Institut für Logik, Komplexität und Deduktionssysteme
der Universität Karlsruhe
Lehrstuhl Prof. A. Waibel

Sprachenidentifikation mit neuronalen Netzen

Studienarbeit
von

Hagen Soltau
(soltau@ira.uka.de)
Betreuerin: Tanja Schultz

Karlsruhe, den 30. August 1999

Zusammenfassung

Es werden Experimente vorgestellt, mit neuronalen Netzen Sprachen zu identifizieren. Dabei werden für die einzelnen Sprachen Spracherkennungssysteme genutzt. Eingabe für die neuronalen Netze bilden die Ausgaben der Spracherkennungssysteme. Mittels eines ML-Klassifikators werden Maßnahmen zur Erhöhung der Trenngüte untersucht. Ein weiterer Schwerpunkt der Experimente liegt in der Merkmalstransformation. Es zeigte sich, daß die Eliminierung von probleminvarianten Eigenschaften deutlich zu einer Verbesserung führt. Das Hinzufügen von Informationen (Ausgabe eines weiteren Spracherkenners) führte ebenfalls zu Verbesserungen. Mit weiteren Experimenten wurde versucht, ein optimales Lernverhalten zu erreichen. Die Einstellung der Schrittweite und des Momentums erwies sich dabei als kritisch. Die Experimente zeigten, daß neuronale Netze mit nur einer verdeckten Schicht ein stabileres Lernverhalten als Netze mit mehreren verdeckten Schichten haben.
Die multilinguale Datenbasis SST (Spontaneous Scheduling Task) wurde zum Trainieren und Testen der Netze verwendet. Sie enthält Dialoge für Terminabsprachen. Bei der Unterscheidung zwischen Deutsch, Englisch, Spanisch und Japanisch wurde eine Identifikationsleistung von 85,4% erreicht.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einführung

Ziel von Spracherkennungssystemen ist es, gesprochene Äußerungen zu erkennen. Dabei werden zunächst die akustischen Sprachsignale (Schallwellen) in elektromagnetische Wellen umgewandelt und anschließend digitalisiert. Die Aufgabe der Vorverarbeitung ist es, aus den digitalisierten Sprachsignalen die wesentlichen Eigenschaften zu extrahieren. Die eigentliche Erkennung basiert auf den in der Sprachvorverarbeitung ermittelten wesentlichen Eigenschaften des Signals.
Bei der Anwendung von multilingualen Spracherkennungssystemen muß die zu erkennende Sprache, d.h die Sprache in welcher gesprochen wurde, identifiziert werden. Anwendungsgebiete von Sprachenidentifikationssystemen sind z.B. Sprachübersetzungsysteme (JANUS) oder auch die Weiterleitung von telefonischen Notrufen (Language Line von AT&T).
Das Gebiet der automatischen Sprachenidentifikation (Language Identification, LID) beschäftigt sich mit diesem Problem. Dabei stützt man sich auf die Erkenntnisse, die auf dem Gebiet der Spracherkennung (Speech Recognition, SR) gesammelt wurden. Ähnlich der Spracherkennung kann man das Problem in zwei Teilprobleme aufspalten:

- Vorverarbeitung des Sprachsignals
- Klassifikation der Sprache

Innerhalb dieser Studienarbeit kamen neuronale Netze als Klassifikator zum Einsatz. Gegenstand dieser Studienarbeit ist die Transformation des Eingaberaumes für das neuronale Netz und der Vergleich verschiedener Netztopologien.
Zum Trainieren und Testen wurde eine mehrsprachige Datenbasis für Dialoge bei Terminabsprachen genutzt (Spontaneous Scheduling Task, SST). Verwendet wurden englische, deutsche, spanische und japanische Dialoge.

2 Grundlagen

Dieses Kapitel beinhaltet die Grundlagen für die nachfolgenden Kapitel. Zunächst wird auf das Spracherkennungsproblem eingegangen um später den Zusammenhang zu dem Sprachenidentfikationsproblem aufzuzeigen. Anschließend werden wichtige Begriffe auf dem Gebiet der neuronalen Netze erklärt.

2.1 Spracherkennung

Wahrscheinlichkeitstheoretisch läßt sich das Erkennungsproblem folgendermaßen formulieren. Sei A das akustische (vorverarbeitete) Signal. Gesucht ist die Wortsequenz $\hat{W} = \hat{w}_1\hat{w}_2\cdots\hat{w}_n$, so daß

$$P(\hat{W}|A) = \max_W P(W|A) \tag{1}$$

ist. Um $P(W|A)$ zu berechnen, wendet man den Satz von Bayes an.

$$P(W|A) = \frac{P(A|W)\cdot P(W)}{P(A)} \tag{2}$$

Für die Maximum-Bildung ist $P(A)$ unerheblich. Es verbleibt $P(A|W)$ und $P(W)$ zu bestimmen. In der akustischen Modellierung werden Modelle für $P(A|W)$ gesucht, die Sprachmodellierung versucht den Term $P(W)$ zu schätzen.

2.2 Akustische Modellierung

Die Erkennung von Sprache wird meist auf die Erkennung kleinerer Spracheinheiten und anschließender Zusammensetzung reduziert. Dies hat den Vorteil, daß es nur Modelle für die (geringere) Anzahl von kleinen Spracheinheiten geben muß. Es werden folgende Ebenen von Spracheinheiten unterschieden:

- Phonem-Ebene
  Phoneme sind konkrete Realisierungen von Sprachlauten. Sie bilden die kleinsten Einheiten in der akustischen Modellierung. Abhängig von Spracherkennungssystemen werden im Deutschen zwischen 40 und 50 Phoneme unterschieden.

- Silben-Ebene
  Durch Koartikulationseffekte hängen die akustischen Ausprägungen von der Lautnachbarschaft ab. Bei Silben kann ein größerer Kontext betrachtet werden. Nachteil ist jedoch eine große Anzahl von Silben, was die Klassifikationsaufgabe dementsprechend erschwert. Zudem entstehen Schwierigkeiten eine adäquate Trainingsmenge zu finden, die auch seltene Silben in ausreichender Anzahl enthält.

- Wort-Ebene
  Spracherkennung, basierend auf Wortebene, ist nur bei einem kleinen Vokabular (z.B. Ziffernerkennung) sinnvoll. Bei einem größeren Wortschatz geht man häufig von einer unteren Ebene aus und verwendet Wörterbücher, aus denen die Zusammensetzung der Worte aus den kleineren Einheiten hervorgeht.

- Satz-Ebene
  Hier gilt das zur Wort-Ebene Gesagte in noch stärkerer Weise. Bei der Annahme von 100000 Worten und ca. 20 Worten pro Satz im Deutschen hätte man eine Klassifikationsaufgabe von 1 aus 10^{100}, was wohl jede Dimension sprengen würde.

Einen Kompromiß zwischen der Phonem- und Silben-Ebene stellen die Triphone dar. Bei Triphonen handelt es sich um Phoneme mit Kontext in Form eines linken und rechten Phonems.

2.3 Sprachmodellierung

Für $W = w_1 \cdots w_n$ ist

$$P(W) = P(w_1) \cdot P(w_2|w_1) \cdot \prod_{k=3}^{n} P(w_k|w_1 \cdots w_{k-1}). \tag{3}$$

Ziel ist es, den Term $P(w_k|w_1 \cdots w_{k-1})$ zu schätzen. Die Anzahl dieser Terme ist expotentiell zu der Länge der Wortsequenzen. Häufig wird deshalb die Historie in Äquivalenzklassen ϕ zusammengefaßt. Bei Bigram Modellen werden die Äquivalenzklassen durch Begrenzung der Historie auf das letzte Zeichen gebildet. Analog können n-gram Modelle gebildet werden.

$$\begin{aligned} P(w_k|w_1\cdots w_{k-1}) &= P(w_k|\phi(w_1\cdots w_{k-1})) \\ \phi_{bigram}(w_1\cdots w_{k-1}) &= w_{k-1} \end{aligned} \tag{4}$$

2.4 Neuronale Netze

Auf dem Gebiet der Neuronalen Netze gibt es vielfältige Ansätze. Im folgenden werden nur FeedForward-Netze betrachtet. Solche Netze bestehen aus mehreren, linear angeordneten Neuronenschichten. Die Neuronen einer Schicht sind untereinander nicht verbunden. Kein Neuron ist mit sich selbst verbunden. Verbindungen bestehen ausschließlich von jedem Neuron einer Schicht zu jedem Neuron der nächst höheren Schicht. Die Schicht, deren Neuronen keine Eingabeverbindungen von anderen Neuronen haben, wird *Input Layer* genannt. Mit *Output Layer* wird die Schicht bezeichnet, deren Neuronen keine Ausgabeverbindungen zu anderen Neuronen haben. Die dazwischenliegenden Schichten werden *Hidden Layer* genannt.
Die Verbindungsgewichte werden mit *Backpropagation* eingestellt. Backpropagation ist ein Verfahren des überwachten Lernens, bei dem Muster (x^μ, y^μ) gelernt werden, wobei x^μ das Eingabemuster und y^μ das Ausgabemuster ist. Ziel ist es, die Verbindungsgewichte so einzustellen, daß bei Anlegen eines Musters x^μ das zugehörige Ausgabemuster y^μ assoziiert wird.
Das Lernen geschieht dabei in zwei Phasen. In der ersten Phase werden die Eingabemuster angelegt und die Neuronenaktivierungen schichtweise (FeedForward) berechnet. In der zweiten Phase wird dann der Fehlergradient für jedes Verbindungsgewicht berechnet und anschließend korrigiert.

2.4.1 FeedForward-Berechnung

Die Netzeingabe net_i des Neurons i ist die mit den Verbindungsgewichten w_{ij} gewichtete Summe der Eingaben s_j der verbundenen Neuronen abzüglich eines Schwellwertes θ_i. Die Wahl der Aktivierungsfunktion f beeinflußt den späteren Gradientenabstieg. Grundforderungen an die Aktivierungsfunktion sind dabei Differenzierbarkeit und Monotonie. Eine häufig und auch hier verwendete Funktion ist die sogenannte logistische Aktivierungsfunktion. Sie hat den Vorteil der einfachen Berechenbarkeit ihrer Ableitung. Die Ableitung ist selbst wieder als Produkt der Funktion darstellbar.

$$\begin{aligned}
& s_i = f(net_i), \text{ wobei } net_i = \textstyle\sum_j (w_{ij} \cdot s_j) - \theta_i \\
& f_{logistic}(x) = \tfrac{1}{1+e^{-x}} \\
& \tfrac{d}{dx} f_{logistic}(x) = \tfrac{e^{-x}}{(1+e^{-x})^2} = f_{logistic}(x) \cdot (1 - f_{logistic}(x))
\end{aligned} \tag{5}$$

2.4.2 Fehler-Rückpropagierung

Bedeutung kommt ebenfalls der Wahl der Fehlerfunktion zu. Für Gradientenabstiegsverfahren sind konvexe Fehlerfunktionen von Vorteil. Im folgenden wird die Summe der quadrierten Fehler (Mean Square Error, MSE) als Fehlermaß E betrachtet. Der Fehlergradient wird dann ausgehend von der Ausgabeschicht zurückverfolgt.

$$\mathrm{E} = \frac{1}{2} \sum_{\mu} \|s^{\mu} - y^{\mu}\|^2 = \frac{1}{2} \sum_{\mu} \sum (s_i^{\mu} - y^{\mu})^2 \tag{6}$$

Bei der Berechnung des Fehlergradienten für die Neuronenaktivierung werden zwei Fälle unterschieden. Der Gradient für Ausgabeneuronen kann direkt durch die Ableitung der Fehlerfunktion E berechnet werden. Bei den Neuronen der unteren Schichten findet die Kettenregel Anwendung.

$$\begin{aligned}
& \tfrac{\partial E}{\partial s_i^{\mu}} = s_i^{\mu} - y^{\mu} \qquad , i \in Ausgabeneuron \\
& \tfrac{\partial E}{\partial s_i^{\mu}} = \textstyle\sum_j \tfrac{\partial E}{\partial net_j^{\mu}} \tfrac{\partial net_j^{\mu}}{\partial s_i^{\mu}} = \textstyle\sum_j \tfrac{\partial E}{\partial net_j^{\mu}} \cdot w_{ij} \qquad , i \notin Ausgabeneuron \\
& \text{und } \tfrac{\partial E}{\partial net_j^{\mu}} = \tfrac{\partial E}{\partial s_j^{\mu}} \tfrac{\partial s_j^{\mu}}{\partial net_j^{\mu}} = \tfrac{\partial E}{\partial s_j^{\mu}} \cdot s_j^{\mu} \cdot (1 - s_j^{\mu})
\end{aligned} \tag{7}$$

Der Fehlergradient für die Gewichte läßt sich durch die Gradienten der Neuronenaktivierung mittels der Kettenregel ausdrücken.

$$\frac{\partial E}{\partial w_{ij}} = \sum_{\mu} \frac{\partial E}{\partial net_j^{\mu}} \frac{\partial net_j^{\mu}}{\partial w_{ij}} = \sum_{\mu} \frac{\partial E}{\partial net_j^{\mu}} \cdot s_i^{\mu} \tag{8}$$

2.4.3 Gewichtsänderung

Die Gewichtsänderungen erfolgen nun so, daß die Gewichte in Richtung des negativen Fehlergradienten verschoben werden. Die Stärke der Verschiebung wird durch die Schrittweite $\triangle$ angegeben. Zusätzlich kann ein Trägheitsmoment α verwendet werden, um die Tendenz einer einmal eingeschlagenen Richtung zu verstärken. So können widersprüchliche Muster, die Zick-Zack-Bewegungen verursachen, behandelt werden. Insgesamt sieht die Korrektur also so aus:

$$w_{ij}^{t+1} = -\triangle \cdot (1-\alpha) \cdot w_{ij}^{t} \cdot \frac{\partial E}{\partial w_{ij}^{t}} + \alpha \cdot w_{ij}^{t} \tag{9}$$

Ein Problem von Gradientenabstiegsverfahren sind flachen Täler des Fehlergebirges. Ist der Fehlergradient relativ klein, so fällt die Gewichtskorrektur relativ schwach aus. Dem läßt sich begegnen, indem die Schrittweite $\triangle$ variabel gestaltet wird.

2.4.4 Musterlernen vs. Epochenlernen

Die bisher vorgestellte Rückpropagierung entspricht einem Epochenlernen. Zunächst werden die Fehler für alle Muster summiert und dann anschließend in einem Schritt die Gewichtskorrekturen durchgeführt. Bei Musterlernen [2] wird nicht erst gewartet bis für alle Muster der Fehlergradient berechnet wurde, sondern die Gewichtskorrektur erfolgt für jedes Muster sofort. Die Korrekturregel sieht dann folgendermaßen aus:

$$w_{ij}^{t+1} = -\triangle \cdot w_{ij}^{t} \cdot \frac{\partial E^{\mu}}{\partial w_{ij}^{t}} + \alpha \cdot w_{ij}^{t} \qquad \text{,wobei } \frac{\partial E^{\mu}}{\partial w_{ij}^{t}} = \frac{\partial E}{\partial net_j^{\mu}} \frac{\partial net_j^{\mu}}{\partial w_{ij}} \tag{10}$$

Diese Variante ist kein echtes Gradientenabstiegsverfahren mehr, da der Fehlergradient nicht mehr der Gradient der Gesamtfehlerfunktion E ist. Wenn der Fehlergradient aber mit dem Gradienten der Gesamtfehlerfunktion im Durchschnitt übereinstimmt, dann läßt sich so ein Beschleunigen des Lernvorgangs erreichen. Es ist offensichtlich, daß der Erfolg bei Musterlernen von der Reihenfolge der präsentierten Muster abhängt. Werden die Muster zufällig aus der Datenmenge gezogen, so wird der durchschnittliche Fehlergradient mit dem Gesamtfehlergradient übereinstimmen. Es gibt allerdings auch Probleme, bei dem ein Musterlernen mit geschickter Wahl der Reihenfolge der Muster erfolgreicher ist.

3 Ein System zur Sprachenidentifikation

Heutige Systeme zur Sprachenidentifikation bauen häufig auf Spracherkennungssysteme auf. Das hier betrachtete Sprachenidentifikationsystem verwendet für jede zu identifizierende Sprache einen eigenen Spracherkenner. Dieses System wird für das multilinguale Sprachübersetzungsystem JANUS eingesetzt. Es stehen deshalb die Spracherkennungssysteme bereits zur Verfügung. Es wird von der Annahme ausgegangen, daß der Erkenner, in dessen Sprache die Äußerung ist, die höchste Wahrscheinlichkeit (bzw. die kleinste Distanz in JANUS) liefert. Ein Klassifikator vergleicht die Ausgaben der Spracherkenner für die Äußerung und entscheidet sich dann für die Sprache, dessen Erkenner die höchste Wahrscheinlichkeit liefert. Das LID-Problem läßt sich dann wahrscheinlichkeitstheoretisch folgendermaßen formulieren. Sei A das akustische (vorverarbeitete) Signal. Gesucht ist die Sprache $\hat{L}$, so daß

$$P(\hat{L}|A) = \max_{L} P(L|A) \tag{11}$$

gilt. Der Term $P(L|A)$ kann so formuliert werden, daß der Bezug zum Spracherkennungsproblem deutlich wird.

$$P(L|A) = \sum_{W \in Vok(L)} P(L, W|A) = \sum_{W \in Vok(L)} P(L|W, A) \cdot P(W|A) \tag{12}$$

Obiger Ausdruck läßt sich weiter vereinfachen, indem statt alle Wortsequenzen W nur die wahrscheinlichste Hypothese $\hat{W}$ betrachtet wird. Dann entfällt die aufwendige Summation. Näheres dazu ist in [1] zu finden.

$$\hat{L} = \operatorname*{argmax}_{L} P(L, \hat{W}|A) \tag{13}$$

Eine weitere Umformung ergibt:

$$\hat{L} = \operatorname*{argmax}_{L} P(A|L, \hat{W}) \cdot P(\hat{W}|L) \cdot P(L) \tag{14}$$

Mit dem akustischen Modell kann $P(A|L, \hat{W})$ berechnet werden, das Sprachmodell liefert $P(\hat{W}|L)$. $P(L)$ gibt die a-priori Wahrscheinlichkeit für die Sprache L an.

Ein Problem dieses Ansatzes ist die Nichtvergleichbarkeit der von Spracherkennern berechneten Wahrscheinlichkeiten. Falls ein Erkenner unabhängig

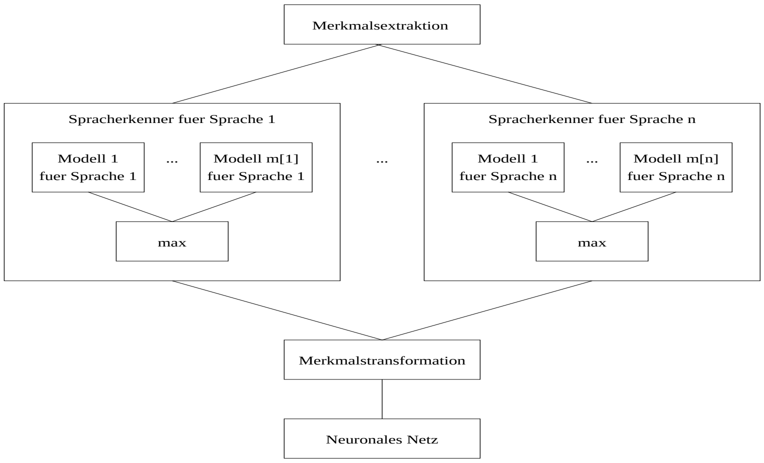

Abbildung 1: LID-System mit neuronalem Netz

von der Akustik geringere Ausgaben als die anderen Erkenner liefert, so wird sich der ML-Klassifikator nie für die Sprache dieses Erkenners entscheiden. Es ist also in jedem Fall eine weitere Verarbeitung der von den Erkennern gelieferten Werte erforderlich, bevor die Werte in den ML-Klassifikator eingehen können. Dies zeigt auch [4].

Der Ansatz wurde nun dahingehend modifiziert, daß statt des ML-Klassifikators ein neuronales Netz verwendet wird. Die Ausgaben der Erkenner werden aber nicht direkt dem Netz übergeben, sondern es erfolgt erst eine Merkmalstransformation. Vorteil dieses Ansatzes ist es, daß die Normierung selbständig gelernt wird. Zusätzlich erhält man ein eher diskriminatives Verhalten durch das neuronale Netz.

4 Experimente

Ausgehend von der im vergangenen Kapitel beschriebenen Systemstruktur des hier betrachteten LID-Systems (siehe Abb. 1) werden verschiedene Netztopologien und Eingabetransformationen verglichen. Zur Messung der Identifikationsleistung wird die mehrsprachige Datenbasis SST herangezogen. Sie wird näher im ersten Abschnitt beschrieben. Im zweiten Abschnitt werden verschiedene Verfahren zur Transformation des Eingaberaumes des neuronalen Netzes verglichen. Im letzten Abschnitt werden verschiedene Netztopologien untersucht.

Die Notation der Daten entspricht der Notation des Abschnitts *neuronale Netze*. Das Eingabemuster x^μ ist ein Vektor der Dimension der Anzahl der Sprachen. Die Reihenfolge ist Deutsch, Englisch, Spanisch und Japanisch. Mit x_i^μ wird auf die i-te Komponente des Vektors zugegriffen und enthält den Score des i-ten Spracherkenners für die μ-te Äußerung. Das Ausgabemuster y^μ ist ein Vektor der gleichen Dimension. Er hat folgende Form:

$$y_i^\mu = \begin{cases} 1 & : \text{ Äußerung } \mu \text{ in Sprache } i \\ 0 & : \text{ sonst} \end{cases}$$

4.1 Die mehrsprachige Datenbasis SST

Die mehrsprachige Datenbasis SST wurde an der Universität Karlsruhe, der Carnegie Mellon University (USA) und ATR International (Japan) aufgenommen. Ziel war es, eine Datenbasis für spontansprachliche Terminabsprachedialoge zu schaffen. Die Gespräche werden dabei von muttersprachlichen Sprechern abgewickelt. Ein Knopfdruckverfahren vergibt das Rederecht, sodaß jeweils nur ein Sprecher sprechen kann. Umfang der Dialoge sind durchschnittlich 10 bis 15 Äußerungen. Die Datenbasis enthält Dialoge in Deutsch, Englisch, Spanisch und Japanisch (Tabelle 1).

Für diese Sprachen stehen jeweils ein phonem- und ein wortbasierter Erkenner zur Verfügung. Die Erkennungsleistungen sind in Tabelle 2 wiedergegeben. Die Identifikationsleistung kann von den Erkennungsleistungen der Spracherkenner abhängen. Zu beachten ist allerdings, daß die Erkennungsleistungen in der Tabelle 2 nicht unbedingt vergleichbar sind. Die Erkennungsleistung der phonembasierten Erkenner sind Phonemerkennungsleistungen. In der rechten Hälfte der Tabelle 2 sind Worterkennungsleistungen

Sprache	Äußerungen	Stunden
Deutsch	12292	30.5
Englisch	7644	6.9
Spanisch	5730	10.7
Japanisch	3311	8.0

Tabelle 1: Die spontansprachliche Terminabsprachedatenbasis SST

Sprache	PmitPT	WmitLM
Deutsch	56.1%	69.6%
Englisch	53.1%	69.0%
Spanisch	52.0%	69.4%
Japanisch	65.5%	70.0%

Tabelle 2: Erkennungsleistungen auf SST

angegeben. In den phonembasierten Erkennern *PmitPT* ist Zusatzwissen in Form von Phonem-Trigram Modellen integriert. Die wortbasierten Erkenner *WmitLM* enthalten Wort-Trigram Sprachmodelle.

Für die Experimente standen unterschiedliche Trainingsumgebungen zur Verfügung. Die Trainingsumgebung TB1 enthält nur Äußerungen in Deutsch, Englisch und Japanisch. Dafür stehen aber insgesamt 1002 Äußerungen zur Verfügung, d.h. 334 pro Sprache. Diese Umgebung wurde später auf alle vier Sprachen erweitert. In der Trainingsgumgebung TB2 gibt es für jede Sprache 314 Äußerungen. Von diesen Daten wurden 70% zum Trainieren und 20% zum Testen verwendet. Die restlichen 10% wurden für eine Validierungsmenge verwendet. Die in den nächsten Abschnitten gezeigten Lernkurven beziehen sich immer auf die Trainingsdaten. Die angegebenen Identifikationsleistungen beruhen aber natürlich auf Testdaten. Dies gilt auch für die im Abschnitt *Ergebnisse mit dem ML-Klassifikator* angegebenen Ergebnisse.

Trainingsumgebung	Sprachen	Größe	Größe pro Sprache
TB1	D,E,J	1002	334
TB2	D,E,S,J	1256	314

Tabelle 3: Trainingsumgebungen für ML-Klassifikator und neuronale Netze

4.2 Transformation der Eingabe

4.2.1 Ergebnisse mit dem ML-Klassifikator

Zunächst wurden Experimente mit dem ML-Klassifikator durchgeführt, um die Trennbarkeit der Daten einzuschätzen. Es werden verschiedene Normierungen verglichen. Es wurde die Trainingsumgebung TB2 (Tabelle 3) vewendet. Die folgenden Tabellen bestehen jeweils aus zwei Hälften. Die linke Hälfte enthält die Ergebnisse für die phonembasierten Erkenner, in der rechten Hälfte wurden die wortbasierten Erkenner verwendet. In der i-ten Zeile und j-ten Spalte steht die Anzahl der Äußerungen der Sprache i, die für Äußerungen der Sprache j gehalten wurden.

	D	E	S	J	D	E	S	J
D	55	0	7	0	0	0	62	0
E	27	0	22	13	0	0	61	1
S	37	0	25	0	0	0	62	0
J	22	0	40	0	0	0	62	0
	PmitPT				WmitLM			

Tabelle 4: ML-Klassifikator ohne Normierung

Die englischen und japanischen Äußerungen werden bei Verwendung der phonembasierten Erkenner nie identifiziert (Tabelle 4, 2. und 4. Spalte der linken Tabellenhälfte). Bei Verwendung der wortbasierten Erkenner produziert der spanische Erkenner Ausgaben in einem Wertebereich, der deutlich die Wertebereiche der anderen Erkenner dominiert (3. Spalte der rechten Tabellenhälfte). Somit wird sich nur für Spanisch entschieden.

Eine Normierung des Wertebereichs scheint also erforderlich zu sein. Der Normierungsfaktor ist die Summe aller Ausgaben eines Erkenners. Interessant ist der Einfluß des Summierungsbereichs. Man kann entweder über die Ausgaben aller Äußerungen summieren oder es werden nur die Äußerungen betrachtet, die in der Sprache gesprochen wurden, für die der Erkenner gebaut wurden ist. Um diesen Einfluß zu untersuchen, setzt sich der Normierungsfaktor aus den Ausgaben für die Äußerungen aller Sprachen zusammen und im zweiten Durchlauf nur aus den Ausgaben für die Äußerungen der Sprache des Erkenners. Die Normierungsfaktoren werden ausschließlich aus den Trainingsdaten gewonnen. Die so ermittelten Faktoren werden auf die

	D	E	S	J	D	E	S	J
D	42	9	11	0	50	11	0	1
E	0	19	17	26	1	59	2	0
S	18	16	28	0	2	25	27	8
J	3	6	46	7	0	9	5	48
	PmitPT				WmitLM			

Tabelle 5: ML-Klassifikator mit Wertebereich-Normierung über alle Sprachen

Testdaten übertragen. Somit sind die Angaben zur Identifikationsleistung also korrekt.

$$\tilde{x}_i^\mu = \frac{x_i^\mu}{\sum_\nu x_i^\nu} \tag{15}$$

$$\tilde{x}_i^\mu = \frac{x_i^\mu}{\sum_{\nu \in L(i)} x_i^\nu} \tag{16}$$

Bei Anwendung der Wertebereich-Normierung ist eine starke Ausprägung der Hauptdiagonalen in der Tabelle 5 erkennbar. So wurden die meisten Äußerungen auch der richtigen Sprache zugeordnet (bei Verwendung der wortbasierten Erkenner). Die Hauptdiagonale der rechten Tabellenhälfte ist mit 50-59-27-48 besetzt. In jeder Sprache gab es 62 Äußerungen. Es wurde also jeweils ein Großteil richtig klassifiziert.

Tabelle 7 zeigt den Vergleich des Summierungsbereichs. Die Identifikationsleistung bei der Normierung des Wertebereichs der Äußerungen der Erkennersprache ist lediglich knapp über der Zufallswahrscheinlichkeit. Wird aber über die Äußerungen aller Sprachen summiert, so ist die Identifikationsleistung bereits $74,2\%$ im Falle wortbasierter Erkenner. Diese Normierung führte zu einer signifikanten Verbesserung. Die Äußerungen in fremder Sprache sollten also mit berücksichtigt werden. Werden die fremden Äußerungen nicht berücksichtigt, so erhält man nur 29,4%.

Am häufigsten wurde Spanisch mit Englisch bei den wortbasierten Erkennern verwechselt. 25 der spanischen Äußerungen wurden der englischen Sprache zugeordnet. Die Unterscheidung von Deutsch und Englisch bereitet anscheinend weniger Schwierigkeiten, obwohl sie der gleichen Sprachfamilie angehören. Auffallend ist die Verwechselung von Japanisch mit Spanisch bei

	D	E	S	J	D	E	S	J
D	14	0	41	7	1	1	53	7
E	3	5	34	20	0	4	57	1
S	6	0	52	4	0	2	59	1
J	8	0	52	2	0	3	53	6
	PmitPT				WmitLM			

Tabelle 6: ML-Klassifikator mit Mittelwert-Normierung

Verwendung der phonembasierten Erkenner. Fast alle japanischen Äußerungen wurden der spanischen Sprache zugerechnet.

Die Anzahl der Verwechselungen sind bei Verwendung der wortbasierten Erkenner geringer als bei Verwendung der phonembasierten Erkenner. Die unterschiedlichen Wertebereiche der phonembasierten Erkenner können durch diese Normierung nicht korrigiert werden. Die Verwechselungen werden eher durch die unterschiedlichen Wertebereiche der Erkenner verursacht, weniger durch die Ähnlichkeiten von Sprachen einer Sprachfamilie.

Zum Vergleich wurde eine Normierung durchgeführt, die Zissman [6] vorschlug. Dabei wird der Mittelwert aller Scores der Erkenner von dem Score abgezogen. Der Mittelwert bezieht sich auf die Sprache des Erkenners. In Formeln ausgedrückt:

$$\tilde{x}_i^\mu = x_i^\mu - \frac{\sum_\nu x_i^\nu}{\#\text{Pattern}} \tag{17}$$

Es zeigt sich, daß das Normieren durch Subtraktion des Mittelwertes nur eine Verbesserung bei Verwendung der wortbasierten Erkenner bringt (28,2%), bei Verwendung der phonembasierten Erkenner ist die Identifikationsleistung sogar geringer (29,4%). Dies liegt vermutlich an den unterschiedlichen Wertebereichen, die die phonembasierten Erkenner produzieren. Darauf deuten auch die Verwechselungen hin. In der Tabelle 6 sind die Verwechselungen bei Anwendung der Mittelwert-Normierung angegeben. Die Dominanz der spanischen Ausgaben bei Verwendung der wortbasierten Erkenner konnte nicht ausgeglichen werden. Die Wertebereich-Normierung erweist sich im Vergleich zu der Mittelwert-Normierung als besser.

Normierungsart	PmitPT	WmitLM
ohne Normierung	32,2%	25%
Wertebereich über alle Sprachen	38,7%	74,2%
Wertebereich über die gesprochene Sprache	32,6%	29,4%
Subtraktion des Mittelwertes	29,4%	28,2%

Tabelle 7: Vergleich der Identifikationsleistungen bezügl. Normierung

4.2.2 Informationsreduktion

Der Begriff Informationsreduktion bezieht sich in diesem Text auf die dem Netz zur Verfügung stehenden Informationen, nicht aber auf die Wissensquellen des Gesamtsystems.

Die Vorverarbeitung der Daten für neuronale Netze beschäftigt sich mit dem Auffinden von probleminvarianten Eigenschaften der Daten. Die Daten sollen so transformiert werden, daß diese probleminvarianten Eigenschaften eliminiert werden. Im folgenden werden zwei Merkmalsextraktionen vorgestellt und miteinander verglichen.

Die Ausgaben der Spracherkenner hängen nicht nur von den Äußerungen selbst ab, sondern auch von der Länge der Äußerungen. Je umfangreicher die Äußerung, desto so geringer wird die Wahrscheinlichkeit der Äußerung. Die Länge der Äußerung ist durch die Anzahl der Frames der Äußerung gegeben, die die Sprachvorverarbeitung (Kurzzeit-FFT) aus der Äußerung erzeugt. In der JANUS-Vorverarbeitung entspricht ein Frame 10 ms. Die Abhängigkeit von der Länge ist aber für das LID-Problem uninteressant. Es wird deshalb eine Längennormierung durchgeführt, indem die Ausgaben durch die Länge der Äußerung geteilt werden.

$$\tilde{x}_i^\mu = \frac{x_i^\mu}{\text{Anzahl der Frames von } x^\mu} \tag{18}$$

Diese normierten Daten enthalten noch eine weitere probleminvariante Eigenschaft. Die Summe der Ausgaben aller Erkenner einer Äußerung gibt an, wie die Erkenner zusammen die Äußerung bewerten. Wird die Summe auf einen konstanten Wert (= 1) für alle Äußerungen gesetzt, so wird auch diese Information aus den Daten entfernt. Der Betrag des Merkmalvektors wird auf 1 normiert.

$$\tilde{x}_i^\mu = \frac{x_i^\mu}{\sum_j x_j^\mu} \qquad (19)$$

Zum Vergleich der Transformationen wurde ein dreischichtiges Netz mit 20 Neuronen in der verdeckten Schicht trainiert. Zu unterscheiden waren die Sprachen Deutsch, Englisch, Spanisch und Japanisch (Trainingsumgebung TB2). Die Testmenge besteht aus 248 Mustern (62 pro Sprache). Zur Spracherkennung wurden die wortbasierten und die phonembasierten Erkenner verwendet.

Reduktionssart	ohne Reduktion	Länge derÄußerung	Vektorbetrag
ohne Normierung			
PmitPT	69,3%	85,4%	72,6%
WmitLM	33%	51,1%	66,9%
Reduktion + Wertebereichsnormierung			
PmitPT	25%	25%	25%
WmitLM	25%	25%	25%
Reduktion + Wertbereichsnorm. + Vektorbetragsn.			
PmitPT	25%	67,7%	68,9%
WmitLM	25%	81,4%	81,4%

Tabelle 8: Vergleich der Identifikationsleistungen bezügl. Informationsreduktion

Zunächst wurde der Einfluß der Informationsreduktion ohne Anwendung einer der im letzten Kapitel vorgestellten Normierungen untersucht. Es ergeben sich große Unterschiede bezüglich der verwendeten Erkenner. Bei den Ausgaben der phonembasierten Erkenner scheint eine Normierung bezüglich der Länge derÄußerung sinnvoll. Es werden 85,4% erreicht. Bei den wortbasierten Erkennern wird dagegen mit der Vektorbetragsnormierung eine größere Identifikationsleistung erreicht (66,9%) als bei der Normierung der Länge derÄußerung (51,1%).

Wird nun nach der Informationsreduktion zusätzlich eine Wertebereichnormierung durchgeführt, so ist trotz der verbesserten Trenngüte das Netz nicht in der Lage, das Klassifikationsproblem zu lernen. Eine Analyse der Wertebereichsnormierung ergab, daß die Eingabemuster einen zu geringen Betrag haben. Die Größenordnung einer Komponente der Eingabemuster ist etwa 1/1000. Wahrscheinlich versagt bei diesen Bereichen der Gradientenabstieg.

Aus diesem Grunde wurde nach der Wertebereichsnormierung noch eine Vektorbetragsnormierung durchgeführt. Nun ergeben sich auch für die wortbasierten Erkenner eine vergleichbare Identifikationsleistung von 81,4%. Der geringe Unterschied zwischen den beiden Reduktionsarten wird durch die gemeinsame zusätzliche Vektorbetragsnormierung verursacht.

Insgesamt sollte also bei Verwendung der phonembasierten Erkenner eine Normierung bezüglich der Länge derÄußerung, aber keine Wertebereichsnormierung durchgeführt werden. Bei Verwendung der wortbasierten Erkenner ist eine Kombination aus Wertebereichsnormierung und Vektorbetragsnormierung sinnvoll.

4.2.3 Vergrößerung des Eingaberaumes

Im letzten Abschnitt wurde gezeigt, wie Informationsreduktionen zur Erhöhung der Identifikationsleistung führen können. Ziel war es, probleminvariante Informationen zu eliminieren. Im folgenden wird untersucht, inwieweit zusätzliche Informationen genutzt werden können, um eine Verbesserung der Leistung zu erreichen.

Zur Trennung von n Sprachen wurden bisher die Ausgaben der n Spracherkenner genutzt. Es stellt sich die Frage, ob die Ausgabe eines weiteren Spracherkenners zur besseren Trennung der Sprachen beiträgt. Da kein weiterer Spracherkenner zur Verfügung steht, wird das Identifikationsproblem modifiziert. Zu unterscheiden sind nunmehr 3 Sprachen. Zur Trennung stehen aber die Ausgaben aller 4 Spracherkenner zur Verfügung. Es wird wie auch in den letzten Abschnitten die Trainingsumgebung TB2 verwendet. Werden die phonembasierten Erkenner verwendet, so wird aufgrund der Ergebnisse des Abschnitts *Informationsreduktion* lediglich eine Längennormierung aber keine Wertebereich- oder Mittelwertnormierung durchgeführt. Bei Verwendung der wortbasierten Erkenner werden Wertebereichsnormierung und Vektorbetragsnormierung kombiniert.

Die Ergebnisse der Tabelle 9 basieren auf einem dreischichtigen Netz, wobei die verdeckte Schicht 20 Neuronen umfaßt. Die Identifikationsleistung steigt deutlich bei Hinzunahme weiterer Informationen bei Anwendung der wortbasierten Erkenner. Bei der Trennung von Englisch, Spanisch und Japanisch fällt dies am deutlichsten auf. Dort steigt die Leistung von 73,6% auf

Sprachen	#Eingabe=3	#Eingabe=4	#Eingabe=3	#Eingabe=4
D-E-S	80,2%	83,3%	76,8%	82,8%
D-E-J	94,1%	89,8%	90,8%	93,5%
D-S-J	82,8%	89,2%	89,2%	89,3%
E-S-J	72,0%	88,7%	73,6%	79,0%
	PmitPT		WmitLM	

Tabelle 9: Vergleich der Identifikationsleistungen bezügl. Vergrößerung des Eingaberaumes

79,0%. Bemerkenswert ist auch die Abhängigkeit von der Klassifikationsaufgabe. So lassen sich die Sprachen Deutsch, Englisch und Japanisch recht gut trennen (90,8% bzw. 93,5%). Umfaßt das Problem aber die Trennung von Englisch und Spanisch (1.te und 4.te Zeile), so fällt eine deutlich geringere Identifikationsleistung auf. Dort ergeben sich auch die größten Verbesserungen durch Hinzunahme der Ausgabe eines weiteren Erkenners.

Auch bei Verwendung der phonembasierten Erkenner zeigt sich eine Leistungssteigerung. So wird eine Verbesserung von 72,0% auf 88,7% bei der Trennung von Englisch, Spanisch und Japanisch erreicht. Nur in der 2.ten Zeile (Deutsch, Englisch und Japanisch) tritt eine Verschlechterung auf. Die Hinzunahme der Ausgaben des spanischen Erkenners verwirrt das neuronale Netz.

Bei Verwendung der phonem- und wortbasierten Erkenner scheint die Klassifikation von Deutsch, Englisch und Japanisch am leichtesten. Probleme bereiten die Ausgaben des spanischen Erkenners. Insbesondere ist die Kombination von Englisch und Spanisch kritisch. Dies ist auch konsistent zu den Ergebnissen, die mit dem ML-Klassifikator erreicht wurden. Die in Tab. 5 angegebenen Verwechselungen waren bei den spanischen Äußerungen besonders hoch. Im Falle wortbasierter Erkenner wurden 25 der 62 Äußerungen der englischen Sprache zugerechet.

4.3 Netztopologien und Lernparameter

4.3.1 Lernrate und Momentum

Das Momentum ist ein Parameter, um bisherige Trends zu unterstützen. Eine Erhöhung des Momentums kann das Lernverhalten wesentlich beschleunigen.

Ausgehend von wortbasierten Daten werden verschiedene Einstellungen des Momentums getestet. Die Untersuchungen beziehen sich sowohl auf nicht normierten Daten, also auch auf wertebereich-normierten Daten kombiniert mit der Vektorbetragsnormierung. Trainiert wird ein dreischichtiges Netz mit 20 verdeckten Neuronen. Zu unterscheiden sind dabei die Sprachen Englisch, Deutsch und Japanisch. Zum Einsatz kommt die umfangreichere Trainingsumgebung TB1. Zur Erzeugung der Ausgaben der Spracherkennungssysteme wurden nur die wortbasierten Erkenner verwendet.

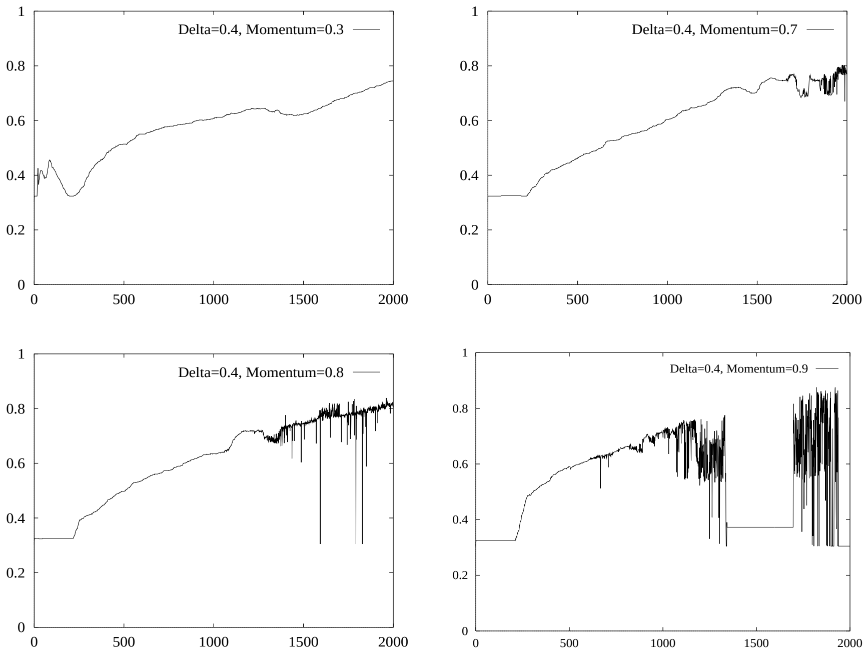

Abbildung 2: Lernkurven bei unterschiedlichen Momentum ohne Wertebereich-Normierung

Die Lernkurven (Abb. 2) sind jeweils von den Trainingsdaten aufgenommen. Wird das Momentum 0,3 gewählt, so erhält man eine relativ stetige Lernkurve. Wird das Momentum höher gesetzt (0,7), so steigt die Lernkurve zwischen der 1000. und 1500. Iteration stärker als bei der Lernkurve mit kleinerem Momentum. Andererseits entstehen Zick-Zack-Bewegungen. Wird das Momentum weiter auf 0,8 vergrößert, so erkauft man sich den stärkeren Anstieg der Kurve mit ausgeprägten Zick-Zack-Bewegungen. Es enstehen aber noch keine lokalen Minima. Bei einem Momentum von 0,9 ändert sich die Richtung des Fehlergradienten fortlaufend. Die Lernkurve besteht fast nur noch aus Zick-Zack- Bewegungen. Fällt das Netz in ein lokales Minimum, so bleibt es dort aufgrund des großen Momentums lange gefangen (etwa 300 Iterationen).

Im folgenden wird nun die Variation des Momentums bei veränderten Daten betrachet. Die Daten sind mit der bereits in Abschnitt *Ergebnisse mit dem ML-Klassifikator* untersuchten Wertebereich-Normierung vorverarbeitet worden.

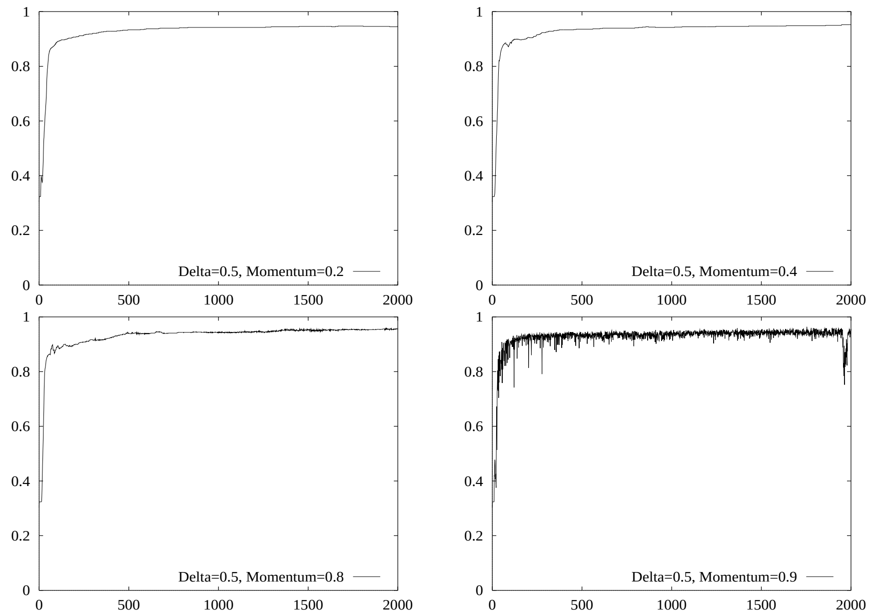

Abbildung 3: Lernkurven bei unterschiedlichen Momentum mit Wertebereich-Normierung

Es scheint, daß das Momentum bei Netzen, die mit wertebereich-normierten Daten trainiert werden, einen geringeren Einfluß hat (vgl. Abb. 3). Selbst bei einem Momentum=0,2 ist die Lernkurve wesentlich steiler (und stetiger) als bei nicht-normierten Daten. Wird das Momentum auf 0,4 bzw. 0,8 weiter erhöht, so steigt die Kurve in den ersten 100 Iterationen nur noch etwas stärker. Erst wenn das Momemtum auf 0,9 gesetzt wird, entstehen kleinere Zick-Zack-Bewegungen. Die Lernkurve geht bereits in den ersten 100 Iterationen steil nach oben. Die Wertebereich-Normierung sollte also nicht nur bei ML-Klassifikatoren angewendet werden, sondern auch bei neuronalen Netzen. Sie erweist sich in jedem Fall als vorteilhaft.

Lernparameter	Δ=0,5
α=0,2	90,4%
α=0,4	94,8%
α=0,8	95,6%
α=0,9	80,5%

Tabelle 10: Vergleich der Identifikationsleistungen bei unterschiedlichen Momentum

Lernparameter	α=0,8	α=0,9
Δ=0,4	94,8%	92,6%
Δ=0,5	95,6%	80,5%
Δ=0,6	87,8%	90,4%
Δ=0,7	80,5%	89,2%

Tabelle 11: Vergleich der Identifikationsleistungen bei unterschiedlicher Schrittweite

In der Tabelle 10 sind die Ergebnisse der mit unterschiedlichen Momentum trainierten Netze für Testdaten angegeben. Mit einem hohem Momentum lassen sich also Verbesserungen der Identifikationsleistung bei wertebereichnormierten Daten erzielen. Dabei fällt der Unterschied am deutlichsten bei dem Übergang von Momentum 0,2 auf 0,4 auf. Die Identifikationsleistung steigt von 90,4% auf 94,8%.

Der Einfluß der Schrittweite auf die Identifikationsleistung ist in Tabelle 11 wiedergegeben. Als Momentum wird 0,8 bzw. 0,9 gewählt. Wird ein Momentum von 0,8 gewählt, so sollte die Schrittweite nicht größer als 0,5 gesetzt werden. Die Identifikationsleistung fällt bei wachsender Schrittweite rapide. So sinkt die Identifikationsleistung auf 80,5% bei einer Schrittweite=0,7. Eine ebenso eindeutige Aussage läßt sich bei einem Momentum=0,9 nicht treffen. Es ist kein Trend erkennbar. Obwohl die Lernkurve (Abb. 4) mit Schrittweite=0,5 relativ gut konvergiert und keine größeren Zick-Zack-Bewegungen aufweist, ist die Identifikationsleistung 80,5%. Wird die Schrittweite weiter erhöht, erreicht auch die Identifikationsleistung wieder einen 'normalen' Bereich. Dies ist um so erstaunlicher, wenn die Lernkurve betrachtet wird. Die Lernkurve mit Schrittweite=0,6 besteht eigentlich nur aus Zick-Zack-Bewegungen. Es ist keine Konvergenz des Lernverhaltens ersicht-

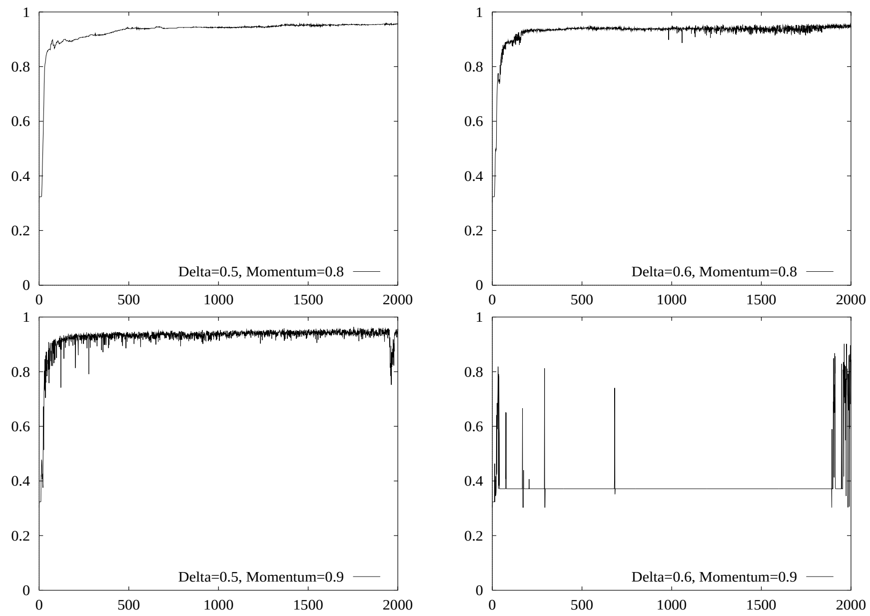

Abbildung 4: Lernkurven bei unterschiedlicher Schrittweite mit Wertebereich-Normierung

lich, die Kurve ist unstetig. Insgesamt erscheint es sinnvoll, die Schrittweite auf 0,5 zu setzen und das Momentum bei 0,8 zu belassen.

4.3.2 Epochenlernen

In diesem Abschnitt wird nun untersucht, inwieweit das bisherige Musterlernen gerechtfertigt war. Allgemein gilt, daß das Epochenlernen stabiler verläuft, dafür aber langsamer ist. Zu beachten ist eine Schrittweiten- und Momentumanpassung. Als Heuristik gilt:

$$\tilde{\Delta} = \frac{\Delta}{\#Muster} \qquad \text{und} \qquad \tilde{\alpha} = \frac{\alpha}{\#Muster} \tag{20}$$

Bei $0,7 \cdot 1002 = 701$ Mustern und den Lernparametern aus dem letzten Abschnitt ergeben sich $\tilde{\Delta} = 0.0007$ und $\tilde{\alpha} = 0.0011$ als vergleichbare Lernparameter. Man beachte die Anzahl der Iterationen in Abb. 5. Es wurde also doppelt soviele Iterationen gelernt. Dabei werden bei Musterlernen als auch

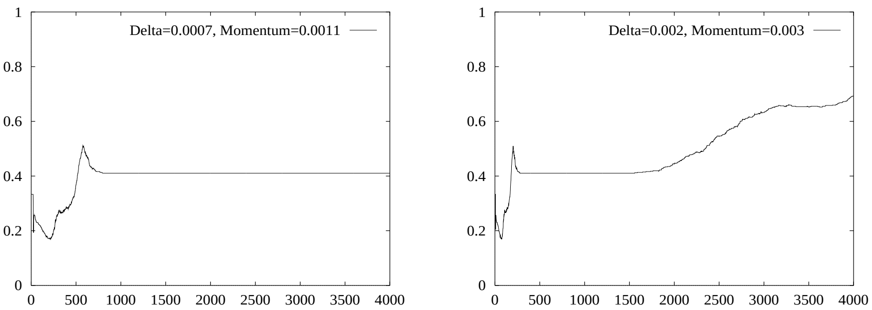

Abbildung 5: Lernkurven bei Epochenlernen

Lernparameter	ID-Leistung
Δ=0,0007 α=0,0011	54,0%
Δ=0,0050 α=0,0005	82,3%
Δ=0,0020 α=0,0030	73,7%

Tabelle 12: Vergleich der Identifikationsleistung bei Epochenlernen

bei Epochenlernen in einer Iteration alle Muster präsentiert. Auch die Identifikationsleistung ist wesentlich schlechter (82,3 %) im Vergleich zu 95,6 % bei Musterlernen. Die Daten wurden mit den gleichen Mitteln vorverarbeitet (Wertebereich-Normierung und VektorBetrag-Normierung). Das Musterlernen ist schneller und erfolgreicher.

4.3.3 Netzgröße

Die bisherigen Experimente bezogen sich auf ein dreischichtiges Netz mit 20 Neuronen in der verdeckten Schicht. In diesem Abschnitt wird nun untersucht, inwieweit eine Veränderung der Netzgröße eine Verbesserung des Lernverhaltens bewirkt. Als Lernparameter werden die im letzten Abschnitt gefundenen Werte (Δ=0,5 und α=0,8) verwendet. Die Experimente basieren ebenfalls auf wertebereichs -und vektorbetragsnormierten Daten. Zu unterscheiden sind die Sprachen Englisch, Deutsch und Japanisch (Trainingsumgebung 1).

Die Tabelle 13 ist wie folgt aufgebaut. In der ersten Spalte steht die Anzahl der Neuronen in der ersten verdeckten Schicht (HL1), in der zweiten Spalte die Anzahl der Neuronen der zweiten verdeckten Schicht (HL2). In den ersten vier Zeilen ist die Anzahl der zweiten verdeckten Schicht 0, d.h. es gibt nur eine verdeckte Schicht.

HL1	HL2	Id-Leistung
5	0	95,6%
10	0	96,1%
20	0	94,8%
30	0	96,5%
10	5	96,5%
10	10	93,9%
15	10	94,8%
20	10	94,3%

Tabelle 13: Vergleich der Identifikationsleistungen bezügl. Netzgröße

Die Lernkurven der vierschichtigen Netze weisen ausgeprägte Zick-Zack-Bewegungen auf. Dies kann mit der großen Anzahl an zu lernenden Netzparametern erklärt werden. Gegenüber Veränderungen der Anzahl der verdeckten Neuronen in den dreischichtigen Netzen ist das Lernverhalten relativ robust. Die Identifikationsleistung schwankt von 93,9% (HL1=10 und HL2=10) bis 96,5% (HL1=30 und HL2=0 bzw. HL1=10 und HL2=5). Eine Aussage zugunsten einer Netztopologie läßt sich nicht treffen. Allerdings sollten aufgrund des Lernverhaltens (Abb. 6) dreischichtige Netze bevorzugt werden.

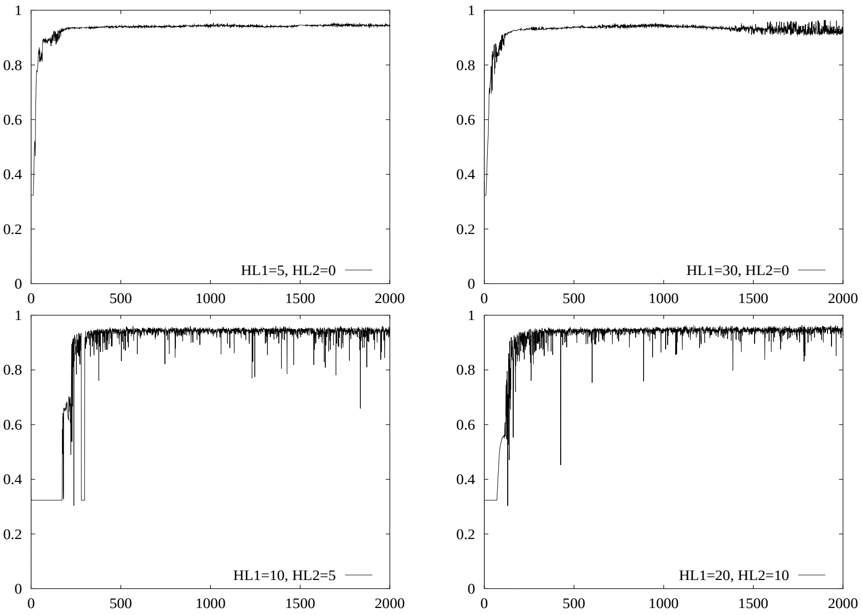

Abbildung 6: Lernkurven mit unterschiedlicher Netztopologie

4.4 Endergebnis

Die bisher gesammelten Erkenntnisse werden zur Bestimmung der Identifikationsleistung des Systems miteinander verbunden. Die phonembasierten Daten werden nur längennormiert. Die Ausgaben der wortbasierten Erkenner werden mit einer Kombination aus Wertebereichsnormierung und Vektorbetragsnormierung transformiert. Trainiert wird ein dreischichtiges Netz mit 20 verdeckten Neuronen. Schrittweite und Momentum werden entsprechend den Ergebnissen des letzten Abschnitts gewählt. Verwendet wird die Trainingsumgebung TB2, die jeweils 314 Äußerungen pro Sprache enthält.

Die Identifikationsleistung beträgt $81,4\%$ bei Verwendung der wortbasierten Erkenner und $85,4\%$ bei den phonembasierten Erkennnern. Gut trennen lassen sich in beiden Fällen die japanischen Äußerungen. Die englischen Muster lassen sich im Falle der phonembasierten Erkenner besser trennen als die deutschen Muster. Umgekehrt ist bei den wortbasierten Daten. Dort lassen sich die deutschen Muster besser trennen.

Schwierigkeiten bereiten die spanischen wortbasierten Äußerungen. So werden 16 der 62 spanischen Äußerungen noch durch das Netz der englischen Sprache zugeordnet. Bei Anwendung des ML-Klassifikators waren es 25 Muster. Das Netz ist also nur sehr schlecht in der Lage, dies zu korrigieren. Die Trennung von deutschen und englischen Äußerungen funktioniert dagegen recht gut. Der ML-Klassifikator verwechselte noch 11 deutschen Muster mit der englischen Sprache. Mit dem Netz konnte dies auf 2 Muster reduziert werden.

	D	E	S	J	D	E	S	J
D	51	4	5	2	58	2	2	0
E	3	56	1	2	3	51	8	0
S	5	9	46	2	3	16	39	4
J	0	3	0	59	0	2	6	54
ID-Leistung	85,4%				81,4%			
	PmitPT				WmitLM			

Tabelle 14: Endergebnis

5 Fazit

Untersucht wurde in dieser Studienarbeit die Anwendung neuronaler Netze zur Identifikation von Sprachen. Dabei wurden im ersten Schritt verschiedene Transformationen des Eingaberaumes des neuronalen Netzes verglichen. Unterschiede ergeben sich bei den phonem- und wortbasierten Daten. So erweist sich bei Verwendung der phonembasierten Erkenner eine Längennormierung als vorteilhaft. Auf eine Wertebereichnormierung sollte bei den phonembasierten Daten verzichtet werden. Die Identifikationsleistungen mit dem ML-Ansatz sind nur knapp oberhalb der Zufallswahrscheinlichkeit. Das neuronale Netz ist dem ML-Klassifikator deutlich überlegen. Es werden 85,4% der Äußerungen richtig klassifiziert.

Die Vorteile der Wertebereichsnormierung werden bei Verwendung der wortbasierten Erkenner deutlich. Im Abschnitt *Ergebnisse mit dem ML-Klassifikator* wird bereits mit dem ML-Ansatz eine Identifikationsleistung von 74,2% erreicht. Bei Verwendung von neuronalen Netzen reicht diese Normierung allerdings nicht aus (siehe Abschnitt *Informationsreduktion*). Zusätzlich ist eine Vektorbetragsnormierung erforderlich, da ansonsten der Gradientenabstieg versagt. Mit einer Kombination aus beiden Normierungen werden 81,4% Identifikationsleistung erreicht.

Als kritisch erweisen sich die spanischen Äußerungen. Ein Großteil der Fehlklassifikationen werden durch die spanischen Äußerungen verursacht. Das Netz kann die Trennung von Spanisch und Englisch nicht oder nur schlecht lernen. Die Trennung von Deutsch und Englisch fällt dem Netz wesentlich leichter. Insgesamt ist der NN-Klassifikator im Vergleich zu dem ML-Klassifikator erfolgreicher.

Neben der Eliminierung probleminvarianter Eigenschaften ist auch die Hinzunahme weiterer Informationen von Vorteil. Die Ergebnisse des Abschnitts *Vergrößerung des Eingaberaumes* zeigen, daß die Hinzunahme der Ausgabe eines weiteren Erkenners eine Verbesserung bringt. Ausnahme bilden auch hier die Ausgaben der spanischen Erkenner.

Im zweiten Schritt wurde dann versucht, die Lernparameter und die Netzgröße richtig einzustellen. Es zeigte sich, daß das Lernverhalten gegenüber Änderungen des Momentums sehr empfindlich ist. Schrittweite und Trägheitsmoment müssen aufeinander abgestimmt sein. Die Parameter lassen sich nur bedingt unabhängig voneinander einstellen. Aber es kommt nicht darauf an, die richtigen Parameter zu finden, sondern die falschen Parameter zu vermeiden. Die Wahl der Netzgröße ist weniger kritisch. Es sollten allerdings nur dreischichtige Netze verwendet werden. Die in Abschnitt *Netzgröße* gezeigten Lernkurven der vierschichtigen Netze deuten auf ein eher unstabiles Lernverhalten hin.

Insgesamt war der Einsatz neuronaler Netze erfolgreich. Vergleicht man die Identifikationsleistung mit den Ergebnissen, die mit dem ML-Klassifikator erreicht wurden, so wurde eine erhebliche Leistungssteigerung erreicht.

Literatur

[1] T.J. Hazen und V.W. Zue: *Automatic Language Identification using a Segment-based Approach* in: Proc. Eurospeech, S. 1303-1306, Berlin 1993.

[2] R. Rojas: *Theorie der neuronalen Netze, Eine systematische Einführung*, Springer Verlag 1993.

[3] G. Ruske: *Automatische Spracherkennung, Methoden der Klassifikation und Merkmalsextraktion*, R. Oldenbourg Verlag 1988.

[4] T. Schultz: *Identifizierung von Sprachen, Exemplarisch aufgezeigt am Beispiel der Sprachen Deutsch, Englisch und Spanisch* Diplomarbeit, Universität Karlsruhe.

[5] T. Schultz, I. Rogina und A. Waibel: *LVCSR-based Language Identification* in: Proc. ICASSP 1996.

[6] M.A. Zissman und E. Singer: *Automatic Language Identification of Telephone Speech Messages using Phoneme Recognition and N-gram Modeling* in: Proc. ICASSP, S. 305-308, volume 1, Adelaide 1994.