

Technische Hochschule Köln

Fakultät für Informatik und Ingenieurwissenschaften

M A S T E R A R B E I T

Evaluation künstlicher neuronaler Netze für
eine Out-of-Stock-Erkennung

Vorgelegt an der TH Köln

Campus Gummersbach

im Studiengang

Wirtschaftsinformatik Verbundstudiengang

ausgearbeitet von:

DENNIS WILLERS

(Matrikelnummer: 11147419)

Erster Prüfer: Prof. Dr. Heide Faeskorn-Woyke

Zweiter Prüfer: Dr. Florian Corzilius

Troisdorf, im Mai 2023

Zusammenfassung

Als Kunde ist es frustrierend in einem Markt vor einem leeren Regal zu stehen. Dem Markt fällt nicht immer auf, dass ein Produkt länger ausverkauft ist, da der gesuchte Artikel im Warenwirtschaftssystem noch mit Bestand gelistet ist. Um dem entgegenzuwirken, soll mithilfe einer Kamera kontrolliert werden, ob eine Out-of-Stock (OOS)-Situation vorliegt.

Diese Masterarbeit evaluiert verschiedene Modellkonfigurationen von Künstlichen-Neuronalen-Netzen (KNNs), um herauszufinden, welche im Markt eine OOS-Situation anhand von Bildern am besten erkennt. Hierfür wurden in sechs Märkten insgesamt 2712 Fotos geschossen, um eine Datenbasis zu schaffen. Die Bilder zeigen eindeutig, ob eine Lücke im Fach eines Regals existiert oder ob das Produkt im Regal vorrätig ist. Basierend auf dem vortrainierten VGG16-Modell von Keras wurden zwei Fully-Connected-Schichten implementiert, in denen sich die 36 betrachteten KNNs hinsichtlich des verwendeten Optimierungsverfahrens und der eingesetzten Aktivierungsfunktionen unterscheiden. Insgesamt wurden im Rahmen der Masterarbeit 216 Modelle generiert, um die Auswirkungen von drei verschiedenen Optimierungsverfahren in Kombination mit zwölf verschiedenen Aktivierungsfunktionspaarungen zu untersuchen. Für die Generierung der Modelle wurde ein nahezu ausgeglichenes Verhältnis von OOS- und In-Stock-Daten verwendet.

Die Evaluation der generierten OOS-Modelle zeigt, dass das Optimierungsverfahren FTRL die schlechtesten Ergebnisse erzielte und daher für diesen Anwendungsfall nicht geeignet ist. Deutlich bessere Ergebnisse erreichen Modellkonfigurationen, die das Optimierungsverfahren Adam oder SGD verwenden. Von den sechs besten Modellkonfigurationen verwenden fünf das Optimierungsverfahren Adam und eine SGD. Diese haben jeweils eine *Accuracy* von mindestens 93% erzielt und konnten den *Recall* für die Klasse OOS zu mindestens 91% vorhersagen.

Da das Datenverhältnis zwischen den OOS- und In-Stock-Daten bei den zuvor generierten Modellen nicht der Realität entspricht, wurden die In-Stock-Bilder augmentiert. Unter Hinzunahme der augmentierten Bilder wurden anschließend für die sechs besten Modellkonfigurationen neue OOS-Modelle generiert. Die Ergebnisse dieser OOS-Modelle weisen keine Konvergenzen auf. Daraus lässt sich ableiten, dass mehr Epochen in der Trainingsphase zu besseren Ergebnissen führen. Dennoch stehen die Ergebnisse des OOS-Modells, welches das Optimierungsverfahren Adam und die Aktivierungsfunktionen Sigmoid und ReLU verwendet, positiv heraus. Es konnte mit einer *Accuracy* von 97,91% und einem *Recall* der Klasse OOS von 87,82% das beste Ergebnis erzielen.

Insgesamt haben mehrere OOS-Modelle das Potenzial sowohl den Umsatz des Marktes als auch die Zufriedenheit der Kunden zu steigern. In einer fortführenden Arbeit sollten die OOS-Modelle im Markt installiert werden, um ihre Leistung unter realen Bedingungen zu bewerten. Die daraus resultierenden Erkenntnisse können für die kontinuierliche Optimierung des Modells verwendet werden.

Abstract

As a customer, it can be frustrating to face an empty shelf in a store. The market does not always realize that a product has been out of stock for a while, as the item is still listed as in stock in the inventory management system. To address this issue, a camera should be used to check for Out-of-Stock (OOS) situations.

This master thesis evaluates different model configurations of Artificial Neural Networks (ANNs) to determine which one best detects OOS situations in the market using images. To create a dataset, 2,712 photos were taken in six stores. The photos clearly show whether there is a gap on the shelf or if the product is in stock. Based on the pre-trained VGG16 model from Keras, two fully connected layers were implemented, with 36 different ANNs differing in the optimization method and activation function pairings. In total, 216 models were generated in this thesis to investigate the effects of three different optimization methods combined with twelve different activation function pairings. An almost balanced ratio of OOS and in-stock data was used to generate these models.

The evaluation of the generated OOS models shows that the FTRL optimization method achieved the least favorable results and is therefore not suitable for this application. Model configurations using the Adam or SGD optimization methods achieve much better results. Of the top six model configurations, five use the Adam optimization method and one uses SGD. They all achieved an accuracy of at least 93% and were able to predict the Recall for the OOS class with at least 91%.


As the data ratio between OOS and in-stock data did not correspond to reality in the previously generated models, the in-stock images were augmented. Including the augmented images, new OOS models were generated for the top six model configurations. The results of these OOS models show no convergences. This suggests that more epochs in the training phase lead to better results. However, the results of the OOS model using the Adam optimization method and the Sigmoid and ReLU activation functions stand out positively. It achieved the best result with an accuracy of 97.91% and a Recall of the OOS class of 87.82%.

Overall, several OOS models have the potential to increase both market sales and customer satisfaction. In a future study, the OOS models should be installed in the market to evaluate their performance under real conditions. The resulting insights can be used for continuous optimization of the model.

Inhaltsverzeichnis

Abkürzungsverzeichnis	VI
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	IX
Listings	X
Disclaimer	XII
1 Einleitung	1
1.1 Ist-Bestand Problematik	2
1.2 Lösungsansatz für die OOS-Erkennung	2
1.3 Vorgehen nach dem CRISP-DM Modell	3
2 Künstliche neuronale Netze	6
2.1 Neuronale Netze in der Biologie	6
2.2 Künstliche Neuronen	8
2.2.1 Definition von künstlichen Neuronen	8
2.2.2 Zusammenspiel zwischen zwei künstlichen Neuronen	8
2.2.3 Topologie von einem KNN	9
2.3 Aktivierungsfunktionen	10
2.3.1 Stufenfunktion	11
2.3.2 Sigmoidfunktion	11
2.3.3 Tangens-Hyperbolicus-Funktion	12
2.3.4 Softmax-Funktion	13
2.3.5 Rectified-Linear-Unit-Funktion	14
2.4 Backpropagation	15
2.5 Optimierungsverfahren	17
2.5.1 SGD	18
2.5.2 Adam	19
2.5.3 FTRL	20
2.6 Deep-Learning-Bibliothek Keras	21
2.6.1 Verwendung von Tensorflow 2 mithilfe der Keras API	22
2.6.2 VGG16-Modell	22
2.7 Klassifizierungsmetriken	23
2.7.1 Confusion-Matrix	24
2.7.2 Accuracy	25
2.7.3 Precision	25

2.7.4	Recall	26
2.7.5	F1-Score	26
2.7.6	Kostenfunktion Binary-Cross-Entropy	28
3	Datengrundlage	29
3.1	Datenbeschaffung	29
3.1.1	Kriterien für die Bilddaten	29
3.1.2	Übersicht der erzeugten In-Stock und OOS-Bilder	30
3.2	Herausforderungen bei den OOS-Daten	31
3.3	Augmentation der Daten	33
3.3.1	Augmentationstypen	33
3.3.2	Trainieren mit augmentierten Bildern	35
3.3.3	Datengrundlage durch das Augmentierten von Bildern erweitern	37
3.4	Aufteilung der Trainings- und Testdaten	38
3.4.1	Holdout-Methode	38
3.4.2	Cross-Validation-Methode	38
3.4.3	Hybride Modellvalidierung	39
4	OOS-Modelle	41
4.1	Rahmenbedingungen für das KNN	41
4.1.1	Hardware-Komponenten	41
4.1.2	Software-Komponenten	42
4.1.3	KNN-Konfiguration	43
4.1.4	KNN-Architektur	44
4.2	Implementierung der OOS-Modellvarianten	46
4.2.1	Implementierung der Aktivierungsfunktion in eine Netz-Schicht	46
4.2.2	Implementierung von Optimierungsverfahren	48
4.3	Implementierung des ML-Prozesses	48
4.3.1	Implementierung des Modelltrainings	49
4.3.2	Implementierung der Modellvalidierung	51
4.4	Generierung von OOS-Modellen	52
4.4.1	Zusammenstellung der Trainingsdaten	52
4.4.2	Validierung der Modellkonfigurationen mittels Cross-Validation	54
4.4.3	Zentrale Methode zur Generierung eines OOS-Modells	56
4.5	Dokumentation der Leistung von OOS-Modellen	57
4.5.1	Protokollieren der Ergebnisse eines OOS-Modells	57
4.5.2	Kompakte Darstellung der Ergebnisse einer Modellkonfiguration	60

5	Evaluation	62
5.1	Bewertung der Metrikergebnisse	62
5.1.1	Kostenfunktionsanalyse der OOS-Modelle	62
5.1.2	Genauigkeit der OOS-Modelle	64
5.1.3	Recall der OOS-Modelle	65
5.1.4	Konvergenzverhalten verschiedener OOS-Modelle	67
5.2	Einfluss der Hardware-Leistung auf die Generierungsdauer	70
5.2.1	Generierungsdauer der OOS-Modelle	70
5.2.2	Hardware-Auslastung	71
5.3	Vergleich der besten Modellkonfigurationen	71
5.3.1	Konvergenzverhalten der besten Modellkonfigurationen	72
5.3.2	Vergleich der Cross-Validierten OOS-Modelle	74
5.3.3	Auswirkung des Datenverhältnisses auf die OOS-Modelle	76
6	Diskussion	80
6.1	Ergebnis	80
6.2	Ausblick	81
	Literatur	XIII
	Anhang	XX
A		XX
B	Cross-Validierung auf Marktebene	XXI
C	Codedokumentationen	XXII
C.1	Docker	XXII
C.1.1	Implementierung Dockerfile	XXII
C.1.2	Implementierung Docker-Compose	XXII
C.2	Augmentation	XXIII
C.2.1	Definierte Parameter für das Augmentieren von Bildern	XXIII
C.2.2	Speichern von augmentierten Bildern	XXIV
C.3	Laden von Daten	XXV
C.3.1	Laden der Testdaten	XXV
C.3.2	Laden der Daten von einem Markt	XXV
C.4	Enums	XXVI
C.4.1	Enum Markt	XXVI
C.4.2	Enum Aktivierungsfunktion	XXVI
C.4.3	Enum Optimierungsverfahren	XXVI
C.5	Klassen	XXVII

C.5.1	Klasse ConfigKNN	XXVII
C.5.2	Klasse CustomCallback	XXVII
C.6	Kommentierte Methoden	XXIX
C.6.1	Kommentierte Version der KNN-Architektur	XXIX
C.6.2	Kommentierte Version der Zusammenstellung von Trainingsdaten	XXX
C.6.3	Kommentierte Version der Validierung von Modellkonfigurationen	XXXI
C.6.4	Kommentierte Version zur Generierung eines OOS-Modells	XXXII
C.7	Generieren der Excel-Tabelle	XXXIV
C.7.1	Initiale Konfiguration einer Excel-Datei	XXXIV
C.7.2	Speichern der Excel-Tabelle	XXXIV
C.7.3	Schreiben der Modellergebnisse in einer Excel-Datei	XXXV
C.7.4	Mittelwert der Modellergebnisse berechnen	XXXVII
C.7.5	Mittelwert der Modellergebnisse dokumentieren	XXXVIII
D	Ergebnisse	XXXIX
D.1	Modellkonfigurationen auf Basis der unveränderten Daten	XXXIX
D.1.1	Adam sigmoid ReLU	XL
D.1.2	Adam sigmoid sigmoid	XLII
D.1.3	Adam sigmoid softmax	XLIV
D.1.4	Adam sigmoid tanh	XLVI
D.1.5	Adam softmax ReLU	XLVIII
D.1.6	Adam softmax sigmoid	L
D.1.7	Adam softmax softmax	LII
D.1.8	Adam softmax tanh	LIV
D.1.9	Adam tanh ReLU	LVI
D.1.10	Adam tanh sigmoid	LVIII
D.1.11	Adam tanh softmax	LX
D.1.12	Adam tanh tanh	LXII
D.1.13	Ftrl sigmoid ReLU	LXIV
D.1.14	Ftrl sigmoid sigmoid	LXVI
D.1.15	Ftrl sigmoid softmax	LXVIII
D.1.16	Ftrl sigmoid tanh	LXX
D.1.17	Ftrl softmax ReLU	LXXII
D.1.18	Ftrl softmax sigmoid	LXXIV
D.1.19	Ftrl softmax softmax	LXXVI
D.1.20	Ftrl softmax tanh	LXXVIII
D.1.21	Ftrl tanh ReLU	LXXX
D.1.22	Ftrl tanh sigmoid	LXXXII
D.1.23	Ftrl tanh softmax	LXXXIV

D.1.24 Ftrl tanh tanh	LXXXVI
D.1.25 SGD sigmoid ReLU	LXXXVIII
D.1.26 SGD sigmoid sigmoid	XC
D.1.27 SGD sigmoid softmax	XCII
D.1.28 SGD sigmoid tanh	XCIV
D.1.29 SGD softmax ReLU	XCVI
D.1.30 SGD softmax sigmoid	XCVIII
D.1.31 SGD softmax softmax	C
D.1.32 SGD softmax tanh	CII
D.1.33 SGD tanh ReLU	CIV
D.1.34 SGD tanh sigmoid	CVI
D.1.35 SGD tanh softmax	CVIII
D.1.36 SGD tanh tanh	CX
D.2 Modellkonfigurationen auf Basis der augmentierten Daten	CXII
D.2.1 Adam sigmoid ReLU	CXII
D.2.2 Adam sigmoid sigmoid	CXIV
D.2.3 Adam sigmoid softmax	CXVI
D.2.4 Adam softmax sigmoid	CXVIII
D.2.5 Adam softmax tanh	CXX
D.2.6 SGD sigmoid ReLU	CXXII

Erklärung über die selbständige Abfassung der Arbeit **CXXIV**

Abkürzungsverzeichnis

Adam Adaptive-Moment-Estimation

AdaGrad Adaptative-Gradient

Adadelta Adaptive-Delta

API Application-Programming-Interface

BCE Binary-Cross-Entropy

BGD Batch-Gradient-Descent

CNN Convolutional-Neural-Network

CPU Central-Processing-Unit

CRISP-DM Cross-Industry-Standard-Process for Data-Mining

CV Computer-Vision

DM Data-Mining

ESL Electronic-Shelf-Labels

Fibu Finanzbuchhaltung

FN False-Negative

FP False-Positive

FTML Follow-The-Moving-Leader

FTRL Follow-The-Regularized-Leader

GPU Graphics-Processing-Unit

KFCV K-Fold Cross-Validation

KI Künstliche-Intelligenz

KNN Künstliche-Neuronale-Netze

Kolo Kolonialwaren

MAE Mean-Absolute-Error

MBGD Mini-Batch-Gradient-Descent

ML Machine-Learning

MNIST Modified-National-Institute-of-Standards-and-Technology

Mopro Molkereiprodukte

MSE Mean-Squared-Error

NPV Negative-Predictive-Values

OG Obst und Gemüse

OGD Online-Gradient-Descent

OOS Out-Of-Stock

PoC Proof-of-Concept

PPV Positive-Predictive-Values

ReLU Rectified-Linear-Unit

RMSProp Root-Mean-Squared-Propagation

Sens Sensitivity

Spec Specificity

SGD Stochastic-Gradient-Descent

TanH Tangens Hyperbolicus

TN True-Negative

TNR True-Negative-Rates


TP True-Positive

TPR True-Positive-Rates

TPU Tensor-Processing-Unit

WaWi Warenwirtschaftssystem

Abbildungsverzeichnis

1	CRISP-DM Vorgehensmodell	4
2	Aufbau eines Neurons	7
3	Künstliches Neuron	8
4	Verbindung zweier künstlicher Neuronen	9
5	Verbindung von künstlichen Neuronen	10
6	Stufenfunktionsgraph	11
7	Sigmoid-Funktionsgraph	12
8	TanH-Funktionsgraph	13
9	ReLU-Funktionsgraph	15
10	Bild ohne OOS-Artikel	29
11	Bild mit OOS-Artikel	29
12	OOS-Varianten	31
13	Regal- und Korbvarianten	32
14	Warenstopper-Varianten	32
15	Horizontal Flip	34
16	Rotation Range	34
17	Random Shifts	34
18	Random Brightness	34
19	Random Zoom	35
20	Alle Parameter	35
21	Cross-Validation mit k=10	39
22	Aufteilung der Märkte in Trainings- und Testdaten	40
23	Quell- und Zielverzeichnisstruktur	52
24	Vergleich der Konvergenz verschiedener Modellkonfigurationen	69
25	Konvergenzen der Top Modellkonfigurationen von Adam	73
26	Kostenfunktion der Top 6 Modellkonfigurationen	74
27	Unterschiede in den Ergebnissen der Cross-Validierung	75
28		XX
29	Cross-Validierung auf Marktebene	XXI

Tabellenverzeichnis

1	Confusion-Matrix	24
2	Anzahl erstellter In-Stock und OOS-Bilder je Markt	30
3	Verwendete Parameter der ImageDataGenerator-Klasse	34
4	Server-Spezifikationen	41
5	Ermittelte Werte der Kostenfunktion Binary-Cross-Entropy	63
6	Accuracy der OOS-Modelle	64
7	Recall der Klasse OOS	65
8	Recall der Klasse In-Stock	66
9	Generierungsdauer der OOS-Modelle	70
10	Evaluation von Modellkonfigurationen auf Basis von augmentierten Daten	77

Listings

1	Laden der Trainingsdaten	36
2	Generieren und Speichern von augmentierten Bildern	37
3	Verwendete Bibliotheken für das OOS-Modell	42
4	Ausschnitt globale Konfigurationseinstellungen	43
5	Ausschnitt der ConfigKNN-Klasse	44
6	Architektur vom KNN	45
7	Generierung der nächsten Output-Schicht	46
8	Ermittlung des aktuellen Optimierungsverfahrens	48
9	Speichern von Informationen während der Trainingsphase	49
10	OOS-Modell trainieren	50
11	OOS-Modell evaluieren	51
12	Zusammenstellung der aktuellen Trainingsdaten	53
13	Iterieren über alle Modellkonfigurationen	54
14	Implementierung der Cross-Validation-Methode	55
15	Hauptmethode zur Generierung eines OOS-Modells	56
16	Ausschnitt vom Schreiben der Modellergebnisse in einer Excel-Tabelle	58
17	Ausschnitt zur Berechnung des Durchschnitts der OOS-Modellergebnisse	60
18	Ausschnitt der Dokumentation der Mittelwertergebnisse von OOS-Modellen	61
19	Dockerfile	XXII
20	docker-compose.yml	XXII
21	Definierte Parameter für das Augmentieren von Bildern	XXIII
22	Speichern von augmentierten Bildern	XXIV
23	Laden der Testdaten	XXV
24	Laden der Marktdaten	XXV
25	Enum Markt	XXVI
26	Enum Aktivierungsfunktion	XXVI
27	Enum Optimierungsverfahren	XXVI
28	Speichern der aktuellen KNN-Konfigurationen	XXVII
29	Custom Callback	XXVII
30	Kommentierte Version der KNN-Architektur	XXIX
31	Kommentierte Version der Zusammenstellung von Trainingsdaten	XXX
32	Kommentierte Version der Validierung von Modellkonfigurationen	XXXI
33	Kommentierte Version zur Generierung eines OOS-Modells	XXXII
34	Erstellen einer Excel-Datei und Festlegen des Arbeitsblattes	XXXIV
35	Speichern der Excel-Datei	XXXIV
36	Schreiben der Modellergebnisse in einer Excel-Tabelle	XXXV
37	Berechnung des Mittelwerts der OOS-Modellergebnisse	XXXVII

38 Dokumentation der Mittelwertergebnisse in einer Excel-Tabelle XXXVIII

Disclaimer

In dieser Arbeit wird darauf geachtet eine neutrale Form zu verwenden. Dies gestaltet sich an einigen Stellen als schwierig, weshalb aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet wurde. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint.

1 Einleitung

„Wenn du etwas so machst, wie du es seit zehn Jahren gemacht hast, dann sind die Chancen groß, dass du es falsch machst.“¹

Obwohl Charles Franklin Kettering in Zeiten der Industrialisierung aufwuchs, ist sein Zitat im heutigen Digitalisierungswandel aktueller denn je. In seiner Zeit veränderten sich Geschäftsprozesse stark, sodass Produkte nun mithilfe von Maschinen anstelle von Hand hergestellt wurden. Als gelernter Ingenieur revolutionierte Kettering den Handel mit der Erfindung der elektrischen Registrierkasse.² Unternehmen, die die elektrische Registrierkasse einführten, konnten ihr Warenwirtschaftssystem (WaWi) und die Finanzbuchhaltung (Fibu) vereinfachen und sich dadurch im Vergleich zur Konkurrenz einen Vorteil verschaffen. Heutzutage ist ein Unternehmen ohne IT-System kaum noch vorstellbar.

Einer der aktuell führenden deutschen und europäischen Handelskonzerne mit rund 380.000 Mitarbeiterinnen und Mitarbeitern ist die REWE Group mit Sitz in Köln. Um das Tagesgeschäft effizient zu gestalten, verfügt die REWE Group über einen großen Bestand von IT-Hardware-Geräten und selbst entwickelten Software-Anwendungen. Dazu zählen unter anderem die 30.000 Scanner-Kassen, welche in den 7.500 Märkten verteilt sind. Die IT-Landschaft wird mit über 1.000 selbst entwickelten Anwendungen immer komplexer. Damit der Markt den Warennachschub termingerecht organisieren kann und somit in den Regalen jederzeit genügend Ware vorhanden ist, werden über 10.000 Datensätze täglich in die Systeme eingespeist.³

Zukünftig konzentrieren sich IT-Prozesse nicht mehr nur auf die Verarbeitung von Informationen, sondern sollen Nutzer bei Entscheidungen unterstützen. Darum nehmen Künstliche-Intelligenz (KI) basierte Systeme vermehrt Einfluss auf die modernen Geschäftsprozesse. Ein Beispiel hierfür ist die marktspezifische Sortimentsplanung. Früher musste die Marktleitung auf ihr Bauchgefühl vertrauen, um zu entscheiden, wie gut und häufig sich ein Artikel verkaufen lässt. Heutzutage kann ein KI-System für jeden Markt individuell prognostizieren, welche Artikel weniger gefragt sind und durch welche Artikel diese ersetzt werden können. Die Marktleitung kann auf Grundlage dieser Informationen entscheiden, ob sie den Artikeltauschvorschlag annehmen möchte. Die erfolgreiche Anwendung dieser Technologie führt zu einer Steigerung des Umsatzes. Dies ist ein Beispiel dafür, dass das Zitat von Kettering nach wie vor zutrifft. Nur wer stetig seine Prozesse optimiert, kann am Markt bestehen.

¹Charles Franklin Kettering (USA, 1876-1958)


²Vgl. Jeffries 1960, S.107






³Vgl. REWE Systems GmbH o.J.

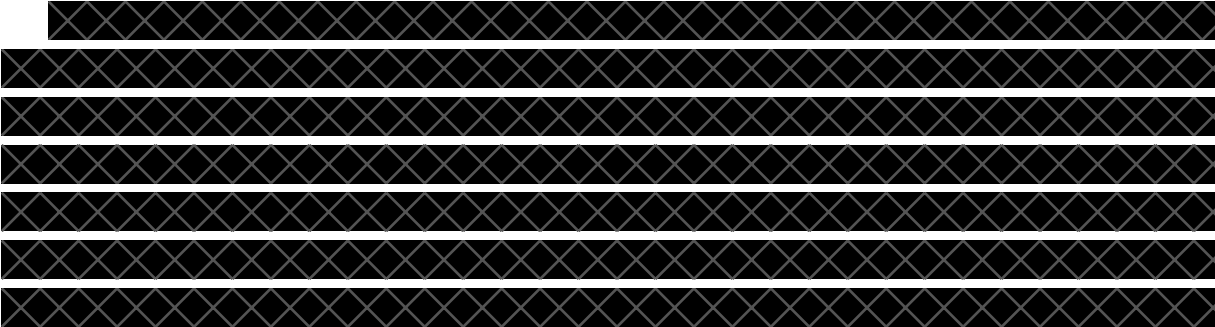
1.1 Ist-Bestand Problematik

Die REWE digital GmbH (ehemals REWE Systems GmbH) ist stetig auf der Suche nach Anwendungsfällen, die mithilfe des Einsatzes von KI basierten Methoden bestehende Marktprozesse optimieren oder neue Marktprozesse erschaffen. Idealerweise kann ein KI-System ein bestehendes Problem lösen. Dafür lohnt sich ein Blick in den Markt. Beim Durchlaufen der Gänge fällt auf, dass viele Lücken im Regalbestand zu finden sind. Für Kunden, die wegen eines bestimmten Artikels in den Markt fahren, kann es besonders ärgerlich sein festzustellen, dass dieser bereits ausverkauft ist. Solche Situationen werden im Handel als Out-Of-Stock (OOS) bezeichnet. In einigen Fällen kann es sogar passieren, dass ein Artikel über einen längeren Zeitraum im Regal nicht verfügbar ist. Laut Aussage der Marktleitung liegt die häufigste Ursache für die Nichtverfügbarkeit von Produkten darin, dass sie im WaWi mit einem Bestand gelistet sind und das Personal daher nicht über den fehlenden Bestand informiert ist. So kann es zu verzögerten Nachbestellungen der jeweiligen Produkte kommen. Die Gründe für eine fehlerhafte Bestandslistung reichen vom Verzählen bei der Abnahme der Ware vom Lieferanten über die fehlerhafte Abrechnung beim Verkauf an der Kasse bis hin zu Diebstahl. Die zahlreichen Möglichkeiten zur Manipulation von Beständen erklären, warum der im WaWi gelistete Bestand nicht immer mit dem tatsächlichen Bestand eines Artikels übereinstimmen. Ein KI-System kann dabei unterstützen diese Diskrepanzen zu identifizieren und so die OOS-Quote im Markt reduzieren.

1.2 Lösungsansatz für die OOS-Erkennung

Um die Ist-Bestandsproblematik zu lösen, kann ein kamerabasiertes KI-System per Bilderkennung feststellen, ob sich eine Lücke im Regal befindet. Wenn eine Lücke erkannt wurde, muss das KI-System diese mit den Daten im WaWi abgleichen. Falls der Artikel im WaWi mit Bestand gelistet ist, muss das KI-System das Personal über die identifizierte Lücke informieren und Handlungsoptionen aufzeigen. Voraussetzung für die OOS-Erkennung ist die Beschaffung des benötigten Echtzeit-Bildmaterials. Hierfür können entweder Kameras in regelmäßigen Abständen an den Regalen angebracht oder ein Roboter mit Kamera in Betrieb genommen werden. Der Roboter fährt dann autonom die Gänge im Markt ab und untersucht die Regale auf Lücken. Hierfür gibt es bereits Anbieter, die sich darauf spezialisiert haben einen solchen Roboter zur Verfügung zu stellen. 



Deshalb forciert die REWE digital GmbH einen eigenen Proof-of-Concept (PoC) für eine OOS-Erkennung. Es ist im ersten Schritt nicht das Ziel einen autonom fahrenden Roboter mit integrierter OOS-Erkennung zu entwickeln. Bevor entsprechende Hardware Komponenten beschafft werden, soll sichergestellt werden, dass eine KI anhand eines Bildes erkennen kann, ob sich im Regal eine Lücke befindet. Eine vorangegangene Projektarbeit zeigte bereits auf Grundlage einer beschränkten Datenbasis, dass ein OOS-Modell gute Ergebnisse bei der Erkennung einer OOS-Situation erzielen kann. Diese Masterarbeit beschäftigt sich mit der Evaluation von OOS-Modellen, die Künstliche-Neuronale-Netze (KNN) als Basis verwenden und sich in ihren eingesetzten Optimierungsverfahren und Aktivierungsfunktionen unterscheiden. Durch das Definieren verschiedener Modellkonfigurationen soll ermittelt werden, welche Einstellung auf Grundlage einer repräsentativen Datenbasis zu den besten Ergebnissen führt. Hierfür eignet sich die Verwendung der Deep-Learning Bibliothek von Keras.

Der Anwendungsfall im Markt fokussiert sich darauf einzelne Fächer in einem Regal als befüllt oder als Regallücke zu klassifizieren. Die Identifizierung des fehlenden Produkts wird nicht betrachtet. Weiterführende Projekte könnten sich mit der Erkennung und dem Scannen von Barcodes an gefundenen OOS-Lücken beschäftigen. Auf Basis der genannten Rahmenbedingungen untersucht diese Masterarbeit folgende Forschungsfrage:

Welche Kombinationen von Optimierungsverfahren und Aktivierungsfunktionen sind am besten geeignet, um mithilfe von künstlichen neuronalen Netzen Modelle für eine Out-Of-Stock-Erkennung im Markt zu generieren?

1.3 Vorgehen nach dem CRISP-DM Modell

Um diese Forschungsfrage zu beantworten, geht diese Masterarbeit nach dem Cross-Industry-Standard-Process for Data-Mining (CRISP-DM)-Modell vor. Dieses Vorgehensmodell entstand aus dem Bedürfnis Daten in einen richtigen logischen Zusammenhang zu bringen. Vor der Einführung der Methodik führten die unterschiedlichen Kenntnisse und Fähigkeiten der verschiedenen Projektteams dazu, dass die Vorgehensweise je nach Data-Mining (DM)-Projekt variiert. Um erfolgreiche Praktiken unternehmensweit zu etablieren,

muss ein Standard für das Vorgehen definiert werden.⁵ Entwickelt wurde der Standard CRISP-DM von den Konzernen IBM und Daimler. Die nachfolgende Abbildung 1 zeigt das Schaubild zum CRISP-DM-Modell.

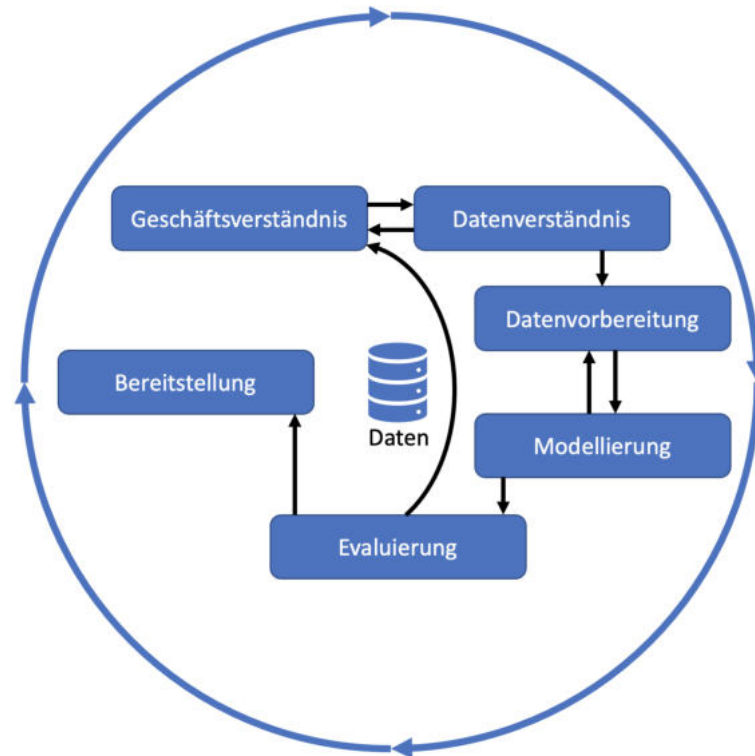


Abbildung 1: CRISP-DM Vorgehensmodell⁶

Heutzutage ist das CRISP-DM-Modell der am häufigsten verwendete Prozessstandard im Bereich der Auswertung von KI-basierten Daten.⁷ Das Modell ist in sechs Phasen unterteilt. Da Projekte seltenst linear verlaufen, sind Rücksprünge in vorherige Phasen erlaubt, um Änderungen oder Anpassungen vorzunehmen. Rücksprünge werden durch die inneren Pfeile gekennzeichnet. Im Gegensatz dazu symbolisieren die äußeren Pfeile den zyklischen Charakter des CRISP-DM-Modells. Um ein besseres Verständnis für die Abhängigkeiten zwischen den Phasen zu erlangen, geht Kapitel 2 auf die theoretischen Grundlagen von KNNs ein und beschreibt wie diese in der Deep-Learning-Bibliothek von Keras genutzt werden.

In der ersten Phase *Geschäftsverständnis* geht es um die Definition der Projektziele. Im Rahmen dieser Masterarbeit ist dies die Erkennung einer OOS-Situation auf Basis von Bildern. Die zweite Phase *Datenverständnis* umfasst die erstmalige Sichtung der Daten. Dabei werden die Daten hinsichtlich ihrer Verfügbarkeit und ihrer Qualität untersucht. Anschließend werden in der dritten Phase *Datenvorbereitung* die Daten bereinigt und als

⁵Vgl. Wirth und Hipp 2000, S. 1 f.

⁶Vgl. IBM 2021

⁷Vgl. Matzka 2021, S. 35

Test- und Trainingsdaten für das Modell bereitgestellt. Die Vorbereitung der Daten ist die aufwendigste und bedeutsamste Phase im Projekt, da sie für die Qualität des Modells ausschlaggebend ist. Kapitel 3 erläutert wie die Schritte zwei und drei des CRISP-DM-Modells auf die OOS- und In-Stock-Daten angewendet wurden.

Anschließend findet in der vierten Phase *Modellierung* das Implementieren und Bauen der KI-Modelle statt. Die Implementierung zu den OOS-Modellen ist in Kapitel 4 beschrieben. Erst in der fünften Phase *Evaluierung* findet eine kritische Betrachtung der Ergebnisse der Modelle statt. Folglich werden die Ergebnisse des OOS-Modelle in Kapitel 5 zusammengetragen und evaluiert. Ist das Unternehmen mit der Qualität eines Modells zufrieden, wird in die sechste Phase *Bereitstellung* übergegangen. Hierzu diskutiert Kapitel 6 abschließend das Potenzial der OOS-Modelle und erörtert die nächsten Schritte, die für den produktiven Einsatz einer OOS-Erkennung erforderlich sind.

2 Künstliche neuronale Netze

Die Grundidee der objektorientierten Programmierung basiert auf der strukturellen Nachbildung der Wirklichkeit. Ähnlich verhält es sich mit KNN. Als Vorbild der neuronalen Netze dient die Biologie. Auf die biologischen Grundlagen der neuronalen Netze geht Kapitel 2.1 ein. Den Schwenk in die Informatik mithilfe von künstlichen Neuronen beschreibt Kapitel 2.2. Wann ein künstliches Neuron in einem KNN aktiv ist, definieren die in Kapitel 2.3 beschriebenen Aktivierungsfunktionen.

Die Lernfähigkeit eines neuronalen Netzes basiert auf der Fehlerminimierung. Mithilfe des in Kapitel 2.4 beschriebenen Backpropagation Verfahrens lernt das KNN die Fehlerrate so gering wie möglich zu halten. Aufbauend darauf wurden die in Kapitel 2.5 beschriebenen Optimierungsverfahren entwickelt, um die Effektivität des Lernprozesses weiter zu steigern. Diese theoretischen Aspekte nutzt die in Kapitel 2.6 beschriebene Deep-Learning-Bibliothek Keras, die Verwendung in der praktischen Ausarbeitung der OOS-Erkennung findet. Abschließend beschreibt Kapitel 2.7 nach welchen Bewertungskriterien ein KNN analysiert werden kann.

2.1 Neuronale Netze in der Biologie

Das menschliche Gehirn besteht aus etwa 86 Milliarden Nervenzellen. Alleine Neuronen stellen 69 Milliarden dieser Nervenzellen dar, die sich im Cerebellum, dem sogenannten Kleinhirn befinden.⁸ Zuständig ist dieser Bereich des Gehirns für das Erlernen von Bewegungsabläufen. Das bedeutet nicht, dass Menschen, die ohne Kleinhirn geboren werden nicht fähig sind zentrale Bewegungsabläufe zu erlernen. Eine Untersuchung ergab, dass ein Mensch ohne Kleinhirn bemerkenswert wenig Defizite aufweist. Das bedeutet, dass das Gehirn es mit den verbliebenen 17 Milliarden Zellen geschafft hat, sich strukturell an die veränderte Umweltbedingung anzupassen. Dieses Beispiel zeigt die besonders hohe Kapazität und Plastizität eines Gehirns.⁹

Die Vielzahl an Netzwerken mit unterschiedlichsten Funktionen im Gehirn lassen sich in bestimmte Areale abbilden, da sie bei allen Menschen ähnlich ausgeprägt sind. Hierbei wird von neuronalen Karten gesprochen. Beispielsweise zeigt die corticale Karte die Areale im Gehirn, die für die menschlichen Sinnesreize zuständig sind.¹⁰ Die verschiedenen neuronalen Netzwerke im Gehirn sind unterschiedlich stark entwickelt. Vergleichbar ist dies mit dem Schienennetzverkehr in Deutschland. Große Städte wie Köln, Frankfurt, Berlin oder München sind Knotenpunkte, die als *hubs* bezeichnet werden. Um den Knotenpunkt herum existieren viele Strecken. Dementsprechend hoch ist die Anzahl der Züge, die in

⁸Vgl. Herculano-Houzel 2009 S. 7

⁹Vgl. Yu u. a. 2015

¹⁰Vgl. Luhmann 2020 S. 26

diese Knotenpunkte hinein und wieder hinaus fahren. Jedoch ist nicht jeder Zug auf jeder Strecke gleich schnell unterwegs. Besteht der Bedarf möglichst schnell zwischen zwei *hubs* zu reisen, ist eine direkte Verbindung von Vorteil. So entstehen Hochgeschwindigkeitsstrecken wie die Verbindung zwischen Köln und Frankfurt. Diese werden als *rich club* bezeichnet, da zwei *hubs* eine schnelle Verbindung zueinander aufgebaut haben. Solche *hubs* und *rich clubs* sind auch im Gehirn zu finden.¹¹

Ein tieferer Blick in ein solches Netzwerk zeigt eine Vielzahl von Neuronen, die miteinander verknüpft sind. Neuronen bestehen aus Ein- und Ausgabeleitungen. Zuständig für die Eingabesignale sind die *Dendriten*. Über elektrochemische Reize empfangen sie die Eingabesignale, die sie an den Prozessor, das *Neuron*, herantragen. Ein Neuron hat durchschnittlich 10.000 Eingabeleitungen. Verstärkt oder gehemmt wird das Signal auf einer Eingabeleitung durch eine *Synapse*. Diese ist als Zelle zwischen einer Dendrite und den Prozessor geschaltet. Aktiv wird das Neuron erst, wenn das elektrische Potential einen Schwellwert überschreitet. Nach Aktivierung des Neurons sendet es über eine einzelne Ausgabeleitung, dem sogenannten *Axon*, einen kurzen elektrischen Impuls. Die Ausgabeleitung kann sich wiederum in verschiedene Zweige aufteilen.¹² Den Aufbau eines Neurons veranschaulicht Abbildung 2.

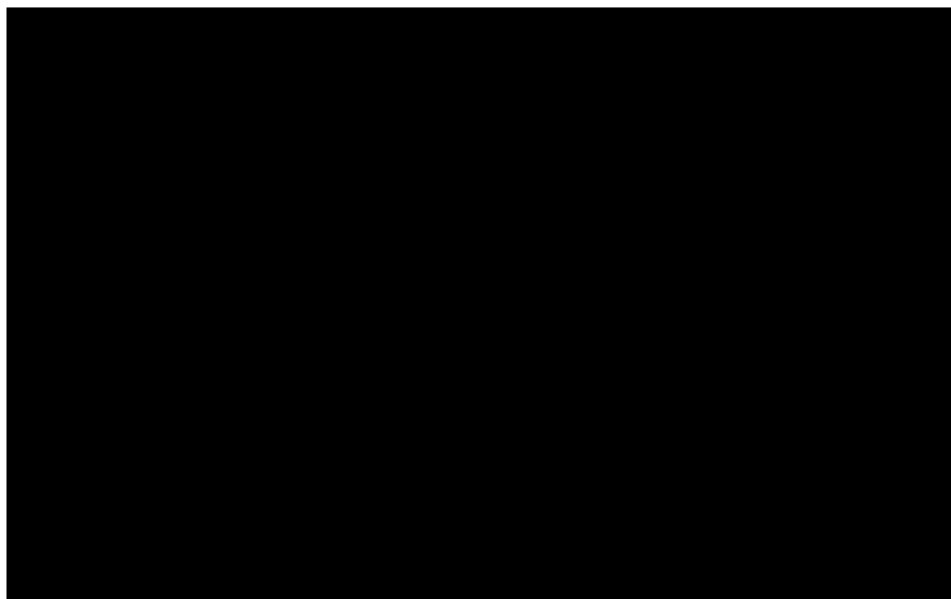


Abbildung 2: Aufbau eines Neurons (a=Axon, d=Dendriten, s=Synapsen)¹³

Den elektrischen Impuls von einem *Axon* empfangen wiederum die daran angeknüpften Neuronen. Sie verarbeiten den elektrischen Impuls weiter. Je mehr Neuronen miteinander verknüpft sind, desto größer ist das neuronale Netzwerk.

¹¹Vgl. Luhmann 2020 S. 50

¹²Vgl. Kinnebrock 2018, S. 14 f.

¹³Kinnebrock 2018, S. 15

2.2 Künstliche Neuronen

Dieses Kapitel skaliert in jedem Unterkapitel die Anzahl der betrachteten künstlichen Neuronen. Zuerst beschreibt Kapitel 2.2.1 die Funktionsweise eines einzelnen künstlichen Neurons. Anschließend geht Kapitel 2.2.2 auf das Zusammenspiel zweier künstlicher Neuronen ein. Abschließend beschreibt Kapitel 2.2.3 die Topologie vieler künstlicher Neuronen in einem KNN.

2.2.1 Definition von künstlichen Neuronen

Um ein Neuron in der Informatik abbilden zu können, müssen die dazugehörigen Bestandteile definiert werden. Ein künstliches Neuron besteht aus $x_1 \dots x_n$ Eingangssignalen mit den dazugehörigen Gewichtungsfaktoren $w_1 \dots w_n$, einer Aktivierungsfunktion f und einem Ausgangssignal y . Ein Beispiel für ein künstliches Neuron ist in Abbildung 3 dargestellt.

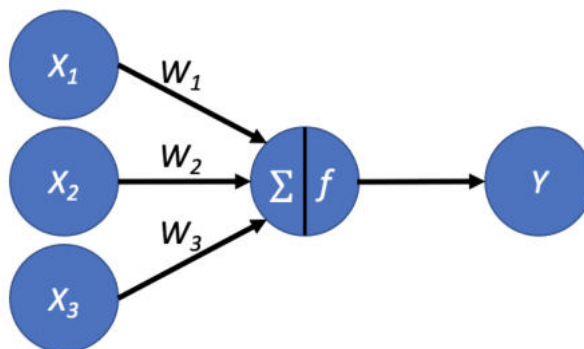


Abbildung 3: Künstliches Neuron¹⁴

Aus der Biologie ist bekannt, dass ein Neuron entweder passiv ist oder nach Aktivierung einen kurzen elektrischen Impuls feuert. Folglich können diese zwei Zustände binär definiert werden. In welchen Zustand die Neuronen geschaltet werden, hängt von den Eingangssignalen ab. Da das Lernverhalten von den Synapsen abhängt, können diese in der Informatik durch negative oder positive Multiplikatoren simuliert werden.¹⁵ Die Summe aller Eingabesignale ergibt einen Wert. Anschließend überprüft die Aktivierungsfunktion, ob das Neuron bei dem summierten Wert aktiv wird. Kapitel 2.3 geht näher auf die Aktivierungsfunktionen ein.

2.2.2 Zusammenspiel zwischen zwei künstlichen Neuronen

Der Austausch von Daten zwischen den Neuronen erfolgt über Schnittstellen. Das resultierende Ergebnis y eines Neurons wird an m verbundene Systemelemente weitergeleitet.

¹⁴Wenighofer und Galler 2018, S. 518

¹⁵Vgl. Kinnebrock 2018, S. 15

Folglich kann das Ausgangssignal eines Neurons u_1 auch das Eingangssignal von Neuron u_2 sein. Ein Beispiel dafür zeigt die nachfolgende Abbildung 4.

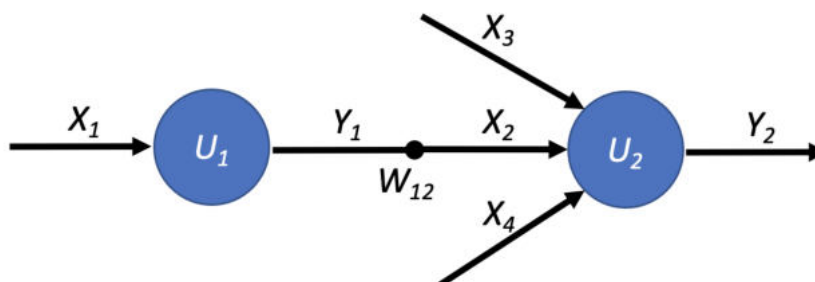


Abbildung 4: Verbindung zweier künstlicher Neuronen¹⁶

Die in der Biologie beschriebenen *hubs* und *rich clubs* werden auch bei künstlichen Neuronen berücksichtigt. Wie stark die Verbindung zwischen zwei Neuronen ist, bestimmt der als numerischer Wert gekennzeichnete Gewichtungsfaktor. Dieser wird für die Multiplikation mit dem Eingangssignal verwendet. Am Beispiel der Neuronen u_1 und u_2 aus Abbildung 4 lässt sich deren Gewichtungsfaktor als w_{12} definieren. Die Zahl 12 setzt sich zusammen aus dem Ausgangssignal y_1 von Neuron u_1 und dem Eingangssignal x_2 von Neuron u_2 .¹⁷ Bei einem positiven Vorzeichen hat die Verbindung zwischen den Neuronen eine verstärkende Wirkung, wohingegen bei einem negativen Vorzeichen die Wirkung hemmend ist.¹⁸ Ist die Verbindungsgewichtung $w_{ij}=0$ erfolgt eine Multiplikation mit 0, was mit einer nicht existierenden Verbindung zwischen den zwei Neuronen gleichzusetzen ist.¹⁹

2.2.3 Topologie von einem KNN

Wie bei einem Gehirn, gibt es auch bei künstlichen Neuronen Verkettungen. Darum steht in einem KNN jedes Neuron n_j in Verbindung zu seinen Vorgängerneuronen $n_1 \dots n_i$ in Abhängigkeit von den zugeordneten Gewichtungen w_{ij} .²⁰ Zur Veranschaulichung werden die Neuronen in verschiedene Schichten unterteilt, bei denen die Neuronen mit der jeweils vorhergehenden und nachfolgenden Schicht verbunden sind.²¹ Die Topologie eines Netzwerks lässt sich wie in Abbildung 5 dargestellt in die Schichten *Input-Schicht*, *Hidden-Schicht* und *Output-Schicht* unterteilen.

¹⁶Vgl. Crone 2010, S. 167

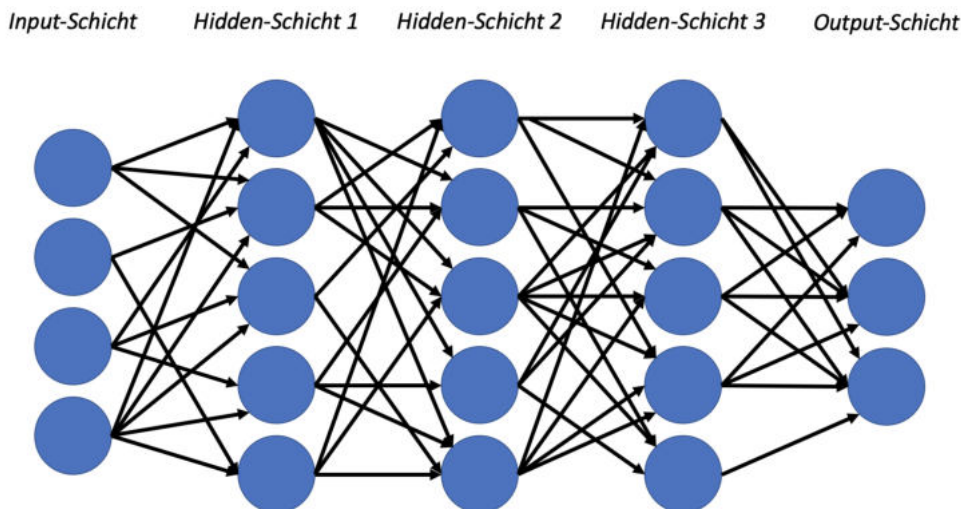
¹⁷Vgl. Zell 2000, S. 70

¹⁸Vgl. Zell 2000, S. 46

¹⁹Vgl. Rashid und Langenau 2017 S. 38

²⁰Vgl. Scherer 1997, S. 47

²¹Vgl. Rashid und Langenau 2017 S. 36

Abbildung 5: Verbindung von künstlichen Neuronen²²

Die *Input-Schicht* besteht aus den eingehenden Daten. Im Kontext der OOS-Erkennung sind es die Regalbilder. Die *Hidden-Schichten* sind verborgene Schichten, da sie keine Verbindung zur Umwelt haben. Es ist von außen schwer nachzuvollziehen wie eine *Hidden-Schicht* aufgebaut ist und aus welchen Gründen es zu dem entwickelten KNN kommt.²³ In der *Output-Schicht* wird das Ergebnis ausgegeben. Bei der OOS-Erkennung steht das Ergebnis z.B. dafür, ob das Bild eine Regallücke darstellt oder nicht.

2.3 Aktivierungsfunktionen

Ein Neuron benötigt eine Aktivierungsfunktion, um das Eingangssignal zu verarbeiten und ab einem gewissen Schwellwert ein Ausgangssignal zu senden. Die Mathematik stellt im Kontext von KNN mehrere solcher Aktivierungsfunktionen zur Verfügung. Am Beispiel der Stufenfunktion wird die Funktionsweise einer Aktivierungsfunktion in Kapitel 2.3.1 erläutert.

Es gibt in der Theorie keine eindeutige Beschreibung, nach welchen Kriterien Aktivierungsfunktionen ausgewählt werden sollten.²⁴ Deshalb untersucht diese Arbeit inwiefern die Auswahl der Aktivierungsfunktion Einfluss auf die Leistung eines KNNs hat. Der Fokus liegt dabei auf den traditionell am häufigsten verwendeten Aktivierungsfunktionen. Dazu zählen die in Kapitel 2.3.2 beschriebene Sigmoidfunktion, die in Kapitel 2.3.3 erläuterte Tangens Hyperbolicus (TanH) Funktion, die in Kapitel 2.3.4 geschilderte Softmax Funktion und schlussendlich die in Kapitel 2.3.5 ausgeführte Rectified-Linear-Unit (ReLU)-Funktion.^{25 26}

²²Wenighofer und Galler 2018, S. 518

²³Vgl. Rauscher 2004 S. 41

²⁴Vgl. Nielsen 2015, S. 124

²⁵Vgl. Zhu und Chen 2020, S. 2 f.

²⁶Vgl. Goodfellow, Bengio und Courville 2016, S. 180 ff.

2.3.1 Stufenfunktion

Die einfachste Aktivierungsfunktion ist die in der nachfolgenden Abbildung 6 dargestellte *Stufenfunktion*.

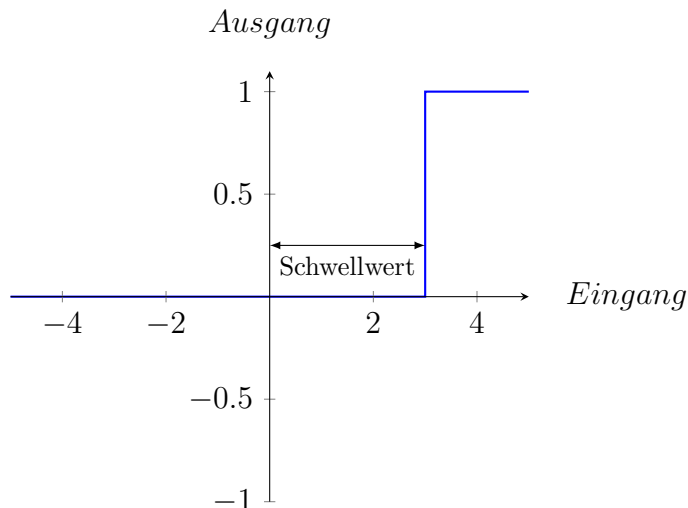


Abbildung 6: Stufenfunktionsgraph²⁷

Die Abbildung zeigt, dass bei niedrigen Eingabewerten die Ausgabe Null beträgt. Erst ab dem Schwellwert $x = 3$ gibt die Funktion den positiven Ausgabewert Eins zurück.

Die Stufenfunktion sollte nicht mit der binären Schwellwertfunktion verwechselt werden. Während der Schwellwert bei der Stufenfunktion nicht eindeutig festgelegt und somit individuell gewählt werden kann, liegt er bei der binären Schwellwertfunktion bei $x = 0$.

Ein weiterer Unterschied ist der definierte Wertebereich. Die Stufenfunktion aus Abbildung 6 nimmt binär die Werte $\{0; 1\}$ an. Im Gegensatz dazu kann der Wertebereich bei der binären Stufenfunktion auch bei $\{-1; 1\}$ liegen.²⁸ Welche Auswirkungen der definierte Wertebereich hat erläutert Kapitel 2.4.

2.3.2 Sigmoidfunktion

Statt einer binären Ausgabe auf Grundlage der Stufenfunktion, können mithilfe einer sanfteren Aktivierungsfunktion kontinuierliche Ausgabewerte erzielt werden. Hierfür eignet sich die in Abbildung 7 dargestellte *Sigmoidfunktion*,

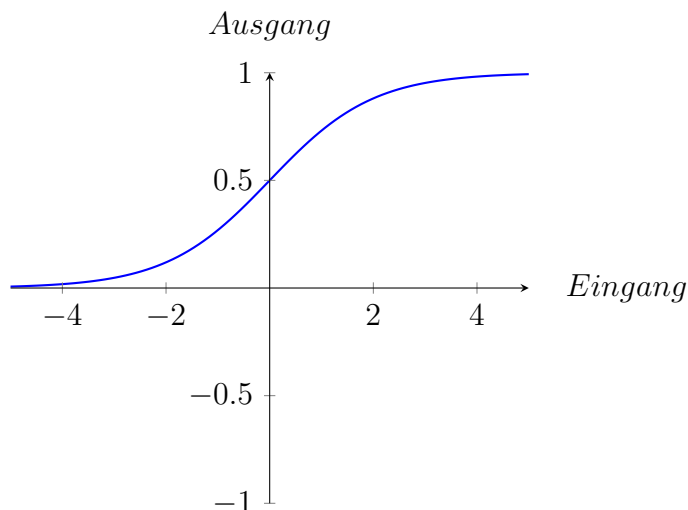
$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

welche eine einfache, aber gebräuchliche Aktivierungsfunktion von KNNs ist. Sie wird auch als logistische Funktion bezeichnet.²⁹

²⁷Rashid und Langenau 2017, S. 33

²⁸Vgl. Crone 2010, S. 172

²⁹Vgl. Rashid und Langenau 2017 S. 33 f.

Abbildung 7: Sigmoid-Funktionsgraph³⁰

Das e in Formel 1 steht für die Eulersche Zahl. Der große Vorteil der *Sigmoidfunktion* liegt in der einfachen Berechenbarkeit im Vergleich zu anderen s-förmigen Funktionen. Die *Sigmoidfunktion* nähert sich den Endzuständen stetig an und hat daher einen Wertebereich von $]0, 1[$.³¹

2.3.3 Tangens-Hyperbolicus-Funktion

Eine weitere Möglichkeit ist die in Abbildung 8 dargestellte Aktivierungsfunktion *TanH*,

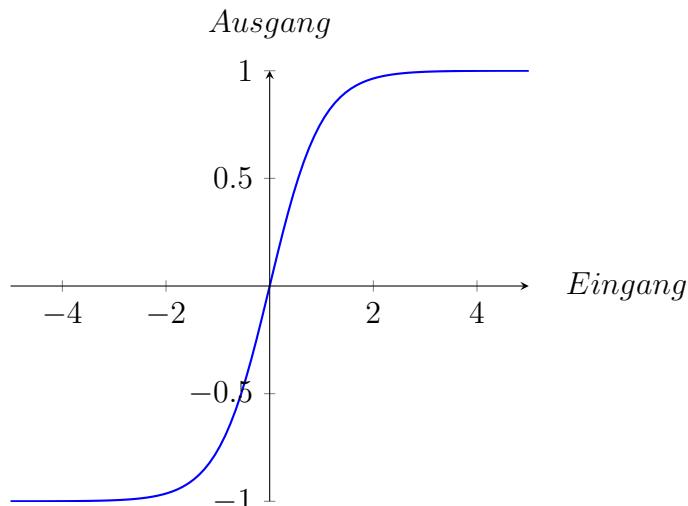
$$\text{TanH}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

welche die Zustände ebenfalls nur näherungsweise erreichen kann. Der Wertebereich liegt jedoch zwischen $] - 1, 1[$.³²

³⁰Rashid und Langenau 2017, S. 34

³¹Vgl. Rashid und Langenau 2017, S. 34 f.

³²Vgl. Crone 2010, S. 172

Abbildung 8: TanH-Funktionsgraph³³

Im Gegensatz zur *Sigmoidfunktion* sättigt die *TanH-Funktion* nicht so stark. Dennoch sind die *Sigmoidfunktion* und der *TanH* nah miteinander verwandt, sodass die Formel auch als

$$\text{TanH}(x) = 2 \times \text{Sigmoid}(2x) - 1 \quad (3)$$

geschrieben werden kann. Aufgrund der nicht so starken Sättigung der *TanH-Funktion* wird diese gegenüber der *Sigmoidfunktion* in der *Hidden-Schicht* bevorzugt. Eine Sättigung nahe Null erschwert nämlich das gradientenbasierte Lernen.³⁴ In der Regel konvergiert ein KNN besser, wenn der Mittelwert der Aktivierungsfunktion nahe 0 ist. Dies ist beim *TanH* der Fall, während der Mittelwert der *Sigmoidfunktion* bei $\frac{1}{2}$ liegt. Darum eignet sich der *TanH* besonders bei großen Merkmalsunterschieden, weil die Gewichte mit weniger Iterationszeiten und folglich einer schnelleren Konvergenz angepasst werden. Je länger ein neuronales Netz trainiert wird, desto besser kann es die Unterschiede zwischen den verschiedenen Merkmalen in den Eingabedaten erkennen und sie voneinander differenzieren.³⁵

2.3.4 Softmax-Funktion

Eine weitere wichtige Ausgabefunktion für KNNs ist die *Softmax-Funktion*. In der *Output-Schicht* wird sie im Vergleich zur *Hidden-Schicht* häufiger eingesetzt, um die Wahrschein-

³³Vgl. Crone 2010, S. 172

³⁴Vgl. Goodfellow, Bengio und Courville 2016, S. 191

³⁵Vgl. Zhu und Chen 2020, S. 2 f.

lichkeitsverteilung über n Klassen zu bestimmen.³⁶ Formal ist die Funktion als

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (4)$$

definiert. Die Formel 4 berechnet die Wahrscheinlichkeit einer Klasse i für den Eingabewert x .³⁷ Werden die Wahrscheinlichkeiten zu einer Eingabe x für alle Klassen i ermittelt, ergibt die Summe aller berechneten Wahrscheinlichkeiten Eins, da

$$\sum_i \text{Softmax}(x_i) = \sum_i \frac{e^{x_i}}{\sum_j e^{x_j}} = \frac{\sum_i e^{x_i}}{\sum_j e^{x_j}} = 1 \quad (5)$$

gilt.³⁸ Die Ausgabewerte können aufgrund der exponentiellen Funktion e nur positiv sein, wodurch der Wertebereich zwischen $[0, 1]$ liegt. Obwohl die *Softmax-Funktion* häufig in der Multiklassen-Klassifizierung angewendet wird, kann sie auch in einer binären Klassifikation genutzt werden. Am Beispiel eines zweidimensionalen Softmax-Vektors

$$\text{Softmax}([1 \ 2]) = \left[\frac{e^1}{e^1+e^2} \quad \frac{e^2}{e^1+e^2} \right] = [0.2689 \ 0.7311] \quad (6)$$

können die Wahrscheinlichkeiten für die binären Klassen berechnet werden.

2.3.5 Rectified-Linear-Unit-Funktion

Als alternative zum *TanH* bietet sich die *ReLU*-Aktivierungsfunktion an. Sie hat sich zu einem Standard für die meisten KNNs entwickelt.³⁹ Durch ihre einfache Formel

$$f(x) = \max(0, x) \quad (7)$$

können KNNs um ein vielfaches schneller trainieren als mit der *TanH*-Aktivierungsfunktion.⁴⁰ Bei der Formel 7 handelt es sich um eine stückweise lineare Funktion, die aus zwei linearen Teilen besteht. Veranschaulicht wird die *ReLU*-Funktion durch die folgende Abbildung 9.

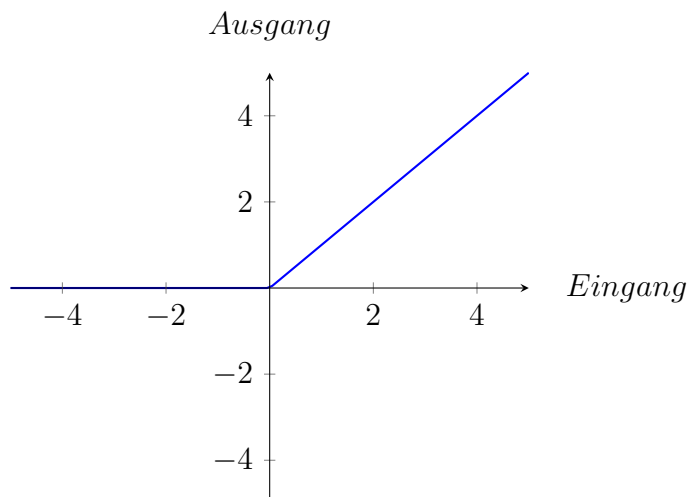
³⁶Vgl. Goodfellow, Bengio und Courville 2016, S. 180 f.

³⁷Vgl. Goodfellow, Bengio und Courville 2016, S. 181

³⁸Vgl. Nielsen 2015, S. 70 ff.

³⁹Vgl. Goodfellow, Bengio und Courville 2016, S. 170

⁴⁰Vgl. Krizhevsky, Sutskever und Hinton 2012, S. 3

Abbildung 9: ReLU-Funktionsgraph⁴¹

Die Abbildung zeigt, dass bei negativen Eingabewerten die Ausgabe der Funktion Null beträgt, während bei positiven Werten der Ausgangswert der Eingabe entspricht. Folglich liegt der Wertebereich bei $[0, \infty]$.⁴² Aufgrund der linearen Eigenschaften, die die *ReLU*-Funktion mitbringt, funktioniert die Optimierung linearer Modelle mithilfe von gradientenbasierten Methoden gut.⁴³

Da eine binäre Klassifizierung nur Werte zwischen Null und Eins erlaubt, ist die *ReLU*-Aktivierungsfunktion nicht für die *Output-Schicht* geeignet. Diese Arbeit betrachtet daher die Leistungsfähigkeit von *ReLU* ausschließlich in der *Hidden-Schicht*, in der sie am häufigsten eingesetzt wird. Dies wird durch frühere Arbeiten zur Bilderkennung gestützt, bei denen ReLU als wichtiger Faktor zur Verbesserung der Leistung identifiziert wurde.⁴⁴

2.4 Backpropagation

Das Backpropagation-Verfahren ist ein Lernverfahren mit dem Ziel das KNN zu optimieren. Es handelt sich hierbei um eine Fehlerrückführungsmethode, welche bei mehrschichtigen Netzen zum Einsatz kommt.⁴⁵ Sie gehört zur Gruppe der überwachten Lernverfahren.⁴⁶ Bei dieser Art von Lernverfahren bestehen die Trainingsdaten aus jeweils einem Input-Wert und dem zugehörigen Output-Wert. Das ermittelte Ergebnis wird mit dem tatsächlichen Output-Wert verglichen.⁴⁷

Am Beispiel eines zweischichtigen Netzes lässt sich die Backpropagation-Methode verdeutlichen. Der erste Schritt ist eine zufällige Bestimmung der Gewichtungsfaktoren. Danach

⁴¹Gabbay und Gabbay 2022, S. 181

⁴²Vgl. Banerjee, Mukherjee und Pasilliao 2020, S. 239

⁴³Vgl. Goodfellow, Bengio und Courville 2016, S. 170

⁴⁴Vgl. Jarrett u. a. 2009, S. 2152

⁴⁵Vgl. Kinnebrock 2018, S. 39

⁴⁶Vgl. Crone 2010, S. 198

⁴⁷Vgl. Zell 2000, S. 93

wird initial vorwärts durch das Netz propagiert. Nun gilt es die Ist-Werte mit den Soll-Werten zu vergleichen. Seien z_i die vorgegebenen Soll-Werte und a_i die Ist-Werte, dann kann die Fehlerrate mit folgender Fehlerfunktion ermittelt werden:

$$E = \frac{1}{2} \cdot \sum_i (z_i - a_i)^2 \quad (8)$$

Je näher E gegen 0 geht, desto exakter arbeitet das Netz.⁴⁸ Nun geht es darum den Fehler von der Output-Schicht bis hin zur Input-Schicht zurück zu propagieren. Sei w_{ij}^1 der Gewichtungsfaktor zwischen der Output-Schicht a und der Hidden-Schicht h und sei w_{ij}^2 der Gewichtungsfaktor zwischen h und der Input-Schicht e . Des Weiteren seien e_i die Input-Werte und h_j die Hidden-Werte. Nun gilt es den Fehler eines Wertes zu definieren mit

$$\epsilon_i = z_i - a_i = \text{OutputWert} - \text{InputWert} = \text{Fehler} \quad (9)$$

Zum Schluss muss ein Lernfaktor α mit $\alpha > 0$ frei gewählt werden. Dann gilt:

$$\Delta w_{ij}^1 = \alpha \cdot \epsilon_i \cdot h_j \quad (10)$$

$$\Delta w_{ij}^2 = \alpha \cdot \sum_m e_m \cdot w_{mi}^1 \cdot e_j \quad (11)$$

Durch das Anpassen der Gewichtungsfaktoren verkleinert sich das Ergebnis der Fehlerfunktion E . Folglich verändert sich das Netz in Richtung eines lokalen Minimums.⁴⁹ Die Formeln haben den Nachteil, dass Gewichtsveränderungen sich auch bei Gewichtungsfaktoren auswirken, die ihrem Zielwert Eins oder Null nahe sind. Um dem entgegenzuwirken, können die Formeln wie folgt präzisiert werden:

$$\Delta w_{ij}^1 = \alpha \cdot \epsilon_i \cdot a_i \cdot (1 - a_i) \cdot h_j \quad (12)$$

$$\Delta w_{ij}^2 = \alpha \cdot \sum_m \epsilon_m \cdot a_m \cdot (1 - a_m) \cdot w_{mi}^1 \cdot h_j \cdot (1 - h_j) \cdot e_j \quad (13)$$

Eine starke Gewichtsveränderung erfolgt bei indifferenten Werten wie 0,5.⁵⁰ Bei Aktivierungsfunktionen wie dem *TanH* wird das Lernen beschleunigt, da bei Nichtaktivierung des Vorgängerneurons u_i mit dem Zustand -1 die Verbindung zwischen den Neuronen mit dem Gewichtungsfaktor w_{ij} aufgrund der Lernregel des Backpropagationverfahrens reduziert wird. Wenn der Zustand der Aktivierungsfunktion 0 ist, hat dies als Produkt in der Lernregel keine Auswirkung auf den Gewichtungsfaktor. Dies kommt z.B. bei der *Sigmoidfunktion* vor.⁵¹ Mithilfe der genannten Definitionen und Formeln lässt sich der Backpropagation Lernalgorithmus wie folgt beschreiben:

⁴⁸Vgl. Kinnebrock 2018, S. 40

⁴⁹Vgl. Kinnebrock 2018, S. 40

⁵⁰Vgl. Kinnebrock 2018, S. 41

⁵¹Vgl. Zell 2000, S. 92

1. Initiale Bestimmung der Gewichte mit Zufallszahlen.
2. Zufällige Vorgabe von Input- und Output-Parametern für die zu lernende Funktion und Berechnung der Belegung h_j in der Hidden-Schicht.
3. Korrektur der Gewichte auf Basis der vorgegebenen Input-Werte e_i und Output-Werte z_i mithilfe der Formeln 10 und 11 oder 12 und 13.
4. Weiteren Lernschritt ab Punkt 2 durchführen oder Lernalgorithmus beenden⁵²

Die Anzahl der Lernschritte ist in der Regel hoch, weshalb das Entwickeln eines KNN rechenintensiv ist. Zur Reduktion der Rechenzeit empfiehlt es sich nicht zu hohe initiale Gewichtungsfaktoren zu bestimmen. Gleiches gilt für die Anzahl der Neuronen in der Hidden-Schicht. Wird sie zu groß gewählt, steigt die Rechenzeit. Wird sie zu niedrig gewählt, kann die Funktion mangels Kapazitäten nicht dargestellt werden. Da es sowohl für die Anzahl der Schichten als auch für die Anzahl der Neuronen pro Schicht für die meisten Anwendungsfälle keine theoretischen Vorgaben gibt, hilft lediglich ein experimentelles Vorgehen.⁵³

2.5 Optimierungsverfahren

Die meisten Optimierungsverfahren basieren auf dem Backpropagation-Algorithmus. Dabei handelt es sich um mathematische Algorithmen, die beim maschinellen Lernen verwendet werden, um Modelle zu trainieren und die verwendeten Parameter zu optimieren. Um die Auswirkungen von Optimierungsverfahren auf die Leistung des OOS-Modells zu untersuchen, werden im Rahmen dieser Masterarbeit drei grundlegend verschiedene Optimierungsverfahren verglichen.

Das einfachste und, sowohl in der Literatur als auch in der Praxis, populärste Verfahren dieser Kategorie ist das Stochastic-Gradient-Descent (SGD)-Optimierungsverfahren, welches in Kapitel 2.5.1 beschrieben wird.⁵⁴ Deutlich komplexer ist die Gruppe der adaptiven Lernverfahren aufgrund der dynamischen Anpassung der Lernrate. Dadurch erzielen diese Optimierungsverfahren i.d.R. bessere Ergebnisse. Die Literatur zeigt, dass das Optimierungsverfahren Adaptive-Moment-Estimation (Adam) für die meisten Anwendungsfälle die besten Ergebnisse unter den adaptiven Lernverfahren erzielt. Aus diesem Grund wird das Verfahren in Kapitel 2.5.2 näher beschrieben und im Rahmen dieser Arbeit verwendet.⁵⁵ Darüber hinaus wurde ein Zeit- und Ressourcenschonendes Optimierungsverfahren gewählt. Dieses gehört zur Gruppe der Online-Lernverfahren, welche besonders für Datensätze mit vielen Nullen, sogenannte sparse Daten, geeignet sind. Es handelt sich

⁵²Vgl. Kinnebrock 2018, S. 41

⁵³Vgl. Kinnebrock 2018, S. 41 f.

⁵⁴Vgl. TensorFlow 2 2022e

⁵⁵Vgl. Ruder 2016 S. 10

hierbei um das in Kapitel 2.5.3 beschriebene Follow-The-Regularized-Leader (FTRL)-Optimierungsverfahren.⁵⁶

2.5.1 SGD

SGD gehört zur Gruppe der gradientenbasierten Optimierungsverfahren. Die Gradienten sind Vektoren, die die Richtung und Steigung einer Funktion an einem bestimmten Punkt im Raum angeben. Die Idee bei der Gradientenabstiegsmethode ist, dass die Parameter iterativ so verändert werden, dass die definierte Kostenfunktion gegen ein Minimum konvergiert.⁵⁷ Eine Kostenfunktion berechnet den Unterschied zwischen den vorhergesagten und den tatsächlichen Werten eines Modells. Folglich gilt je kleiner die Kostenfunktion ist, desto besser ist die Leistung eines Modells mit den verwendeten Daten. Der Gradient angewendet auf die Kostenfunktion zeigt die Richtung, in der die Kosten am schnellsten sinken.^{58 59}

Aufgrund des einfachen Konzepts und der leichten Algorithmus-Implementierung wird das SGD-Optimierungsverfahren häufig in Tutorials und Codebeispielen verwendet.⁶⁰ Im Vergleich zu anderen gradientenbasierten Optimierungsverfahren, wie z.B. dem Batch-Gradient-Descent (BGD), hat das SGD eine kürzere Ausführungszeit, da zum Berechnen der Gradienten nicht für jeden Schritt der komplette Trainingsdatensatz, sondern nur ein einzelner Eintrag verwendet wird.⁶¹ Dies führt zu einer höheren Schwankung, da jeder Schritt des Algorithmus von den ausgewählten Beispielen beeinflusst wird. Dadurch kann der Algorithmus zu potenziell besseren lokalen Minima springen. Gleichzeitig erschwert dieses Verhalten die Konvergenz zum exakten Minimum.⁶² Um dem entgegenzuwirken, ist das Optimierungsverfahren Mini-Batch-Gradient-Descent (MBGD) entwickelt worden, welches die Gradienten auf Basis einer Teilmenge von n Trainingsdatensätzen berechnet. Je nach gewählter Größe der Mini-Batches gelten die genannten Vor- und Nachteile von BGD und SGD. Bei zu großen Mini-Batches kann der Algorithmus einem lokalen Minimum nicht entkommen, wohingegen bei einer zu kleinen Größe das Konvergieren erschwert wird.⁶³

Ein wichtiger Punkt bei der Verwendung von SGD ist die Wahl der Lernrate. Sie gibt an, wie schnell sich die Modellparameter entlang des Gradienten anpassen. Bei einer zu kleinen Lernrate verlangsamt sich die Konvergenz, wohingegen eine zu große Lernrate die Konvergenz behindert, da die Kostenfunktion um das Minimum schwankt oder sogar

⁵⁶Vgl. McMahan u. a. 2013, S. 1 ff.

⁵⁷Vgl. Géron 2019, S. 118

⁵⁸Vgl. Géron 2019, S. 111

⁵⁹Vgl. Goodfellow, Bengio und Courville 2016, S. 80 ff.

⁶⁰Vgl. TensorFlow 2 2022e

⁶¹Vgl. Géron 2019, S. 124

⁶²Vgl. Ruder 2016, S. 2

⁶³Vgl. Géron 2019, S. 127

abweicht.⁶⁴ Folglich besteht die Herausforderung bei der Wahl der Lernrate darin, eine Balance zwischen schnellem und stabilem Konvergieren zu finden.

Eine weitere Möglichkeit die Konvergenz zu beschleunigen bietet die Momentum-Methode. Der Wert des Momentum-Algorithmus bildet den exponentiell abnehmenden gleitenden Durchschnitt vergangener Gradienten ab. Das bedeutet, dass der Durchschnitt vergangener Gradienten kontinuierlich berechnet wird, wobei die Bedeutung von weiter zurückliegenden Gradienten in diesem Durchschnitt exponentiell abnimmt. Das Momentum gibt eine Auskunft darüber, ob ein Gradient übersprungen werden kann, da es sich um ein lokales Minimum handelt.⁶⁵ Mithilfe des Momentum-Parameters in Form von $\frac{1}{1-\alpha}$ kann der Einfluss der Höchstgeschwindigkeit vom Momentum-Algorithmus auf den Gradientenabstiegsalgorithmus abgebildet werden. In der Praxis werden häufig die Werte 0,5, 0,9 und 0,99 verwendet. Am Beispiel von $\alpha = 0,9$ bedeutet dies, dass der Momentum-Algorithmus bis zu Zehn mal schneller sein kann als der Gradientenabstiegsalgorithmus.⁶⁶

2.5.2 Adam

Für neuronale Netze gilt die Lernrate als einer der am schwierigsten festzulegenden Parameter, da die Konfiguration die Leistung des Modells erheblich beeinflusst. Mit Einführung des Momentum-Parameters kann zwar das Problem etwas entschärft werden, jedoch steigert die Verwendung von weiteren Parametern die Komplexität eines Modells. Die Gruppe der adaptiven Lernverfahren löst dieses Problem, indem es die Lernrate während eines Trainingsprozesses auf Basis einer Standardlernrate dynamisch anpasst.⁶⁷ Zu den adaptiven Lernverfahren gehört das Optimierungsverfahren Adam, welches auf den Optimierungsverfahren Adaptive-Gradient (AdaGrad) und Root-Mean-Squared-Propagation (RMSProp) basiert.⁶⁸

AdaGrad passt die Lernrate individuell an, indem es den Kehrwert des Quadrats der bisherigen Gradienten berechnet. Dadurch werden große Gradienten mit kleineren Lernraten und kleine Gradienten mit größeren Lernraten versehen. Im Gegensatz zu dem Momentum-Algorithmus passt AdaGrad die Lernrate ausschließlich basierend auf dem bisherigen Verlauf der Gradienten an.⁶⁹ Dies führt jedoch zum Problem, dass die Lernrate im Laufe des Trainings immer kleiner wird, da die Summe der Quadrate der bisherigen Gradienten im Nenner stetig wächst. Folglich können die Lernraten für spätere Iterationen zu klein sein, was zu einer verlangsamten Konvergenz und im schlimmsten Fall zum Stagnieren des Optimierungsprozesses führen kann.⁷⁰

⁶⁴Vgl. Ruder 2016, S. 1 ff.

⁶⁵Vgl. Goodfellow, Bengio und Courville 2016, S. 292

⁶⁶Vgl. Goodfellow, Bengio und Courville 2016, S. 294

⁶⁷Vgl. Goodfellow, Bengio und Courville 2016, S. 302 f.

⁶⁸Vgl. Kingma und Ba 2014, S. 1

⁶⁹Vgl. Zeiler 2012, S. 2

⁷⁰Vgl. Ruder 2016, S. 6

Die Optimierungsverfahren RMSProp und Adaptive-Delta (Adadelta) wurden unabhängig voneinander entwickelt, um das Problem der verlangsamten Konvergenz in AdaGrad zu lösen. Beide Verfahren verwenden eine laufende Schätzung der zweiten Momente der Gradienten, um die Lernrate anzupassen. Es handelt sich dabei um eine statistische Methode, die den exponentiell abklingenden Durchschnitt des Quadrats der vergangenen Gradienten berechnet. Die neue Lernrate wird bestimmt, indem die bisherige Lernrate durch den berechneten Wert geteilt wird. Mit diesem Verfahren berücksichtigen RMSProp und Adadelta nur eine begrenzte Anzahl von vorherigen Gradienten und tragen somit zu einer beschleunigten Konvergenz bei. Obwohl sich die Formeln von RMSProp und Adadelta unterscheiden und zu unterschiedlichen Ergebnissen führen können, ähneln sich ihre Ansätze.^{71 72 73}

Adam nutzt das Verfahren zur Berechnung der zweiten Momente der Gradienten ebenfalls, um die Lernrate individuell anpassen zu können. Zusätzlich gibt Adam eine laufende Schätzung für die Berechnung der ersten Momente der Gradienten ab. Hierbei handelt es sich um den Durchschnitt der bisherigen Gradienten. Die Verwendung des ersten Moments schafft eine schnelle Anpassung der Lernrate für häufige und große Gradienten, während die Verwendung des zweiten Moments dafür sorgt, dass die Lernrate für seltene und kleine Gradienten weniger angepasst wird.⁷⁴

Die Erfinder von Adam, Diederik Kingma und Jimmy Ba, haben gezeigt, dass Adam ähnlich konvergiert wie SGD mit Momentum. Diese beiden Optimierungsverfahren konvergieren wiederum schneller als AdaGrad.⁷⁵ Außerdem zeigt Adam unabhängig von der Parameterkonfiguration eine gleiche oder bessere Leistung als RMSProp.⁷⁶

2.5.3 FTRL

Anders als bei herkömmlichen Lernverfahren, bei denen alle benötigten Trainingsdaten gesammelt und anschließend zur Modellanpassung verwendet werden, handelt es sich bei FTRL um ein sogenanntes Online-Lernverfahren. Das bedeutet, dass jeder neue Datensatz unabhängig von den anderen betrachtet wird. Dadurch sind Online-Lernverfahren in der Regel schneller und ressourcenschonender als traditionelle Batch-Lernverfahren. Zu jedem neuen Datensatz wird eine Vorhersage getroffen und anschließend mit dem tatsächlichen Ergebnis verglichen. Auf diese Weise aktualisiert und verbessert sich das Modell stetig selbst.⁷⁷ Aufgrund dieser Eigenschaft eignen sich Online-Lernverfahren besonders für Echtzeitanwendungen. FTRL wurde mit der Motivation entwickelt in Echtzeit

⁷¹Vgl. Goodfellow, Bengio und Courville 2016, S. 303 ff.

⁷²Vgl. Ruder 2016, S. 6 ff.

⁷³Vgl. Zeiler 2012, S. 3 ff.

⁷⁴Vgl. Kingma und Ba 2014, S. 2

⁷⁵Vgl. Kingma und Ba 2014, S. 5

⁷⁶Vgl. Kingma und Ba 2014, S. 8

⁷⁷Vgl. Shalev-Shwartz und Ben-David 2014, S. 287

Klickwahrscheinlichkeiten für Online-Anzeigen vorherzusagen.⁷⁸ Auch bestehende Optimierungsverfahren wie SGD können das Prinzip des Online-Lernverfahrens verwenden. Anstelle der Lösung von Batch-Problemen kann die Online-Version von SGD, die Online-Gradient-Descent (OGD) genannt wird, kontinuierlich mit neuen Daten lernen.⁷⁹

FTRL basiert auf einem Regularisierungsterm, der verhindern soll, dass die Gewichte des Modells zu sehr von einer Iteration zur nächsten variieren. Dadurch soll auch eine Überanpassung an eine bestimmte Klasse vermieden werden.⁸⁰

Ähnlich wie die Momentum-Erweiterung bei SGD gibt es bei FTRL verschiedene Varianten wie den FTRL-Proximal oder den Follow-The-Moving-Leader (FTML). Beide Varianten konnten in bestimmten Anwendungsfällen gute Ergebnisse erzielen. So konnte FTML bei den Bildern aus der Modified-National-Institute-of-Standards-and-Technology (MNIST)-Datenbank, die 60.000 handgeschriebene Ziffern umfasst, ähnlich genaue Vorhersagen über die geschriebenen Ziffern treffen wie Adam.⁸¹ Mit FTRL-Proximal wurden verschiedene binäre Klassifizierungsexperimente durchgeführt. So konnte FTRL-Proximal bspw. mit einer *Accuracy* von 98,9% vorhersagen, ob es sich bei einem Datensatz um Nachrichtentexte handelt.⁸²

Diese Arbeit fokussiert sich auf das FTRL-Proximal. Die Besonderheit hierbei ist die zusätzliche Regularisierung auf den aktuellen Datensatz. Es ähnelt dem OGD Optimierungsverfahren, da beide eine lokale Optimierung auf Basis des aktuellen Gradienten ermöglichen. Zusätzlich ermöglicht das FTRL-Proximal eine globale Optimierung des Modells, indem es die Summe aller vorherigen Gradienten berücksichtigt. Durch die Anpassung des bestehenden Modells können bessere Vorhersagen erzielt werden.⁸³

2.6 Deep-Learning-Bibliothek Keras

Mithilfe von Deep-Learning kann ein Computer komplexe Zusammenhänge auf Basis von einfachen Konzepten erlernen. Im Bereich Computer-Vision (CV) kann ein Computer bspw. auf Grundlage der Kombination von Ecken und Konturen Personen auf einem Bild erkennen. Ermöglicht wird dies durch Deep-Learning-Modelle wie KNN.⁸⁴ Die in der Programmiersprache Python geschriebene Deep-Learning-Bibliothek Keras basiert auf KNN und wurde mit dem Schwerpunkt entwickelt, schnelle Experimente mithilfe einer einfachen und minimalistischen Handhabung durchzuführen.⁸⁵

⁷⁸Vgl. McMahan u. a. 2013, S. 1

⁷⁹Vgl. McMahan u. a. 2013, S. 2

⁸⁰Vgl. Luo u. a. 2018, S. 1901 f.

⁸¹Vgl. Zheng und Kwok 2017, S. 6

⁸²Vgl. McMahan 2011, S. 531 f.

⁸³Vgl. McMahan und Streeter 2010, S. 2

⁸⁴Vgl. Goodfellow, Bengio und Courville 2016, S. 5

⁸⁵Vgl. Chollet 2015

Um dies zu ermöglichen, definiert Keras einen Application-Programming-Interface (API)-Standard. Die Vorteile der API werden in Kapitel 2.6.1 näher erläutert. Diese API bietet die Möglichkeit vortrainierte Keras-Applikationen zu verwenden. Eine davon ist das in Kapitel 2.6.2 beschriebene VGG16-Modell, welches in Kapitel 4 als Basis für die Generierung der OOS-Modelle verwendet wird.

2.6.1 Verwendung von Tensorflow 2 mithilfe der Keras API

Keras ist eine populäre High-Level-API für das Bauen und Trainieren von Deep-Learning Modellen. Kennzeichnend für eine High-Level-API sind eine leichte und einsteigerfreundliche Bedienbarkeit, kleiner Code Umfang und schnelle Ergebnisse. Für spezielle Anforderungen sind Low-Level-APIs geeignet. Eine Besonderheit der Keras-Bibliothek ist, dass die API nicht an eine spezielle Implementation gebunden ist. Da es sich bei Keras um eine Open-Source Bibliothek handelt, entwickelt eine aktive Mitgliedergemeinschaft die API stetig weiter. Die Keras API bietet bspw. eine Implementierung für TensorFlow, MX-Net, TypeScript, JavaScript, CNTK, Theano, PlaidML, Scala oder CoreML an.⁸⁶ Da man sich bei der Entwicklung nicht mit verschiedenen Technologien auseinander setzen muss, gelingt der Prozess von der Idee bis hin zu einem Ergebnis schneller.

Für die Entwicklung des OOS-Modells wird die in der Keras API angebotene Implementierung für *Tensorflow 2* verwendet. Bei *Tensorflow 2* handelt es sich ebenfalls um eine Open-Source Plattform mit dem Schwerpunkt Machine-Learning (ML). Sie dient als Infrastrukturschicht für differenzierbare Programmierung und bringt folgende Vorteile:

- Effiziente Ausführung von Low-Level Tensor-Operationen auf Central-Processing-Unit (CPU), Graphics-Processing-Unit (GPU) oder Tensor-Processing-Unit (TPU) Ebene.
- Berechnen der Gradienten von beliebig differenzierbaren Ausdrücken.
- Skalieren der Berechnungen auf verschiedene Endgeräte.
- Exportieren der Programme auf verschiedene Laufzeitumgebungen.

So ist es möglich die erstellten Keras-Modelle bspw. im Browser oder auf ein mobiles Endgerät zu exportieren und auszuführen.⁸⁷

2.6.2 VGG16-Modell

Das VGG16-Modell von Keras ist ein vortrainiertes KNN-Modell für Anwendungsfälle zur Bildklassifizierung.⁸⁸ Das erstellte Modell entstand aus einer Forschung, der die Auswir-

⁸⁶Vgl. Gupta, Gordon und Allison 2018

⁸⁷Vgl. Keras o.J.

⁸⁸Vgl. Simonyan und Zisserman 2014, S. 5 ff.

kung der Tiefe von Faltungsnetzen auf ihre Genauigkeit in der Bilderkennung untersucht. Ein Convolutional-Neural-Network (CNN), zu Deutsch etwa Faltendes-Neuronales-Netzwerk, wird für Mustererkennungsaufgaben wie der Gesichtserkennung oder der Erkennung handgeschriebener Zahlen genutzt.⁸⁹ Die Architektur vom VGG16-Modell basiert auf sehr kleinen 3 x 3 Faltungsfilttern.⁹⁰ Was genau ein Faltungsfilter macht, definiert der Faltungskern. Dieser befindet sich bei einer 3 x 3 Matrix in der Mitte, dem sogenannten Ankerpunkt. Um die einzelnen Werte der Faltungsmatrix berechnen zu können, müssen folgende Definitionen vorgenommen werden. Sei bei einer 3 x 3 Faltungsmatrix ein Bild $I(x, y)$, der Kernel $G(i, j)$ mit $0 < i < 4$ und $0 < j < 4$, der Ankerpunkt $a = 2$, dann kann der Faltungsfilter $H(x, y)$ mithilfe der Gleichung

$$H(x, y) = \sum_{i=1}^3 \sum_{j=1}^3 I(x - i + 2, y - j + 2)G(i, j) \quad (14)$$

berechnet werden.⁹¹ Das VGG16-Modell leitet die Bilder durch mehrere *Faltungsschichten*. Zum Trainieren des Modells werden RGB-Bilder mit einer Größe von 224 x 224 Pixeln verwendet. Zwischen einigen *Faltungsschichten* existieren *Max-Pooling-Schichten*. Diese ermöglichen es dem CNN Merkmale in benachbarten Pixeln zu erkennen und zu Merkmalskarten zusammenzufassen. Nachdem die *Faltungsschichten* abgearbeitet wurden, folgen drei *Fully-Connected-Schichten*. Damit werden die Neuronen aller Schichten bis hin zur Klassifizierungsebene verbunden. Die letzte Schicht ist die *Soft-Max-Schicht*, die eine Wahrscheinlichkeitsbewertung für jede Klasse liefert.⁹² ⁹³ Die Erkenntnis der Forschung war, dass mit 16-19 Gewichtsschichten die besten Ergebnissen im Bereich der Lokalisierung und Klassifizierungen erreicht werden kann. Die leistungsstärksten CNN-Modelle VGG16 und VGG19 sind veröffentlicht worden, um weitere Forschungen zur Verwendung von tiefen visuellen Repräsentationen in CV zu erleichtern.⁹⁴

2.7 Klassifizierungsmetriken

Um die Leistung der verschiedenen Modelle zu analysieren, werden verschiedene Bewertungskriterien herangezogen und miteinander verglichen. Dieses Kapitel stellt die gängigsten Klassifizierungsmetriken vor, welche oft als Maß für die Bewertung eines Modells verwendet werden.

Anhand der in Kapitel 2.7.1 definierten *Confusion-Matrix* können alle nachfolgenden Klassifizierungsmetriken formal definiert werden. Der Klassiker unter den Klassifizie-

⁸⁹Vgl. Matsugu u. a. 2003, S. 555

⁹⁰Vgl. Simonyan und Zisserman 2014, S. 1

⁹¹Vgl. Bradski und Kaehler 2008, S. 144 f.

⁹²Vgl. Simonyan und Zisserman 2014, S. 2

⁹³Vgl. Lu u. a. 2017, S. 38

⁹⁴Vgl. Simonyan und Zisserman 2014, S. 1

rungsmetriken ist die in Kapitel 2.7.2 beschriebene *Accuracy*-Metrik, die eine allgemeine Aussage über die Genauigkeit eines Modells trifft. Dabei werden beide Klassen berücksichtigt. Um jedoch Informationen zur richtigen Vorhersage einer einzelnen Klasse zu erhalten, bietet sich die in Kapitel 2.7.3 erläuterte *Precision*-Metrik an. Die in Kapitel 2.7.4 vorgestellten *Recall*-Metrik überprüft, ob alle Bilder, die zu einer Klasse gehören, erkannt wurden. Aus den Klassifizierungsmetriken *Precision* und *Recall* wird der in Kapitel 2.7.5 beschriebene *F1-Score* berechnet. Abschließend erläutert Kapitel 2.7.6, wie der Fehler zwischen den tatsächlichen und vorhergesagten Werten mithilfe der Kostenfunktion Binary-Cross-Entropy (BCE) berechnet werden kann.

2.7.1 Confusion-Matrix

Um die nachfolgenden Klassifizierungsmetriken in einer Formel ausdrücken zu können, müssen zuerst die verschiedenen Vorhersagefälle definiert werden. Die Basis dafür bildet die in Tabelle 1 dargestellte 2×2 *Confusion-Matrix*.

	Predicted Positive	Predicted Negative
Actual Positive	True-Positiv (TP)	False-Negative (FN)
Actual Negative	False-Positiv (FP)	True-Negative (TN)

Tabelle 1: Confusion-Matrix⁹⁵

Das Ergebnis einer Vorhersage kann immer einer von vier Klassen zugeordnet werden. Die zugehörige Klasse ergibt sich aus dem Zusammenhang zwischen dem vom Modell vorhergesagten Wert und dem tatsächlichen Wert. Nun gibt es zu jeder Klasse eine Beschreibung, die den Zusammenhang zwischen einem vorhergesagten und den tatsächlichen Wert näher verdeutlicht:

- True-Positive (TP) sind Werte, die auch als *Treffer* bezeichnet werden, da ein positiver Wert auch als solches vorhergesagt wurde.
- False-Positive (FP) sind sogenannte *Fehlalarme*, da fehlerhaft ein Ergebnis als korrekt vorhergesagt wurde.
- False-Negative (FN) sind die *Vermissten*, da ein positiver Wert nicht als solcher vorhergesagt wurde.
- True-Negative (TN) werden als *Nicht-Ereignisse* bezeichnet. Hierbei handelt es sich um Werte, die korrekterweise als nicht positiv vorhergesagt wurden.⁹⁶

⁹⁵Vgl. Larner 2021, S. 3

⁹⁶Vgl. Larner 2021, S. 2

2.7.2 Accuracy

Mithilfe der Definitionen aus der *Confusion-Matrix* kann die Formel 15 die *Accuracy*-Metrik formal beschreiben.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

Durch die Betrachtung aller *Treffer* und *Nicht-Ereignisse* im Verhältnis zu der Summe aller getätigten Klassifizierungen, gibt die *Accuracy*-Metrik einen guten Überblick über die Effektivität der binären Klassifizierung.⁹⁷ *Tensorflow 2* bietet die *Accuracy*-Metrik zur Auswertung der Modelle an.⁹⁸

Der Nachteil der *Accuracy*-Metrik ist, dass diese nur bei einer relativ ausgeglichen Klassenverteilung von Nutzen ist, da ansonsten der Wert eine verzerrte Abbildung der Wirklichkeit wiedergibt. Bei einem Klassenverhältnis von 99:1 wird bspw. eine *Accuracy* von 99% angegeben, wenn das Modell ausschließlich das Ergebnis mit der höheren Wahrscheinlichkeit für sämtliche Eingaben ausgibt.⁹⁹ Die folgenden drei Klassifizierungsmetriken bieten eine Lösung für dieses Problem an.

2.7.3 Precision

Um zu bestimmen, wie viele der Treffer richtig kategorisiert wurden, bietet sich die *Precision*-Klassifizierungsmetrik an. Sie gibt eine Auskunft darüber, wie wahrscheinlich es ist, dass der vorhergesagte positive Wert für eine Klasse tatsächlich richtig ist. Wenn eine Person bspw. einen positiven Coronatest erhält, kann die *Precision*-Metrik angeben, wie wahrscheinlich die Person tatsächlich infiziert ist.¹⁰⁰ Auf Grundlage der in der *Confusion-Matrix* definierten Regeln gilt die nachfolgende Formel 16.

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

Es werden weder die *Vermissten* noch die *Nicht-Ereignisse* von der *Precision*-Metrik beachtet.¹⁰¹ Ein Synonym für *Precision* ist Positive-Predictive-Values (PPV). Die Literatur definiert für PPV auch den Negativfall Negative-Predictive-Values (NPV), dessen Formel als

$$NPV = \frac{TN}{TN + FN} \quad (17)$$

⁹⁷Vgl. Sokolova und Lapalme 2009, S. 430

⁹⁸Vgl. TensorFlow 2 2022c

⁹⁹Vgl. Provost und Fawcett 1999, S. 1

¹⁰⁰Vgl. Larner 2021, S. 23 f.

¹⁰¹Vgl. Fawcett 2006, S. 862

definiert ist. Hierbei werden die *Treffer* und *Fehlalarme* nicht berücksichtigt. In der zuvor definierten Analogie bedeutet dies, dass die NPV-Metrik angibt, wie wahrscheinlich eine negativ getestete Person tatsächlich negativ ist.¹⁰² Im Kontext der *Tensorflow 2*-API bildet die *Precision*-Klasse nur den Positivfall ab.¹⁰³ Obwohl keine nativen Funktionen zur Berechnung des NPVs in TensorFlow vorhanden sind, ist es möglich, die Funktionsweise dieser Metrik durch eine benutzerdefinierte Implementierung nachzubilden.

2.7.4 Recall

Ist es weniger wichtig wie präzise eine Klasse vorhergesagt wird, sondern wie vollständig alle richtigen Fälle gefunden werden, so bietet sich die Verwendung der *Recall*-Metrik an. Sie kann die Effektivität einer Klassifizierung bei der Ermittlung positiver Werte mithilfe der Formel

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

bestimmen.¹⁰⁴ Hierbei werden nur die *Treffer* und die *Vermissten* beachtet. Am Beispiel einer Corona-Erkrankung bedeutet dies, dass die *Recall*-Metrik die Wahrscheinlichkeit für die Erfassung eines positiven Coronafalls angibt.

Synonyme für *Recall* sind Sensitivity (Sens) und True-Positive-Rates (TPR). Zu den Synonymen existieren Negativfälle die Specificity (Spec) und True-Negative-Rates (TNR) genannt werden.¹⁰⁵

$$TNR = \frac{TN}{TN + FP} \quad (19)$$

Dementsprechend betrachten die Negativfälle nur die Werte von *Nicht-Ereignisse* und *Fehlalarmen*. In *Tensorflow 2* können die Negativfälle wie bei der *Precision*-Metrik durch eine selbst implementierte Funktion nachgebaut werden, wohingegen für den Positivfall die *Recall*-Klasse von *Tensorflow 2* verwendet werden kann.

2.7.5 F1-Score

Die *F1-Score*-Metrik kombiniert die Eigenschaften der *Precision*- und der *Recall*-Metrik. Sie wird als das *harmonische Mittel* bezeichnet, da sie das Gleichgewicht zwischen der Richtigkeit und der Vollständigkeit bei der Vorhersage einer Klasse bewertet.¹⁰⁶ Die For-

¹⁰²Vgl. Larner 2021, S. 23 f.

¹⁰³Vgl. TensorFlow 2 2022d

¹⁰⁴Vgl. Sokolova und Lapalme 2009, S. 430

¹⁰⁵Vgl. Larner 2021, S.16 ff.

¹⁰⁶Vgl. Géron 2019, S. 92

mel dazu lautet

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (20)$$

und kann weiter vereinfacht werden durch

$$\begin{aligned} &= 2 \times \frac{\frac{TP}{TP+FP} \times \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \\ &= \frac{TP}{TP + \frac{FN+FP}{2}} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN} \end{aligned} \quad (21)$$

Es wird die Relation zwischen den vorhergesagten *Treffern* mit der Summe aus vorhergesagten und fälschlicherweise nicht klassifizierten *Treffern* berechnet.¹⁰⁷ Im Kontext der Corona-Pandemie könnte der *F1-Score* beispielsweise genutzt werden, um die Effektivität eines Diagnosetests zu bewerten, indem sowohl die Anzahl der tatsächlich infizierten Personen als auch die Anzahl der korrekt diagnostizierten Infektionen berücksichtigt wird.

Der Nachteil des *F1-Scores* ist, dass dieser eine gleichwertige Gewichtung von *Precision* und *Recall* voraussetzt. Wenn z.B. ein Klassifikator geeignete Videos für Kinder vorhersagen soll, dann ist es weniger schlimm, dass ab und zu gute Videos abgelehnt werden, solange die gezeigten Videos tatsächlich für Kinder geeignet sind. Das bedeutet, dass der *Recall*-Wert geringer sein darf als der *Precision*-Wert.¹⁰⁸ Um das Verhältnis anpassen zu können, muss die Formel um eine β -Variable ergänzt werden.

$$\begin{aligned} F_{\beta} - Score &= (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \\ &= (1 + \beta^2) \times \frac{\frac{TP}{TP+FP} \times \frac{TP}{TP+FN}}{\beta^2 \times \frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \\ &= \frac{(1 + \beta^2) \times TP}{(1 + \beta^2) \times TP + \beta^2 \times FN + FP} \end{aligned} \quad (22)$$

Ist $\beta < 1$, dann wird die *Precision*-Metrik höher gewichtet, wohingegen bei $\beta > 1$ die *Recall*-Metrik höher gewichtet wird. Im Fall des *F1-Score* ist $\beta = 1$.¹⁰⁹

In *TensorFlow 2* ist der *F1-Score* nicht als Standardmetrik enthalten. Alternativ kann der *F1-Score* mithilfe der `sklearn.metrics`-Bibliothek oder durch eine benutzerdefinierte Implementierung berechnet werden.¹¹⁰

¹⁰⁷Vgl. Sokolova und Lapalme 2009, S. 430

¹⁰⁸Vgl. Géron 2019, S. 93

¹⁰⁹Vgl. Manning, Raghavan und Schütze 2008, S. 154 ff.

¹¹⁰Vgl. scikit-learn o.J.

2.7.6 Kostenfunktion Binary-Cross-Entropy

Eine Kostenfunktion, in der Literatur auch als *Loss-Funktion* bekannt, ist ausschlaggebend für den Erfolg eines KNNs. Je nach verwendeter Kostenfunktion variiert die Leistung eines Modells, da nicht alle Kostenfunktionen gleich gut minimiert werden können. Im Kontext der binären Klassifizierung hat sich BCE als erste Wahl für eine Kostenfunktion durchgesetzt. Andere Kostenfunktionen wie der Mean-Squared-Error (MSE) oder der Mean-Absolute-Error (MAE) haben bei gradientenbasierten Optimierungen oft schlechtere Ergebnisse erzielt.¹¹¹

Für die Formel der Kostenfunktion BCE sei p die vorhergesagte Klasse und y die tatsächliche Klasse. Da es sich um eine binäre Klassifizierung handelt, kann y nur die Werte 0 oder 1 annehmen. Hingegen kann p die Werte annehmen, die die Aktivierungsfunktion in der letzten Schicht definiert. Bei der *Sigmoidfunktion* wären es alle Werte zwischen 0 und 1. Die Differenz zwischen den tatsächlichen und vorhergesagten Klassen kann mit

$$BCE = -y \log(p) - (1 - y) \log(1 - p) \quad (23)$$

berechnet werden. Die Formel 23 berücksichtigt nicht, welche der beiden Klassen 0 oder 1 falsch klassifiziert wurde. Jede Fehlklassifizierung führt zum gleichwertigen Verlustbetrag.¹¹² Die Intuition von BCE ist, dass der Verlustbetrag die Unsicherheit des Modells in Bezug auf die Vorhersage der tatsächlichen Klasse berücksichtigt. Je näher die Vorhersage an den tatsächlichen Wert heranreicht, desto geringer ist der Verlustbetrag.¹¹³

In *TensorFlow 2* ist die Kostenfunktion BCE, die Auskunft über die Sicherheit in der Vorhersage eines Modells gibt, als Standardmetrik enthalten und kann mit der Funktion `tf.keras.losses.BinaryCrossentropy` aufgerufen werden.¹¹⁴

¹¹¹Vgl. Goodfellow, Bengio und Courville 2016, S. 177

¹¹²Vgl. Kulkarni, R und Sharath 2022, S. 319

¹¹³Vgl. Géron 2019, S. 149 f.

¹¹⁴Vgl. TensorFlow 2 2022b

3 Datengrundlage

Die Daten stellen das Fundament eines Modells dar. Wie die geeigneten Stammdaten für die OOS-Erkennung beschafft wurden, erläutert Kapitel 3.1. Auf Basis dieser Daten wurden die größten Herausforderungen für die OOS-Erkennung in Kapitel 3.2 analysiert. Um die verfügbare Datenmenge zu erhöhen, kann die in Kapitel 3.3 beschriebene Augmentation von Daten verwendet werden. Abschließend beschreibt Kapitel 3.4 mit welchen Methoden die Aufteilung der Daten in Trainings- und Testdaten erfolgt.

3.1 Datenbeschaffung

Für die Trainings- und Testphase der im Rahmen dieser Arbeit generierten OOS-Modelle wird entsprechendes Bildmaterial benötigt. Kapitel 3.1.1 erläutert, welchen Kriterien die benötigten Bilder entsprechen müssen und zeigt, wie diese aussehen können. Anschließend gibt Kapitel 3.1.2 eine Übersicht, wie viele Bilder je Markt zur Verfügung stehen.

3.1.1 Kriterien für die Bilddaten

Da bei der REWE Bilddaten nur für den Webauftritt verfügbar sind und diese folglich keine Regale und Regallücken darstellen, mussten entsprechende Bilder für diese Arbeit erstellt werden. Diese müssen gefüllte und leere Fächer in den Regalen der Märkte darstellen, um als Basis für das Modelltraining zu dienen. Hierfür wurden mehrere Märkte aufgesucht, um verschiedene Fotos von den Regalbeständen zu schießen. Jedes Foto wurde so geschossen, dass eine Person eindeutig erkennen kann, ob eine Bestandslücke existiert. Folglich grenzt sich diese Arbeit von Sonderfällen wie falsch positionierten oder schwer zu erkennenden Artikeln ab. Dazu zählen auch leere Kartons oder Flächen für freistehende Getränkeboxen. Ein Beispiel für eindeutig zu erkennende In-Stock und OOS-Artikel zeigen die zwei nachfolgenden Abbildungen 10 und 11:



Abbildung 10: Bild ohne OOS-Artikel



Abbildung 11: Bild mit OOS-Artikel

Die gezeigten Bilder beziehen sich auf Artikel im Getränkemarkt eines REWE-

Marktes. Darüber hinaus wurden auch Bilder von anderen Sortimenten im Markt geschossen, solange sie die Kriterien der eindeutigen Erkennbarkeit einhalten. Dies erfüllen insbesondere Kolonialwaren (Kolo). Zu dieser Kategorie gehören das Trockensortiment wie auch Molkereiprodukte (Mopro). Auch Bilder von Non-Food Artikel können erstellt werden, solange es sich nicht um Artikel von Konzessionären handelt.

Weniger gut geeignet sind Bilder von der Frischetheke, da die Produkte keinen fest zugeordneten Platz haben. Auch schwer zugängliche Flächen oder Werbeständer wurden im Hinblick auf das Zielbild nicht berücksichtigt, da diese bei der Erkennung für autonom fahrende Roboter oder eine angebrachte Kamera weniger gut geeignet sind. Aus dem selben Grund werden Artikel an der Kasse wie Zeitungen, Magazine oder Zigaretten nicht berücksichtigt.

3.1.2 Übersicht der erzeugten In-Stock und OOS-Bilder

Nicht alle REWE Märkte sind identisch eingerichtet. Sie verfügen über unterschiedliche Raumgestaltungen und technologische Ausstattungen. So müssen bspw. einige Märkte manuell Preisänderungen auf Papier ausdrucken und am Regal anbringen, während andere Märkte mithilfe von Electronic-Shelf-Labels (ESL) Preisänderungen elektronisch ans Regal senden. Darüber hinaus können sich die Regale, die die Märkte zum Verräumen ihrer Ware verwenden, in ihrer Form, Farbe und Bauweise unterscheiden. Diese Aspekte haben Auswirkungen auf die Fotos, mit denen das Modell trainiert wird. Um zu verhindern, dass das OOS-Modell ausschließlich bei Märkten mit gleichen Voraussetzungen gut funktioniert, wurden insgesamt *sechs* verschiedene Märkte aufgesucht. Diese beinhalten sowohl zentral geführte, als auch von verschiedenen Kaufmann / Kauffrau geführte Märkte. Aus Datenschutzgründen werden die Märkte nachfolgend Markt A bis Markt F genannt. Eine Übersicht über die Anzahl der erzeugten In-Stock und OOS-Bilder je Markt zeigt die Tabelle 2.

Markt	In-Stock	OOS	Gesamt
A	76	34	110
B	231	177	408
C	369	389	758
D	370	333	703
E	148	158	306
F	231	196	427
6	1425	1287	2712

Tabelle 2: Anzahl erstellter In-Stock und OOS-Bilder je Markt

Insgesamt wurden **2712** Bilder erstellt. In Markt A wurden im Verhältnis zu den anderen Märkten weniger Bilder geschossen, da der im Vorfeld durchgeführte PoC ausschließlich Bilder vom Getränkemarkt verwendet.

Die Datengrundlage besteht zu 47% aus OOS-Bildern. Dies spiegelt nicht die Realität wieder. In der Lebensmittelbranche liegt die OOS-Quote zwischen acht und neun Prozent.¹¹⁵

3.2 Herausforderungen bei den OOS-Daten

Selbst bei der zuvor definierten Eingrenzung gibt es zahlreiche Möglichkeiten, wie eine OOS-Lücke aussehen kann. Dementsprechend ist die größte Herausforderung für ein KNN die Vielzahl an OOS-Varianten als solche zu erkennen. Eine Auswahl an Darstellungen für eine OOS-Situation zeigt Abbildung 12.

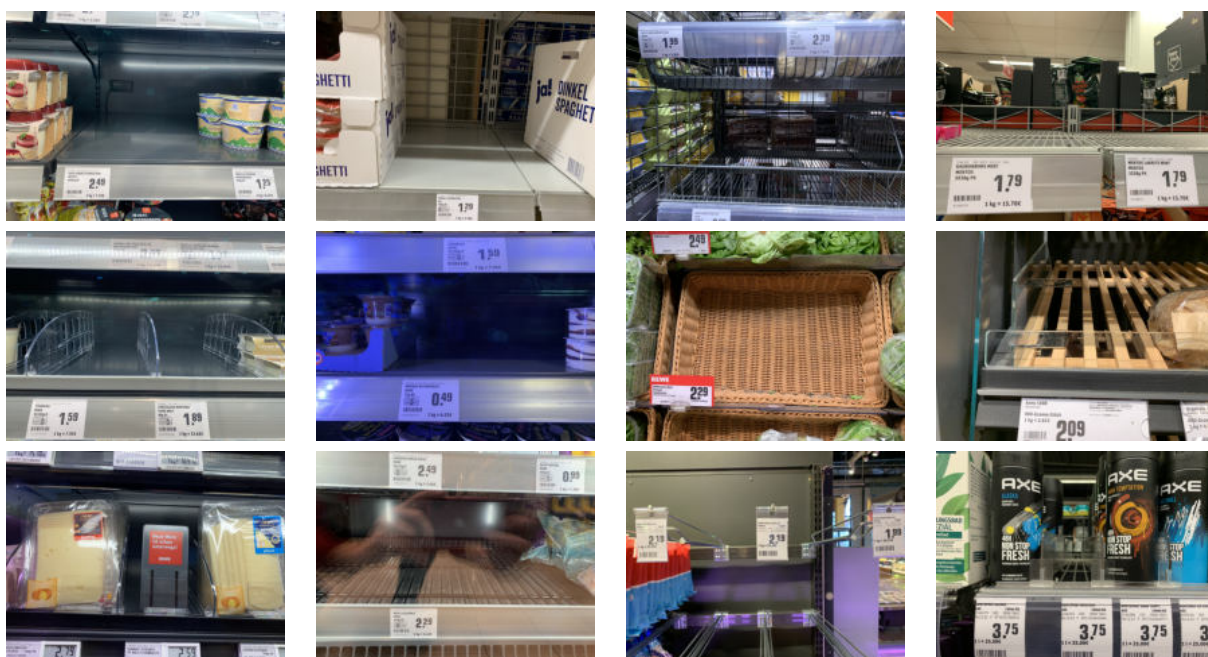


Abbildung 12: OOS-Varianten

Durch helle Farben wie Rot und Gelb sollen Artikel die Aufmerksamkeit und den Appetit von Kunden anregen.¹¹⁶ Um die Artikel möglichst gut zu präsentieren, sind die Regale in einem dezenten Grauton gehalten, wodurch Grautöne markant für eine OOS-Situationen sind. Ausnahmen bilden die Präsentation von Backartikeln und Obst und Gemüse (OG). Zudem unterscheiden sich Regale in ihrem Aufbau. Wie in den Bildern zu erkennen, haben die Rückwände und Unterböden, je nach Regaltyp unterschiedliche Beschaffenheiten. Hier gibt es jeweils offene und geschlossene Varianten, sowie Mischtypen. Offene oder mit Gitter versehene Flächen stellen eine besondere Herausforderung dar, da in dem Fall weitere Artikel auf der anderen Seite eines Regals sichtbar sind. Dies kann zu einer fehlerhaften Klassifizierung des OOS-Artikels als In-Stock führen.

¹¹⁵Vgl. Hofer 2009, S. 4

¹¹⁶Vgl. Singh 2006, S. 787

Bei den Backwaren und OG Artikeln kommt die Schwierigkeit hinzu, dass das Modell die braune Farbe von Regalen und Körben als neue mögliche OOS-Variante erkennen muss. Es gibt bei der REWE nämlich keine einheitliche Vorgabe wann welche Körbe oder Regale verwendet werden und wie diese auszusehen haben. Abbildung 13 verdeutlicht an einigen Beispielen die verschiedenen individuellen Möglichkeiten eines Marktes zur Gestaltung der Regale.



Abbildung 13: Regal- und Korbvarianten

Es sind Regale und Boxen in einem besonders geschnittenen Design zu erkennen. Sie können sich in Farbe, Form und Materialbeschaffenheit unterscheiden. Die Vielzahl an Sonderfällen mit verhältnismäßig wenigen Bildern erschweren das Trainieren eines Modells. Selbst einzelne Elemente wie einen Warenstopper gibt es in verschiedensten Varianten. Warenstopper schieben einen Artikel in einem Regal nach vorne und zeigen sich nachdem der letzte Artikel aus dem Regal entnommen wurde. Einige Beispiele zu den verschiedenen Warenstoppervarianten zeigt Abbildung 14.

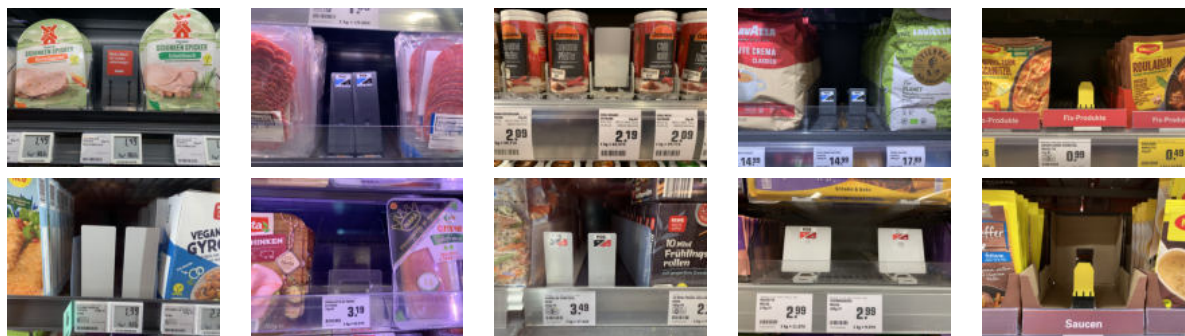


Abbildung 14: Warenstopper-Varianten

Wie in Abbildung 12, 13 und 14 zu erkennen, unterscheidet sich die Breite eines Artikels und somit die Breite einer potenziellen Regallücke stark. Hinzu kommt, dass einige Produkte lediglich in einer Reihe aufgestellt sind, während an anderen Stellen das gleiche Produkt in mehreren Reihen nebeneinander steht. Offensichtlich sind breitere Lücken für einen Menschen leichter zu erkennen. Gleichzeitig besteht die Gefahr, dass ein Artikel mit einer großen Regalplatzfläche fälschlicherweise als OOS erkannt wird, wenn ein Großteil jedoch nicht die komplette Fläche leer ist.

Im Markt gibt es Gegebenheiten bei denen mehrere Artikel nebeneinander OOS sind. Da jedoch nicht das Ziel ist Artikel zu identifizieren, sondern eine OOS-Situation, sind mehrere OOS-Artikel nebeneinander hilfreich für die Erkennung.

Eine weitere Herausforderung sind die Lichtverhältnisse. Abbildung 12 zeigt ein Bild auf dem eine Spiegelung in der Glasscheibe sichtbar ist. Darüber hinaus können Verunreinigungen an der Glasscheibe die Sicht auf den Artikel erschweren. Dementsprechend muss das Modell in der Lage sein eine OOS-Situation bei verschiedenen Beleuchtungs- und Sichtverhältnissen zu erkennen.

3.3 Augmentation der Daten

Bei der Augmentation von Daten werden bestehende Bilder durch verschiedene Methoden verändert und anschließend als neue unabhängige Bilder verwendet. Kapitel 3.3.1 beschreibt die auf die In-Stock-Daten angewendeten Augmentationsarten. Anschließend erläutert Kapitel 3.3.2 aus technischer Sicht wie die augmentierten Bilder während des Trainingsprozesses erzeugt werden. Dabei werden die augmentierten Bilder nur für die Dauer der Verwendung gespeichert. Um eine weitere Datenbasis zu schaffen, die das Datenverhältnis der OOS- und In-Stock-Daten entsprechend der Realität abbildet, beschreibt Kapitel 3.3.3 das hierzu benötigte Speichern von augmentierten In-Stock-Bildern.

3.3.1 Augmentationstypen

Es besteht die Gefahr, dass das Modell sich aufgrund der geringen Anzahl an Daten zu stark an den Trainingsdaten orientiert. Das könnte zur Folge haben, dass die Kostenfunktion für die Testdaten hoch ist, obwohl sie bei den Trainingsdaten viel niedriger war. Bei einem großen Unterschied der Kostenfunktion zwischen den Datensätzen spricht man von *Overfitting*.¹¹⁷ Um dem entgegenzuwirken, bietet es sich an mit augmentierten Daten zu arbeiten.¹¹⁸ Um den manuellen Aufwand so gering wie möglich zu halten, können die vorhandenen Bilder mithilfe der `ImageDataGenerator`-Klasse von der Keras-Bibliothek automatisch augmentiert werden.¹¹⁹ Die Klasse bietet eine Vielzahl von Konfigurationen zur Augmentation der Daten an. Nachfolgend beschreibt die Tabelle 3 alle Parameter, die bei der Augmentation der OOS-Daten angewendet werden.

¹¹⁷Vgl. Goodfellow, Bengio und Courville 2016, S. 110

¹¹⁸Vgl. Abayomi-Alli u. a. 2022, S. 15

¹¹⁹Vgl. Shorten und Khoshgoftaar 2019, S. 43

Parameter	Datentyp	Beschreibung
rescale	Float	Das Bild wird mit dem angegebenen Wert neu skaliert.
horizontal_flip	Boolean	Bei <i>true</i> wird das Bild zufällig gespiegelt.
rotation_range	Integer	Gibt den Gradbereich für eine zufällige Bildrotation an.
fill_mode	String	Definiert entsprechend dem angegebenen Modus (<i>constant</i> , <i>nearest</i> , <i>reflect</i> oder <i>wrap</i>) wie die Pixel außerhalb der Grenzen eines Bildes, die z.B. bei einer Drehung auftreten, gefüllt werden.
width_shift_range	Float	Definiert um wie viel Prozent ein Bild horizontal verschoben wird.
height_shift_range	Float	Definiert um wie viel Prozent ein Bild vertikal verschoben wird.
brightness_range	Float Liste	Definiert einen Bereich für eine Helligkeitsverschiebung. Das Bild kann sowohl dunkler als auch heller werden. Die Liste enthält zwei Float Werte, die den Helligkeitsverschiebungsbereich angeben.
zoom_range	Float	Definiert den Bereich für einen zufälligen Zoom. Mit dem angegebenen Prozentwert kann das Bild sowohl vergrößert als auch verkleinert werden.

Tabelle 3: Verwendete Parameter der ImageDataGenerator-Klasse¹²⁰

Die vollständige Parameterliste ist in der angegebenen Quelle zu finden. Um die Auswirkungen der einzelnen Parameter zu verdeutlichen, veranschaulichen die nachfolgenden Abbildungen wie sich die Originalabbildungen 10 und 11 nach Anwendung der verschiedenen Augmentationstypen verändern.



Abbildung 15: Horizontal Flip

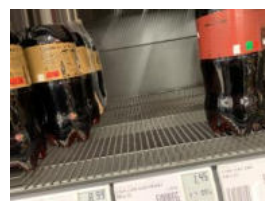


Abbildung 16: Rotation Range

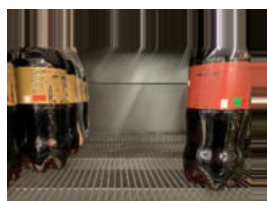


Abbildung 17: Random Shifts

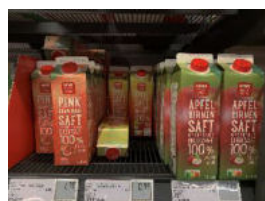


Abbildung 18: Random Brightness

¹²⁰Vgl. TensorFlow 2 2022f



Abbildung 19: Random Zoom



Abbildung 20: Alle Parameter

Bei allen Bildern wurde der `rescale`-Parameter verwendet. Abbildung 15 stellt den `horizontal_flip`-Parameter für den Wert `True` dar. Bei Abbildung 16 wird das Bild zufällig gedreht. Hier wurde der Wert auf maximal 30 Grad begrenzt, da die Kamera niemals seitlich oder auf dem Kopf zum Regal ausgerichtet sein sollte. Die bei der Augmentierung auftretenden Ränder werden durch den `fill_mode`-Parameter mit dem Modus `nearest` aufgefüllt. Die Auswirkungen von den Parametern `width_shift_range` und `height_shift_range` zeigt Abbildung 17. Die Helligkeitsverschiebung mit dem Parameter `brightness_range` stellt Abbildung 18 dar. Das Vergrößern und Verkleinern eines Bildes mithilfe des Parameters `zoom_range` veranschaulicht Abbildung 19. Die abschließende Abbildung 20 zeigt den Effekt, wenn alle Parameter gleichzeitig auf ein Bild einwirken.

3.3.2 Trainieren mit augmentierten Bildern

Die `ImageDataGenerator`-Klasse bietet nicht nur die Möglichkeit Bilder zu augmentieren, sondern kann auch das Aufteilungsverhältnis definieren. Dies erfolgt durch die Angabe des Parameters `validation_split`, welcher den Anteil der Testdaten definiert. Aufgrund der Anzahl verfügbarer Bilder, sowie basierend auf den Empfehlungen der in Kapitel 3.4.1 beschriebenen Holdout-Methode, ist der Wert für den `validation_split` mit `0.3` angegeben. Das bedeutet, dass ein Datensatz in 70% Trainings- und 30% Testdaten eingeteilt wird. Relevant für die Augmentation sind die Trainingsdaten, da das OOS-Modell die Trainingsbilder ausschließlich während der Trainingsphase augmentiert.

Die weiteren Werte für die in Tabelle 3 definierten Parameter sind im Codeausschnitt in Anhang C.2.1 zu finden. Nach Initialisierung der `ImageDataGenerator`-Klasse liefert der Codeausschnitt die Variable `datagen` zurück, welche alle Regeln für das Generieren von augmentierten Bildern enthält. Die Methode `flow_from_directory(...)` übermittelt den Pfad zu den Bilddaten, auf die die Augmentationsregeln angewendet werden sollen. Innerhalb des Pfades sind zwei Unterordner zu finden. In dem einen befinden sich die OOS-Bilder, im anderen die Bilder, die keine OOS-Situation zeigen. So kann die Methode erkennen, wie viele Klassen es gibt und welche Bilder der jeweiligen Klasse angehören. Das nachfolgende Listing 1 veranschaulicht den Aufruf der Methode `flow_from_directory(...)` zum Laden von Trainingsdaten.

```
def get_train_data(datagen, directory):
    train_it = datagen.flow_from_directory(
        directory=directory,
        class_mode='categorical',
        batch_size=64,
        target_size=(224, 224),
        subset='training')
    return train_it
```

Listing 1: Laden der Trainingsdaten

Der Funktion wird der Parameter `directory` mitgegeben, der den Pfad zu den Bilddaten enthält. Des Weiteren wird der Parameter `class_mode` auf `categorical` gesetzt. Dadurch werden die Ergebnisse des KNNs im Kontext der OOS-Erkennung in einem zweidimensionalen Array erfasst. Für die Darstellung des Arrays wird die *NumPy*-Bibliothek verwendet. Wenn vier Bilder klassifiziert werden, sieht der Output im Idealfall bspw. wie folgt aus: `np.array([[1, 0], [1, 0], [0, 1], [0, 1]])`. Das Array enthält vier innere Arrays, die jeweils zwei Werte enthalten. Dabei steht der erste Wert für die In-Stock-Klasse und der zweite Wert für die OOS-Klasse. Im Idealfall ist exakt einer dieser Werte Eins und der andere Null. Im gezeigten Beispiel wurden jeweils zwei Bilder als In-Stock und OOS klassifiziert. Alternativ kann als `class_mode` auch `binary` verwendet werden. Dabei handelt es sich um ein eindimensionales *NumPy*-Array, welches nur die Werte Null oder Eins enthält. Diese Konfiguration eignet sich allerdings nicht für Klassifizierungsmetriken wie den *Recall*, da ansonsten nur die Ergebnisse für die In-Stock- oder die OOS-Klasse berechnet werden können. Um dies zu umgehen, wurde die `categorical` Konfiguration verwendet. Ein Test basierend auf der vorangegangenen Projektarbeit, die ursprünglich mit der `binary` Konfiguration erstellt wurde, zeigt, dass die Änderung des Wertes auf `categorical` keine Auswirkungen auf die Ergebnisse des OOS-Modells hat.

Damit alle Bilder die gleichen Größendimensionen aufweisen, definiert der Parameter `target_size` auf welche Größe die Bilder skaliert werden. Es bietet sich an die Standardgröße zu verwenden, mit der das VGG16-Modell trainiert wurde. Der letzte Parameter `subset` definiert, für welchen Zweck die Bilder verwendet werden. Zu Trainingszwecken muss der Wert `training` eingetragen werden. Hierin besteht auch der einzige Konfigurationsunterschied zu der Generierung der Testdaten. Wie in Anhang C.3.1 zu erkennen, ist für die Testdaten der Wert `validation` einzutragen. Während eines Trainingsdurchlaufs, welcher als *Epoche* bezeichnet wird, trainiert das Modell standardmäßig einmal mit allen Bildern. Jedoch ist es nicht immer vorteilhaft mit allen Bildern gleichzeitig zu trainieren. Mithilfe des `batch_size`-Parameters wird definiert, mit wie vielen Bildern das KNN je Epochenschritt trainiert. Empfohlen wird der Wert 64.¹²¹ Wenn das OOS-Modell mit allen 2712 verfügbaren Bildern trainiert, gibt es pro Epoche 43 Epochenschritte. Die vor-

¹²¹Vgl. Google 2021

genommene Implementierung des OOS-Modells erzeugt während eines Epochenschritts zu jedem verwendeten Trainingsbild ein augmentiertes Bild. Nach Beendigung eines Epochenschritts werden die generierten augmentierten Bilder wieder gelöscht.¹²²

3.3.3 Datengrundlage durch das Augmentierten von Bildern erweitern

Um den Anteil der OOS-Bilder im Verhältnis zu den In-Stock-Bildern auf die in der Lebensmittelbranche vorherrschenden 9% zu senken, kann entweder die Anzahl der In-Stock-Bilder erhöht oder die Anzahl der OOS-Bilder verringert werden. Die verfügbaren Bilder bestehen aus 1425 In-Stock- und 1287 OOS-Bildern. Das Verringern von Bildern führt dazu, dass aufgrund der 1425 In-Stock-Bilder nur 140 OOS-Bilder verwendet werden dürfen. Das bedeutet, dass nur Elf Prozent der 1287 generierten OOS-Bilder für das Modelltraining eingesetzt werden dürfen. Die dadurch verringerte Anzahl an Trainingsdaten führt jedoch zu einer erhöhten *Overfitting*-Gefahr.

Darum bietet es sich an, die Datenbasis durch das Generieren und Speichern von augmentierten Bildern zu erweitern. Um zu unterscheiden, ob ein Modell ausschließlich mit den Originalbildern oder mit den angepassten Datenverhältnis trainiert werden soll, ist eine separate Datenbasis erstellt worden. In dieser sind zu jedem originalen In-Stock-Bild neun augmentierte In-Stock-Bilder. Ein Ausschnitt des Codes zur Generierung und Speicherung von augmentierten Bildern ist in Listing 2 zu finden. Der gesamte Auszug vom Code ist in Anhang C.2.2 dargestellt.

```

for i in range(9):
    train_it = datagen.flow_from_directory(
        directory = directory,
        class_mode='categorical',
        batch_size=1425,
        target_size=(224, 224),
        save_to_dir=save_to_dir,
        save_prefix="augmented_" + str(i) + "_",
        save_format='jpg',
    )
    train_it.next()

```

Listing 2: Generieren und Speichern von augmentierten Bildern

Der Code unterscheidet sich durch eine `batch_size` von 1425 von Listing 1 und weist auf einen Pfad, der lediglich die In-Stock-Bilder enthält. Dafür wurde temporär der `Datengrundlage_Test` Ordner angelegt. Die augmentierten Bilder werden im JPG-Format in der Datenbasis `Datengrundlage_Augmentiert` im dazugehörigen Marktordner unter `!OOS` abgelegt. Gekennzeichnet sind die Bilder durch das Prefix `augmented_`. Mit jedem Aufruf der `next()`-Methode wird zu jedem Bild im Batch ein neues augmentiertes

¹²²Vgl. TensorFlow 2 2022f

Bild generiert. Somit gibt es insgesamt 14250 In-Stock Bilder, was mit 1287 OOS-Bildern eine OOS-Quote von circa 8,3 Prozent ergibt und damit im erwarteten Bereich der Lebensmittelbranche liegt.

3.4 Aufteilung der Trainings- und Testdaten

Die zum Anlernen eines ML-Modells verwendeten Daten werden Trainingsdaten genannt. Um zu überprüfen, wie gut das Modell in der Praxis funktioniert, müssen Daten verwendet werden, mit denen das Modell nicht trainiert wurde. Diese Daten werden Testdaten genannt.¹²³ Es gibt verschiedene Methoden zum Aufteilen einer Datenbasis in Trainings- und Testdaten. Die einfachste und häufigste Methode ist die in Kapitel 3.4.1 beschriebene *Holdout*-Methode. Um zu vermeiden, dass das Ergebnis eines Modells aufgrund der zufälligen Datenteilung ein Sonderfall ist, bietet sich die Verwendung der in Kapitel 3.4.2 beschriebenen *Cross-Validation*-Methode an. Zur Untersuchung der verschiedenen Konfigurationen eines OOS-Modells wurde ein hybrider Ansatz aus der *Holdout*- und der *Cross-Validation*-Methode verfolgt, welcher in Kapitel 3.4.3 erläutert wird.

3.4.1 Holdout-Methode

Bei der *Holdout*-Methode werden die verfügbaren Daten zufällig in zwei disjunkte Teildatensätze aufgeteilt. Der eine Teildatensatz wird zum Trainieren des Modells verwendet, der andere zur Validierung. Diese einfache und schnelle Vorgehensweise ist charakteristisch für die *Holdout*-Methode. Problematisch wird die Methode für kleine Datensätze, da bei einer ungünstigen Aufteilung wichtige Informationen für die Modellbildung verloren gehen. Durch die Augmentierung der Daten können die potenziell negativen Auswirkungen verringert werden. Eine weitere Möglichkeit, um dem Problem entgegenzuwirken, ist das Verhältnis der Aufteilung anzupassen. Es bietet sich bei kleineren Datensätzen an, einen höheren Prozentsatz der Daten für das Training zu verwenden.¹²⁴ Für das OOS-Modell wurde eine 70:30 Aufteilung gewählt.

3.4.2 Cross-Validation-Methode

Es gibt verschiedene Varianten der *Cross-Validation*-Methode. Im Kontext der Klassifizierung und Regression zählt die *K-Fold Cross-Validation* (KFCV)-Methode zu den in der Praxis am häufigsten angewendeten Verfahrenen.¹²⁵

Im Gegensatz zur *Holdout*-Methode teilt die KFCV-Methode die Daten in k gleich große Blöcke und berechnet in jedem der k Durchläufe die Ergebnisse der Klassifizierungsmetriken. In jedem Durchlauf wird das Modell mit $k - 1$ Datenblöcken trainiert und

¹²³Vgl. Awad und Khanna 2015, S. 6

¹²⁴Vgl. Albers u. a. 2009, S. 538 ff.

¹²⁵Vgl. Bergmeir und Benítez 2012, S. 197

anschließend mit dem ausgeschlossenen Datenblock getestet. Nach k Iterationen hat jeder Datenblock einmal die Rolle als Testdatensatz eingenommen.¹²⁶ Der Mittelwert über die Ergebnisse der Iterationen ergibt die Kreuzvalidierungsschätzung des Modells.¹²⁷ Nachfolgend veranschaulicht Abbildung 21 die KFCV-Methode.

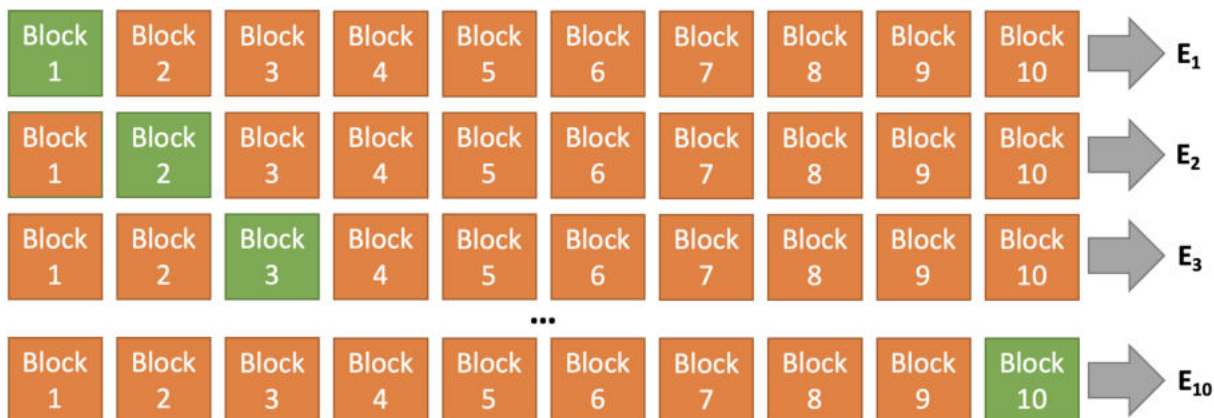


Abbildung 21: Cross-Validation mit $k=10$ ¹²⁸

Im Beispiel wurde ein Wert von $k = 10$ verwendet, der gemäß der Literatur, wie auch $k = 5$, als gut geeignet angesehen wird. Somit ist die Datenbasis in zehn gleich große Blöcke aufgeteilt.¹²⁹ Jede Zeile in der Abbildung stellt eine Iteration dar. Die orangenen Blöcke bilden zusammengesetzt die Trainingsdaten, während die einzelnen grünen Blöcke die Testdaten enthalten. Das Ergebnis jeder Iteration ist als E_k definiert. Der Mittelwert aller Ergebnisse E kann formal als

$$E = \frac{1}{k} \sum_{i=1}^k E_i \quad (24)$$

angegeben werden.¹³⁰ Der Nachteil dieser Methode ist die erhöhte Berechnungszeit, da für jede Iteration ein eigenes Modell anzufertigen ist. Dafür sind die Ergebnisse aussagekräftiger, da das Risiko für einen Sonderfall aufgrund der mehrfachen Teilung der Datensätze reduziert wird.¹³¹

3.4.3 Hybride Modellvalidierung

Bei der Implementierung und Evaluation der OOS-Modelle wurde ein hybrider Ansatz verfolgt, der die theoretischen Konzepte der *Holdout*- und *KFCV*-Methode vereint. Um das OOS-Modell an einem praxisnahen Beispiel testen zu können, ist die Datenbasis nicht

¹²⁶Vgl. Albers u. a. 2009, S. 544 f.

¹²⁷Vgl. Arlot und Celisse 2010, S. 52 ff.

¹²⁸Vgl. Niu u. a. 2018, S. 9

¹²⁹Vgl. Albers u. a. 2009, S. 545

¹³⁰Vgl. Niu u. a. 2018, S. 9

¹³¹Vgl. Albers u. a. 2009, S. 545

in k gleich große Datenblöcke, sondern in Daten je Markt aufgeteilt worden. Von den sechs Märkten, in denen die Daten gesammelt wurden, wurden jeweils fünf zum Trainieren und einer zum Testen des OOS-Modells genutzt.

Der Testmarktdatensatz kommt während der Trainingsphase nicht zum Einsatz. Stattdessen werden die Bilddaten der fünf Trainingsmärkte nach der *Holdout*-Methode zufällig im Verhältnis 70:30 in Trainings- und Testdaten geteilt. Nach jeder Epoche prüft das OOS-Modell seinen Trainingsfortschritt mithilfe dieser Testdaten. So wird sichergestellt, dass die Daten des Testmarktes keinen Einfluss auf die Modellbildung nehmen. Anschließend wird das fertig trainierte Modell verwendet, um die Daten des Testmarktes zu klassifizieren und so das OOS-Modell zu evaluieren. Abbildung 22 veranschaulicht die Generierung eines OOS-Modells mit Markt A als Testmarkt.

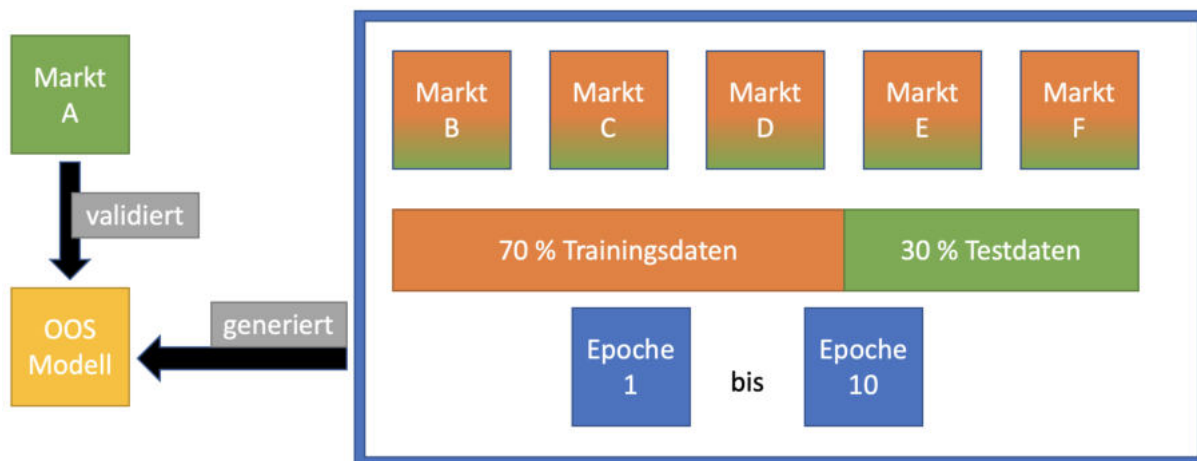


Abbildung 22: Aufteilung der Märkte in Trainings- und Testdaten

Die Trainingsdauer ist hierbei auf Zehn Epochen festgelegt. Diese Rahmenbedingungen gelten für alle OOS-Modellkonfigurationen. Für jede Konfiguration werden sechs OOS-Modelle evaluiert, sodass wie in Anhang B dargestellt nach dem Prinzip der *KFCV*-Methode jeder Markt die Rolle des Testmarktdatensatzes einnimmt. Diese Maßnahmen sollen dazu beitragen, die Qualität und Aussagekraft der Forschungsergebnisse zu erhöhen, da je Modellkonfiguration mehrere Ergebnisse in die Evaluierung einfließen.

4 OOS-Modelle

In diesem Kapitel liegt der Fokus auf der Implementierung der OOS-Modelle. Um einen Vergleich der verschiedenen OOS-Modelle zu ermöglichen, müssen einheitliche Bedingungen geschaffen werden. Dazu erläutert einleitend Kapitel 4.1 die Rahmenbedingungen für jedes KNN. Auf die Unterschiede in der Implementierung der Modellkonfigurationen geht Kapitel 4.2 ein. Nach der Definition der KNNs beschreibt Kapitel 4.3 die Implementierung des ML-Prozesses. Damit nicht das Training jedes Modells manuell gestartet werden muss, erklärt Kapitel 4.4 den automatisierten Durchlauf aller Modellkonfigurationsmöglichkeiten. Die Ergebnisse aller Modelle werden wie in Kapitel 4.5 beschrieben ebenfalls automatisch dokumentiert. Mithilfe der generierten Dokumentationen kann im nächsten Schritt die Evaluation der OOS-Modelle erfolgen.

4.1 Rahmenbedingungen für das KNN

Im nachfolgenden werden die Rahmenbedingungen für jedes KNN erläutert. Um die generierten OOS-Modelle vergleichen und die Ergebnisse unter identischen Umständen reproduzieren zu können, listet Kapitel 4.1.1 die genutzten Hardware-Komponenten und Kapitel 4.1.2 die verwendeten Software-Komponenten auf. Mit den in Kapitel 4.1.3 beschriebenen Parameter wurden die globalen Eigenschaften aller KNNs definiert. Abschließend erläutert Kapitel 4.1.4 die Architektur, auf der alle KNNs basieren.

4.1.1 Hardware-Komponenten

Da das Generieren von KNN rechenintensiv ist, wurde diese auf einem externen Server durchgeführt. Die zugehörigen Server-Spezifikationen sind in Tabelle 4 aufgelistet.

Server-Name	hetzner-ax41
Server-Art	Dediziertes Hosting
Arbeitsspeicher	8 GB
CPU	RYZEN 5-3600 3,6GHz (6 Threads, 3 Kerne)
GPU	Keine
Festplattenart	NVMe SSD
Verwendeter Festplattenspeicher	43,5 GB

Tabelle 4: Server-Spezifikationen

Der Anbieter des Servers nennt sich *Hetzner* und bietet mit dem Server *hetzner-ax41* eine *Dedicated-Hosting*-Dienstleistung an. Das bedeutet, dass im Gegensatz zu *Shared-Hosting*-Diensten ein Server für den Kunden alleine zur Verfügung steht. Durch die enthaltenen Administratorenrechte können Paketabhängigkeiten installiert und der zugehörige Source-Code leicht ausgeführt werden.¹³² Die Angabe zu der CPU-Leistung und dem Ar-

¹³²Vgl. Ackermann 2021, S. 495 f.

beitsspeicher beziehen sich auf die Limitierungen zum Trainieren des Modells. Definiert ist diese in der in Anhang C.1.2 dargestellten `docker-compose.yml`, welche die Ausführung des Programms auf maximal 8 GB Arbeitsspeicher und 3 CPU-Kerne begrenzt.

Die KNNs trainieren auf einer CPU. Alternativ können KNNs auch auf GPUs trainiert werden. Der Vorteil einer GPU ist, dass sich die Trainingsgeschwindigkeit von KNNs durch die parallele Architektur und die hohe Anzahl von Kernen erhöht.¹³³ Da jedoch der Fokus dieser Arbeit auf dem Vergleich verschiedener Modellkonfigurationen liegt, kann durch die Verwendung der gleichen Hardware eine Vergleichbarkeit der Trainingsgeschwindigkeit hergestellt werden.

Ein weiterer Aspekt ist der verwendete Festplattenspeicher. Den meisten Speicherplatz benötigten die drei verwendeten Datentöpfe. Einer dieser Datentöpfe enthält alle verfügbaren Bilddateien. Der zweite enthält darüber hinaus die augmentierten In-Stock-Bilder. Datentopf Nummer Drei enthält dynamisch die Marktdaten, die für das Training des aktuellen Modells verwendet werden. Der dafür benötigte Festplattenspeicher inklusive aller Programmdateien beläuft sich auf circa 43,5 GB, wobei sich dieser Wert je nach verwendeten Daten dynamisch ändert.

4.1.2 Software-Komponenten

Docker ermöglicht eine einfache systemunabhängige Verteilung und Ausführung von Anwendungen.¹³⁴ Mit dem Befehl `docker-compose up` wird ein Docker-Container hochgefahren, um das zugehörige Programm zu starten. Die verwendeten Bibliotheken sind im Anhang C.1.1 dargestellten `Dockerfile` beschrieben. In diesem ist zu erkennen, dass die Python Version 3.10 verwendet wird. Da jederzeit die Gefahr besteht, dass neue Versionen von Bibliotheken zu Kompatibilitätsproblemen mit anderen Bibliotheken führen können, ist in Listing 3 die `requirements.txt` dargestellt, welche die verwendeten Versionen zu jeder genutzten Bibliothek benennt.

```
openpyxl==3.0.10
PyYAML==6.0
scipy==1.10.0
tensorflow==2.11.0
```

Listing 3: Verwendete Bibliotheken für das OOS-Modell

Die Anwendung verwendet insgesamt vier Bibliotheken. Neben der *Tensorflow*-Bibliothek, die wiederum die *Keras*-Bibliothek enthält, muss zusätzlich die *scipy*-Bibliothek installiert werden, da *Tensorflow* viele numerische Berechnungen durchführt, die von *scipy* bereitgestellt werden.¹³⁵ ¹³⁶ Die allgemeine Konfiguration der KNNs sind

¹³³Vgl. Abadi u. a. 2016, S. 10 ff.

¹³⁴Vgl. Hopp und Stern 2021, S. 18 f.

¹³⁵Vgl. TensorFlow 2 2022a

¹³⁶Vgl. SciPy 2022

in einer yaml-Datei beschrieben. Für das Auslesen des Inhalts bietet sich die *PyYAML*-Bibliothek an.¹³⁷ Nach der Ermittlung der Ergebnisse eines OOS-Modells dokumentiert die Anwendung diese in einer Excel-Tabelle. Für das Erstellen, Befüllen und Speichern der Excel-Datei wurde die Bibliothek *openpyxl* verwendet.¹³⁸

4.1.3 KNN-Konfiguration

Um Konfigurationseinstellungen einfach für alle KNNs vornehmen zu können, wurde eine `config.yaml`-Datei entwickelt. Sie dient als zentrale Datei zur Einstellung aller Parameter zum Starten der Anwendung. Ein Ausschnitt der wichtigsten Konfigurationseinstellungen ist in Listing 4 zu finden.

```
bilder:
  original_path: "assets/Bilder/Datengrundlage/"
  #original_path: "assets/Bilder/Datengrundlage-Augmentiert/"
  knn_path: "assets/Bilder/AktuelleTrainingsUndTestdaten/"
knn:
  epochs: 10
  exception_activation_funktion_2: ['ReLU']
result:
  excel_path: "ressourcen/results/excel/"
```

Listing 4: Ausschnitt globale Konfigurationseinstellungen

Die drei eingerückten Zeilen unter dem Punkt `Bilder` definieren den Pfad zu den zuvor erläuterten Datentöpfen. Während der `original_path` den Pfad zur originalen Bilddatenquelle definiert, gibt der `knn_path` den Pfad zu den Bildern an, die für das nächste Training des KNNs aufbereitet werden.

Eine weitere Rahmenbedingung ist die Trainingsdauer. Jedes KNN hat Zehn Epochen Zeit, um seine Kostenfunktion optimal zu entwickeln. Dieser Wert basiert auf den Erfahrungen aus der vorangegangenen Projektarbeit. Das darin untersuchte KNN, welches große Ähnlichkeiten in der Struktur mit den hier trainierten KNNs aufweist, konnte bereits nach wenigen Epochen eine Konvergenz aufweisen.

Des Weiteren kann mit `exception_activation_funktion_2` definiert werden, welche Aktivierungsfunktionen in der letzten Schicht nicht verwendet werden sollen. Wie in Kapitel 2.3.5 erwähnt, ist das für die Aktivierungsfunktion ReLU relevant, da diese nicht für die Output-Schicht eines KNNs geeignet ist.

Jedes trainierte KNN unterscheidet sich in der Zusammensetzung der verwendeten Aktivierungsfunktionen, Optimierungsverfahren oder dem verwendeten Testmarkt. Die Ergebnisse sind in Excel-Tabellen dokumentiert. Diese befinden sich im durch `excel_path`

¹³⁷Vgl. PyYAML o.J.

¹³⁸Vgl. Openpyxl 2022

definierten Pfad. Damit in der Excel-Tabelle dokumentiert werden kann, welche Konfigurationseinstellungen zu welchem Ergebnis gehören, speichert die Anwendung die Konfigurationseinstellungen des aktuell betrachteten KNNs in der Klasse `ConfigKNN`. Die Zusammenstellung eines KNNs wird, wie in Listing 5 dargestellt, als Enum bei der Initialisierung der `ConfigKNN`-Klasse mitgegeben.

```
class ConfigKNN:
    def __init__(self,
                 excluded_folder=
                     Markt.Kein_Markt,
                 activation_function_128_units=
                     Aktivierungsfunktion.ReLU,
                 activation_function_2_units=
                     Aktivierungsfunktion.sigmoid,
                 optimization_method=
                     Optimierungsverfahren.SGD):
        self.excluded_folder = excluded_folder
        ...

    def __str__(self):
        return "optimization_method: " \
            + self.optimization_method.name + ...
```

Listing 5: Ausschnitt der `ConfigKNN`-Klasse

Der vollständige Sourcecode ist in Anhang C.5.1 zu finden. Die darin verwendeten Enums zu den Märkten, Aktivierungsfunktionen und Optimierungsverfahren sind in Anhang C.4 veranschaulicht. Der Parameter `excluded_folder` definiert den Markt, mit dem das OOS-Modell getestet wird. Der im Code definierte Standardwert ist `Kein_Markt`. Das hätte zur Folge, dass das KNN mit den Bildern aller Märkte trainiert. Standardwerte wurden definiert, um mögliche Fehler bei der Initialisierung einer Klasse zu vermeiden. Jedoch werden zur Laufzeit für den Parameter `excluded_folder` die Werte `Markt_A` bis `Markt_F` verwendet.

Als mögliche Werte für Aktivierungsfunktionen und Optimierungsverfahren kommen die im Kapitel 2.3 eingeführten Aktivierungsfunktionen Sigmoid, TanH, Softmax und ReLU sowie die im Kapitel 2.5 erläuterten Optimierungsverfahren SGD, Adam und FTRL infrage. Mit der `__str__()`-Methode werden die aktuellen KNN-Konfigurationen in der Konsole ausgegeben.

4.1.4 KNN-Architektur

Die Architektur der KNNs basiert auf dem in Kapitel 2.6.2 beschriebenen VGG16-Modell. Da das Modell für RGB-Bilder der Größe 224 x 224 Pixel optimiert wurde, werden die verwendeten OOS-Bilder auf diese Größe skaliert.

Beim Laden des VGG16-Modells wird lediglich die Output-Schicht ignoriert, da zwei eigene *Fully-Connected-Schichten* definiert wurden. Diese ermitteln den Output des OOS-Modells und legen mit ihrer Architektur die Rahmenbedingungen für die KNNs fest. Die erste der zwei Schichten besteht aus 128 Neuronen, die die Aktivierungsfunktion verwenden, welche im Parameter `activation_function_128_units` der `ConfigKNN`-Klasse definiert ist. Die letzte Schicht nutzt die im Parameter `activation_function_2_units` definierte Aktivierungsfunktion und fasst alle Stränge zu zwei Neuronen zusammen. In dieser Schicht wird letztendlich entschieden, ob es sich um ein OOS-Bild handelt. Dabei symbolisiert das eine Neuron die In-Stock-Klasse und das andere die OOS-Klasse.

Des Weiteren definiert die Variable `loss_fn` die in Kapitel 2.7.6 beschriebene Kostenfunktion *Binary-Crossentropy*. Zur Minimierung der Kostenfunktion verwendet das KNN das im Parameter `optimization_method` definierte Optimierungsverfahren. Um die Leistung von KNNs beurteilen zu können, werden die in Kapitel 2.7.2 beschriebene *Accuracy*- und die in Kapitel 2.7.4 erläuterte *Recall*-Klassifizierungsmetrik herangezogen. Nachfolgend stellt Listing 6 die Implementierung der Funktion `define_model` dar, die die Konfiguration der KNN-Architektur umfasst.

```
def define_model(config_knn):
    model = tf.keras.applications.VGG16(include_top=False,
        input_shape=(224, 224, 3))
    for layer in model.layers:
        layer.trainable = False

    flat = tf.keras.layers.Flatten()(model.layers[-1].output)
    output_layer_1 = get_next_layer(128, config_knn.
        activation_function_128_units, flat)
    output_layer_2 = get_next_layer(2, config_knn.
        activation_function_2_units, output_layer_1)
    model = tf.keras.models.Model(inputs=model.inputs,
        outputs=output_layer_2)

    opt = get_optimization_method(config_knn.
        optimization_method)
    loss_fn = tf.keras.losses.BinaryCrossentropy()
    recall_not_oos = tf.keras.metrics.Recall(name='
        recall_not_oos', class_id=0)
    recall_oos = tf.keras.metrics.Recall(name='recall_oos',
        class_id=1)

    model.compile(optimizer=opt, loss=loss_fn, metrics=['
        accuracy', recall_not_oos, recall_oos])
    return model
```

Listing 6: Architektur vom KNN

In Anhang C.6.1 ist eine kommentierte Version des Source-Codes zu finden. Das Laden

des bestehenden VGG16-Modells bringt den Vorteil, dass die darin enthaltenen Schichten nicht neu trainiert werden müssen. Stattdessen fokussiert sich das KNN auf das Trainieren der *Fully-Connected-Schichten*. Die Verbindung zwischen dem VGG16-Modell und den *Fully-Connected-Schichten* erfolgt durch die *Flat*-Schicht. Diese verwendet einen Tensor, der als ein multidimensionales mathematisches Objekt betrachtet werden kann.¹³⁹ Der Tensor wird benötigt, um ein dreidimensionales Array mit den Dimensionen Höhe, Breite und Tiefe in ein eindimensionales Array zu überführen.¹⁴⁰ Bei den *Fully-Connected-Schichten* handelt es sich um *Dense*-Schichten. Sie vernetzen alle *Output*-Neuronen der vorherigen Schicht mit den *Input*-Neuronen der jeweiligen *Dense*-Schicht.¹⁴¹

Die OOS-Erkennung legt den Schwerpunkt auf die möglichst vollständige Erkennung aller Lücken, weshalb eine FP-Erkennung weniger schlimm als eine FN-Erkennung ist. Aus diesem Grund wurde sich für die Nutzung der *Recall*- und gegen die Verwendung der *Precision*- und *F1-Score*-Klassifizierungsmetrik entschieden. Zudem wird die *Accuracy*-Klassifizierungsmetrik verwendet, um eine allgemeine Übersicht zu der Leistung der generierten OOS-Modelle zu erhalten.

4.2 Implementierung der OOS-Modellvarianten

Die Methoden `get_next_layer(...)` und `get_optimization_method(...)` aus Listing 5 definieren die Unterschiede in der Konfiguration zwischen den verschiedenen OOS-Modellen. Kapitel 4.2.1 erläutert den Aufbau von `get_next_layer(...)` und geht dabei auf die konfigurierbaren Aktivierungsfunktionen ein, während Kapitel 4.2.2 die Struktur von `get_optimization_method(...)` beschreibt und die Konfiguration der Optimierungsverfahren erläutert.

4.2.1 Implementierung der Aktivierungsfunktion in eine Netz-Schicht

Um das VGG16-Modell um die zwei *Fully-Connected-Schichten* zu erweitern, wird die in Listing 7 dargestellte Methode `get_next_layer(...)` verwendet. Diese benötigt die Neuronenanzahl, die Art der Aktivierungsfunktion und die aktuelle Struktur vom KNN.

```
def get_next_layer(units, activation_function, flat_or_dense):
    if activation_function == Aktivierungsfunktion.ReLU:
        return tf.keras.layers.Dense(units, activation=
            activation_function.name, kernel_initializer='
            he_uniform')(flat_or_dense)
    if activation_function == Aktivierungsfunktion.sigmoid:
        return tf.keras.layers.Dense(units, activation=
            activation_function.name)(flat_or_dense)
    if activation_function == Aktivierungsfunktion.tanh:
```

¹³⁹Vgl. Goodfellow, Bengio und Courville 2016, S. 31

¹⁴⁰Vgl. TensorFlow 2 2023d

¹⁴¹Vgl. TensorFlow 2 2023c

```

    return tf.keras.layers.Dense(units, activation=
        activation_function.name)(flat_or_dense)
if activation_function == Aktivierungsfunktion.softmax:
    return tf.keras.layers.Dense(units, activation=
        activation_function.name)(flat_or_dense)
return None

```

Listing 7: Generierung der nächsten Output-Schicht

Wie bereits in Listing 6 dargestellt, lautet die Anzahl der Neuronen für den Parameter `units` 128 oder 2. Des Weiteren können dem Parameter `activation_function` alle Werte übergeben werden, die im Enum `Aktivierungsfunktion` gelistet sind.

Jede Aktivierungsfunktion muss zu Beginn des Trainings mit einer Gewichtung je Neuron initialisiert werden. Dafür gibt es verschiedene Initialisierer, die für die Zuweisung der Anfangsgewichte verwendet werden können. *Tensorflow* definiert für den Parameter `kernel_init` als Standard den Wert `glorot_uniform`.¹⁴² Der Glorot-Initialisierer zieht Stichproben aus einer Gleichverteilung innerhalb von $[-limit, limit]$, wobei *Limit* mithilfe der Formel 25

$$\text{Limit} = \frac{6}{n_{in} + n_{out}} \quad (25)$$

berechnet wird. n_{in} benennt hierbei die Anzahl der eingehenden Neuronen, wohingegen n_{out} die Anzahl der ausgehenden Neuronen definiert.¹⁴³ Der Glorot-Initialisierer ist für die ReLU-Aktivierungsfunktion in tieferen neuronalen Netzen, wie dem VGG16-Modell, nicht geeignet. Die Varianz der Gewichte kann durch die eingehenden Neuronen besser berechnet werden, weshalb sich der Initialisierer `he_uniform` für die ReLU-Aktivierungsfunktion eignet.¹⁴⁴ Mithilfe der Formel 26

$$\text{Limit} = \frac{6}{n_{in}} \quad (26)$$

kann das *Limit* für den Initialisierer `he_uniform` berechnet werden.¹⁴⁵ Der letzte Parameter `flat_or_dense` definiert welche Art von Schicht dem KNN hinzugefügt wird. Der Name des Parameters verdeutlicht, dass die neue Schicht auf eine *Flat*- oder *Dense*-Schicht aufsetzt. Eine neue Schicht wird bei jedem Aufruf der Methode `get_next_layer(...)` erzeugt und dem KNN hinzugefügt. Schlussendlich gibt die Methode das angepasste KNN als Rückgabewert zurück.

¹⁴²Vgl. TensorFlow 2 2023a

¹⁴³Vgl. TensorFlow 2 2023a

¹⁴⁴Vgl. He u. a. 2015, S. 3

¹⁴⁵Vgl. TensorFlow 2 2023b

4.2.2 Implementierung von Optimierungsverfahren

Bereits in Kapitel 2.5 wurde die Vielzahl an Konfigurationsmöglichkeiten der einzelnen Optimierungsverfahren beschrieben. Aufgrund der vielen Einstellungsmöglichkeiten für die verschiedenen Parameter der Optimierungsverfahren in *Tensorflow*, beschränkt sich diese Arbeit mit wenigen Ausnahmen auf die Standardkonfigurationen.

Jedoch unterscheiden sich die Standardwerte für die Lernrate der verschiedenen Optimierungsverfahren in *Tensorflow*. Während Adam und FTRL beispielsweise eine Standardlernrate von 0.001 haben, liegt diese für SGD bei 0.01.¹⁴⁶ ¹⁴⁷ Um eine einheitliche Grundlage für alle Optimierungsverfahren zu schaffen, wurde die Lernrate für alle mithilfe des Parameters `learning_rate` auf 0.001 gesetzt. Eine niedrige Lernrate führt zwar zu einer langsamen Konvergenz, stellt allerdings sicher, dass das optimale Minimum erreicht wird, was im Rahmen dieser Arbeit im Vordergrund steht.

Um die Geschwindigkeit der Konvergenz von SGD trotzdem zu erhöhen, wurde der Parameter `momentum` aktiviert. Dieser ist Standardmäßig deaktiviert.¹⁴⁸ Aufgrund der weiten Verbreitung in der Praxis wurde der Wert 0.9 für das `momentum` gewählt.¹⁴⁹ Die Implementierung in Listing 8 zeigt, wie die Optimierungsverfahren bestimmt und initialisiert werden.

```
def get_optimization_method(optimization_method):
    if optimization_method == Optimierungsverfahren.SGD:
        return tf.keras.optimizers.SGD(learning_rate=0.001,
                                         momentum=0.9)
    if optimization_method == Optimierungsverfahren.Adam:
        return tf.keras.optimizers.Adam(learning_rate=0.001)
    if optimization_method == Optimierungsverfahren.Ftrl:
        return tf.keras.optimizers.Ftrl(learning_rate=0.001)
    return None
```

Listing 8: Ermittlung des aktuellen Optimierungsverfahrens

Die Methode `get_optimization_method(...)` benötigt als Parameter ein Enum vom Typ `Optimierungsverfahren` und liefert den zugehörigen Optimierer als initialisierte Klasse zurück.

4.3 Implementierung des ML-Prozesses

Der Fokus dieses Kapitels liegt auf der Generierung von OOS-Modellen auf Basis von KNNs, die den ML-Prozess durchlaufen. Ein KNN kann in die Trainingsphase übergehen, sobald deren Architektur mithilfe einer Modellkonfiguration definiert wurde. Wie

¹⁴⁶Vgl. TensorFlow 2 2023f

¹⁴⁷Vgl. TensorFlow 2 2023g

¹⁴⁸Vgl. TensorFlow 2 2023h

¹⁴⁹Vgl. Goodfellow, Bengio und Courville 2016, S. 294

das Training durchgeführt wird, ist in Kapitel 4.3.1 beschrieben. Nach dem Abschluss des Trainings spricht man vom generierten OOS-Modell. Dieses OOS-Modell geht anschließend in die Testphase über, welche Kapitel 4.3.2 näher erläutert.

4.3.1 Implementierung des Modelltrainings

Das Training eines KNNs besteht aus mehreren Epochen, die jeweils aus einer Trainingsphase gefolgt von einer Testphase bestehen. In der dazu verwendeten Methode `fit_model()` in Listing 10 werden viele Variablen verwendet, die in den vorherigen Codeausschnitten eingeführt wurden. Dazu zählt der `config` Parameter, welcher die global definierten Konfigurationseigenschaften aus der `config.yaml` enthält. Zudem ist es erforderlich, das zuvor definierte KNN im Parameter `model` mitzugeben. Darüber hinaus benötigt die Methode die Parameter `train_it` und `test_it`, die die Trainings- und Testdaten beinhalten. Neu ist der letzte Parameter `callback`. Dieser Parameter wird nach jeder Epoche um die Informationen über die Modelleistung und die Trainingsdauer erweitert. An einem vereinfachten Beispiel veranschaulicht Listing 9 wie mithilfe der Klasse `CustomCallback` die Trainingsdauer und die *Accuracy* des OOS-Modells auf die Trainings- und Testdaten pro Epoche in einem Array gespeichert werden.

```
class CustomCallback(tf.keras.callbacks.Callback):
    def __init__(self):
        super().__init__()
        self.epoch_training_accuracies = None
        self.epoch_test_accuracies = None
        self.epoch_durations = None

    def on_train_begin(self, logs=None):
        self.epoch_training_accuracies = []
        self.epoch_test_accuracies = []
        self.epoch_durations = []

    def on_epoch_begin(self, epoch, logs=None):
        self.epoch_start_time = time.time()

    def on_epoch_end(self, epoch, logs=None):
        if logs is None:
            logs = {}
        self.epoch_training_accuracies.append(logs.get('
            accuracy'))
        self.epoch_test_accuracies.append(logs.get('
            val_accuracy'))
        self.epoch_durations.append(time.time() - self.
            epoch_start_time)
```

Listing 9: Speichern von Informationen während der Trainingsphase

Die Klasse `CustomCallback` erbt von der Klasse `tf.keras.callbacks.Callback` und überschreibt somit vorhandene Methoden. Nach der Initialisierung der Klasse werden zu Beginn des Trainings in der Methode `on_train_begin` die Variablen mit einem leeren Array initialisiert. Mit dem Start einer neuen Epoche wird die Methode `on_epoch_begin` aufgerufen. Folglich wird der Epochenstartzeitpunkt in der Variable `epoch_start_time` festgehalten. Sobald eine Epoche endet, wird die Methode `on_epoch_end` ausgeführt. In dieser wird das erreichte Ergebnis der *Accuracy*-Klassifizierungsmetrik für die Trainings- und Testdaten in den Arrays der entsprechenden Variablen festgehalten. Die Epochen-dauer wird in diesem Schritt ebenfalls ermittelt, indem der Epochenstartzeitpunkt vom Epochenendzeitpunkt subtrahiert wird. Das Ergebnis wird dem Array der Variable `epoch_durations` hinzugefügt.

Der vollständige Sourcecode zu der Klasse `CustomCallback` befindet sich im Anhang C.5.2. In dieser werden die ermittelten Informationen um die *Recall*-Klassifizierungsergebnisse und die Kostenfunktionsergebnisse erweitert. Die gesammelten Daten werden für die spätere Dokumentation und Auswertung der OOS-Modellergebnisse verwendet. Der in der Methode `fit_model(...)` definierte Parameter `callback` ist der Klasse `CustomCallback` zugehörig und wird für das Speichern aller Informationen nach jeder Epoche verwendet. Nachdem alle benötigten Parameter für die Methode `fit_model(...)` erläutert wurden, stellt Listing 10 die Konfigurationen für das Training des Modells dar.

```
def fit_model(config, model, train_it, test_it, callback):
    model.fit(
        train_it,
        steps_per_epoch=len(train_it),
        validation_data=test_it,
        validation_steps=len(test_it),
        epochs=config["knn"]["epochs"],
        verbose=1,
        callbacks=[callback]
    )
```

Listing 10: OOS-Modell trainieren

Mit der Ausführung der Methode `model.fit()` beginnt das Training des Modells. Der erste Parameter `train_it` definiert die Trainingsdatensätze. Um in einer Epoche mit allen Daten zu trainieren, müssen alle *Mini-Batches* aus den Trainingsdaten verwendet werden. Die Anzahl der im Training verwendeten *Mini-Batches* entspricht der Länge von `train_it`. Folglich wird `steps_per_epoch` mit der Anzahl der vorhandenen *Mini-Batches* gleichgesetzt. Nach jeder Epoche wird das trainierte Modell getestet. Verwendet wird hierfür das angegebene Testdatenset `test_it`. Hierbei muss ebenfalls die Länge von `test_it` bestimmt werden, um mit allen Testdaten testen zu können.

Während einer Epoche wird zu jedem Trainingsbild ein augmentiertes Bild erstellt.

Die Anzahl der Epochen lässt sich durch das Auslesen der globalen Konfigurationseinstellungen aus der `config.yaml` bestimmen. Da es sich bei der Trainingsphase um einen längeren Prozess handelt, bietet es sich an den Parameter `Verbose` auf 1 zu setzen, damit die Konsolenausgabe die Fortschrittsanzeige für jede Epoche und jeden Epochenschritt darstellt. Auch gibt die Konsole für jede Epoche die Ergebnisse der Klassifizierungsmetriken und der Kostenfunktion aus.¹⁵⁰ Die Methode `fit_model()` liefert kein Ergebnis zurück, da Python einen *Call-By-Object-Reference* Ansatz verfolgt.¹⁵¹ Dadurch verweisen die übergebenen Objekte nur auf den Speicherort. Dementsprechend wirkt sich eine Änderung des Objekts innerhalb der Funktion auch auf das Objekt außerhalb der Funktion aus, weshalb das `model` nicht erneut zurückgegeben werden muss.

4.3.2 Implementierung der Modellvalidierung

Um die Leistung des trainierten OOS-Modells zu überprüfen, wird ein Testmarkt herangezogen. Dieser wurde während der Trainingsphase nicht verwendet. Für das Laden der Bilder wird die gleiche Methode `flow_from_directory(...)` verwendet, wie für das Modelltraining. Allerdings wird der Parameter `subset` nicht definiert, weshalb alle Daten des Marktes geladen werden. Die hierzu implementierte Methode `get_markt_data(...)` ist im Anhang C.3.2 zu finden.

Die Untersuchung der Vorhersageleistung eines OOS-Modells erfolgt mithilfe der in Listing 11 dargestellten Methode `evaluate_model(...)`. Beim ersten Parameter `model` handelt es sich um das trainierte OOS-Modell. Der zweite Parameter `test_it` ist optional und findet nur Verwendung, wenn kein Testmarkt definiert wurde. Für die Generierung der verschiedenen OOS-Modelle ist dieser Parameter nicht notwendig. Zusätzlich werden die Parameter `datagen`, `config` und `config_knn` benötigt. Während `datagen` für den Aufruf der Methode `flow_from_directory(...)` benötigt wird, ermitteln `config` und `config_knn` den Pfad zum definierten Testmarkt. Der Pfad setzt sich aus der in der `config.yaml` konfigurierten Datenbasis sowie dem Namen des Testmarktes zusammen, der aus der Variable `excluded_folder` der Klasse `ConfigKNN` ausgelesen wird.

```
def evaluate_model(model, test_it, datagen, config,
                   config_knn):
    if config_knn.excluded_folder != Markt.Kein_Markt:
        test_it = get_markt_data(datagen, config["bilder"]["
            original_path"] + config_knn.excluded_folder.name)
    all_metrics = model.evaluate(test_it, steps=len(test_it),
                                verbose=1)
    loss, accuracy, recall_not_oos, recall_oos = all_metrics
    print('loss: %.3f' % loss + ' - accuracy: %.3f' %
          accuracy + ' - recall_not_oos: %.3f' % recall_not_oos)
```

¹⁵⁰Vgl. TensorFlow 2 2023e

¹⁵¹Vgl. Python Software Foundation 2023a

```

+ ' - recall_oos: %.3f' % recall_oos)
return all_metrics

```

Listing 11: OOS-Modell evaluieren

Die Methode `evaluate(...)` benötigt ähnlich wie die Methode `fit(...)` zunächst den Testdatensatz, die Anzahl der *Mini-Batches* und die Einstellung des Parameters `Verbose` zur Fortschrittsanzeige. Nach Abschluss des Tests gibt die Methode `evaluate(...)` das Ergebnis der Kostenfunktion, sowie die Ergebnisse der definierten Klassifizierungsmetriken als Array zurück.¹⁵² Zur besseren Visualisierung der Ergebnisse werden diese mithilfe der Definition des Parameters `Verbose=1` auf der Konsole ausgegeben. Die ermittelten Testergebnisse `all_metrics` werden abschließend von der Methode `evaluate_model(...)` bereitgestellt.

4.4 Generierung von OOS-Modellen

Die Generierung von allen betrachteten OOS-Modellen umfasst mehrere Schritte, die aufeinander aufbauen. Zunächst werden die Trainingsdaten auf Grundlage der definierten Konfigurationen zusammengestellt, wie in Kapitel 4.4.1 näher erläutert wird. Um diese Aufgabe in einem Programm automatisch für alle betrachteten Konfigurationsmöglichkeiten durchzuführen, beschreibt Kapitel 4.4.2 die Implementierung der OOS-Modelle auf Grundlage der *Cross-Validation*-Methode. Abschließend wird in Kapitel 4.4.3 auf einer abstrakten Ebene beschrieben, wie die zuvor eingeführten Methoden bei der Generierung eines einzelnen OOS-Modells im Zusammenhang stehen.

4.4.1 Zusammenstellung der Trainingsdaten

Vor der Zusammenstellung der Trainingsdaten wird zunächst die Ordnerstruktur des Quell- und Zielverzeichnis erläutert. Das Quellverzeichnis ist durch den in der `config.yaml` definierten `original_path` beschrieben, wohingegen der `knn_path` das Zielverzeichnis benennt. Am Beispiel vom Quellverzeichnisordner *Datengrundlage* und dem Zielverzeichnisordner *AktuelleTrainingsUndTestdaten* stellt Abbildung 23 die Unterschiede zwischen den Ordnerstrukturen dar:

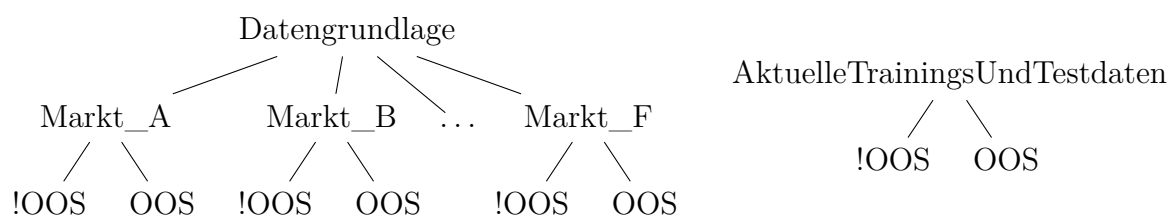


Abbildung 23: Quell- und Zielverzeichnisstruktur

¹⁵²Vgl. TensorFlow 2 2023e

Innerhalb des Ordners *Datengrundlage* sind die Bilder nach In-Stock und OOS je Markt sortiert. Im Unterschied dazu differenziert die Ordnerstruktur *AktuelleTrainings-UndTestdaten* lediglich zwischen OOS und nicht OOS, da hier alle für das KNN verwendeten Bilder unabhängig vom jeweiligen Markt zusammengeführt werden. Eine Ausnahme bildet hierbei der Testmarkt, dessen Daten erst nach der Generierung des OOS-Modells zum testen verwendet werden. Nachfolgend stellt Listing 31 die technische Umsetzung der Zusammenführung der Bilder dar.

```
def copy_images_with_exclusion(src_dir, dest_dir, exclude_dir
=None):
    for file_or_dir in os.listdir(dest_dir):
        file_or_dir_path = os.path.join(dest_dir, file_or_dir
)
        if os.path.isdir(file_or_dir_path):
            shutil.rmtree(file_or_dir_path)
        else:
            os.remove(file_or_dir_path)

    os.makedirs(os.path.join(dest_dir, '!OOS'))
    os.makedirs(os.path.join(dest_dir, 'OOS'))

    for subdir in os.listdir(src_dir):
        subdirString = bytes(subdir, 'utf-8').decode('
unicode_escape')
        exclude_dirString = bytes(exclude_dir.name, 'utf-8').
decode('unicode_escape')
        if subdirString == exclude_dirString:
            continue
        for folder in os.listdir(os.path.join(src_dir, subdir
)):
            if folder == 'OOS':
                for img in os.listdir(os.path.join(src_dir,
subdir, folder)):
                    shutil.copy(os.path.join(src_dir, subdir,
folder, img), os.path.join(dest_dir,
'OOS'))
            elif folder == '!OOS':
                for img in os.listdir(os.path.join(src_dir,
subdir, folder)):
                    shutil.copy(os.path.join(src_dir, subdir,
folder, img), os.path.join(dest_dir,
'!OOS'))
```

Listing 12: Zusammenstellung der aktuellen Trainingsdaten

Eine kommentierte Version des Sourcecodes ist in Anhang C.6.2 zu finden. Die Methode `copy_images_with_exclusion(...)` benötigt als Parameter das Quell- und Zielverzeichnis, sowie den definierten Testmarkt. Zunächst werden alle Dateien und Ordner

mittels einer `for`-Schleife im Zielverzeichnis `dest_dir` gelöscht. Anschließend werden die Ordner `OOS` und `!OOS` im Zielverzeichnis erstellt.

Im nächsten Schritt werden in einer Schleife alle Unterverzeichnisse des Quellverzeichnisses `src_dir` durchlaufen, wobei das Unterverzeichnis `exclude_dir` ausgeschlossen wird. Hierzu werden die Strings vom aktuellen Ordernamen und dem Namen des Testmarktes zunächst ins UTF-8-Format konvertiert und anschließend miteinander verglichen. Falls die beiden Strings identisch sind, wird der aktuelle Ordner nicht berücksichtigt. Andernfalls werden zuerst alle Bilder aus dem Quellverzeichnisordner `OOS` in den Zielverzeichnisordner `OOS` kopiert. Abschließend kopiert die Methode den Ordner `!OOS` nach dem gleichen Vorgehen. Somit enthält der Ordner *Aktuelle TrainingsUndTestdaten* alle für das Modelltraining relevanten Daten.

4.4.2 Validierung der Modellkonfigurationen mittels Cross-Validation

Die automatisierte Zusammenstellung der Trainings- und Testdaten für jedes OOS-Modell erfolgt mithilfe der implementierten *Cross-Validation*-Methode. Um die bestmögliche Modellkonfiguration zu finden, wurde eine automatisierte sequentielle Abarbeitung aller Aktivierungsfunktionen und Optimierungsverfahren implementiert.

Die hierzu entwickelte Methode `run_cross_validation(...)` lässt sich in zwei Abschnitte gliedern. Der erste Teil ist in Listing 13 dargestellt und beschreibt die Zusammenstellung der verschiedenen Modellkonfigurationen.

```
def run_cross_validation():
    with open("ressources/config/config.yaml", "r") as file:
        config = yaml.load(file, Loader=yaml.FullLoader)

    for optimization_method in Optimierungsverfahren:
        for activation_function_2 in Aktivierungsfunktion:
            if activation_function_2.name in config['knn']['exception_activation_funktion_2']:
                continue
            for activation_function_128 in
                Aktivierungsfunktion:
                ...
```

Listing 13: Iterieren über alle Modellkonfigurationen

Als erstes lädt die Methode die Datei `config.yaml`. Anschließend durchläuft sie mehrere verschachtelte `For`-Schleifen. Zunächst iteriert die äußere `For`-Schleife über alle Optimierungsverfahren. Innerhalb dieser Schleife werden in einer weiteren `For`-Schleife alle Aktivierungsfunktionen durchlaufen, wobei auch geprüft wird, ob es sich um die in der `exception_activation_funktion_2` definierten Aktivierungsfunktion handelt. Falls ja, wird diese Konfiguration übersprungen. Andernfalls wird in einer dritten `For`-Schleife

erneut über alle Aktivierungsfunktionen iteriert. Somit ist sichergestellt, dass für jedes Optimierungsverfahren mit jeder möglichen Aktivierungsfunktionspaarung in den *Fully-Connected-Schichten* eine Modellgenerierung stattfindet. Nun geht es im nächsten Schritt darum, für jede festgelegte Modellkonfiguration eine *Cross-Validierung* durchzuführen. Diese ist in Listing 14, dem zweiten Teil der Methode `run_cross_validation(...)`, dargestellt.

```

...
model_evaluate_metrics = []
training_duration_models = []
workbook = get_excel_workbook()
worksheet = get_excel_worksheet(workbook)
r = 1
config_knn = ConfigKNN(
    excluded_folder=Markt.Kein_Markt,
    activation_function_2_units=activation_function_2,
    activation_function_128_units=activation_function_128,
    optimization_method=optimization_method
)
for excluded_folder in Markt:
    if excluded_folder == Markt.Kein_Markt:
        continue
    config_knn.excluded_folder = excluded_folder
    r, evaluate_metrics, training_duration_model = run_model(
        config, config_knn, worksheet, r)
    model_evaluate_metrics.append(evaluate_metrics)
    training_duration_models.append(training_duration_model)
average_evaluate_cross_validation(worksheet, r,
    model_evaluate_metrics, training_duration_models)
save_excel(workbook, config, config_knn)

```

Listing 14: Implementierung der Cross-Validation-Methode

Für die *Cross-Validierung* werden je Modellkonfiguration sechs Modelle generiert. Die Ergebnisse aller Modelle werden in den Arrays `model_evaluate_metrics` und `training_duration_models` festgehalten. Zudem erfolgt die Initialisierung von Variablen, welche für die Dokumentation relevant sind. Dazu zählt die Initialisierung einer Excel-Datei, dem sogenannten `workbook`, und das Festlegen des Blattes innerhalb der Excel-Datei, bezeichnet als `worksheet`. Die dafür verwendeten Methoden sind in Anhang C.7.1 definiert. Die Variable `r` definiert die nächste freie Zeile im Arbeitsblatt.

Im nächsten Schritt werden die Modellkonfigurationen in der Variable `config_knn` festgehalten. Schließlich wird der Testmarkt definiert, welcher gemäß der Regeln der *Cross-Validierung* für jedes Modell variiert. Hierbei wird mithilfe einer `For`-Schleife über das Enum `Markt` iteriert und alle möglichen Testmärkte, mit Ausnahme von *Kein_Markt*, bestimmt. Nach der Definition aller relevanten Konfigurationen werden die OOS-Modelle

mithilfe der Methode `run_model(...)` generiert, welche im nachfolgenden Kapitel näher erläutert wird.

Nach dem Abschluss der *Cross-Validierung* wird ein Durchschnitt über alle ermittelten OOS-Modellergebnisse gebildet. Das Ergebnis wird im `worksheet` dokumentiert und die zugehörige Excel-Datei mithilfe der in Anhang C.7.2 dargestellten Methode gespeichert. Die genaue Vorgehensweise zur Dokumentation erläutert Kapitel 4.5. Durch die verschiedenen Modellkonfigurationsmöglichkeiten entstehen so insgesamt 36 Excel-Dateien. Aufgrund der *Cross-Validierung* wurden insgesamt 216 OOS-Modelle generiert und evaluiert.

Eine kommentierte Version der Methode `run_cross_validation(...)`, die die Zusammenführung beider Teile nochmals veranschaulicht, ist in Anhang C.6.3 zu finden. Diese Methode wird zum Starten der Anwendung aufgerufen.

4.4.3 Zentrale Methode zur Generierung eines OOS-Modells

Viele der bisher definierten Methoden werden in der Methode `run_model(...)` aufgerufen. Diese wird in Listing 15 veranschaulicht. Für den Aufruf der Methode werden vier Parameter benötigt. Bei dem ersten Parameter `config` handelt es sich um die eingelesene Datei `config.yaml`. Der zweite Parameter `config_knn` beschreibt die aktuellen Modellkonfigurationen. Der dritte Parameter `worksheet` wird verwendet, um die Ergebnisse in der Excel-Tabelle zu dokumentieren. Der letzte Parameter `r` definiert in welcher Zeile in der Excel-Tabelle die Dokumentation fortgesetzt wird.

```
def run_model(config, config_knn, worksheet, r):
    copy_images_with_exclusion(config["bilder"]["
        original_path"], config["bilder"]["knn_path"],
        config_knn.excluded_folder)
    callback = CustomCallback()

    start_time = datetime.now()
    model = define_model(config_knn)
    datagen = define_augmentation_rules()
    train_it = get_train_data(datagen, config["bilder"]["
        knn_path"])
    test_it = get_test_data(datagen, config["bilder"]["
        knn_path"])
    fit_model(config, model, train_it, test_it, callback)
    evaluate_metrics = evaluate_model(model, test_it, datagen
        , config, config_knn)
    end_time = datetime.now()

    training_duration_model = end_time - start_time
    training_duration_model_str = format(
        training_duration_model)
```



```
r = create_excel_result(worksheet, r, callback, config,
                        config_knn, evaluate_metrics, training_duration_model)
return r, evaluate_metrics, training_duration_model
```

Listing 15: Hauptmethode zur Generierung eines OOS-Modells

Die Methode stellt zunächst die Trainings- und Testdaten zusammen und erstellt ein Objekt der Klasse `CustomCallback`. Nachfolgend startet die Messung der Zeit, die für die Erstellung eines OOS-Modells benötigt wird.

Daraufhin wird das KNN definiert und die Daten vorbereitet. Hierzu werden mithilfe der definierten Augmentationsregeln die Trainings- und Testdaten geladen. Nun sind alle Informationen vorhanden, die für das Training des KNNs benötigt werden. Nach dem Training des KNNs wird das generierte OOS-Modell auf seine Leistung getestet. Mit dem Abschluss des Tests wird auch die Messung gestoppt.

Die Informationen zum OOS-Modell umfassen die Gesamtdauer des Trainings und Tests, die Resultate der Klassifizierungsmetriken und der Kostenfunktion für jede Epoche, sowie die Testergebnisse des Testmarktes. Diese Informationen werden in der Excel-Tabelle mithilfe der Methode `create_excel_result(...)` aus Kapitel 4.5.1 dokumentiert.

Abschließend gibt die Methode die Zeilennummer zurück, an der die Dokumentation des nächsten OOS-Modells in der Excel-Tabelle fortgesetzt werden kann. Zusätzlich werden die Ergebnisse der Metriken sowie die Dauer des Trainings zurückgegeben. Zur Verbesserung des Verständnisses der verwendeten Methoden und Zusammenhänge stellt Anhang C.6.4 eine kommentierten Version der Methode `run_model(...)` zur Verfügung.

4.5 Dokumentation der Leistung von OOS-Modellen

Im nachfolgenden geht es um die Beschreibung der automatisierten Dokumentation von Modellergebnissen. Zunächst erläutert Kapitel 4.5.1 wie die Ergebnisse eines einzelnen OOS-Modells in einer Excel-Tabelle dokumentiert werden. Da zu einer Modellkonfiguration für die *Cross-Validierung* sechs verschiedene Modelle generiert werden, schildert Kapitel 4.5.2 die Konsolidierung der Ergebnisse, um eine Zusammenfassung über die Leistung der Modellkonfiguration zu liefern.

4.5.1 Protokollieren der Ergebnisse eines OOS-Modells

Bereits zuvor wurde der Aufruf der Methode `create_excel_result(...)` gezeigt. Wie der Name bereits verrät, dient diese Methode dazu, die ermittelten Ergebnisse in einer Excel-Tabelle zu erfassen. Da die Ergebnisse von mehreren Modellen in einer Excel-Tabelle dokumentiert werden, erwartet die Methode als Parameter das aktuelle `worksheet` sowie

die Zeilennummer `r`, an der die Dokumentation fortgesetzt werden soll. Zudem benötigt die Methode Informationen zu den Ergebnissen des OOS-Modells, die während der Trainingsphase durch den `CustomCallback`-Parameter `callback` gesammelt werden.

Darüber hinaus sind die im Parameter `config` enthaltenen globalen Konfigurationen erforderlich, sowie die im Parameter `config_knn` definierten Modellkonfigurationen. Die Ergebnisse des Testmarktes sind im Parameter `evaluate_metrics` gespeichert. Der letzte Parameter `training_duration_model` definiert die benötigte Dauer zum Generieren des OOS-Modells.

Mit diesen Informationen erfolgt die in Anhang C.7.3 dargestellte Dokumentation der Ergebnisse der OOS-Modelle. Eine vereinfachte Version der Methode ist zur Veranschaulichung der Zusammenhänge in Listing 16 abgebildet.

```
def create_excel_result(worksheet, r, callback, config,
                        config_knn, evaluate_metrics, training_duration_model):
    worksheet.column_dimensions["A"].width = 10
    ...
    worksheet.cell(row=r, column=2).value = '
        Optimierungsverfahren'
    ...
    r = r + 1
    worksheet.cell(row=r, column=2).value = config_knn.
        optimization_method.name
    ...
    r = r + 2
    ...
    worksheet.cell(row=r, column=3).value = '
        training_accuracy'
    ...
    r = r + 1
    for i, (loss, accuracy, recall_oos, recall_not_oos,
            val_loss, val_accuracy, val_recall_oos,
            val_recall_not_oos, duration) in enumerate(
        zip(callback.epoch_training_losses, callback.
            epoch_training_accuracies,
            callback.epoch_training_recall_oos, callback.
            epoch_training_recall_not_oos, callback.
            epoch_test_losses, callback.
            epoch_test_accuracies, callback.
            epoch_test_recall_oos, callback.
            epoch_test_recall_not_oos, callback.
            epoch_durations)):
        ...
        worksheet.cell(row=r + i, column=3).value = accuracy
        ...
    r = r + config["knn"]["epochs"] + 1
    worksheet.cell(row=r, column=2).value = 'Trainingsdauer'
```

```

worksheet.cell(row=r, column=3).value =
    training_duration_model
r = r + 3
...
worksheet.cell(row=r, column=3).value = 'accuracy'
...
r = r + 1
...
worksheet.cell(row=r, column=3).value=evaluate_metrics[1]
...
r = r + 4
return r

```

Listing 16: Ausschnitt vom Schreiben der Modellergebnisse in einer Excel-Tabelle

In der ersten Zeile wurde die Beschreibung der einzelnen Modellkonfigurationen nach Festlegung der Zellenbreite für die verwendeten Spalten eingefügt. Die dazugehörigen Werte wurden in der Zeile darunter eingetragen.

Zwei Zeilen darunter beginnt der Abschnitt, der die Ergebnisse je Epoche enthält. Die Spalten sind wie folgt beschriftet:

- *epoch*
- *training_loss*
- *training_accuracy*
- *training_recall_oos*
- *training_recall_not_oos*
- *duration*
- *test_loss*
- *test_accuracy*
- *test_recall_oos*
- *test_recall_not_oos*

Die *duration* bezieht sich auf die Dauer einer Epoche. Die zugehörigen Werte werden für jede Epoche untereinander aufgelistet. Nachdem zwei Zeilen darunter die Gesamtdauer für die Generierung des OOS-Modells protokolliert wurde, beginnt die Dokumentation der Ergebnisse aus dem Testmarktes. Dabei wird der verwendete Testmarkt, die Klassifizierungsmetriken und die Kostenfunktion gefolgt vom jeweiligen Ergebnis angegeben. Damit endet die Dokumentation der Ergebnisse des OOS-Modells. Da die Dokumentation aller Modelle mit derselben Modellkonfiguration in einer Excel-Tabelle erfolgt, wird *r* um vier erhöht, um eine klare Abgrenzung zwischen den Ergebnissen der verschiedenen OOS-Modelle zu schaffen.

Aufgrund der weiteren Verwendung von *r* ist eine explizite Rückgabe der Variable notwendig, da es sich bei der Parameterübergabe um einen primitive Datentypen handelt. Dieser wird im Gegensatz zu Objekten kopiert und enthält folglich keine Referenz auf die Speicherstelle.¹⁵³

¹⁵³Vgl. Python Software Foundation 2023b

4.5.2 Kompakte Darstellung der Ergebnisse einer Modellkonfiguration

Um die verschiedenen Modellkonfigurationen vergleichen zu können, wird für jede Modellkonfiguration der Mittelwert der Leistung über die sechs generierten OOS-Modelle gebildet. Als Datengrundlage dienen die mit dem Testmarkt ermittelten Ergebnisse. Für die Berechnung der Durchschnittswerte wurde die in Listing 17 abgebildete Hilfsmethode `sum_values_into_one_array(...)` verwendet. Eine genauere Beschreibung des Sourcecodes ist in Anhang C.7.4 zu finden. Als Parameter benötigt die Methode die Ergebnisse aller Metriken der OOS-Modelle, die sich als Array im Array `model_evaluate_metrics` wiederfinden. Zudem ist das Array `training_duration_models` mit den Zeiten für die Generierung der einzelnen OOS-Modelle erforderlich.

```
def sum_values_into_one_array(model_evaluate_metrics,
                             training_duration_models):
    sum_array = [0, 0, 0, 0, 0]
    num_arrays = len(model_evaluate_metrics)
    for evaluate_metric in model_evaluate_metrics:
        for i in range(len(evaluate_metric)):
            sum_array[i] += evaluate_metric[i]
    avg_array = [value / num_arrays for value in sum_array]

    add_times = timedelta()
    for training_time in training_duration_models:
        add_times += training_time
    average_duration = add_times / len(
        training_duration_models)
    avg_array[4] = average_duration
    return avg_array
```

Listing 17: Ausschnitt zur Berechnung des Durchschnitts der OOS-Modellergebnisse

Die Variable `sum_array` repräsentiert die vier Metriken und die Generierungsdauer aller OOS-Modelle. Sie enthält die Summe der einzelnen Klassifizierungsmetriken und wird anschließend verwendet, um sie durch die in `num_arrays` definierte Anzahl von OOS-Modellen zu dividieren. Daraus entsteht die neue Variable `avg_array`, die die berechneten Mittelwerte aller Klassifizierungsmetriken enthält.

Das Berechnen des Mittelwerts für die Generierungsdauer erfolgt durch das Aufsummieren der Zeiten in `add_times`. Der ermittelte Durchschnittswert wird an der letzten Stelle des Arrays `avg_array` gespeichert. Im letzten Schritt liefert die Methode die errechneten Mittelwerte als Ergebnis zurück.

Anschließend werden die berechneten Mittelwerte, wie bereits in Listing 14 veranschaulicht, durch die Methode `average_evaluate_cross_validation(...)` protokolliert. Listing 18 zeigt einen Ausschnitt dieser Methode. Der vollständige Sourcecode ist in Anhang C.7.5 zu finden. Der Methode müssen die Parameter `worksheet`, `r`,

`model_evaluate_metrics` und `training_duration_models` übergeben werden.

```
def average_evaluate_cross_validation(worksheet, r,
    model_evaluate_metrics, training_duration_models):
    total_information = sum_values_into_one_array(
        model_evaluate_metrics, training_duration_models)
    ...
    worksheet.cell(row=r, column=3).value='accuracy'
    ...
    r = r + 1
    ...
    worksheet.cell(row=r, column=3).value=total_information
    [1]
    ...
```

Listing 18: Ausschnitt der Dokumentation der Mittelwtergebnisse von OOS-Modellen

Als erstes ermittelt die Methode die Mittelwerte mithilfe der zuvor eingeführten Hilfsmethode `sum_values_into_one_array(...)`. Danach werden die Überschriften für die ermittelten Ergebnisse im `worksheet` definiert. Abschließend werden die zugehörigen Ergebnisse aus der Variable `total_information` in die jeweilige Zelle eingetragen. Damit ist die vollständige Dokumentation der Ergebnisse aller OOS-Modelle der jeweiligen Modellkonfiguration abgeschlossen.

5 Evaluation

In diesem Kapitel werden die Stärken und Schwächen der Modellkonfigurationen hinsichtlich der OOS-Erkennung untersucht. Die unveränderten Bilder dienen als Datengrundlage für die 216 zu generierenden OOS-Modelle, da die alternative mit der augmentierten Datenbasis etwa fünf mal so lange dauert. Erst nach der Bestimmung der top sechs Modellkonfigurationen wurden neue OOS-Modelle, mit den Einstellungen dieser sechs Modellkonfigurationen, auf Grundlage der augmentierten Datenbasis generiert, um den produktiven Einsatz der OOS-Modelle bewerten zu können. Die vollständige Dokumentation aller Ergebnisse ist in Anhang D zu finden.

Einen Überblick über die Leistung der verschiedenen Modellkonfigurationen verschafft Kapitel 5.1. Inwiefern die Auslastung der Hardware Einfluss auf die Generierungsdauer der OOS-Modelle hat, erläutert Kapitel 5.2. Auf Grundlage dieser Erkenntnisse werden in Kapitel 5.3 die top sechs Modellkonfigurationen näher untersucht und evaluiert, um potenzielle Kandidaten für einen produktiven Einsatz zu bestimmen.

5.1 Bewertung der Metrikergebnisse

Die Ergebnisse der Modellkonfigurationen auf Grundlage der unveränderten Datenbasis sind in Anhang D.1 dokumentiert. Darin sind für jedes OOS-Modell die ermittelten Resultate in Tabellen protokolliert.

Die folgenden Kapitel stellen die Durchschnittsergebnisse der verschiedenen Metriken dar, um eine Vergleichbarkeit der verschiedenen Modellkonfigurationen zu ermöglichen. Als erstes vergleicht Kapitel 5.1.1 die ermittelten Ergebnisse der Kostenfunktion miteinander. Anschließend geht Kapitel 5.1.2 auf die *Accuracy* der OOS-Modelle ein. Darauf aufbauend analysiert Kapitel 5.1.3 wie die *Accuracy* im Zusammenhang mit dem *Recall* der Klassen OOS und In-Stock steht. Auf Grundlage der erläuterten Ergebnisse untersucht Kapitel 5.1.4 abschließend eine Auswahl von Modellkonfiguration hinsichtlich ihrer Konvergenz.

5.1.1 Kostenfunktionsanalyse der OOS-Modelle

Eine Übersicht über die Ergebnisse der Kostenfunktion BCE je Modellkonfiguration stellt Tabelle 5 dar. Die Werte in der ersten Spalte, mit der Bezeichnung *2*, definieren die verwendete Aktivierungsfunktion in der letzten Schicht des KNNs. In der zweiten Spalte, mit der Bezeichnung *128*, wird die in der vorletzten Schicht eingesetzte Aktivierungsfunktion genannt. Die Kombination der Werte aus beiden Spalten ergibt die Paarung der verwendeten Aktivierungsfunktionen. Zudem verdeutlicht der Name der Spalte, wie viele Neuronen die jeweilige Schicht einsetzt.

Die letzten drei Spalten Adam, FTRL und SGD spezifizieren das verwendete Optimierungsverfahren. So lässt sich aus der Tabelle der durchschnittliche Wert der Kostenfunktion BCE für jede Modellkonfiguration bestehend aus Aktivierungsfunktionspaarung und Optimierungsverfahren ablesen.

2	128	Adam	FTRL	SGD
Sigmoid	ReLU	0,163340812	0,691030353	0,194024574
Sigmoid	Sigmoid	0,214864537	0,691257725	0,401952068
Sigmoid	Softmax	0,536948899	0,69289877	0,672009071
Sigmoid	TanH	0,271986552	0,689997017	0,216243898
Softmax	ReLU	0,200286491	0,690692484	0,246291767
Softmax	Sigmoid	0,189187984	0,690846711	0,40517507
Softmax	Softmax	0,525350526	0,692895393	0,676919361
Softmax	TanH	0,184851721	0,689589918	0,199021837
TanH	ReLU	7,943368753	3,125029042	7,157918612
TanH	Sigmoid	3,961244946	0,548670183	1,651457528
TanH	Softmax	1,409279307	2,61973989	0,634450451
TanH	TanH	4,45527041	5,287303592	6,021151096

Tabelle 5: Ermittelte Werte der Kostenfunktion Binary-Cross-Entropy

Je näher der ermittelt Wert an Null liegt, desto häufiger stimmt der vorhergesagte Wert mit der tatsächlichen Klasse überein. Die erste Modellkonfiguration *Adam · Sigmoid · ReLU* erreicht mit einem Ergebnis von 0,16 bereits das beste Ergebnis. Am schlechtesten schneidet die Modellkonfiguration *Adam · TanH · ReLU* mit 7,94 ab. Die Modellkonfigurationen, die in der letzten Schicht den TanH verwenden, weisen meistens, mit Werten teilweise weit über eins, mangelhafte Leistungen auf. Eine Ausnahme bildet die Modellkonfiguration *FTRL · TanH · Sigmoid*, welche sogar das beste Ergebnis für das Optimierungsverfahren FTRL liefert.

Auffällig ist beim Optimierungsverfahren FTRL, dass die Wahl der Aktivierungsfunktionspaarung, die in der letzten Schicht die Sigmoid oder die Softmax Aktivierungsfunktion verwenden, zu einem Ergebnis der Kostenfunktion von gerundet 0,69 führt.

Stellt man die Ergebnisse der Optimierungsverfahren Adam und SGD gegenüber, so ist zu erkennen, dass die Wahl der Aktivierungsfunktionspaarung zu ähnlichen Schwankungen im Resultat führen. Bei der Betrachtung der Sigmoid und Softmax Aktivierungsfunktion in Spalte 2 scheinen die Ergebnisse von SGD, mit Ausnahme der Modellkonfiguration *SGD · Sigmoid · TanH*, stets schlechter zu sein als bei Adam.

Ein Blick auf die besten Ergebnisse zeigt, dass Adam die top drei Modellkonfigurationen stellt. SGD belegt danach den vierten Platz mit der Modellkonfiguration *SGD · Sigmoid · ReLU*. Somit definieren Adam und SGD mit derselben Aktivierungsfunktionspaarung ihr jeweils bestes Ergebnis. Die zweitbesten Ergebnisse der beiden Optimierungsverfahren wurden ebenfalls mit derselben Aktivierungsfunktionspaarung *Softmax · TanH*

erreicht. Erst danach unterscheidet sich die Reihenfolge der nächstbesten Aktivierungsfunktionspaarungen.

5.1.2 Genauigkeit der OOS-Modelle

Die *Accuracy*-Klassifizierungsmetrik gibt eine gute Auskunft über die Leistung der OOS-Modelle. Bei der Betrachtung der Ergebnisse gilt es zu berücksichtigen, dass eine zufällige Klassifizierung bereits eine Erfolgswahrscheinlichkeit von 50% hat. Die *Accuracy* der OOS-Modelle ist in Tabelle 6 in Prozent zu jeder Modellkonfiguration abgebildet.

2	128	Adam	FTRL	SGD
Sigmoid	ReLU	93,53%	56,73%	93,01%
Sigmoid	Sigmoid	93,32%	54,91%	88,22%
Sigmoid	Softmax	93,14%	54,91%	68,41%
Sigmoid	TanH	86,58%	54,91%	92,12%
Softmax	ReLU	92,47%	60,78%	89,49%
Softmax	Sigmoid	93,76%	54,91%	88,16%
Softmax	Softmax	92,31%	54,91%	62,63%
Softmax	TanH	93,49%	54,91%	92,46%
TanH	ReLU	51,34%	64,95%	53,55%
TanH	Sigmoid	66,03%	80,24%	74,86%
TanH	Softmax	66,85%	53,55%	65,85%
TanH	TanH	50,90%	59,35%	51,34%

Tabelle 6: Accuracy der OOS-Modelle

Bei der *Accuracy* schneiden Modellkonfigurationen ähnlich gut oder schlecht ab wie bei der Kostenfunktion. Allerdings ist eine Korrelation zwischen der Kostenfunktion und der *Accuracy* nicht zu erkennen. Weiterhin weisen Modellkonfigurationen, die die Aktivierungsfunktion TanH als letzte Schicht verwenden, schwache Resultate auf. Erneut bildet die Modellkonfiguration *FTRL · TanH · Sigmoid* mit 80,24% eine Ausnahme. Im Gesamtvergleich befindet sich dieses Ergebnis jedoch nur auf dem 15ten Platz. Alle anderen Resultate von FTRL sind wesentlich schwächer, als aufgrund der Kostenfunktion anzunehmen war. Gerundet liegen diese zwischen 55% und 65%. Auffällig ist auch, dass sechs Modelle auf die zweite Nachkommastelle genau die gleiche *Accuracy* aufweisen.

In einigen Fällen ist die *Accuracy* besser, als die Werte der Kostenfunktion vermuten lassen. Obwohl die Modellkonfiguration *SGD · TanH · Sigmoid* für die Kostenfunktion den Wert 1,65 erzielt hat, erreicht dieselbe Modellkonfiguration eine *Accuracy* von 75%. Das gleiche gilt auch für die Modellkonfiguration *Adam · Sigmoid · Softmax*, welche das insgesamt fünftbeste Ergebnis, trotz eines Kostenfunktionswertes von 0,53, erreicht. Ein ähnliches Verhalten zeigt auch die Modellkonfiguration *Adam · Softmax · Softmax*. Speziell die letzten zwei Ausnahmen zeigen, dass Modelle Spitzenergebnisse in der *Accuracy*

erzielen können, gleichwohl sie nicht die besten Ergebnisse in der Kostenfunktion aufweisen.

Die zwei besten Resultate von SGD haben genau dieselben Aktivierungsfunktionspaarungen, wie bei den Ergebnissen der Kostenfunktion. Dennoch hat die Modellkonfiguration $SGD \cdot Sigmoid \cdot ReLU$ mit 93,01% insgesamt nur das sechstbeste Ergebnis erreicht. Wiedermal stellt Adam die beste Modellkonfiguration. Jedoch ist es eine andere als bei der Kostenfunktion. Mit 93,76% hat die Modellkonfiguration $Adam \cdot Softmax \cdot Sigmoid$ das beste Ergebnis erreicht. Knapp dahinter, mit 93,53%, befindet sich der Kostenfunktionssieger $Adam \cdot Sigmoid \cdot ReLU$. Insgesamt sechs Modellkonfigurationen erreichen ein Ergebnis von über 93% und liegen dementsprechend nah aneinander.

Die Ergebnisse deuten klar darauf hin, dass das Optimierungsverfahren FTRL vergleichsweise schwache Leistungen erzielt und somit am wenigsten geeignet ist. SGD liefert zwar bei einigen Aktivierungsfunktionspaarungen gute Resultate, jedoch sind diese dem Optimierungsverfahren Adam unterlegen. Eine Ausnahme bildet die Aktivierungsfunktionspaarung $Sigmoid \cdot TanH$. Hier schlägt SGD den Adam um 5,54%. Adam scheint erneut am besten geeignet zu sein, da die fünf Konfigurationen mit der höchsten *Accuracy* dieses Optimierungsverfahren nutzen.

5.1.3 Recall der OOS-Modelle

Für den Anwendungsfall ist es besonders wichtig, dass möglichst alle bestehenden OOS-Situationen erkannt werden. Darum ist es wichtig eine gute Quote für die *Recall*-Klasse OOS zu erreichen. Nachfolgend stellt Tabelle 7 den *Recall* aller Modellkonfigurationen für die Klasse OOS dar. Um die ermittelten Werte in Relation setzen zu können, zeigt Tabelle 8 den *Recall* für die Klasse In-Stock.

2	128	Adam	FTRL	SGD
Sigmoid	ReLU	91,89%	0,00%	92,12%
Sigmoid	Sigmoid	95,25%	0,00%	88,35%
Sigmoid	Softmax	91,93%	0,00%	34,59%
Sigmoid	TanH	76,34%	0,00%	92,41%
Softmax	ReLU	90,61%	23,14%	84,83%
Softmax	Sigmoid	91,57%	0,00%	87,48%
Softmax	Softmax	87,99%	0,00%	21,58%
Softmax	TanH	92,96%	0,00%	91,60%
TanH	ReLU	50,00%	58,41%	33,33%
TanH	Sigmoid	31,15%	46,34%	93,26%
TanH	Softmax	0,00%	0,00%	23,57%
TanH	TanH	81,43%	7,14%	16,67%

Tabelle 7: Recall der Klasse OOS

2	128	Adam	FTRL	SGD
Sigmoid	ReLU	95,11%	95,83%	93,52%
Sigmoid	Sigmoid	91,80%	100,00%	88,60%
Sigmoid	Softmax	94,44%	100,00%	94,52%
Sigmoid	TanH	95,99%	100,00%	92,38%
Softmax	ReLU	94,83%	96,86%	94,23%
Softmax	Sigmoid	96,14%	100,00%	89,62%
Softmax	Softmax	96,61%	100,00%	99,47%
Softmax	TanH	94,42%	100,00%	93,49%
TanH	ReLU	66,67%	45,71%	66,67%
TanH	Sigmoid	65,73%	86,76%	72,41%
TanH	Softmax	0,00%	0,00%	96,44%
TanH	TanH	82,45%	65,79%	47,34%

Tabelle 8: Recall der Klasse In-Stock

Der *Recall* für die Klasse In-Stock liegt meist über dem *Recall* der Klasse OOS. Nur in 4 von 36 Fällen ist die Erkennung der Klasse OOS besser als die der Klasse In-Stock. Mit einem OOS-*Recall* von 95,25% schneidet die Modellkonfiguration *Adam · Sigmoid · Sigmoid* am besten ab. Hinsichtlich der *Accuracy* ist diese Modellkonfiguration nur die viertbeste. Jedoch sind die Abstände zwischen den Spitzenreitern beim *Recall* der Klasse OOS größer als bei der *Accuracy*-Klassifizierungsmetrik. Während bei der *Accuracy* der viertplatzierte nur 0,44% hinter dem besten Ergebnis liegt, beträgt der OOS-*Recall* der zweitplatzierten Modellkonfiguration *SGD · TanH · Sigmoid* 93,26% und somit fast zwei Prozentpunkte weniger.

Die zuletzt genannte Modellkonfiguration ist ein gutes Beispiel für einen trügerisch guten *Recall*. Zwar erzielt die Modellkonfiguration einen guten *Recall* für die Klasse OOS, jedoch sind im Vergleich zu den anderen Modellkonfigurationen die Werte für die *Accuracy* und den *Recall* der Klasse In-Stock deutlich niedriger. Folglich müssen bei der Bewertung einer Modellkonfiguration die Ergebnisse aller Klassifizierungsmetriken berücksichtigt werden. Dies gilt besonders für das Optimierungsverfahren FTRL. Viele dieser Modellkonfigurationen weisen eine identische Genauigkeit auf, was darauf zurückzuführen ist, dass der *Recall* für die Klassen OOS und In-Stock jeweils bei 0 und 100 liegt. Wird in der letzten Schicht die Aktivierungsfunktion Sigmoid oder Softmax verwendet, lernt das Modell mit jeder Aktivierungsfunktionspaarung immer die Klasse In-Stock vorherzusagen, da mehr Bilder für diese Klasse existieren. Über die *Cross-Validierung* aller sechs Modelle ergibt das eine durchschnittliche *Accuracy* von 54,91%.

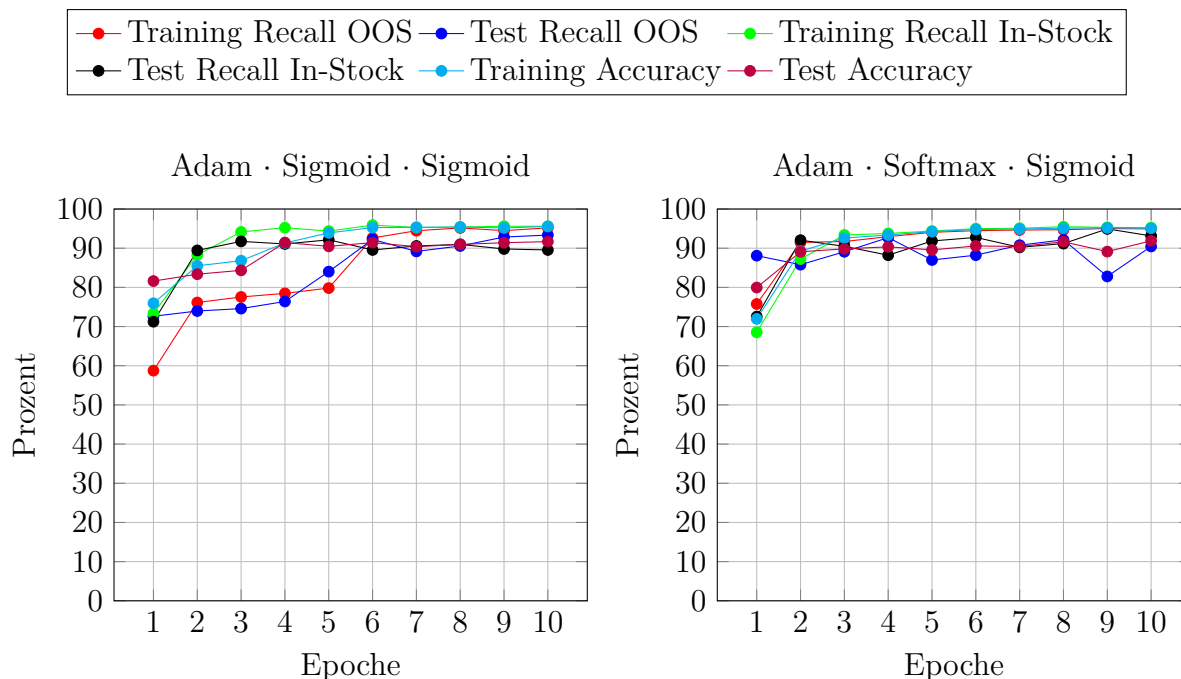
Mit Ausnahme der Modellkonfiguration *Adam · TanH · TanH* existiert bei der Verwendung des TanHs in der letzten Schicht keine Kombination, bei der ein guter *Recall* sowohl für die OOS- als auch für die In-Stock-Klasse erreicht wird. Tatsächlich ist die Ausnahme gleichzeitig auch die Modellkonfiguration, die die schlechteste *Accuracy* aufweist.

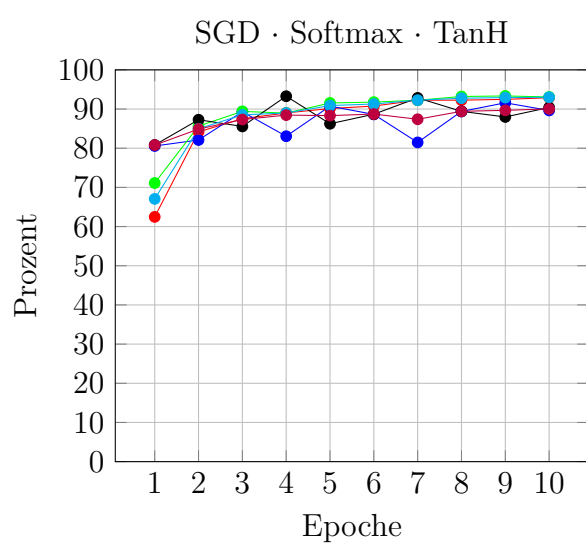
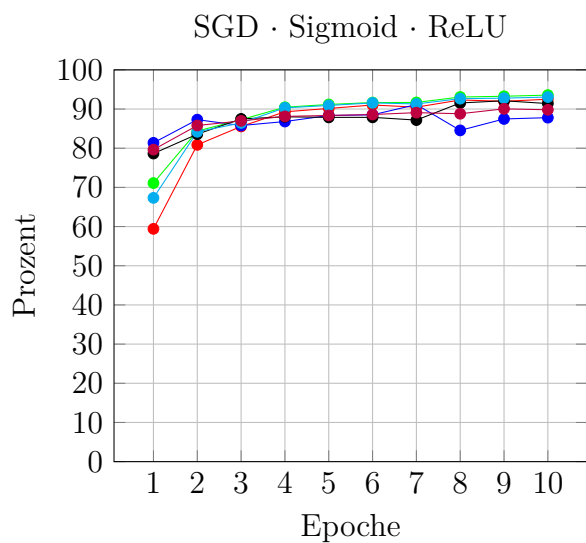
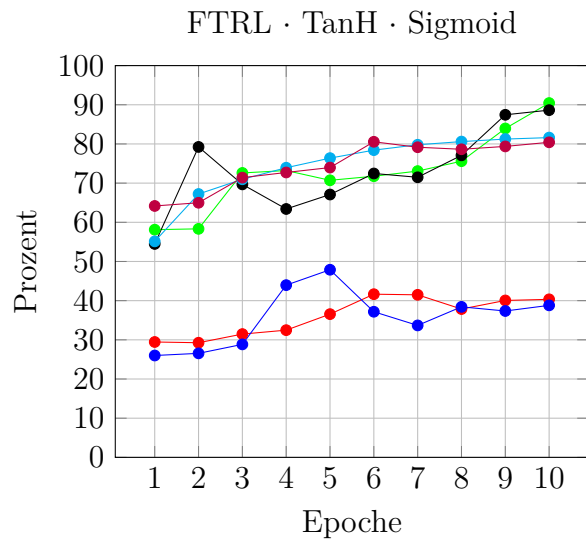
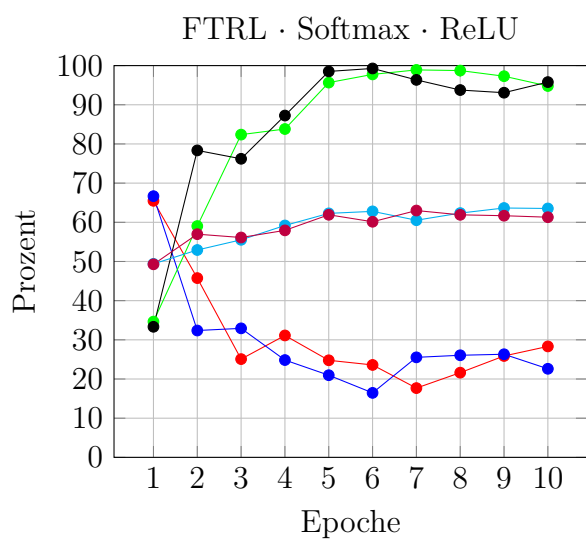
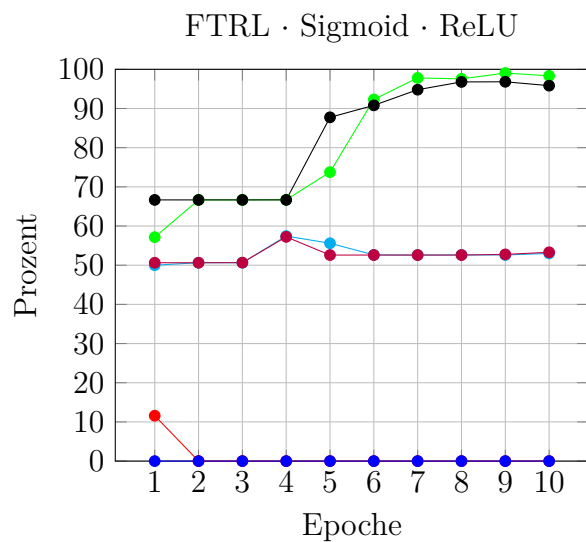
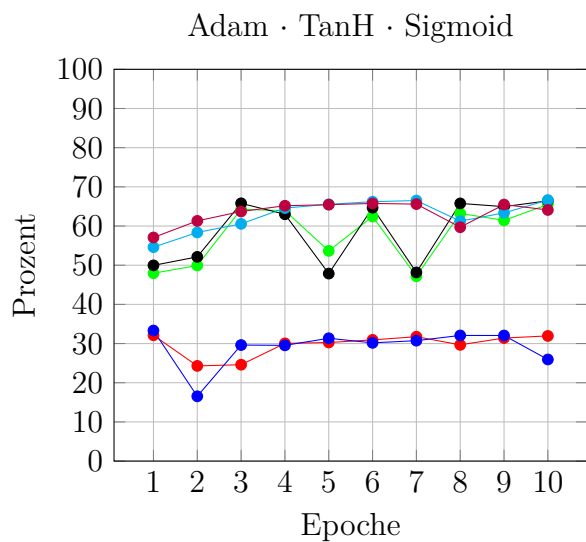
Hingegen zeigen alle Modellkonfigurationen, die in der letzten Schicht die Aktivierungsfunktion Sigmoid oder Softmax verwenden, gute Ergebnisse für die Klasse In-Stock. Die Modellkonfigurationen, die weniger gute Ergebnisse für die Klasse OOS erreichen, schneiden auch bei der *Accuracy* schlechter ab.

5.1.4 Konvergenzverhalten verschiedener OOS-Modelle

Um eine möglichst zuverlässige Vorhersage zu erreichen, ist es entscheidend, die Trainingsphase der Modelle ausreichend lang zu gestalten. In diesem Kapitel wird daher untersucht, ob die Trainingsphase der OOS-Modelle für die verschiedenen Modellkonfigurationen ausreichend lang war. Hierzu wurden die Trainings- und Testergebnisse der *Accuracy*- und *Recall*-Klassifizierungsmetriken je Epoche untersucht. Ist während der Trainingsphase eine Konvergenz zu erkennen, war die Anzahl der definierten Epochen ausreichend.

Damit die Anzahl der zu betrachtenden Modellkonfigurationen überschaubar bleibt, wurden für jedes Optimierungsverfahren drei Modellkonfigurationen ausgewählt, die unterschiedliche Aktivierungsfunktionen in der letzten Schicht verwenden. Da eine allgemein gute Vorhersage von Modellen wichtig ist, wurden Modellkonfigurationen ausgewählt, die eine gute *Accuracy* aufweisen. Zeigen mehrere Modellkonfiguration ähnlich gute Ergebnisse, wurde der OOS-*Recall* hinzukommend betrachtet, da dieser eine aussagekräftige Metrik für die zuverlässige OOS-Erkennung darstellt. Deshalb erhält im nachfolgenden die Modellkonfiguration *Adam · Sigmoid · Sigmoid* gegenüber der Modellkonfiguration *Adam · Sigmoid · ReLU* den Vorzug, obwohl die *Accuracy* schlechter ist. Ein um 3,36% besserer *Recall* überwiegt die geringere *Accuracy* von 0,21%. Abbildung 24 stellt für jede ausgewählte Modellkonfiguration die Entwicklung der verschiedenen Metriken im Laufe der Epochen der Trainingsphase dar.





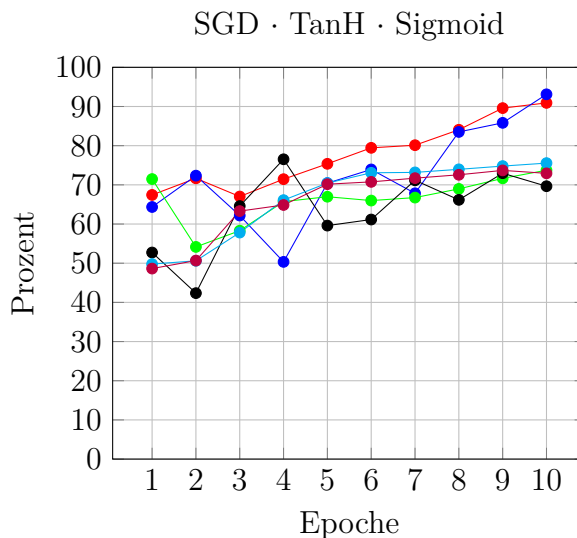


Abbildung 24: Vergleich der Konvergenz verschiedener Modellkonfigurationen

Die verschiedenen Graphen zeigen teils deutliche Unterschiede im Verlauf des Trainings. Während bei vier Graphen die Ergebnisse zu allen Klassifizierungsmetriken bei um die 90% konvergieren, sind bei den anderen Graphen die Abstände zwischen den Ergebnissen der Klassifizierungsmetriken größer.

Am deutlichsten ist es bei der Modellkonfiguration *FTRL · Sigmoid · ReLU* zu erkennen. Je fortgeschrittener das Training ist, desto vollständiger prognostiziert das Modell die Klasse In-Stock, wohingegen die Klasse OOS niemals vorhergesagt wird. Folglich verbessert sich die *Accuracy* nicht.

Bei der Modellkonfiguration *FTRL · Softmax · ReLU* steigt die *Accuracy* leicht, weil auch die Klasse OOS in wenigen Fällen vorhergesagt wird. Die beste Modellkonfiguration mit FTRL unter Verwendung von *TanH · Sigmoid* zeigt, dass eine gute *Accuracy* erreicht werden kann, wenn der *Recall* für die Klasse In-Stock nicht bereits nach wenigen Epochen bei 100% liegt.

Bei allen Graphen ist zu erkennen, dass sich die Trainings- und Testergebnisse für die jeweiligen Klassifizierungsmetriken im Verlauf der Trainingsphasen auf ein ähnliches Niveau angleichen. Das ist besonders bei der Modellkonfiguration *SGD · TanH · Sigmoid* zu sehen, da im Verlauf des Trainings zunächst starke Ausreißer zu beobachten sind, die jedoch gegen Ende des Trainings immer weiter abnehmen.

Es sind auch Unterschiede in der Geschwindigkeit der Konvergenz festzustellen. Während sich die Metriken der Modellkonfiguration *Adam · Sigmoid · Sigmoid* erst in Epoche 6 angleichen, konvergiert die Modellkonfiguration *Adam · Softmax · Sigmoid* bereits in Epoche 3.

Zusammenfassend lässt sich sagen, dass die Trainingsphase ausreichend war, um einen Vergleich der verschiedenen Modellkonfigurationen zu ermöglichen. Es sind keine großen

Veränderungen bei einem längeren Training zu erwarten, da die *Accuracy* bei allen Graphen seit mehreren Epochen stabil ist. Dennoch ist es empfehlenswert das OOS-Modell, das in der Praxis eingesetzt wird, länger zu trainieren, um möglicherweise die Leistung um weitere Prozentpunkt zu steigern.

5.2 Einfluss der Hardware-Leistung auf die Generierungsdauer

Die zur Verfügung gestellte Hardware beeinflusst die Trainingsdauer von OOS-Modellen. Um festzustellen, ob sich auch die Modellkonfiguration auf die Generierungsdauer auswirkt, vergleicht Kapitel 5.2.1 die durchschnittliche Trainingsdauer der verschiedenen Modellkonfigurationen. Aus den dadurch gewonnenen Erkenntnissen erläutert Kapitel 5.2.2 den Zusammenhang zwischen der Auslastung der Hardware-Komponenten und der Generierungsdauer.

5.2.1 Generierungsdauer der OOS-Modelle

Die Generierungsdauer eines OOS-Modells setzt sich aus der benötigten Zeit für die Trainings- und Testphase zusammen. Für jede Modellkonfiguration wurde der Mittelwert über die Generierungsdauer der sechs OOS-Modelle gebildet. Nachfolgend stellt Tabelle 9 die ermittelten Werte für die verschiedenen Modellkonfigurationen dar.

2	128	Adam	FTRL	SGD
Sigmoid	ReLU	1:35:08	1:34:53	1:34:53
Sigmoid	Sigmoid	1:35:08	1:34:51	1:34:51
Sigmoid	Softmax	1:35:16	1:35:06	1:34:32
Sigmoid	TanH	1:34:52	1:34:21	1:34:21
Softmax	ReLU	1:35:05	1:34:35	1:34:50
Softmax	Sigmoid	1:35:17	1:34:41	1:34:52
Softmax	Softmax	1:35:07	1:32:31	1:34:23
Softmax	TanH	1:35:23	1:32:40	1:34:12
TanH	ReLU	1:34:25	1:34:39	1:35:25
TanH	Sigmoid	1:34:35	1:34:50	1:34:26
TanH	Softmax	1:34:29	1:34:38	1:34:32
TanH	TanH	1:34:43	1:35:12	1:34:52

Tabelle 9: Generierungsdauer der OOS-Modelle

Die durchschnittliche Generierungsdauer beträgt eine Stunde, 34 Minuten und 39 Sekunden. Der Unterschied zwischen der längsten und kürzesten Generierungsdauer beträgt zwei Minuten und 54 Sekunden. Ohne die beiden schnellsten Modellkonfigurationen *FTRL · Softmax · Softmax* und *FTRL · Softmax · Softmax* liegt die Differenz nur noch bei einer Minute und 13 Sekunden.

Diese geringe Spanne zeigt, dass die Modellkonfigurationen keine großen Auswirkungen auf die Generierungsdauer der OOS-Modelle haben. Viel mehr wird aus den Werten deutlich, dass bei einer identischen Datenbasis und gleichem Aufbau die Generierungsdauer ähnlich ist.

5.2.2 Hardware-Auslastung

Durch den Einsatz einer Monitoring-Software lässt sich der Ressourcenverbrauch einer Anwendung überwachen. Dies ermöglicht das Nachverfolgen der Hardware-Auslastung während der Generierung der OOS-Modelle. Die Erkenntnisse bezüglich der Generierungsdauer verschiedener Modellkonfigurationen wurden durch die aus den Monitoringdaten gewonnenen Informationen bestätigt. Es sind während der Trainingsphasen keine Unterschiede in der Auslastung der Hardware zu erkennen. Die Hardware-Auslastung umfasst sowohl die Beanspruchung der CPU als auch die des Arbeitsspeichers.

Anhand der Monitoringdaten ist zu erkennen, dass die CPU bei der Ausführung des Programms vollständig beansprucht wurde, während der Arbeitsspeicher im Durchschnitt zu 58% ausgelastet war. Zu keinem Zeitpunkt hat die Anwendung die möglichen acht GB Arbeitsspeicher ausgereizt, sondern schwankte meist zwischen 4,4 GB und 5,2 GB. Dementsprechend führt eine Vergrößerung des Arbeitsspeichers nicht zu einer Verkürzung der Generierungsdauer.

Es scheint jedoch, dass die Generierungsdauer stark von der CPU abhängt. Alle drei Kerne der CPU waren voll ausgelastet. Folglich ermöglicht eine Erhöhung der Anzahl von Kernen, mehr Berechnungen parallel auszuführen und somit die Generierungsdauer der OOS-Modelle zu reduzieren. Dies bestätigt auch die Aussage, dass eine GPU die Trainingszeit merklich reduzieren kann, da eine GPU i.d.R. deutlich mehr Kerne als eine CPU besitzt.

5.3 Vergleich der besten Modellkonfigurationen

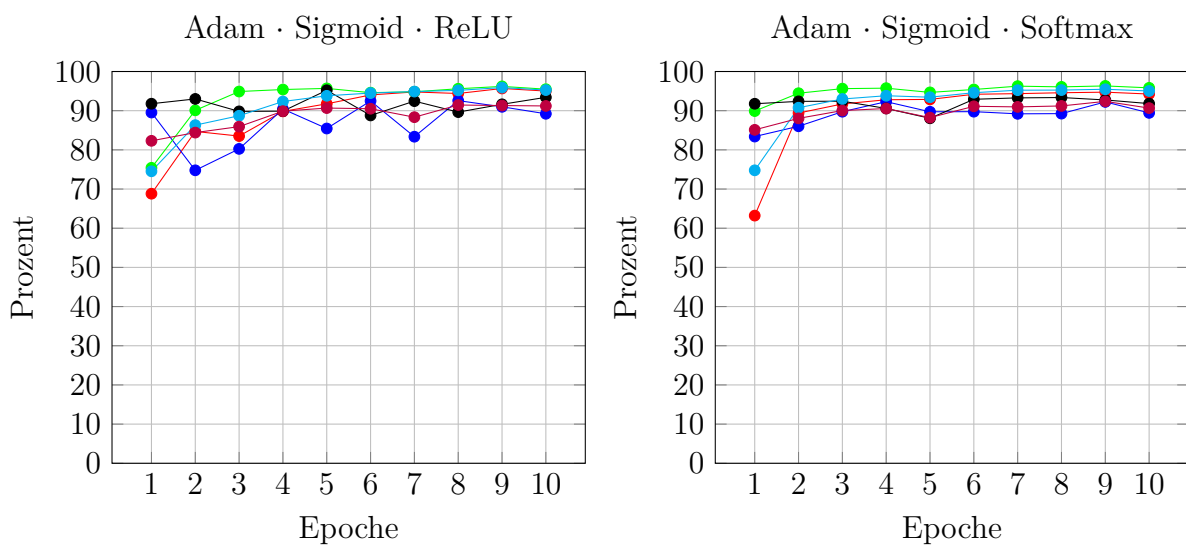
Im Anschluss an die Übersicht der Leistung der verschiedenen Modellkonfigurationen erfolgt eine detaillierte Untersuchung der besten Modellkonfigurationen. Hierzu zählen alle Modellkonfigurationen, die eine *Accuracy* von mindestens 93% erreicht haben. Dazu gehören fünf Modellkonfigurationen von Adam und die beste Modellkonfiguration von SGD. Drei dieser Modellkonfigurationen wurden bisher nicht auf ihre Konvergenz untersucht, weshalb Kapitel 5.3.1 dies nachholt und den Fokus auf die Unterschiede in der Konvergenz zwischen den besten Modellkonfigurationen setzt.

Zudem werden für die besten Modellkonfigurationen die einzelnen OOS-Modelle der verschiedenen Märkte untersucht. Hierbei wird in Kapitel 5.3.2 analysiert, inwiefern die Auswahl des Testmarktes Auswirkungen auf die Leistung der OOS-Modelle hat.

Für die sechs besten Modellkonfigurationen wurden unter Verwendung eines realistischen Datenverhältnisses neue OOS-Modelle generiert. Hierfür wurde die Datenbasis mit den augmentierten Bildern verwendet. Kapitel 5.3.3 untersucht abschließend den Einfluss des Datenverhältnisses auf die erzeugten OOS-Modelle.

5.3.1 Konvergenzverhalten der besten Modellkonfigurationen

Durch die Betrachtung der Konvergenz der besten Modellkonfigurationen lässt sich erkennen, ob eine längere Trainingsdauer zu potenziell besseren Ergebnissen führen kann. Obwohl sich die Ergebnisse der besten Modellkonfigurationen auf einem ähnlichen Niveau bewegen, demonstrieren die Graphen, wie unterschiedlich der Trainingsverlauf der einzelnen Modellkonfigurationen ist. Die Trainings und Testverläufe der Verfahren *Adam · Sigmoid · Sigmoid*, *Adam · Softmax · Sigmoid* und *SGD · Sigmoid · ReLU* können Abbildung 24 entnommen werden. Abbildung 25 zeigt die Verläufe für *Adam · Sigmoid · ReLU*, *Adam · Sigmoid · Softmax* und *Adam · Softmax · TanH*.



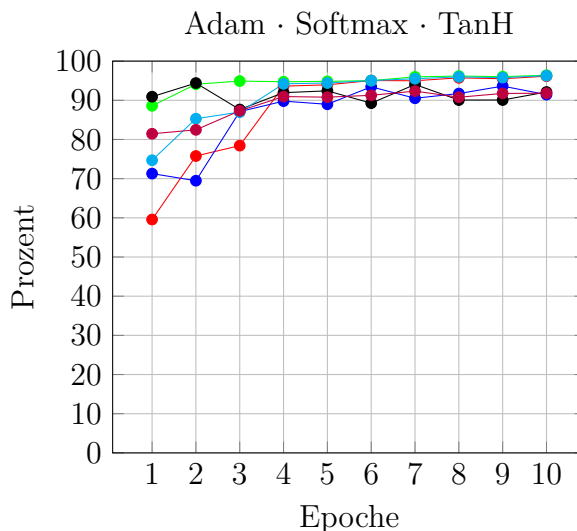


Abbildung 25: Konvergenzen der Top Modellkonfigurationen von Adam

Es ist gut zu erkennen, dass die Geschwindigkeit, mit der sich die Ergebnisse der Klassifizierungsmetriken einem gemeinsamen Wert nähern, je nach Modellkonfiguration unterscheiden. Während sich z.B. die Ergebnisse der Modellkonfigurationen *Adam · Softmax · TanH* und *Adam · Sigmoid · Sigmoid* nach jedem Epochenschritt nur langsam angleichen, sind die Lernkurven der Modellkonfiguration *Adam · Sigmoid · Softmax* steiler und konvergieren dementsprechend früher.

Bei allen top sechs Modellkonfigurationen ist eine Konvergenz zu erkennen. Jedoch unterscheidet sich das Konvergenzverhalten hinsichtlich der Ausreißer. Einige Modellkonfigurationen wie *Adam · Softmax · TanH* zeigen ab dem Konvergenzpunkt nahezu keine Veränderungen, wohingegen andere Modellkonfigurationen trotz einer zu erkennenden Konvergenz in einigen Epochen unerwartete Ausreißer haben. Besonders die Metrik *Test Recall OOS* hat des Öfteren eine Abweichung nach unten, wie in den Modellkonfigurationen *Adam · Softmax · Sigmoid* und *Adam · Sigmoid · ReLU* zu sehen. Diese plötzlichen Schwankungen bieten die Möglichkeit aus einem potenziellen lokalen Minimum auszubrechen und somit ein besseres Minimum zu finden. Andererseits besteht die Gefahr nach einer Epoche zu enden, die schlechtere Klassifizierungsergebnisse erzielt, da sich die Kostenfunktion bspw. in einem schlechteren lokalen Minimum befindet. Des Weiteren ist gut zu erkennen, dass die *Accuracy* im Training stets besser ist als die *Accuracy* im Test. Da der Unterschied zwischen den Test- und Trainingsergebnissen nur gering ist, handelt es sich dabei um ein gutes Resultat.

Bisher wurde nur die Konvergenz der Klassifizierungsmetriken *Accuracy* und *Recall* untersucht. Um weitere Erkenntnisse über das Konvergenzverhalten zu gewinnen, veranschaulicht die nachfolgende Abbildung 26 das Konvergenzverhalten aus der Kostenfunktionsperspektive. Hierzu werden für jede Modellkonfiguration die durchschnittlichen Ergebnisse der Kostenfunktion nach jeder Epoche in einem Graphen dargestellt.

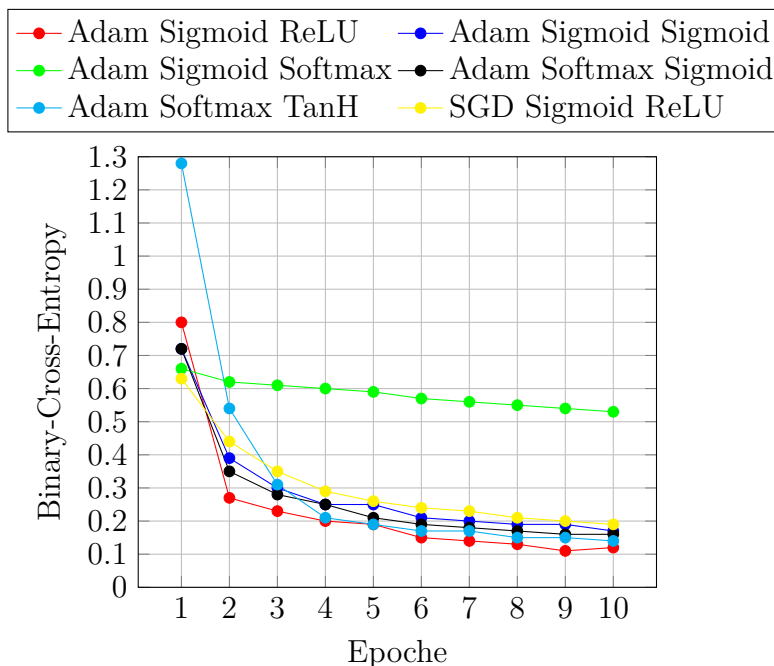


Abbildung 26: Kostenfunktion der Top 6 Modellkonfigurationen

Offensichtlich stellt der Verlauf der Modellkonfiguration *Adam · Sigmoid · Softmax* einen Sonderfall dar. Während alle anderen Modellkonfiguration eine Lernkurve illustrieren, die einer exponentiell fallenden Funktion ähnelt, zeigt der Sonderfall einen leicht abfallenden linearen Verlauf.

Unter den übrigen fünf Modellkonfigurationen sticht *Adam · Softmax · TanH* heraus, da die Kostenfunktion dieser Modellkonfiguration zu Beginn auffallend schlecht ist, sich allerdings ab Epoche drei den anderen schnell annähert. Anschließend weist sie ähnliche Ergebnisse wie die anderen Modellkonfigurationen auf.

Im Gegensatz zum Konvergenzverhalten der *Accuracy*- und *Recall*-Klassifizierungsmetriken ist bei der Kostenfunktion keine eindeutige Konvergenz zu erkennen. Die Verbesserung nimmt mit jeder neuen Epoche stetig ab, jedoch ist keine Stagnation zu erkennen. Dies legt nahe, dass eine Erhöhung der Anzahl von Epochen sinnvoll ist, um sicherzustellen, dass die Kostenfunktion den bestmöglichen Wert erreicht hat und das OOS-Modell optimal trainiert wurde.

5.3.2 Vergleich der Cross-Validierten OOS-Modelle

Mithilfe der *Cross-Validierung* wird sichergestellt, dass Ausnahmeergebnisse einen geringen Einfluss auf das Gesamtergebnis der Modellkonfiguration haben. Dennoch ist es sinnvoll zu untersuchen, welche Auswirkungen die verwendeten Testmärkte auf die Kostenfunktion und die Klassifizierungsmetriken haben. So kann evaluiert werden, wie sich die Ergebnisse der OOS-Modelle von Markt zu Markt unterscheiden.

Hierzu stellt Abbildung 27 für die Kostenfunktion und jede Klassifizierungsmetrik einen Graphen bereit, in dem die Ergebnisse der besten Modellkonfigurationen je Testmarkt veranschaulicht sind.

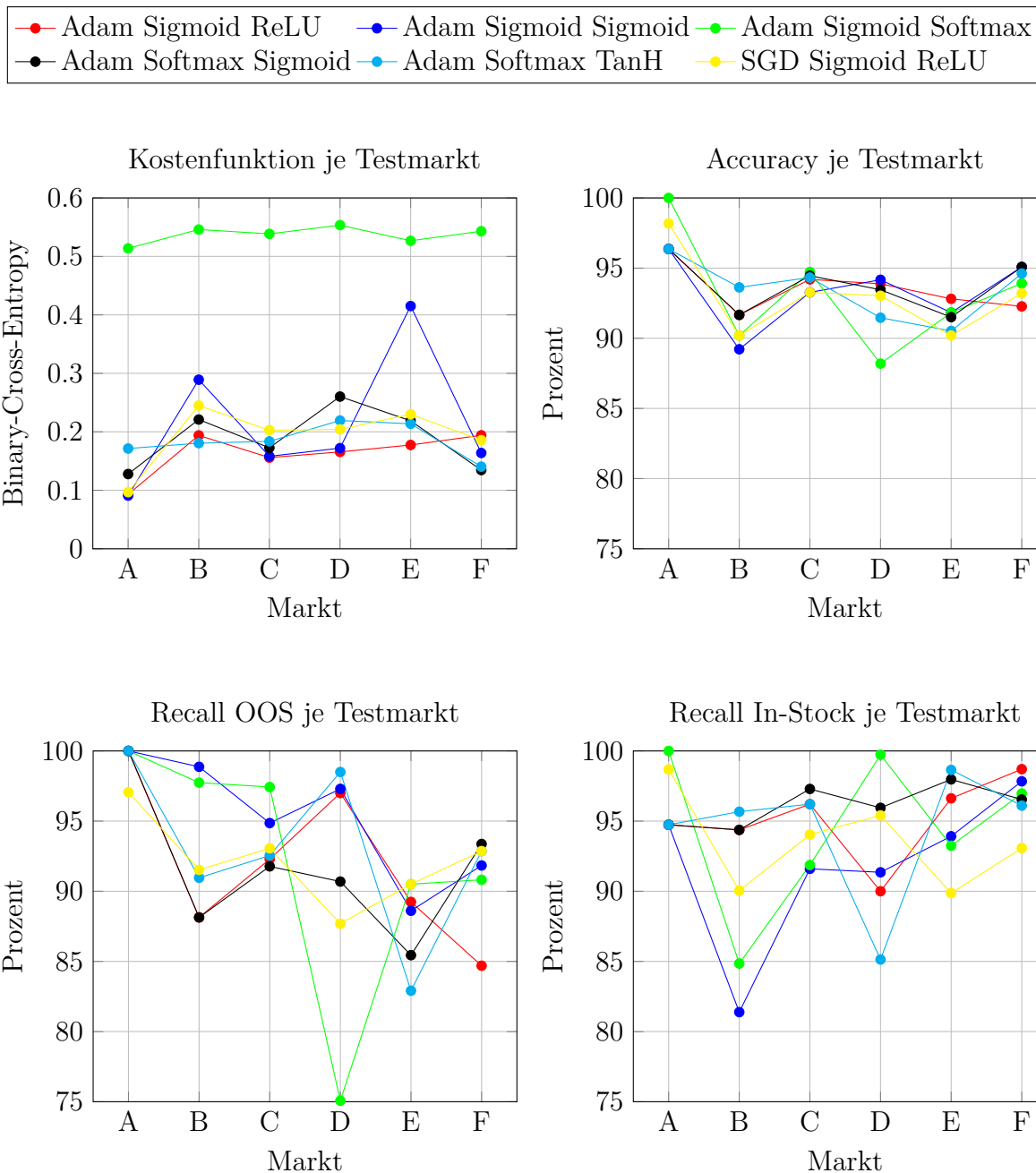


Abbildung 27: Unterschiede in den Ergebnissen der Cross-Validierung

Wie bereits in Abbildung 26 ersichtlich war, liefert die Modellkonfiguration *Adam · Sigmoid · Softmax* schlechtere Resultate für die Kostenfunktion als die übrigen Modellkonfigurationen. Dafür schwanken die Ergebnisse zwischen den Testmärkten im Verhältnis zu den anderen Modellkonfigurationen weniger.

Am größten ist die Schwankung bei der Modellkonfiguration *Adam · Sigmoid · Sigmoid*. Diese erreicht für den Markt A das beste Ergebnis, jedoch auch das schlechteste

für die Märkte B und E. Auffällig ist der besonders starke Ausreißer bei Markt E. Diese signifikante Abweichung spiegelt sich jedoch nicht im Ergebnis der *Accuracy* wieder. Hierbei sind ähnlich gute Resultate wie bei den anderen Modellkonfigurationen erzielt worden, die für den Markt E alle nah beieinander liegen. In den meisten Fällen führt ein guter oder schlechter Kostenfunktionswert zu einem ähnlichen Ergebnis in der *Accuracy*. So hat z.B. die Modellkonfiguration *Adam · Sigmoid · ReLU* das beste Ergebnis für die Kostenfunktion und die *Accuracy* in Markt E erzielt.

Eine weitere Besonderheit stellt wiederum die Modellkonfiguration *Adam · Sigmoid · Softmax* dar. Obwohl die Werte der Kostenfunktion nur geringfügig schwanken, gibt es große Unterschiede in der *Accuracy* der Testmärkte. Tatsächlich wurde für den Markt A eine *Accuracy* von 100% erreicht, wohingegen für den Markt D eine *Accuracy* von nur etwa 88% erzielt wurde. Da auch der *Recall* für den Markt A bei beiden Klassen 100% beträgt, scheint diese Modellkonfiguration besonders gut für diesen Markt und somit das Getränkesortiment geeignet zu sein.

Im Vergleich zu der *Accuracy* sind beim *Recall* beider Klassen die Schwankungen bei allen Modellkonfigurationen deutlich stärker. Bspw. beträgt der maximale Unterschied der *Accuracy* bei der Modellkonfiguration *Adam · Sigmoid · ReLU* 5%, wohingegen die Differenz des *Recalls* der Klasse OOS bis zu 15% Prozent beträgt. Zudem zeigen die *Recall*-Klassen in wenigen Fällen ähnliche Werte. Mit Ausnahme von Markt A führt ein guter *Recall* der einen Klasse oft zum schlechteren *Recall* der anderen Klasse. Zumeist ist der *Recall* der Klasse In-Stock besser als der Klasse OOS. Eine Ausnahme bildet die Modellkonfiguration *Adam · Sigmoid · Softmax*, die nur bei den Märkten E und F einen schlechteren *Recall* für die Klasse OOS erzielt hat.

Da der *Recall* der Klasse OOS zu den wichtigsten Klassifizierungsmetriken gehört, bietet es sich für die OOS-Erkennung an eine Modellkonfiguration zu verwenden, die für Märkte mit ähnlichen Rahmenbedingungen gute Ergebnisse liefert. Bei den betrachteten Testmärkten ist zu erkennen, dass sich die beste Modellkonfiguration von Markt zu Markt unterscheidet. Offensichtlich existieren mehrere Modellkonfigurationen, die über 95% aller OOS-Situation für die Märkte A, B, C und D erkennen. Scheinbar ist der Markt E ein Ausreißer nach unten. Um die Ursachen dafür herauszufinden, bietet es sich in einer folgenden Arbeit an, die Bilder zu untersuchen, die fälschlicherweise als In-Stock klassifiziert wurden. Eventuell lassen sich daraus Merkmale erschließen, die zu der verminderten Leistung führen.

5.3.3 Auswirkung des Datenverhältnisses auf die OOS-Modelle

Es besteht ein Risiko, dass ein Modell in der Praxis schlechter abschneidet, wenn das Verhältnis der trainierten Daten nicht dem der Realität entspricht. Um sicherzustellen, dass die generierten OOS-Modelle für die Praxis geeignet sind, wurden die besten Modellkon-

figurationen nochmals neu mit der augmentierten Datenbasis generiert und getestet. Alle anderen globalen Konfigurationseinstellungen wurden beibehalten. Hierdurch soll überprüft werden, wie sich die Änderung des Datenverhältnisses auf die Ergebnisse auswirkt.

Die Ergebnisse zu den generierten OOS-Modellen auf Grundlage der besten Modellkonfigurationen und der augmentierten Datenbasis sind in Anhang D.2 dokumentiert. Die Tabelle 10 gibt einen Überblick über die Ergebnisse der OOS-Modelle. Dabei werden die durchschnittlichen Ergebnisse der Kostenfunktion BCE, der Klassifizierungsmetriken und der Generierungsdauer für die betrachteten Modellkonfigurationen dargestellt.

	Adam · Sigmoid · ReLU	Adam · Sigmoid · Sigmoid	Adam · Sigmoid · Softmax	Adam · Softmax · Sigmoid	Adam · Softmax · TanH	SGD · Sigmoid · ReLU
Evaluation						
Kostenfunktion BCE	0,0595	0,1609	0,2858	0,1288	0,2427	0,0793
Accuracy	97,91%	93,08%	92,22%	96,00%	93,13%	97,31%
Recall OOS	87,82%	11,48%	0,00%	49,66%	13,46%	77,16%
Recall In-Stock	98,87%	99,97%	100,00%	99,63%	99,77%	99,15%
Generierungsdauer	7:38:01	7:37:49	7:39:22	7:29:31	7:29:41	7:40:42

Tabelle 10: Evaluation von Modellkonfigurationen auf Basis von augmentierten Daten

Auffällig sind die schwachen Resultate vom *Recall* der Klasse OOS. Während alle sechs Modellkonfigurationen mit der ursprünglichen Datenbasis Ergebnisse von über 91% für den OOS-*Recall* lieferten, liegen bei der Verwendung der augmentierten Datenbasis nur zwei Modellkonfigurationen bei über 50%. Tatsächlich unterscheiden sich diese beiden Modellkonfigurationen nur im Optimierungsverfahren. Beide verwenden *Sigmoid · ReLU* als Aktivierungsfunktionspaarung. Anders als bei der unveränderten Datenbasis konnte das Optimierungsverfahren Adam mit einem Ergebnis von 87,82% für die *Recall*-Klasse OOS das Optimierungsverfahren SGD mit einem Resultat von 77,16% übertreffen.

Eine Modellkonfiguration die schlechte Ergebnisse erzielt hat ist *Adam · Sigmoid · Softmax*. Sie sagt niemals die Klasse OOS voraus, sondern entscheidet sich bei jedem Bild für die Klasse In-Stock, weshalb der *Recall* für diese Klasse auch bei 100% liegt. Bei der zugehörigen *Accuracy* von 92,22% handelt es sich um keinen guten Wert, da dies nur den durchschnittlichen Anteil an In-Stock Bildern in dem Datenverhältnis widerspiegelt. Das gleiche gilt auch für das Ergebnis der Kostenfunktion. Durch die starke Verschiebung des Datenverhältnisses liegt der Maßstab für ein gutes Ergebnis viel höher.

Ähnlich unbefriedigende Resultate liefern die restlichen drei Modellkonfigurationen. Der Grund hierfür zeigt sich nach einem genaueren Blick auf die Ergebnisse während der Trainingsphase. Die Modellkonfiguration *Adam · Sigmoid · Sigmoid* hat ausschließlich für den Markt F die Klasse OOS zu 68,88% vorhergesagt. Für die anderen Märkte lag der *Recall* dieser Klasse bei 0. Ein ähnliches Verhalten zeigt auch die Modellkonfiguration *Adam · Softmax · TanH*. Lediglich für den Markt D konnte sie einen *Recall* der Klasse OOS von 80,78% bestimmen. Auch bei der Modellkonfiguration *Adam · Softmax · Sigmoid* ist zu erkennen, dass nur für einzelne Märkte ein OOS-*Recall* über Null ermittelt wurde. Während die Modellkonfiguration für die Märkte B, D, E und F den *Recall* der Klasse OOS bestimmte, lag das Ergebnis für die Märkte A und C bei 0.

Folglich fällt vielen Modellen das Erlernen der Klasse OOS für einzelne Märkte schwer. Während der Trainingsphase ist bei vielen Modellen zu erkennen, dass über mehrere Epochen hinweg der *Recall* für die Klasse OOS Null beträgt. Sobald in einer Epoche der *Recall* der Klasse OOS mit größer Null vorhergesagt wird, steigen in den Folgeepochen die Ergebnisse zu dieser Klassifizierungsmetrik.

In dem Zusammenhang zeigen die Daten, dass die Anzahl der Trainingsepochen zu klein war, da die Werte der Kostenfunktion von Epoche zu Epoche stetig sinken und folglich keine Konvergenz erreicht wurde. Mit dieser Erkenntnis lassen sich die schlechten Ergebnisse der Modellkonfigurationen *Adam · Sigmoid · Sigmoid* und *Adam · Softmax · TanH* auf das Konvergenzverhalten zurückführen, welches sie bei der Datengrundlage mit den augmentierten Bildern aufweisen. Die Lernkurven der beiden Modellkonfigurationen zeigen im Vergleich zu den anderen Modellkonfigurationen nur langsame Verbesserungen. Aus diesen Ergebnissen lässt sich schließen, dass für diese Modellkonfigurationen eine längere Trainingsphase bei erhöhter Datenanzahl notwendig ist, um eine Konvergenz der Kostenfunktion und der Klassifizierungsmetriken zu erreichen.

Folglich handelt es sich bei den Ergebnissen aus der Tabelle 10 nicht um die bestmöglichen Ergebnisse der Modellkonfigurationen. Jedoch zeigt sie, dass sich die Aktivierungsfunktionspaarung *Sigmoid · ReLU* im Vergleich zu den anderen Aktivierungsfunktionspaarungen am schnellsten an die gegebenen Daten anpasst. Dies ist bei der Betrachtung der Generierungsdauer nicht unerheblich, da sich auf Grundlage der augmentierten Daten die Dauer der Trainingsphase verfünffacht hat. Je nachdem wie regelmäßig ein OOS-Modell mit neuen Daten trainiert werden soll und wie umfangreich die Daten sind, beeinflusst dies die Wahl für die in der Praxis geeignete Modellkonfiguration.

Erneut scheint die Generierungsdauer der Modellkonfigurationen *Adam · Softmax · Sigmoid* und *Adam · Softmax · TanH* am kürzesten zu sein. Dementsprechend handelt es sich bei der Generierungsdauer in den unveränderten Daten nicht um Anomalien. Dennoch ist die Differenz zu den anderen Modellkonfigurationen im Verhältnis zur Laufzeit gering. Da die Generierungsdauer auch von der Konvergenz abhängt, bietet sich die Mo-

dellkonfiguration *Adam · Sigmoid · ReLU* an. Offensichtlich erreicht sie am schnellsten gute Ergebnisse, weshalb sie im Vergleich zu den anderen Modellkonfigurationen weniger Epochen in der Trainingsphase benötigt. Darum zählt sie für das regelmäßige Erstellen neuer OOS-Modelle zu den geeignetsten Kandidaten.

Zusammenfassend zeigen die auf den augmentierten Bildern basierenden OOS-Modelle im Vergleich zur unveränderten Datenbasis meist schlechtere Ergebnisse, trotz einer deutlich längeren Generierungsdauer. Es bietet sich an, zu überprüfen, ob die ca. 50:50 Aufteilung bei der unveränderten Datenbasis in der Praxis gleiche oder bessere Ergebnisse liefern kann, wie mit den Datensätzen der Testmärkte. Falls ja, ist bei der Generierung der OOS-Modelle die Anpassung des Datenverhältnisses der Trainingsdaten an die Realität nicht notwendig.

6 Diskussion

In dieser Masterarbeit wurden verschiedene OOS-Modelle mithilfe von KNNs generiert und evaluiert. In Kapitel 6.1 werden die Ergebnisse der generierten OOS-Modelle in Zusammenhang mit der gestellten Forschungsfrage diskutiert. Dabei werden die Leistungen und Grenzen der Modelle in Hinblick auf die OOS-Erkennung betrachtet. Abschließend zeigt Kapitel 6.2 Optionen auf, um die OOS-Modelle produktiv in den Markt zu bringen. Zudem wird auf vertiefende Forschungsthemen eingegangen, die das Ziel haben, die OOS-Modelle weiter anzupassen und zu verbessern. Hierfür werden offene Fragen und Herausforderungen diskutiert und mögliche Ansätze für weiterführende Arbeiten aufgezeigt.

6.1 Ergebnis

Die im Rahmen dieser Masterarbeit entwickelten OOS-Modelle erzielten je nach verwendeter Modellkonfiguration und Datengrundlage unterschiedlich gute Ergebnisse. Am wenigsten geeignet sind die Modellkonfigurationen, die das Optimierungsverfahren FTRL verwenden. Die anderen beiden Optimierungsverfahren Adam und SGD zeigen für viele Modellkonfigurationen deutlich bessere Ergebnisse, weshalb zukünftig eines dieser beiden Verfahren eingesetzt wird. Hinsichtlich der Aktivierungsfunktionen bietet sich in der letzten Schicht der *Sigmoid* oder der *Softmax* an. Sie erzielen im Vergleich zum TanH bessere Resultate.

Um die Forschungsfrage beantworten zu können, welche Kombinationen von Optimierungsverfahren und Aktivierungsfunktionen für das Erkennen einer OOS-Situation geeignet sind, müssen weitere Rahmenbedingungen für den Einsatz eines OOS-Modells geklärt werden. Wenn ein OOS-Modell regelmäßig mit aktuellen Daten neu trainiert werden soll, hat sich die Modellkonfiguration *Adam · Sigmoid · ReLU* aufgrund ihrer schnellen Konvergenz als besonders gut geeignet erwiesen. Zudem gehört sie zu den Modellkonfigurationen, die die besten Ergebnisse für die Kostenfunktion und die Klassifizierungsmetriken erzielen.

Wenn ein Modell einmalig trainiert und für alle Märkte bereitgestellt werden soll, bieten sich mehrere der top sechs untersuchten Modellkonfigurationen an. Da die Ergebnisse mit der augmentierten Datenbasis aufgrund der zu kurzen Trainingsphase nicht aussagekräftig sind, beziehen sich die Erkenntnisse auf die Resultate mit der unveränderten Datenbasis. Hierbei ist zu erkennen, dass die Modellkonfigurationen *Adam · Sigmoid · Sigmoid* und *Adam · Sigmoid · Softmax* für diesen Anwendungsfall nicht geeignet sind, weil sie zu große Schwankungen in den Ergebnissen zwischen den einzelnen Märkten aufweisen. Hingegen eignen sich die anderen vier Modellkonfigurationen *Adam · Sigmoid · ReLU*, *Adam · Softmax · Sigmoid*, *Adam · Softmax · TanH* und *SGD · Sigmoid · ReLU* für das Einsatzszenario.

Eine weitere Option ist, dass unterschiedliche Modellkonfigurationen für verschiedene Märkte individuell bereitgestellt werden. Es ist in den Ergebnissen der *Cross-Validierung* zu erkennen, dass sich die beste Modellkonfiguration je Markt unterscheidet. Gründe für die Unterschiede in den Ergebnissen finden sich im Sortiment. Der Markt A zeigt die besten Ergebnisse im Vergleich zu allen anderen Märkten, da er ausschließlich auf Bildern aus dem Getränkesortiment basiert. Hervorzuheben ist hierbei die Modellkonfiguration *Adam · Sigmoid · Softmax*, da sie perfekte Ergebnisse für den Testmarkt A erzielt. Schlussfolgernd können alle Modellkonfigurationen aus den top sechs für diese Option betrachtet werden, da jede von ihnen in mindestens einem Markt eine gute Leistung erbracht hat.

6.2 Ausblick

Unabhängig davon welche Modellkonfigurationen für die Märkte verwendet werden, zeigen die Ergebnisse, dass eine automatisierte Erkennung von OOS-Situationen möglich ist. Bei der Betrachtung des übergeordneten Ziels, einen selbstgebauten autonom fahrenden Roboter im Markt zu haben, der OOS-Situationen direkt am Regal erkennt, ist mit den verschiedenen OOS-Modellen bereits ein Meilenstein erreicht worden. Im nächsten Schritt muss ein Algorithmus entwickelt werden, der den Roboter in die Lage versetzt, sich selbständig im Markt zu bewegen und Bildmaterial der Regalfächer zu erstellen. Ein weiterer Meilenstein ist die Hardware-Zusammenstellung des Roboters und die Integration des OOS-Modells in den Roboter. Dabei ist es unerheblich, ob das zugrundeliegende OOS-Modell auf der Hardware des Roboters, auf einem Rechner im Markt oder in einem zentralen Rechenzentrum gespeichert ist. Wichtig ist, dass eine Schnittstelle existiert, über die der Roboter mit dem OOS-Modell kommunizieren kann.

Zur Verbesserung des WaWis könnte der Roboter in Kombination mit einer Kamera und einem Barcode-Scanner direkt mitteilen, welcher Artikel fehlt, statt lediglich darauf hinzuweisen, dass es eine OOS-Situation gibt. Dies würde die Bearbeitungszeit von OOS-Situationen verkürzen und somit den monetären Nutzen steigern.

Um eine kontinuierliche Optimierung des OOS-Modells zu erreichen, muss darüber hinaus untersucht werden, wie Falschklassifizierungen korrigiert und wie diese in das Modell integriert werden können. Dadurch sollen die Fälle minimiert werden, in denen das Modell eine Fachkraft fälschlicherweise über eine OOS-Situation informiert. Gleichzeitig muss auch eine falsch klassifizierte In-Stock-Situation gemeldet werden können. Eine Möglichkeit, Zusammenhänge zwischen den Falschklassifizierungen zu finden, besteht darin, die Sortimente mit den häufigsten Fehlklassifizierungen zu analysieren. Möglicherweise können durch die daraus resultierenden Erkenntnisse Rückschlüsse auf die Marktbedingungen gezogen werden, die das Erkennen von OOS-Situationen begünstigen.

In einer weiterführenden Forschungsarbeit kann untersucht werden, ob das Datenverhältnis, mit dem ein OOS-Modell trainiert wird, Auswirkungen auf die erzielten Ergebnisse

hat. Hierzu können die bestehenden top sechs Modellkonfigurationen verwendet werden. Jedoch sollte die Trainingsphase deutlich mehr Epochen haben, um sicherzustellen, dass die Kostenfunktion gegen das bestmögliche Minimum konvergiert.

Eine weitere Möglichkeit die OOS-Modelle zu verbessern ist eine vertiefende Evaluation der möglichen Modellkonfigurationen. Nachdem herausgefunden wurde, dass das Optimierungsverfahren Adam die besten Modellergebnisse erzielt, können in einer fortführenden Arbeit weitere Untersuchungen zur optimalen Einstellung der Parameter durchgeführt werden. Zudem können auch die Varianten von Adam hinsichtlich ihrer Leistung untersucht werden. Dazu zählen die Optimierungsverfahren Adamax, Nadam, AdaBound und AmsGrad.

Darüber hinaus kann erforscht werden, wie sich die Anzahl der Neuronen in den einzelnen Schichten auf die Leistung der OOS-Modelle auswirkt. Außerdem kann evaluiert werden, ob es bessere Alternativen zum VGG16-Modell gibt, auf dem die OOS-Modelle basieren. Zu den möglichen Kandidaten könnte bspw. das VGG19- oder das ResNet-Modell zählen.

Basierend auf den Ergebnissen dieser Masterarbeit ist die Empfehlung, ein entwickeltes OOS-Modell schnellstmöglich im Markt zu implementieren, um praxisnahe Erfahrungen zu sammeln und die Leistung des Modells unter realen Bedingungen zu bewerten. Die daraus resultierenden Erkenntnisse sollten genutzt werden, um das Modell kontinuierlich zu optimieren. Durch eine iterative Verbesserung kann die Leistung des OOS-Modells weiter gesteigert und somit eine erfolgreiche Automatisierung der OOS-Erkennung erreicht werden.

Literatur

- Abadi, Martín u. a. (2016). *TensorFlow: A system for large-scale machine learning*. arXiv: [1605.08695](https://arxiv.org/abs/1605.08695).
- Abayomi-Alli, Olusola O. u. a. (2022). „Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review“. In: *Electronics* 11.22, S. 3795. ISSN: 2079-9292. DOI: [10.3390/electronics11223795](https://doi.org/10.3390/electronics11223795). URL: <http://dx.doi.org/10.3390/electronics11223795>.
- Ackermann, P (2021). *Webentwicklung: Das Handbuch für Fullstack-Entwickler*. Verfügbar von: ProQuest Ebook Central [22. März 2023]. Bonn: Rheinwerk Verlag. URL: <https://www.proquest.com/ebookcentral>.
- Albers, Sönke u. a., Hrsg. (2009). *Methodik der empirischen Forschung*. Gabler Verlag. DOI: [10.1007/978-3-322-96406-9](https://doi.org/10.1007/978-3-322-96406-9). URL: <https://doi.org/10.1007/978-3-322-96406-9>.
- Arlot, Sylvain und Alain Celisse (2010). „A survey of cross-validation procedures for model selection“. In: *Statistics Surveys* 4.none, S. 40 –79. DOI: [10.1214/09-SS054](https://doi.org/10.1214/09-SS054). URL: <https://doi.org/10.1214/09-SS054>.
- Awad, Mariette und Rahul Khanna (2015). *Efficient Learning Machines*. Apress. DOI: [10.1007/978-1-4302-5990-9](https://doi.org/10.1007/978-1-4302-5990-9). URL: <https://doi.org/10.1007/978-1-4302-5990-9>.
- Banerjee, Chaity, Tathagata Mukherjee und Eduardo Pasiliao (2020). „The Multi-Phase ReLU Activation Function“. In: *Proceedings of the 2020 ACM Southeast Conference*. ACM SE '20. Tampa, FL, USA: Association for Computing Machinery, 239–242. ISBN: 9781450371056. DOI: [10.1145/3374135.3385313](https://doi.org/10.1145/3374135.3385313). URL: <https://doi.org/10.1145/3374135.3385313>.
- Bergmeir, Christoph und José M. Benítez (2012). „On the use of cross-validation for time series predictor evaluation“. In: *Information Sciences* 191. Data Mining for Software Trustworthiness, S. 192–213. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2011.12.028>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025511006773>.
- Bradski, Gary R. und Adrian Kaehler (2008). *Learning OpenCV: [computer vision with the OpenCV library ; software that sees]*. 1. ed. HBZ: HT015642354; Verfasserangabe: Gary Bradski and Adrian Kaehler; Umfang: XVII, 555 S. : Ill., graph. Darst.; Quelldatenbank: DE-605; hbz Verbund-ID: HT015642354; (pbk) : £30.99. Beijing ; Köln [u.a.]: O'Reilly. ISBN: 0-596-51613-4; 978-0-596-51613-0. URL: http://digitale-objekte.hbz-nrw.de/storage/2010/01/16/file_93/3613583.pdf.
- Chollet, François (März 2015). *Add initial public version of Keras*. URL: <https://github.com/keras-team/keras/commit/37a1db225420851cc668600c49697d9a2057f098> (besucht am 25. 11. 2022).

- Crone, Sven F. (2010). *Neuronale Netze zur Prognose und Disposition im Handel*. Gabler. DOI: [10.1007/978-3-8349-8631-3](https://doi.org/10.1007/978-3-8349-8631-3). URL: <https://doi.org/10.1007/978-3-8349-8631-3>.
- Fawcett, Tom (2006). „An introduction to ROC analysis“. In: *Pattern Recognition Letters* 27.8. ROC Analysis in Pattern Recognition, S. 861–874. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2005.10.010>. URL: <https://www.sciencedirect.com/science/article/pii/S016786550500303X>.
- Gabbay, Freddy und Freddy Gabbay (2022). *Computational Optimizations for Machine Learning*. HBZ: HT021611261; Umfang: 1 electronic resource (276 pages); Quelldatenbank: DE-605; Online-Ressource [Kann nicht per Fernleihe bestellt werden!]; English; hbz Verbund-ID: HT021611261. Basel: MDPI - Multidisciplinary Digital Publishing Institute. ISBN: 9783036531878. URL: <https://mdpi.com/books/pdfview/book/5018>;<https://directory.doabooks.org/handle/20.500.12854/79633>.
- Goodfellow, Ian, Yoshua Bengio und Aaron Courville (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Google (Juni 2021). *Batch normalization*. URL: <https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#12> (besucht am 11. 12. 2022).
- Gupta, Sandeep, Josh Gordon und Karmel Allison (Dez. 2018). *Standardizing on Keras: Guidance on High-level APIs in TensorFlow 2.0*. URL: <https://blog.tensorflow.org/2018/12/standardizing-on-keras-guidance.html> (besucht am 25. 11. 2022).
- Géron, Aurélien (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Second edition, Third Release. Erscheint auch als: Druck-Ausgabe: Géron, Aurélien: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, ISBN 978-1-492-03264-9. HBZ: HT020257277; Verfasserangabe: Aurélien Géron; Umfang: 1 Online-Ressource (xxv, 819 Seiten) ; Illustrationen; Quelldatenbank: DE-605; Online-Ressource [Kann nicht per Fernleihe bestellt werden!]; hbz Verbund-ID: HT020257277. Beijing ; Boston ; Fannham ; Sebastopol ; Tokyo: O'Reilly. ISBN: 978-1-492-03261-8. URL: http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2245240&site=ehost-live&ebv=EB&ppid=pp_Cover.
- He, Kaiming u. a. (2015). „Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification“. In: *CoRR* abs/1502.01852. arXiv: [1502.01852](https://arxiv.org/abs/1502.01852). URL: <http://arxiv.org/abs/1502.01852>.
- Herculano-Houzel, Suzana (2009). „The human brain in numbers: a linearly scaled-up primate brain“. In: *Novel molecular and cellular approaches to brain mapping*.
- Hofer, Florian (2009). *Management der Filiallogistik im Lebensmitteleinzelhandel*. Hrsg. von Michael Eßig und Wolfgang Stölzle. Gabler. DOI: [10.1007/978-3-8349-9401-1](https://doi.org/10.1007/978-3-8349-9401-1). URL: <https://doi.org/10.1007/978-3-8349-9401-1>.

- Hopp, Hans-M. und Fabian Stern (2021). *Docker Handbuch für Einsteiger*. ger. BMU Verlag. ISBN: 9783966450669. URL: <https://content-select.com/de/portal/media/view/5e467833-bd6c-4c74-96b9-0f63b0dd2d03>.
- IBM (2021). *CRISP-DM-Hilfe*. URL: <https://www.ibm.com/docs/de/spss-modeler/saas?topic=dm-crisp-help-overview> (besucht am 26. 11. 2022).
- Jarrett, Kevin u. a. (2009). „What is the best multi-stage architecture for object recognition?“ In: *2009 IEEE 12th International Conference on Computer Vision*, S. 2146–2153. DOI: [10.1109/ICCV.2009.5459469](https://doi.org/10.1109/ICCV.2009.5459469).
- Jeffries, Zay (1960). *Charles Franklin Kettering - (1876-1958)*. National Academy of Sciences. URL: <http://nasonline.org/publications/biographical-memoirs/memoir-pdfs/kettering-charles.pdf> (besucht am 23. 08. 2022).
- Keras (o.J.). *Keras*. URL: <https://keras.io/about/> (besucht am 26. 11. 2022).
- Kingma, Diederik P. und Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. DOI: [10.48550/ARXIV.1412.6980](https://doi.org/10.48550/ARXIV.1412.6980). URL: <https://arxiv.org/abs/1412.6980>.
- Kinnebrock, Werner (2018). *Neuronale Netze - Grundlagen, Anwendungen, Beispiele*. Berlin, Boston: Oldenbourg Wissenschaftsverlag. ISBN: 9783486786361. DOI: [doi:10.1515/9783486786361](https://doi.org/10.1515/9783486786361). URL: <https://doi.org/10.1515/9783486786361>.
- Krizhevsky, Alex, Ilya Sutskever und Geoffrey E Hinton (2012). „ImageNet Classification with Deep Convolutional Neural Networks“. In: *Advances in Neural Information Processing Systems*. Hrsg. von F. Pereira u. a. Bd. 25. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- Kulkarni, Chinmay Ajit, Mohith R und Shylaja S Sharath (2022). „Custom Binary Cross Entropy Based Anomaly Detection in Bank Transactions Using Deep Convolutional Neural Network“. In: *The 23rd International Conference on Information Integration and Web Intelligence*. iiWAS2021. Linz, Austria: Association for Computing Machinery, 319–323. ISBN: 9781450395564. DOI: [10.1145/3487664.3487708](https://doi.org/10.1145/3487664.3487708). URL: <https://doi.org/10.1145/3487664.3487708>.
- Larner, A.J. (2021). *The 2x2 Matrix*. Springer International Publishing. DOI: [10.1007/978-3-030-74920-0](https://doi.org/10.1007/978-3-030-74920-0). URL: <https://doi.org/10.1007/978-3-030-74920-0>.
- Lu, Le u. a., Hrsg. (2017). *Deep Learning and Convolutional Neural Networks for Medical Image Computing*. Springer International Publishing. DOI: [10.1007/978-3-319-42999-1](https://doi.org/10.1007/978-3-319-42999-1). URL: <https://doi.org/10.1007/978-3-319-42999-1>.
- Luhmann, Heiko J (2020). *Hirnpotentiale*. de. 1. Aufl. Berlin, Germany: Springer.
- Luo, Luo u. a. (2018). „Sketched Follow-The-Regularized-Leader for Online Factorization Machine“. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 1900–1909. ISBN: 9781450355520. DOI: [10.1145/3219819.3220044](https://doi.org/10.1145/3219819.3220044). URL: <https://doi.org/10.1145/3219819.3220044>.

- Manning, Christopher D, Prabhakar Raghavan und Hinrich Schütze (2008). *Introduction to information retrieval*. Cambridge University Press.
- Matsugu, Masakazu u. a. (2003). „Subject independent facial expression recognition with robust face detection using a convolutional neural network“. In: *Neural Networks* 16.5. Advances in Neural Networks Research: IJCNN '03, S. 555–559. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(03\)00115-1](https://doi.org/10.1016/S0893-6080(03)00115-1). URL: <https://www.sciencedirect.com/science/article/pii/S0893608003001151>.
- Matzka, Stephan (2021). *Crashkurs KI im Unternehmen: Alles, was Sie über Data Science wissen müssen*. 1. Auflage 2021. Haufe Fachbuch. HBZ: HT021066346; Umfang: Online-Ressource (162 S.); Quelldatenbank: DE-605; Online-Ressource [Kann nicht per Fernleihe bestellt werden!]; 9783648149201; hbz Verbund-ID: HT021066346. Planegg: Haufe Lexware. ISBN: 9783648149225.
- McMahan, Brendan (2011). „Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization“. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Hrsg. von Geoffrey Gordon, David Dunson und Miroslav Dudík. Bd. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, S. 525–533. URL: <https://proceedings.mlr.press/v15/mcmahan11b.html>.
- McMahan, H. Brendan und Matthew J. Streeter (2010). „Adaptive Bound Optimization for Online Convex Optimization“. In: *CoRR* abs/1002.4908. arXiv: [1002.4908](https://arxiv.org/abs/1002.4908). URL: <http://arxiv.org/abs/1002.4908>.
- McMahan, H. Brendan u. a. (2013). „Ad Click Prediction: a View from the Trenches“. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.
- Nielsen, M.A. (2015). *Neural Networks and Deep Learning*. Determination Press. URL: <https://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf>.
- Niu, Mengting u. a. (2018). „RFAmyloid: A Web Server for Predicting Amyloid Proteins“. In: *International Journal of Molecular Sciences* 19.7, S. 2071. ISSN: 1422-0067. DOI: [10.3390/ijms19072071](https://doi.org/10.3390/ijms19072071). URL: <http://dx.doi.org/10.3390/ijms19072071>.
- Openpyxl (2022). *Openpyxl 3.0.10*. URL: <https://openpyxl.readthedocs.io/en/stable/changes.html#id23> (besucht am 24.03.2022).
- Provost, Foster und Tom Fawcett (Dez. 1999). „Analysis and Visualization of Classifier Performance: Comparison Under Imprecise Class and Cost Distributions“. In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* 43-48.
- Python Software Foundation (2023a). *How do I write a function with output parameters (call by reference)?* URL: <https://docs.python.org/3/faq/programming.html#>

- [how-do-i-write-a-function-with-output-parameters-call-by-reference](#) (besucht am 26. 03. 2023).
- Python Software Foundation (2023b). *Why did changing list y also change list x?* URL: <https://docs.python.org/3/faq/programming.html#why-did-changing-list-y-also-change-list-x> (besucht am 31. 03. 2023).
- PyYAML (o.J.). *PyYAML Documentation*. URL: <https://pyyaml.org/wiki/PyYAMLDocumentation> (besucht am 24. 03. 2023).
- Rashid, T. und F. Langenau (2017). *Neuronale Netze selbst programmieren : Ein verständlicher Einstieg mit Python*. Animals. O'Reilly. ISBN: 9783960090434. URL: <https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1620168&lang=de&site=eds-live>.
- Rauscher, Markus (2004). *Kuenstliche neuronale Netze zur Risikomessung bei Aktien und Renten*. Deutscher Universitaetsverlag. DOI: [10.1007/978-3-322-81863-8](https://doi.org/10.1007/978-3-322-81863-8).
- REWE Systems GmbH (o.J.). *REWE Systems in Zahlen*. URL: <https://www.rewe-systems.com/ueber-uns> (besucht am 26. 08. 2022).
- Ruder, Sebastian (2016). „An overview of gradient descent optimization algorithms“. In: *CoRR* abs/1609.04747. arXiv: [1609.04747](https://arxiv.org/abs/1609.04747). URL: <http://arxiv.org/abs/1609.04747>.
- Scherer, Andreas (1997). *Neuronale Netze*. Teubner Verlag. DOI: [10.1007/978-3-322-86830-5](https://doi.org/10.1007/978-3-322-86830-5).
- scikit-learn (o.J.). *sklearn.metrics.f1_score*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (besucht am 28. 02. 2023).
- SciPy (2022). *SciPy 1.10.0 Release Notes*. URL: <https://docs.scipy.org/doc/scipy/reference/index.html> (besucht am 24. 03. 2023).
- Shalev-Shwartz, Shai und Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. ISBN: 1107057132, 9781107057135.
- Shorten, Connor und Taghi M. Khoshgoftaar (Juli 2019). „A survey on Image Data Augmentation for Deep Learning“. In: *Journal of Big Data* 6.1. DOI: [10.1186/s40537-019-0197-0](https://doi.org/10.1186/s40537-019-0197-0). URL: <https://doi.org/10.1186/s40537-019-0197-0>.
- [REDACTED] (besucht am 30. 12. 2022).
- [REDACTED] (besucht am 26. 08. 2022).
- Simonyan, Karen und Andrew Zisserman (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. DOI: [10.48550/ARXIV.1409.1556](https://arxiv.org/abs/1409.1556). URL: <https://arxiv.org/abs/1409.1556>.

- Singh, Satyendra (2006). „Impact of color on marketing“. In: *Management Decision* 44.6, S. 783–789. DOI: [10.1108/00251740610673332](https://doi.org/10.1108/00251740610673332).
- Sokolova, Marina und Guy Lapalme (2009). „A systematic analysis of performance measures for classification tasks“. In: *Information Processing and Management* 45.4, S. 427–437. ISSN: 0306-4573. DOI: <https://doi.org/10.1016/j.ipm.2009.03.002>. URL: <https://www.sciencedirect.com/science/article/pii/S0306457309000259>.
- TensorFlow 2 (2022a). *TensorFlow 2.11.0*. URL: <https://github.com/tensorflow/tensorflow/releases/tag/v2.11.0> (besucht am 24.03.2023).
- (Dez. 2022b). *tf.keras.losses.BinaryCrossentropy*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/losses/BinaryCrossentropy (besucht am 01.03.2023).
- (Nov. 2022c). *tf.keras.metrics.Accuracy*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/metrics/Accuracy (besucht am 27.02.2023).
- (Nov. 2022d). *tf.keras.metrics.Precision*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/metrics/Precision (besucht am 27.02.2023).
- (2022e). *tf.keras.optimizers.Optimizer*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/optimizers/Optimizer (besucht am 30.01.2022).
- (Nov. 2022f). *tf.keras.preprocessing.image.ImageDataGenerator*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator (besucht am 05.12.2022).
- (2023a). *tf.keras.initializers.GlorotUniform*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/initializers/GlorotUniform (besucht am 25.03.2023).
- (2023b). *tf.keras.initializers.HeUniform*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/initializers/HeUniform (besucht am 25.03.2023).
- (2023c). *tf.keras.layers.Dense*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/layers/Dense (besucht am 25.03.2023).
- (2023d). *tf.keras.layers.Flatten*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/layers/Flatten (besucht am 25.03.2023).
- (2023e). *tf.keras.Model*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/Model (besucht am 26.03.2023).
- (2023f). *tf.keras.optimizers.Adam*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/optimizers/Adam (besucht am 25.03.2023).

- TensorFlow 2 (2023g). *tf.keras.optimizers.experimental.Ftrl*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/optimizers/experimental/Ftrl (besucht am 25.03.2023).
- (2023h). *tf.keras.optimizers.experimental.SGD*. URL: https://www.tensorflow.org/versions/r2.11/api_docs/python/tf/keras/optimizers/experimental/SGD (besucht am 25.03.2023).
- Wenighofer, Robert und Robert Galler (Nov. 2018). „Künstliche neuronale Netze im Einsatz zur Gebrauchstauglichkeitsbewertung des einschaligen maschinellen Tunnelbaus“. In: *BHM Berg- und Hüttenmännische Monatshefte* 163.12, S. 517–523. DOI: [10.1007/s00501-018-0799-4](https://doi.org/10.1007/s00501-018-0799-4). URL: <https://doi.org/10.1007/s00501-018-0799-4>.
- Wirth, Rüdiger und Jochen Hipp (2000). „Crisp-dm: towards a standard process modell for data mining“. In:
- Yu, Feng u. a. (2015). „A new case of complete primary cerebellar agenesis: clinical and imaging findings in a living patient“. en. In: *Brain* 138.Pt 6, e353.
- Zeiler, Matthew D. (2012). „ADADELTA: An Adaptive Learning Rate Method“. In: *CoRR* abs/1212.5701. arXiv: [1212.5701](http://arxiv.org/abs/1212.5701). URL: <http://arxiv.org/abs/1212.5701>.
- Zell, Andreas (2000). „Simulation neuronaler Netze“. HBZ: HT012815527; Verfasserangabe: von Andreas Zell; Umfang: 624 S. : Ill., zahlr. graph. Darst.; Quelldatenbank: DE-605; Teilw. zugl. Stuttgart, Univ., Habil.-Schr.; hbz Verbund-ID: HT012815527. thesis. München [u.a.] ISBN: 3-486-24350-0.
- Zheng, Shuai und James T. Kwok (2017). „Follow the Moving Leader in Deep Learning“. In: *Proceedings of the 34th International Conference on Machine Learning*. Hrsg. von Doina Precup und Yee Whye Teh. Bd. 70. Proceedings of Machine Learning Research. PMLR, S. 4110–4119. URL: <https://proceedings.mlr.press/v70/zheng17a.html>.
- Zhu, Jianhao und Zong Chen (2020). „Comparative analysis of various new activation functions based on convolutional neural network“. In: *Journal of Physics: Conference Series* 1676.1, S. 012228. DOI: [10.1088/1742-6596/1676/1/012228](https://doi.org/10.1088/1742-6596/1676/1/012228). URL: <https://dx.doi.org/10.1088/1742-6596/1676/1/012228>.

Anhang

A [redacted]

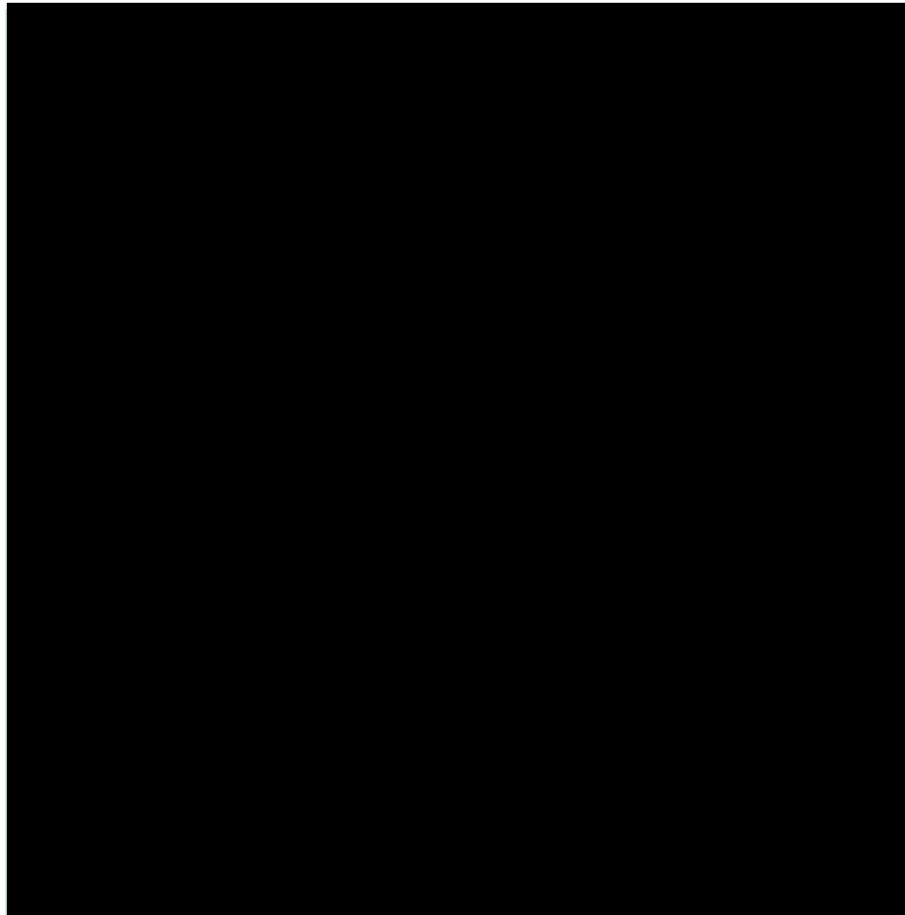
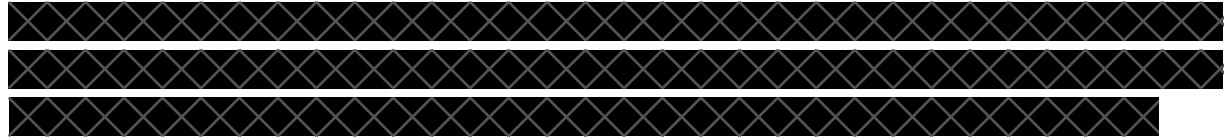


Abbildung 28: [redacted]¹⁵⁴

B Cross-Validierung auf Marktebene

Die nachfolgende Abbildung 29 stellt die sechs Kombinationen dar, mit denen ein OOS-Modell trainiert und getestet wird. Nähere Informationen finden sich in Kapitel 3.4.2 und 3.4.3 wieder.

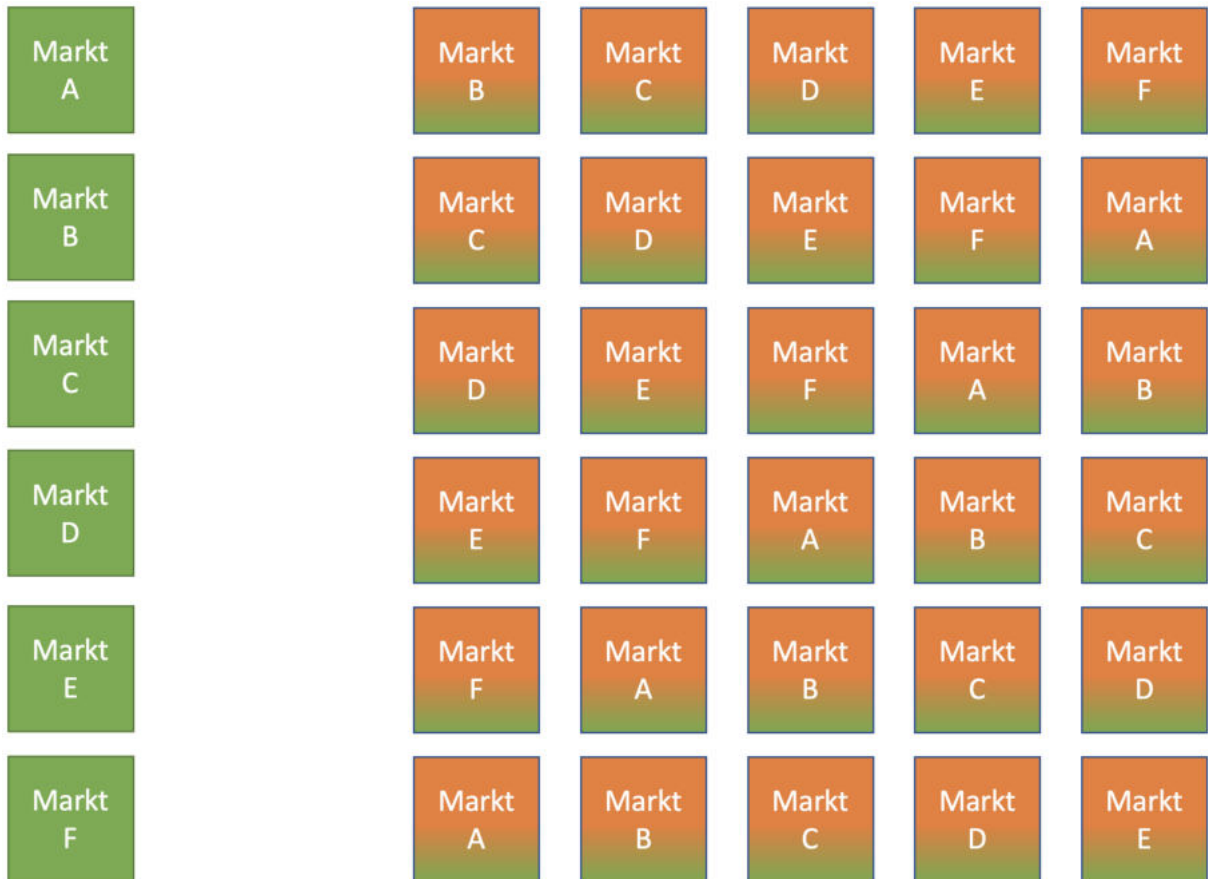


Abbildung 29: Cross-Validierung auf Marktebene

C Codedokumentationen

C.1 Docker

C.1.1 Implementierung Dockerfile

```
FROM python:3.10-slim-bullseye
RUN pip install tensorflow openpyxl PyYAML matplotlib scipy
WORKDIR /
```

Listing 19: Dockerfile

C.1.2 Implementierung Docker-Compose

```
version: "3"
services:
  oos_generator:
    build: .
    volumes:
      - /path/to/folder
    command: [ "python", "main.py" ]
    deploy:
      resources:
        limits:
          cpus: '3'
          memory: 8G
```

Listing 20: docker-compose.yml

C.2 Augmentation

C.2.1 Definierte Parameter für das Augmentieren von Bildern

```
def define_augmentation_rules():
    datagen = ImageDataGenerator(
        rescale=1. / 224,
        horizontal_flip=True,
        rotation_range=30,
        fill_mode='nearest',
        width_shift_range=0.2,
        height_shift_range=0.2,
        brightness_range=[0.5, 1.5],
        zoom_range=0.5,
        validation_split=0.3
    )
    return datagen
```

Listing 21: Definierte Parameter für das Augmentieren von Bildern

C.2.2 Speichern von augmentierten Bildern

```

def generate_augmented_not_oos_pictures():
    for markt in Markt:
        # Pfad zum Ordner "!00S"
        folder_path = "assets/Bilder/Datengrundlage_Test/" +
            markt.name + "/"

        # Erstellen einer Instanz vom ImageDataGenerator
        datagen = get_image_definition()

        # Pfad fuer den neuen Ordner, in dem die
        # augmentierten Bilder gespeichert werden sollen
        save_to_dir = "assets/Bilder/Datengrundlage-
            Augmentiert/" + markt.name + "!00S"

        # Pro Originalbild 9 augmentierte Bilder erstellen
        # und speichern
        for i in range(9):
            train_it = datagen.flow_from_directory(
                folder_path,
                class_mode='binary',
                batch_size=1425,
                target_size=(224, 224),
                save_to_dir=save_to_dir,
                save_prefix="augmented_" + str(i) + "_",
                save_format='jpg',
            )
            train_it.next()

def get_image_definition():
    augmentedImageDefinition = tf.keras.preprocessing.image.
        ImageDataGenerator(
            rescale=1. / 224,
            horizontal_flip=True,
            rotation_range=15,
            fill_mode='nearest',
            width_shift_range=0.2,
            height_shift_range=0.2,
            brightness_range=[0.5, 1.5],
            zoom_range=0.5,
        )
    return augmentedImageDefinition

```

Listing 22: Speichern von augmentierten Bildern

C.3 Laden von Daten

C.3.1 Laden der Testdaten

```
def get_test_data(datagen, directory):  
    test_it = datagen.flow_from_directory(  
        directory=directory,  
        class_mode='categorical',  
        batch_size=64,  
        target_size=(224, 224),  
        subset='validation')  
    return test_it
```

Listing 23: Laden der Testdaten

C.3.2 Laden der Daten von einem Markt

```
def get_markt_data(datagen, directory):  
    markt_data = datagen.flow_from_directory(  
        directory=directory,  
        class_mode='categorical',  
        batch_size=64,  
        target_size=(224, 224))  
    return markt_data
```

Listing 24: Laden der Marktdaten

C.4 Enums

C.4.1 Enum Markt

```
class Markt(Enum):  
    Markt_A = 0,  
    Markt_B = 1,  
    Markt_C = 2,  
    Markt_D = 3,  
    Markt_E = 4,  
    Markt_F = 5  
    Kein_Markt = -1
```

Listing 25: Enum Markt

C.4.2 Enum Aktivierungsfunktion

```
class Aktivierungsfunktion(Enum):  
    ReLU = 0,  
    sigmoid = 1,  
    tanh = 2,  
    softmax = 3
```

Listing 26: Enum Aktivierungsfunktion

C.4.3 Enum Optimierungsverfahren

```
class Optimierungsverfahren(Enum):  
    SGD = 0,  
    Adam = 1,  
    Ftrl = 2
```

Listing 27: Enum Optimierungsverfahren

C.5 Klassen

C.5.1 Klasse ConfigKNN

```

class ConfigKNN:
    def __init__(self,
                 excluded_folder=Markt.Kein_Markt,
                 activation_function_128_units=
                     Aktivierungsfunktion.ReLU,
                 activation_function_2_units=
                     Aktivierungsfunktion.sigmoid,
                 optimization_method=Optimierungsverfahren.
                     SGD):
        self.excluded_folder = excluded_folder
        self.activation_function_128_units =
            activation_function_128_units
        self.activation_function_2_units =
            activation_function_2_units
        self.optimization_method = optimization_method

    def __str__(self):
        return "optimization_method: " \
            + self.optimization_method.name + " -" \
            + activation_function_128_units: " \
            + self.activation_function_128_units.name + ' -' \
            + activation_function_2_units: ' \
            + self.activation_function_2_units.name + ' -' \
            + excluded_folder: ' \
            + self.excluded_folder.name

```

Listing 28: Speichern der aktuellen KNN-Konfigurationen

C.5.2 Klasse CustomCallback

```

class CustomCallback(tf.keras.callbacks.Callback):
    def __init__(self):
        super().__init__()
        self.epoch_start_time = None
        self.epoch_training_losses = None
        self.epoch_training_accuracies = None
        self.epoch_training_recall_oos = None
        self.epoch_training_recall_not_oos = None
        self.epoch_test_losses = None
        self.epoch_test_accuracies = None
        self.epoch_test_recall_oos = None
        self.epoch_test_recall_not_oos = None
        self.epoch_durations = None

```

```
def on_train_begin(self, logs=None):
    self.epoch_training_losses = []
    self.epoch_training_accuracies = []
    self.epoch_training_recall_oos = []
    self.epoch_training_recall_not_oos = []
    self.epoch_test_losses = []
    self.epoch_test_accuracies = []
    self.epoch_test_recall_oos = []
    self.epoch_test_recall_not_oos = []
    self.epoch_durations = []

def on_epoch_begin(self, epoch, logs=None):
    self.epoch_start_time = time.time()

def on_epoch_end(self, epoch, logs=None):
    if logs is None:
        logs = {}
    self.epoch_training_losses.append(logs.get('loss'))
    self.epoch_training_accuracies.append(logs.get('
        accuracy'))
    self.epoch_training_recall_oos.append(logs.get('
        recall_oos'))
    self.epoch_training_recall_not_oos.append(logs.get('
        recall_not_oos'))
    self.epoch_test_losses.append(logs.get('val_loss'))
    self.epoch_test_accuracies.append(logs.get('
        val_accuracy'))
    self.epoch_test_recall_oos.append(logs.get('
        val_recall_oos'))
    self.epoch_test_recall_not_oos.append(logs.get('
        val_recall_not_oos'))
    self.epoch_durations.append(time.time() - self.
        epoch_start_time)
```

Listing 29: Custom Callback

C.6 Kommentierte Methoden

C.6.1 Kommentierte Version der KNN-Architektur

```
def define_model(config_knn):
    # Vortrainierte VGG16-Modell wird geladen
    model = tf.keras.applications.VGG16(include_top=False,
        input_shape=(224, 224, 3))
    # Die geladenen Schichten muessen nicht neu trainiert
    werden
    for layer in model.layers:
        layer.trainable = False
    # Die Fully-Connected-Schichten werden definiert
    flat = tf.keras.layers.Flatten()(model.layers[-1].output)
    output_layer_1 = get_next_layer(128, config_knn.
        activation_function_128_units, flat)
    output_layer_2 = get_next_layer(2, config_knn.
        activation_function_2_units, output_layer_1)

    # Die Schichten des Modells werden definiert
    model = tf.keras.models.Model(inputs=model.inputs,
        outputs=output_layer_2)

    # Das Optimierungsverfahren wird definiert
    opt = get_optimization_method(config_knn.
        optimization_method)
    # Die Klassifizierungsmetriken und die Loss-Funktion
    werden definiert
    loss_fn = tf.keras.losses.BinaryCrossentropy()
    recall_not_oos = tf.keras.metrics.Recall(name='
        recall_not_oos', class_id=0)
    recall_oos = tf.keras.metrics.Recall(name='recall_oos',
        class_id=1)

    # Fuege die Metriken dem Modell hinzu
    model.compile(optimizer=opt, loss=loss_fn, metrics=['
        accuracy', recall_not_oos, recall_oos])
    return model
```

Listing 30: Kommentierte Version der KNN-Architektur

C.6.2 Kommentierte Version der Zusammenstellung von Trainingsdaten

```
def copy_images_with_exclusion(src_dir, dest_dir, exclude_dir
=None):
    # Loesche vorhandene Dateien und Unterordner im
    Zielordner
    for file_or_dir in os.listdir(dest_dir):
        file_or_dir_path = os.path.join(dest_dir, file_or_dir
        )
        if os.path.isdir(file_or_dir_path):
            shutil.rmtree(file_or_dir_path)
        else:
            os.remove(file_or_dir_path)

    # Erstelle die gewuenschte Ordnerstruktur im
    Zielverzeichnis
    os.makedirs(os.path.join(dest_dir, '!00S'))
    os.makedirs(os.path.join(dest_dir, '00S'))

    # Durchlaufe die Unterordner im Quellverzeichnis
    for subdir in os.listdir(src_dir):
        # if subdir == exclude_dir:
        subdirString = bytes(subdir, 'utf-8').decode('
        unicode_escape')
        exclude_dirString = bytes(exclude_dir.name, 'utf-8').
        decode('unicode_escape')
        if subdirString == exclude_dirString:
            continue
        for folder in os.listdir(os.path.join(src_dir, subdir
        )):
            if folder == '00S':
                for img in os.listdir(os.path.join(src_dir,
                subdir, folder)):
                    shutil.copy(os.path.join(src_dir, subdir,
                    folder, img), os.path.join(dest_dir,
                    '00S'))
            elif folder == '!00S':
                for img in os.listdir(os.path.join(src_dir,
                subdir, folder)):
                    shutil.copy(os.path.join(src_dir, subdir,
                    folder, img), os.path.join(dest_dir,
                    '!00S'))
```

Listing 31: Kommentierte Version der Zusammenstellung von Trainingsdaten

C.6.3 Kommentierte Version der Validierung von Modellkonfigurationen

```

def run_cross_validation():
    print("Lese Configurationsdatei")
    # Lesen der Configurationsdatei
    with open("ressources/config/config.yaml", "r") as file:
        config = yaml.load(file, Loader=yaml.FullLoader)

    # Initialisieren der KNN-Modellbau Eigenschaften
    for optimization_method in Optimierungsverfahren:
        for activation_function_2 in Aktivierungsfunktion:
            if activation_function_2.name in config['knn']['
                exception_activation_funktion_2']:
                continue
            for activation_function_128 in
                Aktivierungsfunktion:
                model_evaluate_metrics = []
                training_duration_models = []
                workbook = get_excel_workbook()
                worksheet = get_excel_worksheet(workbook)
                r = 1
                # Definiere die neue Modellkonfiguration
                config_knn = ConfigKNN(
                    excluded_folder=Markt.Kein_Markt,
                    activation_function_2_units=
                        activation_function_2,
                    activation_function_128_units=
                        activation_function_128,
                    optimization_method=optimization_method
                )
                # Starte Cross-Validation
                for excluded_folder in Markt:
                    if excluded_folder == Markt.Kein_Markt:
                        continue
                    config_knn.excluded_folder =
                        excluded_folder
                    print("Konfiguration KNN: " + config_knn.
                        __str__())
                    r, evaluate_metrics,
                        training_duration_model = run_model(
                            config, config_knn, worksheet, r)
                    model_evaluate_metrics.append(
                        evaluate_metrics)
                    training_duration_models.append(
                        training_duration_model)
                print("Berechne den Durchschnitt der Cross-
                    Validation beim Modellbau")

```

```

        average_evaluate_cross_validation(worksheet,
            r, model_evaluate_metrics,
            training_duration_models)
    print("Speichere die Excel Tabelle")
    save_excel(workbook, config, config_knn)

```

Listing 32: Kommentierte Version der Validierung von Modellkonfigurationen

C.6.4 Kommentierte Version zur Generierung eines OOS-Modells

```

def run_model(config, config_knn, worksheet, r):
    print("Generiere Ordner mit OOS und !OOS Daten")
    # Zusammenstellen der Trainings- und Testdaten
    copy_images_with_exclusion(config["bilder"]["original_path"], config["bilder"]["knn_path"],
        config_knn.excluded_folder)
    print("Initialisiere Callback")
    # Erstellen einer Instanz des benutzerdefinierten
    # Callbacks
    callback = CustomCallback()

    print("Starte Zeit")
    start_time = datetime.now()
    # Modell definieren
    print("Definiere KNN Modell")
    model = define_model(config_knn)
    # Augmentationsregeln definieren
    print("Initialisiere ImageDataGenerator")
    datagen = define_augmentation_rules()
    # Bestimmen der Trainings- und Testdaten
    print("Definiere Trainings- und Testdaten Set")
    train_it = get_train_data(datagen, config["bilder"]["knn_path"])
    test_it = get_test_data(datagen, config["bilder"]["knn_path"])
    print("Trainiere das Modell")
    # OOS-Modell trainieren
    fit_model(config, model, train_it, test_it, callback)
    # OOS-Modell testen
    print("Evaluere das Modell")
    evaluate_metrics = evaluate_model(model, test_it, datagen,
        , config, config_knn)

    # Dauer des Trainings und Tests bestimmen
    print("Stoppe die Zeit")
    end_time = datetime.now()
    training_duration_model = end_time - start_time

```

```
training_duration_model_str = format(  
    training_duration_model)  
print('Die Generierung des OOS-Modells dauerte: ' +  
    training_duration_model_str)  
# Dokumentieren der Ergebnisse  
print("Inhalte in die Excel-Tabelle schreiben")  
r = create_excel_result(worksheet, r, callback, config,  
    config_knn, evaluate_metrics, training_duration_model)  
return r, evaluate_metrics, training_duration_model
```

Listing 33: Kommentierte Version zur Generierung eines OOS-Modells

C.7 Generieren der Excel-Tabelle

C.7.1 Initiale Konfiguration einer Excel-Datei

```
def get_excel_workbook():
    # Erstellt eine neue Excel-Arbeitsmappe
    workbook = openpyxl.Workbook()
    return workbook

def get_excel_worksheet(workbook):
    # Erstellt ein neues Arbeitsblatt
    worksheet = workbook.active
    return worksheet
```

Listing 34: Erstellen einer Excel-Datei und Festlegen des Arbeitsblattes

C.7.2 Speichern der Excel-Tabelle

```
def save_excel(workbook, config, config_knn):
    now = datetime.now()
    date_time = now.strftime("%d_%m_%y_%H_%M")
    file_name = config_knn.optimization_method.name + "_" + \
                config_knn.activation_function_2_units.name + \
                "_" + \
                config_knn.activation_function_128_units.name \
                + "_" + date_time + ".xlsx"
    # Speichern Sie die Arbeitsmappe
    workbook.save(config["result"]["excel_path"] + file_name)
```

Listing 35: Speichern der Excel-Datei

C.7.3 Schreiben der Modellergebnisse in einer Excel-Datei

```

def create_excel_result(worksheet, r, callback, config,
    config_knn, evaluate_metrics, training_duration_model):
    # Spaltenbreite setzen
    worksheet.column_dimensions["A"].width = 10
    worksheet.column_dimensions["B"].width = 20
    worksheet.column_dimensions["C"].width = 20
    worksheet.column_dimensions["D"].width = 20
    worksheet.column_dimensions["E"].width = 20
    worksheet.column_dimensions["F"].width = 20
    worksheet.column_dimensions["G"].width = 20
    worksheet.column_dimensions["H"].width = 20
    worksheet.column_dimensions["I"].width = 20
    worksheet.column_dimensions["J"].width = 12

    # Modellkonfigurationen beschreiben
    worksheet.cell(row=r, column=2).value = '
        Optimierungsverfahren'
    worksheet.cell(row=r, column=3).value = '
        Aktivierungsfunktion 128'
    worksheet.cell(row=r, column=4).value = '
        Aktivierungsfunktion 2'
    worksheet.cell(row=r, column=5).value = 'Evaluationsmarkt
        '

    r = r + 1
    worksheet.cell(row=r, column=2).value = config_knn.
        optimization_method.name
    worksheet.cell(row=r, column=3).value = config_knn.
        activation_function_128_units.name
    worksheet.cell(row=r, column=4).value = config_knn.
        activation_function_2_units.name
    worksheet.cell(row=r, column=5).value = config_knn.
        excluded_folder.name
    r = r + 2

    # Titel der Epochenergebnisse definieren
    worksheet.cell(row=r, column=1).value = 'epoch'
    worksheet.cell(row=r, column=2).value = 'training_loss'
    worksheet.cell(row=r, column=3).value = '
        training_accuracy'
    worksheet.cell(row=r, column=4).value = '
        training_recall_oos'
    worksheet.cell(row=r, column=5).value = '
        training_recall_not_oos'
    worksheet.cell(row=r, column=6).value = 'test_loss'
    worksheet.cell(row=r, column=7).value = 'test_accuracy'
    worksheet.cell(row=r, column=8).value = 'test_recall_oos'

```

```

worksheet.cell(row=r, column=9).value = '
    test_recall_not_oos'
worksheet.cell(row=r, column=10).value = 'duration'
r = r + 1

# Dokumentieren der Trainingsergebnisse je Epoche
for i, (loss, accuracy, recall_oos, recall_not_oos,
       val_loss, val_accuracy, val_recall_oos,
       val_recall_not_oos, duration) in enumerate(
zip(callback.epoch_training_losses, callback.
epoch_training_accuracies,
callback.epoch_training_recall_oos, callback.
epoch_training_recall_not_oos,
callback.epoch_test_losses, callback.
epoch_test_accuracies, callback.
epoch_test_recall_oos,
callback.epoch_test_recall_not_oos, callback.
epoch_durations)):
worksheet.cell(row=r + i, column=1).value = i + 1
worksheet.cell(row=r + i, column=2).value = loss
worksheet.cell(row=r + i, column=3).value = accuracy
worksheet.cell(row=r + i, column=4).value =
    recall_oos
worksheet.cell(row=r + i, column=5).value =
    recall_not_oos
worksheet.cell(row=r + i, column=6).value = val_loss
worksheet.cell(row=r + i, column=7).value =
    val_accuracy
worksheet.cell(row=r + i, column=8).value =
    val_recall_oos
worksheet.cell(row=r + i, column=9).value =
    val_recall_not_oos
worksheet.cell(row=r + i, column=10).value = duration
r = r + config["knn"]["epochs"] + 1

# Dokumentieren der Testmarkt Ergebnisse
worksheet.cell(row=r, column=2).value = 'Trainingsdauer'
worksheet.cell(row=r, column=3).value =
    training_duration_model
r = r + 3
worksheet.cell(row=r, column=1).value = 'Evaluation'
worksheet.cell(row=r, column=2).value = 'loss'
worksheet.cell(row=r, column=3).value = 'accuracy'
worksheet.cell(row=r, column=4).value = 'recall_oos'
worksheet.cell(row=r, column=5).value = 'recall_not_oos'
r = r + 1
worksheet.cell(row=r, column=1).value = config_knn.
excluded_folder.name

```

```

worksheet.cell(row=r, column=2).value = evaluate_metrics
    [0]
worksheet.cell(row=r, column=3).value = evaluate_metrics
    [1]
worksheet.cell(row=r, column=4).value = evaluate_metrics
    [3]
worksheet.cell(row=r, column=5).value = evaluate_metrics
    [2]
r = r + 4
return r

```

Listing 36: Schreiben der Modellergebnisse in einer Excel-Tabelle

C.7.4 Mittelwert der Modellergebnisse berechnen

```

def sum_values_into_one_array(model_evaluate_metrics ,
    training_duration_models):
    # Initialisiere Summen-Array mit Nullen
    sum_array = [0, 0, 0, 0, 0]
    # Bestimme die Anzahl der Modelle
    num_arrays = len(model_evaluate_metrics)

    # Iteriere durch das aeussere Array
    for evaluate_metric in model_evaluate_metrics:
        # Fuege die Werte des inneren Arrays zum Summen-Array
        hinzu
        for i in range(len(evaluate_metric)):
            sum_array[i] += evaluate_metric[i]

    # Berechne den Durchschnitt, indem das Summen-Array durch
    die Anzahl der inneren Arrays geteilt wird
    avg_array = [value / num_arrays for value in sum_array]

    # Zeiten aufaddieren
    add_times = timedelta()
    for training_time in training_duration_models:
        add_times += training_time
    # Mittelwert berechnen
    average_duration = add_times / len(
        training_duration_models)
    # Ergebnis dem avg_array hinzufuegen
    avg_array[4] = average_duration
    return avg_array

```

Listing 37: Berechnung des Mittelwerts der OOS-Modellergebnisse

C.7.5 Mittelwert der Modellergebnisse dokumentieren

```
def average_evaluate_cross_validation(worksheet, r,
    model_evaluate_metrics, training_duration_models):
    total_information = sum_values_into_one_array(
        model_evaluate_metrics, training_duration_models)
    worksheet.cell(row=r, column=2).value = 'AVERAGE'
    worksheet.cell(row=r, column=3).value = 'EVALUATE'
    worksheet.cell(row=r, column=4).value = 'METRICS'
    r = r + 1
    worksheet.cell(row=r, column=1).value = 'Models'
    worksheet.cell(row=r, column=2).value = 'loss'
    worksheet.cell(row=r, column=3).value = 'accuracy'
    worksheet.cell(row=r, column=4).value = 'recall_oos'
    worksheet.cell(row=r, column=5).value = 'recall_not_oos'
    worksheet.cell(row=r, column=6).value = 'duration'
    r = r + 1
    worksheet.cell(row=r, column=1).value = len(
        training_duration_models)
    worksheet.cell(row=r, column=2).value = total_information
        [0]
    worksheet.cell(row=r, column=3).value = total_information
        [1]
    worksheet.cell(row=r, column=4).value = total_information
        [3]
    worksheet.cell(row=r, column=5).value = total_information
        [2]
    # Durchschnittswert wieder in Datetime-Objekt umwandeln
    average_datetime = format(total_information[4])
    worksheet.cell(row=r, column=6).value = average_datetime
```

Listing 38: Dokumentation der Mittelweltergebnisse in einer Excel-Tabelle

D Ergebnisse

Die Dokumentation der Ergebnisse teilt sich in Modellkonfigurationen auf, die die unveränderte oder die augmentierte Datenbasis verwendet haben. Nachfolgend listet Kapitel D.1 die Ergebnisse zu den 36 generierten Modellkonfigurationen auf Basis der unveränderten Daten auf. Die Einstellungen der sechs besten Modellkonfigurationen wurden zum Generieren neuer OOS-Modelle auf Basis der augmentierten Daten wiederverwendet. Die dazu erzielten Ergebnisse sind in Kapitel D.2 dargestellt.

D.1 Modellkonfigurationen auf Basis der unveränderten Daten

Im nachfolgenden werden die Ergebnisse der OOS-Modelle zu jeder Modellkonfiguration dargestellt. Aufgrund der Generierung von sechs OOS-Modellen pro Modellkonfiguration wurden entsprechend viele Tabellen erstellt, um die Ergebnisse während der Trainings- und Testphase zu präsentieren.

Die Trainings- und Testergebnisse jeder Epoche umfassen die Resultate der Kostenfunktion BCE, der *Accuracy*, den *Recall* der Klasse OOS und In-Stock, sowie die Dauer der Epoche. Die Ergebnisse werden in Dezimalform angegeben und müssen mit 100 multipliziert werden, um den Prozentsatz zu erhalten. Am Ende jeder Tabelle ist dargestellt, welcher Testmarkt das generierte OOS-Modell getestet hat. Die zugehörigen Ergebnisse sind in der letzten Zeile der jeweiligen Tabellen dargestellt. Die letzte Tabelle zu den jeweiligen Modellkonfiguration unterscheidet sich von den anderen Tabellen. Sie stellt die durchschnittlichen Ergebnisse dar, die die Testmärkte erzielten.

D.1.1 Adam sigmoid ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.728452	0.766868	0.733485	0.738624	0.281327	0.894737	0.909333	0.888614	642.22
2	0.177237	0.937466	0.935080	0.938624	0.243387	0.902439	0.858667	0.945545	638.29
3	0.156426	0.936369	0.933941	0.938624	0.333629	0.861361	0.970667	0.759901	639.09
4	0.132281	0.948985	0.947608	0.953439	0.214599	0.903723	0.909333	0.891089	637.61
5	0.128745	0.953374	0.952164	0.953439	0.256385	0.894737	0.842667	0.945545	637.89
6	0.115523	0.957762	0.954442	0.957672	0.243759	0.899872	0.898667	0.910891	635.81
7	0.103964	0.959408	0.962415	0.955556	0.526773	0.794608	0.584000	0.987624	639.99
8	0.142737	0.941306	0.935080	0.946032	0.228513	0.928113	0.896000	0.957921	637.51
9	0.098808	0.963796	0.961276	0.967196	0.209017	0.920411	0.914667	0.928218	637.68
10	0.097880	0.962699	0.958998	0.964021	0.243522	0.898588	0.880000	0.925743	635.48
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:02	0.093093		0.963636		1		0.947368		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.003827	0.686299	0.687259	0.662679	0.347734	0.852388	0.792793	0.905028	570.17
2	0.229821	0.921885	0.920206	0.922249	0.304229	0.876990	0.852853	0.896648	566.98
3	0.154719	0.953503	0.953668	0.947368	0.275742	0.888567	0.846847	0.927374	568.68
4	0.137390	0.955983	0.961390	0.953349	0.250332	0.910275	0.921922	0.899441	567.09
5	0.121313	0.954743	0.952381	0.954545	0.213083	0.918958	0.867868	0.966480	567.92
6	0.119101	0.960942	0.954955	0.967703	0.276995	0.891462	0.933934	0.863128	568.17
7	0.117412	0.955983	0.963964	0.950957	0.234107	0.905933	0.909910	0.899441	566.71
8	0.098029	0.967142	0.963964	0.971292	0.229890	0.898698	0.942943	0.860335	568.17
9	0.089524	0.969622	0.969112	0.971292	0.218254	0.914617	0.897898	0.927374	567.42
10	0.107141	0.956603	0.958816	0.955742	0.277056	0.875543	0.834835	0.918994	565.59
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:26	0.193899		0.916667		0.881356		0.943723		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.710188	0.809350	0.790143	0.781081	0.256045	0.895727	0.955390	0.908228	485.22
2	0.167996	0.931337	0.939587	0.939189	0.234790	0.904274	0.888476	0.917722	482.51
3	0.147743	0.943024	0.950715	0.935135	0.241299	0.900855	0.951673	0.860759	481.52
4	0.218910	0.911614	0.899841	0.918919	0.364479	0.858120	0.944238	0.781646	482.01
5	0.196069	0.930606	0.928458	0.935135	0.274468	0.876923	0.780669	0.955696	481.13
6	0.176327	0.928415	0.934817	0.922973	0.374939	0.868376	0.966543	0.781646	479.70
7	0.165827	0.934259	0.936407	0.931081	0.275743	0.890598	0.817844	0.949367	481.64
8	0.117897	0.956172	0.953895	0.956757	0.332129	0.871795	0.955390	0.800633	480.58
9	0.115887	0.956172	0.955485	0.956757	0.326352	0.892308	0.933086	0.857595	479.90
10	0.142195	0.945215	0.931638	0.955405	0.199737	0.919658	0.921933	0.914557	481.00
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:24:11	0.155919		0.941953		0.922879		0.962060		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.103875	0.520967	0.278443	0.757781	0.690413	0.524917	0.937063	1	501.00
2	0.665897	0.525231	0.459581	0.740189	0.648690	0.524917	0.006993	0.971519	494.00
3	0.578196	0.624023	0.318862	0.991881	0.514626	0.674419	0.297203	1	497.39
4	0.475050	0.820895	0.688623	0.985115	0.445400	0.883721	0.776224	0.974684	495.10
5	0.439938	0.875622	0.772455	0.979702	0.404226	0.913621	0.849650	0.968354	496.78
6	0.287734	0.915423	0.898204	0.907984	0.219801	0.935216	0.891608	0.962025	495.21
7	0.192975	0.938877	0.929641	0.940460	0.215125	0.908638	0.790210	0.990506	493.19
8	0.173853	0.938877	0.914671	0.947226	0.162951	0.951827	0.902098	0.955696	494.66
9	0.158114	0.942431	0.940120	0.945873	0.168863	0.933555	0.898601	0.952532	496.61
10	0.139781	0.945274	0.947605	0.941813	0.153702	0.938538	0.930070	0.952532	495.40
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:39	0.165600		0.938834		0.969970		0.900000		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.699972	0.852226	0.806574	0.828859	0.272878	0.886269	0.893491	0.877285	595.28
2	0.186539	0.926409	0.910240	0.929530	0.187657	0.933426	0.940828	0.932115	592.20
3	0.188640	0.922849	0.911504	0.929530	0.194614	0.918169	0.920118	0.913838	591.30
4	0.119630	0.957864	0.958281	0.958613	0.205612	0.911234	0.943787	0.887728	591.03
5	0.101247	0.959050	0.951960	0.965324	0.195515	0.922330	0.908284	0.932115	591.75
6	0.119556	0.949555	0.945638	0.954139	0.195100	0.918169	0.952663	0.882507	589.81
7	0.114851	0.952522	0.949431	0.953020	0.296920	0.876560	0.982249	0.780679	590.30
8	0.132828	0.949555	0.948167	0.950783	0.220723	0.920943	0.961538	0.885117	592.02
9	0.107256	0.957270	0.950695	0.963087	0.200293	0.916782	0.863905	0.963446	590.58
10	0.132699	0.948961	0.944374	0.951902	0.197296	0.925104	0.914201	0.937337	589.72
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:42:01	0.177493		0.928105		0.892405		0.966216		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.559358	0.836250	0.832461	0.755981	0.266995	0.884672	0.883792	0.927374	566.00
2	0.180783	0.936250	0.917539	0.935407	0.201600	0.922628	0.938838	0.916201	561.20
3	0.147278	0.943125	0.942408	0.950957	0.215034	0.912409	0.828746	0.927374	562.89
4	0.139082	0.945000	0.933246	0.954545	0.188865	0.925547	0.929664	0.958101	558.70
5	0.129218	0.954375	0.947644	0.953349	0.214914	0.913869	0.877676	0.944134	559.77
6	0.103873	0.958125	0.952880	0.965311	0.189167	0.916788	0.905199	0.927374	560.59
7	0.122005	0.951875	0.945026	0.958134	0.195155	0.924088	0.917431	0.938547	561.90
8	0.107037	0.958125	0.947644	0.964115	0.216875	0.915328	0.896024	0.918994	561.71
9	0.096626	0.965000	0.963351	0.967703	0.252977	0.900730	0.951070	0.868715	561.88
10	0.106913	0.956875	0.959424	0.960526	0.211929	0.915328	0.871560	0.955307	561.48
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:30	0.194040		0.922717		0.846939		0.987013		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:08	0.163341	0.935318	0.918925	0.951063

D.1.2 Adam sigmoid sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.423673	0.851344	0.838269	0.840212	0.279304	0.890886	0.896000	0.873762	641.91
2	0.204098	0.924301	0.930524	0.918519	0.236174	0.920411	0.906667	0.925743	636.83
3	0.162981	0.941854	0.936219	0.947090	0.267805	0.894737	0.882667	0.905941	639.28
4	0.141054	0.948985	0.952164	0.948148	0.217267	0.920411	0.909333	0.933168	636.71
5	0.176047	0.933077	0.935080	0.934392	0.256201	0.901155	0.968000	0.814356	636.98
6	0.135490	0.950631	0.946469	0.950265	0.216085	0.913992	0.954667	0.886139	635.20
7	0.126196	0.953374	0.957859	0.949206	0.210004	0.910141	0.898667	0.925743	634.87
8	0.129000	0.952276	0.956720	0.948148	0.204336	0.917843	0.914667	0.920792	634.68
9	0.121608	0.953922	0.954442	0.954497	0.202274	0.928113	0.920000	0.935644	635.31
10	0.123494	0.952276	0.948747	0.951323	0.193313	0.925546	0.938667	0.913366	636.69
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:36	0.090886		0.963636		1		0.947368		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.531326	0.836950	0.776062	0.854067	0.373202	0.863965	0.843844	0.868715	571.37
2	0.260168	0.919405	0.913771	0.927033	0.327256	0.863965	0.900901	0.843575	566.59
3	0.200876	0.940484	0.934363	0.946172	0.302341	0.887120	0.807808	0.958101	567.51
4	0.153641	0.959082	0.956242	0.959330	0.268368	0.888567	0.918919	0.851955	565.58
5	0.152579	0.946683	0.956242	0.938995	0.280472	0.890014	0.849850	0.921788	564.68
6	0.122603	0.964042	0.956242	0.970096	0.261009	0.895803	0.933934	0.868715	565.90
7	0.115884	0.967142	0.962677	0.970096	0.235175	0.905933	0.906907	0.905028	566.40
8	0.119939	0.961562	0.962677	0.962919	0.270778	0.885673	0.819820	0.938547	565.89
9	0.128266	0.955983	0.956242	0.952153	0.268153	0.894356	0.942943	0.854749	565.48
10	0.104362	0.967142	0.971686	0.961722	0.281810	0.882779	0.942943	0.818436	564.59
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:12	0.289139		0.892157		0.988701		0.813853		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.571264	0.790358	0.682035	0.798649	0.392332	0.858120	0.877323	0.870253	486.27
2	0.263649	0.932798	0.917329	0.937838	0.300322	0.895727	0.881041	0.911392	483.10
3	0.217402	0.931337	0.928458	0.931081	0.284143	0.894017	0.910781	0.876582	482.66
4	0.175686	0.946676	0.939587	0.950000	0.248046	0.899145	0.918216	0.886076	482.20
5	0.200237	0.932798	0.923688	0.937838	0.246685	0.897436	0.836431	0.943038	477.60
6	0.154266	0.948137	0.933227	0.962162	0.232350	0.899145	0.884758	0.924051	481.83
7	0.144557	0.950329	0.957075	0.944595	0.250509	0.894017	0.951673	0.835443	482.29
8	0.128716	0.959094	0.953895	0.960811	0.318189	0.870085	0.947955	0.803797	481.18
9	0.128319	0.953251	0.953895	0.950000	0.243853	0.895727	0.951673	0.844937	479.90
10	0.123364	0.959094	0.953895	0.963513	0.266026	0.894017	0.925651	0.863924	480.79
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:24:15	0.158119		0.932718		0.948586		0.915989		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.717052	0.792466	0.881737	0.883627	0.352030	0.921927	0.856643	0.927215	501.57
2	0.336128	0.915423	0.890719	0.895805	0.320148	0.908638	0.807692	0.965190	496.89
3	0.278004	0.928216	0.919162	0.926928	0.248476	0.936877	0.916084	0.943038	496.00
4	0.236099	0.938166	0.931138	0.943166	0.226999	0.948505	0.923077	0.981013	493.90
5	0.214454	0.942431	0.937126	0.951286	0.207121	0.948505	0.951049	0.949367	495.44
6	0.181846	0.956645	0.956587	0.958051	0.186383	0.953488	0.933566	0.971519	495.98
7	0.177126	0.950960	0.952096	0.949932	0.276805	0.895349	0.783217	0.993671	497.11
8	0.174668	0.948827	0.946108	0.951286	0.185045	0.930233	0.923077	0.939873	494.51
9	0.163092	0.945274	0.932635	0.959405	0.190536	0.933555	0.895105	0.968354	493.48
10	0.155027	0.949538	0.946108	0.951286	0.155243	0.953488	0.947552	0.958861	496.37
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:53	0.172076		0.941679		0.972973		0.913514		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.380871	0.522255	0.031606	0.476510	1.159672	0.531207	0	0	596.53
2	0.988213	0.530564	0	0.717002	0.840273	0.531207	0	0.887728	593.04
3	0.727308	0.530564	0	0.955257	0.637509	0.531207	0	0.929504	591.28
4	0.593605	0.744807	0	0.957494	0.580744	0.900139	0	0.879896	588.73
5	0.554811	0.931751	0.093552	0.949664	0.559294	0.887656	0.585799	0.950392	593.88
6	0.524692	0.945994	0.820480	0.957494	0.531579	0.915395	0.878698	0.851175	592.59
7	0.497291	0.950742	0.891277	0.957494	0.509336	0.911234	0.905325	0.872063	594.49
8	0.471957	0.950148	0.936789	0.948546	0.466532	0.922330	0.911243	0.911227	589.81
9	0.435183	0.946588	0.919090	0.963087	0.438375	0.919556	0.934911	0.887728	591.38
10	0.407389	0.945994	0.936789	0.947427	0.421359	0.916782	0.928994	0.882507	593.81
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:06	0.415137		0.918301		0.886076		0.939189		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.724701	0.763750	0.315445	0.547847	0.427378	0.830657	0.883792	0.734637	566.34
2	0.311261	0.907500	0.916230	0.899522	0.333786	0.881752	0.941896	0.832402	561.61
3	0.241770	0.936250	0.934555	0.942584	0.270858	0.916788	0.957187	0.891061	562.41
4	0.210019	0.942500	0.929319	0.955742	0.221369	0.929927	0.914373	0.932961	562.69
5	0.173707	0.947500	0.943717	0.949761	0.255755	0.903650	0.850153	0.949721	564.69
6	0.163526	0.950625	0.943717	0.955742	0.236488	0.908029	0.948012	0.871508	562.02
7	0.163141	0.947500	0.946335	0.946172	0.248050	0.906569	0.905199	0.902235	560.29
8	0.144968	0.952500	0.954188	0.952153	0.204626	0.931387	0.917431	0.944134	564.09
9	0.135240	0.955000	0.952880	0.956938	0.230443	0.912409	0.926606	0.896648	563.87
10	0.122355	0.959375	0.954188	0.960526	0.191820	0.928467	0.917431	0.935754	563.31
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:45	0.163830		0.950820		0.918367		0.978355		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:08	0.214865	0.933218	0.952450	0.918045

D.1.3 Adam sigmoid softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.670791	0.659901	0.413440	0.937566	0.655414	0.829268	0.842667	1	639.63
2	0.642836	0.909490	0.907745	1	0.641593	0.856226	0.802667	1	634.06
3	0.628622	0.927592	0.920273	1	0.629206	0.878049	0.858667	0.995049	634.09
4	0.613798	0.940757	0.936219	0.984127	0.615818	0.901155	0.917333	0.950495	635.50
5	0.603562	0.927592	0.917995	0.969312	0.616684	0.860077	0.968000	0.787129	633.60
6	0.590235	0.936917	0.945330	0.946032	0.595148	0.893453	0.877333	0.933168	632.69
7	0.572396	0.959408	0.962415	0.964021	0.585791	0.893453	0.840000	0.960396	635.00
8	0.562822	0.954471	0.951025	0.967196	0.575882	0.889602	0.832000	0.957921	634.88
9	0.549562	0.958859	0.954442	0.969312	0.560311	0.912709	0.912000	0.920792	631.57
10	0.538897	0.962150	0.953303	0.975661	0.549049	0.926829	0.898667	0.955446	633.42
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:16	0.513821		1		1		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.648695	0.818351	0.733591	0.891148	0.643779	0.821997	0.972973	0.776536	568.65
2	0.618068	0.919405	0.934363	0.949761	0.619990	0.890014	0.888889	0.916201	561.30
3	0.600340	0.951023	0.958816	0.964115	0.605667	0.901592	0.891892	0.946927	562.87
4	0.586667	0.958462	0.956242	0.968900	0.598023	0.894356	0.867868	0.932961	564.10
5	0.575967	0.956603	0.958816	0.965311	0.598458	0.848046	0.732733	0.977654	563.51
6	0.566704	0.952883	0.958816	0.955742	0.581403	0.900145	0.936937	0.874302	563.22
7	0.553116	0.961562	0.945946	0.982057	0.570384	0.894356	0.882883	0.916201	564.60
8	0.543312	0.960322	0.958816	0.962919	0.560686	0.898698	0.936937	0.874302	564.48
9	0.532122	0.963422	0.963964	0.967703	0.547709	0.916064	0.912913	0.924581	565.02
10	0.521988	0.965282	0.962677	0.967703	0.546312	0.897250	0.957958	0.840782	562.99
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:38:41	0.546024		0.901961		0.977401		0.848485		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.662684	0.705625	0.550079	0.879730	0.638066	0.846154	0.739777	0.930380	480.93
2	0.627535	0.869248	0.841017	0.889189	0.627199	0.861538	0.973978	0.762658	478.30
3	0.610481	0.906501	0.879173	0.929730	0.607260	0.895727	0.892193	0.895570	477.13
4	0.592771	0.934259	0.923688	0.943243	0.600425	0.895727	0.940520	0.857595	479.64
5	0.586663	0.922571	0.923688	0.921622	0.609266	0.835897	0.985130	0.708861	477.87
6	0.573880	0.937180	0.941176	0.933784	0.579462	0.905983	0.866171	0.936709	478.29
7	0.561732	0.946676	0.931638	0.959459	0.573642	0.895727	0.862454	0.924051	480.00
8	0.550357	0.951790	0.941176	0.960811	0.562766	0.907692	0.888476	0.924051	479.49
9	0.543943	0.944485	0.930048	0.956757	0.556172	0.905983	0.884758	0.924051	477.68
10	0.535140	0.950329	0.931638	0.966216	0.553483	0.897436	0.955390	0.848101	478.09
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:25	0.538534		0.947230		0.974293		0.918699		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.658723	0.731343	0.657186	0.860622	0.625849	0.901993	0.912587	0.870253	497.38
2	0.620033	0.895522	0.860778	0.913396	0.610125	0.913621	0.825175	0.981013	492.71
3	0.607942	0.909026	0.892216	0.912043	0.595263	0.930233	0.905594	0.943038	491.69
4	0.597268	0.901208	0.884731	0.913396	0.584688	0.935216	0.958042	0.914557	495.09
5	0.581725	0.928927	0.905689	0.945873	0.577247	0.926910	0.912587	0.939873	493.00
6	0.567919	0.943852	0.926647	0.955345	0.567418	0.928571	0.919580	0.933544	492.89
7	0.556225	0.952381	0.946108	0.955345	0.556338	0.938538	0.895105	0.977848	493.68
8	0.548668	0.949538	0.937126	0.959405	0.547683	0.935216	0.905594	0.958861	543.66
9	0.542885	0.947406	0.938623	0.951286	0.534481	0.951827	0.947552	0.955696	493.52
10	0.542988	0.918977	0.904192	0.932341	0.564285	0.853821	0.699301	0.993671	490.50
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:59	0.553458		0.881935		0.750751		0.997297		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.641177	0.829674	0.841972	0.851230	0.626088	0.875173	0.801775	0.934726	590.97
2	0.608181	0.931157	0.912769	0.942953	0.613496	0.879334	0.789941	0.953003	586.77
3	0.592827	0.946588	0.916561	0.966443	0.595751	0.904300	0.869823	0.932115	589.50
4	0.579075	0.949555	0.931732	0.964206	0.587406	0.908460	0.967456	0.851175	588.62
5	0.575787	0.923442	0.929204	0.913870	0.571027	0.920943	0.946746	0.898172	587.43
6	0.557824	0.947774	0.932996	0.959732	0.558133	0.929265	0.940828	0.916449	591.23
7	0.546580	0.952522	0.944374	0.957494	0.552403	0.922330	0.917160	0.926893	587.40
8	0.534940	0.954303	0.948167	0.959732	0.539915	0.929265	0.905325	0.947781	586.98
9	0.522202	0.960237	0.951960	0.967562	0.532132	0.932039	0.917160	0.939948	587.48
10	0.517062	0.951929	0.953224	0.950783	0.521347	0.938974	0.940828	0.934726	590.52
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:36	0.526740		0.918301		0.905063		0.932432		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.661802	0.741875	0.596859	0.973684	0.646116	0.832117	0.733945	0.994413	560.13
2	0.631057	0.920625	0.907068	0.972488	0.631571	0.881752	0.880734	0.930168	559.08
3	0.616051	0.938125	0.935864	0.965311	0.623445	0.891971	0.966361	0.832402	561.15
4	0.604235	0.945000	0.934555	0.971292	0.609306	0.897810	0.886850	0.927374	559.20
5	0.592530	0.947500	0.937173	0.964115	0.598433	0.902190	0.837920	0.972067	560.12
6	0.581041	0.956875	0.946335	0.973684	0.588820	0.909489	0.844037	0.980447	592.74
7	0.573949	0.941250	0.930628	0.956938	0.578370	0.913869	0.954128	0.891061	559.41
8	0.563553	0.945000	0.938482	0.954545	0.569147	0.912409	0.886850	0.938547	558.12
9	0.551701	0.953750	0.945026	0.966507	0.559041	0.924088	0.960245	0.899441	556.90
10	0.543577	0.953125	0.950262	0.956938	0.551358	0.925547	0.914373	0.935754	559.01
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:40	0.543116		0.939110		0.908163		0.969697		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:16	0.536949	0.931423	0.919279	0.944435

D.1.4 Adam sigmoid tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.254601	0.837082	0.968109	0.972487	1.315300	0.862644	1	1	631.66
2	0.562152	0.936917	0.997722	0.988360	0.431031	0.870347	0.970667	0.918317	629.98
3	0.229224	0.932529	0.965831	0.966138	0.312206	0.890886	0.896000	0.943069	630.81
4	0.167481	0.944048	0.963554	0.958730	0.221586	0.911425	0.933333	0.915842	632.69
5	0.137992	0.950082	0.958998	0.961905	0.232033	0.908858	0.928000	0.913366	630.19
6	0.140723	0.947340	0.954442	0.947090	0.236330	0.913992	0.896000	0.943069	631.23
7	0.109084	0.960505	0.953303	0.970370	0.220960	0.910141	0.848000	0.960396	632.70
8	0.141373	0.944597	0.944191	0.951323	0.238522	0.897304	0.834667	0.965347	633.69
9	0.132864	0.952276	0.955581	0.949206	0.200650	0.915276	0.925333	0.920792	631.67
10	0.111117	0.958310	0.954442	0.955556	0.230040	0.898588	0.938667	0.858911	633.33
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:45	0.156606		0.927273		1		0.894737		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.617366	0.773094	0.957529	0.965311	1.652729	0.836469	1	1	572.41
2	0.907031	0.926844	1	0.997608	0.555737	0.891462	0.975976	0.974860	565.07
3	0.255886	0.952263	0.983269	0.984450	0.306479	0.895803	0.975976	0.910615	564.79
4	0.153921	0.955983	0.967825	0.972488	0.278586	0.904486	0.918919	0.924581	564.29
5	0.141934	0.951643	0.957529	0.970096	0.243907	0.892909	0.936937	0.891061	564.80
6	0.113208	0.962182	0.965251	0.964115	0.235437	0.916064	0.909910	0.932961	561.78
7	0.112015	0.960322	0.958816	0.968900	0.280405	0.887120	0.834835	0.952514	563.91
8	0.131451	0.952883	0.951094	0.961722	0.234668	0.904486	0.921922	0.879888	564.26
9	0.119803	0.961562	0.958816	0.965311	0.240249	0.907381	0.924925	0.891061	564.91
10	0.115708	0.958462	0.962677	0.958134	0.240889	0.895803	0.858859	0.932961	560.31
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:49	0.200705		0.919118		0.881356		0.943723		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.790297	0.459459	0.341812	0.043243	2.232767	0.459829	0.531599	0	481.11
2	1.722944	0.459459	0.761526	0	1.352184	0.459829	0.747212	0	477.11
3	0.904018	0.632579	0.866455	0	0.627178	0.854701	0.795539	0	476.38
4	0.345092	0.943755	0.912560	0.454054	0.334327	0.883761	0.921933	0.787975	477.89
5	0.215472	0.934989	0.912560	0.927027	0.265764	0.900855	0.940520	0.876582	478.18
6	0.181690	0.942294	0.928458	0.945946	0.229196	0.914530	0.884758	0.939873	479.49
7	0.148619	0.956172	0.947536	0.958108	0.291971	0.882051	0.788104	0.955696	478.12
8	0.227596	0.922571	0.918919	0.925676	0.264533	0.914530	0.903346	0.939873	478.98
9	0.163824	0.951059	0.941176	0.959459	0.243047	0.912821	0.947955	0.886076	478.54
10	0.145515	0.955442	0.960254	0.951351	0.272334	0.892308	0.840149	0.930380	478.44
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:18	0.177169		0.944591		0.920308		0.970190		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.545657	0.472637	0.950599	0.040595	2.855402	0.475083	1	0	496.38
2	2.262247	0.474769	1	0	1.715926	0.475083	1	0	493.39
3	1.371527	0.474769	0.217066	0	1.018408	0.475083	0	0	493.11
4	0.819191	0.474769	0	0	0.697213	0.475083	0	0	493.47
5	0.694590	0.516702	0	0.825440	0.694915	0.524917	0	1	492.72
6	0.693075	0.525231	0	1	0.691920	0.524917	0	1	491.47
7	0.692847	0.525231	0	1	0.691989	0.524917	0	1	493.01
8	0.692909	0.525231	0	1	0.691982	0.524917	0	1	492.69
9	0.692359	0.525231	0	1	0.692035	0.524917	0	1	491.71
10	0.693480	0.525231	0	1	0.691918	0.524917	0	1	492.59
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:25	0.691767		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.299202	0.465282	0.957016	0.972036	1.563576	0.468793	1	0.953003	592.32
2	1.052885	0.507418	1	0.953020	0.624808	0.851595	1	0.971279	586.70
3	0.409191	0.929970	0.975980	0.956376	0.348755	0.907073	0.923077	0.942559	588.01
4	0.302224	0.928190	0.907712	0.944072	0.310498	0.907073	0.863905	0.939948	587.44
5	0.263186	0.942433	0.934260	0.945190	0.288115	0.916782	0.905325	0.916449	589.58
6	0.245688	0.938279	0.936789	0.939597	0.286874	0.895978	0.831361	0.963446	587.72
7	0.199512	0.953709	0.949431	0.963087	0.266293	0.914008	0.943787	0.879896	587.59
8	0.196980	0.945401	0.941846	0.946309	0.307036	0.884882	0.789941	0.958225	588.59
9	0.160717	0.960831	0.954488	0.965324	0.268923	0.901526	0.843195	0.955614	589.87
10	0.175139	0.949555	0.946903	0.955257	0.245624	0.914008	0.934911	0.895561	589.82
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:40	0.192324		0.947712		0.936709		0.959459		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.553239	0.522500	0.942408	0.972488	0.856283	0.522628	0.853211	1	563.14
2	0.558626	0.668125	0.925393	1	0.396663	0.889051	0.850153	0.963687	558.77
3	0.297272	0.923125	0.928010	0.935407	0.306359	0.900730	0.856269	0.913408	560.63
4	0.245657	0.937500	0.921466	0.950957	0.268266	0.916788	0.920489	0.910615	560.19
5	0.214845	0.949375	0.946335	0.953349	0.243520	0.915328	0.914373	0.918994	560.38
6	0.196864	0.951875	0.946335	0.960526	0.291337	0.894890	0.969419	0.829609	559.89
7	0.228725	0.926250	0.922775	0.930622	0.295845	0.880292	0.966361	0.804469	558.00
8	0.177317	0.954375	0.948953	0.958134	0.237997	0.924088	0.926606	0.927374	560.30
9	0.159209	0.958125	0.959424	0.962919	0.208734	0.934307	0.954128	0.910615	559.14
10	0.152461	0.958125	0.952880	0.960526	0.236051	0.908029	0.877676	0.927374	560.59
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:14	0.213349		0.929742		0.841837		0.991342		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:52	0.271987	0.865792	0.763368	0.959908

D.1.5 Adam softmax ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.599875	0.797586	0.771071	0.822222	0.299750	0.878049	0.840000	0.913366	637.50
2	0.186310	0.923204	0.924829	0.921693	0.213223	0.910141	0.933333	0.888614	633.99
3	0.143576	0.940208	0.937358	0.942857	0.222536	0.908858	0.901333	0.915842	633.41
4	0.124929	0.956116	0.955581	0.956614	0.250967	0.897304	0.856000	0.935644	630.61
5	0.130049	0.947340	0.946469	0.948148	0.242732	0.889602	0.954667	0.829208	633.80
6	0.145416	0.938563	0.937358	0.939683	0.165249	0.935815	0.933333	0.938119	634.10
7	0.105264	0.959956	0.954442	0.965079	0.229790	0.912709	0.954667	0.873762	632.41
8	0.107579	0.962150	0.961276	0.962963	0.277586	0.906290	0.826667	0.980198	631.32
9	0.112615	0.956665	0.954442	0.958730	0.236211	0.902439	0.821333	0.977723	634.80
10	0.115262	0.955019	0.951025	0.958730	0.196062	0.920411	0.906667	0.933168	632.09
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:53	0.123573		0.954545		1		0.934211		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.675154	0.774953	0.779923	0.770335	0.459270	0.861071	0.831832	0.888268	566.90
2	0.224426	0.939864	0.936937	0.942584	0.294671	0.884226	0.921922	0.849162	564.27
3	0.154827	0.942963	0.935650	0.949761	0.263193	0.895803	0.900901	0.891061	565.31
4	0.146344	0.941104	0.940798	0.941388	0.299276	0.892909	0.855856	0.927374	563.30
5	0.137083	0.943583	0.942085	0.944976	0.312433	0.885673	0.804805	0.960894	564.61
6	0.112600	0.960322	0.960103	0.960526	0.253029	0.908828	0.882883	0.932961	565.69
7	0.092530	0.966522	0.962677	0.970096	0.315731	0.875543	0.927928	0.826816	564.51
8	0.086916	0.968382	0.963964	0.972488	0.290988	0.885673	0.846847	0.921788	563.29
9	0.099773	0.959702	0.963964	0.955742	0.286948	0.894356	0.840841	0.944134	561.09
10	0.102881	0.961562	0.956242	0.966507	0.239306	0.898698	0.966967	0.835196	566.31
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:50	0.256759		0.901961		0.949153		0.865801		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.959685	0.692476	0.661367	0.718919	0.412760	0.827350	0.676580	0.955696	482.93
2	0.263491	0.902118	0.880763	0.920270	0.334536	0.847863	0.951673	0.759494	478.49
3	0.202740	0.931337	0.920509	0.940541	0.264297	0.897436	0.858736	0.930380	480.79
4	0.186948	0.926224	0.910970	0.939189	0.249754	0.904274	0.895911	0.911392	478.17
5	0.168503	0.934989	0.937997	0.932432	0.234720	0.911111	0.955390	0.873418	480.60
6	0.145265	0.942294	0.934817	0.948649	0.220921	0.916239	0.977695	0.863924	481.11
7	0.134036	0.951790	0.947536	0.955405	0.227887	0.897436	0.847584	0.939873	477.91
8	0.126446	0.954711	0.960254	0.950000	0.231061	0.911111	0.936803	0.889241	480.80
9	0.133435	0.955442	0.952305	0.958108	0.246006	0.894017	0.869888	0.914557	480.09
10	0.127876	0.948137	0.945946	0.950000	0.250563	0.887179	0.780669	0.977848	481.30
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:36	0.217343		0.919525		0.848329		0.994580		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.765548	0.729922	0.763473	0.699594	0.308922	0.908638	0.944056	0.876582	497.23
2	0.268331	0.891258	0.899701	0.883627	0.183102	0.933555	0.888112	0.974684	493.32
3	0.161796	0.938166	0.941617	0.935047	0.190052	0.925249	0.870629	0.974684	493.18
4	0.174364	0.933191	0.937126	0.929635	0.187275	0.935216	0.975524	0.898734	493.49
5	0.169547	0.933191	0.934132	0.932341	0.233482	0.895349	0.800699	0.981013	493.79
6	0.162684	0.934613	0.934132	0.935047	0.195700	0.945183	0.923077	0.965190	495.33
7	0.144234	0.948117	0.953593	0.943166	0.179489	0.933555	0.888112	0.974684	493.96
8	0.107848	0.963042	0.961078	0.964817	0.162818	0.926910	0.944056	0.911392	495.72
9	0.144707	0.945984	0.946108	0.945873	0.161324	0.948505	0.940559	0.955696	492.12
10	0.118653	0.960199	0.959581	0.960758	0.134243	0.943522	0.944056	0.943038	490.92
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:33	0.157186		0.930299		0.951952		0.910811		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.817936	0.751929	0.729456	0.771812	0.308449	0.894591	0.920118	0.872063	595.87
2	0.198632	0.927596	0.916561	0.937360	0.259652	0.900139	0.908284	0.892950	589.40
3	0.146723	0.951929	0.945638	0.957494	0.226212	0.898752	0.931953	0.869452	590.31
4	0.144671	0.945401	0.944374	0.946309	0.212579	0.909847	0.872781	0.942559	586.09
5	0.124229	0.953116	0.950695	0.955257	0.208643	0.908460	0.881657	0.932115	589.53
6	0.128690	0.948368	0.946903	0.949664	0.199428	0.932039	0.967456	0.900783	589.46
7	0.130295	0.954303	0.949431	0.958613	0.229505	0.905687	0.825444	0.976501	587.90
8	0.117247	0.957270	0.951960	0.961969	0.167911	0.933426	0.946746	0.921671	589.23
9	0.098982	0.963798	0.960809	0.966443	0.192234	0.932039	0.964497	0.903394	589.60
10	0.107121	0.963798	0.962073	0.965324	0.183240	0.934813	0.914201	0.953003	588.99
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:22	0.176561		0.937908		0.886076		0.993243		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.653910	0.805625	0.798429	0.812201	0.309717	0.890511	0.862385	0.916201	564.79
2	0.169722	0.944375	0.945026	0.943780	0.199242	0.922628	0.908257	0.935754	556.61
3	0.142311	0.942500	0.943717	0.941388	0.228230	0.918248	0.917431	0.918994	557.80
4	0.139547	0.949375	0.945026	0.953349	0.202082	0.924088	0.886850	0.958101	556.62
5	0.108736	0.962500	0.959424	0.965311	0.191028	0.918248	0.966361	0.874302	558.71
6	0.100240	0.961250	0.959424	0.962919	0.214070	0.910949	0.865443	0.952514	558.74
7	0.107471	0.955625	0.956806	0.954545	0.177379	0.932847	0.926606	0.938547	561.69
8	0.118937	0.955000	0.948953	0.960526	0.262918	0.908029	0.944954	0.874302	558.19
9	0.124760	0.952500	0.942408	0.961722	0.363212	0.878832	0.990826	0.776536	559.77
10	0.171123	0.933750	0.934555	0.933014	0.255936	0.915328	0.856269	0.969274	559.02
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:15	0.270298		0.903981		0.801020		0.991342		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:05	0.200286	0.924703	0.906088	0.948331

D.1.6 Adam softmax sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.448472	0.816786	0.799544	0.832804	0.295940	0.883184	0.888000	0.878713	636.74
2	0.203665	0.927592	0.931663	0.923810	0.242665	0.903723	0.888000	0.918317	632.51
3	0.171907	0.942403	0.937358	0.947090	0.239892	0.902439	0.904000	0.900990	633.07
4	0.191311	0.931432	0.922551	0.939683	0.282408	0.863928	0.978667	0.757426	630.72
5	0.149569	0.948985	0.954442	0.943915	0.221785	0.906290	0.896000	0.915842	634.90
6	0.137724	0.947888	0.952164	0.943915	0.273348	0.879332	0.794667	0.957921	633.48
7	0.139244	0.945145	0.939636	0.950265	0.248745	0.906290	0.901333	0.910891	634.34
8	0.110978	0.960505	0.962415	0.958730	0.237006	0.911425	0.936000	0.888614	633.30
9	0.125073	0.955019	0.956720	0.953439	0.538270	0.792041	0.573333	0.995049	632.42
10	0.141425	0.949534	0.947608	0.951323	0.184117	0.921694	0.925333	0.918317	635.06
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:48:41	0.127777		0.963636		1		0.947368		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.636227	0.757595	0.709138	0.802632	0.450929	0.813314	0.849850	0.779330	567.16
2	0.291855	0.921265	0.921493	0.921053	0.341972	0.872648	0.894895	0.851955	562.01
3	0.214062	0.934904	0.935650	0.934211	0.301075	0.882779	0.939940	0.829609	565.33
4	0.164130	0.952263	0.954955	0.949761	0.293699	0.895803	0.876877	0.913408	564.89
5	0.140999	0.957223	0.957529	0.956938	0.323550	0.868307	0.960961	0.782123	565.47
6	0.141623	0.957223	0.960103	0.954545	0.266840	0.890014	0.897898	0.882682	564.90
7	0.124784	0.960942	0.962677	0.959330	0.246922	0.898698	0.843844	0.949721	563.90
8	0.117722	0.959702	0.962677	0.956938	0.285301	0.885673	0.882883	0.888268	562.97
9	0.108696	0.963422	0.966538	0.960526	0.218482	0.921852	0.897898	0.944134	563.69
10	0.130285	0.950403	0.951094	0.949761	0.237234	0.904486	0.885886	0.921788	565.19
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:37:45	0.220999		0.916667		0.881356		0.943723		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.898947	0.728269	0.796502	0.670270	0.515746	0.847863	0.750929	0.930380	482.37
2	0.411245	0.909423	0.871224	0.941892	0.398676	0.912821	0.895911	0.927215	478.99
3	0.341123	0.921110	0.899841	0.939189	0.352116	0.894017	0.940520	0.854430	478.81
4	0.285666	0.934989	0.937997	0.932432	0.320286	0.890598	0.914498	0.870253	478.71
5	0.248024	0.937911	0.936407	0.939189	0.307206	0.885470	0.866171	0.901899	479.22
6	0.211474	0.948137	0.937997	0.956757	0.273818	0.900855	0.936803	0.870253	478.90
7	0.207094	0.937180	0.931638	0.941892	0.325068	0.863248	0.970260	0.772152	481.39
8	0.206513	0.937911	0.933227	0.941892	0.246004	0.912821	0.947955	0.882911	479.38
9	0.174135	0.949598	0.947536	0.951351	0.228354	0.924786	0.903346	0.943038	478.59
10	0.153155	0.953981	0.950715	0.956757	0.235435	0.911111	0.892193	0.927215	480.58
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:27	0.172639		0.944591		0.917738		0.972900		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.098989	0.475480	0.952096	0.044655	0.725461	0.506645	1	0.060127	499.35
2	0.643877	0.762616	0.934132	0.607578	0.564310	0.897010	0.818182	0.968354	494.82
3	0.533703	0.886283	0.868263	0.902571	0.483159	0.906977	0.832168	0.974684	494.18
4	0.464276	0.910448	0.896707	0.922869	0.420214	0.940199	0.909091	0.968354	494.69
5	0.404355	0.920398	0.907186	0.932341	0.376507	0.928571	0.895105	0.958861	492.02
6	0.357737	0.933191	0.917665	0.947226	0.341856	0.941860	0.930070	0.952532	494.26
7	0.319301	0.948827	0.943114	0.953992	0.312773	0.940199	0.919580	0.958861	496.12
8	0.291304	0.943852	0.937126	0.949932	0.278019	0.945183	0.923077	0.965190	495.08
9	0.286831	0.931770	0.931138	0.932341	0.339071	0.878738	0.748252	0.996835	492.59
10	0.263956	0.936745	0.931138	0.941813	0.259755	0.936877	0.884615	0.984177	493.12
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:35	0.260288		0.934566		0.906907		0.959459		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.794502	0.730564	0.504425	0.930649	0.423733	0.875173	0.952663	0.806789	593.10
2	0.336829	0.913947	0.916561	0.911633	0.344372	0.901526	0.920118	0.885117	589.67
3	0.266743	0.925223	0.925411	0.925056	0.317809	0.877947	0.786982	0.958225	590.79
4	0.234797	0.934125	0.935525	0.932886	0.290576	0.895978	0.931953	0.864230	589.29
5	0.206401	0.937092	0.930468	0.942953	0.249200	0.918169	0.857988	0.971279	591.40
6	0.174757	0.950148	0.953224	0.947427	0.257863	0.900139	0.822485	0.968668	588.38
7	0.165017	0.948961	0.948167	0.949664	0.230967	0.915395	0.884615	0.942559	592.64
8	0.163073	0.948368	0.936789	0.958613	0.205533	0.929265	0.887574	0.966057	590.60
9	0.144316	0.954303	0.949431	0.958613	0.214993	0.919556	0.887574	0.947781	615.05
10	0.140625	0.954896	0.954488	0.955257	0.228514	0.908460	0.881657	0.932115	589.90
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:19	0.218847		0.915033		0.854430		0.979730		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.467760	0.808750	0.784031	0.831340	0.324455	0.871533	0.844037	0.896648	562.39
2	0.235770	0.920625	0.913613	0.927033	0.335742	0.856934	0.730887	0.972067	559.41
3	0.179626	0.945625	0.935864	0.954545	0.215575	0.925547	0.941896	0.910615	561.03
4	0.159424	0.939375	0.929319	0.948565	0.191542	0.934307	0.951070	0.918994	559.79
5	0.136131	0.951875	0.956806	0.947368	0.307100	0.867153	0.743119	0.980447	555.80
6	0.142523	0.948750	0.947644	0.949761	0.188277	0.922628	0.911315	0.932961	556.58
7	0.130690	0.951875	0.952880	0.950957	0.228747	0.902190	0.926606	0.879888	555.38
8	0.117164	0.956875	0.950262	0.962919	0.206922	0.913869	0.951070	0.879888	557.08
9	0.120580	0.956875	0.950262	0.962919	0.230498	0.910949	0.957187	0.868715	558.11
10	0.109174	0.959375	0.959424	0.959330	0.184771	0.929927	0.957187	0.905028	559.38
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:57	0.134578		0.950820		0.933673		0.965368		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:17	0.189188	0.937552	0.915684	0.961425

D.1.7 Adam softmax softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.644940	0.806912	0.717540	0.889947	0.624694	0.871630	0.930667	0.816832	634.54
2	0.602630	0.927592	0.916856	0.937566	0.605043	0.883184	0.850667	0.913366	634.68
3	0.583983	0.945694	0.930524	0.959788	0.592311	0.890886	0.842667	0.935644	634.10
4	0.573340	0.934723	0.919134	0.949206	0.577135	0.912709	0.920000	0.905941	630.97
5	0.557352	0.946242	0.937358	0.954497	0.564975	0.908858	0.874667	0.940594	632.54
6	0.546766	0.947888	0.935080	0.959788	0.556611	0.908858	0.866667	0.948020	630.86
7	0.532962	0.953922	0.952164	0.955556	0.547475	0.902439	0.944000	0.863861	634.41
8	0.522771	0.952276	0.954442	0.950265	0.536584	0.908858	0.853333	0.960396	635.86
9	0.508929	0.960505	0.961276	0.959788	0.547889	0.867779	0.994667	0.750000	632.93
10	0.509717	0.937466	0.927107	0.947090	0.528580	0.892169	0.816000	0.962871	634.42
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:56	0.485576		0.963636		0.970588		0.960526		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.635728	0.851209	0.893179	0.812201	0.624300	0.863965	0.924925	0.807263	568.35
2	0.601240	0.941104	0.940798	0.941388	0.604298	0.901592	0.918919	0.885475	562.80
3	0.584184	0.949163	0.953668	0.944976	0.600329	0.862518	0.831832	0.891061	565.99
4	0.575370	0.938004	0.927928	0.947368	0.586463	0.891462	0.954955	0.832402	565.51
5	0.560837	0.948543	0.953668	0.943780	0.567855	0.913169	0.867868	0.955307	596.05
6	0.552138	0.947923	0.940798	0.954545	0.563631	0.897250	0.837838	0.952514	567.53
7	0.537849	0.956603	0.951094	0.961722	0.552267	0.907381	0.900901	0.913408	562.19
8	0.525057	0.964662	0.965251	0.964115	0.545703	0.905933	0.918919	0.893855	562.70
9	0.517478	0.957223	0.945946	0.967703	0.532708	0.913169	0.906907	0.918994	565.20
10	0.501021	0.975201	0.969112	0.980861	0.527251	0.903039	0.864865	0.938547	564.29
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:37:55	0.524148		0.914216		0.887006		0.935065		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.649955	0.771366	0.694754	0.836486	0.628565	0.870085	0.862454	0.876582	484.67
2	0.615360	0.897736	0.896661	0.898649	0.616220	0.864957	0.765799	0.949367	480.69
3	0.599600	0.913075	0.899841	0.924324	0.597520	0.904274	0.921933	0.889241	478.89
4	0.584473	0.927684	0.923688	0.931081	0.588362	0.900855	0.955390	0.854430	479.41
5	0.572812	0.932067	0.922099	0.940541	0.584906	0.888889	0.962825	0.825949	478.87
6	0.560899	0.947407	0.942766	0.951351	0.570548	0.897436	0.918216	0.879747	479.46
7	0.550251	0.951059	0.933227	0.966216	0.560315	0.916239	0.892193	0.936709	478.80
8	0.542486	0.941563	0.926868	0.954054	0.551722	0.921368	0.936803	0.908228	477.82
9	0.532386	0.948868	0.934817	0.960811	0.545062	0.907692	0.925651	0.892405	480.69
10	0.520220	0.957633	0.945946	0.967568	0.547364	0.880342	0.754647	0.987342	480.47
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:12	0.541788		0.897098		0.814910		0.983740		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.649014	0.790334	0.788922	0.791610	0.621464	0.900332	0.891608	0.908228	498.57
2	0.614568	0.917555	0.902695	0.930988	0.607487	0.918605	0.923077	0.914557	491.62
3	0.601542	0.928216	0.928144	0.928281	0.593414	0.936877	0.898601	0.971519	491.69
4	0.591606	0.926795	0.920659	0.932341	0.584095	0.940199	0.919580	0.958861	491.44
5	0.577761	0.944563	0.925150	0.962111	0.573826	0.948505	0.933566	0.962025	493.20
6	0.569105	0.948827	0.943114	0.953992	0.567535	0.935216	0.891608	0.974684	493.29
7	0.558466	0.950960	0.955090	0.947226	0.568201	0.900332	0.821678	0.971519	495.89
8	0.547618	0.955224	0.956587	0.953992	0.554079	0.923588	0.877622	0.965190	492.49
9	0.546190	0.935323	0.916168	0.952639	0.538045	0.948505	0.944056	0.952532	494.06
10	0.536478	0.941720	0.941617	0.941813	0.555162	0.878738	0.755245	0.990506	492.23
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:13	0.538529		0.933144		0.870871		0.989189		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.646811	0.815430	0.798989	0.829978	0.635396	0.834951	0.680473	0.971279	593.59
2	0.613457	0.921068	0.893805	0.945190	0.608009	0.915395	0.878698	0.947781	586.79
3	0.596214	0.940059	0.916561	0.960850	0.599932	0.900139	0.964497	0.843342	594.09
4	0.582171	0.947774	0.939317	0.955257	0.585802	0.908460	0.869823	0.942559	589.20
5	0.570509	0.947181	0.936789	0.956376	0.578221	0.904300	0.908284	0.900783	588.87
6	0.563259	0.933531	0.932996	0.934004	0.566922	0.914008	0.946746	0.885117	591.30
7	0.546475	0.954303	0.948167	0.959732	0.552704	0.919556	0.890533	0.945170	589.71
8	0.535070	0.956677	0.948167	0.964206	0.540730	0.932039	0.890533	0.968668	591.39
9	0.533529	0.937092	0.936789	0.937360	0.541439	0.898752	0.825444	0.963446	590.85
10	0.521832	0.943620	0.921618	0.963087	0.528682	0.923717	0.890533	0.953003	588.92
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:19	0.538467		0.898693		0.822785		0.979730		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.641717	0.838125	0.793194	0.879187	0.628500	0.845255	0.706422	0.972067	563.53
2	0.609240	0.920000	0.914921	0.924641	0.614164	0.875912	0.944954	0.812849	556.93
3	0.595010	0.933125	0.920157	0.944976	0.599522	0.897810	0.932722	0.865922	556.88
4	0.582739	0.938125	0.938482	0.937799	0.591924	0.881752	0.816514	0.941341	558.50
5	0.573061	0.931875	0.924084	0.938995	0.573155	0.919708	0.896024	0.941341	558.01
6	0.559435	0.947500	0.931937	0.961722	0.564008	0.921168	0.920489	0.921788	558.09
7	0.550475	0.945000	0.937173	0.952153	0.567063	0.889051	0.957187	0.826816	561.18
8	0.547703	0.927500	0.913613	0.940191	0.547504	0.919708	0.929664	0.910615	560.80
9	0.533949	0.939375	0.928010	0.949761	0.545402	0.894890	0.911315	0.879888	558.01
10	0.516989	0.961250	0.951571	0.970096	0.541359	0.899270	0.960245	0.843575	561.29
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:04	0.523594		0.932084		0.913265		0.948052		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:07	0.525351	0.923145	0.879904	0.966050

D.1.8 Adam softmax tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.737604	0.510148	0.019362	0.966138	2.204455	0.518614	0	1	638.48
2	1.724548	0.518376	0	1	1.282920	0.518614	0	1	631.72
3	0.852284	0.552386	0.071754	0.998942	0.525399	0.862644	0.746667	0.970297	631.11
4	0.362069	0.945694	0.929385	0.960847	0.351305	0.915276	0.936000	0.896040	630.43
5	0.302694	0.945145	0.941913	0.948148	0.337770	0.901155	0.861333	0.938119	631.69
6	0.269109	0.950631	0.955581	0.946032	0.298495	0.917843	0.925333	0.910891	633.59
7	0.233751	0.966539	0.965831	0.967196	0.292373	0.920411	0.890667	0.948020	634.68
8	0.219441	0.959408	0.955581	0.962963	0.287371	0.908858	0.853333	0.960396	636.49
9	0.207173	0.955568	0.952164	0.958730	0.250908	0.929397	0.914667	0.943069	631.81
10	0.178203	0.970379	0.976082	0.965079	0.251306	0.926829	0.933333	0.920792	636.29
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:48:42	0.171412		0.963636		1		0.947368		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.862833	0.858648	0.828829	0.886364	0.423959	0.869754	0.846847	0.891061	571.32
2	0.216917	0.930564	0.925354	0.935407	0.301437	0.876990	0.819820	0.930168	563.00
3	0.138150	0.949783	0.945946	0.953349	0.286075	0.876990	0.927928	0.829609	566.29
4	0.110441	0.964042	0.961390	0.966507	0.238957	0.900145	0.852853	0.944134	565.75
5	0.133239	0.947923	0.943372	0.952153	0.282826	0.881331	0.915916	0.849162	567.00
6	0.108641	0.962182	0.960103	0.964115	0.303505	0.871201	0.954955	0.793296	566.78
7	0.126767	0.955983	0.945946	0.965311	0.267471	0.895803	0.804805	0.980447	562.61
8	0.111906	0.965902	0.965251	0.966507	0.218919	0.910275	0.918919	0.902235	564.41
9	0.090676	0.968382	0.971686	0.965311	0.230790	0.910275	0.933934	0.888268	566.41
10	0.099205	0.972722	0.970399	0.974880	0.215776	0.916064	0.888889	0.941341	565.30
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:38:58	0.180570		0.936275		0.909605		0.956710		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.927141	0.734843	0.683625	0.778378	0.494157	0.849573	0.869888	0.832278	484.96
2	0.388778	0.894083	0.869634	0.914865	0.359048	0.902564	0.862454	0.936709	478.77
3	0.321982	0.905771	0.883943	0.924324	0.351836	0.883761	0.929368	0.844937	480.98
4	0.264714	0.934989	0.934817	0.935135	0.317302	0.888889	0.843866	0.927215	480.08
5	0.240724	0.935719	0.933227	0.937838	0.282671	0.905983	0.866171	0.939873	479.43
6	0.216955	0.940102	0.941176	0.939189	0.263025	0.921368	0.925651	0.917722	481.89
7	0.211283	0.941563	0.939587	0.943243	0.261222	0.917949	0.892193	0.939873	481.37
8	0.182810	0.953981	0.947536	0.959459	0.325874	0.870085	0.973978	0.781646	477.50
9	0.188927	0.948137	0.945946	0.950000	0.264004	0.899145	0.925651	0.876582	481.49
10	0.159715	0.957633	0.955485	0.959459	0.228937	0.911111	0.907063	0.914557	480.41
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:19	0.183656		0.943272		0.925450		0.962060		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.970278	0.815920	0.747006	0.878214	0.341377	0.892027	0.839161	0.939873	497.76
2	0.250074	0.921819	0.919162	0.924222	0.336928	0.850498	0.699301	0.987342	494.58
3	0.190086	0.937456	0.932635	0.941813	0.270677	0.890365	0.793706	0.977848	491.88
4	0.184832	0.935323	0.928144	0.941813	0.202401	0.936877	0.898601	0.971519	494.31
5	0.150897	0.950960	0.941617	0.959405	0.201773	0.925249	0.884615	0.962025	493.26
6	0.166516	0.939588	0.944611	0.935047	0.176672	0.940199	0.902098	0.974684	493.11
7	0.161720	0.946695	0.934132	0.958051	0.137641	0.963455	0.954545	0.971519	493.90
8	0.131461	0.961620	0.958084	0.964817	0.171026	0.930233	0.884615	0.971519	494.98
9	0.121166	0.960910	0.958084	0.963464	0.148104	0.948505	0.951049	0.946203	489.63
10	0.131974	0.957356	0.950599	0.963464	0.198789	0.923588	0.965035	0.886076	492.30
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:22	0.219333		0.914652		0.984985		0.851351		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.500893	0.836795	0.826802	0.845638	0.276756	0.889043	0.923077	0.859008	592.75
2	0.191929	0.931157	0.922882	0.938479	0.248115	0.907073	0.884615	0.926893	588.81
3	0.172781	0.940653	0.936789	0.944072	0.361715	0.830791	0.991124	0.689295	591.15
4	0.161987	0.941840	0.946903	0.937360	0.220442	0.923717	0.911243	0.934726	592.96
5	0.156116	0.943027	0.936789	0.948546	0.224108	0.901526	0.893491	0.908616	590.50
6	0.128751	0.960237	0.953224	0.966443	0.194692	0.929265	0.934911	0.924282	590.30
7	0.113481	0.962611	0.960809	0.964206	0.208699	0.909847	0.943787	0.879896	587.80
8	0.115004	0.962018	0.962073	0.961969	0.238123	0.901526	0.970414	0.840731	590.70
9	0.116630	0.961424	0.950695	0.970917	0.225437	0.916782	0.940828	0.895561	590.80
10	0.111494	0.961424	0.960809	0.961969	0.195329	0.919556	0.881657	0.953003	591.36
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:50	0.213685		0.905229		0.829114		0.986486		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.668767	0.725625	0.468586	0.960526	0.858027	0.868613	0.798165	0.932961	564.91
2	0.469507	0.921875	0.909686	0.933014	0.326492	0.891971	0.902141	0.882682	557.12
3	0.205238	0.933125	0.934555	0.931818	0.258254	0.894890	0.837920	0.946927	558.20
4	0.192937	0.930000	0.916230	0.942584	0.258959	0.891971	0.944954	0.843575	556.58
5	0.165022	0.941875	0.939791	0.943780	0.198247	0.932847	0.917431	0.946927	560.00
6	0.154858	0.949375	0.947644	0.950957	0.257788	0.896350	0.963303	0.835196	559.79
7	0.146136	0.956875	0.952880	0.960526	0.194290	0.932847	0.944954	0.921788	558.85
8	0.133630	0.955625	0.955497	0.955742	0.205474	0.925547	0.902141	0.946927	558.25
9	0.146840	0.952500	0.951571	0.953349	0.237585	0.900730	0.951070	0.854749	560.10
10	0.129881	0.957500	0.955497	0.959330	0.221997	0.910949	0.911315	0.910615	558.68
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:08	0.140455		0.946136		0.928571		0.961039		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:23	0.184852	0.934866	0.929621	0.944169

D.1.9 Adam tanh ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.444342	0.523862	0.966970	0.970370	7.624619	0.518614	1	1	637.26
2	7.624619	0.518376	1	1	7.624619	0.518614	1	1	631.21
3	7.624619	0.518376	1	1	7.624619	0.518614	1	1	633.38
4	7.624619	0.518376	1	1	7.624619	0.518614	1	1	633.17
5	7.624619	0.518376	1	1	7.624619	0.518614	1	1	632.98
6	7.624619	0.518376	1	1	7.624619	0.518614	1	1	631.79
7	7.624619	0.518376	1	1	7.624619	0.518614	1	1	630.90
8	7.624619	0.518376	1	1	7.624619	0.518614	1	1	634.62
9	7.624619	0.518376	1	1	7.624619	0.518614	1	1	633.08
10	7.624619	0.518376	1	1	7.624619	0.518614	1	1	632.51
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:01	7.624619		0.690909		1		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.798378	0.482951	0.956242	0.038278	7.945990	0.481910	1	0	569.02
2	7.949046	0.481711	1	0	7.945991	0.481910	1	0	562.27
3	7.949045	0.481711	1	0	7.945991	0.481910	1	0	565.29
4	7.949046	0.481711	1	0	7.945990	0.481910	1	0	562.02
5	7.949046	0.481711	1	0	7.945991	0.481910	1	0	560.68
6	7.949046	0.481711	1	0	7.945991	0.481910	1	0	563.61
7	7.949046	0.481711	1	0	7.945991	0.481910	1	0	562.78
8	7.949046	0.481711	1	0	7.945991	0.481910	1	0	564.51
9	7.949046	0.481711	1	0	7.945990	0.481910	1	0	565.31
10	7.949046	0.481711	1	0	7.945990	0.481910	1	0	564.20
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:47	8.683502		0.433824		1		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.862597	0.542732	0.092210	0.963513	7.052442	0.540171	0	1	480.69
2	7.046773	0.540541	0	1	7.052442	0.540171	0	1	479.44
3	7.046772	0.540541	0	1	7.052442	0.540171	0	1	478.88
4	7.046774	0.540541	0	1	7.052442	0.540171	0	1	479.25
5	7.046773	0.540541	0	1	7.052442	0.540171	0	1	479.11
6	7.046772	0.540541	0	1	7.052442	0.540171	0	1	478.85
7	7.046772	0.540541	0	1	7.052442	0.540171	0	1	478.19
8	7.046772	0.540541	0	1	7.052442	0.540171	0	1	476.89
9	7.046772	0.540541	0	1	7.052442	0.540171	0	1	477.90
10	7.046773	0.540541	0	1	7.052441	0.540171	0	1	479.39
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:45	7.870883		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.030171	0.525231	0.035928	0.972936	7.286394	0.524917	0	1	492.89
2	7.281578	0.525231	0	1	7.286392	0.524917	0	1	491.28
3	7.281576	0.525231	0	1	7.286393	0.524917	0	1	490.69
4	7.281576	0.525231	0	1	7.286394	0.524917	0	1	490.71
5	7.281577	0.525231	0	1	7.286393	0.524917	0	1	491.75
6	7.281578	0.525231	0	1	7.286393	0.524917	0	1	490.92
7	7.281576	0.525231	0	1	7.286393	0.524917	0	1	489.58
8	7.281577	0.525231	0	1	7.286393	0.524917	0	1	493.29
9	7.281577	0.525231	0	1	7.286393	0.524917	0	1	493.12
10	7.281576	0.525231	0	1	7.286392	0.524917	0	1	493.14
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:10	7.264939		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.066183	0.525223	0.034134	0.958613	7.189927	0.531207	0	1	592.86
2	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.08
3	7.199788	0.530564	0	1	7.189927	0.531207	0	1	590.68
4	7.199788	0.530564	0	1	7.189928	0.531207	0	1	587.80
5	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.11
6	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.59
7	7.199788	0.530564	0	1	7.189927	0.531207	0	1	590.49
8	7.199787	0.530564	0	1	7.189927	0.531207	0	1	589.98
9	7.199786	0.530564	0	1	7.189927	0.531207	0	1	590.18
10	7.199788	0.530564	0	1	7.189927	0.531207	0	1	590.92
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:41	7.919153		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.786695	0.480625	0.962042	0.049043	8.015590	0.477372	1	0	562.14
2	8.013630	0.477500	1	0	8.015590	0.477372	1	0	555.80
3	8.013632	0.477500	1	0	8.015591	0.477372	1	0	557.58
4	8.013631	0.477500	1	0	8.015590	0.477372	1	0	558.61
5	8.013631	0.477500	1	0	8.015590	0.477372	1	0	558.94
6	8.013632	0.477500	1	0	8.015591	0.477372	1	0	558.96
7	8.013632	0.477500	1	0	8.015590	0.477372	1	0	558.53
8	8.013632	0.477500	1	0	8.015591	0.477372	1	0	559.10
9	8.013630	0.477500	1	0	8.015591	0.477372	1	0	557.41
10	8.013632	0.477500	1	0	8.015591	0.477372	1	0	557.57
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:05	8.297116		0.459016		1		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:25	7.943369	0.513422	0.500000	0.666667

D.1.10 Adam tanh sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.978449	0.701042	0.969248	0.961905	0.843217	0.830552	1	0.997525	636.70
2	0.404166	0.893033	0.952164	0.980952	0.329116	0.863928	0.968000	0.801980	629.38
3	0.255901	0.925946	0.953303	0.941799	0.328748	0.906290	0.896000	0.948020	634.78
4	0.207691	0.925398	0.955581	0.943915	0.271479	0.907574	0.944000	0.925743	633.89
5	0.213909	0.944597	0.952164	0.969312	0.282556	0.915276	0.978667	0.960396	635.53
6	0.186248	0.941854	0.962415	0.960847	0.283127	0.907574	0.912000	0.928218	632.71
7	0.174644	0.957213	0.972665	0.968254	0.299569	0.913992	0.938667	0.930693	630.93
8	0.912988	0.641251	0.850797	0.985185	0.794062	0.539153	0.973333	1	635.38
9	0.517319	0.765771	0.928246	0.998942	0.395210	0.881900	0.941333	0.960396	632.70
10	0.229109	0.942951	0.958998	0.969312	0.590750	0.808729	0.672000	0.995049	630.40
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:49	0.127073		0.963636		0.941176		0.973684		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.575018	0.518289	0	0.958134	6.273712	0.518090	0	1	567.19
2	5.906153	0.518289	0	1	5.506112	0.518090	0	1	563.70
3	5.125104	0.518289	0	1	4.736754	0.518090	0	1	562.35
4	4.399469	0.518289	0	1	4.120402	0.518090	0	1	562.99
5	4.075573	0.518289	0	0.397129	4.066483	0.518090	0	0	562.79
6	4.062947	0.518289	0	0.867225	4.063355	0.518090	0	1	561.08
7	4.062162	0.518289	0	0.546651	4.062972	0.518090	0	1	563.68
8	4.062079	0.518289	0	1	4.062984	0.518090	0	1	564.11
9	4.061940	0.518289	0	0.738038	4.062989	0.518090	0	1	563.38
10	4.061782	0.518289	0	1	4.062970	0.518090	0	1	563.33
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:39	3.690203		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.496196	0.540541	0	0.956757	6.435749	0.540171	0	1	481.72
2	6.194815	0.540541	0	1	5.935841	0.540171	0	1	478.19
3	5.668620	0.540541	0	1	5.376762	0.540171	0	1	478.50
4	5.093117	0.540541	0	1	4.802029	0.540171	0	1	476.89
5	4.525108	0.540541	0	1	4.245150	0.540171	0	1	478.89
6	4.038884	0.540541	0	1	3.900050	0.540171	0	1	476.65
7	3.892200	0.540541	0	0.381081	3.894747	0.540171	0	0	478.23
8	3.889514	0.540541	0	0.855405	3.892185	0.540171	0	1	478.78
9	3.888723	0.540541	0	1	3.891407	0.540171	0	1	478.99
10	3.888715	0.540541	0	1	3.891408	0.540171	0	1	476.89
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:22:56	4.306658		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.068114	0.474058	0.958084	0	4.725742	0.475083	1	0	494.37
2	4.078850	0.507463	0.505988	0.014885	0.699780	0.709302	0.024476	0.325949	492.28
3	0.604685	0.603411	0.522455	0.913396	0.519989	0.812292	0.881119	1	491.21
4	0.467172	0.850036	0.847305	0.891746	0.401879	0.893688	0.828671	0.854430	492.47
5	0.378918	0.884861	0.863772	0.852503	0.316310	0.905316	0.902098	0.911392	492.21
6	0.319222	0.921109	0.893713	0.917456	0.266573	0.928571	0.898601	0.952532	492.19
7	0.245954	0.930348	0.932635	0.935047	0.254954	0.931894	0.905594	0.958861	492.28
8	0.260275	0.931770	0.929641	0.952639	0.200364	0.941860	0.951049	0.946203	492.39
9	0.217745	0.938877	0.956587	0.949932	0.220623	0.940199	0.982517	0.936709	491.61
10	0.207646	0.948827	0.958084	0.960758	0.252476	0.923588	0.884615	0.990506	493.67
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:00	0.218588		0.930299		0.927928		0.970270		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.712475	0.521662	0	0	7.712473	0.538141	0	0	592.74
2	7.712475	0.519881	0	0	7.712473	0.524272	0	0	589.68
3	7.712475	0.522255	0	0	7.712473	0.522885	0	0	587.69
4	7.712475	0.518694	0	0	7.712473	0.528433	0	0	588.51
5	7.712475	0.521662	0	0	7.712473	0.525659	0	0	587.70
6	7.712475	0.528190	0	0	7.712473	0.528433	0	0	587.87
7	7.712475	0.521662	0	0	7.712473	0.509015	0	0	587.72
8	7.712475	0.520475	0	0	7.712473	0.520111	0	0	589.69
9	7.712474	0.515134	0	0	7.712473	0.525659	0	0	587.69
10	7.712475	0.524036	0	0	7.712473	0.532594	0	0	588.96
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:04	7.712474		0.473856		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.712537	0.521250	0	0	7.712474	0.522628	0	0	562.70
2	7.712475	0.522500	0	0	7.712474	0.522628	0	0	556.89
3	7.712475	0.522500	0	0	7.712474	0.522628	0	0	557.48
4	7.712475	0.522500	0	0	7.712474	0.522628	0	0	556.19
5	7.712475	0.522500	0	0	7.712474	0.522628	0	0	557.36
6	7.712475	0.522500	0	0	7.712474	0.522628	0	0	556.12
7	7.712475	0.522500	0	0	7.712474	0.522628	0	0	557.34
8	7.712475	0.522500	0	0	7.712474	0.522628	0	0	559.48
9	7.712475	0.522500	0	0	7.712474	0.522628	0	0	557.20
10	7.712475	0.522500	0	0	7.712474	0.522628	0	0	558.82
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:01	7.712474		0.540984		0		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:35	3.961245	0.660293	0.311517	0.657326

D.1.11 Adam tanh softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.551404	0.481075	0	0	4.414944	0.481386	0	0	635.68
2	4.403006	0.481624	0	0	4.394918	0.481386	0	0	632.86
3	4.386689	0.481624	0	0	4.381892	0.481386	0	0	629.99
4	4.375659	0.481624	0	0	4.373010	0.481386	0	0	633.02
5	4.367901	0.481624	0	0	4.366305	0.481386	0	0	632.88
6	4.353104	0.481624	0	0	4.340190	0.481386	0	0	633.69
7	4.313528	0.481624	0	0	4.314590	0.481386	0	0	631.38
8	4.282333	0.481624	0	0	4.282033	0.481386	0	0	631.60
9	4.255601	0.481624	0	0	4.267859	0.481386	0	0	631.11
10	4.242303	0.481624	0	0	4.249483	0.481386	0	0	634.26
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:56	5.493843		0.309091		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.142946	0.671420	0	0	0.896652	0.810420	0	0	568.01
2	0.829883	0.889647	0	0	0.801748	0.863965	0	0	561.87
3	0.750050	0.934904	0	0	0.744089	0.891462	0	0	561.00
4	0.687815	0.949163	0	0	0.721591	0.862518	0	0	563.10
5	0.650255	0.937384	0	0	0.672059	0.878437	0	0	558.99
6	0.605221	0.946063	0	0	0.675443	0.855282	0	0	563.79
7	0.558372	0.956603	0	0	0.589625	0.910275	0	0	560.39
8	0.525299	0.964042	0	0	0.562868	0.924747	0	0	564.58
9	0.495775	0.961562	0	0	0.541177	0.917511	0	0	563.50
10	0.491090	0.947923	0	0	0.538037	0.904486	0	0	562.20
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:34	0.520416		0.916667		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.159137	0.532505	0	0	0.890214	0.459829	0	0	482.17
2	0.872154	0.459459	0	0	0.853284	0.459829	0	0	479.13
3	0.839365	0.495252	0	0	0.824594	0.540171	0	0	478.40
4	0.813472	0.540541	0	0	0.801922	0.540171	0	0	477.16
5	0.792806	0.540541	0	0	0.783496	0.540171	0	0	476.02
6	0.776200	0.540541	0	0	0.768467	0.540171	0	0	477.43
7	0.762420	0.540541	0	0	0.755979	0.540171	0	0	477.59
8	0.750970	0.540541	0	0	0.745558	0.540171	0	0	477.09
9	0.741365	0.540541	0	0	0.736823	0.540171	0	0	477.30
10	0.733217	0.540541	0	0	0.729484	0.540171	0	0	477.68
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:43	0.732960		0.486807		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.956262	0.572850	0	0	0.864365	0.855482	0	0	491.96
2	0.837573	0.823738	0	0	0.805947	0.898671	0	0	493.10
3	0.779536	0.906183	0	0	0.752391	0.915282	0	0	488.72
4	0.725850	0.930348	0	0	0.704808	0.916944	0	0	488.70
5	0.678023	0.940298	0	0	0.668669	0.915282	0	0	493.65
6	0.640261	0.932480	0	0	0.627406	0.918605	0	0	490.92
7	0.602596	0.939588	0	0	0.594185	0.926910	0	0	491.61
8	0.566984	0.945984	0	0	0.571293	0.920266	0	0	487.59
9	0.533348	0.956645	0	0	0.538021	0.926910	0	0	491.60
10	0.513421	0.944563	0	0	0.539290	0.893688	0	0	491.20
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:02	0.517938		0.914652		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.046061	0.710979	0	0	0.852357	0.746186	0	0	589.04
2	0.798839	0.912760	0	0	0.777836	0.875173	0	0	588.59
3	0.733474	0.931751	0	0	0.737382	0.828017	0	0	590.39
4	0.687586	0.917507	0	0	0.677848	0.890430	0	0	588.90
5	0.632594	0.947774	0	0	0.634356	0.907073	0	0	587.61
6	0.594377	0.940653	0	0	0.595606	0.919556	0	0	590.18
7	0.551865	0.948368	0	0	0.620540	0.847434	0	0	588.00
8	0.522395	0.948961	0	0	0.523612	0.927878	0	0	590.08
9	0.486135	0.956083	0	0	0.519477	0.912621	0	0	587.21
10	0.457316	0.960237	0	0	0.479264	0.934813	0	0	584.21
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:39	0.474546		0.924837		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.069138	0.493750	0	0	0.872566	0.477372	0	0	558.94
2	0.849954	0.477500	0	0	0.828152	0.477372	0	0	557.10
3	0.812420	0.477500	0	0	0.797214	0.477372	0	0	558.22
4	0.785879	0.477500	0	0	0.774722	0.477372	0	0	555.97
5	0.766174	0.477500	0	0	0.757721	0.477372	0	0	555.93
6	0.751151	0.477500	0	0	0.744611	0.477372	0	0	557.88
7	0.739483	0.477500	0	0	0.734361	0.477372	0	0	557.70
8	0.730312	0.477500	0	0	0.726265	0.477372	0	0	559.62
9	0.723054	0.477500	0	0	0.719801	0.477372	0	0	557.12
10	0.717257	0.477500	0	0	0.714614	0.477372	0	0	555.98
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:58	0.715972		0.459016		0		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:29	1.409279	0.668512	0	0

D.1.12 Adam tanh tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.988016	0.512891	0.686788	0.968254	3.909194	0.518614	0.722667	1	633.70
2	3.835374	0.518376	0.904328	1	3.832390	0.518614	0.952000	1	630.79
3	3.941269	0.518376	0.955581	1	4.342512	0.518614	1	1	633.07
4	4.438621	0.518376	1	1	4.338601	0.518614	1	1	631.89
5	4.223755	0.518376	1	1	4.099259	0.518614	1	1	632.07
6	4.082417	0.518376	1	1	4.072317	0.518614	1	1	634.12
7	4.054943	0.518376	1	1	4.043156	0.518614	1	1	634.00
8	4.027668	0.518376	1	1	4.018069	0.518614	1	1	632.29
9	4.000841	0.518376	1	1	3.993765	0.518614	1	1	629.71
10	3.973234	0.518376	1	1	3.968772	0.518614	1	1	631.89
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:05	2.693875		0.690909		1		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.712549	0.478611	0	0	7.712474	0.481910	0	0	565.47
2	7.712475	0.481711	0	0	7.712474	0.481910	0	0	563.76
3	7.712475	0.481711	0	0	7.712474	0.481910	0	0	561.19
4	7.712475	0.481711	0	0	7.712474	0.481910	0	0	563.70
5	7.712475	0.481711	0	0	7.712474	0.481910	0	0	565.18
6	7.712475	0.481711	0	0	7.712474	0.481910	0	0	563.67
7	7.712475	0.481711	0	0	7.712474	0.481910	0	0	563.31
8	7.712475	0.481711	0	0	7.712474	0.481910	0	0	564.72
9	7.712475	0.481711	0	0	7.712474	0.481910	0	0	565.09
10	7.712475	0.481711	0	0	7.712474	0.481910	0	0	567.05
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:50	7.712474		0.433824		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.352196	0.463842	0.976153	0.886486	4.437799	0.459829	1	1	485.00
2	4.362791	0.459459	1	0.859459	4.334100	0.459829	1	0.816456	480.88
3	4.266832	0.459459	1	0.928378	4.412472	0.459829	1	0.996835	483.40
4	4.394865	0.459459	1	0.982432	4.356502	0.459829	1	0.908228	479.69
5	4.291216	0.459459	1	0.939189	4.297771	0.459829	1	0.835443	478.80
6	4.235934	0.459459	1	0.929730	4.440566	0.459829	1	0.825949	477.49
7	4.235875	0.459459	1	0.944595	4.304495	0.459829	1	0.898734	477.52
8	4.213205	0.459459	1	0.954054	4.261257	0.459829	1	0.895570	481.51
9	4.215704	0.459459	1	0.952703	4.305834	0.459829	1	0.841772	480.77
10	4.213059	0.459459	1	0.959459	4.265906	0.459829	1	0.949367	479.70
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:44	3.803554		0.513193		1		0.975610		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.693588	0.468372	0.937126	0.706360	4.231127	0.475083	1	0.952532	498.23
2	4.189637	0.474769	1	0.901218	4.154014	0.475083	1	0.889241	491.07
3	4.191221	0.474769	1	0.941813	4.224801	0.475083	1	0.974684	493.39
4	4.177782	0.474769	1	0.925575	4.149148	0.475083	1	0.958861	494.89
5	4.128103	0.474769	1	0.939107	4.141212	0.475083	1	0.943038	494.62
6	4.115956	0.474769	1	0.953992	4.134193	0.475083	1	0.977848	494.18
7	4.137319	0.474769	1	0.953992	4.136631	0.475083	1	0.977848	494.30
8	4.128668	0.474769	1	0.936401	4.127552	0.475083	1	0.908228	491.24
9	4.098639	0.474769	1	0.955345	4.150671	0.475083	1	0.958861	493.82
10	4.125329	0.474769	1	0.963464	4.138741	0.475083	1	0.977848	493.41
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:36	4.109037		0.473684		1		0.975676		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.028187	0.528190	0.756005	0.965324	3.764326	0.531207	0.943787	1	592.51
2	3.739497	0.530564	0.908976	1	3.741462	0.531207	0.961538	1	591.90
3	3.852400	0.530564	0.969659	1	3.812553	0.531207	0.937870	1	590.60
4	3.774150	0.530564	0.884956	1	3.828130	0.531207	0.825444	1	589.53
5	3.718082	0.530564	0.929204	1	3.838464	0.531207	0.801775	1	590.79
6	3.696128	0.530564	0.927939	1	3.707146	0.531207	0.937870	1	589.20
7	3.706388	0.530564	0.924147	1	3.730623	0.531207	0.970414	1	589.08
8	3.696235	0.530564	0.943110	1	3.726982	0.531207	0.926036	1	588.69
9	3.777906	0.530564	0.967130	1	3.747300	0.531207	0.931953	1	590.10
10	3.686100	0.530564	0.953224	1	3.695863	0.531207	0.937870	1	589.92
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:47	4.068248		0.483660		0.886076		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.314090	0.474375	0.962042	0.763158	4.279122	0.477372	1	0.567039	562.89
2	4.243182	0.477500	1	0.858852	4.240544	0.477372	1	0.932961	557.33
3	4.209716	0.477500	1	0.912679	4.216650	0.477372	1	0.734637	559.09
4	4.151111	0.477500	1	0.931818	4.207540	0.477372	1	0.840782	558.30
5	4.097239	0.477500	1	0.953349	4.209046	0.477372	1	0.851955	559.28
6	4.116466	0.477500	1	0.938995	4.123690	0.477372	1	0.966480	559.80
7	4.117118	0.477500	1	0.941388	4.172773	0.477372	1	0.877095	562.49
8	4.078740	0.477500	1	0.964115	4.199550	0.477372	1	0.907821	562.17
9	4.108130	0.477500	1	0.961722	4.163533	0.477372	1	0.983240	560.12
10	4.112747	0.477500	1	0.960526	4.192461	0.477372	1	0.997207	558.63
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:14	4.344434		0.459016		1		0.995671		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:43	4.455270	0.509048	0.814346	0.824493

D.1.13 Ftrl sigmoid ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.696336	0.480527	0.177677	0.032804	0.693217	0.481386	0	0	633.28
2	0.693161	0.481624	0	0	0.693125	0.481386	0	0	630.89
3	0.693072	0.481624	0	0	0.693046	0.481386	0	0	631.91
4	0.692971	0.724081	0	0	0.692971	0.783055	0	0	628.33
5	0.692882	0.695008	0	0.040212	0.692874	0.518614	0	0.264851	629.08
6	0.692793	0.521119	0	0.538624	0.692807	0.518614	0	0.448020	634.40
7	0.692674	0.518376	0	0.867725	0.692683	0.518614	0	0.688119	634.38
8	0.692560	0.520571	0	0.853968	0.692581	0.518614	0	0.806931	633.88
9	0.692360	0.519473	0	0.942857	0.692399	0.528883	0	0.809406	631.03
10	0.692159	0.543609	0	0.900529	0.692229	0.560976	0	0.750000	633.58
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:40	0.691387		0.800000		0		0.750000		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.699712	0.513949	0.167310	0.958134	0.692656	0.518090	0	1	568.37
2	0.692575	0.518289	0	1	0.692485	0.518090	0	1	562.91
3	0.692323	0.518289	0	1	0.692234	0.518090	0	1	560.83
4	0.692051	0.518289	0	1	0.691976	0.518090	0	1	562.67
5	0.691764	0.518289	0	1	0.691692	0.518090	0	1	562.78
6	0.691352	0.518289	0	1	0.691293	0.518090	0	1	560.31
7	0.690879	0.518289	0	1	0.690948	0.518090	0	1	562.32
8	0.690363	0.518289	0	1	0.690354	0.518090	0	1	565.89
9	0.689790	0.518289	0	1	0.689713	0.518090	0	1	561.78
10	0.688951	0.518289	0	1	0.689079	0.518090	0	1	562.70
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:37:35	0.686689		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.702772	0.459459	0.068362	0.043243	0.693952	0.459829	0	0	483.00
2	0.693601	0.459459	0	0	0.693378	0.459829	0	0	478.12
3	0.693111	0.459459	0	0	0.692943	0.461538	0	0	476.38
4	0.692758	0.626004	0	0.002703	0.692677	0.553846	0	0	476.79
5	0.692341	0.544193	0	0.385135	0.691727	0.540171	0	1	479.71
6	0.691022	0.540541	0	1	0.690576	0.540171	0	1	475.89
7	0.689775	0.540541	0	1	0.689490	0.540171	0	1	475.30
8	0.688728	0.540541	0	1	0.688322	0.540171	0	1	479.86
9	0.687537	0.540541	0	1	0.687253	0.540171	0	1	479.20
10	0.686374	0.540541	0	1	0.686149	0.540171	0	1	477.07
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:13	0.690706		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.697233	0.521677	0.037425	0.952639	0.692954	0.524917	0	1	496.31
2	0.692918	0.525231	0	1	0.692868	0.524917	0	1	492.88
3	0.692820	0.525231	0	1	0.692763	0.524917	0	1	491.88
4	0.692735	0.525231	0	1	0.692645	0.524917	0	1	491.30
5	0.692608	0.525231	0	1	0.692511	0.524917	0	1	491.43
6	0.692461	0.525231	0	1	0.692362	0.524917	0	1	492.50
7	0.692304	0.525231	0	1	0.692152	0.524917	0	1	492.79
8	0.692110	0.525231	0	1	0.691912	0.524917	0	1	492.10
9	0.691872	0.525231	0	1	0.691724	0.524917	0	1	493.48
10	0.691629	0.525231	0	1	0.691445	0.524917	0	1	491.29
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:12	0.691372		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.695331	0.527003	0.036662	0.739374	0.692660	0.531207	0	1	594.64
2	0.692554	0.530564	0	1	0.692404	0.531207	0	1	588.29
3	0.692294	0.530564	0	1	0.692091	0.531207	0	1	589.79
4	0.691958	0.530564	0	1	0.691740	0.531207	0	1	587.73
5	0.691584	0.530564	0	1	0.691325	0.531207	0	1	586.40
6	0.691087	0.530564	0	1	0.690837	0.531207	0	1	589.82
7	0.690611	0.530564	0	1	0.690253	0.531207	0	1	585.97
8	0.690006	0.530564	0	1	0.689659	0.531207	0	1	584.69
9	0.689431	0.530564	0	1	0.689064	0.531207	0	1	585.61
10	0.688634	0.530564	0	1	0.688474	0.531207	0	1	588.16
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:30	0.693225		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.695289	0.498125	0.208115	0.703349	0.693128	0.522628	0	1	563.38
2	0.693120	0.522500	0	1	0.693109	0.522628	0	1	557.21
3	0.693102	0.522500	0	1	0.693091	0.522628	0	1	557.16
4	0.693084	0.522500	0	1	0.693074	0.522628	0	1	555.51
5	0.693066	0.522500	0	1	0.693055	0.522628	0	1	555.82
6	0.693047	0.522500	0	1	0.693037	0.522628	0	1	558.18
7	0.693029	0.522500	0	1	0.693017	0.522628	0	1	558.52
8	0.693008	0.522500	0	1	0.692996	0.522628	0	1	558.40
9	0.692986	0.522500	0	1	0.692972	0.522628	0	1	557.57
10	0.692961	0.522500	0	1	0.692946	0.522628	0	1	561.32
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:05	0.692804		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:53	0.691030	0.567324	0	0.958333

D.1.14 Ftrl sigmoid sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694767	0.518376	0.030752	1	0.692776	0.518614	0	1	637.59
2	0.692784	0.518376	0	1	0.692605	0.518614	0	1	634.11
3	0.692663	0.518376	0	1	0.692517	0.518614	0	1	633.13
4	0.692606	0.518376	0	1	0.692473	0.518614	0	1	634.39
5	0.692518	0.518376	0	1	0.692448	0.518614	0	1	634.49
6	0.692537	0.518376	0	1	0.692429	0.518614	0	1	635.89
7	0.692479	0.518376	0	1	0.692415	0.518614	0	1	636.60
8	0.692500	0.518376	0	1	0.692404	0.518614	0	1	632.61
9	0.692502	0.518376	0	1	0.692403	0.518614	0	1	632.21
10	0.692454	0.518376	0	1	0.692394	0.518614	0	1	633.03
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:48:01	0.680731		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694773	0.518289	0.041184	0.952153	0.692813	0.518090	0	1	568.15
2	0.692773	0.518289	0	1	0.692648	0.518090	0	1	564.98
3	0.692688	0.518289	0	1	0.692559	0.518090	0	1	564.07
4	0.692709	0.518289	0	1	0.692576	0.518090	0	1	562.91
5	0.692632	0.518289	0	1	0.692547	0.518090	0	1	564.21
6	0.692606	0.518289	0	1	0.692537	0.518090	0	1	565.48
7	0.692670	0.518289	0	1	0.692503	0.518090	0	1	562.99
8	0.692570	0.518289	0	1	0.692512	0.518090	0	1	561.94
9	0.692524	0.518289	0	1	0.692523	0.518090	0	1	565.32
10	0.692532	0.518289	0	1	0.692498	0.518090	0	1	566.17
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:52	0.689816		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692163	0.540541	0.033386	0.952703	0.691698	0.540171	0	1	483.11
2	0.691292	0.540541	0	1	0.690878	0.540171	0	1	477.39
3	0.690701	0.540541	0	1	0.690477	0.540171	0	1	480.55
4	0.690439	0.540541	0	1	0.690178	0.540171	0	1	477.91
5	0.690140	0.540541	0	1	0.690048	0.540171	0	1	479.08
6	0.690063	0.540541	0	1	0.689962	0.540171	0	1	481.12
7	0.690031	0.540541	0	1	0.689896	0.540171	0	1	480.14
8	0.689880	0.540541	0	1	0.689863	0.540171	0	1	478.61
9	0.689861	0.540541	0	1	0.689824	0.540171	0	1	480.48
10	0.689828	0.540541	0	1	0.689809	0.540171	0	1	480.20
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:31	0.698277		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.695132	0.525231	0.043413	1	0.692681	0.524917	0	1	498.10
2	0.692613	0.525231	0	1	0.692340	0.524917	0	1	493.38
3	0.692333	0.525231	0	1	0.692138	0.524917	0	1	492.11
4	0.692143	0.525231	0	1	0.692033	0.524917	0	1	493.29
5	0.692055	0.525231	0	1	0.691972	0.524917	0	1	493.59
6	0.691980	0.525231	0	1	0.691925	0.524917	0	1	493.40
7	0.691952	0.525231	0	1	0.691893	0.524917	0	1	491.97
8	0.691926	0.525231	0	1	0.691877	0.524917	0	1	492.31
9	0.691938	0.525231	0	1	0.691858	0.524917	0	1	492.69
10	0.691901	0.525231	0	1	0.691848	0.524917	0	1	493.90
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:11	0.691718		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.698774	0.530564	0	0.914989	0.691955	0.531207	0	1	593.73
2	0.691924	0.530564	0	1	0.691657	0.531207	0	1	586.93
3	0.691627	0.530564	0	1	0.691362	0.531207	0	1	589.33
4	0.691354	0.530564	0	1	0.691201	0.531207	0	1	589.16
5	0.691295	0.530564	0	1	0.691118	0.531207	0	1	591.11
6	0.691163	0.530564	0	1	0.691033	0.531207	0	1	591.58
7	0.691140	0.530564	0	1	0.690993	0.531207	0	1	587.59
8	0.691063	0.530564	0	1	0.690950	0.531207	0	1	589.58
9	0.691030	0.530564	0	1	0.690900	0.531207	0	1	588.74
10	0.690996	0.530564	0	1	0.690882	0.531207	0	1	588.69
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:12	0.696768		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.699490	0.522500	0	0.959330	0.692424	0.522628	0	1	562.96
2	0.692391	0.522500	0	1	0.692262	0.522628	0	1	558.49
3	0.692246	0.522500	0	1	0.692145	0.522628	0	1	559.09
4	0.692187	0.522500	0	1	0.692074	0.522628	0	1	560.98
5	0.692071	0.522500	0	1	0.692014	0.522628	0	1	558.51
6	0.692047	0.522500	0	1	0.691982	0.522628	0	1	559.03
7	0.692041	0.522500	0	1	0.691934	0.522628	0	1	559.89
8	0.691963	0.522500	0	1	0.691911	0.522628	0	1	561.29
9	0.691925	0.522500	0	1	0.691857	0.522628	0	1	559.72
10	0.691886	0.522500	0	1	0.691836	0.522628	0	1	559.61
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:16	0.690237		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:51	0.691258	0.549142	0	1

D.1.15 Ftrl sigmoid softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693155	0.512891	0.027335	0.983069	0.693133	0.518614	0	1	638.31
2	0.693131	0.518376	0	1	0.693121	0.518614	0	1	634.88
3	0.693117	0.518376	0	1	0.693109	0.518614	0	1	631.80
4	0.693106	0.518376	0	1	0.693098	0.518614	0	1	634.29
5	0.693096	0.518376	0	1	0.693088	0.518614	0	1	631.20
6	0.693087	0.518376	0	1	0.693080	0.518614	0	1	630.28
7	0.693078	0.518376	0	1	0.693071	0.518614	0	1	632.67
8	0.693070	0.518376	0	1	0.693063	0.518614	0	1	632.22
9	0.693062	0.518376	0	1	0.693055	0.518614	0	1	633.99
10	0.693054	0.518376	0	1	0.693048	0.518614	0	1	632.39
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:12	0.692091		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693147	0.508370	0.195624	0.796651	0.693137	0.518090	0	1	566.38
2	0.693132	0.518289	0	1	0.693124	0.518090	0	1	564.49
3	0.693121	0.518289	0	1	0.693117	0.518090	0	1	566.00
4	0.693116	0.518289	0	1	0.693109	0.518090	0	1	563.02
5	0.693106	0.518289	0	1	0.693099	0.518090	0	1	562.27
6	0.693095	0.518289	0	1	0.693088	0.518090	0	1	566.02
7	0.693085	0.518289	0	1	0.693081	0.518090	0	1	567.60
8	0.693076	0.518289	0	1	0.693072	0.518090	0	1	562.39
9	0.693069	0.518289	0	1	0.693066	0.518090	0	1	566.50
10	0.693063	0.518289	0	1	0.693060	0.518090	0	1	563.41
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:38:45	0.692821		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693131	0.540541	0.050874	0.956757	0.693093	0.540171	0	1	482.23
2	0.693074	0.540541	0	1	0.693048	0.540171	0	1	477.78
3	0.693030	0.540541	0	1	0.693009	0.540171	0	1	479.20
4	0.692990	0.540541	0	1	0.692971	0.540171	0	1	479.82
5	0.692954	0.540541	0	1	0.692934	0.540171	0	1	478.69
6	0.692917	0.540541	0	1	0.692899	0.540171	0	1	478.19
7	0.692884	0.540541	0	1	0.692868	0.540171	0	1	479.39
8	0.692852	0.540541	0	1	0.692839	0.540171	0	1	480.27
9	0.692824	0.540541	0	1	0.692812	0.540171	0	1	477.89
10	0.692797	0.540541	0	1	0.692785	0.540171	0	1	477.33
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:27	0.693280		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693140	0.510306	0.263473	0.723951	0.693127	0.524917	0	1	548.97
2	0.693118	0.525231	0	1	0.693108	0.524917	0	1	491.59
3	0.693102	0.525231	0	1	0.693090	0.524917	0	1	493.69
4	0.693085	0.525231	0	1	0.693075	0.524917	0	1	493.61
5	0.693069	0.525231	0	1	0.693060	0.524917	0	1	493.90
6	0.693054	0.525231	0	1	0.693047	0.524917	0	1	494.78
7	0.693042	0.525231	0	1	0.693034	0.524917	0	1	494.79
8	0.693028	0.525231	0	1	0.693022	0.524917	0	1	493.56
9	0.693017	0.525231	0	1	0.693011	0.524917	0	1	495.04
10	0.693006	0.525231	0	1	0.693000	0.524917	0	1	493.89
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:26:23	0.692991		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693148	0.524036	0.030341	0.959732	0.693109	0.531207	0	1	589.64
2	0.693094	0.530564	0	1	0.693074	0.531207	0	1	587.93
3	0.693063	0.530564	0	1	0.693044	0.531207	0	1	591.09
4	0.693035	0.530564	0	1	0.693017	0.531207	0	1	589.60
5	0.693010	0.530564	0	1	0.692993	0.531207	0	1	588.70
6	0.692989	0.530564	0	1	0.692972	0.531207	0	1	587.70
7	0.692967	0.530564	0	1	0.692951	0.531207	0	1	587.70
8	0.692947	0.530564	0	1	0.692931	0.531207	0	1	590.80
9	0.692925	0.530564	0	1	0.692908	0.531207	0	1	590.33
10	0.692907	0.530564	0	1	0.692890	0.531207	0	1	589.98
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:42	0.693296		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693132	0.523750	0.018325	0.971292	0.693129	0.522628	0	1	564.34
2	0.693123	0.522500	0	1	0.693112	0.522628	0	1	558.19
3	0.693107	0.522500	0	1	0.693098	0.522628	0	1	559.74
4	0.693094	0.522500	0	1	0.693084	0.522628	0	1	561.56
5	0.693081	0.522500	0	1	0.693072	0.522628	0	1	559.01
6	0.693068	0.522500	0	1	0.693060	0.522628	0	1	559.08
7	0.693057	0.522500	0	1	0.693049	0.522628	0	1	559.39
8	0.693047	0.522500	0	1	0.693039	0.522628	0	1	559.21
9	0.693036	0.522500	0	1	0.693029	0.522628	0	1	559.87
10	0.693027	0.522500	0	1	0.693020	0.522628	0	1	557.61
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:10	0.692913		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:06	0.692899	0.549142	0	1

D.1.16 Ftrl sigmoid tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.695504	0.517279	0.037585	0.019048	0.693120	0.521181	0	0	637.91
2	0.693095	0.541964	0	0.029630	0.693071	0.518614	0	0.002475	632.87
3	0.693040	0.518376	0	0.526984	0.693017	0.518614	0	1	634.09
4	0.692984	0.518376	0	1	0.692957	0.518614	0	1	634.30
5	0.692916	0.518376	0	1	0.692889	0.518614	0	1	636.65
6	0.692823	0.518376	0	1	0.692787	0.518614	0	1	633.64
7	0.692718	0.518376	0	1	0.692685	0.518614	0	1	632.08
8	0.692571	0.518376	0	1	0.692525	0.518614	0	1	633.09
9	0.692398	0.518376	0	1	0.692348	0.518614	0	1	631.21
10	0.692174	0.518376	0	1	0.692107	0.518614	0	1	636.06
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:22	0.687557		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.696244	0.520149	0.963964	0.958134	0.692570	0.518090	1	1	568.13
2	0.692421	0.518289	0.332046	1	0.692286	0.518090	0	1	567.41
3	0.692191	0.518289	0	1	0.691995	0.518090	0	1	565.11
4	0.691739	0.518289	0	1	0.691630	0.518090	0	1	565.88
5	0.691299	0.518289	0	1	0.691231	0.518090	0	1	566.09
6	0.690779	0.518289	0	1	0.690762	0.518090	0	1	567.70
7	0.690172	0.518289	0	1	0.690148	0.518090	0	1	564.76
8	0.689363	0.518289	0	1	0.689236	0.518090	0	1	564.91
9	0.688400	0.518289	0	1	0.688554	0.518090	0	1	566.68
10	0.687121	0.518289	0	1	0.687194	0.518090	0	1	564.99
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:08	0.685578		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.699049	0.458729	0.949126	0.959459	0.693171	0.459829	1	1	480.54
2	0.692802	0.539810	0.985692	1	0.692350	0.540171	0.985130	1	478.21
3	0.692057	0.540541	0.146264	1	0.691642	0.540171	0	1	480.05
4	0.691403	0.540541	0	1	0.691064	0.540171	0	1	479.21
5	0.690790	0.540541	0	1	0.690486	0.540171	0	1	479.60
6	0.690142	0.540541	0	1	0.689832	0.540171	0	1	479.99
7	0.689553	0.540541	0	1	0.689300	0.540171	0	1	479.00
8	0.688982	0.540541	0	1	0.688881	0.540171	0	1	479.71
9	0.688497	0.540541	0	1	0.688161	0.540171	0	1	482.94
10	0.687738	0.540541	0	1	0.687532	0.540171	0	1	480.55
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:36	0.693980		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694034	0.479033	0.958084	0.046008	0.693165	0.475083	1	0	497.06
2	0.693138	0.496091	1	0.497970	0.693088	0.524917	0.965035	1	492.62
3	0.693075	0.525231	0.092814	1	0.693018	0.524917	0	1	492.77
4	0.693000	0.525231	0	1	0.692947	0.524917	0	1	494.18
5	0.692917	0.525231	0	1	0.692864	0.524917	0	1	494.60
6	0.692847	0.525231	0	1	0.692778	0.524917	0	1	494.68
7	0.692754	0.525231	0	1	0.692680	0.524917	0	1	494.96
8	0.692646	0.525231	0	1	0.692566	0.524917	0	1	494.11
9	0.692533	0.525231	0	1	0.692422	0.524917	0	1	494.70
10	0.692391	0.525231	0	1	0.692257	0.524917	0	1	493.69
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:20	0.692264		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694942	0.531157	0.035398	0.967562	0.692886	0.531207	0	1	593.68
2	0.692835	0.530564	0	1	0.692714	0.531207	0	1	591.16
3	0.692630	0.530564	0	1	0.692500	0.531207	0	1	589.41
4	0.692418	0.530564	0	1	0.692222	0.531207	0	1	590.83
5	0.692157	0.530564	0	1	0.691944	0.531207	0	1	590.58
6	0.691808	0.530564	0	1	0.691555	0.531207	0	1	592.07
7	0.691428	0.530564	0	1	0.691170	0.531207	0	1	589.90
8	0.690960	0.530564	0	1	0.690723	0.531207	0	1	587.99
9	0.690537	0.530564	0	1	0.690141	0.531207	0	1	589.51
10	0.689974	0.530564	0	1	0.689623	0.531207	0	1	589.49
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:48	0.693324		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.698275	0.472500	0.954188	0.956938	0.693333	0.477372	1	1	561.36
2	0.693095	0.480000	1	1	0.692878	0.534307	1	1	556.79
3	0.692651	0.524375	0.658377	1	0.692477	0.522628	0.703364	1	556.62
4	0.692248	0.522500	0.324607	1	0.692096	0.522628	0	1	557.77
5	0.691827	0.522500	0	1	0.691710	0.522628	0	1	556.71
6	0.691340	0.522500	0	1	0.691249	0.522628	0	1	560.21
7	0.690867	0.522500	0	1	0.690726	0.522628	0	1	556.43
8	0.690311	0.522500	0	1	0.690227	0.522628	0	1	555.95
9	0.689598	0.522500	0	1	0.689642	0.522628	0	1	556.53
10	0.688885	0.522500	0	1	0.688823	0.522628	0	1	553.90
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:54	0.687280		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:21	0.689997	0.549142	0	1

D.1.17 Ftrl softmax ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.696120	0.483818	0.965831	0.035979	0.693243	0.481386	1	0	634.88
2	0.693126	0.481624	1	0	0.693074	0.481386	1	0	631.49
3	0.692969	0.481624	1	0	0.692936	0.481386	1	0	632.78
4	0.692817	0.600110	0.989749	0.238095	0.692816	0.613607	0.994667	0.259901	630.71
5	0.692649	0.835985	0.879271	0.795767	0.692665	0.816431	0.690667	0.933168	631.10
6	0.692461	0.852441	0.793850	0.906878	0.692498	0.734275	0.477333	0.972772	633.08
7	0.692247	0.834339	0.711845	0.948148	0.692318	0.811297	0.666667	0.945545	634.48
8	0.692002	0.811849	0.664009	0.949206	0.692109	0.799743	0.960000	0.650990	633.07
9	0.691717	0.864509	0.847380	0.880423	0.691807	0.781771	0.954667	0.621287	635.62
10	0.691312	0.842567	0.948747	0.743915	0.691454	0.836970	0.909333	0.769802	635.40
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:13	0.691052		0.863636		0.970588		0.815789		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.697249	0.481711	0.983269	0.015550	0.692835	0.481910	1	0	568.17
2	0.692481	0.599504	0.985843	0.240431	0.692333	0.817656	0.942943	0.701117	565.83
3	0.691957	0.731556	0.504505	0.942584	0.691980	0.767004	0.975976	0.572626	566.39
4	0.691412	0.832610	0.877735	0.790670	0.691383	0.743850	0.495495	0.974860	565.48
5	0.690677	0.781773	0.607465	0.943780	0.690703	0.780029	0.567568	0.977654	563.89
6	0.690012	0.796032	0.621622	0.958134	0.690103	0.755427	0.510511	0.983240	565.38
7	0.689040	0.678859	0.348777	0.985646	0.689457	0.849494	0.864865	0.835196	563.82
8	0.687983	0.810291	0.633205	0.974880	0.688248	0.795948	0.603604	0.974860	565.28
9	0.686559	0.835710	0.705277	0.956938	0.686987	0.800289	0.624625	0.963687	566.39
10	0.684946	0.850589	0.750322	0.943780	0.685690	0.722142	0.447447	0.977654	565.31
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:04	0.684959		0.745098		0.418079		0.995671		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694601	0.456538	0.949126	0.037838	0.693482	0.459829	1	0	484.55
2	0.693078	0.529584	0.656598	0.421622	0.692402	0.540171	0	1	480.51
3	0.691922	0.540541	0	1	0.691414	0.540171	0	1	479.91
4	0.691010	0.540541	0	1	0.690618	0.540171	0	1	479.99
5	0.690204	0.540541	0	1	0.689833	0.540171	0	1	479.48
6	0.689271	0.540541	0	1	0.688892	0.540171	0	1	481.69
7	0.688410	0.540541	0	1	0.688171	0.540171	0	1	478.73
8	0.687522	0.540541	0	1	0.687178	0.540171	0	1	479.57
9	0.686499	0.540541	0	1	0.686485	0.540171	0	1	478.49
10	0.685595	0.540541	0	1	0.685654	0.540171	0	1	481.59
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:24:16	0.691297		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693737	0.518124	0.014970	0.972936	0.692778	0.524917	0	1	499.18
2	0.692734	0.525231	0	1	0.692644	0.524917	0	1	493.07
3	0.692601	0.525231	0	1	0.692492	0.524917	0	1	494.20
4	0.692429	0.525231	0	1	0.692304	0.524917	0	1	494.52
5	0.692253	0.525231	0	1	0.692117	0.524917	0	1	494.76
6	0.692029	0.525231	0	1	0.691859	0.524917	0	1	493.49
7	0.691767	0.525231	0	1	0.691587	0.524917	0	1	492.50
8	0.691472	0.525231	0	1	0.691259	0.524917	0	1	492.10
9	0.691206	0.525231	0	1	0.690890	0.524917	0	1	492.90
10	0.690872	0.525231	0	1	0.690471	0.524917	0	1	494.19
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:18	0.690560		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694166	0.536498	0.046776	0.969799	0.692958	0.531207	0	1	594.91
2	0.692918	0.530564	0	1	0.692856	0.531207	0	1	591.29
3	0.692808	0.530564	0	1	0.692723	0.531207	0	1	593.90
4	0.692663	0.530564	0	1	0.692571	0.531207	0	1	591.83
5	0.692484	0.530564	0	1	0.692359	0.531207	0	1	589.29
6	0.692233	0.530564	0	1	0.692075	0.531207	0	1	590.90
7	0.691935	0.530564	0	1	0.691744	0.531207	0	1	587.83
8	0.691577	0.530564	0	1	0.691358	0.531207	0	1	590.10
9	0.691205	0.530564	0	1	0.690981	0.531207	0	1	589.19
10	0.690613	0.530564	0	1	0.690333	0.531207	0	1	591.28
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:25	0.693479		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692578	0.486875	0.968586	0.046651	0.693146	0.477372	1	0	563.94
2	0.693134	0.509375	0.103403	0.880383	0.693122	0.522628	0	1	559.58
3	0.693116	0.522500	0	1	0.693101	0.522628	0	1	558.02
4	0.693093	0.522500	0	1	0.693081	0.522628	0	1	557.06
5	0.693076	0.522500	0	1	0.693062	0.522628	0	1	555.93
6	0.693054	0.522500	0	1	0.693044	0.522628	0	1	557.10
7	0.693037	0.522500	0	1	0.693024	0.522628	0	1	559.22
8	0.693015	0.522500	0	1	0.693002	0.522628	0	1	559.31
9	0.692991	0.522500	0	1	0.692976	0.522628	0	1	559.58
10	0.692966	0.522500	0	1	0.692949	0.522628	0	1	560.81
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:14	0.692809		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:35	0.690692	0.607750	0.231445	0.968577

D.1.18 Ftrl softmax sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.700613	0.482721	1	0.002116	0.693110	0.498074	0.994667	0.037129	635.65
2	0.693017	0.518925	0.071754	0.934392	0.692742	0.518614	0	1	634.31
3	0.692723	0.518376	0	1	0.692473	0.518614	0	1	633.62
4	0.692428	0.518376	0	1	0.692305	0.518614	0	1	635.18
5	0.692317	0.518376	0	1	0.692187	0.518614	0	1	631.97
6	0.692148	0.518376	0	1	0.692115	0.518614	0	1	629.93
7	0.692046	0.518376	0	1	0.692011	0.518614	0	1	632.98
8	0.691966	0.518376	0	1	0.691936	0.518614	0	1	635.40
9	0.691917	0.518376	0	1	0.691841	0.518614	0	1	634.18
10	0.691764	0.518376	0	1	0.691791	0.518614	0	1	631.72
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:52	0.679590		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694708	0.522009	0.045045	0.965311	0.692818	0.518090	0	1	567.26
2	0.692829	0.518289	0	1	0.692635	0.518090	0	1	560.62
3	0.692695	0.518289	0	1	0.692555	0.518090	0	1	562.82
4	0.692619	0.518289	0	1	0.692527	0.518090	0	1	563.36
5	0.692627	0.518289	0	1	0.692507	0.518090	0	1	565.40
6	0.692540	0.518289	0	1	0.692492	0.518090	0	1	562.51
7	0.692556	0.518289	0	1	0.692469	0.518090	0	1	562.89
8	0.692559	0.518289	0	1	0.692463	0.518090	0	1	563.89
9	0.692553	0.518289	0	1	0.692449	0.518090	0	1	564.78
10	0.692475	0.518289	0	1	0.692442	0.518090	0	1	561.81
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:39	0.689035		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.697950	0.514974	0.246423	0.743243	0.692084	0.540171	0	1	482.79
2	0.691717	0.540541	0	1	0.691230	0.540171	0	1	479.81
3	0.691008	0.540541	0	1	0.690617	0.540171	0	1	479.30
4	0.690481	0.540541	0	1	0.690295	0.540171	0	1	479.75
5	0.690269	0.540541	0	1	0.690097	0.540171	0	1	479.03
6	0.690048	0.540541	0	1	0.689954	0.540171	0	1	477.19
7	0.689970	0.540541	0	1	0.689868	0.540171	0	1	481.78
8	0.689831	0.540541	0	1	0.689832	0.540171	0	1	479.20
9	0.689792	0.540541	0	1	0.689783	0.540171	0	1	478.80
10	0.689740	0.540541	0	1	0.689746	0.540171	0	1	479.79
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:10	0.697900		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.703416	0.484009	0.824850	0.175913	0.693015	0.524917	0	1	496.65
2	0.692817	0.525231	0	1	0.692487	0.524917	0	1	492.81
3	0.692410	0.525231	0	1	0.692160	0.524917	0	1	492.88
4	0.692113	0.525231	0	1	0.691948	0.524917	0	1	490.08
5	0.691956	0.525231	0	1	0.691808	0.524917	0	1	494.54
6	0.691820	0.525231	0	1	0.691710	0.524917	0	1	494.49
7	0.691744	0.525231	0	1	0.691628	0.524917	0	1	493.01
8	0.691652	0.525231	0	1	0.691563	0.524917	0	1	490.14
9	0.691638	0.525231	0	1	0.691514	0.524917	0	1	491.99
10	0.691543	0.525231	0	1	0.691480	0.524917	0	1	492.40
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:29	0.691322		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694591	0.530564	0	1	0.692133	0.531207	0	1	592.32
2	0.691985	0.530564	0	1	0.691739	0.531207	0	1	587.60
3	0.691758	0.530564	0	1	0.691440	0.531207	0	1	591.08
4	0.691539	0.530564	0	1	0.691301	0.531207	0	1	588.59
5	0.691403	0.530564	0	1	0.691224	0.531207	0	1	590.11
6	0.691316	0.530564	0	1	0.691168	0.531207	0	1	591.77
7	0.691306	0.530564	0	1	0.691146	0.531207	0	1	589.92
8	0.691243	0.530564	0	1	0.691101	0.531207	0	1	591.09
9	0.691239	0.530564	0	1	0.691088	0.531207	0	1	590.29
10	0.691229	0.530564	0	1	0.691073	0.531207	0	1	587.68
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:44	0.696735		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692701	0.522500	0	1	0.692700	0.522628	0	1	560.69
2	0.692681	0.522500	0	1	0.692437	0.522628	0	1	557.09
3	0.692460	0.522500	0	1	0.692286	0.522628	0	1	558.00
4	0.692343	0.522500	0	1	0.692201	0.522628	0	1	556.38
5	0.692232	0.522500	0	1	0.692153	0.522628	0	1	559.11
6	0.692190	0.522500	0	1	0.692126	0.522628	0	1	559.30
7	0.692246	0.522500	0	1	0.692113	0.522628	0	1	561.30
8	0.692184	0.522500	0	1	0.692094	0.522628	0	1	561.13
9	0.692175	0.522500	0	1	0.692088	0.522628	0	1	559.34
10	0.692180	0.522500	0	1	0.692084	0.522628	0	1	560.33
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:12	0.690497		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:41	0.690847	0.549142	0	1

D.1.19 Ftrl softmax softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693141	0.516182	0.089977	0.912169	0.693131	0.518614	0	1	622.64
2	0.693127	0.518376	0	1	0.693117	0.518614	0	1	620.48
3	0.693116	0.518376	0	1	0.693109	0.518614	0	1	619.20
4	0.693106	0.518376	0	1	0.693098	0.518614	0	1	621.30
5	0.693096	0.518376	0	1	0.693089	0.518614	0	1	618.98
6	0.693087	0.518376	0	1	0.693080	0.518614	0	1	619.90
7	0.693078	0.518376	0	1	0.693071	0.518614	0	1	617.49
8	0.693071	0.518376	0	1	0.693064	0.518614	0	1	619.88
9	0.693063	0.518376	0	1	0.693057	0.518614	0	1	616.70
10	0.693056	0.518376	0	1	0.693050	0.518614	0	1	619.09
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:43:54	0.692117		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693146	0.506510	0.059202	0.922249	0.693131	0.518090	0	1	555.80
2	0.693126	0.518289	0	1	0.693118	0.518090	0	1	551.10
3	0.693116	0.518289	0	1	0.693111	0.518090	0	1	552.90
4	0.693107	0.518289	0	1	0.693101	0.518090	0	1	550.78
5	0.693098	0.518289	0	1	0.693092	0.518090	0	1	552.49
6	0.693089	0.518289	0	1	0.693083	0.518090	0	1	552.70
7	0.693080	0.518289	0	1	0.693075	0.518090	0	1	554.47
8	0.693075	0.518289	0	1	0.693070	0.518090	0	1	553.01
9	0.693069	0.518289	0	1	0.693061	0.518090	0	1	551.37
10	0.693058	0.518289	0	1	0.693054	0.518090	0	1	551.90
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:33:52	0.692798		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693155	0.534697	0.039746	0.955405	0.693090	0.540171	0	1	471.93
2	0.693071	0.540541	0	1	0.693042	0.540171	0	1	468.69
3	0.693026	0.540541	0	1	0.693005	0.540171	0	1	470.99
4	0.692989	0.540541	0	1	0.692968	0.540171	0	1	470.91
5	0.692954	0.540541	0	1	0.692938	0.540171	0	1	468.89
6	0.692924	0.540541	0	1	0.692909	0.540171	0	1	470.48
7	0.692894	0.540541	0	1	0.692879	0.540171	0	1	472.10
8	0.692864	0.540541	0	1	0.692849	0.540171	0	1	470.48
9	0.692835	0.540541	0	1	0.692824	0.540171	0	1	470.30
10	0.692809	0.540541	0	1	0.692798	0.540171	0	1	472.80
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:21:35	0.693275		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693140	0.523099	0.002994	0.993234	0.693127	0.524917	0	1	490.63
2	0.693120	0.525231	0	1	0.693108	0.524917	0	1	484.81
3	0.693101	0.525231	0	1	0.693091	0.524917	0	1	484.38
4	0.693086	0.525231	0	1	0.693076	0.524917	0	1	484.31
5	0.693070	0.525231	0	1	0.693061	0.524917	0	1	484.27
6	0.693057	0.525231	0	1	0.693048	0.524917	0	1	485.61
7	0.693042	0.525231	0	1	0.693036	0.524917	0	1	485.10
8	0.693031	0.525231	0	1	0.693024	0.524917	0	1	485.89
9	0.693018	0.525231	0	1	0.693012	0.524917	0	1	485.69
10	0.693007	0.525231	0	1	0.693001	0.524917	0	1	484.40
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:51	0.692993		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693128	0.530564	0	1	0.693103	0.531207	0	1	584.15
2	0.693091	0.530564	0	1	0.693073	0.531207	0	1	582.30
3	0.693062	0.530564	0	1	0.693043	0.531207	0	1	581.89
4	0.693034	0.530564	0	1	0.693016	0.531207	0	1	584.10
5	0.693011	0.530564	0	1	0.692993	0.531207	0	1	582.79
6	0.692988	0.530564	0	1	0.692973	0.531207	0	1	580.90
7	0.692968	0.530564	0	1	0.692951	0.531207	0	1	583.08
8	0.692950	0.530564	0	1	0.692934	0.531207	0	1	583.51
9	0.692931	0.530564	0	1	0.692916	0.531207	0	1	583.48
10	0.692913	0.530564	0	1	0.692896	0.531207	0	1	585.00
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:38:33	0.693292		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693126	0.520625	0.017016	0.980861	0.693128	0.522628	0	1	557.40
2	0.693122	0.522500	0	1	0.693110	0.522628	0	1	547.80
3	0.693105	0.522500	0	1	0.693095	0.522628	0	1	547.80
4	0.693091	0.522500	0	1	0.693081	0.522628	0	1	548.98
5	0.693076	0.522500	0	1	0.693068	0.522628	0	1	547.39
6	0.693063	0.522500	0	1	0.693055	0.522628	0	1	548.70
7	0.693052	0.522500	0	1	0.693043	0.522628	0	1	548.48
8	0.693039	0.522500	0	1	0.693032	0.522628	0	1	548.49
9	0.693030	0.522500	0	1	0.693021	0.522628	0	1	548.06
10	0.693019	0.522500	0	1	0.693012	0.522628	0	1	548.29
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:33:21	0.692898		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:32:31	0.692895	0.549142	0	1

D.1.20 Ftrl softmax tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694113	0.484366	0.966970	0.035979	0.693184	0.481386	1	0	632.68
2	0.693117	0.481624	1	0	0.693068	0.481386	1	0	622.09
3	0.693012	0.531541	0.936219	0.155556	0.692969	0.518614	0	1	620.21
4	0.692898	0.518376	0	1	0.692855	0.518614	0	1	621.96
5	0.692752	0.518376	0	1	0.692725	0.518614	0	1	622.99
6	0.692570	0.518376	0	1	0.692537	0.518614	0	1	621.41
7	0.692375	0.518376	0	1	0.692332	0.518614	0	1	621.49
8	0.692080	0.518376	0	1	0.692071	0.518614	0	1	619.88
9	0.691774	0.518376	0	1	0.691725	0.518614	0	1	620.20
10	0.691338	0.518376	0	1	0.691309	0.518614	0	1	620.39
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:44:22	0.685221		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694920	0.517049	0.036036	0.964115	0.692609	0.518090	0	1	558.26
2	0.692530	0.518289	0	1	0.692380	0.518090	0	1	549.51
3	0.692233	0.518289	0	1	0.692142	0.518090	0	1	550.41
4	0.691961	0.518289	0	1	0.691863	0.518090	0	1	550.77
5	0.691552	0.518289	0	1	0.691532	0.518090	0	1	552.41
6	0.691154	0.518289	0	1	0.691197	0.518090	0	1	551.47
7	0.690660	0.518289	0	1	0.690587	0.518090	0	1	550.61
8	0.690066	0.518289	0	1	0.689918	0.518090	0	1	551.69
9	0.689244	0.518289	0	1	0.689357	0.518090	0	1	552.69
10	0.688212	0.518289	0	1	0.688216	0.518090	0	1	551.61
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:33:48	0.685416		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694639	0.543462	0.039746	0.971622	0.693019	0.540171	0	1	471.67
2	0.692970	0.540541	0	1	0.692906	0.540171	0	1	469.09
3	0.692844	0.540541	0	1	0.692770	0.540171	0	1	467.49
4	0.692714	0.540541	0	1	0.692645	0.540171	0	1	468.80
5	0.692570	0.540541	0	1	0.692480	0.540171	0	1	469.60
6	0.692370	0.540541	0	1	0.692248	0.540171	0	1	469.30
7	0.692103	0.540541	0	1	0.691957	0.540171	0	1	469.48
8	0.691785	0.540541	0	1	0.691596	0.540171	0	1	470.30
9	0.691420	0.540541	0	1	0.691228	0.540171	0	1	470.30
10	0.690996	0.540541	0	1	0.690706	0.540171	0	1	469.90
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:21:56	0.693505		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694512	0.524520	0.032934	0.968877	0.692866	0.524917	0	1	488.68
2	0.692828	0.525231	0	1	0.692743	0.524917	0	1	485.70
3	0.692695	0.525231	0	1	0.692593	0.524917	0	1	483.30
4	0.692533	0.525231	0	1	0.692438	0.524917	0	1	483.41
5	0.692374	0.525231	0	1	0.692238	0.524917	0	1	482.20
6	0.692176	0.525231	0	1	0.692020	0.524917	0	1	485.50
7	0.691981	0.525231	0	1	0.691820	0.524917	0	1	484.59
8	0.691757	0.525231	0	1	0.691494	0.524917	0	1	485.10
9	0.691505	0.525231	0	1	0.691248	0.524917	0	1	485.80
10	0.691210	0.525231	0	1	0.690904	0.524917	0	1	487.91
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:24:06	0.690874		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694717	0.529377	0.034134	0.967562	0.692619	0.531207	0	1	585.52
2	0.692537	0.530564	0	1	0.692310	0.531207	0	1	579.68
3	0.692209	0.530564	0	1	0.691963	0.531207	0	1	581.68
4	0.691872	0.530564	0	1	0.691580	0.531207	0	1	584.29
5	0.691446	0.530564	0	1	0.691167	0.531207	0	1	579.61
6	0.690972	0.530564	0	1	0.690756	0.531207	0	1	581.49
7	0.690588	0.530564	0	1	0.690189	0.531207	0	1	583.79
8	0.689986	0.530564	0	1	0.689702	0.531207	0	1	581.18
9	0.689386	0.530564	0	1	0.689051	0.531207	0	1	584.80
10	0.688681	0.530564	0	1	0.688451	0.531207	0	1	582.90
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:38:26	0.692861		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.694075	0.524375	0.041885	0.965311	0.692887	0.522628	0	1	558.02
2	0.692843	0.522500	0	1	0.692755	0.522628	0	1	547.79
3	0.692728	0.522500	0	1	0.692629	0.522628	0	1	548.79
4	0.692583	0.522500	0	1	0.692462	0.522628	0	1	547.68
5	0.692410	0.522500	0	1	0.692293	0.522628	0	1	546.91
6	0.692196	0.522500	0	1	0.692065	0.522628	0	1	550.28
7	0.691964	0.522500	0	1	0.691850	0.522628	0	1	547.99
8	0.691697	0.522500	0	1	0.691561	0.522628	0	1	548.31
9	0.691380	0.522500	0	1	0.691159	0.522628	0	1	547.66
10	0.691043	0.522500	0	1	0.690837	0.522628	0	1	545.82
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:33:18	0.689663		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:32:40	0.689590	0.549142	0	1

D.1.21 Ftrl tanh ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.360243	0.481624	0.394077	0	4.329015	0.481386	0.002667	0	635.25
2	4.313348	0.481624	0.427107	0	4.313512	0.481386	0.685333	0	630.98
3	4.293345	0.481624	0.620729	0	4.295588	0.481386	0.602667	0	631.72
4	4.274024	0.481624	0.685649	0	4.278212	0.481386	0.442667	0	632.15
5	4.254036	0.481624	0.699317	0	4.260524	0.481386	0.677333	0	634.44
6	4.233931	0.481624	0.761959	0	4.248369	0.481386	0.773333	0	633.53
7	4.222883	0.481624	0.782460	0	4.236049	0.481386	0.794667	0	633.53
8	4.203344	0.481624	0.837130	0	4.232119	0.481386	0.818667	0	631.07
9	4.193788	0.484366	0.838269	0	4.225477	0.481386	0.768000	0	632.30
10	4.180093	0.481624	0.865604	0	4.213272	0.481386	0.858667	0	629.07
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:48:32	5.501236		0.309091		0.970588		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.897477	0.512089	0.036036	0.953349	0.651308	0.518090	0	0.958101	566.16
2	0.608764	0.692498	0	0.937799	0.604425	0.798842	0	0.824022	562.98
3	0.552343	0.885307	0.015444	0.860048	0.559431	0.810420	0.003003	0.952514	562.50
4	0.506363	0.882207	0.314028	0.843301	0.544542	0.778582	0.618619	0.991620	562.80
5	0.461096	0.904526	0.574003	0.880383	0.476968	0.881331	0.435435	0.882682	562.25
6	0.420227	0.913825	0.644788	0.900718	0.462217	0.878437	0.627628	0.815642	563.79
7	0.387119	0.918785	0.723295	0.907895	0.436911	0.872648	0.843844	0.821229	565.51
8	0.374097	0.918785	0.783784	0.924641	0.422587	0.866860	0.843844	0.826816	561.72
9	0.345384	0.928084	0.822394	0.919856	0.407919	0.872648	0.684685	0.941341	563.18
10	0.339953	0.916305	0.818533	0.928230	0.405183	0.875543	0.762763	0.896648	563.09
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:40	0.405642		0.872549		0.683616		0.883117		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.532340	0.459459	0.467409	0	4.472592	0.459829	0.769517	0	482.71
2	4.451679	0.459459	0.543720	0	4.442337	0.461538	0.107807	0	478.77
3	4.419669	0.468225	0.605723	0	4.419651	0.464957	0.947955	0	477.80
4	4.395176	0.508400	0.745628	0	4.390584	0.502564	0.791822	0	477.09
5	4.374373	0.546384	0.764706	0	4.383843	0.514530	0.873606	0	477.90
6	4.364897	0.585829	0.815580	0	4.366809	0.533333	0.884758	0	479.53
7	4.362978	0.588020	0.850556	0	4.362521	0.553846	0.884758	0	480.25
8	4.343068	0.607743	0.871224	0	4.356937	0.591453	0.821561	0	477.93
9	4.335041	0.636961	0.850556	0	4.347540	0.601709	0.903346	0	478.48
10	4.329929	0.656684	0.885533	0	4.346220	0.613675	0.832714	0	478.40
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:46	3.932103		0.679420		0.874036		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.092366	0.599858	0.769461	0.449256	0.594719	0.855482	0.786713	0.737342	496.55
2	0.574784	0.800284	0.764970	0.650880	0.523394	0.887043	0.772727	0.886076	491.90
3	0.505808	0.865672	0.817365	0.719892	0.458551	0.915282	0.881119	0.794304	490.79
4	0.457550	0.872779	0.842814	0.741543	0.415740	0.911960	0.930070	0.778481	491.22
5	0.414296	0.884151	0.884731	0.772666	0.505621	0.720930	1	0.696203	492.05
6	0.389770	0.896944	0.917665	0.815968	0.363910	0.913621	0.923077	0.797468	492.00
7	0.356558	0.904051	0.892216	0.840325	0.345343	0.911960	0.923077	0.851266	494.46
8	0.332252	0.915423	0.913174	0.871448	0.303597	0.930233	0.940559	0.879747	494.53
9	0.315463	0.901919	0.904192	0.863329	0.284269	0.928571	0.923077	0.901899	492.99
10	0.300726	0.913291	0.914671	0.883627	0.313077	0.905316	0.888112	0.905063	494.29
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:10	0.283471		0.923186		0.909910		0.900000		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.000654	0.530564	0	0.814318	3.899395	0.531207	0	0.929504	590.78
2	3.873420	0.530564	0	0.824385	3.860260	0.531207	0	0.642298	591.50
3	3.833136	0.530564	0	0.850112	3.832562	0.531207	0	0.757180	590.19
4	3.810052	0.530564	0	0.871365	3.807459	0.531207	0	0.892950	590.21
5	3.789895	0.575668	0	0.892617	3.793776	0.585298	0	0.879896	589.57
6	3.777029	0.645697	0	0.895973	3.793205	0.654646	0	0.835509	592.29
7	3.765996	0.684866	0	0.914989	3.798011	0.683773	0	0.845953	588.02
8	3.761385	0.694955	0	0.911633	3.777169	0.667129	0	0.898172	586.89
9	3.752421	0.716914	0	0.921700	3.779274	0.656033	0	0.919060	589.56
10	3.744194	0.734718	0	0.934004	3.791543	0.693481	0	0.890339	586.48
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:46	4.119604		0.653595		0		0.959459		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.608100	0.480000	0	0.028708	4.450335	0.477372	0	0	565.57
2	4.409847	0.477500	0	0	4.390339	0.477372	0	0	559.80
3	4.379716	0.477500	0	0	4.378288	0.477372	0	0	557.51
4	4.373284	0.477500	0	0	4.375209	0.477372	0	0	559.03
5	4.370981	0.477500	0.002618	0	4.373271	0.477372	0	0	558.17
6	4.369464	0.477500	0.018325	0	4.372460	0.477372	0	0	560.48
7	4.368529	0.477500	0.023560	0	4.371426	0.477372	0.030581	0	556.22
8	4.367444	0.477500	0.094241	0	4.370401	0.477372	0.051988	0	554.61
9	4.366254	0.477500	0.126963	0	4.369884	0.477372	0.079511	0	554.30
10	4.365202	0.477500	0.151832	0	4.369266	0.477372	0.088685	0	555.39
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:57	4.508118		0.459016		0.066327		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:39	3.125029	0.649476	0.584079	0.457096

D.1.22 Ftrl tanh sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.705514	0.518925	0	0.964021	2.495467	0.518614	0	1	635.78
2	2.199669	0.518376	0.025057	1	1.840639	0.518614	0.205333	1	630.14
3	1.455185	0.518376	0.264237	1	1.065484	0.518614	0.098667	1	631.26
4	0.817403	0.540867	0.193622	0.928042	0.746081	0.566110	0.304000	0.777228	632.33
5	0.692111	0.640702	0.346241	0.794709	0.689576	0.609756	0.490667	0.757426	634.38
6	0.668017	0.652770	0.430524	0.820106	0.651757	0.662388	0.581333	0.873762	633.19
7	0.607795	0.725727	0.536446	0.844444	0.644770	0.699615	0.445333	0.722772	634.01
8	0.589319	0.722984	0.529613	0.867725	0.612668	0.715019	0.426667	0.789604	632.69
9	0.576037	0.763028	0.561503	0.793651	0.590925	0.738126	0.456000	0.814356	631.84
10	0.540736	0.783324	0.562642	0.791534	0.604527	0.753530	0.554667	0.757426	629.50
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:48:33	0.587508		0.781818		0.941176		0.671053		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.900234	0.521389	0.037323	0.788278	0.673441	0.520984	0	0.837989	569.18
2	0.634469	0.711717	0	0.744019	0.623753	0.739508	0	0.832402	560.38
3	0.590132	0.864848	0	0.757177	0.592761	0.865412	0	0.712291	565.30
4	0.550622	0.907626	0	0.811005	0.569036	0.858177	0	0.734637	564.00
5	0.530323	0.898326	0	0.844498	0.549505	0.863965	0	0.863128	567.18
6	0.513698	0.907006	0	0.860048	0.544049	0.861071	0	0.765363	565.30
7	0.507683	0.902666	0	0.876794	0.524142	0.874095	0	0.837989	567.52
8	0.490805	0.910725	0	0.909091	0.529420	0.839363	0	0.863128	564.98
9	0.479796	0.917545	0.012870	0.893541	0.526454	0.871201	0.012012	0.818436	568.09
10	0.466678	0.928704	0.084942	0.917464	0.506128	0.888567	0.042042	0.854749	563.40
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:57	0.525301		0.830882		0.056497		0.796537		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.093738	0.547115	0.772655	0.041892	0.650161	0.801709	0.568773	0	482.28
2	0.627899	0.754565	0.542130	0.208108	0.612236	0.707692	0.405204	1	478.00
3	0.592312	0.782323	0.623211	1	0.589910	0.796581	0.643123	1	479.14
4	0.570924	0.811541	0.715421	1	0.570876	0.835897	0.773234	1	479.19
5	0.550908	0.835646	0.742448	1	0.554387	0.829060	0.795539	1	478.70
6	0.533179	0.861213	0.798092	1	0.548431	0.842735	0.773234	1	481.10
7	0.529095	0.863404	0.804452	1	0.538458	0.864957	0.862454	1	480.06
8	0.513843	0.878013	0.850556	1	0.522829	0.887179	0.888476	1	479.00
9	0.509444	0.893353	0.855326	1	0.531800	0.868376	0.914498	1	484.29
10	0.501273	0.883857	0.856916	1	0.520372	0.887179	0.888476	1	480.70
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:16	0.499251		0.903694		0.894602		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.005222	0.599858	0	0.787551	0.642800	0.848837	0	0.746835	498.40
2	0.629260	0.803838	0.196108	0.740189	0.619549	0.664452	0	0.971519	494.70
3	0.605056	0.808102	0.019461	0.775372	0.582006	0.860465	0.020979	0.753165	493.56
4	0.576846	0.839375	0.064371	0.802436	0.555581	0.832226	0.643357	0.626582	492.12
5	0.550249	0.869225	0.185629	0.807848	0.533104	0.845515	0.741259	0.674051	494.09
6	0.537678	0.856432	0.387725	0.792963	0.514235	0.887043	0.171329	0.901899	495.67
7	0.519185	0.864250	0.371257	0.836265	0.502974	0.905316	0.055944	0.927215	491.91
8	0.499186	0.884861	0.238024	0.852503	0.483563	0.905316	0.475524	0.867089	495.50
9	0.491440	0.885572	0.404192	0.838972	0.470136	0.918605	0.447552	0.889241	494.29
10	0.476288	0.884151	0.444611	0.866035	0.466130	0.901993	0.517483	0.854430	492.11
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:23	0.462531		0.910384		0.603604		0.832432		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.899435	0.645104	0.951960	0.894855	0.665804	0.682386	0.991124	0.684073	596.99
2	0.647664	0.767359	0.992415	0.807606	0.646977	0.791956	0.982249	0.950392	589.49
3	0.629694	0.807715	0.981037	0.823266	0.631973	0.765603	0.967456	0.715405	592.80
4	0.605714	0.858754	0.974716	0.851230	0.611859	0.793343	0.917160	0.665796	588.20
5	0.588535	0.847478	0.919090	0.796421	0.599167	0.812760	0.846154	0.731071	587.60
6	0.570623	0.874184	0.882427	0.832215	0.583766	0.851595	0.704142	0.806789	588.20
7	0.556404	0.867656	0.777497	0.826622	0.562864	0.869626	0.656805	0.801567	589.09
8	0.540524	0.877151	0.653603	0.826622	0.562245	0.848821	0.514793	0.806789	590.27
9	0.527152	0.886053	0.562579	0.841163	0.555091	0.843273	0.411243	0.746736	587.91
10	0.519032	0.894956	0.471555	0.853468	0.537910	0.871013	0.325444	0.851175	589.98
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:43	0.533198		0.846405		0.284810		0.905405		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.877959	0.475625	0.005236	0.010766	0.764322	0.477372	0	0	562.50
2	0.738609	0.477500	0	0	0.720149	0.477372	0	0	561.30
3	0.710945	0.477500	0	0	0.703898	0.477372	0	0	557.60
4	0.699439	0.477500	0	0	0.696331	0.477372	0	0	560.99
5	0.694067	0.491250	0	0	0.692713	0.477372	0	0	559.89
6	0.691170	0.553750	0	0	0.690616	0.728467	0	0	561.20
7	0.689346	0.563750	0	0	0.688931	0.535766	0	0	559.69
8	0.688168	0.562500	0	0.080144	0.688142	0.522628	0	0.298883	559.78
9	0.686850	0.528750	0.006545	0.669856	0.687302	0.522628	0	0.977654	559.90
10	0.686221	0.522500	0	0.997608	0.686386	0.522628	0	1	556.50
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:11	0.684233		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:50	0.548670	0.802361	0.463448	0.867571

D.1.23 Ftrl tanh softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.566857	0.513988	0	0	1.417746	0.518614	0	0	635.11
2	1.392497	0.518376	0	0	1.370914	0.518614	0	0	633.87
3	1.355082	0.518376	0	0	1.339471	0.518614	0	0	631.39
4	1.326973	0.518376	0	0	1.314950	0.518614	0	0	635.82
5	1.304697	0.518376	0	0	1.294657	0.518614	0	0	633.70
6	1.286048	0.518376	0	0	1.277203	0.518614	0	0	634.48
7	1.269660	0.518376	0	0	1.261882	0.518614	0	0	631.90
8	1.255124	0.518376	0	0	1.248182	0.518614	0	0	631.61
9	1.242030	0.518376	0	0	1.235814	0.518614	0	0	633.45
10	1.230333	0.518376	0	0	1.224503	0.518614	0	0	631.40
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:25	1.206267		0.690909		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.308082	0.514569	0	0	4.237564	0.518090	0	0	564.12
2	4.216500	0.518289	0	0	4.205305	0.518090	0	0	564.96
3	4.200152	0.518289	0	0	4.197086	0.518090	0	0	562.09
4	4.193824	0.518289	0	0	4.192920	0.518090	0	0	563.49
5	4.190181	0.518289	0	0	4.189542	0.518090	0	0	564.70
6	4.187163	0.518289	0	0	4.187057	0.518090	0	0	564.69
7	4.184658	0.518289	0	0	4.184765	0.518090	0	0	566.14
8	4.182523	0.518289	0	0	4.182805	0.518090	0	0	563.73
9	4.180643	0.518289	0	0	4.180977	0.518090	0	0	565.87
10	4.179025	0.518289	0	0	4.179426	0.518090	0	0	562.60
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:37:28	3.840774		0.566176		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.241523	0.540541	0	0	4.150172	0.540171	0	0	484.47
2	4.139745	0.540541	0	0	4.135744	0.540171	0	0	479.71
3	4.129167	0.540541	0	0	4.127334	0.540171	0	0	478.12
4	4.121655	0.540541	0	0	4.120662	0.540171	0	0	482.08
5	4.115579	0.540541	0	0	4.115173	0.540171	0	0	479.78
6	4.110296	0.540541	0	0	4.110226	0.540171	0	0	480.31
7	4.105573	0.540541	0	0	4.105801	0.540171	0	0	479.87
8	4.101403	0.540541	0	0	4.101790	0.540171	0	0	479.00
9	4.097564	0.540541	0	0	4.098154	0.540171	0	0	479.01
10	4.094060	0.540541	0	0	4.094756	0.540171	0	0	480.30
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:16	4.460258		0.486807		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.339826	0.526652	0	0	4.245661	0.524917	0	0	499.78
2	4.214508	0.525231	0	0	4.190954	0.524917	0	0	492.00
3	4.175890	0.525231	0	0	4.168633	0.524917	0	0	493.38
4	4.161200	0.525231	0	0	4.159142	0.524917	0	0	493.40
5	4.154373	0.525231	0	0	4.153784	0.524917	0	0	492.37
6	4.149848	0.525231	0	0	4.150262	0.524917	0	0	491.94
7	4.146534	0.525231	0	0	4.147275	0.524917	0	0	494.98
8	4.143777	0.525231	0	0	4.144765	0.524917	0	0	493.97
9	4.141483	0.525231	0	0	4.142589	0.524917	0	0	492.81
10	4.139398	0.525231	0	0	4.140686	0.524917	0	0	492.16
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:17	4.130743		0.526316		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.303030	0.525816	0	0	1.117082	0.531207	0	0	593.55
2	1.103927	0.530564	0	0	1.093496	0.531207	0	0	590.08
3	1.085848	0.530564	0	0	1.078375	0.531207	0	0	591.71
4	1.072502	0.530564	0	0	1.066548	0.531207	0	0	589.85
5	1.061670	0.530564	0	0	1.056534	0.531207	0	0	590.91
6	1.052225	0.530564	0	0	1.047816	0.531207	0	0	589.48
7	1.043955	0.530564	0	0	1.040062	0.531207	0	0	588.89
8	1.036533	0.530564	0	0	1.032976	0.531207	0	0	592.31
9	1.029848	0.530564	0	0	1.026445	0.531207	0	0	590.32
10	1.023566	0.530564	0	0	1.020416	0.531207	0	0	589.81
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:20	1.025225		0.483660		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.352381	0.477500	0	0	1.157174	0.477372	0	0	561.71
2	1.140887	0.477500	0	0	1.128098	0.477372	0	0	560.99
3	1.119768	0.477500	0	0	1.111983	0.477372	0	0	558.96
4	1.105825	0.477500	0	0	1.099828	0.477372	0	0	560.53
5	1.094776	0.477500	0	0	1.089727	0.477372	0	0	555.28
6	1.085414	0.477500	0	0	1.081017	0.477372	0	0	557.72
7	1.077160	0.477500	0	0	1.073224	0.477372	0	0	556.31
8	1.069771	0.477500	0	0	1.066171	0.477372	0	0	559.49
9	1.062994	0.477500	0	0	1.059754	0.477372	0	0	558.49
10	1.056806	0.477500	0	0	1.053774	0.477372	0	0	559.42
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:02	1.055173		0.459016		0		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:38	2.619740	0.535481	0	0

D.1.24 Ftrl tanh tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.712475	0.518925	0	0	7.712474	0.518614	0	0	637.79
2	7.712475	0.518376	0	0	7.712474	0.518614	0	0	633.19
3	7.712475	0.518376	0	0	7.712474	0.518614	0	0	635.82
4	7.712475	0.518376	0	0	7.712474	0.518614	0	0	633.28
5	7.712475	0.518376	0	0	7.712474	0.518614	0	0	634.60
6	7.712475	0.518376	0	0	7.712474	0.518614	0	0	634.50
7	7.712475	0.518376	0	0	7.712474	0.518614	0	0	634.37
8	7.712475	0.518376	0	0	7.712474	0.518614	0	0	633.78
9	7.712475	0.518376	0	0	7.712474	0.518614	0	0	636.00
10	7.712475	0.518376	0	0	7.712474	0.518614	0	0	634.70
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:29	7.712474		0.690909		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.072675	0.519529	0	0.709330	4.026358	0.518090	0	0.991620	567.44
2	3.994797	0.518289	0	0.739234	3.988652	0.518090	0	0.575419	565.53
3	3.951177	0.522629	0	0.766746	3.958828	0.520984	0	0.745810	565.98
4	3.919210	0.562926	0	0.839713	3.949788	0.575977	0	0.673184	563.63
5	4.382006	0.562926	0	0.913876	4.652033	0.518090	0	1	565.00
6	4.602137	0.518289	0	1	4.553744	0.518090	0	1	565.69
7	4.509349	0.518289	0	1	4.467811	0.518090	0	1	562.78
8	4.427100	0.518289	0	1	4.389880	0.518090	0	1	565.12
9	4.353435	0.518289	0	1	4.318809	0.518090	0	1	564.18
10	4.279951	0.518289	0	1	4.240287	0.518090	0	1	561.82
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:40	3.826760		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.051983	0.539080	0	0.752703	3.882253	0.540171	0	1	484.10
2	3.875337	0.540541	0	0.920270	3.872542	0.540171	0	0.971519	479.22
3	3.862124	0.540541	0	0.885135	3.859702	0.540171	0	0.974684	478.01
4	3.848631	0.540541	0	0.835135	3.848219	0.540171	0	0.800633	481.55
5	3.832479	0.540541	0	0.820270	3.831145	0.540171	0	0.822785	479.40
6	3.813123	0.540541	0	0.809459	3.817879	0.540171	0	0.642405	479.53
7	3.795142	0.541271	0	0.814865	3.806665	0.541880	0	0.553797	481.02
8	3.778031	0.550037	0	0.778378	3.783035	0.541880	0	0.873418	478.99
9	3.761254	0.568298	0	0.832432	3.770595	0.552137	0	0.803797	481.09
10	3.742433	0.590942	0	0.847297	3.768700	0.543590	0	0.955696	480.79
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:57	4.166793		0.489446		0		0.964770		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.411794	0.478323	0.031437	0.043302	7.712474	0.475083	0	0	499.09
2	7.712474	0.474769	0	0	7.712474	0.475083	0	0	494.09
3	7.712474	0.474769	0	0	7.712474	0.475083	0	0	492.21
4	7.712474	0.474769	0	0	7.712474	0.475083	0	0	494.20
5	7.712474	0.474769	0	0	7.712474	0.475083	0	0	494.58
6	7.712474	0.474769	0	0	7.712474	0.475083	0	0	495.82
7	7.712474	0.474769	0	0	7.712474	0.475083	0	0	493.94
8	7.712474	0.474769	0	0	7.712474	0.475083	0	0	494.22
9	7.712474	0.474769	0	0	7.712474	0.475083	0	0	490.38
10	7.712474	0.474769	0	0	7.712474	0.475083	0	0	491.64
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:20	7.712474		0.473684		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.225183	0.529377	0	0.958613	7.189928	0.531207	0	1	593.12
2	7.199788	0.530564	0	1	7.189927	0.531207	0	1	590.16
3	7.199787	0.530564	0	1	7.189927	0.531207	0	1	589.26
4	7.199787	0.530564	0	1	7.189928	0.531207	0	1	589.70
5	7.199788	0.530564	0	1	7.189927	0.531207	0	1	588.88
6	7.199787	0.530564	0	1	7.189927	0.531207	0	1	589.48
7	7.199787	0.530564	0	1	7.189927	0.531207	0	1	590.32
8	7.199787	0.530564	0	1	7.189927	0.531207	0	1	589.98
9	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.34
10	7.199787	0.530564	0	1	7.189927	0.531207	0	1	589.30
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:44	7.919153		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.878103	0.530625	0.037958	0.772727	3.963932	0.522628	0	0.835196	563.27
2	3.934111	0.526250	0	0.769139	3.924875	0.522628	0	0.916201	557.39
3	3.746410	0.585000	0	0.806220	0.732157	0.524088	0	0.997207	558.07
4	0.556333	0.823750	0	0.845694	0.538589	0.836496	0	0.879888	558.50
5	0.498109	0.888125	0	0.882775	0.510146	0.843796	0	0.916201	558.00
6	0.464264	0.891250	0.073298	0.894737	0.482029	0.884672	0.113150	0.868715	558.49
7	0.456129	0.880000	0.336387	0.921053	0.459039	0.896350	0.446483	0.879888	558.29
8	0.419597	0.898125	0.500000	0.915072	0.525824	0.867153	0.648318	0.754190	558.09
9	0.409405	0.912500	0.579843	0.919856	0.442448	0.851095	0.657492	0.815642	558.27
10	0.363696	0.915000	0.667539	0.931818	0.410234	0.871533	0.553517	0.932961	559.00
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:01	0.386167		0.857143		0.428571		0.982684		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:12	5.287304	0.593503	0.071429	0.657909

D.1.25 SGD sigmoid ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.612916	0.647833	0.589977	0.697354	0.545362	0.732991	0.984000	0.700495	638.07
2	0.428177	0.813494	0.779043	0.832804	0.403404	0.840822	0.925333	0.792079	634.69
3	0.317818	0.872737	0.879271	0.871958	0.371271	0.851091	0.744000	0.915842	637.17
4	0.272840	0.898519	0.894077	0.892063	0.339171	0.865212	0.757333	0.918317	636.99
5	0.239369	0.918267	0.914579	0.919577	0.313478	0.871630	0.765333	0.950495	634.61
6	0.228034	0.913878	0.903189	0.918519	0.291558	0.880616	0.842667	0.903465	636.88
7	0.207572	0.924849	0.921412	0.919577	0.241960	0.916560	0.922667	0.913366	637.92
8	0.198820	0.934174	0.938497	0.933333	0.359364	0.849807	0.720000	0.955446	635.57
9	0.198602	0.919912	0.911162	0.930159	0.270035	0.883184	0.853333	0.908416	635.76
10	0.167999	0.937466	0.937358	0.936508	0.307127	0.867779	0.765333	0.955446	636.01
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:46	0.096770		0.981818		0.970588		0.986842		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.614362	0.709857	0.652510	0.721292	0.522063	0.848046	0.765766	0.745810	574.66
2	0.405212	0.889027	0.868726	0.862440	0.392477	0.855282	0.951952	0.818436	569.20
3	0.317976	0.890887	0.895753	0.886364	0.335307	0.901592	0.846847	0.905028	567.88
4	0.251610	0.924985	0.918919	0.912679	0.333500	0.866860	0.807808	0.938547	567.22
5	0.219416	0.938004	0.930502	0.936603	0.307299	0.888567	0.861862	0.913408	565.66
6	0.212606	0.929324	0.917632	0.927033	0.295607	0.901592	0.951952	0.857542	566.92
7	0.215607	0.914445	0.924067	0.900718	0.312024	0.868307	0.936937	0.798883	565.30
8	0.180932	0.936764	0.934363	0.937799	0.268657	0.904486	0.888889	0.905028	568.49
9	0.163081	0.942343	0.931789	0.950957	0.280683	0.897250	0.873874	0.921788	565.22
10	0.171410	0.933664	0.930502	0.943780	0.270017	0.882779	0.867868	0.913408	566.10
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:37:19	0.244908		0.901961		0.915254		0.900433		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.643554	0.689554	0.526232	0.804054	0.554833	0.793162	0.565056	0.892405	485.10
2	0.476092	0.831994	0.742448	0.877027	0.433756	0.868376	0.862454	0.857595	482.79
3	0.387560	0.863404	0.853736	0.872973	0.387369	0.871795	0.825279	0.939873	481.11
4	0.331380	0.883126	0.855326	0.898649	0.368298	0.858120	0.892193	0.768987	481.58
5	0.279210	0.912345	0.898251	0.909459	0.352853	0.859829	0.899628	0.822785	480.29
6	0.269211	0.906501	0.895072	0.908108	0.305978	0.887179	0.881041	0.882911	480.90
7	0.269168	0.896275	0.861685	0.921622	0.324743	0.863248	0.795539	0.920886	480.87
8	0.235714	0.921841	0.912560	0.931081	0.304166	0.873504	0.944238	0.797468	480.62
9	0.229287	0.921110	0.918919	0.924324	0.284151	0.880342	0.866171	0.895570	482.89
10	0.221867	0.917458	0.899841	0.932432	0.259861	0.914530	0.903346	0.911392	481.79
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:32	0.202654		0.932718		0.930591		0.940379		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.655923	0.634684	0.582335	0.654939	0.559329	0.815615	0.727273	0.851266	500.90
2	0.515020	0.793888	0.758982	0.788904	0.420249	0.883721	0.905594	0.787975	495.89
3	0.430129	0.820185	0.787425	0.847091	0.354654	0.882060	0.916084	0.867089	495.01
4	0.336419	0.885572	0.893713	0.879567	0.296259	0.920266	0.898601	0.933544	495.78
5	0.301747	0.902630	0.902695	0.905277	0.265709	0.930233	0.930070	0.905063	496.52
6	0.283038	0.906183	0.913174	0.895805	0.282172	0.885382	0.765734	0.984177	495.28
7	0.273991	0.899787	0.887725	0.905277	0.259002	0.897010	0.979021	0.832278	495.60
8	0.241008	0.914712	0.908683	0.916103	0.221501	0.925249	0.888112	0.952532	494.57
9	0.213848	0.930348	0.925150	0.930988	0.196638	0.940199	0.937063	0.939873	494.42
10	0.212542	0.932480	0.934132	0.926928	0.203326	0.928571	0.867133	0.968354	493.78
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:44	0.204190		0.930299		0.876877		0.954054		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.645773	0.670623	0.571429	0.674497	0.543385	0.796117	0.931953	0.835509	597.30
2	0.447844	0.844510	0.829330	0.846756	0.428925	0.822469	0.680473	0.934726	593.49
3	0.346316	0.865282	0.847029	0.873602	0.380481	0.847434	0.952663	0.746736	591.69
4	0.278646	0.918694	0.902655	0.925056	0.321340	0.890430	0.923077	0.864230	591.62
5	0.266225	0.897329	0.888748	0.906040	0.310602	0.871013	0.908284	0.853786	593.40
6	0.228384	0.916914	0.912769	0.932886	0.290029	0.877947	0.955621	0.804178	591.40
7	0.222980	0.915727	0.914033	0.920582	0.269402	0.898752	0.920118	0.882507	592.11
8	0.205937	0.925816	0.924147	0.930649	0.286279	0.887656	0.855030	0.921671	592.31
9	0.201912	0.921662	0.907712	0.922819	0.258340	0.904300	0.840237	0.947781	591.49
10	0.197138	0.927596	0.920354	0.935123	0.250818	0.904300	0.949704	0.869452	593.69
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:14	0.229939		0.901961		0.905063		0.898649		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.604291	0.686875	0.642670	0.714115	0.496024	0.792701	0.908257	0.692737	565.16
2	0.394003	0.878750	0.873037	0.854067	0.363138	0.877372	0.911315	0.824022	562.68
3	0.325393	0.874375	0.870419	0.885167	0.338858	0.864234	0.862385	0.874302	561.71
4	0.265766	0.907500	0.892670	0.919856	0.300385	0.886131	0.929664	0.857542	561.29
5	0.277095	0.886250	0.874346	0.893541	0.290481	0.880292	0.935780	0.826816	559.36
6	0.217656	0.921250	0.920157	0.917464	0.274486	0.883212	0.914373	0.840782	553.18
7	0.210235	0.926875	0.920157	0.933014	0.263606	0.900730	0.914373	0.882682	550.81
8	0.210446	0.925625	0.920157	0.935407	0.277436	0.887591	0.776758	0.960894	553.59
9	0.182964	0.931250	0.924084	0.936603	0.246010	0.897810	0.877676	0.913408	552.18
10	0.178282	0.935000	0.928010	0.938995	0.253949	0.890511	0.914373	0.865922	551.59
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:43	0.185687		0.932084		0.928571		0.930736		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:53	0.194025	0.930140	0.921158	0.935182

D.1.26 SGD sigmoid sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.719143	0.517828	0.436219	0.627513	0.665876	0.527600	0.008000	0.975248	631.60
2	0.637372	0.742183	0.593394	0.851852	0.626415	0.771502	0.573333	0.844059	626.60
3	0.588329	0.839276	0.748292	0.910053	0.587294	0.795892	0.618667	0.952970	627.40
4	0.545509	0.860121	0.801822	0.903704	0.558596	0.818999	0.778667	0.853960	626.97
5	0.509940	0.873286	0.843964	0.897354	0.533217	0.825417	0.808000	0.836634	627.70
6	0.475111	0.890291	0.855353	0.908995	0.512425	0.829268	0.850667	0.784653	628.51
7	0.452983	0.872189	0.870159	0.878307	0.472002	0.875481	0.869333	0.878713	627.68
8	0.427505	0.883708	0.884966	0.881481	0.467840	0.833119	0.893333	0.789604	629.60
9	0.406769	0.884805	0.873576	0.898413	0.461089	0.830552	0.760000	0.903465	628.59
10	0.383816	0.891388	0.883827	0.893122	0.449292	0.838254	0.850667	0.821782	628.88
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:45:22	0.322302		0.936364		1		0.907895		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.701694	0.535028	0.483912	0.517943	0.688302	0.483357	0.981982	0	562.01
2	0.626906	0.724737	0.700129	0.740431	0.621695	0.767004	0.864865	0.628492	562.82
3	0.581796	0.760694	0.711712	0.794258	0.583499	0.821997	0.738739	0.868715	568.79
4	0.529206	0.897086	0.855856	0.911483	0.547433	0.849494	0.825826	0.851955	568.91
5	0.493794	0.882827	0.888031	0.876794	0.520035	0.861071	0.894895	0.815642	567.40
6	0.459287	0.887167	0.881596	0.887560	0.491026	0.837916	0.897898	0.801676	566.57
7	0.431690	0.901426	0.904762	0.903110	0.474025	0.858177	0.885886	0.826816	565.59
8	0.411302	0.898326	0.891892	0.906699	0.467712	0.827786	0.951952	0.706704	566.09
9	0.390698	0.902666	0.907336	0.894737	0.440835	0.858177	0.873874	0.843575	567.00
10	0.370279	0.911345	0.906049	0.900718	0.427514	0.859624	0.792793	0.907821	566.88
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:10	0.437351		0.833333		0.740113		0.917749		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.714367	0.545654	0.313196	0.647297	0.675662	0.588034	0.970260	1	485.22
2	0.653927	0.673484	0.451510	0.832432	0.633893	0.605128	0.327138	1	480.53
3	0.617106	0.749452	0.600954	0.881081	0.617097	0.565812	0.055762	1	483.48
4	0.585931	0.774288	0.570747	0.937838	0.576096	0.758974	0.494424	0.984177	481.21
5	0.552395	0.839299	0.693164	0.937838	0.557594	0.854701	0.888476	0.835443	483.20
6	0.525053	0.850256	0.764706	0.924324	0.528914	0.858120	0.936803	0.791139	481.98
7	0.505850	0.848064	0.825119	0.867568	0.519889	0.825641	0.947955	0.724684	481.90
8	0.475860	0.862673	0.791733	0.914865	0.492182	0.870085	0.884758	0.848101	480.69
9	0.465344	0.858291	0.815580	0.901351	0.491790	0.825641	0.907063	0.759494	481.50
10	0.441256	0.872169	0.844197	0.909459	0.471617	0.856410	0.869888	0.848101	481.60
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:56	0.450361		0.869393		0.840617		0.905149		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693200	0.519545	0.381737	0.687415	0.677986	0.518272	0.982517	0.253165	499.40
2	0.657787	0.636816	0.477545	0.792963	0.639218	0.689369	0.982517	0.408228	495.18
3	0.623166	0.699360	0.564371	0.809202	0.597424	0.835548	0.95105	0.655063	495.01
4	0.590794	0.796731	0.791916	0.805142	0.566854	0.852159	0.951049	0.721519	494.82
5	0.562030	0.808102	0.754491	0.838972	0.542303	0.853821	0.961538	0.743671	495.31
6	0.530041	0.841507	0.818862	0.868742	0.509520	0.892027	0.951049	0.851266	496.42
7	0.502098	0.862118	0.836826	0.889039	0.481020	0.887043	0.846154	0.905063	497.60
8	0.480134	0.880597	0.866766	0.883627	0.465759	0.892027	0.884615	0.882911	496.68
9	0.464008	0.865672	0.844311	0.884980	0.452125	0.880399	0.933566	0.829114	496.11
10	0.443789	0.872779	0.866766	0.870095	0.429091	0.880399	0.944056	0.829114	495.50
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:55	0.429955		0.894737		0.966967		0.818919		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.704489	0.545401	0.509482	0.621924	0.655854	0.558946	0.097633	0.986945	597.19
2	0.628373	0.706825	0.571429	0.826622	0.606467	0.743412	0.437870	0.953003	592.79
3	0.579809	0.766172	0.670038	0.841163	0.576886	0.771151	0.547337	0.932115	592.71
4	0.530286	0.858160	0.806574	0.894855	0.526997	0.854369	0.949704	0.775457	591.08
5	0.499068	0.863501	0.841972	0.879195	0.502069	0.866852	0.778107	0.926893	590.30
6	0.460610	0.885460	0.873578	0.900447	0.476651	0.858530	0.905325	0.788512	588.63
7	0.436823	0.879525	0.865992	0.891499	0.455601	0.866852	0.917160	0.830287	593.60
8	0.414675	0.897923	0.888748	0.903803	0.448389	0.829404	0.905325	0.741514	592.09
9	0.393527	0.891395	0.887484	0.894855	0.426340	0.857143	0.917160	0.793734	593.10
10	0.372938	0.902077	0.882427	0.916107	0.413354	0.864078	0.943787	0.778068	592.53
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:10	0.383781		0.879085		0.936709		0.831081		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.719023	0.539375	0.543194	0.526316	0.670883	0.699270	0.171254	0.013966	565.06
2	0.637875	0.672500	0.526178	0.722488	0.614332	0.813139	0.507645	0.837989	562.90
3	0.579958	0.833125	0.709424	0.904306	0.572365	0.846715	0.752294	0.882682	563.40
4	0.536314	0.877500	0.876963	0.869617	0.530963	0.856934	0.770642	0.910615	561.53
5	0.494390	0.885625	0.849476	0.918660	0.516447	0.824818	0.892966	0.782123	562.08
6	0.474158	0.876875	0.870419	0.873206	0.491116	0.845255	0.871560	0.812849	561.61
7	0.443168	0.892500	0.859948	0.910287	0.470506	0.836496	0.883792	0.801676	561.63
8	0.424373	0.881250	0.888743	0.875598	0.445512	0.852555	0.874618	0.837989	562.00
9	0.404693	0.890625	0.879581	0.905502	0.441487	0.839416	0.905199	0.770950	561.69
10	0.388447	0.894375	0.896597	0.892344	0.421704	0.855474	0.813456	0.893855	562.30
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:36	0.387962		0.880562		0.816327		0.935065		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:51	0.401952	0.882246	0.883455	0.885976

D.1.27 SGD sigmoid softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693142	0.495886	0.410023	0.873016	0.692794	0.517330	0.141333	0.975248	638.00
2	0.692646	0.521668	0.109339	0.979894	0.692483	0.518614	0.032000	0.992574	633.98
3	0.692277	0.518376	0.019362	0.996825	0.692045	0.518614	0	1	634.98
4	0.691713	0.518376	0	1	0.691231	0.518614	0	1	633.63
5	0.690777	0.518376	0	1	0.690004	0.518614	0	1	633.38
6	0.688957	0.518376	0	1	0.686726	0.518614	0	1	633.77
7	0.684837	0.519473	0.006834	1	0.682147	0.523748	0.042667	1	633.10
8	0.677741	0.612178	0.318907	0.997884	0.674627	0.752246	0.704000	0.985148	632.47
9	0.668240	0.784970	0.734624	0.977778	0.667787	0.772786	0.813333	0.955446	633.91
10	0.659626	0.830499	0.794989	0.980952	0.660699	0.767651	0.709333	0.982673	631.58
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:10	0.640042		0.890909		0.911765		0.960526		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693337	0.481711	0.998713	0.001196	0.693501	0.481910	0.996997	0.002793	567.70
2	0.693116	0.482951	0.976834	0.022727	0.693088	0.496382	0.954955	0.050279	563.66
3	0.692853	0.507130	0.864865	0.159091	0.692882	0.586107	0.825826	0.293296	562.71
4	0.692584	0.603224	0.651223	0.489234	0.692569	0.599132	0.471471	0.737430	562.30
5	0.692451	0.598884	0.431145	0.764354	0.692242	0.580318	0.345345	0.905028	564.56
6	0.692081	0.531928	0.157014	0.954545	0.692147	0.518090	0.051051	0.994413	562.52
7	0.691506	0.523249	0.084942	0.997608	0.691527	0.518090	0.048048	1	565.49
8	0.690522	0.518909	0.011583	1	0.690149	0.518090	0	1	562.30
9	0.689196	0.518289	0.001287	1	0.689307	0.518090	0	1	561.37
10	0.687451	0.518289	0	1	0.687523	0.518090	0	1	563.42
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:40	0.682761		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693099	0.516435	0.780604	0.797297	0.693050	0.514530	0.427509	0.971519	480.43
2	0.692478	0.544193	0.228935	0.987838	0.692178	0.540171	0.026022	1	477.01
3	0.691550	0.540541	0.009539	1	0.690851	0.540171	0	1	477.99
4	0.690219	0.540541	0	1	0.688905	0.540171	0	1	478.89
5	0.687927	0.540541	0	1	0.687234	0.540171	0	1	479.10
6	0.686766	0.540541	0	1	0.684994	0.540171	0	1	478.25
7	0.684719	0.540541	0	1	0.683978	0.540171	0	1	479.42
8	0.682549	0.540541	0	1	0.681606	0.540171	0	1	478.48
9	0.679044	0.546384	0.007949	1	0.676402	0.623932	0.156134	0.984177	479.09
10	0.670759	0.733382	0.419714	0.937838	0.667790	0.772650	0.665428	0.819620	479.61
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:01	0.667206		0.786280		0.614396		0.888889		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693083	0.523099	0.010479	0.009472	0.692910	0.539867	0.010490	0.041139	493.67
2	0.692892	0.534470	0	0.096076	0.692746	0.536545	0	0.199367	491.71
3	0.692618	0.527363	0	0.297700	0.692497	0.524917	0	0.477848	493.70
4	0.692509	0.526652	0	0.500677	0.692221	0.524917	0	0.686709	493.28
5	0.692102	0.525231	0	0.745602	0.692034	0.524917	0	0.857595	493.29
6	0.691946	0.525231	0	0.943166	0.691718	0.524917	0	0.981013	493.39
7	0.691771	0.525231	0	0.970230	0.691325	0.524917	0	0.996835	492.70
8	0.691238	0.525231	0	0.989175	0.690959	0.524917	0	0.990506	490.92
9	0.690696	0.525231	0	0.976996	0.689975	0.524917	0	0.965190	490.02
10	0.689447	0.525231	0	0.966170	0.688168	0.524917	0	0.987342	492.10
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:23	0.687647		0.526316		0		0.983784		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693012	0.522255	0.299621	0.034676	0.692972	0.536754	0.124260	0.078329	590.07
2	0.692669	0.561424	0.035398	0.391499	0.692576	0.532594	0	0.738903	589.06
3	0.692143	0.531157	0	0.913870	0.692028	0.531207	0	0.984334	588.91
4	0.691872	0.530564	0	0.997763	0.691568	0.531207	0	1	586.80
5	0.691193	0.530564	0	1	0.690756	0.531207	0	1	583.77
6	0.689667	0.530564	0	1	0.688490	0.531207	0	1	590.41
7	0.687617	0.530564	0	1	0.686636	0.531207	0	1	589.61
8	0.684989	0.530564	0	1	0.684500	0.531207	0	1	589.23
9	0.681106	0.535905	0.002528	0.998881	0.679272	0.567268	0.035503	0.984334	587.95
10	0.669831	0.728783	0.335019	0.926175	0.670126	0.785021	0.662722	0.723238	587.52
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:59	0.670262		0.794118		0.544304		0.837838		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692710	0.522500	0	0.460526	0.692711	0.522628	0	0.628492	560.66
2	0.692507	0.522500	0	0.753588	0.692549	0.522628	0	0.918994	557.64
3	0.692480	0.522500	0	0.961722	0.692381	0.522628	0	0.994413	557.67
4	0.692225	0.522500	0	0.998804	0.692185	0.522628	0	1	556.71
5	0.691983	0.522500	0	1	0.692024	0.522628	0	1	559.29
6	0.691778	0.522500	0	1	0.691656	0.522628	0	1	557.29
7	0.691366	0.522500	0	1	0.691404	0.522628	0	1	558.31
8	0.690855	0.522500	0	1	0.690561	0.522628	0	1	558.77
9	0.689846	0.522500	0	1	0.688863	0.522628	0	1	559.50
10	0.687648	0.522500	0	1	0.686451	0.524088	0.006116	1	558.49
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:01	0.684137		0.540984		0.005102		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:32	0.672009	0.684130	0.345928	0.945173

D.1.28 SGD sigmoid tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.592038	0.684037	0.623007	0.725926	0.442861	0.854942	0.848000	0.784653	633.58
2	0.358336	0.869995	0.870159	0.875132	0.366163	0.870347	0.874667	0.849010	627.70
3	0.278598	0.910587	0.903189	0.905820	0.339383	0.865212	0.898667	0.844059	631.31
4	0.247872	0.914427	0.913440	0.914286	0.304185	0.883184	0.810667	0.955446	630.57
5	0.225833	0.922106	0.920273	0.921693	0.330500	0.870347	0.768000	0.950495	631.37
6	0.201252	0.933626	0.927107	0.934392	0.294860	0.876765	0.882667	0.891089	631.93
7	0.223313	0.906199	0.906606	0.900529	0.285878	0.890886	0.922667	0.861386	631.19
8	0.200317	0.929786	0.929385	0.926984	0.395723	0.813864	0.616000	0.985148	632.49
9	0.208839	0.917170	0.911162	0.922751	0.257220	0.898588	0.861333	0.938119	628.88
10	0.172470	0.943500	0.938497	0.940741	0.253588	0.894737	0.917333	0.876238	631.63
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:45:53	0.184128		0.918182		1		0.894737		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.662351	0.670800	0.664093	0.651914	0.549394	0.767004	0.882883	0.519553	566.59
2	0.427849	0.867948	0.873874	0.840909	0.423700	0.842258	0.894895	0.748603	563.72
3	0.325660	0.888407	0.881596	0.882775	0.370776	0.855282	0.762763	0.958101	565.55
4	0.273527	0.906386	0.909910	0.894737	0.358012	0.850941	0.912913	0.801676	563.70
5	0.241352	0.924985	0.913771	0.924641	0.311433	0.876990	0.933934	0.815642	562.01
6	0.226209	0.917545	0.926641	0.909091	0.307029	0.875543	0.792793	0.932961	560.10
7	0.192189	0.937384	0.922780	0.942584	0.345077	0.840810	0.948949	0.726257	562.29
8	0.195418	0.934284	0.931789	0.925837	0.285071	0.876990	0.906907	0.837989	561.39
9	0.175081	0.937384	0.939511	0.943780	0.273052	0.890014	0.888889	0.868715	563.06
10	0.176096	0.938004	0.934363	0.935407	0.276911	0.885673	0.828829	0.932961	559.23
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:37	0.255916		0.906863		0.847458		0.948052		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.645868	0.649379	0.508744	0.691892	0.502834	0.852991	0.721190	0.838608	480.89
2	0.423991	0.854638	0.831479	0.881081	0.390030	0.868376	0.847584	0.901899	476.56
3	0.340468	0.878744	0.850556	0.886486	0.347431	0.875214	0.884758	0.863924	476.41
4	0.276519	0.905040	0.885533	0.917568	0.364931	0.854701	0.955390	0.753165	476.70
5	0.276548	0.888240	0.858506	0.914865	0.344462	0.849573	0.933086	0.772152	476.99
6	0.254541	0.904310	0.904611	0.904054	0.310497	0.866667	0.828996	0.911392	476.41
7	0.227357	0.915267	0.915739	0.910811	0.272681	0.895727	0.843866	0.943038	478.43
8	0.229477	0.913806	0.904611	0.928378	0.278258	0.892308	0.855019	0.911392	476.78
9	0.213202	0.926954	0.909380	0.937838	0.286343	0.885470	0.921933	0.848101	477.47
10	0.198095	0.928415	0.934817	0.914865	0.268193	0.897436	0.877323	0.914557	476.82
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:11	0.209074		0.928760		0.884319		0.962060		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.639522	0.601990	0.553892	0.663058	0.472051	0.848837	0.863636	0.829114	497.37
2	0.439895	0.804549	0.794910	0.806495	0.344594	0.882060	0.923077	0.825949	491.79
3	0.343346	0.869936	0.872755	0.866035	0.258468	0.933555	0.919580	0.949367	491.12
4	0.309275	0.879176	0.878743	0.871448	0.254751	0.913621	0.919580	0.924051	493.08
5	0.258632	0.906894	0.893713	0.914750	0.231599	0.930233	0.898601	0.958861	491.90
6	0.249863	0.909737	0.899701	0.913396	0.214189	0.943522	0.947552	0.920886	490.78
7	0.229700	0.914001	0.908683	0.920162	0.204541	0.938538	0.919580	0.946203	490.51
8	0.217652	0.919687	0.923653	0.907984	0.275739	0.893688	0.776224	0.987342	490.91
9	0.214698	0.921109	0.922156	0.932341	0.199967	0.925249	0.947552	0.908228	490.77
10	0.194263	0.928216	0.925150	0.939107	0.198623	0.928571	0.895105	0.962025	491.49
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:08	0.192509		0.937411		0.939940		0.927027		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.585761	0.717507	0.675095	0.700224	0.474699	0.822469	0.928994	0.780679	591.70
2	0.377424	0.858754	0.853350	0.847875	0.370119	0.855756	0.875740	0.804178	588.00
3	0.293462	0.891988	0.882427	0.894855	0.309190	0.880721	0.846154	0.919060	587.10
4	0.236715	0.925816	0.922882	0.927293	0.290131	0.893204	0.902367	0.874674	584.51
5	0.232966	0.914540	0.900126	0.917226	0.283759	0.898752	0.943787	0.859008	589.58
6	0.218033	0.921662	0.926675	0.921700	0.259307	0.891817	0.887574	0.913838	587.00
7	0.193779	0.938872	0.935525	0.939597	0.250369	0.897365	0.908284	0.890339	587.12
8	0.192165	0.925816	0.915297	0.931767	0.314535	0.855756	0.955621	0.762402	588.44
9	0.185513	0.934125	0.932996	0.927293	0.234046	0.900139	0.931953	0.885117	589.79
10	0.172312	0.937685	0.926675	0.946309	0.246251	0.902913	0.943787	0.866841	590.49
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:08	0.214672		0.924837		0.924051		0.918919		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.725217	0.540000	0.515707	0.570574	0.619156	0.756204	0.703364	0.960894	563.14
2	0.531966	0.823125	0.831152	0.740431	0.486176	0.826277	0.828746	0.770950	557.53
3	0.414182	0.855625	0.840314	0.862440	0.436548	0.818978	0.737003	0.918994	559.20
4	0.345520	0.881250	0.878272	0.888756	0.367286	0.867153	0.871560	0.863128	558.97
5	0.288048	0.906875	0.908377	0.904306	0.335791	0.874453	0.859327	0.868715	558.09
6	0.283203	0.894375	0.884817	0.904306	0.327398	0.893431	0.837920	0.946927	558.01
7	0.265796	0.902500	0.903141	0.910287	0.341626	0.851095	0.752294	0.958101	558.17
8	0.232815	0.920000	0.914921	0.923445	0.297920	0.884672	0.917431	0.854749	559.94
9	0.212185	0.936875	0.924084	0.940191	0.314068	0.868613	0.911315	0.840782	558.78
10	0.215246	0.923750	0.912304	0.929426	0.289596	0.880292	0.963303	0.812849	559.51
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:07	0.241165		0.911007		0.948980		0.891775		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:21	0.216244	0.921177	0.924124	0.923762

D.1.29 SGD softmax ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.625626	0.653319	0.626424	0.678307	0.535025	0.765083	0.944000	0.599010	636.58
2	0.412167	0.855732	0.861048	0.850794	0.406220	0.842105	0.912000	0.777228	633.17
3	0.318969	0.886999	0.883827	0.889947	0.355979	0.861361	0.874667	0.849010	633.40
4	0.273743	0.906199	0.902050	0.910053	0.329216	0.872914	0.848000	0.896040	634.30
5	0.245825	0.913878	0.911162	0.916402	0.329934	0.876765	0.800000	0.948020	633.31
6	0.215150	0.927592	0.925968	0.929101	0.313027	0.856226	0.752000	0.952970	633.58
7	0.213111	0.919364	0.923690	0.915344	0.282316	0.874198	0.840000	0.905941	635.00
8	0.210920	0.920461	0.920273	0.920635	0.295045	0.884467	0.930667	0.841584	632.49
9	0.209596	0.921009	0.928246	0.914286	0.271993	0.883184	0.893333	0.873762	632.12
10	0.182650	0.927592	0.929385	0.925926	0.280932	0.879332	0.917333	0.844059	635.69
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:10	0.248234		0.900000		0.970588		0.868421		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.588345	0.731556	0.691120	0.769139	0.508308	0.826339	0.750751	0.896648	572.44
2	0.400112	0.870428	0.862291	0.877990	0.427971	0.807525	0.636637	0.966480	563.08
3	0.311192	0.887167	0.894466	0.880383	0.423936	0.804631	0.639640	0.958101	564.89
4	0.268450	0.901426	0.898327	0.904306	0.357258	0.850941	0.753754	0.941341	564.89
5	0.224676	0.926844	0.920206	0.933014	0.323941	0.874095	0.924925	0.826816	564.80
6	0.208252	0.929944	0.926641	0.933014	0.340352	0.855282	0.891892	0.821229	565.01
7	0.192561	0.937384	0.943372	0.931818	0.305840	0.884226	0.828829	0.935754	564.25
8	0.187667	0.942343	0.939511	0.944976	0.310079	0.866860	0.813814	0.916201	562.43
9	0.175618	0.942343	0.931789	0.952153	0.302932	0.871201	0.963964	0.784916	565.70
10	0.168109	0.944203	0.939511	0.948565	0.253105	0.913169	0.900901	0.924581	563.18
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:37:52	0.264767		0.889706		0.853107		0.917749		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.663710	0.650110	0.513514	0.766216	0.554031	0.777778	0.665428	0.873418	482.92
2	0.469070	0.804967	0.779014	0.827027	0.426029	0.849573	0.832714	0.863924	478.83
3	0.365129	0.855369	0.829889	0.877027	0.377428	0.856410	0.858736	0.854430	478.45
4	0.309305	0.878013	0.866455	0.887838	0.347782	0.871795	0.791822	0.939873	476.16
5	0.292362	0.883126	0.863275	0.900000	0.321481	0.878632	0.892193	0.867089	479.99
6	0.257355	0.905771	0.887122	0.921622	0.290577	0.890598	0.914498	0.870253	480.01
7	0.226709	0.920380	0.901431	0.936486	0.298456	0.875214	0.933086	0.825949	479.69
8	0.233746	0.905771	0.909380	0.902703	0.289742	0.883761	0.810409	0.946203	476.87
9	0.219026	0.924032	0.912560	0.933784	0.258890	0.900855	0.869888	0.927215	479.41
10	0.194331	0.936450	0.931638	0.940541	0.254301	0.885470	0.847584	0.917722	477.47
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:46	0.210884		0.922164		0.881748		0.964770		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.643579	0.621891	0.523952	0.710419	0.563173	0.722591	0.986014	0.484177	493.92
2	0.468644	0.815920	0.833832	0.799729	0.367491	0.880399	0.902098	0.860759	492.71
3	0.370789	0.867093	0.856287	0.876861	0.309845	0.903654	0.954545	0.857595	492.61
4	0.318948	0.882729	0.878743	0.886333	0.260316	0.926910	0.937063	0.917722	492.38
5	0.288945	0.899076	0.913174	0.886333	0.258249	0.916944	0.930070	0.905063	492.70
6	0.267684	0.912580	0.907186	0.917456	0.260557	0.906977	0.863636	0.946203	493.76
7	0.244745	0.909026	0.892216	0.924222	0.252777	0.898671	0.965035	0.838608	492.01
8	0.231592	0.911869	0.919162	0.905277	0.237528	0.900332	0.821678	0.971519	494.00
9	0.207734	0.926795	0.913174	0.939107	0.196872	0.931894	0.930070	0.933544	492.88
10	0.225472	0.913291	0.908683	0.917456	0.223520	0.910299	0.839161	0.974684	494.69
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:11	0.196348		0.927454		0.909910		0.943243		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.599198	0.703858	0.603034	0.793065	0.485366	0.830791	0.730769	0.919060	593.03
2	0.402386	0.855193	0.828066	0.879195	0.374310	0.875173	0.899408	0.853786	587.28
3	0.312347	0.891988	0.882427	0.900447	0.358105	0.847434	0.695266	0.981723	591.81
4	0.275432	0.894956	0.874842	0.912752	0.313351	0.873786	0.769231	0.966057	588.39
5	0.245669	0.910979	0.902655	0.918344	0.256065	0.904300	0.928994	0.882507	590.79
6	0.242064	0.903858	0.908976	0.899329	0.289629	0.880721	0.775148	0.973890	589.87
7	0.224214	0.910979	0.903919	0.917226	0.275698	0.882108	0.931953	0.838120	589.70
8	0.193888	0.931157	0.922882	0.938479	0.233204	0.911234	0.852071	0.963446	589.09
9	0.199554	0.924036	0.921618	0.926175	0.218773	0.918169	0.911243	0.924282	591.21
10	0.176061	0.934718	0.930468	0.938479	0.222629	0.916782	0.899408	0.932115	588.20
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:43	0.212696		0.898693		0.841772		0.959459		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.599051	0.686250	0.612565	0.753588	0.493263	0.808759	0.917431	0.709497	565.15
2	0.387469	0.858125	0.849476	0.866029	0.375176	0.864234	0.874618	0.854749	560.08
3	0.308944	0.891875	0.888743	0.894737	0.363091	0.846715	0.773700	0.913408	560.31
4	0.282395	0.886875	0.883508	0.889952	0.305027	0.871533	0.896024	0.849162	559.80
5	0.231186	0.923750	0.921466	0.925837	0.293750	0.886131	0.819572	0.946927	561.20
6	0.238227	0.905000	0.883508	0.924641	0.273028	0.893431	0.892966	0.893855	560.39
7	0.203428	0.922500	0.929319	0.916268	0.253422	0.900730	0.914373	0.888268	560.40
8	0.219221	0.915625	0.908377	0.922249	0.262024	0.905109	0.969419	0.846369	560.37
9	0.202737	0.918125	0.920157	0.916268	0.281872	0.881752	0.960245	0.810056	559.19
10	0.194232	0.933750	0.934555	0.933014	0.352924	0.839416	0.688073	0.977654	559.50
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:21	0.344822		0.831382		0.632653		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:50	0.246292	0.894900	0.848296	0.942274

D.1.30 SGD softmax sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.679741	0.570488	0.386105	0.741799	0.651006	0.748395	0.557333	0.925743	638.10
2	0.620539	0.773450	0.691344	0.849735	0.602322	0.813864	0.709333	0.910891	631.21
3	0.574689	0.803620	0.830296	0.778836	0.569458	0.788190	0.600000	0.962871	633.60
4	0.526165	0.835985	0.773349	0.894180	0.534953	0.842105	0.840000	0.844059	632.33
5	0.485159	0.873286	0.889522	0.858201	0.517288	0.812580	0.690667	0.925743	631.97
6	0.459709	0.868349	0.846241	0.888889	0.482419	0.845956	0.896000	0.799505	634.08
7	0.435505	0.884257	0.891800	0.877249	0.470809	0.847240	0.789333	0.900990	633.29
8	0.408108	0.884805	0.882688	0.886773	0.457917	0.834403	0.930667	0.745049	633.01
9	0.386979	0.900713	0.905467	0.896296	0.431025	0.847240	0.805333	0.886139	630.12
10	0.367672	0.905101	0.898633	0.911111	0.420865	0.863928	0.904000	0.826733	633.60
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:48:40	0.340907		0.909091		1		0.868421		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692321	0.574086	0.601030	0.549043	0.656071	0.607815	0.195195	0.991620	565.14
2	0.625644	0.700558	0.534106	0.855263	0.605778	0.826339	0.756757	0.891061	563.90
3	0.570004	0.845009	0.831403	0.857656	0.572322	0.784370	0.951952	0.628492	564.92
4	0.526668	0.838190	0.880309	0.799043	0.526022	0.840810	0.906907	0.779330	565.47
5	0.478268	0.885927	0.881596	0.889952	0.499110	0.850941	0.762763	0.932961	565.17
6	0.447691	0.900806	0.921493	0.881579	0.473832	0.862518	0.912913	0.815642	566.57
7	0.417427	0.905146	0.895753	0.913876	0.462356	0.853835	0.897898	0.812849	564.90
8	0.405397	0.875387	0.897040	0.855263	0.449686	0.839363	0.912913	0.770950	566.10
9	0.383267	0.900186	0.889318	0.910287	0.428235	0.848046	0.861862	0.835196	565.48
10	0.361913	0.897706	0.912484	0.883971	0.408719	0.868307	0.885886	0.851955	565.50
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:01	0.443478		0.833333		0.813559		0.848485		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.674917	0.564646	0.119237	0.943243	0.654756	0.623932	0.189591	0.993671	479.52
2	0.636133	0.681519	0.329094	0.981081	0.616994	0.851282	0.840149	0.860759	477.59
3	0.592948	0.792549	0.658188	0.906757	0.581713	0.842735	0.925651	0.772152	478.13
4	0.552151	0.840760	0.739269	0.927027	0.545976	0.863248	0.847584	0.876582	479.79
5	0.523322	0.848795	0.796502	0.893243	0.521201	0.870085	0.907063	0.838608	477.55
6	0.493466	0.871439	0.809221	0.924324	0.502438	0.842735	0.959108	0.743671	480.85
7	0.470881	0.869978	0.845787	0.890541	0.479822	0.863248	0.847584	0.876582	478.26
8	0.452934	0.879474	0.833068	0.918919	0.460555	0.871795	0.799257	0.933544	479.91
9	0.431034	0.878744	0.825119	0.924324	0.457338	0.864957	0.828996	0.895570	479.46
10	0.402953	0.896275	0.856916	0.929730	0.436411	0.854701	0.892193	0.822785	479.08
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:29	0.398951		0.901055		0.897172		0.905149		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.688213	0.570718	0.344311	0.775372	0.663648	0.730897	0.601399	0.848101	498.76
2	0.650529	0.687989	0.390719	0.956698	0.622383	0.850498	0.758741	0.933544	494.10
3	0.611125	0.810235	0.733533	0.879567	0.591866	0.820598	0.678322	0.949367	495.84
4	0.576534	0.835821	0.784431	0.882273	0.552636	0.875415	0.807692	0.936709	495.10
5	0.549828	0.835110	0.806886	0.860622	0.523731	0.858804	0.783217	0.927215	494.67
6	0.517706	0.866382	0.824850	0.903924	0.498837	0.887043	0.895105	0.879747	494.87
7	0.495116	0.854300	0.821856	0.883627	0.485023	0.843854	0.972028	0.727848	493.90
8	0.483331	0.839375	0.824850	0.852503	0.462491	0.873754	0.940559	0.813291	492.89
9	0.456463	0.863539	0.857784	0.868742	0.445414	0.883721	0.930070	0.841772	492.42
10	0.443478	0.873490	0.857784	0.887686	0.416194	0.913621	0.874126	0.949367	493.42
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:50	0.423913		0.883357		0.852853		0.910811		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.689026	0.573294	0.470291	0.664430	0.658529	0.755895	0.644970	0.853786	594.01
2	0.633317	0.767359	0.563843	0.947427	0.614206	0.847434	0.792899	0.895561	590.50
3	0.584873	0.848071	0.801517	0.889262	0.589418	0.680999	0.331361	0.989556	589.34
4	0.554730	0.790504	0.686473	0.882550	0.548476	0.834951	0.724852	0.932115	587.90
5	0.515712	0.869436	0.852086	0.884787	0.518526	0.841886	0.908284	0.783290	589.89
6	0.487063	0.877151	0.830594	0.918344	0.505174	0.843273	0.860947	0.827676	591.17
7	0.461640	0.875964	0.853350	0.895973	0.470375	0.859917	0.846154	0.872063	589.49
8	0.444284	0.862908	0.831858	0.890380	0.448594	0.872399	0.890533	0.856397	588.91
9	0.417531	0.883680	0.891277	0.876957	0.445091	0.871013	0.920118	0.827676	591.18
10	0.388810	0.897329	0.871049	0.920582	0.424922	0.884882	0.878698	0.890339	588.30
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:18	0.418826		0.872549		0.797468		0.952703		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.724935	0.510000	0.518325	0.502392	0.668320	0.769343	0.737003	0.798883	564.71
2	0.640726	0.780625	0.641361	0.907895	0.629307	0.712409	0.422018	0.977654	557.69
3	0.599074	0.831250	0.717277	0.935407	0.597201	0.810219	0.715596	0.896648	556.98
4	0.558149	0.819375	0.841623	0.799043	0.566226	0.789781	0.587156	0.974860	557.60
5	0.526132	0.816875	0.808901	0.824163	0.530072	0.840876	0.740061	0.932961	558.42
6	0.503332	0.857500	0.882199	0.834928	0.507069	0.836496	0.825688	0.846369	558.77
7	0.463935	0.888750	0.865183	0.910287	0.494442	0.823358	0.923547	0.731844	558.41
8	0.444956	0.876250	0.895288	0.858852	0.469326	0.851095	0.844037	0.857542	558.29
9	0.428239	0.887500	0.861257	0.911483	0.449686	0.859854	0.892966	0.829609	555.30
10	0.402941	0.893125	0.905759	0.881579	0.440379	0.848175	0.847095	0.849162	553.58
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:52	0.404975		0.889930		0.887755		0.891775		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:52	0.405175	0.881553	0.874801	0.896224

D.1.31 SGD softmax softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693343	0.482172	0.998861	0.002116	0.693325	0.486521	0.992000	0.017327	635.35
2	0.693015	0.525507	0.924829	0.154497	0.692963	0.540436	0.709333	0.383663	634.10
3	0.692635	0.601207	0.510251	0.685714	0.692581	0.551990	0.208000	0.871287	629.71
4	0.692228	0.537027	0.082005	0.959788	0.692026	0.519897	0.002667	1	630.72
5	0.691462	0.518376	0	1	0.691233	0.518614	0	1	633.40
6	0.689881	0.518376	0	1	0.688258	0.518614	0	1	629.47
7	0.686865	0.518376	0	1	0.685396	0.518614	0	1	633.13
8	0.681764	0.530993	0.026196	1	0.679075	0.618742	0.229333	0.980198	634.67
9	0.672239	0.717499	0.460137	0.956614	0.671230	0.703466	0.416000	0.970297	631.82
10	0.662182	0.795941	0.634396	0.946032	0.663344	0.722721	0.456000	0.970297	632.01
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:05	0.641242		0.900000		0.705882		0.986842		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692948	0.518289	0.001287	0.998804	0.692716	0.518090	0	1	564.40
2	0.692712	0.518289	0	1	0.692341	0.518090	0	1	562.99
3	0.692378	0.518289	0	1	0.692055	0.518090	0	1	562.20
4	0.692065	0.518289	0	1	0.691450	0.518090	0	1	564.97
5	0.691244	0.518289	0	1	0.690220	0.518090	0	1	564.00
6	0.689834	0.518289	0	1	0.688327	0.518090	0	1	565.40
7	0.687780	0.518289	0	1	0.686870	0.518090	0	1	562.88
8	0.685884	0.518289	0	1	0.684612	0.518090	0	1	564.50
9	0.682819	0.518289	0	1	0.682908	0.519537	0.003003	1	563.14
10	0.679323	0.559826	0.087516	0.998804	0.676190	0.697540	0.387387	0.986033	564.06
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:43	0.675536		0.720588		0.372881		0.987013		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693284	0.474069	0.227345	0.683784	0.693144	0.516239	0.011152	0.946203	483.25
2	0.692788	0.539810	0.011129	0.989189	0.692673	0.538462	0	0.996835	478.20
3	0.692494	0.541271	0.001590	1	0.692439	0.540171	0	1	477.47
4	0.691972	0.540541	0	1	0.692045	0.540171	0	1	477.38
5	0.691667	0.540541	0	1	0.691549	0.540171	0	1	478.01
6	0.691236	0.540541	0	1	0.691259	0.540171	0	1	478.78
7	0.690811	0.540541	0	1	0.690759	0.540171	0	1	479.41
8	0.690294	0.540541	0	1	0.690263	0.540171	0	1	478.46
9	0.689710	0.540541	0	1	0.689354	0.540171	0	1	478.10
10	0.688983	0.540541	0	1	0.688722	0.540171	0	1	477.72
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:22:59	0.693537		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692918	0.525231	0	1	0.692836	0.524917	0	1	494.33
2	0.692747	0.525231	0	1	0.692399	0.524917	0	1	491.50
3	0.692327	0.525231	0	1	0.692208	0.524917	0	1	491.49
4	0.691951	0.525231	0	1	0.691625	0.524917	0	1	490.77
5	0.690980	0.525231	0	1	0.690488	0.524917	0	1	488.70
6	0.689438	0.525231	0	1	0.689002	0.524917	0	1	491.91
7	0.688331	0.525231	0	1	0.686938	0.524917	0	1	492.80
8	0.686393	0.525231	0	1	0.685904	0.524917	0	1	492.28
9	0.684372	0.525231	0	1	0.682866	0.524917	0	1	493.89
10	0.680800	0.537313	0.026946	0.998647	0.677594	0.601329	0.160839	1	492.09
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:23	0.676997		0.625889		0.216216		0.994595		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.692986	0.508605	0.180784	0.798658	0.692637	0.531207	0.005917	0.994778	590.31
2	0.692482	0.531751	0.003793	0.998881	0.692246	0.531207	0	1	586.97
3	0.692133	0.530564	0	1	0.691699	0.531207	0	1	589.29
4	0.691691	0.530564	0	1	0.691179	0.531207	0	1	588.11
5	0.691265	0.530564	0	1	0.690890	0.531207	0	1	587.80
6	0.690770	0.530564	0	1	0.690177	0.531207	0	1	589.70
7	0.690224	0.530564	0	1	0.689609	0.531207	0	1	589.67
8	0.689674	0.530564	0	1	0.689069	0.531207	0	1	587.16
9	0.688884	0.530564	0	1	0.688178	0.531207	0	1	588.03
10	0.687417	0.530564	0	1	0.685590	0.531207	0	1	588.40
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:04	0.690725		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.693145	0.500625	0.243455	0.735646	0.693177	0.502190	0.070336	0.896648	562.19
2	0.692935	0.521250	0.020942	0.978469	0.692916	0.522628	0	1	557.36
3	0.692786	0.522500	0.001309	0.998804	0.692546	0.522628	0	1	560.21
4	0.692487	0.522500	0	1	0.692491	0.522628	0	1	558.79
5	0.692294	0.522500	0	1	0.692141	0.522628	0	1	558.72
6	0.691979	0.522500	0	1	0.691850	0.522628	0	1	561.29
7	0.691727	0.522500	0	1	0.691344	0.522628	0	1	559.12
8	0.691173	0.522500	0	1	0.690687	0.522628	0	1	558.79
9	0.689858	0.522500	0	1	0.688761	0.522628	0	1	555.88
10	0.687961	0.522500	0	1	0.686067	0.522628	0	1	555.81
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:04	0.683480		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:23	0.676919	0.626321	0.215830	0.994742

D.1.32 SGD softmax tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.562401	0.727921	0.732346	0.723810	0.507440	0.745828	0.952000	0.554455	638.79
2	0.368474	0.846407	0.846241	0.846561	0.369411	0.852375	0.904000	0.804455	634.89
3	0.290003	0.893033	0.887244	0.898413	0.331940	0.854942	0.946667	0.769802	630.89
4	0.264080	0.890291	0.892938	0.887831	0.305476	0.883184	0.824000	0.938119	629.07
5	0.237165	0.905101	0.908884	0.901587	0.288170	0.894737	0.877333	0.910891	630.70
6	0.208639	0.922655	0.920273	0.924868	0.332859	0.847240	0.976000	0.727723	631.33
7	0.206414	0.919912	0.931663	0.908995	0.274610	0.884467	0.842667	0.923267	634.82
8	0.176912	0.944597	0.935080	0.953439	0.265958	0.892169	0.882667	0.900990	630.30
9	0.178159	0.927592	0.925968	0.929101	0.258035	0.892169	0.936000	0.851485	633.00
10	0.164501	0.939660	0.938497	0.940741	0.248266	0.905006	0.882667	0.925743	634.10
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:08	0.118560		0.954545		0.970588		0.947368		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.570074	0.738996	0.752896	0.726077	0.473114	0.819103	0.732733	0.899441	566.97
2	0.363016	0.887167	0.880309	0.893541	0.387751	0.861071	0.882883	0.840782	563.19
3	0.282965	0.910105	0.907336	0.912679	0.339703	0.869754	0.795796	0.938547	562.69
4	0.262982	0.905146	0.903475	0.906699	0.318525	0.878437	0.882883	0.874302	565.24
5	0.227040	0.924365	0.924067	0.924641	0.285361	0.894356	0.912913	0.877095	563.78
6	0.230567	0.908246	0.895753	0.919856	0.282559	0.891462	0.957958	0.829609	564.00
7	0.194509	0.932424	0.939511	0.925837	0.332605	0.855282	0.747748	0.955307	564.37
8	0.185227	0.932424	0.929215	0.935407	0.305370	0.868307	0.936937	0.804469	565.29
9	0.161244	0.947923	0.940798	0.954545	0.274362	0.895803	0.918919	0.874302	566.18
10	0.167852	0.941724	0.951094	0.933014	0.288963	0.879884	0.903904	0.857542	563.21
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:52	0.278061		0.877451		0.892655		0.865801		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.658028	0.597516	0.484897	0.693243	0.537959	0.827350	0.754647	0.889241	480.16
2	0.456809	0.827611	0.801272	0.850000	0.425735	0.832479	0.758364	0.895570	480.39
3	0.367714	0.859752	0.837838	0.878378	0.329629	0.895727	0.907063	0.886076	478.22
4	0.321931	0.868517	0.864865	0.871622	0.317063	0.880342	0.843866	0.911392	479.45
5	0.269136	0.910153	0.891892	0.925676	0.340299	0.858120	0.910781	0.813291	478.40
6	0.250850	0.913806	0.906200	0.920270	0.285480	0.895727	0.832714	0.949367	478.00
7	0.241788	0.913806	0.893482	0.931081	0.302709	0.866667	0.955390	0.791139	479.18
8	0.222408	0.920380	0.915739	0.924324	0.285441	0.890598	0.910781	0.873418	479.91
9	0.216021	0.916728	0.917329	0.916216	0.300723	0.873504	0.944238	0.813291	477.76
10	0.217829	0.921110	0.904611	0.935135	0.276779	0.878632	0.825279	0.924051	478.81
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:26	0.212913		0.920844		0.866324		0.978320		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.645096	0.621180	0.534431	0.699594	0.497900	0.858804	0.790210	0.920886	495.91
2	0.448113	0.831556	0.829341	0.833559	0.370950	0.873754	0.947552	0.806962	491.01
3	0.355814	0.878465	0.874251	0.882273	0.318660	0.893688	0.853147	0.930380	492.77
4	0.310977	0.883440	0.893713	0.874154	0.295269	0.893688	0.804196	0.974684	493.18
5	0.291769	0.894101	0.880240	0.906631	0.248581	0.921927	0.919580	0.924051	490.83
6	0.257238	0.909737	0.910180	0.909337	0.237415	0.923588	0.909091	0.936709	495.09
7	0.230094	0.931770	0.931138	0.932341	0.218796	0.930233	0.884615	0.971519	492.31
8	0.225220	0.921109	0.922156	0.920162	0.199045	0.938538	0.975524	0.905063	490.45
9	0.219252	0.921109	0.914671	0.926928	0.213106	0.926910	0.919580	0.933544	492.32
10	0.206874	0.924662	0.922156	0.926928	0.218468	0.925249	0.888112	0.958861	493.79
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:07	0.201831		0.934566		0.942943		0.927027		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.596434	0.650445	0.621997	0.675615	0.467885	0.808599	0.659763	0.939948	594.07
2	0.356547	0.869436	0.858407	0.879195	0.392574	0.821082	0.662722	0.960836	585.59
3	0.311493	0.878338	0.863464	0.891499	0.321628	0.869626	0.943787	0.804178	588.89
4	0.254992	0.897329	0.888748	0.904922	0.318154	0.879334	0.763314	0.981723	588.60
5	0.257430	0.900297	0.881163	0.917226	0.317784	0.854369	0.976331	0.746736	592.09
6	0.220015	0.918694	0.919090	0.918344	0.243253	0.911234	0.905325	0.916449	589.97
7	0.209114	0.923442	0.924147	0.922819	0.306579	0.859917	0.733728	0.971279	586.43
8	0.201296	0.927003	0.915297	0.937360	0.259943	0.883495	0.813609	0.945170	588.56
9	0.186258	0.932344	0.922882	0.940716	0.244755	0.904300	0.917160	0.892950	589.62
10	0.177152	0.932344	0.936789	0.928412	0.248610	0.902913	0.973373	0.840731	590.39
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:38	0.187639		0.928105		0.930380		0.925676		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.604138	0.688125	0.621728	0.748804	0.504744	0.788321	0.944954	0.645251	561.51
2	0.407024	0.835625	0.835079	0.836124	0.391100	0.852555	0.770642	0.927374	557.97
3	0.315651	0.891250	0.880890	0.900718	0.352664	0.855474	0.911315	0.804469	556.90
4	0.276727	0.894375	0.896597	0.892344	0.292268	0.891971	0.865443	0.916201	558.07
5	0.238645	0.919375	0.921466	0.917464	0.310121	0.875912	0.847095	0.902235	559.52
6	0.248414	0.903750	0.893979	0.912679	0.326535	0.854015	0.737003	0.960894	558.02
7	0.221402	0.914375	0.914921	0.913876	0.298857	0.846715	0.724771	0.958101	556.73
8	0.221037	0.920000	0.918848	0.921053	0.261496	0.891971	0.844037	0.935754	556.68
9	0.191291	0.930000	0.926702	0.933014	0.254633	0.887591	0.859327	0.913408	557.43
10	0.205385	0.920625	0.920157	0.921053	0.226155	0.910949	0.905199	0.916201	557.60
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:03	0.195127		0.932084		0.892857		0.965368		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:12	0.199022	0.924599	0.915958	0.934927

D.1.33 SGD tanh ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.027322	0.511794	0.133257	0.960847	7.383068	0.518614	0	1	634.41
2	7.386707	0.518376	0	1	7.383068	0.518614	0	1	628.89
3	7.386707	0.518376	0	1	7.383068	0.518614	0	1	631.82
4	7.386708	0.518376	0	1	7.383067	0.518614	0	1	632.58
5	7.386707	0.518376	0	1	7.383067	0.518614	0	1	631.79
6	7.386708	0.518376	0	1	7.383068	0.518614	0	1	632.90
7	7.386707	0.518376	0	1	7.383068	0.518614	0	1	631.63
8	7.386707	0.518376	0	1	7.383069	0.518614	0	1	631.47
9	7.386708	0.518376	0	1	7.383068	0.518614	0	1	634.20
10	7.386708	0.518376	0	1	7.383068	0.518614	0	1	634.89
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:49:26	4.740556		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.925029	0.516429	0.097812	0.966507	7.391103	0.518090	0	1	567.58
2	7.388048	0.518289	0	1	7.391103	0.518090	0	1	562.93
3	7.388048	0.518289	0	1	7.391102	0.518090	0	1	563.54
4	7.388049	0.518289	0	1	7.391103	0.518090	0	1	563.49
5	7.388048	0.518289	0	1	7.391103	0.518090	0	1	563.40
6	7.388048	0.518289	0	1	7.391103	0.518090	0	1	562.51
7	7.388048	0.518289	0	1	7.391104	0.518090	0	1	563.60
8	7.388048	0.518289	0	1	7.391103	0.518090	0	1	565.49
9	7.388048	0.518289	0	1	7.391103	0.518090	0	1	564.77
10	7.388050	0.518289	0	1	7.391103	0.518090	0	1	562.58
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:38:42	6.653592		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.382592	0.539080	0.958665	0.960811	7.624619	0.540171	1	1	480.27
2	7.624619	0.540541	1	1	7.624619	0.540171	1	1	475.60
3	7.624619	0.540541	1	1	7.624619	0.540171	1	1	478.96
4	7.624619	0.540541	1	1	7.624619	0.540171	1	1	477.41
5	7.624619	0.540541	1	1	7.624619	0.540171	1	1	478.39
6	7.624619	0.540541	1	1	7.624619	0.540171	1	1	477.18
7	7.624619	0.540541	1	1	7.624619	0.540171	1	1	478.03
8	7.624619	0.540541	1	1	7.624619	0.540171	1	1	481.65
9	7.624619	0.540541	1	1	7.624619	0.540171	1	1	478.90
10	7.624619	0.540541	1	1	7.624619	0.540171	1	1	480.11
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:22:59	7.624619		0.486807		1		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.680506	0.528785	0.193114	0.131258	7.712474	0.524917	0	0	492.40
2	7.712474	0.525231	0	0	7.712474	0.524917	0	0	489.39
3	7.712474	0.525231	0	0	7.712474	0.524917	0	0	490.58
4	7.712474	0.525231	0	0	7.712474	0.524917	0	0	491.62
5	7.712474	0.525231	0	0	7.712474	0.524917	0	0	491.62
6	7.712474	0.525231	0	0	7.712474	0.524917	0	0	490.90
7	7.712474	0.525231	0	0	7.712474	0.524917	0	0	490.84
8	7.712474	0.525231	0	0	7.712474	0.524917	0	0	492.17
9	7.712474	0.525231	0	0	7.712474	0.524917	0	0	490.10
10	7.712474	0.525231	0	0	7.712474	0.524917	0	0	492.18
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:12	7.712474		0.526316		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.902081	0.524036	0.135272	0.956376	7.189927	0.531207	0	1	593.35
2	7.199788	0.530564	0	1	7.189927	0.531207	0	1	588.80
3	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.91
4	7.199787	0.530564	0	1	7.189928	0.531207	0	1	585.82
5	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.34
6	7.199787	0.530564	0	1	7.189927	0.531207	0	1	587.99
7	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.98
8	7.199787	0.530564	0	1	7.189927	0.531207	0	1	588.29
9	7.199787	0.530564	0	1	7.189927	0.531207	0	1	587.59
10	7.199787	0.530564	0	1	7.189927	0.531207	0	1	590.10
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:14	7.919154		0.483660		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.712237	0.482500	0.971204	0.034689	8.015591	0.477372	1	0	561.38
2	8.013631	0.477500	1	0	8.015591	0.477372	1	0	555.81
3	8.013631	0.477500	1	0	8.015591	0.477372	1	0	557.77
4	8.013632	0.477500	1	0	8.015590	0.477372	1	0	554.99
5	8.013632	0.477500	1	0	8.015592	0.477372	1	0	558.79
6	8.013631	0.477500	1	0	8.015590	0.477372	1	0	555.05
7	8.013632	0.477500	1	0	8.015590	0.477372	1	0	561.05
8	8.013632	0.477500	1	0	8.015591	0.477372	1	0	556.94
9	8.013632	0.477500	1	0	8.015591	0.477372	1	0	559.76
10	8.013632	0.477500	1	0	8.015591	0.477372	1	0	560.92
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:59	8.297116		0.459016		1		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:35:25	7.157919	0.535481	0.333333	0.666667

D.1.34 SGD tanh sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.015520	0.499726	0.781321	0.900529	0.823875	0.518614	0.005333	1	633.90
2	0.719398	0.544158	0.570615	0.649735	0.681793	0.519897	0.576000	0	630.66
3	0.649224	0.715304	0.478360	0.702645	0.640620	0.630295	0.122667	0.905941	632.54
4	0.602673	0.798135	0.605923	0.790476	0.599734	0.754814	0.434667	0.915842	633.89
5	0.528516	0.818980	0.717540	0.774603	0.541960	0.801027	0.530667	0.779703	633.49
6	0.456450	0.854635	0.805239	0.786243	0.500520	0.772786	0.952000	0.542079	632.40
7	0.419742	0.840922	0.884966	0.804233	0.536522	0.811297	0.594667	0.948020	630.18
8	0.359949	0.868897	0.894077	0.850794	0.423018	0.834403	0.922667	0.631188	632.78
9	0.322003	0.901810	0.908884	0.880423	0.377583	0.875481	0.890667	0.863861	633.40
10	0.296701	0.904553	0.904328	0.908995	0.431680	0.847240	0.930667	0.757426	631.91
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:57	0.325097		0.863636		1		0.802632		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.376729	0.517049	0.553411	0.960526	2.474809	0.518090	0.012012	1	567.63
2	1.308699	0.532548	0.521236	0.764354	0.677516	0.518090	0.714715	1	563.02
3	0.673726	0.570986	0.546976	0.686603	0.644765	0.807525	0.516517	0	563.68
4	0.609926	0.778053	0.674389	0.759569	0.605496	0.810420	0.660661	0.969274	564.00
5	0.553336	0.856169	0.767053	0.877990	0.563690	0.843705	0.801802	0.187151	562.98
6	0.509180	0.869188	0.817246	0.802632	0.511740	0.849494	0.894895	0.681564	559.20
7	0.464558	0.872908	0.858430	0.738038	0.481637	0.846599	0.747748	0.851955	560.00
8	0.423829	0.885927	0.880309	0.805024	0.509214	0.845152	0.642643	0.759777	559.40
9	0.357482	0.903286	0.922780	0.795455	0.398640	0.868307	0.873874	0.737430	563.10
10	0.330582	0.915065	0.925354	0.842105	0.386999	0.874095	0.873874	0.835196	560.31
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:29	0.435276		0.865196		0.864407		0.831169		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.485930	0.459459	0.575517	0	4.420006	0.459829	0.843866	0	480.36
2	4.391735	0.459459	0.817170	0	4.373889	0.459829	0.836431	0	478.23
3	4.373042	0.459459	0.866455	0	4.346422	0.459829	0.769517	0	476.08
4	4.761868	0.460190	0.831479	0	6.132229	0.459829	0.026022	0	476.99
5	4.813642	0.460190	0.887122	0	4.428581	0.459829	0.910781	0	477.29
6	4.638205	0.459459	0.798092	0	4.548171	0.459829	0	0	478.22
7	4.467241	0.459459	0.602544	0	4.451830	0.459829	0.074349	0	476.34
8	4.404589	0.459459	0.578696	0	4.388969	0.459829	0.802974	0	477.69
9	4.351618	0.459459	0.852146	0	4.377525	0.459829	0.843866	0	478.57
10	4.349631	0.459459	0.860095	0	4.865078	0.459829	1	0	476.32
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:11	4.349597		0.513193		1		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.693806	0.478323	0.940120	0.757781	4.329464	0.475083	1	0.063291	495.89
2	4.335068	0.474769	1	0.537212	4.299144	0.475083	1	0.781646	492.19
3	4.295321	0.474769	1	0.782138	4.270969	0.475083	1	0.971519	491.97
4	4.257604	0.474769	1	0.779432	4.230316	0.475083	1	0.876582	489.70
5	4.225754	0.474769	1	0.801083	4.206243	0.475083	1	0.886076	493.13
6	4.203267	0.474769	1	0.820027	4.178712	0.475083	1	0.905063	492.59
7	4.193039	0.474769	1	0.845737	4.180143	0.475083	1	0.981013	493.50
8	4.178253	0.474769	1	0.891746	4.158368	0.475083	1	0.939873	492.58
9	4.168115	0.474769	1	0.901218	4.216926	0.475083	1	0.993671	491.61
10	4.177198	0.474769	1	0.912043	4.142870	0.475083	1	0.946203	492.00
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:16	4.146661		0.473684		1		0.932432		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	2.517016	0.509792	0.761062	0.705817	1.835507	0.468793	1	1	590.17
2	1.023132	0.483086	0.710493	0.663311	0.748710	0.468793	1	0.759791	587.90
3	0.694370	0.551929	0.597977	0.807606	0.670615	0.793343	0.991124	0.997389	586.70
4	0.658429	0.685460	0.481669	0.794183	0.646117	0.575589	0.002959	0.989556	589.97
5	0.622776	0.769139	0.340076	0.825503	0.606060	0.805825	0.245562	0.848564	588.92
6	0.573563	0.846291	0.486726	0.791946	0.563039	0.819695	0.671598	0.626632	587.89
7	0.509696	0.856973	0.579014	0.814318	0.502519	0.852982	0.742604	0.678851	612.15
8	0.444184	0.863501	0.786346	0.778524	0.436458	0.865465	0.819527	0.806789	585.81
9	0.400410	0.857567	0.792668	0.875839	0.391762	0.864078	0.810651	0.866841	588.28
10	0.345577	0.886647	0.864728	0.876957	0.367610	0.875173	0.837278	0.874674	586.20
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:39:47	0.366003		0.879085		0.772152		0.925676		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	3.627293	0.522500	0.434555	0.964115	0.811899	0.477372	1	0.100559	561.68
2	0.745942	0.542500	0.680628	0.635167	0.692298	0.598540	0.214067	0	556.90
3	0.642293	0.696875	0.531414	0.519139	0.624156	0.632117	0.330275	1	558.91
4	0.588798	0.769375	0.693717	0.819378	0.586547	0.816058	0.896024	0.840782	557.50
5	0.544080	0.847500	0.810209	0.739234	0.534160	0.824818	0.737003	0.874302	558.50
6	0.473637	0.880625	0.859948	0.758373	0.464570	0.868613	0.917431	0.913408	556.61
7	0.419111	0.885625	0.882199	0.802632	0.449646	0.856934	0.911315	0.810056	557.42
8	0.388199	0.885000	0.904450	0.812201	0.415613	0.874453	0.822630	0.832402	558.13
9	0.373396	0.890625	0.899215	0.846890	0.471408	0.878832	0.730887	0.916201	558.56
10	0.336437	0.892500	0.900524	0.883971	0.383080	0.845255	0.944954	0.765363	557.21
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:57	0.286110		0.896955		0.959184		0.852814		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:26	1.651458	0.748625	0.932624	0.724120

D.1.35 SGD tanh softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.917909	0.515085	0	0.488889	0.694969	0.518614	0	1	632.75
2	0.695581	0.518376	0.289294	1	0.694609	0.518614	0	1	629.53
3	0.693879	0.518376	0	1	0.693057	0.518614	0	1	633.16
4	0.693272	0.518376	0	1	0.692551	0.518614	0	1	628.01
5	0.692952	0.518376	0	1	0.692518	0.518614	0	1	628.60
6	0.692811	0.518376	0	1	0.692462	0.518614	0	1	633.22
7	0.692532	0.518376	0	1	0.692459	0.518614	0	1	631.51
8	0.692573	0.518376	0	1	0.692488	0.518614	0	1	629.99
9	0.692543	0.518376	0	1	0.692458	0.518614	0	1	632.47
10	0.692522	0.518376	0	1	0.692456	0.518614	0	1	631.23
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:47:32	0.680299		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.221580	0.481711	0.480051	0.040670	0.698761	0.481910	1	1	568.94
2	0.698793	0.481711	1	1	0.697344	0.481910	1	1	563.10
3	0.695592	0.518289	1	1	0.693959	0.518090	1	1	561.50
4	0.693367	0.518289	1	1	0.692863	0.518090	0	1	561.90
5	0.692927	0.518289	0	1	0.692625	0.518090	0	1	564.49
6	0.692592	0.518289	0	1	0.692520	0.518090	0	1	561.28
7	0.692567	0.518289	0	1	0.692524	0.518090	0	1	561.10
8	0.692524	0.518289	0	1	0.692496	0.518090	0	1	561.48
9	0.692545	0.518289	0	1	0.692518	0.518090	0	1	560.68
10	0.692729	0.518289	0	1	0.692575	0.518090	0	1	563.80
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:35:34	0.687860		0.566176		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.357827	0.498904	0.440382	0.537838	0.735949	0.459829	1	1	480.48
2	0.746449	0.459459	1	1	0.741851	0.459829	1	1	476.56
3	0.729487	0.459459	1	1	0.716410	0.459829	1	1	477.00
4	0.707964	0.493791	1	1	0.700285	0.540171	1	1	478.53
5	0.696476	0.540541	1	1	0.693477	0.540171	1	1	477.30
6	0.692109	0.540541	0.553259	1	0.690848	0.540171	0	1	479.13
7	0.690419	0.540541	0	1	0.690143	0.540171	0	1	479.48
8	0.690100	0.540541	0	1	0.689965	0.540171	0	1	478.21
9	0.689912	0.540541	0	1	0.689929	0.540171	0	1	476.79
10	0.689916	0.540541	0	1	0.689928	0.540171	0	1	478.40
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:23:17	0.698978		0.486807		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.022960	0.474058	0.034431	0	0.695568	0.475083	0.486014	0	497.23
2	0.685618	0.547974	0.094311	0.566982	0.678189	0.606312	0.300699	1	491.80
3	0.674362	0.639659	0.513473	0.990528	0.663403	0.715947	0.541958	1	492.12
4	0.657477	0.744847	0.613772	0.989175	0.640556	0.807309	0.688811	0.987342	492.72
5	0.633173	0.792466	0.670659	0.964817	0.610959	0.870432	0.783217	0.962025	494.26
6	0.604423	0.829424	0.700599	0.930988	0.581223	0.863787	0.783217	0.911392	493.31
7	0.574417	0.849325	0.764970	0.878214	0.546735	0.897010	0.926573	0.822785	490.99
8	0.536210	0.868515	0.802395	0.878214	0.507920	0.900332	0.814685	0.936709	493.77
9	0.499637	0.889126	0.845808	0.871448	0.490512	0.858804	0.681818	0.984177	491.70
10	0.469466	0.881308	0.812874	0.878214	0.430703	0.926910	0.933566	0.857595	492.72
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:28	0.444112		0.914652		0.951952		0.813514		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.825116	0.531157	0.192162	0.629754	0.706306	0.531207	1	1	593.10
2	0.706380	0.530564	1	1	0.702406	0.531207	1	1	588.89
3	0.698494	0.530564	0.257901	1	0.695029	0.531207	0	1	587.16
4	0.693628	0.530564	0	1	0.691948	0.531207	0	1	588.81
5	0.691755	0.530564	0	1	0.690998	0.531207	0	1	587.36
6	0.690132	0.530564	0	1	0.686678	0.531207	0	1	589.54
7	0.683460	0.530564	0	1	0.677279	0.531207	0	1	588.38
8	0.668194	0.530564	0	1	0.657884	0.531207	0	1	586.21
9	0.644646	0.635015	0.152971	0.993289	0.635316	0.836338	0.677515	0.845953	584.70
10	0.610766	0.827300	0.610619	0.929530	0.602788	0.793343	0.538462	0.926893	584.80
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:40:31	0.605058		0.751634		0.462025		0.972973		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.000113	0.476250	0.835079	0.806220	0.961518	0.477372	1	1	561.44
2	0.943409	0.477500	1	1	0.901285	0.477372	1	1	558.49
3	0.856027	0.477500	1	1	0.810919	0.477372	1	1	555.90
4	0.779130	0.477500	1	1	0.749305	0.477372	1	1	556.38
5	0.731215	0.477500	1	1	0.715674	0.477372	1	1	554.83
6	0.707210	0.477500	1	1	0.699949	0.477372	1	1	555.79
7	0.696948	0.508750	1	1	0.694283	0.522628	1	1	559.60
8	0.693503	0.522500	0.867801	1	0.692626	0.522628	0	1	556.35
9	0.692558	0.522500	0	1	0.692273	0.522628	0	1	559.77
10	0.692449	0.522500	0	1	0.692143	0.522628	0	1	557.01
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:50	0.690396		0.540984		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:32	0.634450	0.658527	0.235663	0.964414

D.1.36 SGD tanh tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	1.747801	0.546901	0.732346	0.713228	0.627277	0.720154	0.794667	0.611386	635.68
2	0.531812	0.805815	0.691344	0.807407	0.504259	0.765083	0.922667	0.522277	632.79
3	0.356102	0.869995	0.874715	0.857143	0.369829	0.870347	0.901333	0.866337	633.80
4	0.736304	0.774547	0.943052	0.922751	1.877848	0.518614	0	1	632.21
5	1.186220	0.517279	0.552392	0.897354	0.698456	0.518614	0	1	631.96
6	0.630893	0.681843	0.617312	0.643386	0.567847	0.831836	0.909333	0.935644	631.12
7	0.472888	0.870543	0.822323	0.801058	0.467303	0.812580	0.629333	0.933168	632.37
8	0.336398	0.907296	0.891800	0.892063	0.379039	0.871630	0.885333	0.928218	633.91
9	0.276944	0.919912	0.922551	0.928042	0.381237	0.875481	0.856000	0.876238	632.59
10	1.053258	0.681843	0.850797	0.971429	0.768648	0.518614	0	1	629.80
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:46:56	0.657009		0.690909		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	5.502038	0.476131	0.885457	0.052632	6.230278	0.481910	1	0	567.81
2	7.808516	0.481711	1	0	7.945990	0.481910	1	0	562.98
3	7.949045	0.481711	1	0	7.945991	0.481910	1	0	561.96
4	7.949046	0.481711	1	0	7.945991	0.481910	1	0	562.73
5	7.949045	0.481711	1	0	7.945992	0.481910	1	0	558.20
6	7.949046	0.481711	1	0	7.945991	0.481910	1	0	563.72
7	7.949045	0.481711	1	0	7.945991	0.481910	1	0	563.61
8	7.949046	0.481711	1	0	7.945990	0.481910	1	0	563.36
9	7.949045	0.481711	1	0	7.945991	0.481910	1	0	564.31
10	7.949046	0.481711	1	0	7.945991	0.481910	1	0	563.30
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:36:42	8.683501		0.433824		1		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	4.499171	0.544193	0.058824	0.689189	3.886141	0.540171	0	0.873418	480.24
2	3.900005	0.540541	0	0.705405	3.893409	0.540171	0	1	477.89
3	3.881378	0.540541	0	0.795946	3.871913	0.540171	0	0.911392	477.80
4	3.854360	0.540541	0	0.885135	3.836397	0.540171	0	0.920886	476.26
5	3.789613	0.540541	0	0.791892	3.811267	0.540171	0	0.575949	475.64
6	3.753958	0.540541	0	0.872973	3.764019	0.540171	0	0.971519	478.04
7	3.748483	0.540541	0	0.851351	3.725858	0.540171	0	0.933544	476.99
8	3.711550	0.540541	0	0.902703	3.710262	0.540171	0	0.870253	479.80
9	3.672090	0.540541	0	0.927027	3.716189	0.540171	0	0.892405	477.23
10	3.687132	0.540541	0	0.948649	3.886905	0.540171	0	0.727848	478.01
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:24:05	4.096508		0.486807		0		0.840108		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	7.098271	0.518124	0.038922	0.959405	7.286393	0.524917	0	1	494.74
2	7.281577	0.525231	0	1	7.286392	0.524917	0	1	491.32
3	7.281578	0.525231	0	1	7.286393	0.524917	0	1	488.19
4	7.281577	0.525231	0	1	7.286394	0.524917	0	1	487.61
5	7.281577	0.525231	0	1	7.286394	0.524917	0	1	492.77
6	7.281577	0.525231	0	1	7.286393	0.524917	0	1	491.79
7	7.281576	0.525231	0	1	7.286394	0.524917	0	1	491.70
8	7.281577	0.525231	0	1	7.286394	0.524917	0	1	488.04
9	7.281577	0.525231	0	1	7.286393	0.524917	0	1	490.32
10	7.281578	0.525231	0	1	7.286392	0.524917	0	1	491.02
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:25:43	7.264939		0.526316		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.583716	0.522849	0.063211	0.268456	7.712473	0.531207	0	0	588.69
2	7.712475	0.530564	0	0	7.712473	0.531207	0	0	587.24
3	7.712475	0.530564	0	0	7.712473	0.531207	0	0	589.27
4	7.712475	0.530564	0	0	7.712473	0.531207	0	0	587.70
5	7.712475	0.530564	0	0	7.712473	0.531207	0	0	588.33
6	7.712474	0.530564	0	0	7.712473	0.531207	0	0	587.39
7	7.712475	0.530564	0	0	7.712473	0.531207	0	0	585.94
8	7.712475	0.530564	0	0	7.712473	0.531207	0	0	586.77
9	7.712474	0.530564	0	0	7.712473	0.531207	0	0	586.20
10	7.712475	0.530564	0	0	7.712473	0.531207	0	0	587.49
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:41:00	7.712474		0.483660		0		0		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	6.743059	0.496250	0.092932	0.192584	7.712474	0.477372	0	0	558.99
2	7.712475	0.477500	0	0	7.712474	0.477372	0	0	554.11
3	7.712475	0.477500	0	0	7.712474	0.477372	0	0	554.98
4	7.712475	0.477500	0	0	7.712474	0.477372	0	0	556.08
5	7.712475	0.477500	0	0	7.712474	0.477372	0	0	557.60
6	7.712475	0.477500	0	0	7.712474	0.477372	0	0	557.89
7	7.712475	0.477500	0	0	7.712474	0.477372	0	0	557.68
8	7.712475	0.477500	0	0	7.712474	0.477372	0	0	559.81
9	7.712475	0.477500	0	0	7.712474	0.477372	0	0	558.38
10	7.712475	0.477500	0	0	7.712474	0.477372	0	0	559.19
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
1:34:46	7.712474		0.459016		0		0		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
1:34:52	6.021151	0.513422	0.166667	0.473351

D.2 Modellkonfigurationen auf Basis der augmentierten Daten

Im nachfolgenden werden die Ergebnisse der sechs besten Modellkonfigurationen auf Basis der augmentierten Daten dargestellt. Die hierzu erzeugten Tabellen sind nach dem gleichen Schema aufgebaut wie im vorherigen Kapitel.

D.2.1 Adam sigmoid ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.207696	0.957853	0.555809	0.985386	0.108760	0.957485	0.474667	0.996788	3064.05
2	0.060212	0.979750	0.823462	0.992269	0.102533	0.960651	0.856000	0.964418	3064.39
3	0.050925	0.983044	0.863326	0.993222	0.097512	0.963817	0.877333	0.969113	3062.54
4	0.056957	0.979169	0.856492	0.990469	0.092783	0.969697	0.722667	0.995799	3060.05
5	0.038210	0.983916	0.876993	0.992905	0.083471	0.968566	0.880000	0.977267	3062.02
6	0.042106	0.984691	0.891800	0.993222	0.074615	0.971054	0.821333	0.984680	3059.09
7	0.035576	0.988567	0.913440	0.994917	0.072155	0.974898	0.810667	0.989375	3059.80
8	0.035086	0.987598	0.910023	0.994917	0.088028	0.970828	0.829333	0.984927	3066.17
9	0.029819	0.990020	0.924829	0.995446	0.091968	0.968792	0.757333	0.988386	3061.22
10	0.042190	0.984207	0.899772	0.991740	0.113231	0.957259	0.914667	0.960217	3058.42
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:34:37	0.045225		0.984887		1		0.984211		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.203140	0.946908	0.356499	0.991027	0.083459	0.968838	0.693694	0.993300	2715.37
2	0.060217	0.980624	0.831403	0.994496	0.082180	0.967050	0.798799	0.982133	2706.69
3	0.052209	0.981390	0.846847	0.993659	0.081800	0.971903	0.792793	0.990508	2712.38
4	0.036283	0.986973	0.898327	0.995334	0.069548	0.975479	0.792793	0.992462	2710.31
5	0.035812	0.986973	0.903475	0.994855	0.103535	0.960409	0.915916	0.962870	2711.15
6	0.033180	0.987739	0.908623	0.995932	0.073805	0.972669	0.831832	0.987437	2711.59
7	0.034180	0.989053	0.915058	0.995932	0.080732	0.971648	0.798799	0.986879	2744.99
8	0.031774	0.988725	0.916345	0.995095	0.075346	0.973946	0.840841	0.988275	2704.51
9	0.033513	0.988396	0.917632	0.994496	0.087560	0.972669	0.822823	0.987437	2706.80
10	0.031228	0.988396	0.921493	0.994736	0.083795	0.969604	0.741742	0.990229	2709.68
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:41:01	0.076246		0.971049		0.694915		0.992208		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.225120	0.948635	0.561208	0.980655	0.099063	0.963340	0.646840	0.998106	2382.00
2	0.068873	0.979304	0.791733	0.994453	0.101613	0.967704	0.535316	0.978535	2371.63
3	0.049107	0.984416	0.861685	0.996077	0.120609	0.949084	0.855019	0.966856	2374.28
4	0.046697	0.983917	0.856916	0.995400	0.084473	0.974978	0.736059	0.979798	2378.09
5	0.038280	0.986286	0.882353	0.994724	0.187793	0.923480	0.947955	0.929609	2372.62
6	0.043063	0.983917	0.864865	0.993777	0.141554	0.942101	0.940520	0.947917	2372.65
7	0.051025	0.981673	0.855326	0.992965	0.080725	0.970323	0.728625	0.991162	2369.72
8	0.040601	0.985289	0.882353	0.994048	0.137093	0.946174	0.929368	0.953598	2377.57
9	0.043041	0.984665	0.872814	0.993506	0.079987	0.970614	0.784387	0.984533	2376.15
10	0.033275	0.987034	0.890302	0.994724	0.089115	0.967122	0.862454	0.979482	2378.55
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:53:00	0.063026		0.980632		0.866324		0.994038		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.247230	0.942009	0.314371	0.992552	0.098127	0.968125	0.559441	0.990205	2390.45
2	0.065270	0.980132	0.793413	0.994448	0.114193	0.955665	0.888112	0.954818	2384.50
3	0.054725	0.982739	0.851796	0.993230	0.070283	0.974210	0.821678	0.993997	2384.20
4	0.048330	0.985099	0.877245	0.993907	0.069794	0.974210	0.779720	0.983570	2382.89
5	0.048480	0.982615	0.862275	0.992552	0.082184	0.968994	0.912587	0.970616	2386.28
6	0.039053	0.987458	0.908683	0.994177	0.080267	0.970733	0.825175	0.983570	2387.41
7	0.042223	0.985968	0.890719	0.994042	0.072813	0.974790	0.835664	0.986730	2389.01
8	0.036075	0.986837	0.898204	0.994042	0.126234	0.949290	0.944056	0.950711	2385.30
9	0.031235	0.988327	0.911677	0.994719	0.107199	0.959722	0.919580	0.961453	2386.78
10	0.037357	0.985720	0.896707	0.994177	0.091052	0.966097	0.909091	0.972196	2382.13
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:52:45	0.074777		0.972229		0.924925		0.976757		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.153450	0.961870	0.642225	0.990603	0.090194	0.968338	0.671598	0.994779	2883.25
2	0.055793	0.982426	0.839444	0.993735	0.086541	0.969297	0.792899	0.987210	2883.01
3	0.054256	0.982323	0.847029	0.992952	0.078985	0.976013	0.784024	0.994257	2882.48
4	0.042743	0.985303	0.884956	0.993735	0.086239	0.966179	0.863905	0.973114	2886.59
5	0.036144	0.987770	0.901391	0.995078	0.095338	0.965219	0.920118	0.974419	2885.80
6	0.049146	0.982014	0.876106	0.990715	0.105673	0.959463	0.914201	0.972331	2885.90
7	0.050306	0.985098	0.887484	0.992393	0.222631	0.935476	0.973373	0.972853	2883.16
8	0.049524	0.985817	0.859671	0.992728	0.094339	0.967138	0.866864	0.966327	2882.02
9	0.040412	0.986742	0.900126	0.993623	0.142680	0.942912	0.946746	0.942313	2886.54
10	0.031793	0.988078	0.908976	0.994854	0.067720	0.974334	0.863905	0.980945	2887.01
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:07:04	0.055836		0.979853		0.854430		0.992568		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.205151	0.954944	0.623037	0.988873	0.097505	0.965208	0.596330	0.989391	2708.60
2	0.051377	0.982789	0.854712	0.995573	0.078796	0.973395	0.773700	0.980179	2701.79
3	0.050236	0.981912	0.848168	0.994257	0.074127	0.973651	0.828746	0.986879	2699.51
4	0.044331	0.985091	0.876963	0.995214	0.083603	0.970581	0.678899	0.995533	2707.50
5	0.046289	0.984324	0.879581	0.993778	0.093999	0.966999	0.785933	0.982133	2706.70
6	0.042571	0.985420	0.888743	0.993898	0.098202	0.964697	0.816514	0.979900	2698.67
7	0.048271	0.982679	0.873037	0.992821	0.103551	0.960604	0.892966	0.967616	2687.18
8	0.037361	0.986187	0.892670	0.994736	0.092174	0.965208	0.816514	0.977387	2685.23
9	0.037156	0.986516	0.903141	0.994257	0.098025	0.970325	0.697248	0.994417	2684.00
10	0.035031	0.988051	0.910995	0.995214	0.112444	0.959580	0.886850	0.967337	2686.66
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:39:35	0.042092		0.986033		0.928571		0.992208		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
7:38:01	0.059534	0.979114	0.878194	0.988665

D.2.2 Adam sigmoid sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.315663	0.914931	0	1	0.291141	0.915197	0	1	3069.61
2	0.285062	0.914931	0	1	0.278429	0.915197	0	1	3066.67
3	0.272951	0.914931	0	1	0.268155	0.915197	0	1	3065.53
4	0.261011	0.914931	0	1	0.257890	0.915197	0	1	3062.79
5	0.249861	0.914931	0	1	0.249719	0.915197	0	1	3064.27
6	0.239925	0.914931	0	1	0.243033	0.915197	0	1	3068.40
7	0.228896	0.914931	0	1	0.231982	0.915197	0	1	3063.81
8	0.220751	0.914931	0	1	0.225041	0.915197	0	1	3064.64
9	0.210379	0.914931	0	1	0.226809	0.915197	0	1	3062.75
10	0.204421	0.914931	0	1	0.202889	0.915197	0	1	3065.39
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:33:42	0.132254		0.957179		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.317183	0.914943	0	0.999880	0.291281	0.914943	0	1	2720.05
2	0.289903	0.914943	0	1	0.286800	0.914943	0	1	2713.41
3	0.280583	0.914943	0	1	0.275667	0.914943	0	1	2712.75
4	0.270469	0.914943	0	1	0.268253	0.914943	0	1	2712.03
5	0.259468	0.914943	0	1	0.257297	0.914943	0	1	2712.31
6	0.249600	0.914943	0	1	0.248958	0.914943	0	1	2710.77
7	0.239429	0.914943	0	1	0.239752	0.914943	0	1	2711.82
8	0.230633	0.914943	0	1	0.232632	0.914943	0	1	2708.57
9	0.222557	0.914943	0	1	0.225071	0.914943	0	1	2709.01
10	0.214931	0.914943	0	1	0.217659	0.914943	0	1	2710.29
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:40:48	0.197497		0.928830		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.295047	0.920459	0	0.992154	0.269019	0.921734	0	1	2387.16
2	0.264908	0.921581	0	1	0.261170	0.921734	0	1	2376.09
3	0.255138	0.921581	0	1	0.247771	0.921734	0	1	2381.49
4	0.236298	0.921581	0	1	0.231997	0.921734	0	1	2380.07
5	0.216480	0.921581	0	1	0.215137	0.921734	0	1	2376.91
6	0.201932	0.921581	0	1	0.202309	0.921734	0	1	2375.98
7	0.190523	0.921581	0	1	0.191679	0.921734	0	1	2375.98
8	0.174208	0.921581	0	1	0.179462	0.921734	0	1	2382.25
9	0.164106	0.921581	0	1	0.166319	0.921734	0	1	2378.98
10	0.156102	0.921581	0	1	0.159476	0.921734	0	1	2381.49
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:51:35	0.174477		0.904633		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.342496	0.917050	0	0.998510	0.285798	0.917125	0	1	2393.21
2	0.286591	0.917050	0	1	0.286987	0.917125	0	1	2392.09
3	0.283427	0.917050	0	1	0.277053	0.917125	0	1	2393.05
4	0.265034	0.917050	0	1	0.253851	0.917125	0	1	2388.59
5	0.240026	0.917050	0	1	0.233833	0.917125	0	1	2387.69
6	0.220133	0.917050	0	1	0.222639	0.917125	0	1	2389.09
7	0.200524	0.917050	0	1	0.193402	0.917125	0	1	2389.67
8	0.172065	0.917050	0	1	0.172541	0.917125	0	1	2391.40
9	0.148324	0.917050	0	1	0.161069	0.917125	0	1	2387.91
10	0.131384	0.917050	0.008982	1	0.142847	0.917125	0	1	2388.43
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:53:26	0.136284		0.917431		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.300714	0.915005	0.006321	1	0.275760	0.918925	0	1	2885.91
2	0.269718	0.918705	0	1	0.265296	0.918925	0	1	2889.11
3	0.260174	0.918705	0	1	0.256314	0.918925	0	1	2889.74
4	0.250679	0.918705	0	1	0.248334	0.918925	0	1	2888.32
5	0.240736	0.918705	0	1	0.243217	0.918925	0	1	2885.70
6	0.233486	0.918705	0	1	0.235898	0.918925	0	1	2885.60
7	0.224554	0.918705	0	1	0.225552	0.918925	0	1	2889.90
8	0.217259	0.918705	0	1	0.219395	0.918925	0	1	2887.88
9	0.206345	0.918705	0	1	0.214116	0.918925	0	1	2892.89
10	0.199207	0.918705	0	1	0.208882	0.918925	0	1	2890.90
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:07:35	0.233306		0.903541		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.289287	0.916246	0	0.994736	0.243811	0.916347	0	1	2715.43
2	0.213891	0.916246	0	1	0.203394	0.916347	0	1	2704.31
3	0.174985	0.916246	0	1	0.170669	0.916347	0	1	2710.69
4	0.148480	0.916246	0.280105	1	0.175001	0.916347	0.914373	1	2709.94
5	0.126170	0.929840	0.827225	1	0.144995	0.916347	0.547401	1	2706.92
6	0.109154	0.977856	0.844241	1	0.120962	0.970837	0.816514	0.994975	2706.87
7	0.093477	0.984652	0.861257	0.998325	0.132098	0.960860	0.908257	0.978504	2691.48
8	0.079605	0.986845	0.899215	0.998086	0.109020	0.970069	0.788991	0.990508	2690.22
9	0.075056	0.986955	0.896597	0.996530	0.099165	0.973651	0.810398	0.989671	2686.80
10	0.068378	0.986955	0.901832	0.996171	0.095430	0.974930	0.758410	0.996092	2691.89
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:39:47	0.091778		0.973264		0.688776		0.998268		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
7:37:49	0.160933	0.930813	0.114796	0.999711

D.2.3 Adam sigmoid softmax

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.564528	0.914931	0	1	0.514731	0.915197	0	1	3065.99
2	0.477315	0.914931	0	1	0.443417	0.915197	0	1	3114.09
3	0.417773	0.914931	0	1	0.394538	0.915197	0	1	3060.70
4	0.376969	0.914931	0	1	0.360766	0.915197	0	1	3056.00
5	0.348862	0.914931	0	1	0.337605	0.915197	0	1	3057.16
6	0.329687	0.914931	0	1	0.321857	0.915197	0	1	3053.85
7	0.316658	0.914931	0	1	0.311266	0.915197	0	1	3061.72
8	0.307877	0.914931	0	1	0.304043	0.915197	0	1	3066.09
9	0.301926	0.914931	0	1	0.299128	0.915197	0	1	3065.79
10	0.297966	0.914931	0	1	0.295874	0.915197	0	1	3066.23
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:34:57	0.211078		0.957179		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.563931	0.910454	0.009009	0.997846	0.518955	0.914943	0	1	2721.69
2	0.484600	0.914943	0	1	0.453195	0.914943	0	1	2715.18
3	0.428489	0.914943	0	1	0.405801	0.914943	0	1	2716.59
4	0.388273	0.914943	0	1	0.372071	0.914943	0	1	2714.80
5	0.359516	0.914943	0	1	0.347982	0.914943	0	1	2713.81
6	0.339037	0.914943	0	1	0.330868	0.914943	0	1	2721.28
7	0.324502	0.914943	0	1	0.318731	0.914943	0	1	2717.69
8	0.314169	0.914943	0	1	0.310086	0.914943	0	1	2716.78
9	0.306893	0.914943	0	1	0.304003	0.914943	0	1	2719.10
10	0.301780	0.914943	0	1	0.299716	0.914943	0	1	2722.26
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:42:40	0.272920		0.928830		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.569820	0.915597	0.007949	0.992424	0.528100	0.921734	0	1	2391.16
2	0.495290	0.921581	0	1	0.464560	0.921734	0	1	2387.03
3	0.439854	0.921581	0	1	0.416653	0.921734	0	1	2390.00
4	0.398094	0.921581	0	1	0.380644	0.921734	0	1	2388.79
5	0.366750	0.921581	0	1	0.353683	0.921734	0	1	2386.99
6	0.343203	0.921581	0	1	0.333237	0.921734	0	1	2377.34
7	0.325587	0.921581	0	1	0.318007	0.921734	0	1	2372.77
8	0.312392	0.921581	0	1	0.306586	0.921734	0	1	2376.40
9	0.302541	0.921581	0	1	0.298222	0.921734	0	1	2394.60
10	0.295177	0.921581	0	1	0.291849	0.921734	0	1	2392.59
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:52:59	0.323187		0.904633		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.570503	0.913697	0.002994	0.996886	0.529313	0.917125	0	1	2398.80
2	0.496985	0.917050	0	1	0.467007	0.917125	0	1	2402.41
3	0.442913	0.917050	0	1	0.420254	0.917125	0	1	2396.78
4	0.402242	0.917050	0	1	0.385485	0.917125	0	1	2396.43
5	0.371896	0.917050	0	1	0.359255	0.917125	0	1	2397.22
6	0.349319	0.917050	0	1	0.340049	0.917125	0	1	2395.65
7	0.332554	0.917050	0	1	0.325405	0.917125	0	1	2397.38
8	0.320060	0.917050	0	1	0.314861	0.917125	0	1	2396.08
9	0.310768	0.917050	0	1	0.306862	0.917125	0	1	2398.79
10	0.303881	0.917050	0	1	0.300858	0.917125	0	1	2397.05
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:56:00	0.300301		0.917431		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.562728	0.918705	0	1	0.514742	0.918925	0	1	2896.95
2	0.478387	0.918705	0	1	0.445482	0.918925	0	1	2892.51
3	0.419468	0.918705	0	1	0.395797	0.918925	0	1	2891.64
4	0.377535	0.918705	0	1	0.360811	0.918925	0	1	2895.10
5	0.348322	0.918705	0	1	0.336479	0.918925	0	1	2892.46
6	0.327721	0.918705	0	1	0.319350	0.918925	0	1	2894.21
7	0.313498	0.918705	0	1	0.307633	0.918925	0	1	2893.56
8	0.303619	0.918705	0	1	0.299253	0.918925	0	1	2895.73
9	0.296702	0.918705	0	1	0.293743	0.918925	0	1	2893.90
10	0.291968	0.918705	0	1	0.289548	0.918925	0	1	2890.81
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:08:33	0.320130		0.903541		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.575027	0.916246	0	0.997368	0.528851	0.916347	0	1	2716.15
2	0.493238	0.916246	0	1	0.460640	0.916347	0	1	2714.20
3	0.434688	0.916246	0	1	0.411138	0.916347	0	1	2714.41
4	0.392506	0.916246	0	1	0.375455	0.916347	0	1	2716.68
5	0.362264	0.916246	0	1	0.349917	0.916347	0	1	2714.34
6	0.340484	0.916246	0	1	0.331680	0.916347	0	1	2711.40
7	0.324889	0.916246	0	1	0.318491	0.916347	0	1	2717.39
8	0.313715	0.916246	0	1	0.309079	0.916347	0	1	2710.77
9	0.305810	0.916246	0	1	0.302506	0.916347	0	1	2710.51
10	0.300186	0.916246	0	1	0.297782	0.916347	0	1	2712.09
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:41:06	0.287378		0.921788		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
7:39:22	0.285832	0.922234	0	1

D.2.4 Adam softmax sigmoid

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.309704	0.909892	0.006834	0.993858	0.283782	0.915197	0	1	3011.76
2	0.278093	0.914931	0	1	0.272425	0.915197	0	1	3007.60
3	0.266994	0.914931	0	1	0.262827	0.915197	0	1	3005.29
4	0.256740	0.914931	0	1	0.254800	0.915197	0	1	3005.78
5	0.246168	0.914931	0	1	0.243513	0.915197	0	1	3006.90
6	0.236589	0.914931	0	1	0.234188	0.915197	0	1	3009.19
7	0.227010	0.914931	0	1	0.226475	0.915197	0	1	3009.27
8	0.216775	0.914931	0	1	0.219628	0.915197	0	1	3012.31
9	0.208543	0.914931	0	1	0.212836	0.915197	0	1	3008.98
10	0.201202	0.914931	0	1	0.204726	0.915197	0	1	3005.62
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:24:08	0.133347		0.957179		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.306199	0.914943	0	1	0.252475	0.914943	0	1	2659.33
2	0.232437	0.914943	0	1	0.222595	0.914943	0	1	2657.70
3	0.201153	0.914943	0	1	0.198717	0.914943	0	1	2659.51
4	0.170526	0.914943	0	1	0.168347	0.914943	0	1	2659.68
5	0.149007	0.914943	0	1	0.152939	0.914943	0	1	2660.20
6	0.133729	0.917132	0.025740	1	0.136797	0.961430	0.570571	0.997767	2664.79
7	0.117185	0.965517	0.604891	0.999043	0.123928	0.967050	0.627628	0.998604	2665.31
8	0.100897	0.985878	0.853282	0.998205	0.115847	0.972925	0.801802	0.988833	2667.21
9	0.089708	0.988068	0.889318	0.997248	0.139977	0.959387	0.900901	0.964824	2666.55
10	0.085533	0.986097	0.876448	0.996291	0.112618	0.970115	0.672673	0.997767	2666.81
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:32:15	0.108657		0.969039		0.593220		0.997836		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.304642	0.913851	0.001590	0.991477	0.269814	0.921734	0	1	2342.89
2	0.265975	0.921581	0	1	0.261184	0.921734	0	1	2339.71
3	0.258151	0.921581	0	1	0.256090	0.921734	0	1	2336.38
4	0.249584	0.921581	0	1	0.247225	0.921734	0	1	2335.69
5	0.242083	0.921581	0	1	0.239668	0.921734	0	1	2336.51
6	0.234046	0.921581	0	1	0.237978	0.921734	0	1	2337.49
7	0.225911	0.921581	0	1	0.226270	0.921734	0	1	2336.50
8	0.217853	0.921581	0	1	0.218940	0.921734	0	1	2337.87
9	0.211492	0.921581	0	1	0.213323	0.921734	0	1	2339.71
10	0.204457	0.921581	0	1	0.207833	0.921734	0	1	2340.00
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:43:55	0.229794		0.904633		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.299442	0.910592	0.008982	0.992146	0.262977	0.917125	0	1	2352.37
2	0.243856	0.917050	0	1	0.232620	0.917125	0	1	2351.62
3	0.215374	0.917050	0	1	0.206199	0.917125	0	1	2347.49
4	0.189497	0.917050	0	1	0.184410	0.917125	0	1	2349.30
5	0.166385	0.917050	0	1	0.167193	0.917125	0	1	2350.59
6	0.144620	0.917050	0	1	0.151200	0.917125	0	1	2349.18
7	0.128234	0.917050	0	1	0.141957	0.917125	0	1	2342.79
8	0.111806	0.976655	0.729042	0.999052	0.123390	0.974500	0.790210	0.991153	2343.99
9	0.106144	0.983112	0.827844	0.997156	0.134176	0.964938	0.902098	0.970616	2344.39
10	0.096551	0.982615	0.842814	0.995261	0.113472	0.972472	0.800699	0.987994	2346.89
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:45:19	0.109393		0.976692		0.819820		0.990811		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.312449	0.918705	0	1	0.275632	0.918925	0	1	2839.00
2	0.270761	0.918705	0	1	0.267008	0.918925	0	1	2837.20
3	0.255950	0.918705	0	1	0.241335	0.918925	0	1	2840.37
4	0.224825	0.918705	0	1	0.216085	0.918925	0	1	2839.63
5	0.195257	0.918705	0	1	0.192279	0.918925	0	1	2837.75
6	0.171424	0.918705	0	1	0.168288	0.918925	0	1	2837.91
7	0.153335	0.918705	0	1	0.162524	0.918925	0	1	2836.17
8	0.131883	0.918705	0	1	0.142183	0.918925	0	1	2835.80
9	0.117503	0.925385	0.082174	1	0.130797	0.918925	0	1	2830.80
10	0.106878	0.971326	0.677623	0.997315	0.121335	0.972655	0.816568	0.986427	2830.49
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:58:21	0.123089		0.973748		0.791139		0.993243		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.307430	0.916246	0	1	0.252926	0.916347	0	1	2657.84
2	0.218282	0.916246	0	1	0.199919	0.916347	0	1	2658.40
3	0.167170	0.916246	0	1	0.162758	0.916347	0	1	2659.07
4	0.135917	0.924030	0.096859	0.999641	0.137392	0.965464	0.636086	0.995533	2664.11
5	0.111020	0.978404	0.781414	0.996411	0.116763	0.972883	0.770642	0.991346	2663.60
6	0.090342	0.984652	0.848168	0.997128	0.104483	0.974162	0.810398	0.989112	2662.93
7	0.078634	0.985091	0.874346	0.995214	0.107204	0.965208	0.856269	0.975154	2664.07
8	0.068241	0.986516	0.882199	0.996052	0.138285	0.940650	0.923547	0.942211	2663.80
9	0.060338	0.987174	0.893979	0.995693	0.096353	0.966488	0.877676	0.974595	2665.70
10	0.054695	0.989805	0.904450	0.997607	0.086418	0.970837	0.828746	0.983808	2662.51
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:33:10	0.068402		0.978452		0.775510		0.995671		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
7:29:31	0.128781	0.959957	0.496615	0.996260

D.2.5 Adam softmax tanh

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.403148	0.914640	0	0.999682	0.290540	0.915197	0	1	3004.48
2	0.291733	0.914931	0	1	0.292005	0.915197	0	1	3000.21
3	0.291679	0.914931	0	1	0.290408	0.915197	0	1	3003.29
4	0.291585	0.914931	0	1	0.290632	0.915197	0	1	3000.19
5	0.292100	0.914931	0	1	0.290367	0.915197	0	1	3002.51
6	0.292070	0.914931	0	1	0.290920	0.915197	0	1	3006.13
7	0.291810	0.914931	0	1	0.291117	0.915197	0	1	3006.31
8	0.291857	0.914931	0	1	0.290361	0.915197	0	1	3008.18
9	0.292333	0.914931	0	1	0.293143	0.915197	0	1	3012.20
10	0.291945	0.914931	0	1	0.290383	0.915197	0	1	3010.91
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:24:01	0.191786		0.957179		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.428334	0.913191	0.002574	0.997846	0.292097	0.914943	0	1	2670.34
2	0.291525	0.914943	0	1	0.291533	0.914943	0	1	2662.00
3	0.291370	0.914943	0	1	0.291319	0.914943	0	1	2660.70
4	0.291809	0.914943	0	1	0.291665	0.914943	0	1	2661.91
5	0.291247	0.914943	0	1	0.291734	0.914943	0	1	2662.39
6	0.292149	0.914943	0	1	0.290999	0.914943	0	1	2663.19
7	0.292472	0.914943	0	1	0.291099	0.914943	0	1	2665.79
8	0.292102	0.914943	0	1	0.291174	0.914943	0	1	2666.90
9	0.291450	0.914943	0	1	0.291312	0.914943	0	1	2669.99
10	0.291719	0.914943	0	1	0.290975	0.914943	0	1	2672.85
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:32:45	0.257641		0.928830		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.428776	0.916843	0.006359	0.994318	0.274939	0.921734	0	1	2352.90
2	0.275413	0.921581	0	1	0.274813	0.921734	0	1	2350.18
3	0.275505	0.921581	0	1	0.274516	0.921734	0	1	2349.31
4	0.276167	0.921581	0	1	0.274696	0.921734	0	1	2342.39
5	0.275495	0.921581	0	1	0.275043	0.921734	0	1	2336.38
6	0.275260	0.921581	0	1	0.276232	0.921734	0	1	2336.11
7	0.275477	0.921581	0	1	0.276312	0.921734	0	1	2337.09
8	0.276554	0.921581	0	1	0.274532	0.921734	0	1	2339.00
9	0.275429	0.921581	0	1	0.274528	0.921734	0	1	2339.58
10	0.275798	0.921581	0	1	0.275514	0.921734	0	1	2339.98
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:44:40	0.320603		0.904633		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.430021	0.915808	0.001497	0.998510	0.239991	0.917125	0	1	2357.99
2	0.214097	0.917050	0	1	0.201170	0.917125	0	1	2355.39
3	0.172829	0.917050	0	1	0.174833	0.917125	0	1	2355.30
4	0.145553	0.917050	0	1	0.135223	0.917125	0	1	2354.46
5	0.124887	0.943748	0.336826	0.998646	0.126548	0.973051	0.755245	0.992733	2356.52
6	0.106828	0.980132	0.791916	0.997156	0.114910	0.972762	0.863636	0.982622	2358.57
7	0.091189	0.984230	0.842814	0.997021	0.099085	0.976239	0.825175	0.989889	2351.41
8	0.080674	0.983484	0.844311	0.996073	0.151981	0.936540	0.940559	0.936177	2350.10
9	0.075488	0.981125	0.809880	0.996615	0.105713	0.964938	0.839161	0.976303	2346.38
10	0.073864	0.981746	0.824850	0.995938	0.090841	0.967835	0.776224	0.985150	2346.41
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:45:54	0.089122		0.971733		0.807808		0.986486		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.388868	0.914080	0.007585	0.994295	0.281470	0.918925	0	1	2840.90
2	0.282710	0.918705	0	1	0.281468	0.918925	0	1	2837.00
3	0.282505	0.918705	0	1	0.281587	0.918925	0	1	2838.89
4	0.282482	0.918705	0	1	0.281423	0.918925	0	1	2839.80
5	0.283039	0.918705	0	1	0.281448	0.918925	0	1	2840.09
6	0.282527	0.918705	0	1	0.283264	0.918925	0	1	2841.59
7	0.282548	0.918705	0	1	0.282211	0.918925	0	1	2838.08
8	0.282575	0.918705	0	1	0.282129	0.918925	0	1	2841.25
9	0.283520	0.918705	0	1	0.281646	0.918925	0	1	2841.91
10	0.282824	0.918705	0	1	0.282281	0.918925	0	1	2835.88
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:58:50	0.322069		0.903541		0		1		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.403111	0.916137	0	0.999880	0.287753	0.916347	0	1	2656.17
2	0.288236	0.916246	0	1	0.288120	0.916347	0	1	2655.29
3	0.288758	0.916246	0	1	0.290473	0.916347	0	1	2656.78
4	0.288490	0.916246	0	1	0.287636	0.916347	0	1	2657.82
5	0.288993	0.916246	0	1	0.287607	0.916347	0	1	2658.36
6	0.288963	0.916246	0	1	0.287670	0.916347	0	1	2662.94
7	0.289283	0.916246	0	1	0.288119	0.916347	0	1	2664.17
8	0.288226	0.916246	0	1	0.288183	0.916347	0	1	2663.48
9	0.288821	0.916246	0	1	0.287629	0.916347	0	1	2662.79
10	0.288278	0.916246	0	1	0.289178	0.916347	0	1	2665.49
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:31:56	0.275018		0.921788		0		1		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
7:29:41	0.242707	0.931284	0.134635	0.997748

D.2.6 SGD sigmoid ReLU

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.263095	0.913962	0.017084	0.996294	0.237041	0.917910	0.098667	1	3092.00
2	0.172830	0.929949	0.266515	1	0.171273	0.943238	0.458667	0.999012	3093.30
3	0.124070	0.958047	0.553531	0.999153	0.134771	0.959294	0.632000	0.991105	3100.27
4	0.096407	0.968123	0.656036	0.998517	0.141707	0.957938	0.792000	0.978256	3099.28
5	0.079602	0.974227	0.733485	0.997458	0.109758	0.965626	0.730667	0.988634	3098.51
6	0.075817	0.974130	0.756264	0.995446	0.110006	0.963139	0.770667	0.986657	3104.38
7	0.067402	0.977425	0.779043	0.995976	0.100061	0.961782	0.632000	0.990610	3103.94
8	0.066207	0.977715	0.782460	0.996082	0.093173	0.965174	0.698667	0.990363	3103.35
9	0.064310	0.977037	0.797267	0.994281	0.087978	0.969697	0.781333	0.985421	3101.82
10	0.057902	0.980138	0.820046	0.995235	0.114962	0.959068	0.850667	0.968866	3103.80
Testmarkt: Markt A									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:40:11	0.067934		0.982368		0.970588		0.982895		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.272322	0.909031	0.006435	0.995573	0.232324	0.914943	0	1	2745.82
2	0.182870	0.923043	0.030888	0.999641	0.180901	0.923116	0.027027	0.999721	2737.02
3	0.132885	0.951067	0.359073	0.998804	0.145459	0.950958	0.465465	0.991904	2738.40
4	0.103086	0.965408	0.597169	0.997248	0.121373	0.951980	0.459459	0.997767	2739.16
5	0.086682	0.971319	0.680824	0.995932	0.115359	0.955045	0.483483	0.998046	2739.13
6	0.077163	0.975698	0.720721	0.997128	0.108974	0.958365	0.528529	0.997208	2736.38
7	0.074729	0.976245	0.761905	0.995573	0.109054	0.963218	0.765766	0.980737	2738.88
8	0.062834	0.979748	0.794080	0.996171	0.094180	0.966794	0.744745	0.987996	2740.01
9	0.063289	0.978544	0.782497	0.996530	0.115628	0.959642	0.843844	0.967895	2742.08
10	0.064143	0.977340	0.779923	0.994975	0.089046	0.970115	0.804805	0.986041	2744.32
Testmarkt: Markt B									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:46:12	0.083766		0.968637		0.717514		0.987446		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.257307	0.914849	0.003180	0.999188	0.216305	0.921734	0	1	2412.15
2	0.174664	0.930433	0.066773	1	0.170454	0.947047	0.260223	0.997790	2408.79
3	0.130511	0.951004	0.340223	0.999188	0.143484	0.938318	0.193309	0.999053	2407.91
4	0.105889	0.962349	0.510334	0.998647	0.120165	0.956648	0.468401	0.995896	2400.42
5	0.098053	0.966962	0.596184	0.996889	0.146885	0.952866	0.750929	0.964015	2401.97
6	0.088245	0.970951	0.647059	0.997430	0.106318	0.960722	0.557621	0.993056	2401.80
7	0.072542	0.976936	0.731320	0.996618	0.108537	0.962758	0.743494	0.980114	2403.58
8	0.070261	0.977185	0.732909	0.996212	0.107547	0.963340	0.780669	0.977588	2402.90
9	0.070389	0.974941	0.732909	0.995130	0.131322	0.952575	0.855019	0.957071	2404.20
10	0.061939	0.978556	0.767886	0.996077	0.090430	0.966250	0.617100	0.995265	2402.89
Testmarkt: Markt C									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:55:27	0.087528		0.968865		0.670951		0.998645		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.273270	0.911834	0.004491	0.992282	0.213104	0.917125	0.006993	1	2420.68
2	0.174394	0.927605	0.190120	1	0.169615	0.925819	0.139860	1	2420.76
3	0.130252	0.950950	0.471557	0.999729	0.149887	0.962330	0.699301	0.993049	2420.23
4	0.104175	0.965355	0.630240	0.998375	0.114729	0.957114	0.545455	0.995577	2418.27
5	0.086857	0.973550	0.718563	0.998104	0.128705	0.942915	0.325175	0.999684	2418.69
6	0.080775	0.972681	0.721557	0.996886	0.102465	0.962040	0.629371	0.993049	2402.18
7	0.076518	0.974171	0.739521	0.996209	0.107539	0.955665	0.493007	0.998104	2396.31
8	0.068959	0.977151	0.776946	0.996209	0.108989	0.959432	0.811189	0.974724	2391.49
9	0.062337	0.979511	0.800898	0.996750	0.098249	0.963489	0.790210	0.980727	2387.20
10	0.062572	0.979014	0.803892	0.995938	0.090670	0.966676	0.821678	0.981991	2376.00
Testmarkt: Markt D									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
6:54:47	0.082049		0.973221		0.837838		0.987297		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.256157	0.913977	0.015171	0.993735	0.210504	0.919645	0.071006	1	2882.09
2	0.163153	0.937513	0.283186	0.998322	0.153866	0.940034	0.325444	0.997651	2879.40
3	0.110251	0.962076	0.532238	0.999329	0.139142	0.940513	0.289941	1	2879.18
4	0.089674	0.970195	0.668774	0.997203	0.119086	0.960422	0.713018	0.981206	2880.80
5	0.081418	0.971326	0.699115	0.996308	0.143059	0.954665	0.860947	0.968677	2887.41
6	0.073693	0.975540	0.747155	0.996532	0.105732	0.963540	0.766272	0.981728	2893.99
7	0.069694	0.977903	0.764855	0.996980	0.106443	0.964980	0.801775	0.979901	2899.49
8	0.063026	0.980987	0.793932	0.996980	0.103713	0.964020	0.855030	0.974419	2903.62
9	0.059626	0.980473	0.805310	0.995861	0.091407	0.965219	0.775148	0.981989	2898.07
10	0.057632	0.980062	0.805310	0.995637	0.084224	0.970257	0.674556	0.995040	2888.81
Testmarkt: Markt E									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
8:07:23	0.095906		0.965201		0.626582		0.998649		

Epoche	Training				Test				Dauer
	BCE	Accuracy	Recall OOS	Recall In-Stock	BCE	Accuracy	Recall OOS	Recall In-Stock	
1	0.257787	0.913944	0.010471	0.993778	0.208203	0.917370	0.021407	1	2707.37
2	0.161853	0.936966	0.291885	1	0.163795	0.931440	0.220183	0.999721	2700.22
3	0.118697	0.957575	0.536649	0.998804	0.151601	0.952162	0.645260	0.982691	2700.76
4	0.098119	0.967332	0.636126	0.997966	0.149459	0.951138	0.749235	0.974874	2699.49
5	0.087546	0.971278	0.700262	0.997727	0.112035	0.957534	0.571865	0.994137	2701.49
6	0.076335	0.975115	0.736911	0.996889	0.129636	0.954720	0.804281	0.971803	2703.11
7	0.074050	0.975115	0.757853	0.994975	0.099654	0.962139	0.626911	0.994137	2706.89
8	0.068713	0.977856	0.770942	0.996171	0.101407	0.964697	0.779817	0.981575	2714.38
9	0.064005	0.977417	0.781414	0.995573	0.096163	0.963929	0.645260	0.994417	2721.90
10	0.058903	0.981254	0.821990	0.996052	0.106236	0.961371	0.828746	0.973758	2728.99
Testmarkt: Markt F									
Generierungsdauer	BCE		Accuracy		Recall OOS		Recall In-Stock		
7:40:11	0.058866		0.980048		0.806122		0.993939		

Durchschnittliche Modelltestergebnisse				
Generierungsdauer	BCE	Accuracy	Recall OOS	Recall In-Stock
7:40:42	0.079341	0.973057	0.771599	0.991478

Erklärung über die selbständige Abfassung der Arbeit

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht.

Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Troisdorf, 03.05.2023,

(Ort, Datum, Unterschrift)

A handwritten signature in black ink, appearing to read 'D. J. J.', written over a horizontal line.

