Multilinguale Spracherkennung mit tiefen neuronalen Netzen

1st Björn Beha
Hochschule Furtwangen University
Furtwangen, Germany
Bjoern.Beha@hs-furtwangen.de

2nd Phillip Ginter
Hochschule Furtwangen University
Furtwangen, Germany
phillip.ginter@hs-furtwangen.de

3rd Suhay Sevinc
Hochschule Furtwangen University
Furtwangen, Germany
Suhay.Sevinc@hs-furtwangen.de

***Zusammenfassung*—Dieser Artikel beschäftigt sich mit dem Ansatz des tiefen maschinellen Lernens im Bereich der multilingualen Spracherkennung. Die Arbeit setzt sich mit der Forschungsfrage auseinander, welches Verfahren heute genutzt wird, um aktuelle Spracherkennungssysteme zu realisieren. Aus dieser Untersuchung geht hervor, dass die hohe Genauigkeit der heutigen Spracherkennung über Long short-term memory-Netzwerke erreicht wird, die sich an vorherige Ereignisse erinnern können. Die Arbeit zeigt, warum diese Art von neuronalen Netzen ideal für Sequenzen von Tönen geeignet ist. In diesem Kontext wird untersucht, wie die Erkennung mehrerer Sprachen funktioniert, auch wenn nur geringe Mengen an Trainingsdaten verfügbar sind. Vor allem diese Knappheit von Ressourcen stellt eine Herausforderung dar, da ohne eine Menge von markierten Datensätzen keine Muster und Gesetzmäßigkeiten der Sprache erkannt und beurteilt werden können. Die Abhandlung zeigt, dass dies über das gemeinsame Nutzen einzelner Töne gelöst wird und wie sich ein solches System trainieren lässt. Weiterhin bestehende Herausforderungen werden diskutiert und es wird geklärt, welche Ansätze man in Zukunft verfolgt, um eine natürliche Interaktion zwischen Mensch und Maschine zu ermöglichen.**

***Index Terms*—Tiefes Lernen, tiefe neuronale Netze, Spracherkennung, Recurrent Neural Netzworks, Long short-term memory, LSTM**

I. Einleitung

Systeme zur Spracherkennung finden eine zunehmende Verbreitung und Beliebtheit im alltäglichen Leben. Das Spektrum dieser Anwendungen ist dabei Vielfältig und reicht vom Diktieren von Nachrichten über das Steuern von Geräten bis hin zum Einsatz in Autos. Dabei ist die Qualität der Spracherkennung und die Reaktion des Systems ein entscheidender Faktor, um die Interaktion so natürlich wie möglich zu gestalten [1]. Allerdings ergibt sich hier ein Hindernis für mehrsprachige Nutzer. Die natürliche Interaktion wird limitiert, da automatische Spracherkennungssysteme den Anwender auf eine voreingestellte Sprache beschränken. In den meisten herkömmlichen Systemen werden Sprachen sowie Dialekte unabhängig voneinander betrachtet. Es wird für jede Sprache ein separates akustisches Modell trainiert [Paper A Real-Time End-to-End Multilingual Speech Recognition Architecture]. Bei weltweit etwa 7000 gesprochenen Sprachen ist es daher nur konsequent, multilinguale Spracherkennungssysteme zu entwickeln [2]. Allerdings erfordert ein solches System einen entsprechenden Satz an markierten Trainingsdaten, um wiederkehrende Muster der Sprache zu erkennen. Dieser Umstand sorgt für erhebliche Qualitätsunterschiede zwischen den Sprachen, da nicht alle Sprachen über solche Datensätze verfügen. Um die Knappheit der beschrifteten Trainingsdaten zu kompensieren, nutzt man den Ansatz der geteilten Hidden Layer [Paper Using Language Adaptive]. Dieser Ansatz stützt sich auf das Zusammenführen aller Daten, um so eine gemeinsame Nutzung der Phoneme zu gewährleisten. Phoneme stellen dabei eine abstrakte Repräsentation aller Laute einer Sprache dar . Um genaue akustische Modelle für eine große Anzahl von Sprachen effizient und effektiv zu trainieren, um die Kosten, die beim Training dieser Modelle entstehen zu reduzieren und um neue Anwendungsszenarien zu unterstützen, besteht ein wachsendes Interesse an der Entwicklung mehrsprachiger Spracherkennungssysteme [1].

Zu Beginn erfolgt die Erläuterung des grundlegenden Aufbaus eines solchen Systems. Dabei werden die Unterschiede zu einem monolingualen Systems hervorgehoben. Darauf aufbauend wird erläutert, wie die Identifikation sowie die Erkennung von Sprachen funktioniert. Anschließend werden die Recurrent Neural Networks sowie deren Erweiterung, die Long short-term memory-Netzwerke beschrieben, welche heute im Bereich der multilingalen Spracherkennung verwendet werden. Dabei wird beleuchtet, wie dieses funktioniert und welche Vorteile gegenüber anderen tiefen neuronalen Netzen bestehen. Das Training eines solchen Modells folgt im Anschluss mit einer abschließenden Diskussion bezüglich weiterhin bestehender Probleme und zukünftiger Ansätze.

II. Verwandte Arbeiten

Dieses Kapitel stellt exemplarisch wichtige Arbeiten vor, welche mit dem Thema des Artikels in Beziehung stehen. Es gibt viele Forschungsarbeiten auf dem Gebiet der mehrsprachigen und sprachübergreifenden Spracherkennung. Der Artikel konzentriert sich allerdings nur auf diejenigen, die Recurrent Neural Networks bzw. LSTM-Netzwerke verwenden. Der Schwerpunkt dieser Arbeit liegt bei dem Untersuchen dieser Verfahren zur Realisierung entsprechender Systeme sowie die sich hier ergebenden Vorteile gegenüber bisheriger Verfahren. Dabei wird kein detaillierter Vergleich verschiedener Modelle aufgeführt. Die Abhandlung lehnt an das Werk Automatic

Speech Recognition - A Deep Learning Approach von Dong Yu und Li Deng [1] an. Das Buch liefert einen genauen Überblick über die Thematik. Die Grundlagen bezüglich automatischer Spracherkennungssysteme, konventioneller Ansätze und Trainingsverfahren sowie die Architektur mehrsprachiger Systeme werden hier beschrieben. Ein weiteres für diesen Artikel interessantes Werk ist das Buch Sprachverarbeitung von Beat Pfister und Tobias Kaufmann, in welchem Grundlagen und Methoden der Sprachsynthese und Spracherkennung genau beschrieben werden. Des Weiteren gibt es die Arbeit von (...) [https://arxiv.org/abs/1703.07090]. Hier wird ein alternatives LSTM-Netzwerk vorgeschlagen, welches den konventionellen Ansatz übertrifft. Auch in den Arbeiten von (...) [x4] und (...) [x5] werden Recurrent Neural Networks im Bezug auf Spracherkennung untersucht. Dabei wird ebenfalls die Effizienz von LSTM-Netzwerken unter anderem durch Experimente bestätigt.

III. Hintergrund

Ein multilinguales Spracherkennungssystem besteht aus mehreren Komponenten. Abbildung 1 illustriert dabei die Pipeline dieser Module. Die tatsächliche Erkennung der Sprache ist dabei trivial. Die Schallwellen, die beim Sprechen produziert werden, lassen sich über einen elektroakustischen Wandler (Mikrophon) in ein elektrisches Signal umwandeln. Dieses elektrische Tonsignal wird daraufhin in digitalisiert bzw. in Bits konvertiert (sampling) und über Vorverarbeitung entsprechend aufbereitet, um es in ein neuronales Netz zu speisen [4]. Diese Daten lassen sich anschließend in Sequenzen aufteilen, aus denen wiederum die benötigten Features extrahiert werden. Abbildung 1 stellt diesen Teilschritt als Spektrogramm dar, welches das gesamte Frequenzspektrum visualisiert. Die Feature-Vektoren müssen hier so gewählt werden, dass die kleinste, effizienteste Menge für die Sprachverarbeitung herausgefiltert wird. Unnötige Informationen müssen bereits vor dem Einsatz des Decoders entfernt werden. Die gewonnenen Features dienen schließlich als Eingabe für die Sprachidentifikation. Die daraus gewonnene Information bezüglich der gesprochenen Sprache in Kombination mit diesen Features werden wiederum als Eingabe für den Decoder genutzt. Unter Zuhilfenahme des akustischen Modells sowie des Sprach- und Lexikalmodells, wird der gesprochene Text analysiert und bewertet, um eine Vorhersage des Gesprochenen zu treffen. Die drei Modelle werden von Bäckström [3] wie folgt beschrieben:

- *Akkustikmodell.* Die Menge an Daten, die das neuronale Netz darüber informiert wie die Zusammenhänge zwischen Phonemen und einem konkreten Audiosignal sind. Die Phoneme können sowohl kontextabhängig als auch kontextfrei sein. Erlernt wird das Modell durch Audioaufnahmen und den zugehörigen Abschriften – die akkustischen Trainingsdaten.
- *Lexikalmodell.* Dieses Modell bildet eine Sequenz von Phonemen – die durch das Akustikmodell gewonnenen wurden – auf gültige Wörter einer Sprache ab. Hierfür werden textuelle Trainingsdaten eingesetzt.
- *Sprachmodell.* Die Wahrscheinlichkeit für einen syntaktisch und semantisch korrekten Satz wird genutzt um aus Sequenzen von Wörtern – die durch das Lexikalmodell gewonnen wurden – gültige Wortsequenzen zu bilden. Beispielsweise folgt auf das englische Wort „thank“ mit einer sehr hohen Wahrscheinlichkeit das Wort „you“ oder „god“. Für das Training dieses Modells werden textuelle Trainingsdaten eingesetzt.

Jedes dieser drei Modelle muss separat trainiert werden. Das führt zu einer erhöhten Komplexität verglichen mit dem Trainieren eines einzelnen gemeinsamen Modells. Darum wird in den letzten Jahren vermehrt der Ansatz verfolgt, Ende-zu-Ende-Systeme zu entwickeln. Dabei werden die drei Modelle als ein System trainiert und eingesetzt. Chan et al. [4] und Prabhavalkar et al. [5] zeigen hierzu verbesserte Ergebnisse, verglichen mit mehreren Systemen mit einzelnen Modellen.

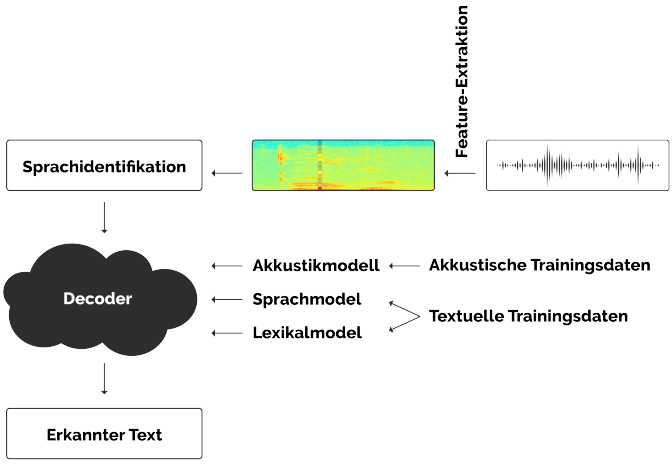

Abbildung 1. Pipeline eines Spracherkennungssystems (Eigene Darstellung, in Anlehnung an: [3])

IV. Sprachidentifikation

Systeme zur Sprachidentifikation werden eingesetzt um die Sprache eines Audiosignals zu klassifizieren. Um genauere Vorhersagen zu treffen ist dies der erste Schritt in multilingualen Spracherkennungssystemen. Erst mit der Identifikation können eingehende Laute entsprechend zugeordnet werden. Ausdrücke und Grammatikregeln lassen sich somit Ableiten und erhöhen die Präzision der Systeme [6]. Die Einsatzgebiete lassen sich laut Zissman et al. [7] in zwei Kategorien einteilen, dem Vorverarbeiten für maschinelle Systeme und die Vorverarbeitung für menschliche Zuhörer. Unter ersterem wird ein sprachgesteuertes System verstanden, welches beim Einsprechen des Texts die Identifikation durchführt um anschließend mithilfe des korrekten Sprachmodells die gesprochene Sprache erkennt. Eine Vorverarbeitung für menschliche Zuhörer geht gleich vor. Der Unterschied liegt darin, dass die weitere Verarbeitung nicht von der Maschine vorgenommen wird, sondern durch einen Menschen. Die Erkennung dient nur zum entsprechenden Delegieren.
Eine Sprache wird von Menschen und Maschinen anhand der

Unterschiede zwischen den Sprachen identifiziert. Zissman et al. [7] nennt als Unterschiede die folgenden vier Charakteristika:

- *Phonologie.* Hier wird die Häufigkeit und Verteilung von Phonemen und Phonen betrachtet. Ein Phon ist der tatsächlich produzierte Ton, der beim Sprechen entsteht.
- *Morphologie.* Sprachen unterscheiden sich in den Wortstämmen, dem Vokabular und der Art, wie Wörter geformt werden.
- *Syntax.* Sätze haben in unterschiedlichen Sprachen, unterschiedliche Satzstrukturen.
- *Prosodie.* Tempo, Rhythmus, Pausen und Tonhöhen unterscheiden sich von Sprache zu Sprache.

A. Architektur

Die Umsetzung der Sprachidentifikation spiegelt sich in der gewählten Architektur eines automatischen Spracherkennungssystems wieder. Gonzalez-Dominguez et al. [8] unterscheidet hierbei zwischen drei möglichen Umsetzungen.

Die erste ist es, ein universelles Modell zu trainieren, indem alle Sprachen als Eingabe möglich sind. Die Hidden Layer eines neuronalen Netzes teilen sich die Repräsentationen für Phoneme und in den Ausgabeschichten wird die Sprache erkannt [1]. Vorteile ergeben sich durch die gemeinsame Nutzung von Phonemen. Das funktioniert, da es zwischen den einzelnen Sprachen jeweils gleiche Phoneme gibt, die somit nicht mehr erneut erlernt werden müssen. Möchte man eine neue Sprache trainieren, so kann man auf die bereits vorhandenen Strukturen aufbauen und diese mitnutzen. Außerdem wird das gesamte System nicht so komplex, wie es mit mehreren einzelnen monolinguale Systeme der Fall wäre [6].

Eine weitere Möglichkeit – der Identifikation einer Sprache – ist der Einsatz eines dedizierten Systems. In Abbildung 1 ist dieser Architekturansatz dargestellt. Hierbei wird anhand eines Teilstückes des Eingabesignals die Sprache bestimmt. Niesler et al. [9] gibt für dieses Teilstück eine durchschnittliche Dauer von 2,3 Sekunden an, bestätigt jedoch auch, dass mit zunehmender Länge die Genauigkeit zunimmt. Für die die konkrete Implementierung eines dedizierten Systems zur Sprachidentifikation gibt es mehrere Ansätze [9]. Neben dem Einsatz eines Gaussian mixture models (GMM) kann man auch Neuronale Netze dafür einsetzen. Diese beiden Ansätze können wiederum in zwei Kategorien unterteilt werden; die Erkennung von Wörtern oder von Phons. Ein Nachteil der Spracherkennung anhand eines Teilstückes ist eine erhöhte Latenz. Diese kann abhängig von der zu erzielenden Genauigkeit unterschiedlich lang sein. Sollte außerdem die Sprache in diesem Schritt falsch erkannt worden sein, breitet sich der Fehler aus und führt zu einem möglicherweise falschen Endresultat.

Die dritte und zugleich letztgenannte Möglichkeit setzt auf mehrere monolinguale Spracherkennungssysteme [8]. Das Eingangssignal wird simultan von mehreren Systemen mit jeweils eigenen Modellen verarbeitet. Anhand der größten Übereinstimmung mit einer Sprache wird am Ende dann die passende Sprache ausgewählt. Das löst die Probleme des vorherigen Ansatzes und mit einer höheren Wahrscheinlichkeit wird die richtige Sprache ausgewählt. Nachteilig ist der erhöhte Rechenaufwand durch dein Einsatz mehrerer Systeme.

In den nachfolgenden Kapiteln gehen wir von dem ersten Ansatz aus, da es das Teilen von Ressourcen ermöglicht, einen einheitlich Ansatz bietet und es zunehmendes Interesse in diesem Gebiet gibt [1], [10].

V. Multilinguale Spracherkennung

Die Kernidee der mehrsprachigen Spracherkennung ist bei den verschiedenen Architekturen dieselbe. Die Hidden-Layer des Deep Neural Networks können als intelligentes Merkmalsextraktionsmodul betrachtet werden, welches aus mehreren Quellsprachen trainiert wird. Nur die Ausgabeschicht liefert eine direkte Übereinstimmung mit den relevanten Klassen. So lassen sich die Extraktoren für eine Reihe verschiedener Sprachen gemeinsam nutzen. Wie im folgenden Kapitel erläutert wird, lässt sich somit besonders das Problem beim Lernen der tiefen neuronalen Netze entgegenwirken. Diese lassen sich aufgrund ihrer Parameter und dem sogenannten Backpropagation-Algorithmus langsamer trainieren als andere Modelle. Ein weiterer Vorteil, den dieser Ansatz bietet ist, dass auch mit Sprachen, die nur einen geringen Satz an markierten Trainingsdaten bietet, erlernt werden können. Indem Elemente anderer Sprachen gemeinsam genutzt und übertragen werden, lässt sich dieses Problem kompensieren. Merkmale, die aus diesen neuronalen Netzen extrahiert werden, lassen sich kombinieren, um so die Erkennungsgenauigkeit zu verbessern [1]. Eine gemeinsame Nutzung dieser Elemente wird ermöglicht, indem Phoneme zwischen den Sprachen zusammen genutzt werden. Phoneme sind als kleinste, bedeutungsunterscheidende Einheiten der Lautsprache definiert. Phoneme werden zur Repräsentation der Aussprache genutzt. Um die genannten Ansätze der Sprachidentifikation sowie die der Spracherkennung zu nutzen, müssen Beziehungen zwischen den akustischen Signalen der Sprachen erkannt werden. Jede Sprache besitzt dabei ihre eigenen Charakteristika.

In der Sprachübergreifenden Erkennung gibt es einen Satz aus trainierten sowie untrainierten bzw. schlecht trainierten Phonemen, die erkannt werden müssen. Die Töne einer Sprache werden mit einem ähnlichen bzw. dem ähnlichsten trainierten Ton einer anderen Sprache ersetzt. Angenommen ein System wurde auf die deutsche Sprache trainiert, kennt das Phonem /y/, welches im Wort süß vorkommt nicht, lässt sich ein ähnlicher Ton einer anderen Sprache nutzen, um das Wort trotzdem korrekt vorherzusagen. Unbekannte Wörter lassen sich somit aus bekannten Phonemen zusammensetzen [5].

Ein Transfer dieses Modells ist trivial. Es wird eine neue Softmax-Schicht angelegt und auf eine bestimmte Sprache trainiert, während das gesamte Netzwerk auf die neue Sprache abgestimmt wird. Die Softmax-Funktion wird hierbei lediglich zur Klassifikation verwendet und sorgt dafür, dass der Output immer in einem gleichen Bereich liegt [6]. Die Ausgabeknoten dieser Schicht entsprechen den Senonen der Zielsprache. Senonen beschreiben hier lediglich das Betrachten des lautlichen Kontextes der einzelnen Phoneme [1]. Diese Kontexte können

komplex sein [7].
Ein entsprechende Architektur ist in Abbildung (...) illustriert. Sie zeigt die gemeinsam genutzten Schichten, die die Merkmale extrahieren sowie die unterschiedlichen Input-Datensätze der einzelnen Sprachen. Jede Sprache hat ihre eigene Softmax-Ebene. Wird ein neuer Datensatz in das System gegeben, werden nur die sprachspezifische Schicht sowie die Hidden Layer angepasst. Andere Softmax-Schichten bleiben intakt. Kommt eine weitere Sprache hinzu, wird leidglich eine neue Softmax-Ebene an das vorhandene Netzwerk angefügt und trainiert, wie es in der Abbildung zu sehen ist [1].

Tatsächlich bringt dieser Ansatz eine Verbesserung gegenüber monolingualer Netzwerke. Ein Vergleich eines monolingualen Deep Neural Networks und eines multilingualen Deep Neural Networks ist in Tabelle (...) aufgeführt. Das monolinguale Netzwerk wurde hierbei nur mit jeweils einer der Sprachen französisch, deutsch, spanisch und italienisch trainiert, während das multilinguale System mit allen vier Sprachen trainiert wurde. Dabei wird die prozentuale Wortfehlerrate (Word error rate, WER) angegeben. Es ist zu erkennen, dass das multilinguale System das monolinguale in allen Sprachen übertrifft. Diese Verbesserung ist dem sprachübergreifenden Wissen zuzuschreiben [1]. Die Steigerungen sind zusätzlich relativ in Prozent angegeben.

In [2] wurden eine Reihe weiterer Versuche durchgeführt, um die Wirksamkeit eines solchen Systems zu evaluieren. Dabei wurde zwei verschiedene Zielsprachen verwendet. Zum einen das amerikanische Englisch, welches phonetisch nahe an den europäischen Sprachen der oben aufgeführten Tabelle liegt und Mandarin-Chinesisch, welches kaum Gemeinsamkeiten zu europäischen Sprachen bietet. Beim genauen Vorhersagen des gesprochenen kommt die Sprachidentifikation ins Spiel, durch welche Wörter ausgeschlossen werden können, die ebenfalls in Frage kommen, allerdings zum Wortschatz einer anderen Sprache gehören. Es wird hier mit statistischen Modellen gearbeitet, um anzugeben mit welcher Wahrscheinlichkeit welches Wort vorkommt oder aufeinander folgen können. Dabei gibt es verschiedene Lösungsansätze, um das gesprochene vorherzusagen. Oft werden tiefe neuronale Netze in Verbindung mit Hidden Markov-Modellen eingesetzt. Diese hybriden Systeme werden in der Literatur oft untersucht und beschrieben. Ein allerdings leistungsfähigeres Modell bieten die Recurrent Neural Networks. Diese Form von neuronalen Netzen werden heutzutage eingesetzt und erreichen hohe Genauigkeiten [1].

VI. Recurrent Neural Networks

Im Bereich der Spracherkennung werden heutzutage sogenannte Recurrent Neural Networks eingesetzt, durch welche die Netzwerke ihre Erkennungsgenauigkeit erreichen. Das Modell dieser Netze erlaubt gerichtete, zyklische Verbindungen zwischen den Neuronen, wodurch es mit einem temporalen Verhalten ausgestattet wird. Recurrent Networks sind somit ideal zum Lernen von Datensequenzen geeignet. Sprache, also kontinuierliche Audiostreams fallen somit ebenfalls in das Anwendungsgebiet. Diese Form von neuronalen Netzen unterscheidet sich grundlegend von einem Feed Forward Deep Neural Network, da es nicht nur basierend auf Eingaben arbeitet, sondern auch auf interne Zustände zurückgreift. Diese internen Zustände speichern die vergangenen Informationen in der zeitlichen Reihenfolge, in welcher diese verarbeitet wurden. Somit ist ein Recurrent Neural Network dynamischer, als ein Deep Neural Network, welches lediglich eine statische Eingabe-Ausgabe-Transformation durchführt. Dabei wird eine Erweiterung des Backpropagation-Algorithmus eingesetzt. Die Backpropagation-Through-Time-Methode sorgt für das Berechnen der Gradienten. Diese werden im Gegensatz zum Standardalgorithmus über die einzelnen Zeitschritte aufsummiert. In dieser Erweiterung des Backpropagation, welche in Recurrent Neural Networks eingesetzt wird, werden lediglich die Parameter einzelnen Zeitschritte zwischen den Ebenen geteilt. In Abbildung (...) ist ein vereinfachtes Modell illustriert [1].

Die Abbildung zeigt links ein Netzwerk A, welches über eine Rückkopplung verfügt. Dieses Netzwerk bekommt einen Input x und gibt einen Output bzw. Zustand E zurück. So kann die Information von einem Schritt zum nächsten gelangen. Das ausgerollte Netzwerk wird rechts daneben dargestellt. Es zeigt eine Folge von Iterationen. S bezeichnet den jeweiligen Schritt und E den entsprechenden Hidden State, welcher sich beim Eingeben von Daten zum Zeitpunkt s ergibt. Ein Recurrent Neural Network gibt somit nicht nur den Input an die nächste Iteration, sondern zusätzlich den daraus resultierenden Zustand. Vorhergehende Schritte beeinflussen so die darauf folgenden [1].
Dies führt allerdings zu einem Problem. Dadurch, dass die Zustände immer weiter angepasst werden, verschwindet bzw. verschwimmt im Laufe der Zeit Information. Dies ist als Vanishing Gradient Problem bekannt und ergibt sich dadurch, dass Recurrent Neural Networks nicht in der Lage sind, auf Informationen zurückzugreifen, die weit in der Vergangenheit liegen. Der Kontext kann demnach bereits vergessen worden sein. Dieses Phänomen wird an Abbildung (...) deutlich. Das einmalige Anwenden der Sigmoidfunktion sorgt dafür, dass ein beliebiger Eingabewert zwischen -1 und 1 liegt. Wendet man die Funktion mehrmals an, flacht die Kurve ab und es kann keine Änderung mehr erkannt werden. Die Ausgaben streben alle den gleichen Wert an [https://deeplearning4j.org/lstm.html].

So kann eine inkorrekte Vorhersage stattfinden. Aufgrund dessen wurden Long-Short-Term-Memory-Netzwerke entwickelt, die zur Lösung des Problems beitragen. Dabei werden Recurrent Neural Networks mit einer Speicherstruktur erweitert, was zur Namensgebung führt. Diese Netzwerke sind in der Lage, anhand des Kontextes zukünftige Wörter vorherzusagen und so ihre Genauigkeit zu erhöhen. Auch beim Lernen und Erkennen verrauschter, geräuschverzerrter oder hallender Aufnahmen bzw. schlechten Bedingungen beim bearbeiten der Merkmale kann diese Form von Netzwerken bessere Ergebnisse erzielen [1]. Die Verbesserte Genauigkeit gegenüber gewöhnlichen tiefen neuronalen Netzen wird in verschiedenen Arbeiten belegt [1][3][x6].
Diese Form von Netzwerken erlauben die Erkennung zeitlich ausgedehnter Muster und von Zusammenhängen zeitlich

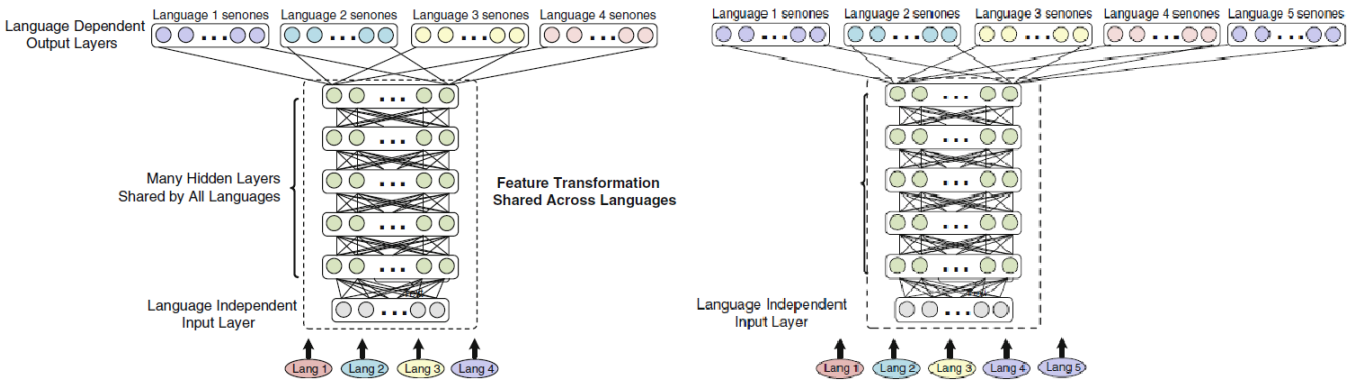

Abbildung 2. Hinzufügen einer neuen Sprache [11]

	FRA	DEU	ESP	ITA
Test set size (words)	40k	37k	18k	31k
Monolingual DNN WER	28.1%	24.0%	30.6%	24.3%
Mulitlingual DNN WER	27.1% (-3.6%)	22.7 (-5.4%)	29.4 (-3.9%)	23.5% (-3.3%)

Tabelle I
RELATIVE WORTFEHLERRATE

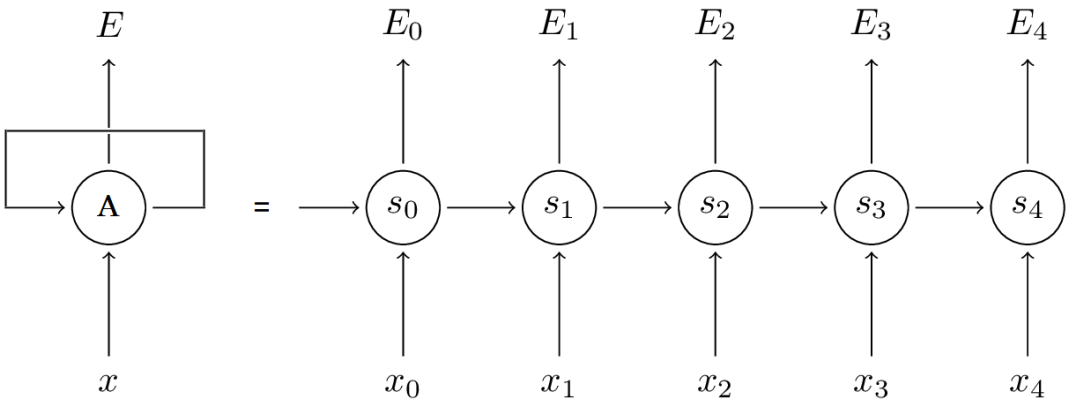

Abbildung 3. Modell des Recurrent Neural Network [11]

getrennter Ereignisse. Somit eignen sie sich um Zeitreihen zu verarbeiten und vorherzusagen. Sogar, wenn zwischen wichtigen Ereignissen Verzögerungen liegen, die eine unbekannte Länge aufweisen. Die grundsätzliche Idee dabei ist es über elementweise Multiplikationen den Informationsfluss in dem Netzwerk zu steuern. Eine LSTM-Zelle kann als intelligente Netzwerkeinheit betrachtet werden, welche Informationen über einen Zeitraum speichern kann. Dies wird durch eine Gating-Struktur erreicht. Die Information passiert verschiedene Gatter. Es wird bestimmt, wann es wichtig ist, sich an eine vorhergehende Eingabe zu erinnern, wann sich die Zelle Informationen weiter merken oder diese vergessen sollte und wann sie die Information ausgibt. Ein Gatter ist dabei nichts weiter, als eine Reihe von Multiplikationen bzw. Matrixoperationen [1]. Das System ist somit in der Lage aus dem Kontext heraus genaue Vorhersagen zu treffen, wodurch die Spracherkennung präziser wird. Eine Darstellung einer LSTM-Zelle ist Abbildung (...) zu sehen. Der Input x wird zur Zeit t von mehreren Quellen in die Zelle eingespeist. Dabei wird x an alle Gatter übergeben. Jedes Gatter i (Input), f (Forget), c (Memory cell), o (Output) und h (Hidden vector bzw. der resultierende Zustand) haben dabei ihre eigene Gewichtungsfunktion [x4].

Allerdings ist es selbst heute nicht möglich das Spracherkennungsproblem allgemein zu lösen. Spracherkennungssysteme werden somit nur für bestimmte Anwendungsfälle oder Szenarien konzipiert. Mit einer solchen Spezialisierung auf entsprechende Anwendungsgebiete können zum höhere Genauigkeiten erreicht werden. Zudem wird nicht so viel Rechenleistung und Speicher benötigt [4]. Vor allem bei der multilingualen Spracherkennung besteht die Schwierigkeit Gemeinsamkeiten verschiedener Sprachen zu nutzen, um Sprachen mit wenig Trainingsdaten mit einer ausreichenden Genauigkeit anzubieten. Es gilt die Sprachen zu finden, die zur

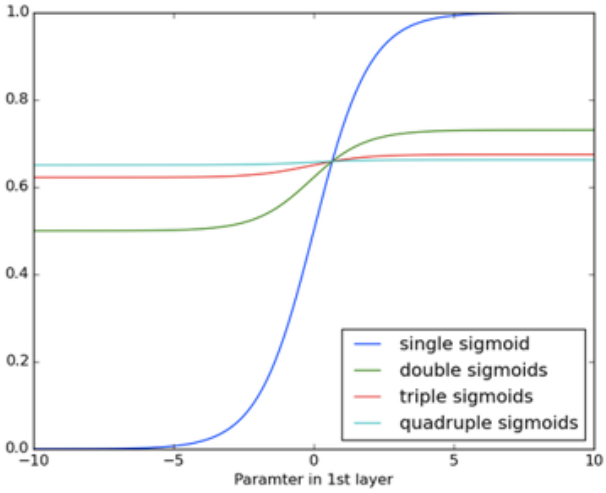

Abbildung 4. Modell des Recurrent Neural Network [deeplearning4j.org/lstm.html] [11]

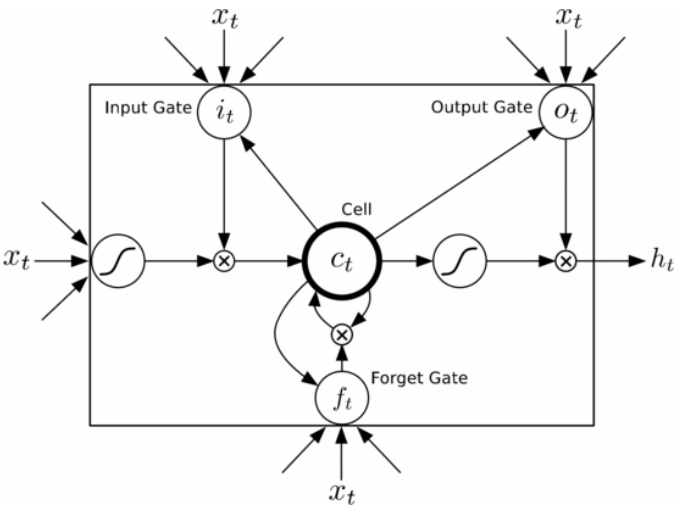

Abbildung 5. Modell des Recurrent Neural Network [deeplearning4j.org/lstm.html] [11]

besten Erkennungsleistung der neuen Sprache führen. Dabei müssen Beziehungen zwischen den Sprachen erkannt werden. Problematisch ist auch, dass gleiche Phoneme je nach Sprecher und Sprache variieren, was dazu führt, dass Phoneme nur im Kontext betrachtet werden (sogenannte Triphon-Zustände) [1]. Spracherkennungen verschiedener Unternehmen erreicht heute niedrige Wortfehlerraten (Google liegt bei 4,9%), was auf die Menge von Trainingsmaterial zurückzuführen ist [x1] [x2] [x3]. In der Literatur wird außerdem gezeigt, dass verschiedene Kombinationen von tiefen neuronalen Netzen und LSTM-Netzwerken zu einer weiteren Verbesserung führen [x4].

VII. Trainingsvorgang

Der Trainingsvorgang basiert auf ein mehrschichtiges tiefen neuronalen Netzwerks. Das Netzwerk aus Neuronen besteht aus drei Schichten:

Input-Schicht
Hidden-Schicht
Output-Schicht

Die Input-Schicht stellt die Eingangsdaten dar, welche als Trainingsmaterial dient. Bei diesen Daten handelt es sich um Sprachaufnahmen. Bei Bedarf können diese Aufnahmen durch Filter vorverarbeitet werden. Anschließend werden die beschrifteten Daten in die Netztopologie eingespeist. Eine Vorklassifizierung der Sprache führt zu einer erhöhten Spracherkennungsrate von mehreren Sprachen, da diese Methode sich für mehrere Klassifiziuerungsklassen eignet. In der Hidden-Schicht geschieht das Training. Hier werden die Phoneme der Sprachen extrahiert und gelernt. Dabei wird die Sigmoid-Funktion als Aktivitätsfunktion eingesetzt (s. Formel 1). Diese Funktion beschreibt den Korrelation zwischen Input-Wert und Aktivitätslevel eines Neurons dar. Zudem wird der Input-Wert auf die X-Achse eingetragen. Auf die Y-Achse wird der zugehörige Aktivitätslevel eingetragen. Der Aktivitätslevel wird durch eine Ausgabefunktion in den Output transformiert, den das Neuron an andere Neuronen weitersendet [13]. Das Netz wird beginnend von der Input-Schicht bis Output-Schicht vollständig durchlaufen.

$$sigm(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Sobald die Output-Schicht erreicht ist, wird das Netz rückwärts durchlaufen. Dieses Verfahren wird auch Gradientenabstiegsverfahren genannt und wird benötigt, um fehlerhafte Kantengewichte herauszufinden und anzupassen. Die Kantengewichte des Netzes werden mit null initialisiert. Die Ableitung der Sigmoid-Funktion wird bei der Korrekturberechnung notwendig (s. Formel 2). Bei größeren Datenmengen entsteht ein Nachteil, welches sich auf die Wissensausprägung des Netzes auswirkt. Beim rückwärts durchlaufen entsteht ein Wissensverlust [12]. Dieser Verlust wird durch das Maxima der Ableitung $sigm(x)'$ repräsentiert. Dieser kann bis zu 25 % betragen. Der entstehende Verlust würde die Klassifizierungsrate des Trainingsmodels reduzieren, welches in Abbildung 6 dargestellt ist [14].

$$sigm(x)' = \frac{e^x}{(e^x + 1)^2} \tag{2}$$

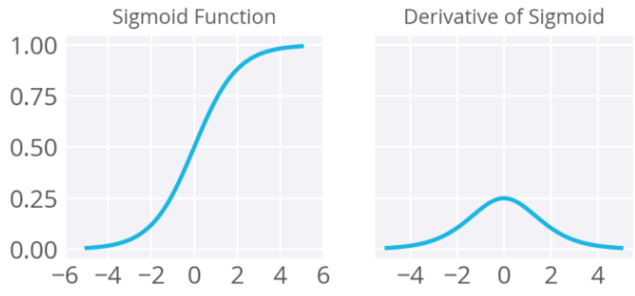

Abbildung 6. Darstellung der Sigmoid-Funktion und dessen Ableitung [14]

Anstelle der Sigmoid-Funktion als Aktivitätsfunktion wird in den State-Of-the-Art-Deep-Learning-Netzen rectified linear

units ($ReLUs$) verwendet (s. Formel 3). Diese Funktion ist dem menschlichen Neuron am ähnlichsten und bringt zudem eine erhöhte Verarbeitungsgeschwindigkeit mit sich [15]. Die Berechnung der Kantengewichte erfolgt durch Formel 4).

$$y_j = ReLU(x_j) = max(0, x_j) \tag{3}$$

$$x_j = b_j + \sum x_{ij} * y_j \tag{4}$$

Als Nächstes folgt die Output-Schicht, welches die Eingangsdaten zu den Klassen (Zielsprachen) zuordnet. Diese Schicht ist als Softlayer konfiguriert, welches die Klassen in eine eindimensionale Matrix kategorisiert. Dabei ist die Matrix in dem Zahlenintervall $[0, 1]$ normalisiert. Die endgültige Sprachidentifikation geschieht über normalisierte Werte, welches in Abbildung 7 dargestellt wird. Die Werte können in Wahrscheinlichkeiten ausgedrückt werden, in dem die Matrixwerte mit dem Faktor 100 multipliziert werden [14]. Die Vorhersa-

Abbildung 7. Klassenzuordnung über Wahrscheinlichkeiten in der Softmax-Konfiguration [14]

gen der Output-Schicht geschieht durch die Funktion $p(j)$ (s. Formel 5). Dabei steht der Index l für die jeweilige Sprache.

$$p(j) = \frac{exp(x_j)}{\sum_l exp(x_l)} \tag{5}$$

Für den vorhin erwähnten Gradientenabstiegsverfahren wird ebenfalls eine Kostenfunktion benötigt. Diese geschieht durch Cross-Entropy-Loss-Funktion.

$$C = \sum_l t_j * log(p_j) \tag{6}$$

Diese Funktion misst die Abweichungen der Kantengewichte der Netztopologie und passt diese rückwirkend an. Der Cross-Entropy-Verlust nimmt zu, wenn der vorhergesagte Wert von der tatsächlichen Beschriftung abweicht [16]. Bei t_j handelt es sich um die Klasse, für die der Verlust berechnet wird [11].

A. Netztopologie

Die Netztopologie beschreibt die Infrastruktur des Netzes. Die Auswahl der Topologie bestimmt die Qualität des Trainingsvorgangs. Eine zu geringe Anzahl der Neuronen führt zu einer niedrigen Spracherkennungsrate. Wiederum eine zu hohe Anzahl würde zu überhöhten Trainingsdauer führen. Aufgrund dessen fallen Topologien von Ansatz zu Ansatz unterschiedlich aus, welche unterschiedliche Spracherkennungsresultate liefern [12]. In dieser Arbeit wird der Topologienvorschlag von Gonzales et al. betrachtet. Für die Eingangsdaten werden 40 Filterbanken verwendet. In der Input-Schicht werden 26 Neuronen eingesetzt. Um unerwünschte Latenzzeiten zu vermeiden wird ein asymmetrischer Kontext verwendet. Die Hidden-Schicht beträgt vier Ebenen mit einer Gesamtzahl von 2560 Neuronen. Die Output-Schicht enthält wie bereits erwähnt eine Softmax-Konfiguration, dessen Dimension der Anzahl der Zielsprachen entspricht. Dies ist bei der Erkennung von multilingualen Sprachen eine erforderliche Konfiguration. [11].

B. Verbesserung des Trainingsverfahrens durch Multitasking learning (MTL)

Bei maschinellem Lernen wird der Fokus auf das Optimieren bestimmte Metriken, wie beispielsweise Klassifizierungsgenauigkeit und Trainingsdauer. Daraufhin wird das Modell soweit optimiert, bis die Leistung des Modells nicht mehr gesteigert werden kann [17]. Das Lernen der einzelnen Sprachen läuft sequenziell ab. Hier setzt das Multitasking Learning (MTL) ein. Es werden mehrere Lernaufgaben gleichzeitig erledigt statt sequentiell, um das Trainingsvefahren effizienter zu gestalten. Das führt zu einer verbesserten Lerneffizienz und Vorhersagegenauigkeit. Im Klassifizierungskontext zielt MTL darauf ab, die Leistung mehrerer Klassifizierungsaufgaben zu verbessern, indem sie gemeinsam erlernt werden [18]. Ein Beispiel hierfür ist ein Spamfilter. Der Schlüssel zur erfolgreichen Anwendung von MTL besteht darin, dass die Aufgaben miteinander verknüpft werden. Dies bedeutet nicht, dass die Aufgaben ähnlich sein müssen. Stattdessen bedeutet es, dass Aufgaben auf verschiedene Ebenen abstrahiert und geteilt werden. Dabei kann das Wissen zwischen Aufgaben übertragen werden, welches die Trainingsdauer deutlich verkürzt. MTL ist vor allem nützlich, wenn die Größe des Trainingssatzes im Vergleich zur Modellgröße klein ist. Dabei wird grundsätzlich zwei Arten von MTL unterschieden: Hard parameter sharing und soft parameter sharing.

Hard parameter sharing stellt das meist genutzte Art dar [17]. Es wird normalerweise auf die Hidden-Schicht angewendet, indem die Aufgaben gemeinsam gelernt werden, während die spezifischen Aufgaben separat gelernt werden. Dies wird in Abbildung 8 dargestellt. Dieser Ansatz reduziert das Risiko von overfitting erheblich. Je mehr Aufgaben gleichzeitig gelernt wird, desto mehr muss das Modell eine Repräsentation finden, die alle Aufgaben erfassen muss. Dadurch ist die Chance auf overfitting deutlich geringer [17] [18].

VIII. Diskussion und Ausblick

Die menschliche Sprache ist der natürlichste Weg etwas zu kommunizieren, so ist es nicht verwunderlich, dass das Interesse an dem Deep Learning-Ansatz zur Spracherkennung und den damit verbundenen Anwendungen steigt. In dieser Arbeit wurde das Gegenstück zu den konventionellen, stochastischen Modellen beleuchtet - die Recurrent Neural Networks mit der Erweiterung der LSTM-Struktur. Dabei geht hervor, dass bei diesen Netzen die einzelnen Neuronen nicht isoliert betrachtet werden

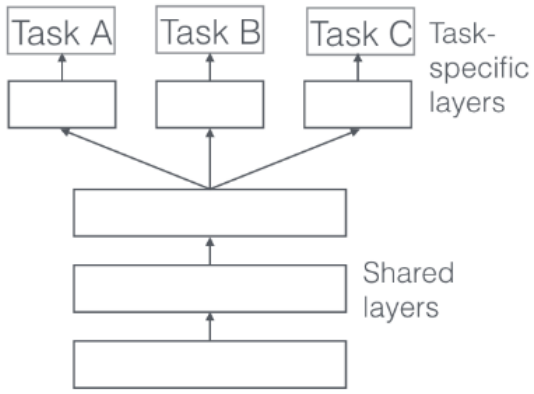

Abbildung 8. Hard parameter sharing auf die Hidden-Schicht angewendet [14].

können. Vielmehr hängt deren Zustand und Aktivierung von den Aktivitäten anderer Neuronen bzw. Zellen ab. Vorhergehende Ereignissen beeinflussen den Zustand. Durch die dynamische Rekursion und den Gattern wird ein Gedächtnis geschaffen, mit welchem sich die Netze an vergangene Zustände erinnern können und aufgrund dieser Erfahrungen genauere Vorhersagen treffen. Vor allem bei Datensequenzen ist dies hilfreich. Da die menschliche Sprache lediglich eine Sequenz von Tönen ist, eignet sich diese Form von Netzwerken ideal. Es ging außerdem hervor, dass mit der multilingualen Spracherkennung bessere Ergebnisse erzielt werden, als mit der monolingualen Erkennung. Dies ist auf das gemeinsame Nutzen der Phoneme zurückzuführen. Heutige Genauigkeiten beim Erkennen von Sprachen erreichen die Wort-Fehler-Rate eines Menschen. Somit ist das reine Verstehen bald nicht mehr das hauptsächliche Problem. Die Autoren sind der Meinung, dass eine natürliche Interaktion mit einem Spracherkennungssystem dennoch schwierig bleibt, solange das System keine Kenntnisse über seine Umwelt hat. Beispielsweise klingen im Deutschen die Worte Meer und mehr gleich, haben jedoch nichts gemeinsam. Diese Homophone lassen sich zwar verstehen, das Spracherkennungssystem erkennt allerdings nicht den Kontext und es könnte zu einer inkorrekten Vorhersage führen. Es bestehen auch weitere, zahlreiche limitierende Faktoren. Wie in der Arbeit beschrieben zählen hierzu vor allem auch Sprachen, die keine ausreichenden Ressourcen bieten. Das Mapping von Wörtern und Sequenzen aus Phonemen braucht Linguistikexperten und stellt eine Herausforderung dar. Schließlich müssen sämtliche Phoneme der Sprache identifiziert werden. Auch der Stil beim Sprechen verändert sich ständig und ist nie gleich zwischen unterschiedlichen Sprechern. Gesprochene Wörter beeinflussen die Betonung der nächsten Worte.

Die Zukunft im Bereich des maschinellen Lernens bleibt spannend. Wir sind überzeugt, dass in Zukunft fortschrittlichere Deep Learning-Architekturen für effektivere Spracherkennungssysteme entwickelt werden, die den hier diskutierten Netzwerken in vielerlei Hinsicht überlegen sind. Das Verständnis über die Struktur der Sprache, deren Dynamik und ihrer Repräsentation treiben den Forschungsfortschritt weiter voran. Ansätze, die in der Literatur zu finden sind, gehen davon aus, weitere Informationsquellen einzubeziehen, um die Qualität weiter zu verbessern. In diesem Zusammenhang wird in [1] das nutzen visueller Daten erwähnt. Dabei werden Merkmale aus interessanten Gesichtsregionen extrahiert. Da visuelle Informationen unabhängig von akustischem Rauschen sind, soll hier eine Verbesserung erzielt werden. Offen bleibt die Frage, welche Ansätze in Zukunft entwickelt werden, um die Interaktion mit Spracherkennungssystemen zu einem natürlichen Prozess zu machen. Da Maschinen die Welt nicht verstehen wie wir, ist es schwierig aus Tönen den gesamten Kontext zu verstehen. Weitere Forschungen können hier anknüpfen und sich mit potentiellen Möglichkeiten zur Lösung dieses Problems auseinandersetzen.

LITERATUR

[1] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.

[2] G. F. Simons and C. D. Fennig, Eds., *Ethnologue: Languages of the World, Twenty-first edition*. Dallas, Texas: SIL International. Online version, 2018. [Online]. Available: http://www.ethnologue.com

[3] T. Bäckström, "Speech recognition overview," 2016, abgerufen am 15. Juni 2018. [Online]. Available: https://mycourses.aalto.fi/pluginfile.php/366508/mod_resource/content/1/slides_recognition_handout.pdf

[4] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015. [Online]. Available: http://arxiv.org/abs/1508.01211

[5] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," 2017. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/pdfs/0233.PDF

[6] C. Bartz, T. Herold, H. Yang, and C. Meinel, "Language identification using deep convolutional recurrent neural networks," *CoRR*, vol. abs/1708.04811, 2017. [Online]. Available: http://arxiv.org/abs/1708.04811

[7] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Commun.*, vol. 35, no. 1-2, pp. 115–124, Aug. 2001. [Online]. Available: http://dx.doi.org/10.1016/S0167-6393(00)00099-6

[8] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. J. Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 749–759, June 2015.

[9] T. Niesler and D. Willett, "Language identification and multilingual speech recognition using discriminatively trained acoustic models," 2006.

[10] S. Hara and H. Nishizaki, "Acoustic modeling with a shared phoneme set for multilingual speech recognition without code-switching," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec 2017, pp. 1617–1620.

[11] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. J. Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 749–759, 2015.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] neuronalesnetz, "Neuronale netze - eine einführung - aktivität," 2018, (Accessed on 06/24/2018).

[14] GitHub, "Kulbear/deep-learning-nano-foundation," 2017, (Accessed on 2018-04-15). [Online]. Available: https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions

[15] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *38th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, 2013.

[16] M. Cheatsheet, "Loss functions — ml cheatsheet documentation," 2017, (Accessed on 2018-05-16). [Online]. Available: http://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html

[17] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, (Accessed on 2018-05-16). [Online]. Available: http://ruder.io/multi-task/

[18] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. H. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," 2015.