

Multilinguale Spracherkennung mit tiefen neuronalen Netzen

1st Björn Beha

dept. name of organization (of Aff.)

name of organization (of Aff.)

City, Country

email address

2nd Philipp Ginter

dept. name of organization (of Aff.)

name of organization (of Aff.)

City, Country

email address

3rd Suhay Sevinc

dept. name of organization (of Aff.)

name of organization (of Aff.)

City, Country

email address

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 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 Networks, Long short-term memory, LSTM

I. EINFÜHRUNG

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 [2]. Allerdings ergibt sich hier ein Hindernis für mehrsprachige Nutzer. Die natürliche Interaktion wird dadurch behindert, da viele ASR-Systeme 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 [3]. 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 [2]. Zu Beginn erfolgt eine Erläuterung bestimmter Grundlagen, die für das weitere Verständnis der Arbeit wichtig sind. 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 multilingualen 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 [2] an. Das Buch liefert einen genauen Überblick über die Thematik. Die Grundlagen bezüglich auto-

matischer 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

III. HINTERGRUND

A. Pipeline

Die Abbildung 1 zeigt die Komponenten, aus dem ein multilinguales Spracherkennungssystem besteht. Am Anfang steht ein analoges Audiosignal das digitalisiert wird. Aus diesen Daten werden aus kleinen Sequenzen Feature-Vektoren extrahiert. In Abbildung 1 ist als Beispielausgabe für diesen Teilschritt ein Spektrogramm dargestellt – bei dem die einzelnen Frequenzen bildlich dargestellt werden. Die Feature-Vektoren müssen so gewählt werden, dass die kleinste effizienteste Menge für die Sprachverarbeitung entsteht und unnötige Informationen bereits vor dem Decoder herausgefiltert werden. Die gewonnenen Feature-Vektoren werden als Eingabe für die Sprachidentifikation genutzt. Die Information über die identifizierte Sprache kombiniert mit den Feature-Vektoren bilden die Eingabe für den Decoder. Der Decoder bildet aus dieser Eingabe, dem Akustik-, Sprach- und Lexikalmodell den gesprochenen Text. Die drei Modelle werden von [4] wie folgt beschrieben:

- *Akkustikmodell.* Die gesprochene Sprache wird abgebildet durch einzelne Phoneme.
- *Lexikalmodell.* Alle gültigen Wörter einer Sprache.
- *Sprachmodell.* Die Wahrscheinlichkeit für einen syntaktisch und semantisch korrekten Satz. Beispielsweise folgt auf das englische Wort „thank“ mit einer sehr hohen Wahrscheinlichkeit das Wort „you“ oder „god“.

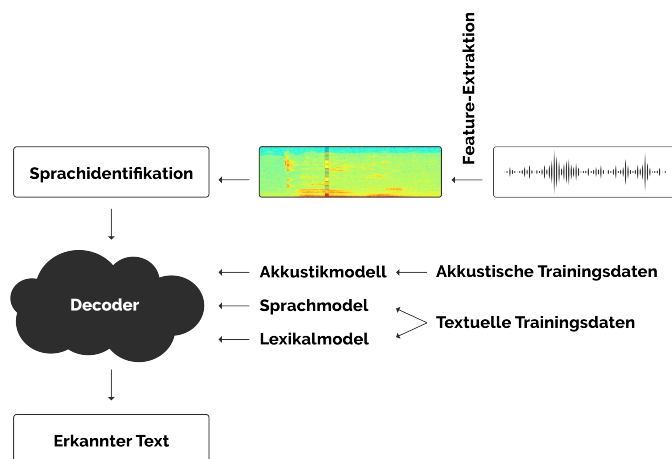


Abbildung 1. Pipeline eines Spracherkennungssystems (Eigene Darstellung, in Anlehnung an: [4])

Die Trainingsdaten für die beschriebenen Modelle werden in die folgenden zwei Gruppen unterteilt [4]:

- *Akkustische Trainingsdaten.* Diese Daten werden genutzt um das Eingangssignal auf Phoneme abzubilden.
- *Textuelle Trainingsdaten.* Bilden Grammatik und semantische Informationen in den Modellen ab.

IV. SPRACHIDENTIFIKATION

A. Überblick

Systeme zur Sprachidentifikation werden eingesetzt um die Sprache eines Audiosignals zu klassifizieren. Üblicherweise ist dies der erste Schritt in automatischen multilingualen Spracherkennungssystemen. Ohne die richtige Ausgangssprache, können Ausdrücke und Grammatikregeln nicht erkannt werden können [5]. Die Einsatzgebiete lassen sich laut Zissman et al. [6] in zwei Kategorien einteilen. Zum Einen ist es die Vorverarbeitung für maschinelle Systeme und zum Anderen für menschliche Zuhörer. Ein Beispiel für ersteres könnte ein sprachgesteuertes System – an einem Flughafen - zur Flugauskunft sein. Ohne menschliches Zutun könnte das System zuerst die Sprachidentifikation ausführen und danach mithilfe des korrekten Sprachmodells die gesprochene Sprache erkennen. Eine Vorverarbeitung für menschliche Zuhörer könnte ein Notrufsystem sein, bei dem zuerst die Sprache des Anrufers erkannt. Anhand dieser wird der Anruf dann an einen Mitarbeiter der für diese Sprache zuständig ist weitergeleitet. Besonders relevant ist solch ein System, da in einem Notfall jede Minute über Leben und Tod entscheiden kann.

B. Wie wird eine Sprache identifiziert?

Eine Sprache wird von Menschen und Maschinen anhand der Unterschiede zwischen den Sprachen identifiziert. Zissman et al. [6] 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.

C. Architektur

Die Umsetzung der Sprachidentifikation spiegelt sich in der gewählten Architektur eines automatischen Spracherkennungssystems wieder. Gonzalez-Dominguez et al. [7] 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 [2]. 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 [5].

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. [8] gibt für dieses Teilstück eine durchschnittliche Dauer von 2,3 Sekunden an, bestätigt jedoch auch, dass mit zunehmender Länge die Genauigkeit zunimmt. Für die konkrete Implementierung eines dedizierten Systems zur Sprachidentifikation gibt es mehrere Ansätze [8]. 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 [7]. 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 den 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 [2], [9].

V. MULTILINGUALE SPRACHERKENNUNG

Die Kernidee der mehrsprachigen Spracherkennung ist bei den verschiedenen Architekturen dieselbe. Die Hidden-Layer des Deep Neural Networks können als ein 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 vorherigen Kapitel bereits erläutert wurde, 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 übertragen werden. Merkmale, die aus diesen neuronalen Netzen extrahiert werden, lassen sich kombinieren, um so die Erkennungsgenauigkeit zu verbessern

[1]. Eine gemeinsame Nutzung wird ermöglicht, indem Phoneme gemeinsam genutzt werden. Phoneme sind als kleinste, bedeutungsunterscheidende Einheiten der Lautsprache definiert. Phoneme werden zur Repräsentation der Aussprache genutzt. Um obige Ansätze 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 müssen mit einem ähnlichen bzw. dem ähnlichsten trainierten Ton einer anderen Sprache ersetzt werden. Beispielsweise gibt es den Ton /y/, welcher im Wort ‚süß‘ vorkommt. Wenn ein System nun mit der deutschen Sprache genutzt wird, welches nur in anderen Sprachen trainiert wurde, muss der ähnlichste Sound zu /y/ gefunden werden [5]. So lassen sich phonetische Klassen aus mehreren Sprachen unterscheiden. Ein Transfer des Modells ist trivial. Es wird lediglich eine neue Softmax-Schicht angelegt und trainiert. Die Softmax-Funktion wird zur Klassifikation verwendet [6]. Die Ausgabenknoten dieser Schicht entsprechen dann den Senonen der Zielsprache. Senonen beschreiben lediglich das Betrachten des lautlichen Kontextes der einzelnen Phoneme und stellen gebundene Triphonzustände dar [1]. Die Kontexte können komplex sein [7]. Die Softmax-Schicht wird nur auf die entsprechende Sprache trainiert. Weitere Verbesserungen lassen sich erzielen, indem das gesamte Netzwerk zusätzlich auf die neue Sprache abgestimmt wird. Ein solche Architektur ist in Abbildung (...) illustriert. Sie zeigt die gemeinsam genutzten Schichten, die die Merkmale extrahieren sowie die unterschiedlichen Input-Datensätze. Jede Sprache hat ihre eigene Softmax-Ebene. Mit der Softmax-Funktion lassen sich hier die Zustände des akustischen Modelles vorhersagen bzw. entsprechende Wahrscheinlichkeiten schätzen. Wird ein neuer Datensatz in das System gegeben, werden nur die sprachspezifische Schicht sowie die Hidden Layer angepasst. Andere Softmax-Schichten bleiben intakt. Nach dem trainieren der fünf Sprachen ist das System in der Lage diese fünf Sprachen zu erkennen. Die Erweiterung um eine Sprache ist trivial. Kommt eine weitere Sprache hinzu, wird lediglich eine neue Softmax-Ebene an das vorhandene Netzwerk angefügt und trainiert [1].

Ein Vergleich eines monolingualen Deep Neural Networks und eines multilingualen Deep Neural Networks ist in Tabelle (...) aufgeführt. Das monolinguale Netzwerk wurde hierbei nur mit der entsprechenden Sprache 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].

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 Tabelle (...) liegt und

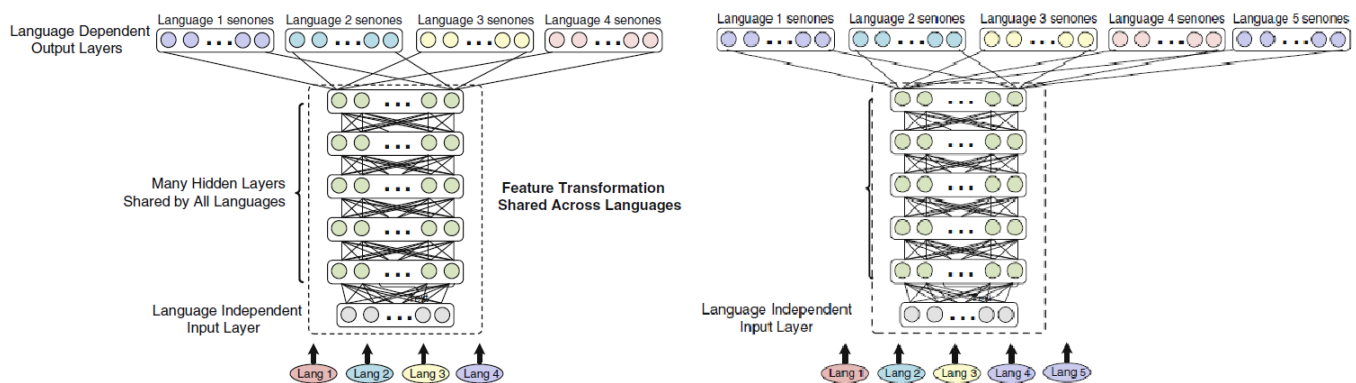


Abbildung 2. Hinzufügen einer neuen Sprache [10]

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

Tabelle 1

RELATIVE WORTFEHLERRATE

Mandarin-Chinesisch, welches weit von den europäischen Sprachen entfernt ist. Die tatsächliche Erkennung der Sprache ist dabei trivial. Die Schallwellen, die beim Sprechen produziert werden, lassen sich über einen elektroakustischen Wandler (Mikrofon) in ein elektrisches Signal umwandeln. Dieses elektrische Tonsignal wird daraufhin in Zahlen bzw. Bits konvertiert (sampling) und über Vorverarbeitung entsprechend aufbereitet, um es in ein neuronales Netz zu speisen [4]. Beim genauen Vorhersagen des gesprochenen kommt die Sprachidentifikation ins Spiel, durch welche Wörter ausgeschlossen werden, die ebenfalls in Frage kommen, allerdings zum Wortschatz einer anderen Sprache gehören. Hat ein System z.B. die deutsche Sprache erkannt, wird für das Wort „Hallo“ immer „Hallo“ vorhergesagt, da dies naheliegender ist. 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 Spracherkennungsgenauigkeit 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 dieser Netzwerke. Diese Form von neuronalen Netzen unterscheidet sich grundlegend von dem Feed-Forward-DNN, 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 RNN deutlich 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 Standard-Algorithmus 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 eine Folge von Iterationen. Der Input ist in obiger Darstellung x_s bezeichnet den Schritt und E den Hidden State, welcher sich beim Eingeben des Inputs ergibt. Ein Recurrent Network gibt somit nicht nur den Input an die nächste Iteration, sondern Input sowie den resultierenden Zustand E . Somit beeinflussen die vorhergehenden Schritte die folgenden. Dies führt zu einem Problem -dem Verschwinden von Information bzw. dem Vanishing Gradient Problem, welches sich dadurch ergibt, dass RNNs nicht in der Lage sind, auf Informationen zurückzugreifen, die weit in der Vergangenheit liegen. Wenn eine Datensequenz lange ist und das System versucht die gesagten Worte vorherzusagen, kann es sein, dass der Kontext bereits vergessen wurde und eine inkorrekte Vorhersage stattfindet. Somit kommt eine erweiterte Form des RNNs zum Einsatz. Dieses wird Long-Short-Term-Memory (LSTM) genannt und erzielt enorm gute Ergebnis-

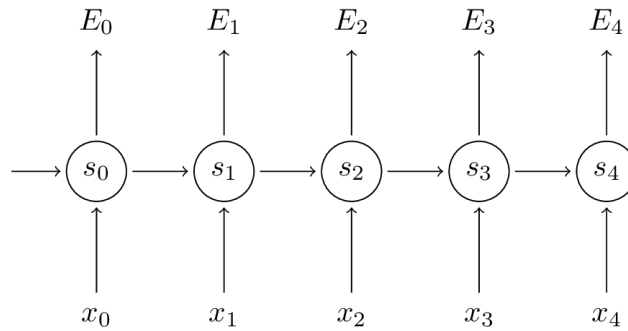


Abbildung 3. Modell des Recurrent Neural Network [10]

se in automatischen Spracherkennungssystemen [1][3]. Diese Netzwerke sind somit in der Lage anhand des Kontextes zukünftige Wörter vorherzusagen und so ihre Genauigkeit zu erhöhen. Auch mit verrauschten Aufnahmen oder schlechteren Bedingungen beim Aufnehmen des Gesprochenen kann diese Form von Netzwerken bessere Ergebnisse erzielen. Aufgrund dessen wurden LSTM-Netzwerke entwickelt, die zur Lösung des Problems beitragen. Dabei werden Recurrent Neural Networks mit einer Speicherstruktur erweitert, was zur namensgebenden Lang-Kurzzeit-Speicherung führt. Diese erlauben die Erkennung zeitlich ausgedehnter Muster und das Erkennen von Zusammenhängen von zeitlich getrennten Ereignissen. Somit eignen sich die Netzwerke um Zeitreihen zu verarbeiten und vorherzusagen. Sogar, wenn zwischen wichtigen Ereignissen sehr lange Verzögerungen liegen, die eine unbekannte Länge aufweisen. Auch bei der Erlernung geräuschverzierter und hallender Sprachmerkmale kann dieses Modell genutzt werden [1]. Die grundsätzliche Idee dabei ist es über elementweise Multiplikationen den Informationsfluss in dem Netzwerk zu steuern. Es kann als komplexe und intelligente Netzwerkeinheit betrachtet werden, welche Informationen über einen langen Zeitraum speichern kann. Dies wird durch die Gating-Struktur erreicht, die bestimmt, wann die Eingabe signifikant genug ist, um sich daran zu erinnern, wann sie sich die Information weiter merken oder vergessen sollte und wann sie die Information ausgeben sollte. Dies geschieht über verschiedene Gates innerhalb einer LSTM-Zelle (...). Ein Gate 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 Spracherkennung deutlich präziser wird. 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 einen höhere Genauigkeiten erreicht werden und zum anderen 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 besten Erkennungsleistung der neuen Sprache führen. Dabei müssen Beziehungen zwischen den Sprachen erkannt werden. Problematisch ist auch, dass gleiche Phoneme je nach Sprecher, Sprache etc. variieren, was dazu führt, dass Phoneme nur im Kontext betrachtet werden (Triphone).

VII. TRAININGSVORGANG

A. Trainingsvorgang

Der Trainingsvorgang basiert auf ein vollständig verbundenes mehrschichtiges Deep Learning-Netzwerk. Dieses Netzwerk aus Neuronen besteht aus drei Schichten:

Input-Schicht
Hidden-Schicht
Output-Schicht

Die Input-Schicht stellt dabei die Eingangsdaten dar, welche als Trainingsmaterial für den Trainingsvorgang verwendet werden. Bei diesen Daten handelt es sich um Sprachaufnahmen. Bei Bedarf können diese Sprachaufnahmen dementsprechend vor-verarbeitet werden, wie zum Beispiel durch Einsatz von Filtern. Anschließend können die Daten in die Netztopologie beschriftet eingespeist werden. Eine Vorklassifizierung der Sprache führt zu erhöhten Erkennungsrate [11]. In der Hidden-Schicht geschieht das eigentlich Training, welches normalerweise durch die Sigmoid-Funktion aktiviert wird [11]. Die Schichten sind durch Kanten miteinander verbunden. Die Hier wird das Netz beginnend von der Input-Schicht bis Output-Schicht vollständig durchlaufen.

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

Sobald die Output-Schicht erreicht ist, wird das Netz rückwärts durchlaufen. Dieses Verfahren wird auch Backpropagation-Algorithmus genannt und wird benötigt, um fehlerhafte Kantengewichte herauszufinden und anzupassen. Die Kantengewichte des Netzes werden anfangs mit null initialisiert, sodass die Kantengewichtungen korrigiert werden müssen. Die Ableitung der Sigmoid-Funktion kommt bei der Korrekturberechnung zum Einsatz V-A. Diese Funktion ist für eine geringe

Datenmenge geeignet. Bei größeren Datenmengen entsteht ein Nachteil, welches sich auf die Wissensausprägung des Netzes auswirkt. Beim Rückwärts durchlaufen entsteht ein Wissensverlust [11]. Dieser Verlust wird durch das Maxima der Ableitung der verwendeten Funktion repräsentiert. Dieser kann bis zu 25 % betragen. Der entstehende Verlust würde die Klassifizierungsrate des Trainingsmodells reduzieren, welches in Abbildung 4 dargestellt ist [12].

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

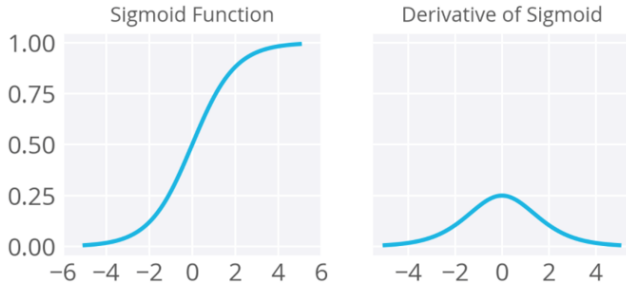


Abbildung 4. Darstellung der Sigmoid-Funktion und dessen Ableitung [12]

Anstelle der Sigmoid-Funktion wird in den modernsten Deep-Learning-Netzen Rectified linear Units (*ReLU*s) verwendet. Diese Funktion ist dem menschlichen Neuron am ähnlichsten und bringt zudem eine erhöhte Verarbeitungsgeschwindigkeit mit sich [13]. Frameworks wie TensorFlow und TFLearn stellen diese Funktionalität bereits standardmäßig zur Verfügung, sodass diese Funktion nicht selber implementiert werden muss.

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

$$x_j = b_j + \sum x_{ij} * y_j \quad (1)$$

Als Nächstes folgt die Output-Schicht, welches die Eingangsdaten dann zu den Klassen (Vorhersagen) zuordnet. Diese Schicht ist als Softlayer konfiguriert, welches die Klassen in eine eindimensionale Matrix kategorisiert. Eine Klasse steht für eine Sprache, die gelernt werden soll. Dabei ist die Matrix in dem Zahlenintervall $[0, 1]$ normalisiert. Die endgültige Sprachidentifikation geschieht über normalisierte Werte, welches in Abbildung 5 dargestellt wird. Die Werte können in Wahrscheinlichkeiten ausgedrückt werden, in dem diese mit dem Faktor 100 multipliziert werden [12]. Die



Abbildung 5. Klassenzuordnung über Wahrscheinlichkeiten in der Softmax-Konfiguration [12]

Vorhersagen der Output-Schicht geschieht durch die Funktion $p(j)$. Dabei steht der Index l für die jeweilige Klasse, also die Sprache, die gelernt werden soll.

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

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

$$C := \sum_l t_j * \log(p_j)$$

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 (label) abweicht [14]. Bei t_j handelt es sich um die Klasse, für die der Verlust berechnet wird [10].

B. Netztopologie

Die Netztopologie beschreibt die Infrastruktur des Netzes. Die Auswahl der Topologie bestimmt die Qualität des Trainingsvorgangs, d. h. die Anzahl der Neuronen. Eine zu geringe Anzahl führt zu einem niedrigen Klassifizierungsrate. Eine zu hohe Anzahl würde zu überhöhten Trainingsdauer führen. Aufgrund dessen fallen Topologien von Ansatz zu Ansatz unterschiedlich aus, welche unterschiedliche Spracherkennungsergebnisse liefern. In dieser Arbeit wird der Topologievorschlag von Gonzales et al. betrachtet. Für die Eingangsdaten werden 40 Filterbanken verwendet. Diese werden benötigt, um die Daten sampeln zu können. Als Input-Schicht wird werden 26 Knoten eingesetzt. Um unerwünschte Latenzzeiten der Frames zu vermeiden wird ein asymmetrischer Kontext verwendet. Die Hidden-Schicht beträgt vier Ebenen mit einer Gesamtzahl von 2560 Knoten. 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.

C. Verbesserung des Trainingsverfahrens durch Multitasking learning (MTL)

Bei maschinelles Lernen wird der Fokus gesetzt, bestimmte Metriken, wie beispielsweise Klassifizierungsgenauigkeit und Trainingsdauer, zu optimieren. Daraufhin wird das Modell soweit optimiert, bis die Leistung des Modells nicht mehr gesteigert werden kann [15]. 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 Trainingsverfahren 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 [16]. Ein Beispiel hierfür ist ein Spamfilter. Der Schlüssel zur erfolgreichen Anwendung von MTL besteht darin, dass die Aufgaben miteinander verknüpft

werden können. Dies bedeutet nicht, dass die Aufgaben ähnlich sein müssen. Stattdessen bedeutet es, dass Aufgaben auf verschiedene Ebenen abstrahiert und geteilt werden. Wenn die Lernaufgaben tatsächlich ähnlich sind, können sie gemeinsam gelernt werden. Dabei kann das Wissen zwischen Aufgaben auf andere Lernaufgaben übertragen werden, welches die Trainingsdauer deutlich verkürzt. MTL ist vor allem dann 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 [15]. 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 6 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 [15] [16].

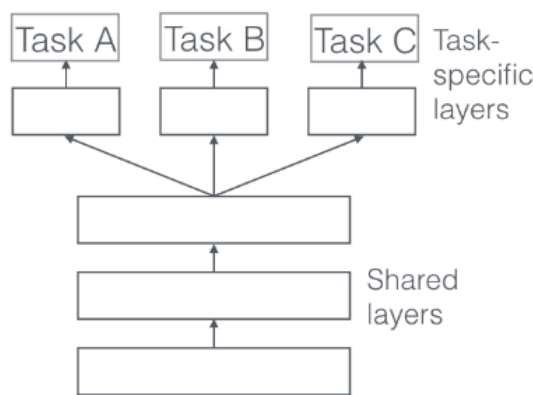


Abbildung 6. Hard parameter sharing auf die Hidden-Schicht angewendet [12].

VIII. DISKUSSION UND AUSBLICK

Die menschliche Sprache ist der natürlichste Weg etwas zu kommunizieren, so ist es nicht wunderlich, dass das Interesse an dem Deep Learning-Ansatz für Spracherkennung und damit verbundene 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 können. Vielmehr hängt deren Zustand und Aktivierung von den Aktivitäten anderer Neuronen ab. Vorhergehenden Ereignissen beeinflussen den Zustand. Durch die dynamische Rekursion 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 sehr hilfreich und da die menschliche Sprache lediglich eine Sequenz von Tönen ist, eignet sich diese Form von Netzwerken ideal. Somit sind diese Netzwerke robust gegenüber Störgeräuschen. Es ging 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 nicht das Hauptsächliche Problem. Die Autoren sind der Meinung, dass eine natürliche Interaktion mit einem Spracherkennungssystem schwierig bleibt, solange das System keine Kenntnisse über die Welt 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 eine 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 [2] 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] R. Harris, *Voice Interaction Design: Crafting the New Conversational Speech Systems*, ser. Morgan Kaufmann series in interactive technologies. Elsevier Science, 2004. [Online]. Available: <https://books.google.de/books?id=92ISybAfXagC>
- [2] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.
- [3] 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>

- [4] 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
- [5] 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>
- [6] 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](http://dx.doi.org/10.1016/S0167-6393(00)00099-6)
- [7] 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.
- [8] T. Niesler and D. Willett, "Language identification and multilingual speech recognition using discriminatively trained acoustic models," 2006.
- [9] 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.
- [10] 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.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [12] 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>
- [13] 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.
- [14] 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
- [15] 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/>
- [16] 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.