Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 82 (1991)

Heft: 21

Artikel: Wie künstliche Neuronen natürliche Sprache erkennen

Autor: Pietro, Gianni N. di

DOI: https://doi.org/10.5169/seals-903029

Wie künstliche Neuronen natürliche Sprache erkennen

Gianni N. di Pietro

Der jüngste Ansatz in der maschinellen Erkennung gesprochener Sprache bedient sich der Neuronalen Netze. Der vorliegende Artikel vermittelt in Form einer exemplarischen Auswahl von Techniken eine Übersicht über die Verwendung Neuronaler Netze in der Spracherkennung.

L'approche la plus récente pour reconnaître à la machine la parole se base sur les réseaux neuronaux. Cet article donne, à l'aide d'exemples choisis de techniques, un aperçu des différentes utilisations des réseaux neuronaux dans la domaine de la reconnaissance de la parole.

Adresse des Autors

Gianni N. di Pietro, Dipl. Informatik-Ing. ETH, Leiter Fachgruppe Spracherkennung, Ascom Tech AG, Bielstrasse 122, 4502 Solothurn, Telefon 065 24 28 84.

Sprache konnektionistisch (mit Neuronalen Netzen) zu modellieren, ist der jüngste Ansatz in der Spracherkennung. Man verspricht sich davon eine Verbesserung der Erkennungsraten, einerseits wegen der inhärent diskriminativen Natur der Trainingsmethoden von Neuronalen Netzen, andererseits weil diese generellere nichtlineare Strukturen zur Verfügung stellen, als die auf stochastischen Modellen basierenden Methoden bieten. Einfache Experimente haben denn auch gezeigt, dass Neuronale Netze in Teilaspekten der Spracherkennung den bisher bekannten Methoden zumindest ebenbürtig sind. Zudem bietet der – vor allem gegenüber stochastischen Modellen – intuitiv zugänglichere Formalismus einen besonderen Reiz.

Wie werden nun Neuronale Netze in der Spracherkennung eingesetzt? Eine erste grobe Unterteilung lässt sich in rein neuronale Ansätze und solche, bei denen die Netze in Kombination mit anderen Methoden eingesetzt werden, machen. Im ersten Fall kann man unterscheiden, ob man ein Sprach-Segment (Wort, Phonem usw.) statisch zu erkennen versucht, oder ob man der dynamischen, also zeitlich veränderlichen Natur der Sprache in irgendeiner Form Rechnung trägt. Im zweiten Fall kann festgestellt werden, dass man Neuronale Netze mit praktisch allen bekannten Spracherkennungs-Methoden zu kombinieren versucht: Als Teil eines Dynamic Time Warping-Ansatzes, in Kombination mit Hidden Markov Models und zusammen mit Methoden der künstlichen Intelligenz. Für eine Einführung in diese Techniken sei auf den Artikel [1] verwiesen. Eine detailliertere Studie zur Verwendung Neuronaler Netze in der Spracherkennung findet sich in [2].

Ein einfaches Spracherkennungsmodell

Zur Beschreibung des Ablaufs der Erkennung von Sprache oder von Sprachbestandteilen kann das einfache Modell von Bild 1 dienlich sein.

Ein Segment eines Sprachsignals, das zum Beispiel ein Wort, eine Silbe oder ein Phonem umfassen kann, wird, nach einer geeigneten Filterung und Digitalisierung, einer *Merkmalsextraktion* unterzogen. Diese dient zur Reduktion der anfallenden Sprachdaten und zur Verringerung der Redundanz. Die hauptsächlich anzutreffenden Verfahren sind Kurzzeit-Fouriertransformation (was ein sog. Spektrogramm liefert), Linear Predictive Coding, Cepstrum-Analyse, Verfahren, die an das menschliche Gehörsystem angelehnt sind (Simulationen der Cochlea, Cochleagramm) sowie extrahierte artikulatorische Parameter wie stimmhaft/stimmlos, Reibelaut, Zischlaut usw.

Die so entstandenen Muster müssen dann klassiert werden. Als Methoden zur *Klassierung* kommen in Frage:

- Dynamic Time Warping (DTW), eine auf dynamischer Programmierung beruhende Methode,
- Hidden Markov Models (HMM), eine probabilistische Methode,
- Regelbasierte Techniken aus der künstlichen Intelligenz (KI), vor allem die Verwendung eines Regelwerks, das die Fähigkeit eines Experten, Spektrogramme zu «lesen», beschreibt sowie
- die Neuronalen Netze.

Spracherkennung mit Neuronalen Netzen ist ein zweistufiger Prozess: In der Trainingsphase (oberer Teil von Bild 1) wird ein Neuronales Netz mit Mustern aus einer geeigneten Datenbank so konditioniert, dass es nach

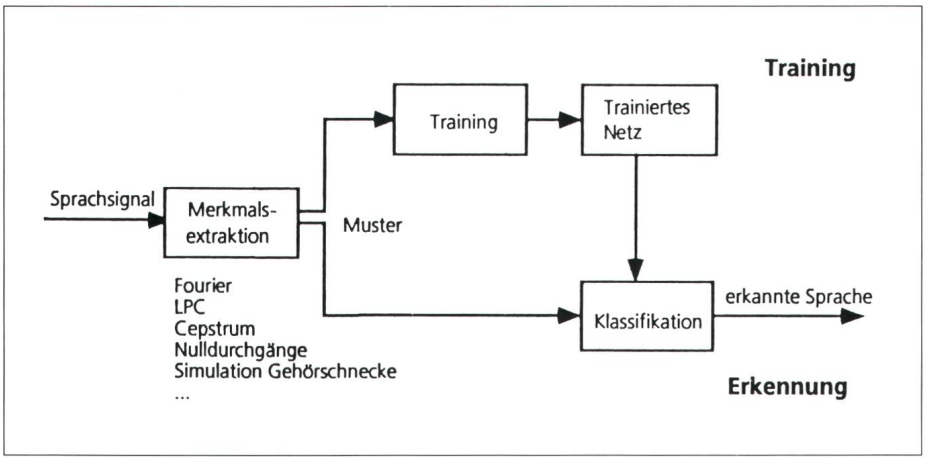


Bild 1 Verarbeitungsschritte bei der neuronalen Spracherkennung

und nach in die Lage kommt, die gelernten sowie neue Muster richtig zu klassieren. In diesem Punkt unterscheidet sich der neuronale Ansatz wenig vom probabilistischen: Eine vollständige Datenbasis, welche eine repräsentative Auswahl der später zu erkennenden Muster umfasst, wird genauso vorausgesetzt wie eine intelligente Trainingsstrategie.

In der Betriebsphase wird das so gewonnene Neuronale Netz zur Klassierung eingesetzt.

Neuronale Netze

Wir definieren hier einige Begriffe, die bei der Betrachtung der verwendeten Modelle verwendet werden (Bild 2). In konnektionistischen Modellen entspricht der Aktivierungsgrad $a_j(t)$ eines Verarbeitungselements j (zu einer bestimmten Zeit t) etwa dem Membranpotential eines Neurons. Das Verarbeitungselement erhält eine oder mehrere Eingaben und gibt sie dem nächsten logischen Abschnitt weiter. Dieser berechnet die Übertragungsfunktion $net(\cdot)$

$$n_j(t) = net(\mathbf{x}, \mathbf{m}) = net(\eta_1,\eta_2,\ldots,\eta_n,\mu_1,\mu_2,\ldots,\mu_n) \quad (1)$$

der erhaltenen Signale (meist die Summe der mit den entsprechenden μ_{jn} gewichteten Signale). Der nächste Abschnitt berechnet die Aktivierungsfunktion $F(\cdot)$ des künstlichen Neurons

$$a_j(t) = F_j[n_j(t)] \quad (2)$$

und (optional) eine Ausgabefunktion $f(\cdot)$

$$\eta_j(t) = f_j[a_j(t)], \quad (3)$$

welche ein einzelnes Signal $\eta_j(t)$ liefert.

Die Art, wie die einzelnen Elemente miteinander zu einem Netz verschaltet sind, wird als Netzwerkstruktur bezeichnet. Das Lernen geschieht in Neuronalen Netzen durch gezielte Veränderung der Gewichte μ_{jn} in den Verbindungen. Je nachdem, ob das Lernen automatisch vor sich geht oder es der Überwachung durch den Benutzer bedarf, spricht man von einem beaufsichtigten oder von einem unbeaufsichtigten Lernverfahren. Für eine Übersicht über die einzelnen Modelle sei im übrigen auf die Einführung [3] oder [4] verwiesen.

Klassierung statischer Muster

Klassierung statischer Muster bedeutet, dass Sprachmerkmale als Ganzes einem Neuronalen Netz zur Klassierung präsentiert werden. Dieses Vorgehen nimmt keinen Bezug auf Gegebenheiten in der Nachbarschaft des untersuchten Sprachsegments und ebensowenig auf die dynamische Natur der Sprache. Deshalb bleibt der Einsatz der beschriebenen Methoden auf die Klassierung einzelner Wörter aus einem bescheidenen Vokabular beschränkt.

Multilayer Perceptron

Eines der einfachsten Modelle, welche zur Klassierung von Sprachmustern eingesetzt werden, ist zweifellos das Multilayer Perceptron. Das Spektrum der damit klassierten Muster reicht von ganzen Wörtern (z.B. chinesische Zahlen in [5] – hier bestimmen darüber hinaus Überlegungen aus der perzeptiven Phonetik die Netzwerkstruktur) über Vokale [6] zum ganzen Satz von Phonemen [7]. Das Vorgehen beim Training der Netze ist bekannt: Ein Muster wird an die Eingabeschicht angelegt, die Aktivierung schichtweise bis zur Ausgabeschicht propagiert; hier wird sie mit der gewünschten Aktivierung verglichen und die Verbindungen werden entsprechend angepasst (Backpropagation). Das wird mit allen Mustern aus der Datenbasis wiederholt, bis sich die Erkennungsrate stabilisiert. Die Anzahl Ausgabeelemente ist durch die Anzahl Klassen festgelegt, die Anzahl Eingabeelemente durch die Auflösung der Muster.

Bis auf wenige Ausnahmen wird bei Verwendung dieser Methode im Vergleich zu traditionellen Methoden von besseren Erkennungsresultaten berichtet.

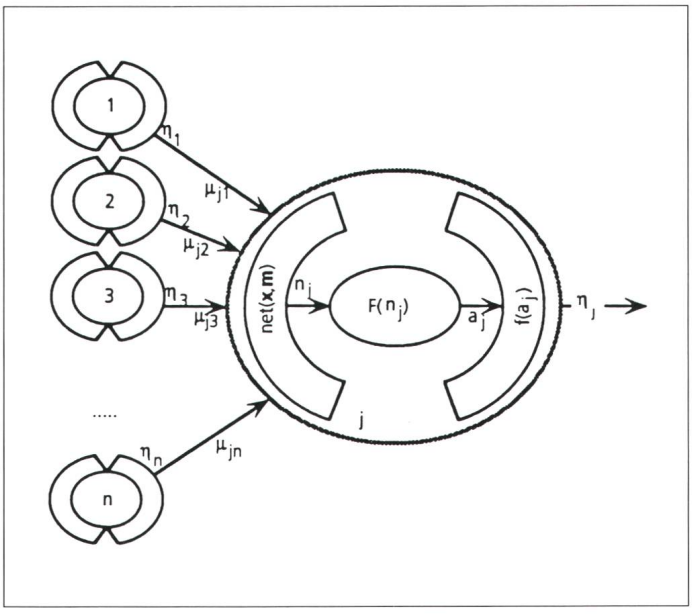


Bild 2 Ein neuronales Verarbeitungselement

Learning Vector Quantizer

Teuvo Kohonen von der Universität Helsinki, bekannt durch seine selbstorganisierenden Karten, auf die wir weiter unten zu sprechen kommen, hat in drei Etappen in den Jahren 1986–1990 seinen Learning Vector Quantizer entwickelt und verfeinert. Die Vektorquantisierung bildet eine Sequenz von kontinuierlichen oder diskreten (Muster-)Vektoren auf eine digitale Sequenz ab, die für die Übertragung über einen digitalen Kanal geeignet ist. Das Ziel (Datenkompression für die Übermittlung) wird durch Auffinden von optimalen Vertretern aller möglichen Klassen und deren anschliessender Codierung erreicht. Der Prozess des Vergleichs eines gegebenen Vektors mit den Referenzvektoren kann aber genausogut für die Klassierung des Vektors verwendet werden.

Kohonen hat nun ein beaufsichtigtes Lernverfahren entwickelt, bei dem von einer neuronal orientierten Struktur Referenzmuster gelernt werden: Diese Muster, welche während des Trainings errechnet werden, sitzen in den Verbindungsgewichten zu den (einschichtig organisierten) Verarbeitungselementen des Vektor-Quantisierers. Jedes dieser Elemente ist dabei für eine Klasse (also z.B. ein Wort) zuständig. Sie berechnen die Distanz D zwischen dem angelegten Muster und demjenigen in den Verbindungsgewichten und passen letztere den angelegten Vektoren an:

$$\text{net}(\mathbf{x}, \mathbf{m}) = D(\mathbf{x}, \mathbf{m}) \qquad (4)$$

Das Vorgehen beim Training wird hier anhand des ersten Modells aus dem Jahr 1986 beschrieben:

$$m_c(t+1) = m_c(t) + \alpha(t)\cdot[x(t) - m_c(t)], \qquad (5a)$$

falls $\boldsymbol{x}(t)$ und $\boldsymbol{m}_c(t)$ zur selben Klasse gehören, sonst

$$m_c(t+1) = m_c(t) - \alpha(t)\cdot[x(t) - m_c(t)], \qquad (5b)$$

und für $i \neq c$. $\quad m_i(t+1) = m_i(t), \qquad (5c)$

$\boldsymbol{x}(t)$ ist das angelegte Muster, $\boldsymbol{m}_c(t)$ das in die zum Element c führenden Verbindungen gespeicherte Muster. Mit $\alpha(t)$ hat man einen Parameter zur Steuerung des Lernverhaltens zur Verfügung. Experimente in sprecherunabhängiger Einzelworterkennung liefern mit diesen Verfahren ähnliche Resultate wie diejenigen mit Multilayer Perceptrons.

In [8] wird die Idee des Learning Vector Quantizers (LVQ) aufgegriffen und so erweitert, dass das Netz gegenüber Verschiebungen des Musters unempfindlich wird. Anstelle des einzelnen Elements pro Klasse werden für jede mögliche Position des Musters mehrere Verarbeitungselemente verwendet. Deren gemittelte Aktivierung stellt die Evidenz für ein bestimmtes Muster dar. Die so erhaltenen Referenzmuster dienen der Erkennung durch ein auf Hidden Markov Models basierendes System. Mit diesem Verfahren werden zum Beispiel alle japanischen Konsonanten in mehr als 97% der Fälle richtig erkannt.

Glossar einiger wichtiger Begriffe

Training/Lernen: Gezielte Modifizierung der Verbindungen in einem neuronalen Netz. Dieses soll dadurch in die Lage kommen, ein Eingabemuster auf ein zugehöriges Ausgabemuster abzubilden, bzw. das Eingabemuster zu klassieren.

Beaufsichtigtes Lernen: Hier wird das Ausgabemuster von aussen vorgegeben, d.h. in jedem Trainingsschritt muss die Zuordnung von Eingabe- und Ausgabemuster festgelegt werden.

Unbeaufsichtigtes Lernen: Durch einen selbstorganisierenden Prozess finden unbeaufsichtigte Lernalgorithmen eine geeignete Abbildung von Eingabe- und Ausgabemustern. Eine vorgegebene Bewertungsfunktion gibt an, wie gut die aktuelle Zuordnung der gestellten Aufgabe gerecht wird.

Muster/Muster-Vektor: Muster liegen in unserem Fall als Matrizen (mit der Zeit auf der horizontalen Achse) oder Vektoren von Werten vor. Es kann sich dabei zum Beispiel um ein Spektrogramm oder um einen zeitlichen Ausschnitt davon handeln.

Klasse: Gruppe, Einheit mit gemeinsamen, sich von anderen unterscheidenden Merkmalen. Wir fassen beispielsweise alle möglichen Ausprägungen des gleichen Wortes oder des gleichen Wortausschnitts (Phonems) in einer Klasse zusammen.

Referenz-Muster/Referenz-Vektor: Vertreter (zentrales Element) einer bestimmten Klasse. Dieser kann sich zum Beispiel dadurch auszeichnen, dass seine Distanz zu allen Klassenmitgliedern minimal ist.

Distanz: Mass der Unterschiedlichkeit zweier Vektoren, zum Beispiel:

$$D_E(\mathbf{x}, \mathbf{m}) = \sqrt[\xi]{\sum_{i=1}^{n} |\eta_i - \mu_i|^{\xi}}$$

Beachte: für $\xi = 2$ erhalten wir die euklidische Distanz.

Grammatik: Teil der Sprachwissenschaft, der sich mit den sprachlichen Formen und deren Funktion im Satz, mit den Gesetzmässigkeiten, dem Bau einer Sprache beschäftigt.

Syntax: Charakterisierung der grammatisch wohlgeformten Sätze einer Sprache, bzw. der korrekten Verknüpfung sprachlicher Einheiten im Satz.

Semantik: Lehre von der Bedeutung sprachlicher Zeichen und Zeichenfolgen. Hauptsächlich an die Ebene des Wortes gebunden, erfasst aber alle Schichten und Bereiche der Sprache.

Pragmatik: Beschreibung des Gebrauchs von Äusserungen, der Absichten, die mit Äusserungen verfolgt werden, und der Wirkungen, die Äusserungen auf Hörer haben. Anders als bei der Semantik wird hier die kommunikative Kompetenz von Menschen untersucht.

Hidden Markov Models (HMM): Eine statistische Beschreibung einer Folge von Symbolen, etwa von Muster-Vektoren. Die Wahrscheinlichkeit, dass eine bestimmte Folge beobachtet wird – unter der Voraussetzung, dass sie von einem HMM generiert wurde – entscheidet über die Zugehörigkeit zu einer bestimmten Klasse [1].

Dynamic Time Warping (DTW): Ein auf dynamischer Zeitorientierung basierendes Verfahren zum Vergleichen von Mustern [1].

Klassierung dynamischer Muster

Neuere neuronale Klassifikatoren, die kurze Verzögerungen, eine wie auch immer geartete Integration über die Zeit oder rückführende Verbindungen enthalten, wurden eigens zu Zwecken der Spracherkennung entwickelt, allen voran, zumindest was die Verbreitung durch publizistische Tätigkeit anbelangt, Alex Waibels sogenannte Time-Delay Neural Networks. Aus den Multilayer Perceptrons ent-

wickelte man ebenfalls die rekursiven Netze, das heisst solche mit rückführenden Verbindungen, bei welchen Klassierungsresultate früherer Perioden auf die aktuellen einen Einfluss ausüben. Einen eigenen Ansatz stellen Kohonens selbstorganisierende topologische Karten dar. Diese bilden eine Sequenz von Sprachabschnitten auf eine zweidimensionale Anordnung von Verarbeitungselementen ab.

Time-Delay Neural Nets

TDNN basieren auf Multilayer Perceptrons. Den Verarbeitungselementen werden aber Verzögerungen D_1 bis D_N zugeordnet (Bild 3). Die Ausgaben

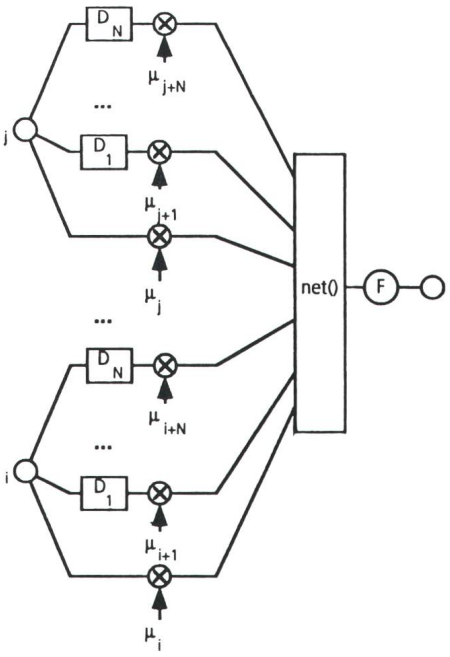


Bild 3 Ein Time-Delay Neural Network-Verarbeitungselement

der j Verarbeitungselemente werden jeweils mit mehreren Gewichten multipliziert. Auf diese Weise ist es einem Element möglich, die aktuelle Eingabe mit den vergangenen in Verbindung zu setzen und zu vergleichen. Lernen geschieht wie bei den Multilayer Perceptrons durch Backpropagation.

Versuche zur Erkennung der Phoneme /b/, /d/ und /g/ in verschiedenen phonetischen Kontexten führten in [9] auf eine sprecherabhängige Erkennungsrate von 98,5%, was weit über den 93,7% eines auf Hidden Markow-Models basierenden Ansatzes liegt. Die Idee wurde in der Folge weiterentwickelt in Richtung auf modulare Netze hin, welche je nur für einen Teil der zu erkennenden Phoneme zuständig sind und durch geschickte Zusammenschaltung grössere Aufgaben zu bewältigen vermögen [10]. Dass das Verfahren auch zur sprecherunabhängigen Erkennung tauglich ist, wurde in [11] demonstriert.

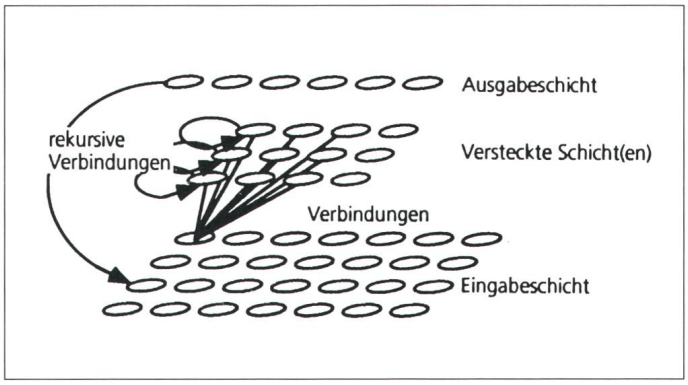


Bild 4 Prinzip der rekursiven Netze

Rekursive Netze

Ein anderer Weg, zeitliche Abhängigkeiten zu berücksichtigen, besteht darin, in Multilayer Perceptrons rückführende Verbindungen einzubauen (Bild 4). Die dadurch entstehende Verkomplizierung des Trainings sowie die entstehenden Stabilitätsprobleme seien hier nur am Rande erwähnt. In [12] stellte man fest, dass durch diese rückführenden Verbindungen der unweigerlichen Variationen in Sprechgeschwindigkeit und -rhythmus ähnlich wie beim Dynamic Time Warping Rechnung getragen werden konnte. (Diese Variationen können nicht nur zwischen verschiedenen Personen, sondern auch bei Äusserungen derselben Person zu verschiedenen Zeitpunkten sehr gross sein.)

Die weiter oben beschriebene Aufgabe, die Phoneme /b/, /d/ und /g/ zu erkennen, wurde mit rekursiven Netzen in [13] mit einer Erkennungsrate von 98% gemeistert.

Phonotopische Karten

Die kompetitiven Netze und die Feature Maps arbeiten nach einem ähnlichen Prinzip wie die LVQ, jedoch unbeaufsichtigt. Die Anzahl Eingangsverbindungen ist wiederum durch die Dimension der Mustervektoren gegeben. Die Anzahl Elemente ist mindestens so gross wie die Anzahl Klassen, oft aber grösser. Die Aktivierungsfunktion berechnet die Distanz zwischen dem angelegten Vektor und dem Vektor der Gewichte in den Verbindungen.

Der «Gewinner» mit der kleinsten Distanz wird (im Falle eines kompetitiven Netzes) auf die höchste Aktivierung gesetzt, alle anderen auf die niedrigste:

$$\eta_j(\mathbf{x}, \mathbf{m}_j) = \begin{cases} 1, \text{ wenn } D(\mathbf{x}, \mathbf{m}_j) < D(\mathbf{x}, \mathbf{m}_i) \text{ für alle } i \\ 0, \text{ sonst} \end{cases} \quad (6)$$

Beim Lernen werden die Verbindungen zum Gewinner verändert:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \varepsilon(t) \cdot [\mathbf{x} - \mathbf{m}_j(t)] \cdot \eta_j. \quad (7)$$

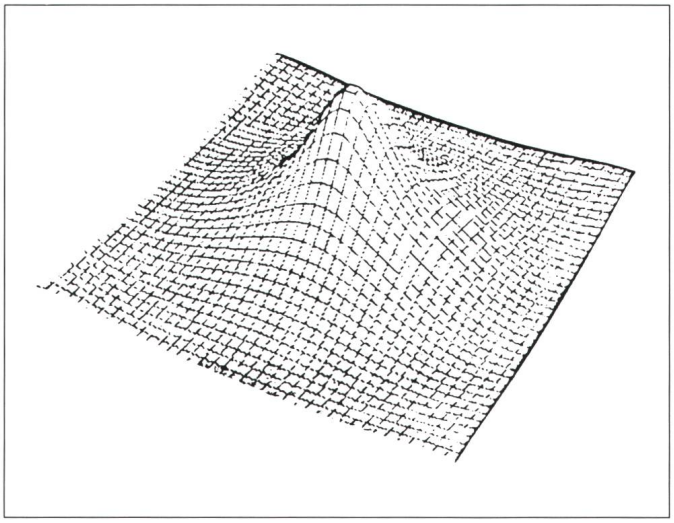


Bild 5 Nachbarschaftsfunktion «mexikanischer Hut»

Der Parameter $\varepsilon(t)$ ist die Lernrate. Er kann mit der Anzahl Iterationen gegen Null gehen.

Kohonens Feature Maps (Self Organizing Feature Maps) [14] gehen auch von der Distanz zwischen den Gewichten und dem angelegten Vektor aus und berechnen die Ausgabe der Elemente entsprechend. Das Lernen allerdings berücksichtigt nicht nur den Gewinner, sondern auch seine topologische Nachbarschaft $N(\cdot)$:

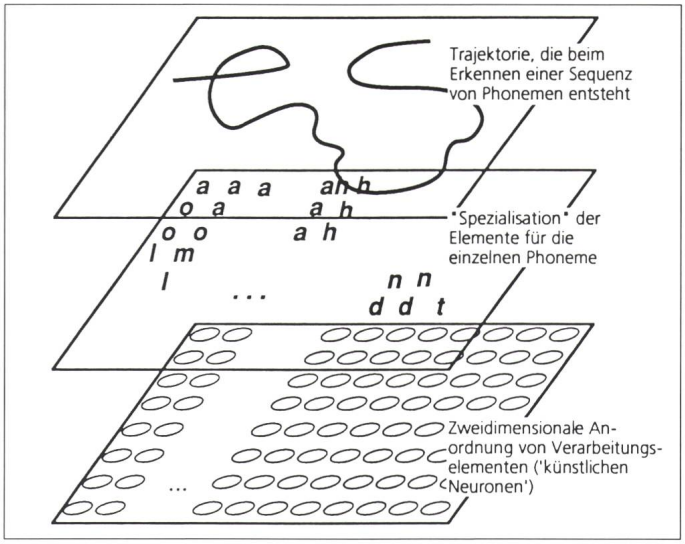
$$m_i(t+1) = \begin{cases} m_i(t) + \varepsilon(t)\cdot[x - m_i(t)] \cdot N_j(t), \\ \text{wenn } i \in N_j(t) \\ m_i(t), \text{ sonst} \end{cases} \quad (8)$$

Die Funktion $\varepsilon(t)$, die sogenannte Verstärkungssequenz, strebt langsam gegen 0. Die Nachbarschaftsfunktion $N_j(t)$ definiert die Nachbarschaft des Elementes j (in Form des berühmten «mexikanischen Hutes», Bild 5) und kann sich zeitlich so verändern, dass die Nachbarschaft mit zunehmenden Iterationen kleiner wird.

Dieser Miteinbezug benachbarter Elemente bewirkt geordnete und lokalisierbare interne Repräsentationen. Das hat nicht nur den Vorteil, dass sie für den menschlichen Betrachter anschaulicher werden, sondern dass ähnliche Muster auch benachbarte Zonen der Feature Map aktivieren. Darüber hinaus ist die Ausdehnung der jeweils aktivierten Zone mit der relativen Häufigkeit des zugehörigen Mustervektors korreliert.

Phonotopische Karten setzen die Information, welche in einer Sequenz von Eingabemustern steckt, in Pfade durch eine zweidimensionale Anordnung von Elementen um (Bild 6). Diese müssen weiterinterpretiert werden. Dafür benutzt Kohonen eine kontext-sensitive stochastische Grammatik, deren Regeln durch ein von ihm entwickeltes Verfahren aus Beispielen automatisch generiert wurde. Probleme beim Erkennen von Explosivlauten umgeht er durch Verwendung einer weiteren Karte, die nur auf diese Laute spezialisiert ist. Das System setzt auf diese Weise ein Sprachsignal in eine phonetische und dann in eine orthographische Transkription um. Das wird dadurch erleichtert, dass für die untersuchten Sprachen – Finnisch und Japanisch – die Orthographie fast identisch mit der phonetischen Transkription ist. Je nach Sprecher und Schwierigkeit des Textes kommt das System auf eine Erkennungsrate von 92 bis 97%.

Bild 6
Phonotopische Karte



Kombination mit herkömmlichen Techniken

Aus der Kombination von Neuronalen Netzen mit herkömmlichen Techniken erhofft man sich eine Verquickung der Vorteile beider Welten unter Umgehung der jeweiligen Nachteile. Aus der Fülle der Ansätze seien hier nur einige wenige herausgepickt.

So wird beispielsweise in [15] ein Multilayer Perceptron (MLP) in einem Dynamic Time Waping (DTW)-Ansatz anstelle der Kostenfunktion zur Überprüfung der Übereinstimmung der zu vergleichenden Vektoren verwendet. Das Lernen zeitverschobener sowie zeitverzerrter Muster wird durch die Verwendung von DTW hinfällig. Bei der sprecherunabhängigen Erkennung der japanischen Ziffern erreicht das System eine Rate von 98 bis 99%.

Auch in Kombination mit Hidden Markov Models (HMM) lassen sich Neuronale Netze einsetzen: Beispielsweise werden Multilayer Perceptrons dazu verwendet, in HMM die Auftretens-Wahrscheinlichkeit eines Symbols in einem bestimmten Zustand zu berechnen [16]. Oder es wird von einem Time-Delay Neural Net- oder einem Learning Vector Quantizer-Netz die bei HMM-Systemen notwendige vorgängige Vektorquantisierung übernommen [17].

Darüber hinaus weisen theoretische Arbeiten von Bourlard darauf hin, dass Hidden Markov Models und Neuronale Netze nicht so verschieden sind: In [18] zeigt er auf, dass ein Netz, welches rückführende Verbindungen aufweist und den linken und rechten Kontext des zu klassifizierenden Sprachabschnitts miteinbezieht, einem bestimmten HMM-Modell äquivalent ist.

Vertikale Integration

Spracherkennung, vor allem die Erkennung zusammenhängender Sprache, umfasst je länger je mehr nicht nur die auf Merkmalsextraktion aufbauende akustisch-phonetische Decodierung, von der bisher die Rede war, sondern noch eine ganze Hierarchie von Komponenten, die aufeinander aufbauen und sich gegenseitig beeinflussen (Bild 7). Das geht von der lexikalischen Analyse, wo Phonemketten lexikalischen Einheiten zugeordnet werden, über Syntax, Semantik zur internen Wissensrepräsentation, wenn eine Art «Verständnis» der Sprache gefordert ist (etwa in Echtzeit-Übersetzungssystemen, wie sie die japanische Forschung zu konstruieren beabsichtigt). Beschäftigt man sich hingegen zum Beispiel mit sprachgesteuerten Auskunftssystemen, so gilt es, robuste Dialogsteuerungs-Algorithmen bereitzustellen, welche auch mit der Eigenschaft von Menschen, sich ungenau, abgehackt und unentschlossen zu äussern, fertig werden.

In Bild 7 sind diejenigen Bereiche grau unterlegt, bei welchen Neuronale Netze zur Anwendung kommen. Der Einsatz Neuronaler Netze zur Modellierung höherer linguistischer Ebenen steckt in den Anfängen. In [19] wird festgehalten, dass linguistisches Wissen über die hierarchische Natur von sprachlichen Repräsentationen in Spracherkennungssystemen (jeglicher Art) einzig zur Beschränkung der Ab-

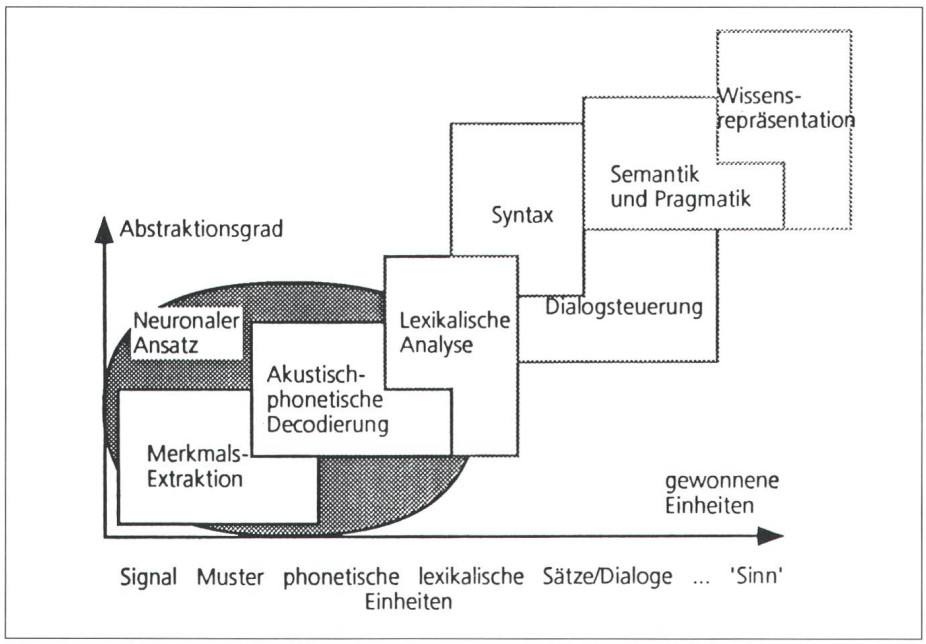


Bild 7 Baukasten der Spracherkennungskomponenten

folge phonetischer Symbole eingesetzt wird. Es wird vorgeschlagen, jedes Stück verfügbaren linguistischen Wissens sorgfältig auf dessen Verwendbarkeit in einem Spracherkennungssystem zu untersuchen und Architekturen – allen voran Neuronale Netze – zu entwerfen, welche dieses Wissen nutzen können.

In ersten Versuchen wird auf einer (mehr oder weniger traditionellen) phonetischen Komponente, die aber aus dem Signal mehrere Merkmale extrahiert, wie etwa Periodizität, Aperiodizität, Übergänge, Dauer, Beginn einer Silbe, eine neuronale lexikalische Komponente aufgebaut, welche phonologisches und syntaktisches Wissen berücksichtigt.

Kritik und Ausblick

Es ist bekannt, dass ausgeklügelte Klassierungswerkzeuge vonnöten sind, weil das Wissen, wie man bei der Merkmalsextraktion an die relevanten Informationen gelangen kann, nicht vorhanden ist. Fortschritte in der Vorverarbeitung (Datenkompression und Merkmalsextraktion ohne Verlust wesentlicher Information) sind demnach Voraussetzung für die darauf aufbauende Erkennung.

Wie es scheint, eignen sich die meisten Neuronalen Netze in ihrer heutigen Form nicht für die Modellierung zeitlicher Sequenzen. Vor allem Kontextabhängigkeiten und möglicherweise relevante Informationen aus länger zurückliegenden sprachlichen Äusserungen werden in den derzeitigen Modellen kaum berücksichtigt. Eine mögliche Lösung dieses Problems liegt in der Erforschung rekursiver Netze. Die genaue Arbeitsweise (und das Potential) dieser Netze ist aber zurzeit im wesentlichen noch unbekannt.

Je grösser die zu lösende Aufgabe ist, um so schwieriger wird es, eine Sprachdatenbasis von Hand zu segmentieren und zu markieren. Hier müssen Verfahren gefunden werden, die unbeaufsichtigt mit grösseren Mengen von sequentiell angebotener Sprache umgehen können.

Grosse Anstrengungen müssen vor allem gemacht werden, um linguistisches Wissen in neuronale Spracherkennungssysteme zu integrieren. Mögliche Fragen hierzu sind:

- Wie integriert man aus der Verarbeitung natürlicher Sprache bekannte Analysenmethoden in einen neuronalen Ansatz?
- Wie modelliert man die in solchen Spracherkennungssystemen inhärente Ungewissheit auf höherer Ebene?
- Ist es möglich, lexikalische, syntaktische und pragmatische Analysen neuronal zu modellieren, und wenn ja, auf welche Weise?

Auf algorithmischer Ebene ist die Suche nach schnellen Lernverfahren, speziell im Hinblick auf grössere Mengen Trainingsmaterial und umfangreichere Probleme dringend vonnöten. Und dazu gehört auch die Handware-Realisierung von aussichtsreich scheinenden Netzen, um die langsame und fehleranfällige Simulation auf sequentiellen Rechnern zu umgehen.

Verdankung

Dieser Aufsatz geht aus einer Studie hervor, die von Juli 1990 bis Juni 1991 von der Gruppe Auris der Ascom Tech für die GD PTT angefertigt wurde.

Bibliographie

[1] *Stephen E. Levinson, David B. Roe:* A Perspective on Speech Recognition. IEEE Communications Magazine. Jan. 1990. S. 28–34.
[2] *Richard P. Lippmann:* Review of Neural Networks for Speech Recognition. Neural Computation. Vol. 1. 1989. S. 1–38.
[3] *J.-F. Leber, M.B. Matthews:* Neurale Netzwerke: Eine Übersicht. Bulletin SEV/VSE 80 (1989)13. S. 923–932.
[4] *Jakob Bernasconi:* Neuronale Netzwerke: Theorie und Praxis. Bulletin SEV/VSE 82 (1991)15. S. 11–16.
[5] *Haïyan Yé, Shengrui Wang, François Robert:* A PCMN Neural Network for Isolated Word Recognition. Speech Communications. 9 (1990) 2. S. 141–153.
[6] *Mahesan Niranjan, Frank Faliside:* Neural network and radial basis functions in classifying static speech patterns. Computer Speech and Language (1990). S. 275–289.
[7] *Hong C. Leung, Victor W. Zue:* Applications of Error Back-Propagation to Phonetic Classification. In: D.S. Touretzky: Advances in Neural Information Processing Systems 1. S. 207–214.
[8] *E. McDermott, H. Iwamida, S. Katagiri, Y. Tohkura:* Shift-Tolerant LVQ and Hybrid LVQ-HMM for Phoneme Recognition. In: Alex Waibel. Kai-Fu Lee: Readings in Speech Recognition. S. 425–438.
[9] *Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano* und *Kevin J. Lang:* Phoneme Recognition Using Time-Delay Neural Networks. IEEE Trans. on Acoustics. Speech and Signal Processing (1989) 3. S. 328–339.
[10] *Alex Waibel, Hidefumi Sawai, Kiyohiro Shikano:* Consonant Recognition by Modular Construction of Large Phonemic Time-Delay Neural Networks. Proceedings of the ICASSP Conference 1989. 9. S. 3.9.
[11] *J.B. Hampshire II, A.H. Waibel:* A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks. IEEE Trans. on Neural Networks. 1 (1990) 2. S. 216–228.
[12] *H. Bourlard, C.J. Wellekens:* Speech Dynamics and Recurrent Neural Networks. Proceedings of the ICASSP Conference 1989. 1. S. 1.9.
[13] *Raymond L. Watrous, Bruce Ladendorf, Gary Kuhn:* Complete Gradient Optimization of a Recurrent Network applied to /b, /d, /g/Discrimination. Zur Publikation eingegeben beim Journal of the Acoustic Society of America.
[14] *Teuvo Kohonen:* Self-Organization and Associative Memory. 3. Aufl., Springer. 1989.
[15] *Hiroaki Sakoe, Ryosuke Isotani, Kazunaga Yoshida, Ken-ichi Iso, Takao Watanabe:* Speaker-Independent Word Recognition Using Dynamic Programming Neural Networks. Proceedings of the ICASSP Conference 1989. 1. S. 1.8.
[16] *N. Morgan, H. Bourlard:* Continuous Speech Recognition Using Multilayer Perceptrons with Hidden Markov Models. Proceedings of the ICASSP Conference 1990. Albuquerque. S. 8.1.
[17] *W. Ma, D. Van Compernolle:* TDNN Labeling for a HMM Recognizer. Proceedings of the ICASSP Conference 1990. Albuquerque. S. 8.3.
[18] *H. Bourlard, C.J. Wellekens:* Links Between Markov Models and Multilayer Perceptrons. Philips Research Laboratory. September 1988.
[19] *M. Huckvale:* Exploiting Speech Knowledge in Neural Nets for Recognition. Speech Communications. Vol. 9. Nr. 1. 1990. S. 1–14.