Universität Hamburg
Fakultät für Mathematik,
Informatik und Naturwissenschaften
Department Informatik

Masterarbeit

Inkrementelles Lernen mit Ensembles aus neuronalen Netzen für variierende Datensätze in diskreten Zeitschritten

Felix Fröhlich

3froehli@informatik.uni-hamburg.de
Studiengang Master of Science Informatik
Matr.-Nr. 6523076
Fachsemester 6

Erstgutachter Universität Hamburg:	Prof. Dr. Chris Biemann
Zweitgutachter Universität Hamburg:	Dr-Ing. Gregor Wiedemann

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Wir haben die Möglichkeit zur nahezu ständigen, sozialen Teilhabe im Internet. Die Grundlage dafür haben hauptsächlich Social-Media-Applikationen und zum Teil auch Nachrichtenseiten mit Diskussionsplattformen geschaffen. Sie stellen eine technologische Infrastruktur bereit, die es den Nutzern erlaubt, beispielsweise Kommentare und Kritiken in textueller Form zu verfassen oder Fotos und Videos zu veröffentlichen. Die gewöhnlich leichte Bedienbarkeit und die oftmals durchführbare Nutzung ohne ersichtliche Kosten, bilden niedrige Hürden für Einsteiger ihre Daten zu teilen.
Die Betreiber solcher Plattformen werden dadurch mit einem kontinuierlichen Strom an Daten konfrontiert, der skalierbare Systeme und Ressourcen für die Persistenz erfordert. Die stetig wachsenden Datenmengen bieten jedoch auch das Potential, durch Analysen und Algorithmen, neues Wissen zu generieren und das Nutzererlebnis zu verbessern.
Daten-getriebene Web-Applikationen stellen dynamische Inhalte dar, die abhängig von Nutzereingaben und den gesammelten Informationen sind. Der derzeitige Wandel von statischen Web-Seiten zu dynamischen Web-Applikationen bringt neue Anforderungen an die verarbeitenden Algorithmen mit sich. Modelle des maschinellen Lernens werden nicht mehr einmalig trainiert, sondern über einen zeitlichen Verlauf hinweg und aktualisieren ihre Parameter in jedem Zeitschritt. Die durch Algorithmen unterstützte Moderation von Nutzerinhalten erfordert eine schnelle Adaption der Modelle, da sprachliche Ausdrücke und Modewörter einem ständigen Wandel unterliegen.
Inkrementelle Lernmethoden stellen eine Alternative zum klassischen Trainingsprozess, bei dem das Modell einmalig mit einer Datenbasis für eine festgelegte Anzahl von Epochen trainiert wird, dar. Stattdessen wird der Algorithmus beim inkrementellen Lernen in diskreten Zeitschritten mit Teildatensätzen trainiert, die sich über den zeitlichen Verlauf verändern. Es bedarf somit einer Klasse von Algorithmen, die in mehreren Zeitschritten trainiert werden kann.
Algorithmen, die mit zeitbedingten Daten arbeiten, müssen zwei wesentliche Merkmale aufweisen: Zum einen muss Wissen aus neuen Daten schnell gelernt werden und zum anderen darf altes Wissen nicht zu schnell vergessen werden, da auch Datenpunkte aus vorherigen Zeitschritten wertvolle Informationen für zukünftige Prädiktionen bereitstellen können.
Künstliche Intelligenz rückt nun schon seit einigen Jahren in den Fokus der Öffentlichkeit, durch die beachtlichen Ergebnisse neuronaler Netze. Die, teils übermenschliche, Performanz von Deep-Learning-Netzen, wie sie bei AlphaGo beobachtet wurde, offenbart ihre Potentiale für weitreichende Anwendungsgebiete. Die Bedeutung von KI wird

längst von Unternehmen weltweit anerkannt und findet bereits zahlreichen Bereichen eingesetzt Anwendung. Auch die erwähnten Social-Media-Plattformen nutzen intelligente Algorithmen, um beispielsweise Nutzerverhalten voraussagen zu können oder unangemessene Inhalte zu filtern.
Die herausragende Performanz neuronaler Netze, die mit großen Datenmengen trainieren, stellt die Frage nach ihrer Eignung für inkrementelles Lernen auf einer zeitlichen Achse. Ein Algorithmus für inkrementelles Lernen weist nach Losing, Hammer und Wersing folgende drei Merkmale auf [Losing u. a. 2016]: Erstens ist der Lernvorgang im Zeitschritt t durch das Lernen im vorherigen Zeitschritt beeinflusst und das Modell wird nicht von Grund auf neu trainiert.Zweitens behält der Algorithmus zuvor gelerntes Wissen und vergisst es nicht im Verlauf des Lernprozesses. Drittens hat der Algorithmus eine begrenzte Aufnahmeapazität und kann sich nicht einfach den gesamten Datenstrom merken. Inkrementelle Lern-Algorithmen, die nicht auf neuronalen Netzen basieren, wie beispielsweise Incremental Support Vector Machines [Cauwenberghs u. Poggio 2001] oder On-Line Random Forest [Saffari u. a. 2009], sind zum Teil Erweiterungen klassischer Lernverfahren.
Verschiedene inkrementelle Lernverfahren mit neuronalen Netzen werden in der Arbeit von Karaoguz verglichen [Karaoguz 2018]. Darin zeigt sich, dass inkrementelles Lernen mit Data Pooling, ein Verfahren, das in einem späteren Kapitel erläutert wird, die beste Performanz erzielt. Beim Data Pooling besteht die Problematik, dass das neuronale Netz zur Überanpassung neigt, da das Training in jedem Zeitschritt mit allen zuvor gesammelten Datenpunkten durchgeführt wird. Das neuronalen Netz erhält Datenpunkte aus früheren Zeitschritten öfter als Eingabe und passt seine Kantengewichte dementsprechend an. Ältere Datenpunkte werden auswendig gelernt und das Netz macht schlechtere Prognosen für unbekannte Daten. Neues Wissen wird schlechter adaptiert.
Die Anwendung eines einzelnen Netzes mit Data Pooling für zeit-abhängige Webplattformen mit kontinuierlichen Datenströmen birgt Risiken hinsichtlich Stabilität und Plastizität, allein durch zugrunde liegende Eigenschaften neuronaler Netze, die in einem späteren Kapitel erläutert werden. Daher stellt die Frage nach weiteren Methoden, um Prädikatoren zu trainieren.
Ensemble-Modelle bestehen aus mehreren Algorithmen, deren einzelne Ergebnisse auf verschiedene Weisen aggregiert werden. Im Vergleich zu einzelnen Prädikatoren, wie einem neuronalen Netz, sind sie stabiler und liefern bessere Vorhersagen [Lessmann u. a. 2015]. Die Gewinner des Netflix-Preises für Film-Empfehlungen, erzielen die herausragende Treffergenauigkeit ihres Modells durch die Qualität der einzelnen Algorithmen, aber auch vor allem durch den Ensemble-Ansatz [Töscher u. Jahrer 2009].
Die häufigsten Verfahren zur Kombination verschiedener Modelle sind Bagging, Boosting und Stacking.
Ensembles aus neuronalen Netzen für inkrementelles Lernen sind bereits 2001 mit dem Algorithmus Learn++ vorgestellt worden [Polikar u. a. 2000]. Die Grundlage sind da-

bei sogenannte schwache Lerner, Modelle, die nur gering bessere Performanz erzielen, als eine zufällige Klassenzuordnung. Die Vorhersagen dieser schwachen Lerner, die aus einfachen neuronalen Netzen bestehen, werden zu einer finalen Vorhersage kombiniert [Polikar u. a. 2001]. Dieses Ensemble-Verfahren ist inspiriert durch den Adaboost-Algorithmus, auf den in einem späteren Kapitel eingegangen wird.
Learn++ verwendet keine komplexen, tiefen neuronalen Netze wie LSTMs oder CNNs, die ebenfalls in späteren Kapiteln noch erklärt werden, sondern Multi Layer Perceptrons (MLP). MLPs sind vollständig verbundene Netze, doch die Erfolge von AlpaGo und Co basieren auf tiefen, raffinierten Netzstrukturen mit speziellen Schichten.
Auch die Erweiterung Learn++.MT erreichte beispielsweise nur schwache Performanz bezüglich eines Optical-Character-Recognition-Datasets. Stattdessen scheinen SVM-Learn++ und SVMLearn++.MT bessere Werte zu erzielen, doch diese Algorithmen basieren auf Support Vector Machines. [Hulley u. Marwala 2007]
Statt einzelne, schwache MLP-Lerner zu kombinieren, stellt sich die Frage, ob sich Ensembles aus tiefen neuronalen Netzen mit komplexen Strukturen sich für inkrementelles Lernen eignen. Dabei ist insbesondere das Stacking-Verfahren von Interesse, das mehrere Schichten von Klassifikatoren bildet und bereits in klassischen Lernverfahren erfoglreiche Anwendung findet [Wolpert 1992]. Diese Arbeit konzentriert sich vorrangig auf textuelle Daten und deren Klassifikation durch neuronale Netze. Das betrifft vor allem die Klassifikation von Nutzerkommentaren, wobei es sich um die Zuordnung zu einer Stimmung handelt.
Ensembles aus komplexen, tiefen neuronalen Netzen und ihre Performanz hinsichtlich inkrementeller Lernverfahren sind Hauptuntersuchungsgegenstand dieser Arbeit. Dabei wird besonders auf die Arbeit von Karaoguz und den darin behandelten inkrementellen Lernmethoden Bezug genommen. Auf Stacking basierende Ensembles werden mit einzelnen neuronalen Netzen verglichen, die mit einem Datenpool trainieren, da diese Lernmethode die besten Resultate erzielte. Die Datenströme werden durch eine Reihe von Datenblöcken in zeitlicher Reihenfolge simuliert. Dadurch lernen die Netze in diskreten Zeitschritten, die in der Regel Tagen entsprechen. Ein solcher Versuchsaufbau lässt sich am ehesten auf eine vergleichbare Situation in der Realität übertragen und würde nur eine minimale Adaption für die Umsetzung bedeuten. Im folgenden Abschnitt wird der Hauptuntersuchungsgegenstand in kleinere Forschungsfragen granuliert, die versuchen genügend unterschiedlichen Aspekte des Ensemble-Ansatzes hervorzuheben.

1.2 Zielsetzung

Viele wissenschaftliche Arbeiten befassen sich bereits mit gängigen Algorithmen für inkrementelles Lernen, wie Incremental Support Vector Machines, die nicht auf neuronalen Netzen basieren, sowie den Learn++-Varianten mit einfachen, neuronalen Netzen als schwache Lerner.
Ensembles aus komplexeren, neuronalen Netzen für inkrementelles Lernen in diskreten Zeitschritten, bilden einen weiteren, möglichen Ansatz und sind der Hauptuntersuchungsgegenstand dieser Arbeit.
Inkrementelle Lernverfahren neuronaler Netze, wie adaptives Lernen mit einem Datenpool, eignen sich als Vergleichsmodelle für die Performanz, um die Evaluationsmetriken einordnen zu können.
Folgende Forschungsfragen werden im Rahmen dieser Arbeit untersucht:

- Können Ensemble-Modell aus neuronalen Deep-Learning-Netzen vergleichbare oder bessere Vorhersagen als ein neuronales Netz mit Data Pooling im Bereich des inkrementellen Lernens liefern?

- Wie kann die Performanz dieser Ensemble-Modelle für unbalancierte Datensätze im Vergleich zum einzelnen neuronalen Netz bewertet werden? Klassifizieren Ensembles unterrepräsentierte Klassen besser oder schlechter als ein einzelnes Netz?

- Welchen Einfluss haben die Basisnetze als Bestandteil des Ensembles auf die Performanz? Wie wirkt sich die Anzahl der einzelnen Netze auf die Vorhersagequalität aus?

- Sind heterogene Basisnetze grundsätzlich besser als Ensembles aus homogenen Basisnetzen? Ist es wichtiger das Ensemble aus verschiedenen Modellen wie CNNs oder LSTMs zusammenzusetzen oder ist die Diversität der Basisnetze von geringerer Bedeutung? Welche Metriken werden davon besondern beeinflusst?

- Welche inkrementelle Lernmethode mit Ensembles aus neuronalen Netzen führt zu besseren Vorhersagen? Welchen Einfluss hat die jeweilige Methode auf unterschiedliche Evaluationsmetriken?

- Wie kann der zeitliche Aufwand beurteilt werden, den das Training von Ensemble-Modellen betrifft? Gibt es Möglichkeiten Ensemble-Modelle schneller zu trainieren als einen einzelnen Klassifikator?

- Können die zu untersuchenden Lernverfahren mit Ensembles ähnlich gute Klassifikatoren erzeugen wie klassische Methoden mit einer Trainings- und Testmenge?

1.3 Aufbau der Arbeit

In diesem Abschnitt werden die einzelnen Bestandteile dieser Arbeit der Reihenfolge nach erläutert, um einen Überblick zu geben und Orientierungspunkte zu setzen.
Nach dem Einleitungsteil, bestehend aus den Unterkapiteln *Motivation*, *Zielsetzung* und *Aufbau der Arbeit*, beginnt das Grundlagen-Kapitel.
Das Grundlagen-Kapitel vermittelt die theoretischen, grundlegenden Konzepte und Algorithmen, die Bestandteile dieser Arbeit sind. Dabei bilden Inkrementelles Lernen, Ensemble-Lernen und neuronale Netze die drei Hauptbestandteile des Grundlagenkapitels.
Das folgende, dritte Kapitel *Verwandte Arbeiten* befasst sich mit wissenschaftlichen Publikationen, die ähnliche Thematiken wie die vorliegende Arbeit aufgreifen. Die Inhalte der verwandten Arbeiten sind in solcher Weise relevant, dass sie Einfluss auf Entscheidungen und Experimente haben, die diese Arbeit betreffen. Zudem bilden sie dafür einen Referenzraum und untersützen die Einordnung der untersuchten Thematik.
In dem vierten Kapitel *Inkrementelles Lernen mit Ensembles aus neuronalen Netzen* werden die durchzuführenden Experimente aufgelistet und erläutert. Weiterhin werden die verwendeten Datensätze und Metriken vorgestellt, die diese Experimente betreffen.
Kapitel fünf behandelt die Evaluation der zuvor ausgeführten Experimente und präsentiert deren Ergebnisse.
Das sechste Kapitel umfasst eine abschließends Betrachtung. Dieses beantwortet die Forschungsfragen aus Kapitel ein und bewertet die Ergebnisse der Experimente aus dem vorherigen Kapitel. Darüber hinaus gibt das letzte Kapitel einen Anstoß bezüglich möglicher zukünftiger Arbeiten, die auf den Erkenntnissen dieser Arbeit aufbauen und die Erforschung der Thematik fortführen könnten.

2 Grundlagen

Inhalt dieses Kapitels ist die Erläuterung der grundlegenden Verfahren und Algorithmen, die die Basis für die Experimente in Kapitel vier bilden. Die drei Themenschwerpunkte sind Inkrementelles Lernen, Ensemble-Lernen und neuronale Netze.

2.1 Inkrementelles Lernen

Während beim klassischen Lernverfahren das jeweilige Modell, wie z.B. ein neuronales Netz, einmalig trainiert wird, bezieht sich inkrementelles Lernen (IL) auf eine Folge von Trainingsschritten. Der Prädikator erhält in diskreten Zeitschritten variierende Datensätze mit denen er kontinuierlich trainiert. Anwendung findet inkrementelles Lernen in Umgebungen, die kontinuierlich Daten generieren. Auch bei Systemen mit wiederholender Abfolge von Nutzeraktionen und dem daraus resultierenden Modeltraining (Human-in-a-loop), ist inkrementelles Lernen notwendig.
Die Herausforderung bei diesem Verfahren ist es, neues Wissen zu adaptieren und bereits Gelerntes nicht zu vergessen.
Inkrementelles Lernen kann auf verschiedene Weisen durchgeführt werden.

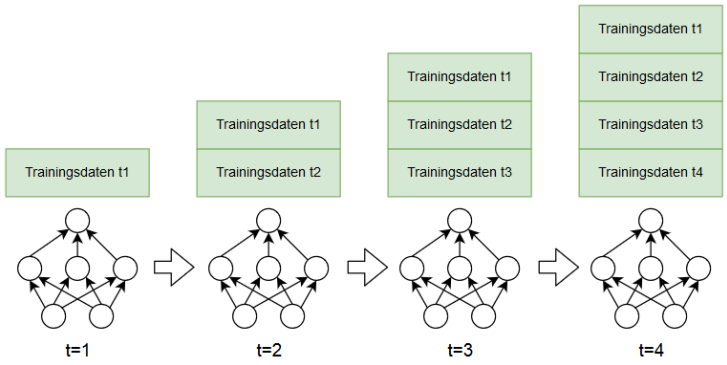

Abbildung 2.1: Inkrementelles Lernen mit Daten-Pooling

Beim IL mit Daten-Pooling trainiert das entsprechende Modell mit einem anwachsenden Datensatz. In jedem Zeitschritt werden Daten zu einer Menge hinzugefügt, die dem Prädikator als Trainingseingabe dient. Dabei werden gelernte Parameter oder Attribute, wie Kantengewichte neuronaler Netze, nach jedem Zeitschritt beibehalten. Dieses Vorgehen

bietet den Vorteil, dass Wissen über ältere Datenpunkte beibehalten wird, da diese Teile der in der Datensammlung bestehen bleiben und bei jedem Trainingsschritt genutzt werden. Ein Nachteil ist die Überanpassung bezüglich der älteren Datenpunkte. Die Kantengewichte passen sich eher an frühere Datenmengen an, da sie diese in mehreren Trainingsphasen gesehen haben. Daraus folgt eine schlechtere Adaption neuer, unbekannter Daten. Ein weiterer, offensichtlicher Nachteil ist die steigende Trainingsdauer.
Beim IL ohne Daten-Pooling wird das Training jeweils nur mit den aktuellen Daten durchgeführt, die neu hinzugekommen sind. Auch hierbei werden gelernte Gewichte von einem Trainingsschritt in den nächsten übertragen.

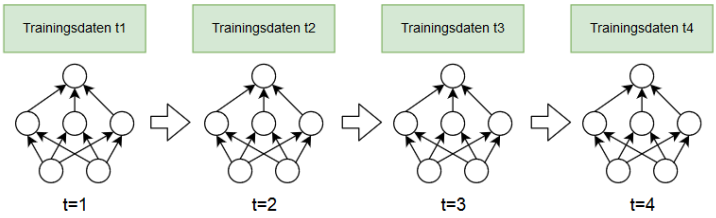

Abbildung 2.2: Inkrementelles Lernen ohne Daten-Pooling

Der Vorteil hierbei ist die konstante Trainingsdauer, da die Datenmenge in ihrer Größe meistens gleich bleibt. Weiterhin neigt das Modell nicht dazu, ältere Datenpunkte auswendig zu lernen, da es diese nur bei jeweils einem Training sieht. Die Nachteile sind dementsprechend zum einen eine mögliche Unteranpassung bei niedriger Epochenanzahl. Die Datenpunkte werden nicht häufig genug zum Trainieren genutzt und die Kantengewichte des Modells können nicht richtig angepasst werden. Zum anderen tendiert das entsprechende Modell dazu, Wissen aus vorherigen Datensätzen zu vergessen und zu überschreiben [Karaoguz 2018].

2.2 Ensemble-Lernen

Ensemble-Lernen ist ein Oberbegriff, der verschiedene Verfahren wie Bagging [Breiman 1994], Boosting [Freund u. Schapire 1996] oder Stacking [Wolpert 1992] umfasst. Ihre Gemeinsamkeit besteht in der Nutzung mehrerer Prädikatoren, denen je nach Implementation ganz unterschiedliche Algorithmen zugrunde liegen können. Durch die Kombination diverser Klassifikatoren können individuelle Stärken genutzt und individuelle Schwächen in ihrer Wirkung vermindert werden. In den folgenden Abschnitten werden die einzelnen Verfahren beschrieben, wobei insbesondere auf das Stacking eingegangen wird, da ein Untersuchungsgegenstand dieser Arbeit ist.

2.2.1 Bootstrap Aggregating (Bagging)

Bagging ist ein Akronym für Bootstrap Aggregating und wurde von Leo Breiman in seinem Paper *Bagging Predictors* [Breiman 1994] dargelegt. Das grundlegende Bootstrap-Verfahren [Efron u. Tibshirani 1994] sei wie folgt definiert:
Sei L die Grundmenge an Trainingsdaten, der eine statistische Verteilung mit einer theoretischen Verteilungsfunktion F zugrunde liegt. Nun werden k verschiedene Teilmengen L_k zufällig aus der Grundmenge extrahiert, die Rückschlüsse auf eine zugrunde liegende Verteilung erlauben. Dabei können einzelne Datenpunkte mehrfach entnommen werden, da sie Teil der Grundmenge bleiben. Aus den verschiedenen Teilmengen lässt sich eine empirische Verteilungsfunktion berechnen, die sich im besten Fall der theoretischen Verteilungsfunktion annähert.
Bagging nutzt eine Menge von k Prädikatoren $\{p(x,L_k)\}$, wobei x die Eingabe und L_k ein partieller Teiltrainingsdatensatz aus der Grundmenge L ist. Die Ausgaben $\{y_k\}$ der Prädikatoren werden mit unterschiedlichen Verfahren aggregiert. Eine Form der Aggregation ist zum Beispiel das arithmetische Mittel der Ausgabewerte.
Breiman untersuchte Bagging für verschiedene Machine-Learning-Modelle, wie zum Beispiel dem Nearest Neighbor Klassifikator und beobachtete einen Anstieg des Wertes für die Treffergenauigkeit (engl. Accuracy). Bagging lässt sich leicht implementieren, wenn ein Machine-Learning-Modell bereits vorhanden ist, wobei die Gesamtstruktur dadurch komplexer und schwieriger zu interpretieren ist. Die Verwendung mehrerer Prädikatoren für Trainingsteilmengen reduziert Varianz und damit eine Überanpassung and den Datensatz. Ein weiterer Vorteil, den Bagging bietet, ist seine einfache Parallelisierbarkeit. Die Teilmodelle können parallel trainiert werden, da keine Abhängigkeiten untereinander existieren.

2.2.2 Boosting

Das Boosting-Verfahren ist insbesondere durch den AdaBoost-Algorithmus (Adaptive Boosting) [Freund u. Schapire 1996] bekannt geworden. Ähnlich wie beim Bagging werden zuerst zufällig Teilmengen aus dem Traningsdatensatz entnommen und darauf unterschiedliche Prädikatoren trainiert. Diese Prädikatoren sind schwache Lerner, d.h. sie sind nur ein wenig besser als der Zufall. Während beim Bagging ein paralleles Trainieren möglich ist, verläuft das Boosting-Verfahren sequentiell. Dies hat den folgenden Grund: Zu Beginn sind alle Datenpunkte eines Datensatzes gleich gewichtet. Nachdem ein erster Prädikator einen Datensatz verarbeitet hat, werden falsch klassifizierte Datenpunkte mit höheren Gewichten versehen. Dadurch werden nachfolgende Prädikatoren in die Richtung gelenkt, zuletzt falsch klassifizierte Datenpunkte stärker zu berücksichtigen und

dadurch ihre Vorhersagen dementsprechend zu verbessern. Das Gewicht eines Datenpunktes aktualisiert sich nach der folgenden Formel:

$$w_{t+1}(i) = \frac{w_t(i)}{Z} e^{-\alpha^t h^t(x) y(x)} \tag{2.1}$$

Dabei entspricht ht dem aktuellen, schwachen Lerner, Alpha dem Gewicht des Lerners selbst und Z einer Konstante, die die Datenpunktgewichte normalisiert.
Nachdem eine Menge von schwachen Lernern sequentiell trainiert wurde, bilden ihre Hypothesen eine finale Hypothese H(x):

$$H(x) = sign(\alpha^1 h^1(x) + \alpha^2 h^2(x) + ... + \alpha^T h^T(x)) \tag{2.2}$$

2.2.3 Stacking

Stacking oder Stacked Generalization [Wolpert 1992] nutzt mehrere, hierarchisch angenordnete Prädikatoren. Das Verfahren eignet sich für Klassifikation und Regression. Dabei bilden die Vorhersagen der unteren Basis-Prädikatoren neue Trainingsdaten, die höheren Meta-Prädikatoren als Eingabe dienen. Obere Hierarchieebenen müssen daher nicht an besondere Eigenschaften der ursprünglichen Eingabedaten angepasst werden und können aus einfachen Algorithmen, wie der logistischen Regression, bestehen. Dies ist insbesondere bei variablen Datenpunktlängen der Fall, sofern der entsprechende Basis-Prädikator die Eingabe nicht auf eine Ausgabesequenz abbildet. Die Ausgabe eines Basis-Prädikators besteht im Falle einer Klassifikation in den meisten Fällen aus vorhergesagten Klasse und einer dazugehörigen Wahrscheinlichkeit.
Die Out-Of-Fold-Vorhersagen eines Prädikators bilden die Datenbasis für einen Meta-Klassifikator. Ähnlich wie bei der K-fold-Cross-Validation werden Datenpunkte und zugeordnete Klassen in gleich große, disjunkte Teilmengen segmentiert, deren genaue Anzahl durch einen vorher definierter Parameter definiert wird. Bei *n* Teilmengen wird ein temporärer Basis-Klassifikator auf jeweils *n-1* Teilmengen trainiert und trifft Vorhersagen für die *n*-te, übrige Teilmenge. Die prognostizierten Klassenlabel auf Basis der ausgeschlossenen Teilmenge, werden pro Iteration zusammengefügt und entsprechen am Ende der Anzahl der Trainingspunkte. Sie bilden damit die Traningsdaten des Meta-Klassifikators.
Neben der Prognose für die, ihm unbekannten Teilmenge, liefert jeder temporäre Out-Of-Fold-Klassifikator auch eine Vorraussage für das gesamte Testset. Die Werte werden am Ende gemittelt und dienen als Testdaten für den Meta-Klassifikator. Um den Aufwand und die Komplexität zu verringern, können die Vorhersagen für das Testset auch von einem einzelnen Klassifikator getroffen werden, der mit den Trainingsdaten trainiert wurde.

Der Begriff *Blending* findet erstmals im Zusammenhang mit den Gewinnern des Netflix-Preises für Filmempfehlungsalgorithmen [Töscher u. Jahrer 2009] Verwendung. Teilweise werden die Begriffe Stacking und Blending synonym gebraucht, doch beschreibt er auch eine Simplifizierung des Stacking-Verfahrens. Die Vereinfachung liegt im Verzicht von Out-Of-Fold-Vorhersagen. Stattdessen teilt man den Datensatz nur in eine Trainings- und Hold-Out-Menge. Die Trainingsmenge ist für die Basisklassifikatoren bestimmt, die nach dem Training darauf Vorhersagen für die unbekannte Hold-Out-Menge machen.
Wolpert empfiehlt in seiner wissenschaftlichen Arbeit zum Stacking, dass die Basis-Klassifikatoren in ihren Ausprägungen möglichst variieren. Das bezieht statistische Klassifzierungsmethoden mit ein wie generative und diskriminierende Algorithmen. Theoretisch lernen diverse Basisprädikatoren dann unterschiedliche Merkmale eines Datensatzes.

2.3 Neuronale Netze

Der folgende Abschnitt ist den unterschiedlichen Aspekten und Ausprägungen neuronaler Netze wie dem Verzerrung-Varianz-Dilemma und Deep Learning gewidmet. Weiterhin werden spezielle Netztypen beschrieben, die in den Experimenten dieser Arbeit als Basisprädikatoren für die Ensembles Anwendung finden.

2.3.1 Das Verzerrung-Varianz-Dilemma

Beim Training von neuronalen Netzen mit überwachtem Lernen können zwei verschiedene Auffälligkeiten beobachtet werden, die die Performanz der Modelle entscheidend beeinflussen [Geman u. a. 1992].
Verzerrung beschreibt den Zustand, in dem das Netz die Zusammenhänge von Eingabe- und Zielwert nicht lernt und keine sinnvolle Abbildung approximieren kann. Diese Auffälligkeit wird in Bezug auf einen Algorithmus auch als Unteranpassung bezeichnet.
Varianz ist eine Folge von Überanpassung und beschreibt ein Modell, das seine Trainingsdaten nahezu auswendig gelernt hat, aber dafür nicht oder nur schlecht generalisieren kann. Die Vorhersagen für unbekannte Datenpunkte weichen hinsichtlich ihrer Metriken deutlich von den aufgezeichneten Trainingswerten ab, sodass das neuronale Netz für die Klassifizierung oder Regression neuer Daten ungeeignet ist.
Verzerrung und Varianz sind beides Fehlerquellen, die durch den Algorithmus minimiert werden sollen. Dies führt zum dem eigentlichen Verzerrung-Varianz-Dilemma, da neuronale Netze, abhängig von verschiedenen Trainingsparametern, entweder Funktionen lernen, die die Beziehungen von Eingabe- und Zielwerten nur bedingt modellieren, aber

dadurch besser generalisieren können oder die Trainingsdaten auswendig lernen und dadurch schlechte Vorhersagen für unbekannte Datenpunkte machen. Bei einer geringen Zahl von Trainingsepochen neigt das jeweilige Netz eher zu Unteranpassung. Zu viele Trainingsiterationen wiederum führen zur Überanpassung. Die Entscheidung für ein Modell, das die Trainingsdaten genau beschreibt, macht es womöglich ungeeignet für neue, ungesehene Datenpunkte. Die Wahl eines gut generalisierenden Modells führt in der Regel dazu, dass Beziehungen innerhalb der Trainingsdaten nicht richtig gelernt werden.
Das Verzerrung-Varianz-Dilemma macht es notwendig, dass der Trainingsprozess des neuronalen Netzes früzeitig gestoppt wird, um Überanpassung zu vermeiden. Gleichzeitig muss lang genug trainiert werden, um den Bias zu minimieren. Eine Möglichkeit den richtigen Zeitpunkt für das Beenden des Trainings zu finden, ist durch eine unbekannte Testmenge gegeben. Nach jeder Epoche kann das neuronale Netz Vorhersagen für diese Testmenge machen. Diese werden ausgewertet und ein Test-Fehler kann berechnet werden. Sinkt der Testfehler während des Trainings, dann kann das Netz weiter trainieren. Fängt er nach weiteren Epochen plötzlich an zu steigen, sollte das Training abgebrochen werden, damit die Fähigkeit des Netzes zu Generalisieren gewährleistet bleibt.
Eine weitere Möglichkeit, Überanpassung zu verhindern, ist der sogenannte Drop-Out-Wert [Srivastava u. a. 2014]. Dieser gibt an, wie viel Prozent der Neuronenverbindungen während des Trainings zufällig deaktivert werden sollen. Dadurch werden bestimmte Kantengewichte nicht aktualisiert und das Netz verhält sich wie eine Menge von Teilnetzen. Bei jedem Trainingsdurchlauf werden die Neuronenverbindungen wieder neu zufällig ausgeschaltet.

2.3.2 Deep Learning

Ein Begriff, der die Debatten bezüglich des Fortschreites künstlicher Intelligenz und maschinellen Lernens geprägt hat, ist Deep Learning. Er beschreibt Lernverfahren verschiedener Typen von neuronalen Netzen mit höherer Komplexität gegenüber dem einfachen Perzeptron oder flachen, neuronalen Netzen. Flache neuronale Netze zeichnen sich dadurch aus, dass sie lediglich eine Eingabe- und Ausgabeschicht aus Perzeptronen besitzen und dadurch in ihrer Fähigkeit, Funktionen zu approximieren, begrenzt sind.
Ein Deep-Learning-Modell besitzt mindestens eine weitere, „versteckte“ Schicht, wodurch eine Sequenz von Berechnungen nach einer Eingabe erfolgt [Schmidhuber 2014]. Daraus folgt die Einstufung als universelle Funktionsapproximatoren. Ein neuronales Netz mit mindestens einer versteckten Schicht und einer begrenzten Anzahl von Neuronen, kann jede kontinuierliche Funktion im euklidischen Raum approximieren [Csáji 2001].
Ein Klassifikator, basierend auf einem flachen neuronalen Netz, kann nur Klassen tren-

nen, die einfach linear separierbar sind. Schon das logische XOR lässt sich nicht mehr lernen. Durch das Hinzufügen einer weiteren Schicht Neuronen, lernt das Netz zuerst die Features OR und NAND und kombiniert sie dann zu der komplexeren XOR-Funktion.
Tiefe, neuronale Netze lernen in den unteren Schichten nach der Eingabe selber Features. Dadurch kam es beispielsweise im Bereich der Bildverarbeitung zur Deep Learning Revolution [Felsberg 2017].
Feature Engineering, das Erstellen von eigenen Features (wie zum Beispiel die Form eines Ohres zur Erkennung von Säugetieren), die wie Schablonen in diskreten Abständen mit Teilabschnitten eines Eingabebildes multipliziert werden, ist nicht mehr erforderlich. CNNs lernen in unteren Schichten Formen, wie Bögen, und in höheren Schichten komplexe Konstrukte, wie Gesichter.
Die erhöhte Komplexität und Tiefe eines neuronalen Netzes ist verbunden mit einer Vielzahl von Rechenoperationen und damit höherer Laufzeit. Mehrdimensionale, viel Speicherplatz beanspruchende Trainingsdaten, wie Bilder, in Kombination mit Deep Learning führen zu Trainingszeiten von mehreren Wochen [Hara u. a. 2017].
Die Verarbeitung textueller Daten durch neuronale Netze erfolgt deutlich schneller, wobei auch hier die numerische Abstraktion im Vektorraum und die Datenmenge einen deutlichen Einfluss auf raschen Trainingsfortschritt haben. Neben dem stetigen Anstieg der Leistung von Zentralprozessoren und der Möglichkeit Rechenoperationen auf mehreren Prozessor-Kernen zu parallelisieren, werden auch Grafikkarten bzw. GPUs für das Trainieren neuronaler Netze eingesetzt. Da GPUs für Vektor- und Matrizenmultiplikationen optimiert sind, um das Skalieren, Transformieren und Drehen zwei- und dreidimensionale Objekte effizient zu berechnen, eignen sie sich auch für Berechnungen neuronaler Netze [Luo u. a. 2005].

2.3.3 Convolutional Neural Networks (CNNs)

Die Entwicklung der CNNs ist inspiriert durch die Endeckung spezieller Neuronen im visuellen Kortex, durch die Wissenschaftler Torsten Weisel und David Hubel [Hubel u. Wiesel 1968]. Diese Neuronen reagieren individuell auf Teilabschnitte des Sichtfeldes durch das rezeptive Feld, welches wiederum Neuronen repräsentiert, die direkt mit ihnen verbunden sind.
Die spezielle Architektur von CNNs, verbunden mit Backpropagation und selbstlernenden Filtern, fand erstmals Anwendung in der Arbeit von Le Cun für die Erkennung handgeschriebener Postleitzahlen [LeCun u. a. 1989].
Ein gewöhnliches neuronales Netz multipliziert den jeweiligen, gesamten Datenpunkt mit den Kantengewichten seiner Eingabeschicht. Dagegen besteht die Konvolutionssschicht eines CNNs aus einer Menge von Filtern, deren Größe nur einem Teilabschnitt der Eingabe entspricht. Diese Filter, bestehend aus zufällig initialisierten Neuronen, bewegen

sich mit einer fest definierten Schrittweite über den Datenpunkt und multiplizieren ihre Kantengewichte pro Schritt nur mit dem jeweiligen Ausschnitt. Die Größe der Filter muss an die Dimensionen der Eingabedaten angepasst werden. Bei textuellen Daten ist der Filter eindimensional und wandert über die Sequenzlänge, während Filter für die Bildverarbeitung zweidimensional sind und über die Höhe und Breite gleiten. Das Trainieren der Filter erlaubt ein selbstständiges Lernen von Merkmalen, wie beispielsweise Linienformen in der Bilderkennung. Jeder Filter erzeugt bei der Iteration über das Eingabebild eine Filter-Map, die die hervorgehobenen Merkmale des jeweiligen Filters enthält und der nächsten Schicht als Eingabe dient. Nach der Konvolutionsschicht folgt in der Regel eine Pooling-Schicht, die die Größe der Filter-Maps durch die Berechnung eines Maximal- oder Durchschnittswerts für kleinere Segmente reduziert. Dadurch werden zum einen Komplexität und Resourcenbedarf verringert und zum anderen bewirkt es eine Invarianz bezüglich einer Verschiebung. Daher kann bei der Bilderkennung ein Objekt an unterschiedlichen absoluten Koordinaten positioniert sein und trotzdem lokalisiert werden. Bei der Verarbeitung textueller Daten können Wortkombinationen an unterschiedlichen Stellen im Eingabetext vorkommen und den gleichen gelernten Filter aktivieren. Die Kombination von Konvolutions- und Pooling-Schicht kann in fast beliebiger Anzahl miteinder verknüpft werden und bildet dadurch tiefe neuronale Netze, die in jeder weiteren, höheren Schicht komplexere Merkmale lernen. Damit das CNN als Klassifikator funktionieren kann, muss mindenstens eine vollständig verbundene Neuronenschicht and die Konvolutions- und Poolingschichten angeknüpft werden.

2.3.4 Long Short-Term Memory (LSTM)

Gewöhnliche Feed-Forward-Netze, wie das Multi-Layer-Perceptron (MLP) benötigen eine fest definierte Eingabegröße. Variable Eingaben, wie unterschiedlich lange Sequenzen von Datenpunkten, lassen sich nicht verarbeiten. Es können eine feste Sequenzlänge definiert und kürzere Sequenzen mit Standardwerten ergänzt werden, jedoch würde das MLP in diesem Fall variierende Reihenfolgen der gleichen Datenpunkte als unterschiedliche Eingaben interpretieren. Zusammenhänge von zeitlich-aufeinanderfolgenden Daten werden durch MLPs nicht erkannt, denn jede Eingabe wird unabhängig von der vorherigen betrachtet.
Daher etablierten sich Recurrent Neural Networks mit einer Recurrent-Layer, einer zusätzlichen Neuronenschicht, die Wissen aus vorherigen Zeitschritten repräsentiert. Bei jedem Zeitschritt ist diese rekurrente Schicht eine Ergänzung zur momentanen Eingabe.
Nach einer langen Sequenz muss der Backpropagation-Algorithmus auf sämtliche Schichten zu allen Zeitpunkten angewendet werden. Dadurch wird der Fehlerwert entweder verschwindend gering oder nimmt drastisch zu. Dieses Vanishing-Gradient-Problem [Hochreiter 1998] verhindert, dass RNNs größere Zusammenhänge in sequen-

tiellen Daten erkennen können. Der Zusammenhang kurzer Wörter- oder Buchstabenfolgen lässt sich für diese Algorithmen erschließen, doch Anfang und Ende eines Dokumentes kann ein gewöhnliches RNN nicht mehr verbinden.
Die Brisanz des Vanishing-Gradient-Problems wurde von Sepp Hochreiter und Jürgen Schmidhuber erkannt und durch Long Short-Term Memory-Zellen gelöst. Durch sogenannte Gates, Neuronenschichten mit einer Sigmoid-Aktivierungsfunktion, wird kontrolliert welche alten Informationen das Netz vergessen und welche neuen Informationen es integrieren soll [Hochreiter u. Schmidhuber 1997].
LSTM-Zellen verfügen über einen Cell-State und können so Wissen über lange Zeiträume speichern und in einen Kontext setzen. Weiterhin speichert ein Hidden-State Informationen über die jüngsten Glieder einer Sequenzfolge. Hidden-State und Cell-State werden bei jedem Zeitschritt weitergereicht und mit der momentanen Eingabe aktualisiert.
Die erste Komponente der LSTM-Zelle besteht aus einer Schicht, die das Forget-Gate genannt wird. Die aktuelle Eingabe und der vorherige Hidden-State werden mit den Gewichten dieser Schicht multipliziert und durchlaufen die Sigmoid-Funktion. Die Ausgabe entspricht der Größe des Cell-States wobei für jede Cell-State-Position ein Wert zwischen 0 und 1 ausgegeben wird. Ein Wert nahe der 1 bedeutet, dass die Information an dieser Stelle behalten werden soll, während Werte nahe der 0 angeben, dass die zugehörigen Informationen vergessen werden sollen [Olah 2015].

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.3}$$

W_f und b_f stehen für die Gewichtsmatrix und den Bias des jeweiligen Forget-Gates.
Als Nächstes durchlaufen Eingabe und vorheriger Hidden-State die Input-Gate-Schicht, um zu bestimmen, welche Positionen aus dem Cell-State mit neuen Werten aktualisiert werden sollen. Wie das Forget-Gate besteht auch diese Schicht aus Neuronen mit einer Sigmoid-Funktion.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.4}$$

Neben dem Input-Gate werden Eingabe und vorheriger Hidden-State auch mit den Kantengewichten einer weiteren Schicht multipliziert, die eine tanh-Aktivierungsfunktion aufweist. Dadurch werden Kandidatenwerte für den neuen Cell-State berechnet.

$$\tilde{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C) \tag{2.5}$$

Die Summe aus der Ausgabe des Forget-Gates multipliziert mit dem alten Cell-State und dem Produkt von Input-Gate und Cell-State-Kandidat ergibt den neuen Cell State.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.6}$$

Nun muss nur noch der neue Hidden-State berechnet werden. Die dafür zuständige Output-Gate-Schicht berechnet sich wieder aus aktueller Eingabe, dem alten Hidden-State und einer dritten Sigmoid-Aktivierungsfunktion. Anschließend werden die Ergebnisse mit dem Cell-State, der eine tanh-Funktion durchläuft, multipliziert.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.7}$$

$$h_t = o_t * tanh(C_t) \tag{2.8}$$

Die Formeln sind im wesentlichen aus dem Artikel von Olah [Olah 2015] übernommen. Durch die, hier formalisierten, Gates wird das Problem des verschwindenden Gradienten beseitigt und die Netze können lange Sequenzlängen lernen. In den späteren Experimenten dieser Arbeit werden LSTM-Netze als Basis-Klassifikatoren für heterogene Stacking-Ensembles eingesetzt.

2.3.5 Gated Recurrent Units (GRU)

Eine weitere Möglichkeit das Vanishing-Gradient-Problem rekurrenter, neuronaler Netze zu lösen bietet das Gated Recurrent Neural Network [Cho u. a. 2014] [Chung u. a. 2014]. Die Hauptkomponente, die Gated Reccurent Unit (GRU), besteht im Vergleich zum LSTM nur aus zwei Gates, die den Informationsfluss kontrollieren. Das Update Gate kontrolliert welche Informationen aus vorherigen Zeitschritten übernommen werden. Dafür werden die aktuelle Eingabe und der Hidden State jeweils mit ihren Gewichten multipliziert, addiert und dann der Sigmoid-Funktion übergeben.

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \tag{2.9}$$

Das Reset Gate bestimmt, wie viel von den alten Informationen vergessen werden soll und errechnet sich ähnlich, wie die Ausgabe des Update Gates. Nur die Gewichte-Matrizen ändern sich.

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \tag{2.10}$$

Anschließend wird ein vorläufiger, neuer Hidden State ermittelt. Dabei wird aus dem Ergebnis der Multiplikation vom Hidden State mit einer Gewichtsmatrix das Hadamard-Produkt berechnet. Dieses wird es mit dem Produkt aus aktueller Eingabe und Gewichtsmatrix addiert und alles der Tanh-Aktivierungsfunktion übergeben.

$$h_t^{'} = tanh(Wx_t + r_t \circ Uh_{t-1}) \tag{2.11}$$

Im finalen Schritt wird der tatsächliche neue Hidden State aus dem Update Gate, sowie dem zurückliegenden und vorläufigen Hidden State berechnet.

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ h_t^{'} \tag{2.12}$$

Wie bei dem LSTM kann dieser neue Hidden State anschließend mit einer vollständig verbundenen Schicht verknüpft werden, um beispielsweise klassifizieren zu können.

3 Verwandte Arbeiten

3.1 Adaptive Learning Strategies for Neural Paraphrase Generation

Ethem Can Karaoguz untersucht in seiner Masterarbeit neuronale Netze aus dem Deep-Learning-Bereich im Kontext eines simulierten Datenstroms mit adaptiven Lernverfahren. Dabei vergleicht er inkrementell-lernende neuronale Netze mit einem Basis-Netz, dass in jedem Zeitschritt re-initialisiert und neu trainiert wird [Karaoguz 2018].
Die Netze bilden eine Eingabesequenz auf eine Ausgabesequenz ab und lernen Sätze zu paraphrasieren. Das Sequence-to-Sequence- bzw. Encoder-Decoder-Modell besteht aus einer Reihe von LSTMs, die zuerst die Eingabe in einen internen Zustand kodieren und daraus dann die Ausgabe berechnen.
Verschiedene adaptive Verfahren, namentlich inkrementelles, Transfer- und aktives Lernen, werden als Strategie geprüft und hinsichtlich ihrer Tauglichkeit verglichen. Als Metrik wird der BLEU-Score benutzt, der die Qualität einer maschinellen Übersetzung bezüglich Sprachtexten als Ein- und Ausgabe misst.
Der Simulation des Datenstroms erfolgt, indem eine gegebene Datenmenge in zufällig verteilte Untermengen separiert wird. Vier verschiedene Datensätze werden für die Evaluation des Models genutzt. Dabei zeigt sich, dass die BLEU-Metrik sich sowohl beim inkrementellen Lernen, mit und ohne Datenpool, als auch beim Vergleichsmodell über den Zeitverlauf verbessert, bezogen auf die Datenmenge QUORA (Quora Question Pairs)und dem Microsoft Research Paraphrase Corpus (MSR). Dennoch treten in einigen Zeitschritten beim MSR-Datensatz Anomalien auf, in denen sich der Wert kurzzeitig verschlechtert. Das Modell mit Datenpool performt besser als das Vergleichsmodell, welches in jedem Zeitschritt die Daten neu lernt. Weiterhin übertrifft die Performanz des Datenpool-Models das Vergleichsmodell auch, wenn es mit einer geringeren Datenmenge trainiert.
Experimente mit inkrementellem Lernen und Netzwerkerweiterung, bei der nach einer bestimmten Anzahl von Zeitschritten eine neue Neuronenschicht hinzugefügt und die vorderste Schicht nach der Eingabe "eingefroren" wird, zeigen keine Verbesserung im BLEU-Score gegenüber dem Datenpool-Modell in den Metriken.

3.2 Learn++

Der Learn++-Algorithmus basiert im Wesentlichen auf dem, in Kapitel 2 vorgestellten, AdaBoost-Algorithmus mit mehren Modifikationen, sodass er auch auf inkrementelles Lernen angewandt werden kann [Polikar u. a. 2000] [Polikar u. a. 2001] [Polikar u. a. 2002]. Learn++ ist, wie Adaboost, ein Ensemble aus sogenannten schwachen Lernern, die sequentiell trainiert werden. Im Falle einer binären Klassifikation zeichnet sich ein schwacher Lerner dadurch aus, dass er nur etwas mehr als 50 Prozent der Datenpunkte richtig klassifiziert. Dabei trainiert jeder schwache Lerner mit einem Teildatensatz der verfügbaren Datenbasis. Die einzelnen Datenpunkte haben Gewichte, die anfangs mit gleichen Werten initialisiert werden und sich nach jeder Klassifikation ändern. Wird ein Datenpunkt falsch klassifiziert, erhöht sich sein Gewicht und die Wahrscheinlichkeit steigt, dass er für den nächsten Klassifikator als Traingsdatenpunkt ausgewählt wird. Dadurch fokussieren sich die folgenden Klassifikatoren stärker auf die schwer zu lernenden Datenpunkte. Im Gegensatz zu Adaboost, dessen Formel zum Gewichtsupdate (siehe Formel 2.1) nur den vorherigen Klassifikator und dessen Gewichtung miteinbezieht, berechnet sich das neue Gewicht in Learn++ wie folgt:

$$w_{t+1}(i) = w_{t+1}(i) B_t^{1-[|H_t(x_i) \neq y_i|]} \tag{3.1}$$

Dabei ist H_t die zusammengesetzte Hypothese aus den schwachen Lernern und ihren Gewichtungen. B_t wird vorher wie folgt aus E_t, dem normalisierten Fehler der Komposition, berechnet:

$$B_t = \frac{E_t}{1 - E_t} \tag{3.2}$$

Die spezielle Formel 3.1 zur Gewichtsberechnung ist besonders gut geeignet für inkrementelles Lernen und ermöglicht dem Ensemble auch neue, unbekannte Klassen zu erlernen. Das geschieht dadurch, dass die Gewichtsaktualisierung abhängig ist von der jeweiligen zusammengesetzten Hypothese H_t für die Datenpunkte im Zeitschritt t. Neue Datensätze mit unbekannten Klassenlabeln werden dadurch von H_t falsch klassifiziert und mit hoher Wahrscheinlichkeit für den nächsten Trainingsdatensatz ausgewählt. Adaboost's Gewichtsaktualisierung ist in diesem Fall deutlich ineffizienter, da sie nur vom letzten schwachen Lerner abhängig ist und nicht von der gesamten Komposition der Hypothesen.
Während des inkrementellen Lernens, wird für jeden Datensatz zum jeweiligen Zeitpunkt eine zusammengesetzte Hypothese mit gewichteter Mehrheit gebildet. Aus diesen Hypothesen bildet sich schließlich die endgültige Hypothese H_{final}:

$$H_{final} = argmax \sum_{k=1}^{K} \sum_{t:H_t(x)=y} log \frac{1}{B_t} \tag{3.3}$$

Zwar bestehen die schwachen Lerner in den, zu Learn++ zugehörigen, wissenschaftlichen Arbeiten, aus mehrschichtigen Perzeptronen, aber theoretisch, und darauf weisen die Autoren auch hin, können die Prädikatoren durch andere Algorithmen ausgetauscht werden. Da Learn++ nur mit schwachen Lernern trainiert, müssen keine Hyperparameter optimiert werden, wie es bei komplexen, starken Prädikatoren der Fall wäre. Die einzelnen Trainingszeiten sind kürzer als die eines komplexen Klassifikators, der die Entscheidungsgrenze in späteren Zeitschritten feinjustiert und dadurch eher zu Überanpassung neigt, als ein schwacher Lerner, der im Falle einer Klassifikation die Klassen nur grob unterscheidet.

4 Inkrementelles Lernen mit Ensembles aus neuronalen Netzen

In diesem Kapitel erfolgt eine Erläuterung der einzelnen, in den Abschnitten 4.5 und 4.6 vorgestellten, Experimente, die im Rahmen dieser Arbeit durchgeführt werden. Vor den eigentlichen Beschreibungen dieser Experimente werden die Datensätze und die Metriken definiert, die zur Evaluation der Verfahren gebraucht werden.

4.1 Repräsentation der Daten

Die Daten, die in textueller Form vorliegen, werden von einfachen Strings in Wort-Token und anschließend in multidimensionale Wortvektoren umgewandelt. Die genutzten Wortvektoren wurden nicht in dieser Arbeit trainiert. Für diese Arbeit werden FastText-Vektoren ausgewählt, deren Vorteil gegenüber Verfahren wie Word2Vec [Mikolov u. a. 2013] darin besteht, dass Buchstaben als Wortuntermengen in Vektorrepräsentationen umgewandelt werden [Grave u. a. 2017]. Dadurch können auch Wörter vektorisiert werden, die nicht im ursprünglichen Trainingskorpus enthalten waren. Die verwendete, konkrete Softwarebibliothek, die benutzt wurde, um ein bereits vortrainiertes FastText-Model einzubinden, ist das FLAIR-NLP-Framework von Zalando Research [Akbik u. a. 2018]. Die verwendeten Wortvektoren haben jeweils eine Größe von 300 Dimensionen.

4.2 Datensätze

4.2.1 IMDB-Review-Datenset

Das IMDB-Review-Datenset enhält 50.000 Filmbewertungen von denen die eine Hälfte positiv und die andere negativ ist [Maas u. a. 2011]. Dabei gibt es nicht mehr als 30 Bewertungen pro Film. Die Daten sind wiederum in eine Trainings- und eine Testmenge von jeweils 25.000 Datenpunkten unterteilt. Diese Unterteilung dient als Grundlage zur Bewertung der Modelle des dazugehörigen Papers.
Ausschlaggebend für die Auswahl dieser Daten war zum einen die Größe dieses Datensets. Dadurch ermöglicht es die Simulation von mehr Zeitschritten mit jeweils genügend Datenpunkten für das Training der einzelnen Iteration. Zum anderen ist das Datenset

balanciert und bildet damit einen Gegensatz zu den anderen beiden Datensätzen. Die Metriken zur Evaluation binärer Klassifikatoren können einfach berechnet werden.
Zudem liefert die mit dem Datensatz verbundene, wissenschaftliche Arbeit durch ihre evaluierten Modelle wertvolle Vergleichsmaßstäbe. Dadurch können die Ensemble-Modelle nicht nur hinsichtlich inkrementeller Lernverfahren, sondern auch mit klassischen Klassifikatoren verglichen werden.
Jeder Datensatz besteht aus durchschnittlich 238 Wörtern in der Trainingsmenge beziehungsweise 232 Wörtern in der Testmenge.

4.2.2 Yahoo News Annotated Comments Corpus

Diese Sammlung von Kommentaren auf Nachrichtenseiten umfasst ungefähr 522.000 Datenpunkte [Napoles u. a. 2017] von denen jedoch nur ein geringer Anteil mit Klassenlabeln versehen ist.
Die in dieser Arbeit verwendeten, sogenannten Experten-Annotationen umfassen 23.383 Kommentare mit den vier Klassen *Negative*, *Neutral*, *Mixed* und *Positive*. Einige der Datenpunkte sind jedoch mit verschiedenen Labeln gleichzeitig versehen, da die Zuordnung teilweise schwierig ist. Die Datenpunkte sind mit einem Unix-Zeitstempel versehen, sodass dadurch der Datenstrom simuliert werden kann, der realistischer ist, als die zufallsbasierte Einteilung der Filmkritiken des IMBD-Datensates.
Ausreichend Daten für die Trainingszyklen der neuronalen Netze liefern nur die Kommentare in dem Zeitraum vom 01.04.2016 bis zum 05.05.2016.

4.2.3 Facebook Dataset Collection

Diese Sammlung von Facebook-Daten enthält eine Reihe von Artikeln, Kommentaren und dazugehörigen Antworten, die im Zeitraum von Mai bis Juli 2017 verfasst wurden [Wiedemann 2017]. Die Inhalte wurden von unterschiedlichen Medienseiten sowie den Seiten rechter Parteien heruntergeladen. Kriterien für die Auswahl dieser Inhalte waren Schlüsselwörter, wie *Asyl*, *Gewalt* und *Terror*.
Insgesamt beträgt die Anzahl der Kommentare 42.626 und ist in die drei Klassen *None*, *discrimination speech act (DSA)* und *counter speech act (CSA)* unterteilt. Dabei gehören die meisten Datenpunkte zur *None*-Klasse (34.917), während *DSA* und *CSA* jeweils mit 3.930 bzw 3.779 Kommentaren vergleichsweise unterrepräsentiert sind.
Alle Datenpunkte sind mit einem Zeitstempel versehen, sodass eine zeitliche Abfolge sehr gut simuliert werden kann. Die Kommentare vom 28.04.2017 bis zum 18.06.2017 werden verwendet, da hier die jeweilige Datenmenge von geügender Größe ist.

4.3 Evaluationsmetriken

Bevor die einzelnen Metriken Gegenstand dieses Abschnittes werden, wird auf eine Besonderheit des inkrementellen Lernens hingewiesen und die damit verbundenen veränderten Anwendungsmöglichkeiten. Beim klassischen, überwachten Lernen wird ein Modell mit einem Teildatensatz einmalig trainiert und dann auf einer anderen Datenbasis evaluiert. Beim inkrementellen Lernen können die Modelle nach jedem Zeitschritt evaluiert werden. Dabei können beispielsweise die Metriken einer gewöhnlichen Testmenge oder für die Daten des nächsten Zeitschrittes, sofern diese vorhanden sind, berechnet werden. Es ist also nicht ausreichend nach dem Ablauf aller Zeitschritte einmalig Modelle zu evaluieren. Entscheidend ist vielmehr der Werteverlauf entscheidend beziehungsweise der Durchschnittswert dieser Verlaufswerte.
Die Metrik der Treffergenauigkeit oder Vertrauenswahrscheinlichkeit (engl. Accuracy) gibt das Verhältnis von richtig klassifizierten Datenpunkten zur Menge aller Datenpunkte an. Datenpunkte sind richtig klassifiziert, wenn die Vorhersage des Prädikators mit dem vorhandenen Klassenlabel übereinstimmt.

$$\textbf{Treffergenauigkeit} = \frac{\textbf{richtig klassifiziert}}{\textbf{Menge aller Datenpunkte}} \tag{4.1}$$

Die Treffergenauigkeit für alle drei Datensätze wird auf zwei Arten bestimmt:
Im ersten Fall wird das jeweilige Modell nach jedem Zeitschritt mit dem Testdatensatz evaluiert. Anschließend werden die Genauigkeitswerte gemittelt und ergeben einen Durchschnittswert. Im zweiten Fall wird das Modell nach jedem Zeitschritt mit den Daten des folgenden Zeitschrittes evaluiert und im letzten Zeitschritt mit der Testmenge. Auch hier ergibt sich aus den gesammelten Genauigkeiten ein Durchschnittswert.
Neben der Treffergenauigkeit sind folgende weitere Metriken zu ergänzen: Der positive prädiktive Wert (engl. Precision), die Sensivität (engl. Recall) und das F-Maß. Bei der Annahme zweier Klassen A und B berechnet sich der positive prädikative Wert aus den richtig klassifizierten A-Datenpunkten im Verhältnis zu allen Datenpunkten, die mit A gelabelt wurden:

$$\textbf{Positiver prädikt. Wert} = \frac{\textbf{richtig(Label A)}}{\textbf{richtig(Label A) + falsch(Label A)}} \tag{4.2}$$

Die Sensitivität berechnet sich aus den Datenpunkten, die richtig mit Klasse A gelabelt geteilt durch die Summe genau dieser Datenpunkte mit den Datenpunkten, die fälschlicherweise mit B gelabelt wurden und eigentlich zu A gehörten.

$$\textbf{Sensivität} = \frac{\textbf{richtig(Label A)}}{\textbf{richtig(Label A) + falsch(Label B)}} \tag{4.3}$$

Das F-Maß ist das gewichtete harmonische Mittel vom positiven prädikativen Wert (PPW)und der Sensitivität.

$$\textbf{F-Maß} = \frac{\textbf{2 x PPW x Sensitivität}}{\textbf{PPW + Sensitivität}} \tag{4.4}$$

Die Berechnung dieser Metriken erfolgt auch für Datenmengen mit mehreren Klassen, wobei dann PPW, Sensitivität und F-Maß für jede einzelne Klasse berechnet werden.

4.4 Das Vergleichsmodell

Um die Ensemble-Modelle mit einem einfachen Prädikator vergleichen zu können, bedarf es eines Vergleichsmodells. Dieses einzelne, neuronale Netz wird mit Data-Pooling trainiert, da es sich in der Arbeit von Karaoguz als das vorläufig beste, iterative Lernverfahren herausstellte. Das neuronale Netz besteht aus mehreren Schichten: Die erste Konvolutions- oder Faltungsschicht lernt mit 100 Filtern und einer Kernel-Größe von 3. Die Ränder der Eingabe werden mit Nullen aufgefüllt, sodass die resultierenden Feature-Maps in der ersten Dimension die Eingabengröße beibehalten. Die Aktivierungsfunktion ist ReLU. Die zweite Schicht berechnet Global Average Pooling auf die Feature Maps und übergibt das Ergebnis an eine vollständig verbundene Schicht mit 128 Neuronen und ReLU als Aktivierungsfunktion. Diese Schicht ist mit der Ausgabeschicht verbunden, die aus c Neuronen besteht, wobei c abhängt von der Anzahl der Klassen des jeweiligen Datensatzes. Da der IMDB-Datensatz nur zwei verschiedene Klassenlabel enthält, reicht ihr ein einzelnes Neuron in der Ausgabeschicht mit einer Sigmoid-Aktivierungsfunktion. Die Loss-Funktion zur Berechnung des Fehlers ist bïnare Kreuzentropie. Bei dem Yahoo- und Facebook-Datensatz besteht die Ausgabeschicht aus 4 bzw. 3 Neuronen mit Softmax als Aktiverungsfunktion und multi-variante Kreuzentropie zur Berechnung des Fehlers. Die Optimierungfunktion für das Gradientenabstiegsverfahren ist Root-Mean-Square-Propagation (RMSprop).

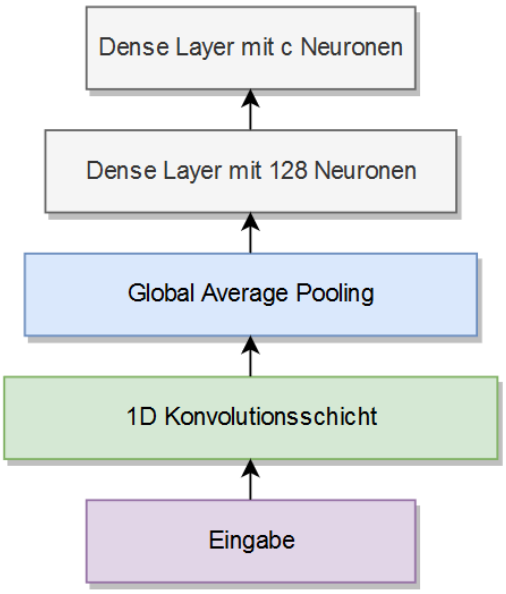

Abbildung 4.1: Vergleichsmodell sowie grundlegende Architektur der CNN-Basisnetze der Ensembles

4.5 Ensembles aus homogenen neuronalen Netzen

Die neuronalen Netze als Basis-Prädikatoren in dieser Experimentierreihe teilen alle die selbe Architektur. Diese entspricht dem vorher beschriebenen Vergleichsmodell in Abschnitt 4.4, dargestellt in Abbildung 4.1. Auch die Hyperparameter wie Lernrate, Größe der Batches und Anzahl der Trainingsepochen sind die gleichen.
Nach den individuellen Trainingsvarianten der Basisnetze, berechnen sie die Vorhersagen für die Hold-out-Mengen und die Testmenge.
Die logistische Regression wird mit den konkatenierten Ausgaben der Basisnetze auf diese Hold-out-Mengen trainiert. Die Werte der Ausgabeneuronen werden konkateniert und der trainierten, logistischen Regression übergeben, die eine finale Vorhersage trifft.

4.5.1 Stacking-Ensemble aus n Basisnetzen mit geteiltem Datenpool

Die Ensemble-Architektur in dieser Experimentierreihe basiert auf Stacking. Als Basis-Klassifikatoren dienen n neuronale Netze und als Meta-Prädikator wird logistische Regression eingesetzt.
Um die Durchführung der Trainingsprozesse zu beschleunigen, wird eine Sequenzlänge von 35 festgelegt. Das bedeutet, dass nur die ersten 35 Wörter jedes Datenpunktes von

dem jeweiligen neuronalen Netz verarbeitet werden. Neben verkürzten Traningszeiten wird auch die Arbeitsspeicherbelastung reduziert, da jeder zusätzliche Sequenzschritt einem weiteren Wortvektor mit 300 Dimensionen pro Datenpunkt entspricht. Sowohl Vergleichsmodell als auch die Basis-Prädikatoren lernen mit der gleichen Sequenzlänge. Die kurze Sequenzlänge wirkt sich insbesondere auf die Metriken bezüglich der IMDB-Daten aus, da diese durchschnittlich aus 238 beziehungsweise 232 Wörtern bestehen. Deshalb reichen die evaluierten Metriken nicht an die Ergebnisse des dazugehörigen Papers heran. In dem späteren Experiment 4.6.2 wird ein optimiertes Ensemble mit größerer Sequenzlänge vorgestellt. Ziel der Experimente avor ist nicht die Optimierung der Evaluationsmetriken. Stattdessen sollen das Vergleichsmodell und das Ensemble aus gleichen Basisnetzen evaluiert werden. Das inkrementelle Lernverfahren mit geteiltem Datenpool ähnelt dem IL mit einem Datenpool insofern, dass die Modelle mit einem Datensatz trainiert werden, der in jedem Zeitschritt vergrößert wird. Dieser Datensatz wird bei diesem Ensemble-Modell auf n Basisnetze aufgeteilt. Kein Basis-Prädikator trainiert mit dem gesamten Datenpool, sondern nur mit einer Teilmenge. Die Aufteilung des Datenpools entspricht dem K-Fold-Prinzip. Sei beispielsweise n=3, dann wird der Pool in drei Teildatensätze separiert und jedes Basisnetz trainiert mit zwei Dritteln des Pools. Dabei wird pro Netz ein anderes Drittel ausgelassen.

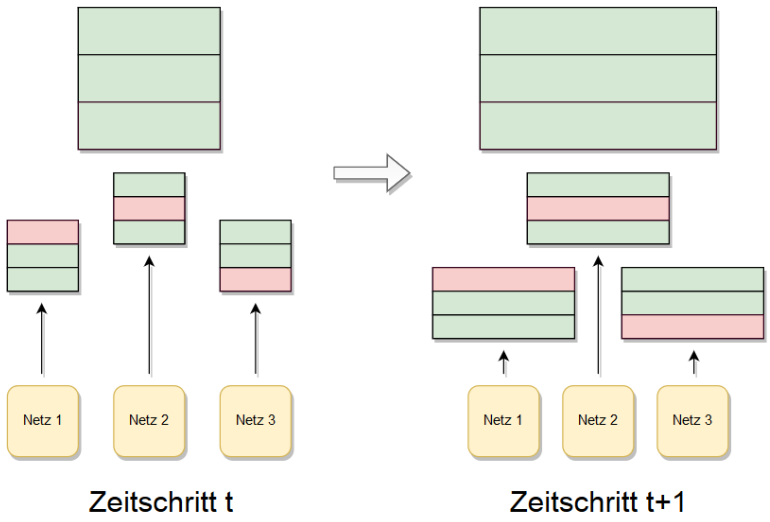

Abbildung 4.2: Ensemble-Lernen mit geteiltem Datenpool und drei Basisnetzen

Die Hold-Out-Menge dient zum Trainieren des Meta-Klassifikators. Dazu trainieren die Basismodelle zuerst nach dem K-Fold-Prinzip und treffen dann Vorhersagen für die Hold-Out-Menge. Die Anwendung und Größe dieser Menge ist jedoch abhängig von dem Datensatz. Bei den IMDB-Daten werden in jeder Epoche 20 Prozent aus dem Datenpool entnommen und bilden die Hold-Out-Menge, die zum Trainieren des Meta-Klassifikators gebraucht wird. Der Yahoo- und Facebook-Datensatz sind nicht balanciert

und enthalten in den einzelnen Zeitschritten teilweise wenig Datenpunkte, sodass ein Auszug von 20 Prozent als Training für den Meta-Klassifikator nicht ausreicht. Die Experimente haben gezeigt, dass die Ensembles gut performen, wenn man dass Prinzip der Hold-Out-Menge nicht anwendet. Das Training der Basisnetze erfolgt für alle drei Datensätze gleich nach dem K-Fold-Prinzip. Bei den Yahoo- und Facebook-Daten wird aber der gesamte Datenpool als Training für den Meta-Klassifikator verwendet.

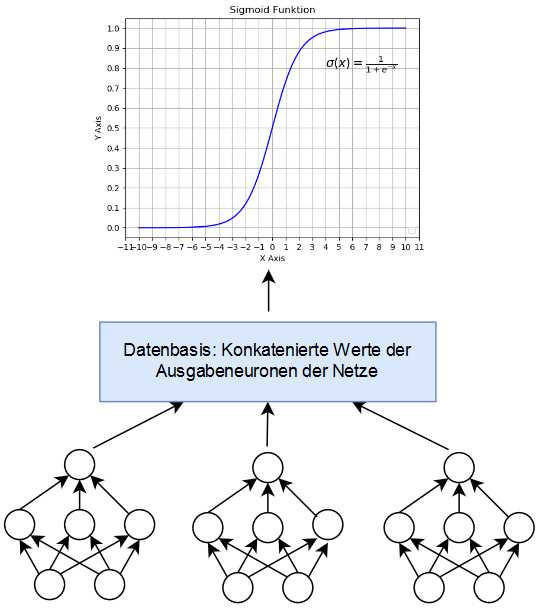

Abbildung 4.3: Aufbau des Stacking-Ensembels mit Datenpool

Im folgenden wird auf die Eigenheiten bezüglich der Experimente zu den einzelnen Datenmengen eingegangen.
Die Ensembles für den Movie-Review-Datensatz werden jeweils mit zwei, drei, vier, fünf und sieben neuronalen Netzen trainiert, die sich den Datenpool nach dem oben beschriebenen K-Fold-Prinzip teilen. Jedes Ensemble-Modell wird fünfmal ausgeführt, um eventuelle Zufallseinflüsse durch initialisierte Kantengewichte zu berücksichtigen. Nach jedem Zeitschritt werden die Metriken berechnet und aufgezeichnet. Um viele Zeitschritte mit möglichst vielen Daten zu simulieren, werden 40.000 Datenpunkte des IMDB-Movie-Review-Datensatzes zum Training verwendet und 10.000 Datenpunkte als Testmenge für die Metriken. Die Trainingsdaten werden gemischt und zur Simulation von 20 Zeitschritten verwendet. Zur Simulation des wachenden Datenpools wird ein Index genutzt, der sich in jeder Iteration um 2000 Punkte verschiebt. Das heißt in der ersten Iteration wird mit 2000 Daten trainiert, in der nächsten dann mit 4000 Daten und so weiter.
Der Yahoo-Datensatz ist auf 34 Tage verteilt, die am 01.04.2016 beginnen und am 04.05.2016 enden. Die Ensembles werden hierbei mit jeweils zwei, drei, vier und fünf

neuronalen Netzen trainiert, da eine weitere Erhöhung der Basismodelle in den Experimenten keine Verbesserung bringt. Als Testmenge dient in jedem Zeitschritt der Datensatz des zukünftigen Zeitschrittes. Im Gegensatz zu den Ensembles der IMDB-Movie-Review-Daten, die immer auf die gleiche Testmenge evaluiert werden, erhalten die Yahoo-Ensembles in jedem Zeitschritt eine andere, unbekannte Datenmenge. Die Entscheidung, die Daten des jeweiligen nächsten Tages als Testmenge zur Performanzauswertung zu verwenden entspricht am ehesten einem realistischen Szenario, bei der ein Modell für die Bewertung von Datenströmen in diskreten Zeitschritten eingesetzt wird.
Die verwendeten Facebook-Daten befinden sich in einem Zeitraum vom 28.04.2017 bis zum 18.06.2017 und umfassen 50 Tage. Genau wie bei den Yahoo-Daten ist die Testmenge jeweils der Datensatz des Folgetages im momentanen Zeitschritt. Wie bei den Facebook-Daten, bestehen auch diese Ensembles aus jeweils zwei, drei, vier und fünf neuronalen Netzen, da eine weitere Erhöhung der Netzanzahl keine Verbesserung der Metriken bringt.

4.5.2 Stacking-Ensemble aus n Basisnetzen ohne Datenpool

Dieses Verfahren ähnelt sehr dem einfachen Training eines neuronalen Netzes ohne Daten-Pooling. Statt eines Datenpools ist zu jedem Zeitpunkt nur ein Teildatensatz vorhanden, der sich von den vorherigen Datensätzen unterscheidet. Bei diesem Ensemble-Modell trainiert ein Basisnetz mit dem aktuellen Teildatensatz, während die anderen Basisnetze mit den Daten vorheriger Zeitschritte trainieren.
Je mehr Netze hinzugefügt werden, desto mehr Zeitschritte werden aus der Vergangenheit von einem zugehörigen Teilnetz als Trainingsdaten verwendet. Das jeweilige Netz trainiert aber ausschließlich mit seinem zugewiesenen Zeitschritt und lernt währenddessen nicht mit den Daten der anderen Netze. Die Basisnetze werden deshalb auch nicht mit den K-Fold-Prinzip trainiert. Stattdessen lernen die Netze jeweils mit dem gesamten Teildatensatz. Im Anschluss daran, werden für die aktuellen Daten des Yahoo- und Facebook-Datensatzes zum Zeitpunkt t von allen Ensembles Vorhersagen gemacht, die dem Meta-Klassifikator als Trainingsdaten übergeben werden. Für den IMDB-Datensatz wird ein hybrides Trainigsverfahren ausgewählt, bei dem die Datensätze mit jedem Zeitschritt zu einem Datenpool hinzugefügt wird, der nur dem Meta-Klassifikator zur Verfügung steht. Dadurch trainiert der Meta-Klassifikator mit Data-Pooling, während die Basisnetze ohne Datenpool trainieren.

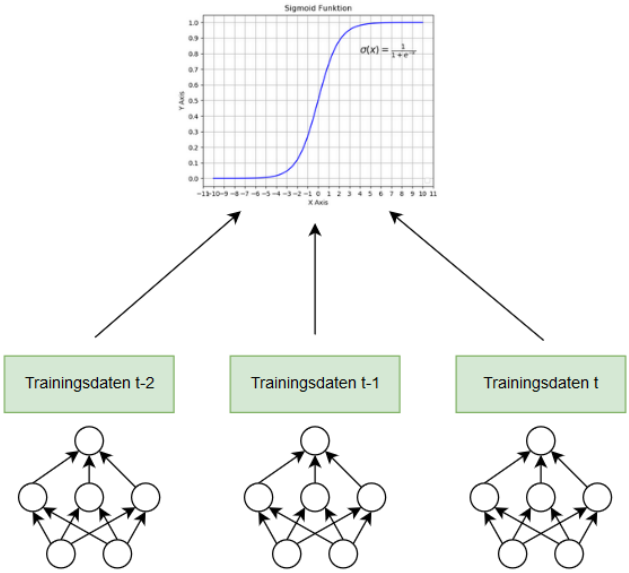

Abbildung 4.4: Ensemble ohne Data-Pooling

Bezüglich des IMDB-Datensatzes werden die gleichen 40.000 Datenpunkte als Training und die restlichen 10.000 zum Testen verwendet, wie in dem Experiment davor. Die 40.000 Datenpunkte werden in 20 distinkte Teildatensätze separiert und simulieren damit unterschiedliche Daten zu den unterschiedlichen Zeitschritten. Aus dem Yahoo-Datensatz werden Datenpunkte vom 01.04.2016 bis zum 05.05.2016 genommen. Anhand des Zeitstempels werden die Daten in Teildatensätze separiert. Dadurch repräsentiert jeder Teildatensatz die Kommentare für einen Tag. Das gleiche Verfahren wird für die Facebook-Daten für den Zeitraum 28.04.2017 bis zum 18.06.2017 angewendet. Die Evaluation der IMDB-Datensatz-Ensembles erfolgt durch die separate Testmenge während die anderen beiden Datensätze die Metriken jeweils durch die Vorhersagen für die Kommentare berechnen, die einen Zeitschritt in der Zunkunft liegen.

4.6 Ensembles aus heterogenen neuronalen Netzen

4.6.1 Stacking-Ensemble aus n Basisnetzen mit geteiltem Datenpool

Dieses Experiment ist dem Experiment 4.5.1 sehr ähnlich und unterscheidet sich hauptsächlich in der Auswahl der Basisklassifikatoren. Hyperparameter für das CNN, Sequenzlänge und Epochenanzahl werden aus dem Experiment 4.5.1 übernommen. Statt homogenen Basisnetzen, die nur aus CNNs bestehen, sind die grundlegenden Prädika-

toren dieses Stacking-Ensembles divers. Neben CNNs bestehen diese aus LSTMs und GRU-Zellen. Logistische Regression wird wieder als Meta-Klassifikator eingesetzt und trainiert mit den Prädiktionen der Basisnetze.

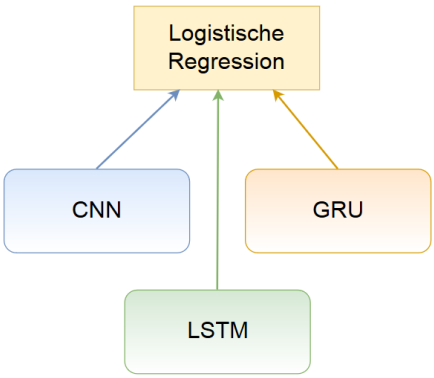

Abbildung 4.5: Ensemble aus drei heterogenen Basisnetzen

Genau wie bei dem Ensemble aus homogenen Basisnetzen mit Daten-Pooling, gibt es bei den Datensätzen einen Unterschied hinsichtlich der Trainingsmenge des Meta-Klassifikators. Der Meta-Klassifikator des IMDB-Datensatzes trainiert in jedem Zeitschritt nur mit einer Hold-Out-Menge, die den Prädiktionen der Basisnetze für 20 Prozent des Datenpools entspricht. Die Meta-Klassifikatoren des Yahoo- und Facebook-Datensatzes trainieren mit den Vorhersagen ihrer Basisnetze für den gesamten, verfügbaren Datenpool. Dieser Unterschied ist darin begründet, dass der Datenpool des IMDB-Datensatzes größer ist, weil zum einen mehr Daten vorhanden sind und zum anderen der Pool gleichmäßiger wächst.
Das Training der Basisnetze erfolgt nach dem K-Fold-Prinzip, das in Experiment 4.5.1 erläutert und in Abbildung 4.2 skizziert wurde.
Die Ensembles für den IMDB-Datensatz bestehen aus zwei bis sieben Netzen, während die Ensembles der anderen Datenmenge aus zwei bis fünf Netzen bestehen. Die Bestimmung der Netztypen erfolgt dabei nach einem festgelegten Schema. Ein Ensemble aus zwei Basisnetzen besteht aus einem CNN und einem LSTM. Bei drei Netzen wird eine GRU-Zelle hinzugefügt. Nach dem selben Muster werden dann weitere Netze ergänzt. Dadurch ist das vierte Basisnetz wieder ein CNN und das fünfte ein LSTM. Die Reihenfolge setzt sich so weiter fort.
Der Hidden-State der LSTM- und GRU-Zellen besteht aus jeweils 128 Neuronen. Die Aufteilung der Datensätze sei hier nur kurz erwähnt, denn sie ist identisch mit der Aufteilung aus dem Experiment 4.5.1. Auch die Auswahl der jeweiligen Testmengen ist gleich. Der IMDB-Datensatz wird mit einer fest definierten Trainingsmenge evaluiert, während die Metriken des Yahoo- und Facebook-Datensatzes mit den Vorhersagen

für den nächsten Zeitschritt berechnet werden. Dazu ist der IMDB-Datensatz wieder in 40.000 Datenpunkte für Training und Testen unterteilt. Die 40.000 Datenpunkte werden in 20 Teildatensätze separiert aus denen dann ein wachsender Datenpool generiert wird. Dazu entspricht der erste Teildatensatz dem Datenpool in Zeitschritt 1 und im nächsten Zeitschritt wird Teildatensatz 2 konkateniert. Die Datenmenge wächst dadurch in jedem Zeitschritt stetig um 2000 Datenpunkte.
Der Yahoo- und Facebook-Datensatz haben konkrete Zeitstempel, die genügen um die Datenmenge in tagesabhängige Teildatensätze zu separieren. In jedem Zeitschritt wird der aktuelle Tagesdatensatz zum Datenpool hinzugefügt. Für die Yahoo-Daten wird der Zeitraum vom 1.04.2016 bis zum 5.05.2016 genutzt, wodurch 34 Zeitschritte zur Verfügung stehen. Bei den Facebook-Daten ist der Zeitraum vom 28.04.2017 bis zum 18.06.2017 für das inkrementelle Lernen von Bedeutung. Dadurch wird hierbei mit 50 Zeitschritten trainiert.

4.6.2 Adaptiertes Stacking-Ensemble aus 3 Basisnetzen mit geteiltem Datenpool

Die bisherigen Experimente decken die verschiedenen Lernverfahren von Ensembles aus neuronalen Netzen in Bezug zu einem Vergleichsmodell ab, das ebenfalls mit inkrementellem Lernen trainiert wird.
Um die Qualität dieses Ansatzes aus Ensembles mit inkrementellem Lernen gegenüber einem klassischen Lernverfahren bewerten zu können, werden in diesem Experiment beide Trainingsmethoden gegenübergestellt. Das klassische Lernverfahren lässt sich wie folgt beschreiben: Ein Modell trainiert einmalig mit einer Trainingsmenge und macht anschließend Vorhersagen für eine Testmenge.
Zur dieser speziellen Gegenüberstellung beider Verfahren bedarf es einer Adaption der bisherigen Parameter, da diese für einen Vergleich verschiedener inkrementeller Lernverfahren optimiert sind.
Als Datenmenge wird der IMDB-Review-Datensatz gewählt, da in der zugehörigen wissenschaftlichen Arbeit bereits die Ergebniswerte der Evaluationsmetriken enthalten sind.
Das gewählte Lernverfahren, das mit dem klassischen Ansatz verglichen wird, ist inkrementelles Ensemble-Lernen mit einem Datenpool, da die Experimente ohne Datenpool im Vergleich dazu schlechtere Performanz aufweisen. Es wird ein Ensemble aus 3 heterogenen Basisnetzen gewählt, da pro Zeitschritt mit mehreren Epochen trainiert wird und daher auch die rekurrenten neuronalen Netze, die vergleichsweise langsamer lernen, ihre spezifischen Eigenschaften einbringen können.
Die Basisnetze bestehen aus einem CNN mit derselben Architektur, die auch in den vorherigen Experimenten verwendet wurde, sowie einem LSTM und einer GRU-Zelle. Die Architektur entspricht im wesentlichen dem Modell in Abbildung 4.5.

Der größte Einflussfaktor auf die Qualität der Evaluationsmetriken hat in diesem Experiment die Sequenzlänge. Während in vorherigen Experimenten die Sequenzlänge von 35 Einheiten bewusst gewählt wurde, damit die Trainingsprozesse beschleunigt ablaufen und die inkrementellen Lernverfahren schneller miteinander verglichen werden können, ist in diesem Experiment die Treffergenauigkeit der wichtigste Faktor. Die Sequenzlänge entspricht der Anzahl der Wörter, die die neuronalen Netze als Eingabe erhalten. Die meisten Filmkritiken sind deutlich länger als 35 Wörter (durschnittlich 238 bzw. 232), weshalb eine so kurze Sequenzlänge womöglich zu wenig Information enthält, um gute Klassifikationsergebnisse zu erreichen. Eine Sequenzlänge von 210 Wörtern könnte ausreichen, um die vorgegebenen Metriken aus der Arbeit zu übertreffen.
Die Trainingsdaten werden auf 10 Zeitschritte verteilt und nach jedem dieser Zeitschritte wird das trainierte Ensemble mit den Testdaten evaluiert. Trainings- und Testdaten sind identisch mit den aufgeteilten Datenpunkten aus der zugehörigen wissenschaftlichen Arbeit.
Die Anzahl der Trainingsepochen beträgt fÃ$\frac{1}{4}$nf und nicht eine Epoche wie in dem vorherigen Experimenten, da der Datenpool sowie die Sequenzlänge von Anfang an größer sind.
Da das Ensemble nur in zehn Zeitschritten lernt, wird zum Trainieren der logistischen Regression als Meta-Klassifikators keine Hold-Out-Menge verwendet. Stattdessen machen die Basisnetze ihre Vorsagen für den gesamten Datensatz und diese Prädiktionen werden dann der logistischen Regression zum Training übergeben.

5 Ergebnisse

5.1 Ensembles aus homogenen neuronalen Netzen

5.1.1 Stacking-Ensemble aus n Basisnetzen mit geteiltem Datenpool

Diese Evaluation beruht auf dem in Abschnitt 4.5.1 vorgestellten Experiment.
Da für die Evaluation dieses Experiments drei divergente Datensätze genutzt wurden, werden diese nacheinander im Bezug zum Vergleichsmodell betrachtet.
Zuerst erfolgt die Auswertung beider Ansätze auf dem IMDB-Datensatz. Hierbei werden die durchschnittlichen Werte der Metriken, bezogen auf alle 20 Zeitschritte sowie auf die letzten 10 Zeitschritte, präsentiert. Dies ist während des Experiments als erforderlich festgestellt worden, da sich die Werte des Vergleichsmodells in späteren Zeitschritten in eine stabilere Richtung entwickeln.

Tabelle 5.1: IMDB-Datensatz: Metriken mit unveränderter Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Vergleichsmodell	0.749	0.745	**0.779**	0.748
Ensemble mit 2 Netzen	0.755	0.749	0.767	0.757
Ensemble mit 3 Netzen	0.764	0.761	0.771	0.766
Ensemble mit 4 Netzen	0.767	0.762	0.776	0.769
Ensemble mit 5 Netzen	0.767	0.762	0.776	0.769
Ensemble mit 7 Netzen	**0.770**	**0.766**	**0.779**	**0.772**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle, bezogen auf alle 20 Zeitschritte, dar. Pro Modell wurden jeweils fünf Durchläufe erfasst, wobei nach jedem Zeitschritt die Metriken aufgezeichnet wurden. Jeder dieser Durchläufe ergibt einen Durchschnittswert, der wiederum gemittelt wird.

Die aufgezeichneten Metriken aus der Tabelle 5.1 bezüglich der IMDB-Datenbasis zeigen deutlich auf, dass das Ensemble-Modell mit geteiltem Datenpool, hinsichtlich einer unbekannten Testmenge, durchgehend bessere Performanz aufweist, sofern man den gesamten Zeitverlauf einbezieht. Schon ab einer Menge von zwei Netzen als Basis-Prädikatoren

übertrifft das Ensemble das Vergleichsmodell in allen Metriken außer der Sensitivität. Weiterhin zeigt sich, dass die Ensemble-Modelle mit zunehmender Anzahl der Basisnetze ihre Metriken weiter verbessern. Das Ensemble aus sieben Netzen weist die beste Performanz auf und erzielt die gleiche Durchschnittssensitivität wie das Vergleichsmodell.

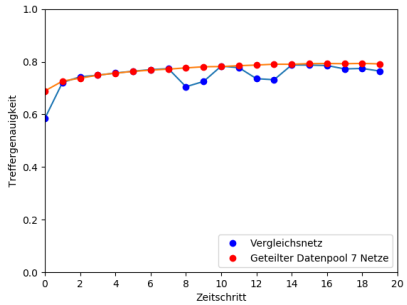

Abbildung 5.1: Treffergenauigkeit

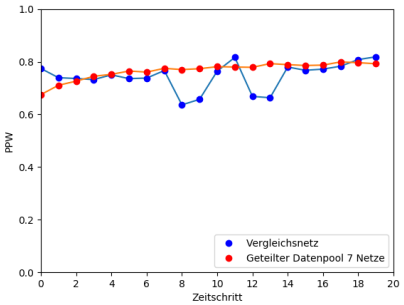

Abbildung 5.2: PPW

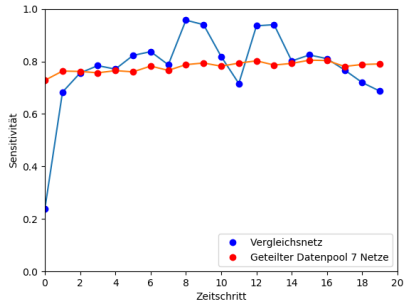

Abbildung 5.3: Sensitivität

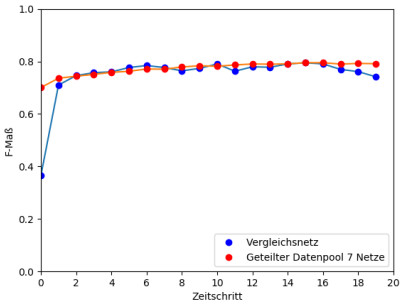

Abbildung 5.4: F-Maß

Neben den Durschnittswerten ist eine Beurteilung der Metriken der zu vergleichenden Verfahren auch hinsichtlich des zeitlichen Verlaufs notwendig.
Die Betrachtung der Werte des Vergleichsmodells (Abbildung 5.1 bis 5.4) zeigt einen volatilen Verlauf auf, der sich jedoch in den späteren Zeitschritten etwas stabilisiert. Zwar schwanken die Werte noch deutlich stärker als die des Ensembles, doch diese Varianz nimmt in ihrer Intensität mit zunehmender Zeit ab. Besonders auffällig sind die starken Schwankungn des PPW der Sensitivität, die sich auch auf das F-Maß auswirkt. Auch noch in den letzten Zeitschritten zeigt sich an deren Werteverlauf, dass das Vergleichsmodell deutlich instabiler ist. Bei der Betrachtung des PPW-Verlaufs und der Sensitivität des Vergleichsmodells, zeigt sich, dass die Werte jeweils in die Gegenrichtung ausschlagen. Ein hoher PPW-Wert ist immer mit einem nidrigen Sensitivitätswert verbunden und umgekehrt. Die Vorhersagen des Netzes sind abwechselnd für eine Klasse gut und für die andere fehlerhaft. Das Netz scheint gelernte Merkmale einer Klasse aus einem vorherigen Zeitschritt im nächsten Zeitschritt wieder zu überschreiben. Die starke Varianz bezüglich dieser Metriken in der ersten Hälfte des Zeitverlaufs ausgeprägter.
Bei der genaueren Betrachtung der Ergebnisse des Vergleichsmodells, angewendet auf

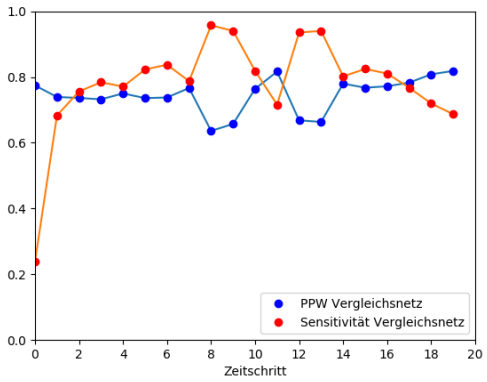

Abbildung 5.5: Verlauf der Metriken des Vergleichsmodells

die Movie-Review-Daten, bezogen auf den zeitlichen Verlauf, fällt auch auf, dass auch hier die Anomalien auftreten, die Karaoguz in seiner Arbeit in Bezug auf sein Datenpool-Models auf den MSR-Datensatz beobachtet hat [Karaoguz 2018]. Damit kann Karaoguz' Vermutung, dass die Instabilität der Ergebnisse an der Unvereinbarkeit seines Modells mit dem MSR-Datensatz lag, hinterfragt werden. Stattdessen ist zu vermuten, dass einzelne neuronale Netze mit einem Datenpool dazu neigen können, in ihren Evaluationsmetriken Instabilität zu erzeugen. Dies ist womöglich auf die Überanpassung zurückzuführen. In späteren (auch früheren) Zeitschritten schwanken der Sensitivitäts- und PPW-Wert.

Die Abnahme der Varianz im späteren Verlauf des Vergleichsmodells, erfordert eine Bewertung der späteren Zeitschritte beider Ansätze. Auch hier werden zunächst die Durchschnittsmetriken dargestellt.

Tabelle 5.2: IMDB-Datensatz: Metriken bezogen auf die letzten 10 Zeitschritte mit unveränderter Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Vergleichsmodell	0.769	0.664	**0.802**	0.775
Ensemble mit 2 Netzen	0.774	0.766	0.791	0.778
Ensemble mit 3 Netzen	0.784	0.786	0.780	0.783
Ensemble mit 4 Netzen	0.786	0.783	0.793	0.787
Ensemble mit 5 Netzen	0.786	0.782	0.792	0.787
Ensemble mit 7 Netzen	**0.789**	**0.788**	0.792	**0.790**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle, bezogen auf die letzten zehn Zeitschritte, dar. Pro Modell wurden jeweils fünf Durchläufe erfasst, wobei nach jedem Zeitschritt die Metriken aufgezeichnet wurden. Jeder dieser Durchläufe ergibt einen Durchschnittswert, der wiederum gemittelt wird.

Dabei zeigt sich, dass die Ensemble-Modelle bezüglich der Treffergenauigkeit, des positiven prädiktiven Wertes und des F-Maßes weiterhin besser abschneiden, als das einfache CNN, jedoch wird die Sensitivität übertroffen. Auffällig ist auch bei dieser Betrachtung die Stabilität des Ensemble-Modells im Vergleich zum einfachen Netz. Die Werte der Metriken eines spezifischen Ensembles weichen kaum voneinander ab. Diese stabile Vorhersagen verfestigen sich mit zunehmender Anzahl an Basisnetzen.

Die bisherigen Ergebnisse betreffen einen balancierten Datensatz, ohne eine wirkliche zeitliche Dimension. Die Zeitachse ist nur simuliert und erlaubt daher keine logische oder sinnhafte Verknüpfung der Teildatensätze untereinander. Die Datensätze zum Zeitschritt *t* und Zeitschritt *t+1* sind zufällig generiert und können keine zeitliche Entwicklung hinsichtlich ihres Inhalts widerspiegeln. Daher wird nun die Evaluation der Datensätze vorgenommen, die eine tatsächliche zeitliche Dimension durch Zeitstempel inne haben und eine Beurteilung der inkrementellen Lernverfahren zulassen, die realistischer ist.
Der Yahoo-Datensatz besteht aus 34 Zeitschritten, die tatsächlichen Tagen entsprechen und daher eine genauere Beurteilung der Ensemble-Modelle auf inkrementelles Lernen möglich machen. Statt einer einzelnen separaten Testmenge wird jeweils der zukünftige, noch unbekannte Zeitschritt zum Berechnen der Metriken genutzt.
In den folgenden Tabellen 5.3 und 5.4 werden für das Vergleichsmodell sowie die Ensembles die durchschnittlichen Metriken dargestellt. Für jede der vier Klassen sind PPW, Sensitivität (abgekürzt Sen) und das F-Maß angegeben. Die Klassen sind abgekürzt mit K1 bis K4 und die Treffergenauigkeit wurde mit TG bezeichnet.

Tabelle 5.3: Yahoo-Datensatz bezogen auf 34 Zeitschritte: TG, K1 und K2

	TG	PPW K1	Sen K1	F-Maß K1	PPW K2	Sen K2	F-Maß K2
Vergleichsmodell	0.519	0.589	**0.814**	0.670	0.197	0.087	0.095
Ensemble 2 Netze	**0.536**	0.604	0.795	**0.683**	**0.266**	0.087	0.127
Ensemble 3 Netze	0.530	0.605	0.775	0.676	0.258	0.096	0.136
Ensemble 4 Netze	0.528	0.607	0.770	0.676	0.257	0.102	0.143
Ensemble 5 Netze	0.531	**0.608**	0.769	0.676	0.260	**0.111**	**0.152**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle in Bezug auf 34 Zeitschritte für die Klassen 1 und 2 dar.

Tabelle 5.4: Yahoo-Datensatz bezogen auf 34 Zeitschritte: K3 und K4

	PPW K3	Sen K3	F-Maß K3	PPW K4	Sen K4	F-Maß K4
Vergleichsmodell	0.364	0.308	0.303	0.275	0.079	0.110
Ensemble 2 Netze	**0.414**	0.398	0.392	0.299	0.096	0.135
Ensemble 3 Netze	0.397	0.405	0.395	**0.315**	0.118	0.161
Ensemble 4 Netze	0.393	0.404	0.392	0.302	0.118	0.160
Ensemble 5 Netze	0.403	**0.408**	**0.399**	0.306	**0.126**	**0.169**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 34 Zeitschritte für die Klassen 3 und 4 dar.

Bei der Betrachtung der Durchschnittswerte des Vergleichsmodells sowie der Werte der Ensembles fällt auf, dass, unabhängig von der Anzahl der Teilnetze, die Metriken der Ensembles bessere Performanz aufweisen. Mit Ausnahme der Sensitivität von Klasse 1, die der Sensitivität der negativen Kommentare entspricht, liegen sämtliche Werte der Ensembles über denen des Vergleichsmodells. Die besten Metriken erzielt das Ensemble aus zwei bzw. fünf Teilnetzen. Das Ensemble aus zwei Teilnetzen kann negative Kommentare am besten vorhersagen (F-Maß), während das Ensemble aus fünf Netzen die anderen drei Klassen (gemischte, neutrale und positive Kommentare) deutlich besser prognostizieren kann.

Die Werteverläufe der F-Maße aller vier Klassen bestätigen die Beobachtung, dass die Ensembles bessere Performanz erzielen, als das Vergleichsmodell (in diesem Fall das Ensemble aus fünf Basisnetzen). Die negativen Kommentare werden von beiden Architekturen zuverlässig prognostiziert. Deutliche Unterschiede zeigen sich in den F-Maßen der anderen Klasse. Sowie bei den gemischten, neutralen als auch den positiven Kommentaren lernt das Ensemble-Modell früher eine Klassenunterscheidung. Positive Kommentare werden von dem Ensemble schon ab dem fünften Zeitschritt gelernt, während das Vergleichsmodell erst ab dem neunten Tag ein F-Maß berechnet, dass über Null liegt. Bei den Verläufen der F-Maß-Werte für die Klasse der gemischten Kommentare (Klasse 2),

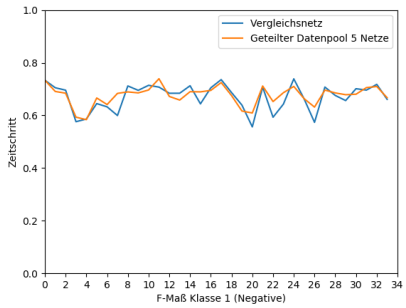

Abbildung 5.6: Yahoo-Datensatz F-Maß Klasse 1

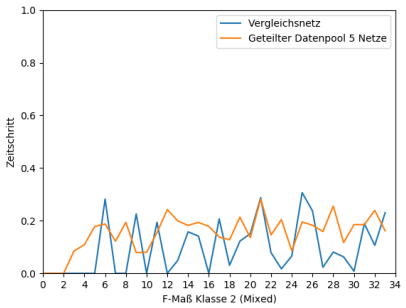

Abbildung 5.7: Yahoo-Datensatz F-Maß Klasse 2

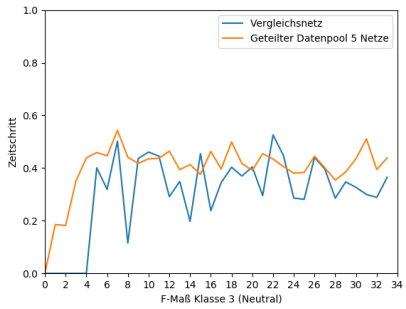

Abbildung 5.8: Yahoo-Datensatz F-Maß Klasse 3

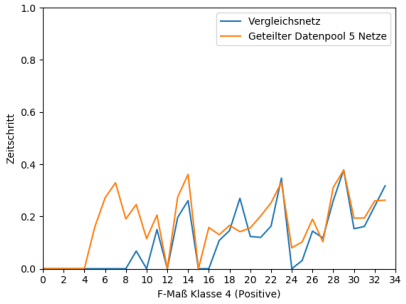

Abbildung 5.9: Yahoo-Datensatz F-Maß Klasse 4

lernt das Ensemble schon ab der dritten Epoche eine Klassenunterscheidung. Das Vergleichsnetz prognostiziert erst ab der sechsten Epoche einen Wert, der über Null liegt. Die Klasse der gemischten Kommentare scheint das einzelne neuronale Netz nur sehr schwer zu lernen. In den Zeitschritten 7, 8, 10, 12, 16 und 30 liegt das F-Maß bei Null wobei die Werte des Ensembles ab der dritten Epoche immer über Null liegen.

Die Experimente mit dem Facebook-Datensatz beziehen sich auf 50 Zeitschritte. Die Prognose umfasst drei Klassen, für die in den folgenden Tabellen Metriken gemittelt werden.

Tabelle 5.5: Facebook-Datensatz bezogen auf 50 Zeitschritte: TG und K1

	TG	PPW K1	Sen K1	F-Maß K1
Vergleichsmodell	0.607	0.454	0.251	0.285
Ensemble mit 2 Netzen	0.620	0.481	0.477	0.458
Ensemble mit 3 Netzen	0.619	0.462	0.468	0.452
Ensemble mit 4 Netzen	**0.635**	**0.500**	**0.499**	**0.475**
Ensemble mit 5 Netzen	0.624	0.494	0.488	0.470

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 50 Zeitschritte für die Klasse 1 dar.

Tabelle 5.6: Facebook-Datensatz bezogen auf 50 Zeitschritte: K2 und K3

	PPW K2	Sen K2	F-Maß K2	PPW K3	Sen K3	F-Maß K3
Vergleichsmodell	0.432	0.372	0.360	0.739	0.678	0.698
Ensemble mit 2 Netzen	0.454	0.450	0.442	0.748	0.769	**0.748**
Ensemble mit 3 Netzen	0.469	0.459	0.447	**0.751**	0.766	0.746
Ensemble mit 4 Netzen	**0.471**	**0.466**	**0.457**	0.742	**0.773**	0.744
Ensemble mit 5 Netzen	0.441	0.462	0.441	0.735	0.743	0.731

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 50 Zeitschritte für die Klassen 2 und 3 dar.

Die Durchschnittsmetriken bezogen auf den Facebook-Datensatz in Tabellee 5.5 bis 5.6 bestätigen die Observation, welche auch schon hinsichtlich des Yahoo-Daten gemacht wurde: Die Ensemble-Modelle weisen nahezu überall bessere Werte auf. Eine deutliche Differenz weisen Sensitivität und F-Maß von Vergleichsnetz und Ensemble-Modellen bezüglich der Klasse 1 auf, die *CSA* (*counter speech act*) repräsentiert. Hier weicht die Sensitivität des Vergleichsmodells mindestens 0.217 Punkte von den Werten des Ensembles ab. Die Ensembles erkennen somit über 20 Prozent mehr CSA-Kommentare aus der Menge aller CSA-Kommentare. Die Differenz des F-Maßes entspricht mindestens 0.167 Punkte, berechnet mit dem Wert aus dem Ensemble bestehend aus drei Basisnetzen.

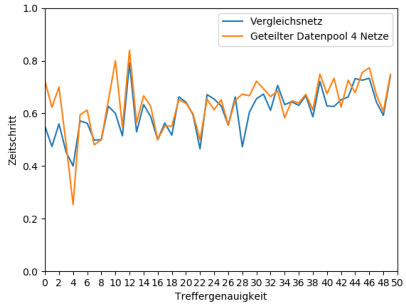

Abbildung 5.10: Facebook-Datensatz Treffergenauigkeit

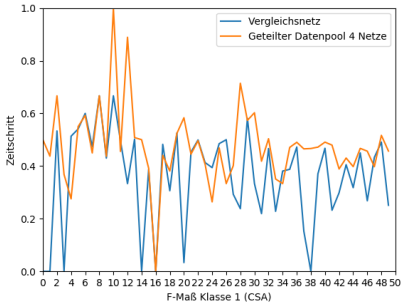

Abbildung 5.11: Facebook-Datensatz F-Maß Klasse 1

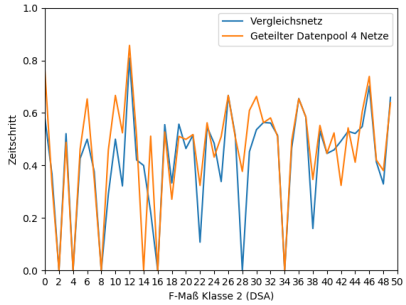

Abbildung 5.12: Facebook-Datensatz F-Maß Klasse 2

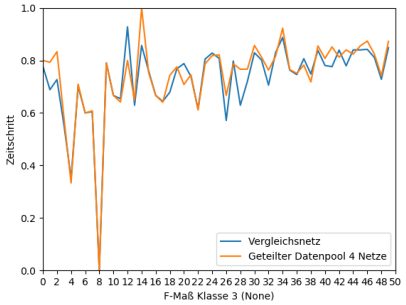

Abbildung 5.13: Facebook-Datensatz F-Maß Klasse 3

Die Werteverläufe der Metriken bezüglich des Facebook-Datensatzes unterstützen die Beobachtung, die bereits zuvor bei der Betrachtung der Tabellen 5.5 und 5.6 gemacht wurde. Das Ensemble-Modell erzielt im Verlauf, in der Mehrheit der Zeitschritte, höhere Werte als das Vergleichsmodell. Insbesondere der Verlauf in Klasse 1 (*CSA*) zeigt auf, dass das Ensemble hier deutlich bessere Klassenvorhersagen macht. Die Treffergenauigkeit sowie das F-Maß der Klasse 3 (*None*) beider Architekturen zeigen einen deutlichen positiven Trend auf.

Abschließend lassen sich die vorliegenden Ergebnisse für alle drei Datensätze insofern interpretieren, dass Ensemble-Modelle mit einem geteilten Datenpool im Bereich des inkrementellen Lernens bessere Vorhersagen machen können als ein einzelnes neuronales Netz. Auch bei zeitabhängigen, unbalancierten Datensätzen übertrifft die Ensemble-Architektur das einzelne Vergleichsmodell. Die Ensembles sind auch in der Lage die Klassen der unbalancierten Datenmengen besser zu lernen, die unterrepräsentiert sind.

Die zeitlichen Ressourcen werden hier nur von einem theoretischen Standpunkt aus betrachtet und beruhen nicht auf den Ergebnissen der Experimente. Beim geteilten Datenpool trainiert jedes Basisnetz des Ensembles mit einem Teildatensatz. Bei n Netzen erhält

jedes Teilnetz *(n-1)/n* Anteile des gesamten Datenpools. Während bei zwei Basisnetzen jeder Basisklassifikator noch mit der Hälfte der Daten trainiert, steigt mit zunehmender Netzanzahl die anteilige Trainingsmenge. Bei der sequentiellen Ausführung steigt die Zeit, die für das Training benötigt wird, da jedes weitere Basisnetz pro Zeitschritt für eine Epoche trainiert wird. In diesem Fall bringt das Ensemble mit geteiltem Datenpool keinen zeitlichen Vorteil. Das Training der Basisnetze bietet jedoch die Möglichkeit parallel ausgeführt zu werden, da diese Vorgänge voneinander unabhängig sind. Das Trainieren der logistischen Regression als Meta-Klassifikator, bedarf kaum zeitlicher Ressourcen. Dennoch bedarf die Aufteilung des Datenpools und die Organisation der neuronalen Netze in Threads beziehungsweise Prozessen Zeit, wodurch zu vermuten ist, dass paralleles Trainieren des Ensembles ungefähr gleich schnell durchzuführen ist, wie das Training des Vergleichsmodells. Dies müsste aber in anderen Experimenten außerhalb dieser Arbeit untersucht werden.

5.1.2 Stacking-Ensemble aus n Basisnetzen ohne Datenpool

Ähnlich wie bei der Auswertung der vorherigen Experimente werden nacheinander die Metriken in Bezug auf die unterschiedlichen Datensätze präsentiert und evaluiert. Diese Evaluation bezieht sich auf das vorgestellte Experiment in Abschnitt 4.5.2.
Die Performanz von bis zu fünf Basisnetzen des Ensembles wird hinsichtlich der IMDB-Daten dem Vergleichsmodell gegenüber gestellt.

Tabelle 5.7: IMDB-Datensatz: Metriken mit unveränderter Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Vergleichsmodell	0.749	0.745	**0.779**	0.748
Ensemble mit 2 Netzen	0.764	0.758	0.773	0.766
Ensemble mit 3 Netzen	**0.766**	**0.765**	0.766	0.766
Ensemble mit 4 Netzen	0.**766**	**0.765**	0.769	**0.767**
Ensemble mit 5 Netzen	0.763	0.758	0.773	0.765

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf alle 20 Zeitschritte dar. Pro Modell wurden jeweils fünf Durchläufe erfasst, wobei nach jedem Zeitschritt die Metriken aufgezeichnet wurden. Jeder dieser Durchläufe ergibt einen Durchschnittswert, der wiederum gemittelt wird.

Der Tabelle 5.7 kann entnommen werden, dass die Ensemble-Modelle durchschnittlich bessere Performanz, als das einzelne Vergleichsmodell aufweisen. In allen Metriken, außer der Sensitivität, erreichen die Ensemble-Modelle höhere Werte.

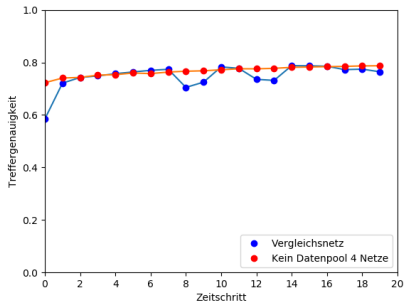

Abbildung 5.14: Treffergenauigkeit

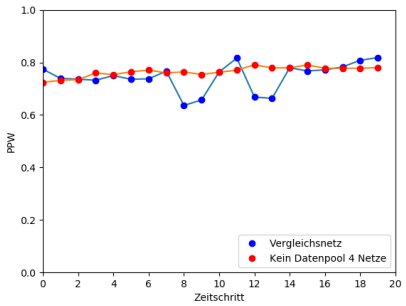

Abbildung 5.15: PPW

Die Wertverläufe in Abbildung 5.14 und 5.15 zeigen einen stabilen Verlauf des Ensemble-Modells mit vier Basisnetzen auf und bestätigen eine bessere Performanz gegenüber dem Vergleichsmodell. Die Werte der Treffergenauigkeit und des F-Maßes verlaufen ähnlich. Der PPW-Wert wird von dem Ensemble überboten, während das einzelne neuronale Netz öfter hohe Werte bei der Sensitivität erreicht. Hinsichtlich der vorliegenden Ergebnisse zu dem IMDB-Datensatz muss noch erwähnt werden, dass nur die Basisnetze ohne Datenpool trainiert wurden. Der Meta-Klassifikator verwendet wachsenden Datenpool zum

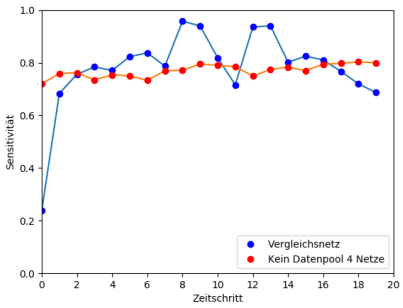

Abbildung 5.16: Sensitivität

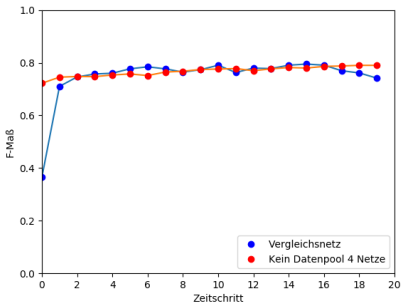

Abbildung 5.17: F-Maß

Training. Diese Eigenheit besteht bei den Modellen für die folgenden beiden Datensätze nicht.

Tabelle 5.8: IMDB-Datensatz: Metriken bezogen auf die letzten 10 Zeitschritte mit unveränderter Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Vergleichsmodell	0.769	0.664	**0.802**	0.775
Ensemble mit 2 Netzen	0.777	0.769	0.791	0.780
Ensemble mit 3 Netzen	0.779	0.774	0.789	**0.781**
Ensemble mit 4 Netzen	**0.780**	**0.778**	0.784	**0.781**
Ensemble mit 5 Netzen	0.777	0.769	0.790	0.779

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf alle 20 Zeitschritte dar. Pro Modell wurden jeweils fünf Durchläufe erfasst wobei nach jedem Zeitschritt die Metriken aufgezeichnet wurden. Jeder dieser Durchläufe ergibt einen Durchschnittswert, der wiederum gemittelt wird.

Die überwiegend besseren Metriken der Ensembles werden auch durch die Werte hinsichtlich der letzten 10 Zeitschritte in Tabelle 5.8 gestützt. Anders als bei den Ensembles mit geteiltem Datenpool, steigt die Performanz hierbei nicht mit zunehmender Anzahl an Basisnetzen. Das Ensemble aus vier Basisnetzen übertrifft die Werte des Ensembles mit fünf Basisnetzen. Diese Beobachtung kann auch schon in der Tabelle 5.7 gemacht werden.

Nach der Betrachtung der IMDB-Datensatz-Metriken, werden die Ensembles ohne Datenpool hinsichtlich ihrer Performanz bezüglich des Yahoo-Datensatzes evaluiert.

Tabelle 5.9: Yahoo-Datensatz bezogen auf 34 Zeitschritte: TG, K1 und K2

	TG	PPW K1	Sen K1	F-Maß K1	PPW K2	Sen K2	F-Maß K2
Vergleichsmodell	0.519	**0.589**	0.814	0.670	0.197	**0.087**	**0.095**
Ensemble 2 Netze	**0.524**	0.569	**0.833**	**0.674**	0.155	0.052	0.066
Ensemble 3 Netze	**0.524**	0.575	0.820	**0.674**	**0.200**	0.055	0.069
Ensemble 5 Netze	**0.524**	0.573	0.820	0.672	0.173	0.055	0.070

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 34 Zeitschritte dar.

Tabelle 5.10: Yahoo-Datensatz bezogen auf 34 Zeitschritte: K3 und K4

	PPW K3	Sen K3	F-Maß K3	PPW K4	Sen K4	F-Maß K4
Vergleichsmodell	0.364	0.308	0.303	**0.275**	**0.079**	**0.110**
Ensemble 2 Netze	**0.396**	0.309	0.335	0.185	0.030	0.046
Ensemble 3 Netze	0.394	**0.337**	**0.353**	0.150	0.023	0.037
Ensemble 5 Netze	**0.396**	0.331	0.349	0.157	0.036	0.053

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 34 Zeitschritte dar.

Die Tabellen 5.9 und 5.10 zeigen, dass die Metriken der Ensembles ohne Datenpool weniger deutlich von den Vorhersagen des Vergleichsmodells abweichen und daher nicht von einer generellen Verbesserung durch Ensembles ausgegangen werden kann. Das Vergleichsmodell mit Datenpool kann genauso wenig sämtliche Metriken übertreffen. Es erzielt die besten Werte für die Klassen 2 (*Mixed*) und 4 (*Positive*), während die Ensembles bessere Vorhersagen für die Klassen 1 (*Negative*) und 3 (*Neutral*) machen. Anders als die Ensembles mit geteiltem Datenpool, können die Ensembles ohne Datenpool die unterrepräsentierten Klassen weniger gut vorhersagen, treffen jedoch gute Vorhersagen für die vorherrschende Klasse der negativen Kommentare.

Tabelle 5.11: Facebook-Datensatz bezogen auf 50 Zeitschritte: TG und K1

	TG	PPW K1	Sen K1	F-Maß K1
Vergleichsmodell	**0.607**	**0.454**	0.251	0.285
Ensemble mit 2 Netzen	0.589	0.422	0.366	0.348
Ensemble mit 3 Netzen	0.578	0.403	0.351	0.331
Ensemble mit 4 Netzen	0.589	0.411	0.360	0.343
Ensemble mit 5 Netzen	0.598	0.448	**0.396**	**0.379**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 50 Zeitschritte dar.

Tabelle 5.12: Facebook-Datensatz bezogen auf 50 Zeitschritte: K2 und K3

	PPW K2	Sen K2	F-Maß K2	PPW K3	Sen K3	F-Maß K3
Vergleichsmodell	**0.432**	0.372	0.360	**0.739**	0.678	0.698
Ensemble mit 2 Netzen	0.409	0.366	0.362	0.677	0.756	0.696
Ensemble mit 3 Netzen	0.402	0.352	0.352	0.670	**0.758**	0.693
Ensemble mit 4 Netzen	0.403	**0.382**	0.370	0.681	0.754	**0.699**
Ensemble mit 5 Netzen	0.419	0.366	**0.373**	0.688	0.752	**0.699**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 50 Zeitschritte dar.

Die Ergebnisse der Ensembles ohne Datenpool werden durch die Metriken in Tabelle 5.11 und 5.12 dahin gehend bestärkt, dass dieser Ansatz bezüglich inkrementeller, nicht-balanchierter Datensätze nicht deutlich besser ist, als das Vergleichsmodell. Dennoch sind die durchschnittlichen F-Maße der Ensembles für alle drei Klassen höher, als die es einzelnen neuronalen Netzes mit Datenpool. Das Vergleichsmodell ist dabei präziser, was sich anhand der PPW-Werte für alle drei Klassen ablesen lässt, die höher sind als die der Ensembles.

5.2 Ensembles aus heterogenen neuronalen Netzen

5.2.1 Stacking-Ensemble aus n Basisnetzen mit geteiltem Datenpool

Die Auswertung der heterogenen Ensembles mit Datenpool beginnt mit dem IMDB-Datensatz, gefolgt von den Yahoo- und Facebook-Daten.

Tabelle 5.13: IMDB-Datensatz: Metriken mit unveränderter Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Vergleichsmodell	0.749	0.745	**0.779**	0.748
Ensemble mit 2 Netzen	0.749	0.747	0.754	0.750
Ensemble mit 3 Netzen	0.758	0.752	0.771	0.761
Ensemble mit 5 Netzen	**0.769**	**0.766**	0.774	**0.770**
Ensemble mit 7 Netzen	**0.769**	**0.766**	0.774	**0.770**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf alle 20 Zeitschritte dar. Pro Modell wurden jeweils fünf Durchläufe erfasst, wobei nach jedem Zeitschritt die Metriken aufgezeichnet wurden. Jeder dieser Durchläufe ergibt einen Durchschnittswert, der wiederum gemittelt wird.

Tabelle 5.14: IMDB-Datensatz: Metriken bezogen auf die letzten 10 Zeitschritte mit unveränderter Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Vergleichsmodell	0.769	0.664	**0.802**	0.775
Ensemble mit 2 Netzen	0.775	0.774	0.776	0.775
Ensemble mit 3 Netzen	0.775	0.772	0.781	0.776
Ensemble mit 4 Netzen	0.786	0.783	0.793	0.787
Ensemble mit 5 Netzen	**0.790**	**0.788**	0.794	**0.791**
Ensemble mit 7 Netzen	0.788	**0.788**	0.787	0.788

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen die letzten 10 Zeitschritte dar. Pro Modell wurden jeweils fïf Durchläufe erfasst, wobei nach jedem Zeitschritt die Metriken aufgezeichnet wurden. Jeder dieser Durchläufe ergibt einen Durchschnittswert, der wiederum gemittelt wird.

Auch die Metriken der heterogenen Ensembles in den Tabellen 5.13 und 5.14 unterstützen die Vermutung, dass die genannten Modellkombinationen mit geteiltem Datenpool

bessere Vorhersagen machen, als das einzelne Vergleichsmodell. Wie bei den Ensembles mit homogenen Netzen und geteiltem Datenpool kann das einzelne neuronale Netz nur bei der Sensitivität herausragen. Alle anderen Metriken werden von den Ensembles übertroffen. Bis zu einer Anzahl von fünf Basisnetzen steigern sich die Metriken kontinuierlich.

Die Metriken des Yahoo-Datensatzes geben weiteren Aufschluss über die Performanz der Ensembles in Bezug auf inkrementelles Lernen mit tatsächlichen zeitabhängigen Daten.

Tabelle 5.15: Yahoo-Datensatz bezogen auf 34 Zeitschritte: TG, K1 und K2

	TG	PPW K1	Sen K1	F-Maß K1	PPW K2	Sen K2	F-Maß K2
Vergleichsmodell	0.519	0.589	**0.814**	0.670	0.197	0.087	0.095
Ensemble 2 Netze	0.533	0.595	0.809	0.682	0.276	0.075	0.113
Ensemble 3 Netze	0.533	0.602	0.790	0.681	0.264	0.084	0.124
Ensemble 4 Netze	**0.536**	**0.607**	0.787	**0.683**	0.266	**0.088**	0.129
Ensemble 5 Netze	0.534	0.605	0.785	0.680	**0.271**	0.093	**0.135**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 34 Zeitschritte dar.

Tabelle 5.16: Yahoo-Datensatz bezogen auf 34 Zeitschritte: K3 und K4

	PPW K3	Sen K3	F-Maß K3	PPW K4	Sen K4	F-Maß K4
Vergleichsmodell	0.364	0.308	0.303	0.275	0.079	0.110
Ensemble 2 Netze	0.388	0.356	0.363	0.374	0.150	0.204
Ensemble 3 Netze	0.**415**	0.385	0.384	0.403	0.159	0.215
Ensemble 4 Netze	0.405	**0.394**	**0.391**	**0.418**	**0.190**	**0.249**
Ensemble 5 Netze	0.404	0.388	0.386	0.399	0.178	0.237

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 34 Zeitschritte dar.

Die Metriken des Yahoo-Datensatzes in Tabelle 5.15 und 5.16 zeigen ebenfalls auf, dass die Ensemble-Modelle das einzelne neuronale Netz in allen Metriken, außer der Sensitivität für negative Kommentare, deutlich überragen. In drei Klassen mit deutlich weniger Datenpunkten (K2 bis K4) übertreffen die Ensembles alle Metriken hinsichtlich PPW, Sensitivität und F-Maß.
Abschließend werden noch die Evaluationsergebnisse hinsichtlich des Facebook-Datensatzes

untersucht.

Tabelle 5.17: Facebook-Datensatz bezogen auf 50 Zeitschritte: TG und K1

	TG	PPW K1	Sen K1	F-Maß K1
Vergleichsmodell	0.607	0.454	0.251	0.285
Ensemble mit 2 Netzen	**0.625**	**0.478**	**0.480**	**0.459**
Ensemble mit 3 Netzen	0.612	0.465	0.465	0.441
Ensemble mit 4 Netzen	0.621	0.465	0.464	0.449
Ensemble mit 5 Netzen	0.619	0.452	0.453	0.439

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 50 Zeitschritte dar.

Aus der Tabelle 5.17 lässt sich schnell ablesen, dass ausnahmslos alle Ensembles bessere Werte in den Metriken der ersten Klasse *CSA* erzielen als das einzelne Modell. Das Ensemble aus zwei Basisnetzen, ein CNN und ein LSTM, überragt dabei mit seiner Performanz die anderen Ensembles. Die Hinzunahme weiterer Basisnetze bringt für die Werte der Klasse *CSA* keine Verbesserung.

Tabelle 5.18: Facebook-Datensatz bezogen auf 50 Zeitschritte: K2 und K3

	PPW K2	Sen K2	F-Maß K2	PPW K3	Sen K3	F-Maß K3
Vergleichsmodell	0.432	0.372	0.360	0.739	0.678	0.698
Ensemble mit 2 Netzen	0.443	0.471	0.449	0.750	**0.756**	0.743
Ensemble mit 3 Netzen	0.421	0.470	0.437	0.748	0.750	0.738
Ensemble mit 4 Netzen	0.434	**0.495**	**0.454**	0.759	0.736	0.739
Ensemble mit 5 Netzen	**0.451**	0.475	0.452	**0.766**	0.754	**0.752**

Diese Tabelle stellt die durchschnittlichen Metriken des Vergleichsmodells sowie verschiedener Ensemble-Modelle bezogen auf 50 Zeitschritte dar.

Dagegen verbessern sich die Werte für die Klasse 2 (*DSA*) und 3 (*None*) durch das Hinzufügen weiterer Netze zum Ensemble. Das Ensemble aus fünf heterogenen Netzen erzielt den höchsten positiven prädiktiven Wert für diese beiden Klassen und das beste F-Maß für die Klasse *None*. Das Ensemble aus vier Netzen erzielt die höchsten Werte für die Sensitivität und das F-Maß der Klasse *DSA*.

5.2.2 Adaptiertes Stacking-Ensemble aus 3 Basisnetzen mit geteiltem Datenpool

Die Evaluationsmetriken beziehen sich in diesem Experiment auf eine, in der wissenschaftlichen Arbeit von Maas [Maas u. a. 2011] definierte Testmenge als Teil des vorgestellten IMDB-Datensatzes. Die ersten vier Werte der folgenden Tabelle gehören zu Modellen, die in der Arbeit von Maas vorgestellt wurden, wogegen das Ensemble aus 3 Netzen das Modell aus dieser Arbeit ist.

Tabelle 5.19: IMDB-Datensatz Evaluation auf Testmenge

	Treffergenauigkeit	PPW	Sensitivität	F-Maß
Latent Dirichlet Allocation	67.42	-	-	-
Maas Modell	87.44	-	-	-
Maas Modell+ Bag of Words	88.33	-	-	-
Maas Modell + ungelabelte Daten + BoW	88.90	-	-	-
Ensemble mit 3 Netzen	**89.06**	89.00	89.13	89.06

Diese Tabelle stellt die Metrikenwerte der Modelle aus der Arbeit von Maas und dem Ensemble Modell aus dieser Arbeit dar.

Die Latent Dirichlet Allocation [Blei u. a. 2003] erzielt in der Tabelle 5.19 mit Abstand den schlechtesten Wert für die Treffergenauigkeit. Das folgende, grundlegende Modell aus der Arbeit von Maas erreicht bereits eine deutlich höhere Treffergenauigkeit. Das dritte Model entspricht dem Maas-Modell, wobei die Eingabetexte zusätzlich in einer Bag-of-Words-Representation vorliegen und verarbeitet werden. Der vierte Ansatz verbessert das Modell nicht nur durch Eingabedaten in BoW-Representation, sodern auch mithilfe nicht-gelabelter Datenpunkte. Dies ist in sofern hilfreich, da das Maas-Modell auch Komponenten aus dem Bereich des nicht-überwachten Lernens umfasst. Die Hinzugabe nicht-gelabelter Datenpunkte erreicht in der Arbeit von Maas die höchste Treffergenauigkeit, doch dieser Wert wird von dem Ensemble mit Daten-Pooling übertroffen. Der Wert des Ensembles ist der Durschnittswert des jeweils letzten Ausgabewertes aller drei Experimentdurchläufe nach 10 Zeitschritten. Das Ensemble erreicht die höchste Treffergenauigkeit trotz der Beschränkung der Sequenzlänge auf 210 Wörter und ohne Hinzufügen zusätlicher Datenpunkte, wie es bei dem letzten Maas-Modell mit nicht-gelabelten Daten geschehen ist. Es ist zu vermuten, dass die Erweiterung der Sequenzlänge für das Ensemble-Modell die Metrikenwerte weiter verbessern könnte.

6 Fazit

In dieser Arbeit wurden Stacking-Ensembles aus neuronalen Netzen für inkrementelle Lernverfahren untersucht. Die angewendeten Verfahren sind Adaptionen der bisherigen Verfahren für inkrementelles Lernen mit einem einzelnen neuronalen Netz. Um eine Verbesserung hinsichtlich der Performanz beobachten zu können, wurden die Ensembles gegenüber einem Vergleichsnetz evaluiert, welches durch inkrementelles Lernen mit einem Datenpool trainiert wurde.
Im Folgenden wird auf die, im ersten Kapitel gestellten Forschungsfragen eingegangen und resümiert, ob die Auswertung der Experimente darauf Antworten gibt.
Die erste Frage, die sich stellte: Können Ensemble-Modelle aus neuronalen Deep-Learning-Netzen vergleichbare oder bessere Vorhersagen als ein neuronales Netz mit Data Pooling im Bereich des inkrementellen Lernens liefern? Die Ergebnisse der Ensembles in Kapitel 5 mit geteiltem Datenpool zeigen, dass sie das einzelne neuronale Netz bezüglich der Evaluationsmetriken deutlich übertreffen. Dies betrifft nicht nur die Treffergenauigkeit, sondern auch den positiven prädiktiven Wert und das F-Maß. Ensemble-Modelle aus neuronalen Netzen können im Bereich des inkrementellen Lernens bessere Vorhersagen machen als ein neuronales Netz. Es muss jedoch differenziert werden zwischen den Ensembles mit und ohne Datenpool. Die Ensembles mit geteilten Datenpool waren, außer bei einigen Fällen bezüglich der Sensitivität, immer besser als das Vergleichsnetz. Treffergenauigkeit und F-Maß waren immer besser. Diese Beoachtung lässt sich bei allen drei Datenmengen machen.
Die zweite Forschungsfrage befasste sich mit der Performanz der behandelten Ensembles für unbalancierte Datensätze im Vergleich zu einem einzelnen neuronalen Netz. Weiterhin stellte sich die Frage, ob Ensembles besser darin sein könnten, unterrepresäntierte Klassen zu lernen?
Auch diese Frage lässt sich durch die vorliegenden Ergebnisse bejahen. Da die Ensemble-Modelle mit geteiltem Datenpool in dieser Arbeit generell besser performen, als das Vergleichsmodell, trifft dies auch auf die unbalancierten Datensätze zu, die Teil der evaluierten Gesamtmenge an Daten sind. Die Yahoo- und Facebook-Kommentare sind nicht balanciert und für beide Datensätze weisen die Ensembles aus homogenen und heterogenen Basisnetzen mit geteiltem Datenpool eine bessere Perfomanz auf, als das einzelne neuronale Netz. Alle unterrepräsentierten Klassen dieser Datensätze werden von den Ensembles besser gelernt. Sowohl der positive prädiktive Wert, die Sensitivität als auch das F-Maß der Ensembles sind deutlich höher, als die Werte des Vergleichsmodells. Dieses macht in beiden unterrepräsentierten Klassen nicht-balancierter Datensätze schlechtere Prognosen, als die Stacking-Ensembles.
Die dritte Forschungsfrage widmete sich der Anzahl der Basisnetze der Ensembles und ihren möglichen Einfluss auf die Vorhersagequalität. Bei den Experimenten und deren

Auswertung lässt sich feststellen, dass die Performanz, die die Anzahl der Basisnetze bewirkt, von dem jeweiligen Datensatz abhängt. Dennoch ist bei homogenen Ensembles mit geteiltem Datenpool die Tendenz zu erkennen, dass mehr Netze die Vorhersagen verbessern können. Bei diesen Ensembles, angewendet auf die IMDB-Filmbewertungen, werden bis zu sieben Netze untersucht, da sich die Performanz mit jedem weiteren Netz steigert. Auch die Yahoo-Kommentare werden mit mehreren Netzen besser klassifiziert. Die Facebook-Kommentare werden mit einem Ensemble aus vier Netzen am besten klassifiziert, ein fünftes Basisnetz steigert die Performanz nicht. Die Steigerung der Anzahl der Basisnetze hat den geringsten Einfluss auf Ensembles ohne Datenpool. Hier lässt sich nicht die Tendenz beobachten, dass mehr Netze bessere Vorhersagen machen. Der Zusammenhang zwischen der Netzanzahl und guten Prognosen scheint hier noch stärker abhängig vom jeweiligen Datensatz zu sein. Die Anzahl der Basisnetze ist als zusätzlicher Hyperparameter von Stacking-Ensembles einzustufen, der einer spezifischen Feinjustierung bedarf. In dem Ensembles mit heterogenen Basisnetzen zeigte sich bezüglich des Facebook-Datensatzes auch, dass die variierende Netzanzahl Einfluss auf die Metriken der einzelnen Klassen hat. Die Bestimmung der Anzahl der Basisnetze kann, nach einem evaluierten Experiment davon abhängig gemacht werden, welche Klasse das Ensemble besonders gut vorhersagen soll.
Die vierte Frage zielte auf die Annahme ab, dass Ensembles mit heterogenen Basisklassifikatoren bessere Vorhersagen machen, als Ensembles mit homogenen Klassifikatoren. Möglichst uneinheitliche, variantenreiche Algorithmen lernen im besten Fall ganz unterschiedliche Aspekte des Datensatzes und verbessern dadurch das Gesamtergebnis. Diese Annahme kann durch die vorliegenden Resultate der Experimente nicht bestätigt werden. Die Ensembles mit homogenen Basisnetzen prognostizieren Klassenzugehörigkeiten für den IMDB- und den Facebook-Datensatz besser, als heterogene Ensembles. Beide Ensemble-Arten mit geteiltem Datenpool, weichen in ihren Prognosen weniger voneinander ab als zu dem einzelnen neuronalen Netz. Die gering schlechtere Performanz der heterogenen Ensembles kann auch eine Folge der Auswahl der Basisnetze sein. Bei den LSTMs und GRU-Zellen lässt sich ein langsamerer Lernprozess beobachten, als bei den CNNs. Da die homogenen Basisnetze CNNs als Basistypen haben, können sie mit weniger Trainingsepochen genauer klassifizieren. Die Ensembles werden pro Zeitschritt nur eine Epoche trainiert. Die Annahme, dass mehr Trainingsepochen die Vorhersagen der LSTMs und GRU-Zellen verbessern, müsste in weiteren Experimenten gezeigt werden. Zu viele Epochen beim geteilten Datenpool könnten zu einer Überanpassung führen, die die Prognosen wieder verschlechtert. Die bessere Performanz der heterogenen Ensembles hinsichtlich der Yahoo-Daten zeigt jedoch auch, dass die Wahl der Ensemble-Architektur abhängig vom Datensatz sein kann. Da es sich des Weiteren bei neuronalen Netzen um diskriminative Modelle handelt und traditionelle Ensembles oft aus diskriminativen und generativen Klassifikatoren bestehen, könnte dies den geringen Unterschied der Evalutionsmetriken beider Ansätze erklären. Die besondere

Eigenschaft tiefer neuronaler Netze selbstständig Merkmale zu lernen, die hohe Dimensionalität durch Wortvektoren und der Zufallsfaktor bei der Gewichtsinitialisierung können Klassifikatoren hervorbringen, die trotz gleicher Basisarchitektur, ganz unterschiedliche Aspekte der Trainingsdaten gelernt haben.
Die fünfte Forschungsfrage, die gestellt wurde, behandelt die unterschiedlichen inkrementellen Lernverfahren mit Ensembles aus neuronalen Netzen und deren zu untersuchende Unterschiede hinsichtlich der Qualität ihrer Evaluationsmetriken. In dieser Arbeit wurden zwei grundsätzlich verschiedene Verfahren dargelegt und in Experimenten evaluiert: Inkrementelles Ensemble-Lernen mit einem Datenpool und inkrementelles Ensemble-Lernen ohne einen Datenpool. Die Begutachtung beider Verfahren zeigt, dass inkrementelles Ensemble-Lernen mit Datenpool deutlich bessere Werte in den Metriken erzielt, als ohne Datenpool. Die Ensembles ohne Datenpool machen für alle drei Datensätze Vorhersagen von geringerer Qualität, als die Ensembles mit Datenpool, unabhängig davon, ob die Basisnetze Diversität aufweisen oder nicht.
Die sechste Frage behandelte die Beurteilung der zeitlichen Komponente und eine damit verbundene mögliche Zeitersparnis, die ein Ensemble-Modell einbringen könnte. Die Ensemble-Modelle mit Daten-Pooling können theoretisch eine Zeitersparnis zur Folge haben, sofern die Basis-Modelle parallel trainiert werden. Der zusätzliche Aufwand, das Training zu Parallelisieren, müsste jedoch miteinbezogen werden. Weiterhin ist die Zeit, die mit dem parallelen Training eingespart wird, nur bei einer geringen Anzahl von Basisnetzen relevant, denn bei k Netzen trainiert jedes Netz mit $k\text{-}1/k$ Anteilen der gesamten Datenmenge. Die Ensembles ohne Datenpool bringen theoretisch langfristig eine deutliche Zeitersparnis ein, da hier die Trainingsdauer nur abhängig ist von wenigen Teildatensätzen zu bestimmten Zeitpunkten. Bei einer konstanten Größe der Daten pro Zeitschritt bleibt auch die Trainingszeit nahezu gleich. Auch die Teilnetze der Ensembles ohne Datenpool können parallel trainiert werden.
Die siebte, letzte Frage hatte die Performanz vom inkrementellen Ensemble-Lernen gegenüber einem klassischen Lernverfahren zum Gegenstand. Können inkrementelle Lernverfahren mit Ensembles ähnlich gute Klassifikatoren erzeugen wie klassische Methoden mit einer Trainings- und Testmenge? Die Resultate des letzten Experiments mit dem Ensemble aus heterogenen Basisnetzen und Data-Pooling, angewendet auf die Trainings- und Testmenge aus der wissenschaftlichen Arbeit von Maas [Maas u. a. 2011] zeigen, dass die angewendete Methode mit einem heterogenen Ensemble eine überragende Treffergenauigkeit erzielt. Das Ensemble übertrifft die Metrikenwerte sämtlicher Modelle aus der Arbeit von Maas, trotz einer beschränkten Sequenzlänge von 210 Wörtern pro Datenpunkt.
Die in dieser Arbeit untersuchten Stacking-Ensembles für inkrementelles Lernen, bilden nur eine geringe Teilmenge der möglichen Architekturen, sodass in Zukunft viele weitere Forschungsfragen gestellt und durch Experimente evaluiert werden können. Die Vielzahl der Netztypen als Basisklassifikatoren sowie die mögliche Auswahl verschiedener

Meta-Klassifikatoren bringen Herausfordungen mit sich, da die Kombinationsmöglichkeiten hier kaum absehbar sind. Weiterhin können Stacking-Ensembles mit mehr als zwei Schichten untersucht werden. Die zeitliche Komponente des inkrementellen Lernens fügt eine weitere Dimension hinsichtlich der Komplexität hinzu und ermöglicht eine Unmenge von denkbaren Auswahlkriterien für die Trainingsdatensätze. Teilnetze könnten mit saisonalen Daten, auf Untergruppen von Nutzertypen oder mit verschiendenen Arten von Transfer-Learning, bei denen die Netze mit distinkten Datensätzen trainiert werden, lernen.
Die beschriebenen, möglichen Szenarien zeigen die Herausfoderung und die Problematik auf, die Stacking-Ensembles im Bereich des inkrementellen Lernens mit sich bringen. Der hohe Grad an Komplexität, durch die vielen Optionen, macht es deutlich schwieriger für einen Datensatz eine optimale Architektur zu finden und erfordert zusätzliche Ressourcen bei der Parametersuche. Dennoch kann ein Ensemble ein einzelnes Modell hinsichtlich der Performanz übertreffen, ohne dass Hyperparameter sowie Details der Architektur und des inkrementellen Lernansatzes optimiert sein müssen. Ist ausschließlich die Verbesserung der Metriken das Ziel ohne einen Anspruch auf ein stark optimierte Ensemble, dann sind die hier vorgestellten Verfahren mit Data-Pooling erfolgversprechend.
In einer weiterführenden Arbeiten könnten die untersuchten Stacking-Ensembles mit inkrementellen Lernverfahren verglichen werden, die nicht auf neuronalen Netzen basieren. Dadurch können weitere Erkenntnisse in Bezug auf die Performanz gewonnen werden.
Tiefe, komplexe neuronale Netze als Basisklassifikatoren sind jedoch durch ihre Fähigkeit, selbstständig Merkmale zu lernen, klassischen maschinellen Lernalgorithmen überlegen. Dadurch können etwaige Ensemble-Modelle einfach adaptiert werden, um beispielsweise statt Text, Bilder oder Audiospuren zu klassifizieren.
Die Anwendung von Stacking-Ensembles mit tiefen neuronalen Netzen bleibt auch in naher Zukunft ein spannendes Forschungsfeld, wenn neue Netzarchitekturen in den wissenschaftlichen Fokus rücken.

A Anhang

Der Quellcode für sämtliche Experimente ist frei verfügbar und befindet sich unter folgender URL:

$$https://github.com/Cyberlander/ILmEanNfvDidZ \tag{A.1}$$

Literaturverzeichnis

[Akbik u. a. 2018] AKBIK, Alan ; BLYTHE, Duncan ; VOLLGRAF, Roland: Contextual String Embeddings for Sequence Labeling. In: *COLING 2018, 27th International Conference on Computational Linguistics*, 2018, S. 1638–1649

[Blei u. a. 2003] BLEI, David M. ; NG, Andrew Y. ; JORDAN, Michael I.: Latent Dirichlet Allocation. (2003)

[Breiman 1994] BREIMAN, Leo: Bagging Predictors. (1994)

[Cauwenberghs u. Poggio 2001] CAUWENBERGHS, Gert ; POGGIO, Tomaso: Incremental and Decremental Support Vector Machine Learning. (2001)

[Cho u. a. 2014] CHO, Kyunghyun ; MERRI ¨ENBOER, Bart van ; GULCEHRE, Caglar ; BAHDANAU, Dzmitry ; BOUGARES, Fethi ; SCHWENK, Holger ; BENGIO, Yoshua: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. (2014)

[Chung u. a. 2014] CHUNG, Junyoung ; GULCEHRE, Caglar ; CHO, KyungHyun ; BENGIO, Yoshua: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. (2014)

[Csáji 2001] CSÁJI, Balázs C.: Approximation with Artificial Neural Networks. (2001)

[Efron u. Tibshirani 1994] EFRON, Bradley ; TIBSHIRANI, Robert J.: An Introduction to the Bootstrap. (1994)

[Felsberg 2017] FELSBERG, Michael: Five years after the Deep Learning revolution of computer vision: State of the art methods for online image and video analysis. (2017)

[Freund u. Schapire 1996] FREUND, Yoav ; SCHAPIRE, Robert E.: Experiments with a New Boosting Algorithm. (1996)

[Geman u. a. 1992] GEMAN, Stuart ; BIENSTOCK, Elie ; DOURSAT, Rene: Neural Networks and the Bias/Variance Dilemma. (1992)

[Grave u. a. 2017] GRAVE, Piotr Bojanowski E. ; JOULIN, Armand ; MIKOLOV, Tomas: Enriching Word Vectors with Subword Information. (2017)

[Hara u. a. 2017] HARA, Kensho ; KATAOKA, Hirokatsu ; SATOH, Yutaka: Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition. (2017)

[Hochreiter 1998] HOCHREITER, Sepp: The Vanishing Gradient Problem During Learning Recurrent Neural Nets And Problem Solutions. (1998)

[Hochreiter u. Schmidhuber 1997] HOCHREITER, Sepp ; SCHMIDHUBER, Jürgen: Long Short-Term Memory. (1997)

[Hubel u. Wiesel 1968] HUBEL, David H. ; WIESEL, Torsten N.: Receptive Fields and Functional Architecture of Monkey Striate Cortex. (1968)

[Hulley u. Marwala 2007] HULLEY, Gregor ; MARWALA, Tshilidzi: Evolving Classifiers: Methods for Incremental Learning. (2007)

[Karaoguz 2018] KARAOGUZ, Ethem C.: Adaptive Learning Strategies for Neural Paraphrase Generation. (2018)

[LeCun u. a. 1989] LECUN, Y. ; BOSER, B. ; DENKER, J. S. ; HENDERSON, D. ; HOWARD, R. E. ; HUBBARD, W. ; JACKEL, L. D.: Backpropagation Applied to Handwritten Zip Code Recognition. (1989)

[Lessmann u. a. 2015] LESSMANN, Stefan ; BAESENS, Bart ; SEOW, Hsin-Vonn ; THOMAS, Lyn C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. (2015)

[Losing u. a. 2016] LOSING, Viktor ; HAMMER, Barbara ; WERSING, Heiko: Choosing the Best Algorithm for an Incremental On-line Learning Task. (2016)

[Luo u. a. 2005] LUO, Zhongwen ; LIU, Hongzhi ; WU, Xincai: Artificial Neural Network Computation on Graphic Process Unit. (2005)

[Maas u. a. 2011] MAAS, Andrew L. ; DALY, Raymond E. ; PHAM, Peter T. ; HUANG, Dan ; NG, Andrew Y. ; POTTS, Christopher: Learning Word Vectors for Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA : Association for Computational Linguistics, June 2011, 142–150

[Mikolov u. a. 2013] MIKOLOV, Tomas ; SUTSKEVER, Ilya ; CHEN, Kai ; CORRADO, Greg ; DEAN, Jeffrey: Distributed Representations of Words and Phrases and their Compositionality. (2013)

[Napoles u. a. 2017] NAPOLES, Courtney ; TETREAULT, Joel ; ROSATA, Enrica ; PROVEN-

ZALE, Brian ; PAPPU, Aasish: Finding Good Conversations Online: The Yahoo News Annotated Comments Corpus. In: *Proceedings of The 11th Linguistic Annotation Workshop*. Valencia, Spain : Association for Computational Linguistics, April 2017, S. 13–23

[Olah 2015] OLAH, Christopher: Understanding LSTM Networks. (2015)

[Polikar u. a. 2002] POLIKAR, Robi ; BYORICK, Jeff ; KRAUSE, Stefan ; MARINO, Anthony ; MORETON, Michael: Learn++: A Classifier Independent Learning Algorithm for Supervised Neural Networks. (2002)

[Polikar u. a. 2000] POLIKAR, Robi ; UDPA, Lalita ; UDPA, Satish S. ; HONAVAR, Vasant: Learn++: An Incremental Learning Algorithm for Multilayer Perceptron Networks. (2000)

[Polikar u. a. 2001] POLIKAR, Robi ; UDPA, Lalita ; UDPA, Satish S. ; HONAVAR, Vasant: Learn++: An Incremental Learning Algorithm for Supervised Neural Networks. (2001)

[Saffari u. a. 2009] SAFFARI, Amir ; LEISTNER, Christian ; SANTNER, Jakob ; GODEC, Martin ; BISCHOF, Horst: On-line Random Forests. (2009)

[Schmidhuber 2014] SCHMIDHUBER, Jürgen: Deep Learning in Neural Networks: An Overview. (2014)

[Srivastava u. a. 2014] SRIVASTAVA, Nitish ; HINTON, Geoffrey ; KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; SALAKHUTDINOV, Ruslan: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. (2014)

[Töscher u. Jahrer 2009] TÖSCHER, Andreas ; JAHRER, Michael: The BigChaos Solution to the Netflix Grand Prize. (2009)

[Wiedemann 2017] WIEDEMANN, Gregor: FB dataset collection. (2017)

[Wolpert 1992] WOLPERT, David H.: Stacked Generalization. (1992)

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Hamburg, den _____ Unterschrift: _____