KfK 4062
April 1986

Ein schneller Algorithmus zur phonetischen Segmentation kontinuierlich gesprochener Sprache

D. Smidt
Institut für Reaktorentwicklung

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE
Institut für Reaktorentwicklung

KfK 4062

Ein schneller Algorithmus zur phonetischen Segmentation kontinuierlich gesprochener Sprache

D. Smidt

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Zusammenfassung

Für die schnelle Klassifikation von Phonemen in akustischen Sprachspektren wurde das Verfahren des differentiellen Lernens (DL-Verfahren) entwickelt und in Verbindung mit einem einfachen Algorithmus zum Erkennen kontinuierlich gesprochener Sätze erprobt. Im DL-Verfahren wird im Lernschritt nach einer Fehlklassifikation jeweils nur die eine Musterkomponente zur Erstellung einer neuen Regel verwendet, die am stärksten vom Referenzwert abweicht. Mehrere solcher Regeln werden sukzessive konjunktiv oder disjunktiv zusammengefaßt. Die Erprobung zeigt zunächst für einen Sprecher ein gutes Klassifikationsvermögen und eine sehr große Schnelligkeit. Die Erweiterung um weitere Merkmale, die automatisch nach ihrer Relevanz ausgewählt werden, wird diskutiert. In einem abschließenden Ausblick wird gezeigt, daß Prozesse, die dem DL-Verfahren entsprechen, auch in der Mustererkennung durch lebende Wesen mit der damit verbundenen Fähigkeit der Generalisierung und Differenzierung eine Rolle spielen könnten.

Abstract

A Fast Algorithm for the Phonemic Segmentation of Continuous Speech

The method of Differential Learning (DL-method) has been applied to the fast phonemic classification of acoustic speech spectra. The method was also tested in connection with a simple algorithm for the recognition of continuous speech. In every learning step of the DL-method only that single pattern component will be used for a new rule, which deviates most from the reference value. Several rules of this type will be connected in a conjunctive or disjunctive way. First tests with a single speaker demonstrate good classification and a very large speed. The inclusion of automatically selected additional features according to their relevance is discussed. In a final outlook there will be shown a correspondence between processes related to the DL-method and pattern recognition in living beings with their ability for generalization and differentiation.

Inhaltsverzeichnis

1. Einleitung

Systeme zur Erkennung kontinuierlich gesprochener Sprache führen Gegensatz zu Systemen zur Einzelworterkennung vor der lexikalischen Bearbeitung eine Segmentation der akustischen Eingabedaten durch. Hierzu wird entweder die in Form von Spektrumsdaten vorliegende Information in geeignete zeitliche Abschnitte zerlegt und mit Hilfe von Methoden der dynamischen Programmierung zeitlich den Referenzdaten angepaßt und mit ihnen verglichen, oder es wird vorab eine Klassifikation nach phonetischen Untereinheiten vorgenommen und deren Ergebnis dann mit den Satzmustern verglichen.

Das früheste erfolgreiche Beispiel für den ersten Weg ist HARPY /1,2/. Auch von anderen Gruppen sind Systeme entwickelt worden, die mit HARPY in dem Aspekt übereinstimmen, daß die syntaktische Information in die akustische Analyse einbezogen wird, z. B. /3/. Die Verfahren sind aufwendig, aber recht zuverlässig.

Um den Rechenaufwand zu verkleinern, werden aber zunehmend nach der zweiten Methode Segmentierer und akustische Klassifikatoren eingesetzt, die ohne Hinzunahme des syntaktischen und semantischen Wissens auskommen. Ein frühes Beispiel hierfür ist der "centisecond acoustic processor" aus dem Watson-Research-Center der IBM /4/, aber auch neuere deutsche Arbeiten basieren auf diesem Prinzip /5,6,7/. Hier werden in einem reinen bottom-up-Verfahren phonetische Klassen als Labels zugewiesen und die nachfolgenden Stufen des Erkennungsprozesses arbeiten nur noch mit diesen.

Die für die Segmentierung erforderliche phonetische Klassifikation erfolgt für jedes der z. B. im 10 ms-Takt aufgenommenen, vorbearbeiteten und geglätteten akustischen Spektren. Dem phonetischen Klassifikator steht hier also jeweils nur ein Kurzzeitspektrum zur Verfügung, während im Gegensatz dazu ein Prozessor wie bei HARPY oder ein Einzelwort-Klassifikator auf eine charakteristische zeitliche Folge mehrerer solcher Kurzzeitspektren

angewendet wird. Das dadurch bewirkte Informationsdefizit bei der phonetischen Klassifikation bewirkt, daß die Worterkennung hier immer eine höhere Fehlerrate hat als etwa die Einzelworterkennung. Insbesondere im Bereich der mit großen zeitlichen Veränderungen von Lautstärke und Spektrum verbundenen Konsonanten ist die phonetische Klassifikation hier notorisch unsicher.

Man könnte nun versuchen, die eliminierte zeitliche Spektralinformation durch geeignete Operationen zu kondensieren, damit zusätzliche Musterkomponenten zu generieren und so den zeitlichen Verlauf direkt in die Klassifikation einzubeziehen. So könnte man etwa Nasale, Plosive - und hier die Unterscheidung zwischen stimmhaft und stimmlos - und andere zeitlich determinierte phonetische Klassen exakter bestimmen. Da aber die so gewonnene Information im allgemeinen nicht mit den anderen Komponenten des Mustervektors commensurabel ist, wird sie nicht in die eigentliche Musterklassifikation einbezogen, sondern durch einen eigenen Satz von - meistens vom Programmierer ad-hoc aufgebauten - Regeln bewertet. Wenn auch die zunächst eliminierte Information über den zeitlichen Verlauf der Spektren in Form von heuristischen Regeln in der Prä- und Postsektion mit dem Klassifikator verbunden wird, kann sie nie ganz genutzt werden.

In diesem Bericht soll ein neuer Algorithmus, das Verfahren des differentiellen Lernens, kurz DL-Verfahren, auf die phonetische Segmentation angewandt werden/8/. Er ermöglicht

- kurze Rechenzeiten durch Konzentration auf die jeweils relevanten Teile des Musters und damit
- Echtzeit-Klassifikation
- Implementation auf einem Mikrorechner
- Erweiterbarkeit auf beliebige zusätzliche Merkmale.

Der Algorithmus wurde im Rahmen eines Systems zur Erkennung kontinuierlich gesprochener, aneinander geketteter Teilsätze erprobt.

2. Das Verfahren des differentiellen Lernens

2.1 Das Prinzip des Algorithmus

Es sei $\bar{x}$ der zu klassifizierende Mustervektor aus n Komponenten oder Merkmalen. Dann besteht jeder Klassifikator aus einer skalaren Entscheidungsfunktion derart, daß

$$(2.1) \qquad f_{lm}(\bar{x}) \begin{cases} \geq 0 & \bar{x} \Longrightarrow K_l \\ < 0 & \bar{x} \Longrightarrow K_m \end{cases}$$

$\bar{x}$ je nach dem Ergebnis der Klasse K_l oder K_m zugewiesen wird. (Das Zeichen $\Longrightarrow$ steht hier für: "Wird zugewiesen zur Klasse...."). In dem n-dimensionalen Musterraum definiert $\bar{x}$ einen Punkt, $f_{lm}(\bar{x}) = 0$ eine Hyperfläche, die den Musterraum in zwei Halbräume teilt, die jeweils einer Klasse zugeordnet sind. Die einfachste überhaupt mögliche Entscheidungsfunktion nach (2.1) wäre

$$(2.2) \qquad x_i - b \begin{cases} \geq 0 & \bar{x} \Longrightarrow K_l \\ < 0 & \bar{x} \Longrightarrow K_m \end{cases}$$

wo x_i eine Komponente von $\bar{x}$ ist und b eine skalare Konstante.

Statt nun, wie sonst in der Theorie der Musterklassifikation, die Funktion $f_{lm}(\bar{x})$ auf eine möglichst effektive Unterscheidung der Klassen zu optimieren, geht das DL-Verfahren von der Einfachstfunktion (2.2) aus. Die Vorgehensweise ist die folgende:

Gegeben seien 2 Mustervektoren $\bar{x}^l$ und $\bar{x}^m$, von denen a priori bekannt ist, daß sie zur Klasse K_l bzw. K_m gehören.
Dann wird bei $\bar{x}^l$ die Komponente gesucht, für die sich

$$\max_j \left(| x_j^l - x_j^m | \right) \text{ ergibt} \tag{2.3}$$

Der Index dieser Komponente sei i.
Dann wird

$$b = \frac{x_i^l + x_i^m}{2} \tag{2.4}$$

und für die Klassifikation weiterer $\bar{x}$ sind dann nur die Werte von i und b für die Abfrage nach 2.2 erforderlich. Die i-te Komponente, die den größten Unterschied zum Muster der anderen

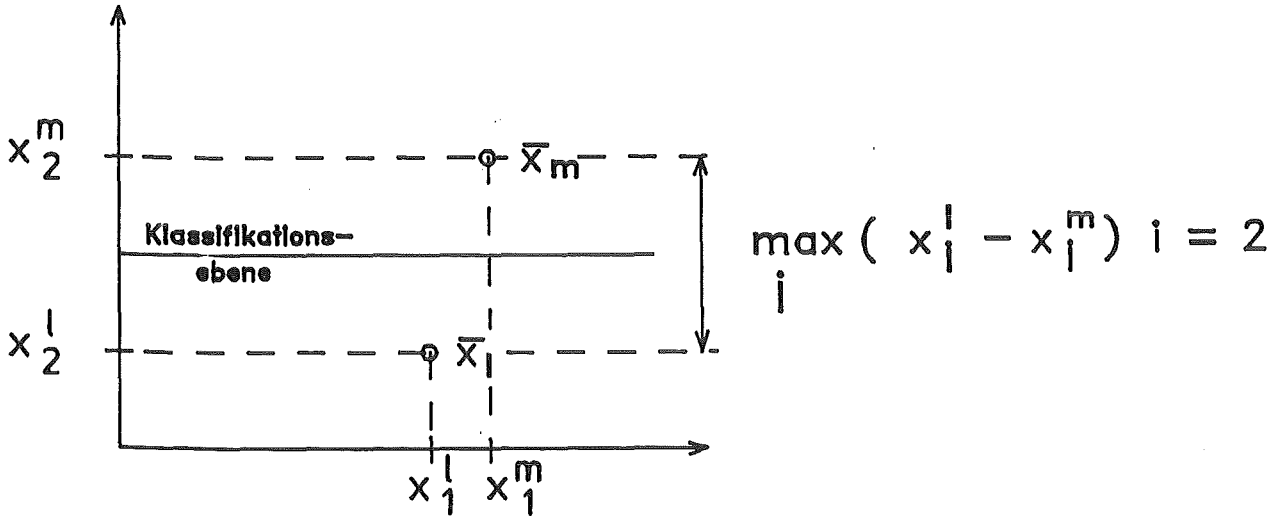

Bild 1: Differentielles Lernen für einen zweidimenionalen Mustervektor

Klasse ergibt, wird relevante Komponente genannt. Weil diese im Lernvorgang durch Differenzbildung nach (2.3) gefunden wird, wurde dem ganzen Verfahren der Name "Differenzielles Lernen" gegeben. Die Klassifikation nach (2.2) geschieht, wie Bild 1 für den zweidimensionalen Fall veranschaulicht, durch eine Hyperebene, die auf der x_i-Achse senkrecht steht und von den Mustern $\bar{x}^l$ und $\bar{x}^m$ in der i-Richtung den gleichen Abstand hat.

Es ist nicht zu erwarten, daß dieser einfache Algorithmus für die korrekte Klassifikation mehrerer schwankungsbehafteter Muster ausreicht. Deshalb wird das Verfahren bei jeder Fehlklassifikation, auch bei jeder neu zu lernenden Klasse wiederholt, und die Einzelabfragen werden als Baumstruktur verkettet.

Bild 2 zeigt ein Beispiel für eine solche Struktur. Als erstes werde etwa durch ein b_1 für die Komponente i_1 ein Kriterium für die Unterscheidung der Phoneme a und e gefunden. Bei dessen Erfüllung werde gelegentlich o als a fehlklassifiziert, so daß ein neues Kriterium mit i_2 und b_2 nötig wird usw. Bei der Klassifikation wird solange von Abfrage zu Abfrage fortgeschritten, bis ein terminaler Knoten erreicht wird. Dabei kann durchaus, wie etwa beim o, ein Phonem auf beiden Seiten einer Verzweigung stehen.

Das nacheinander erfolgende Abfragen der Kriterien entspricht konjunktiv verknüpften Regeln über die Größe einzelner Komponenten. So gilt nach Bild 2 $\bar{x} \Rightarrow$ 'a' dann, wenn
$(x_{i1} > b_1) \wedge (x_{i2} < b_2)$.

Bei Vorkommen hinter mehreren Verzweigungen:

$$\bar{x} \Rightarrow \text{'o' dann, wenn } (x_{i1} > b_1) \wedge (x_{i2} \geq b_2) \vee (x_{i1} \leq b_1) \wedge (x_{i3} \geq b_3) \wedge (x_{i4} < b_4)$$

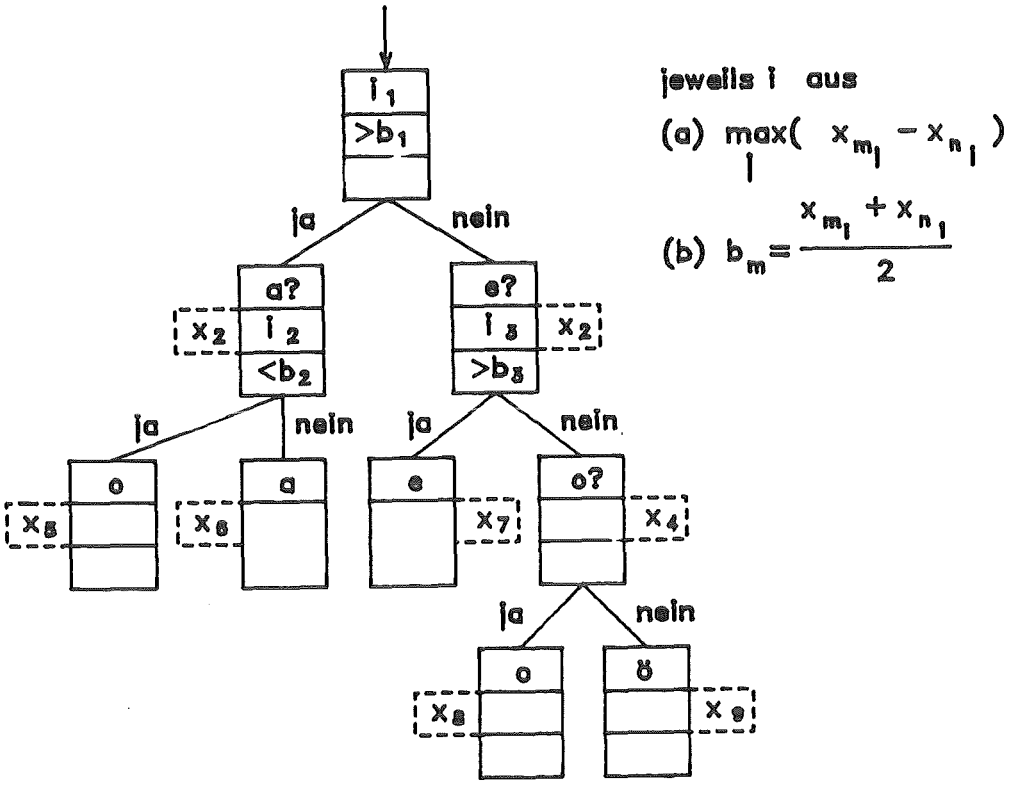

Bild 2: Prinzip der Verkettung von Entscheidungskriterien nach (2.2)

Man kann leicht erkennen, daß auf diese Weise im Prinzip ein beliebig detaillierter Satz von Regeln erstellt werden kann, der die einzelnen Klassen voneinander abgrenzt. Für die Definition einer neuen Klassengrenze sind nach (2.3) zwei Vektoren erforderlich. Einer von ihnen ist der input $\bar{x}$. Der andere muß in Verbindung mit dem letzten Knoten abgespeichert sein. Beispiel:

Für ein neues $\bar{x}$ gelte $x_i > b_1$. Der folgende Knoten sei terminal und ergäbe $\bar{x} \Longrightarrow$ 'a'. Ist diese Klassifikation falsch, so muß dem System nur gesagt werden z. B. $\bar{x} \Longrightarrow$ 'o'. Dann laufen die folgenden Operationen ab:

1. (2.3) mit $\bar{x}$ und $\bar{x}_2$
2. b_2 aus (2.4) eintragen
3. neue terminale Knoten 'o' und 'a' bilden
4. $\bar{x}$ als $\bar{x}_5$, $\bar{x}_2$ als $\bar{x}_6$ eintragen.

Das vollständige Muster, das zur Definition einer neuen Klasse führt, muß nur für den nächsten Lernschritt aufbewahrt werden, für die Klassifikation wird es nicht benötigt.

Entscheidend ist dabei, daß die vollständigen Referenzmuster nur für den Lernprozeß aufbewahrt werden müssen, beim Klassifikationsvorgang aber nur einzelne Komponenten abgefragt werden müssen. Das sollte kurze Rechenzeiten erlauben.

2.2 Kritische Fragen

Wegen der Möglichkeit, praktisch unbegrenzt neue Regeln konjunktiv und disjunktiv zu verknüpfen und dabei jeweils die am stärksten differenzierende Komponente dem Lernen einer neuen Regel zugrundezulegen, ist die prinzipielle Eignung eine solchen Klassifikators auch für kompliziert geformte Klassengrenzen evident.

Nicht so klar ist die Beantwortung der Frage, ob dies auch im Rahmen einer praktisch vertretbaren Zahl von Schritten möglich ist. Damit hängen die folgenden Fragen zusammen:

1. Offensichtlich ist die sich ergebende Struktur von der Reihenfolge abhängig, in der beim Lernprozeß die einzelnen Klassenvertreter eingebracht werden. Da es hier keine a priori-Kriterien gibt, ist diese Reihenfolge praktisch zufällig. Ergeben sich daraus unvertretbar hohe Abfragzahlen, weil Klassengrenzen zu oft korrigiert werden müssen?

2. Können Fehler des Lehrers in der Klassenzuweisung korrigiert werden?

3. Da sich jeder Lernschritt nur auf eine Komponente des Musters bezieht - wird die Lernphase zu lang?

4. Im Gegensatz zu anderen Klassifikatoren wird hier ein und nur ein terminaler Knoten erreicht. Es wird also auch ein und nur

ein Klassifikationsergebnis geliefert, statt mehrerer Ergebnisse mit unterschiedlichen Bewertungsziffern. Welchen Einfluß hat das auf die nachfolgende Wort- und Satzerkennung?

Die Beantwortung dieser Fragen kann nur experimentell erfolgen, und es wurde deshalb an einen nach dem DL-Verfahren arbeitenden Phonetik-Klassifikator/Segmentierer ein Wort- und Satzerkennungssystem angeschlossen und zunächst für einen Sprecher getestet.

3. Implementation eines Spracherkennungssystems

Bild 3 gibt ein Übersichtsschema über das System. Es wird durch Sprache gesteuert und kann nach Verarbeitung eines Satzes sofort

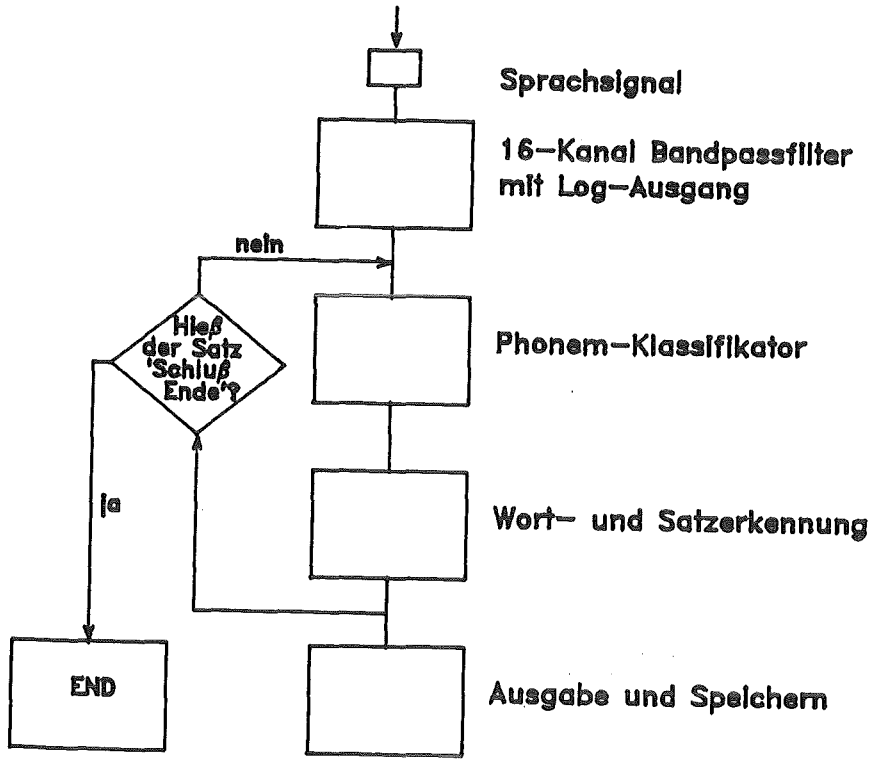

Bild 3: Übersichtsschema des Spracherkennungssystems

den nächsten aufnehmen bis es durch das gesprochene Kommando 'Schluß Aus' abgeschaltet wird. Der Rechner arbeitete mit dem INTEL-80286 Mikro-Prozessor, alle Programme wurden in PASCAL erstellt. Die einzelnen Bereiche werden in den folgenden Abschnitten genauer beschrieben.

3.1 Akustische Analyse

Die Eingabe erfolgt über ein Mikrofon, das mit einer "schiefen Nierencharakteristik" auf den Mund des Sprechers gerichtet und daher einigermaßen unempfindlich gegenüber Nebengeräuschen ist.

Da die Implementation auf einen Mikrorechner zugeschnitten sein sollte, wurde die Frequenzanalyse durch einen eigenen, mit Analog-Bandpaßfiltern aufgebauten 16-Kanal-Spektral-Analysator durchgeführt. Er liefert logarithmische, mit etwa 150 Hertz tiefpaßgefilterte Ausgangssignale. Die Logarithmierung reduziert den Einfluß von Lautstärkeschwankungen und den Unterschied des Outputs im unteren und oberen Frequenzbereich. Bild 4 zeigt die 16 Frequenzbänder, Bild 5 zwei typische Spektren.

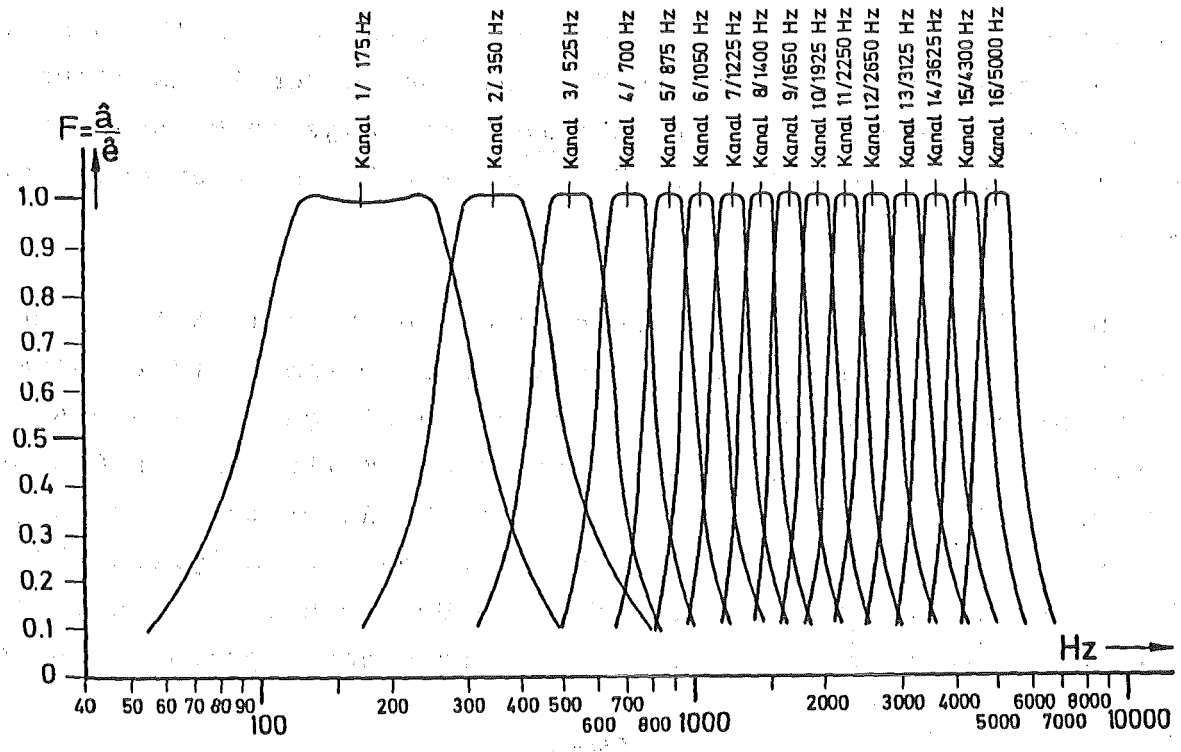

Bild 4: 16-kanalige Filterbank f. Spracherkennung angepaßt an die vom menschlichem Ohr gebildeten Frequenzgruppen

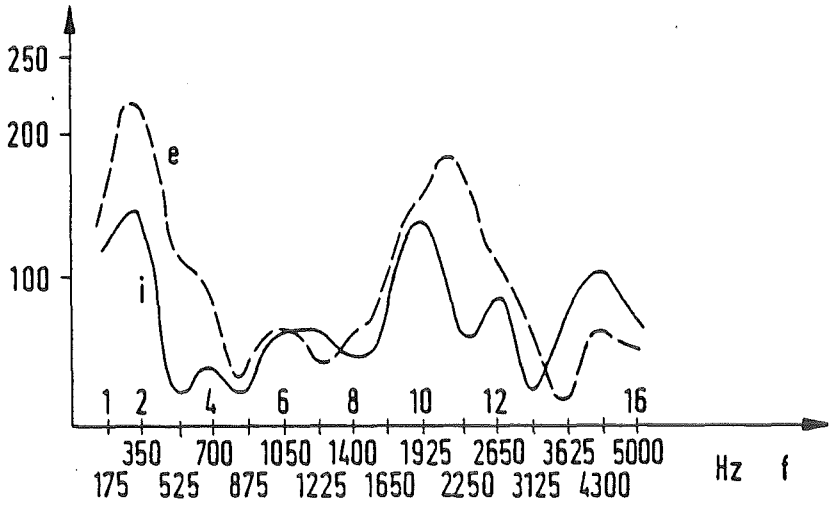

Bild 5: Spektrogramme der Laute e und i.
Man erkennt die typische Schwierigkeit, bei zusätzlichen Schwankungen die in weiten Teilen gleichen Spektren zu unterscheiden.

3.2 Phonetische Klassifikation

Der Rechner ruft die digitalisierten Spektraldaten alle 10 ms ab und legt sie, sobald die Lautstärke als Summe der 16 Kanalausgänge einen Schwellwert überschreitet, so lange im Rechner ab, bis ein weiterer Schwellwert für mehr als 300 ms unterschritten wird.

Danach werden die Daten nacheinander dem DL-Klassifikator zugeführt. Eine Präselektion nach den Kriterien stimmhaft/stimmlos ist nicht erforderlich, hat sich sogar in einigen Fällen als störend herausgestellt, da die allgemeinen Kriterien der Präselektion in Spezialfällen versagten. Der DL-Algorithmus erlaubte eine sehr viel präzisere Abstimmung der Regeln. Lediglich für die Detektion von Plosiven und für die Unterscheidung von p, t, k einerseits und b, d, g andererseits wurden noch Spezialoperationen

mit ad-hoc-Kriterien eingesetzt, die später in das DL-Verfahren einbezogen werden sollen. In der Weiterentwicklung sollen auch diese Entscheidungen in den DL-Ablauf eingebunden werden.

Die Klassifikation erfolgt nach einzelnen Phonemen, wie sie in Tab. 1 zusammengestellt sind. In einer Liste werden nacheinander alle Phoneme eingetragen, die mindestens über 3 der 10 ms-Takte angedauert haben, dazu ihr Anfangs- und Endtakt. Diese Tabelle ist Ausgangspunkt der nachfolgenden Satzanalyse.

Für das Lernen des Phonemklassifikators wird ein Satz von etwa 40 zweisilbigen Wörtern benutzt, die so ausgewählt sind, daß sie alle Phoneme enthalten und eine automatische Segmentation nach dem Verlauf der Summe über die 16 Kanäle ermöglichen. Danach müssen einige Phonemkombinationen durch handsegmentierte Lernworte noch nachgearbeitet werden, doch beschränkt sich dieser Aufwand für einen Sprecher auf etwa 20 weitere Fälle.

Wenn die gespeicherten Spektraldaten in die Phonemliste umgesetzt sind, ist die phonetische Klassifikation abgeschlossen und die Satzerkennung beginnt.

3.3 Satzanalyse

Während die phonetische Analyse als reines bottom-up-Verfahren arbeitet, funktioniert die Satzanalyse als top-down-Verfahren. Es werden mögliche und im Kontext in den Satz gehörende Phoneme mit den in der Phonemliste aufgezeichneten verglichen. Werden sie an der richtigen Stelle angetroffen, so wird ein Zähler erhöht. Am Ende wird der Satz mit dem höchsten Zählerbetrag als Ergebnis geliefert.

Phoneme	Aussprache wie
a	vater, halle
e	leben
(ä)	essen
i	liebe
I	Schiff
o	boot
O	offen, vater
u	rufen
V (ö)	löwe
/ (ü)	ypern
f	fahne
s	wasser
S	esche
x	echo
w	ewig
m	oma
n	ana
t	eta
k	ekel
p	opa
d	eden
g	ego
b	abel
z	besen
E	schwa

Tab. 1 Verwendete Phoneme

Das hier entwickelte Satzerkennungssystem sollte lediglich zur Erprobung des DL-Verfahrens zur Phonemklassifikation dienen. Es ist deshalb einfach gehalten und nicht so weitgehend optimiert wie andere Systeme (z. B. /5, 9, 10, 11/). Wie bei HARPY /1/, aber auf Phonemsymbole als Inputs ausgerichtet, sind Syntax und Semantik durch die Existenz vorgegebener, wenn auch fast beliebig erweiterbarer Sätze festgelegt.

Die Sätze sind in Teilstücken in den Knoten eines Netzwerks angeordnet. Die Erkennung des ersten Teilsatzes führt automatisch zum Weiterschalten auf den nächsten Knoten, in dem mit anderen Konkurrenten der nächste Teilsatz zur Verfügung steht. Solche Netzwerke sind nicht neu, nur werden üblicherweise den Knoten Worte und nicht Wortketten bzw. Teilsätze zugeordnet. Es war aber dadurch möglich, die Koartikulationsprobleme zwischen meist ohnehin gemeinsam vorkommenden Wortgruppen (3. Artikel und Substantiv) phonetisch fest zu erfassen. Bild 6 zeigt einen vereinfachten Netzwerkausschnitt.

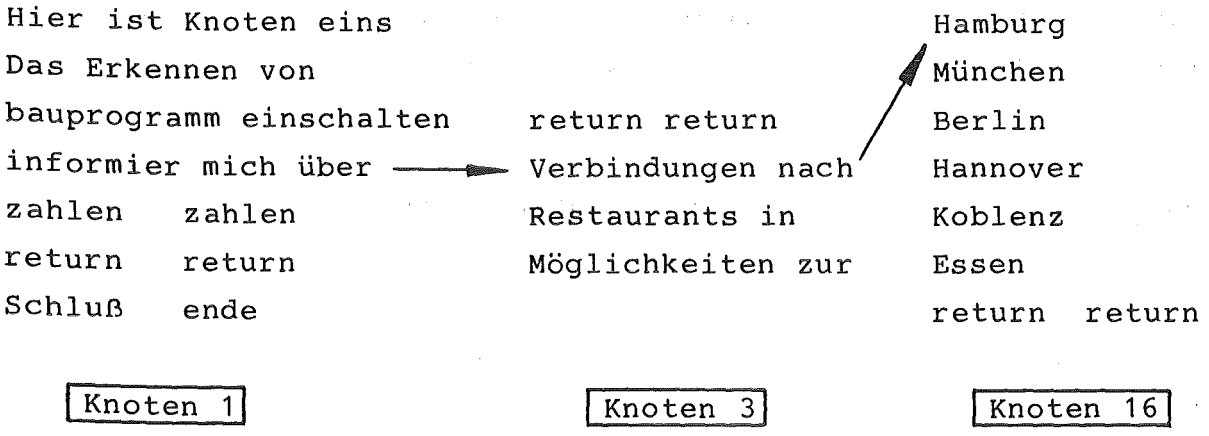

Bild 6: Ausschnitt aus dem Satz-Netzwerk

Die Teilsätze sind durch Zeiger mit ihren Nachbarn im gleichen Knoten und mit Folgeknoten verknüpft. Die zu einem Teilsatz eines Knotens gehörenden Worte sind getrennt und nur über Zeiger verbunden gespeichert. Einzelne Worte sind also die kleinsten Untereinheiten. Durch Sprechen von "return, return" kann man aus jedem Knoten zum Knoten 1 zurückkehren, von dem aus durch "Schluß, Ende" die Sitzung beendet werden kann.

Bei jedem Wort steht seine phonetische Transskription. Diese ist silbenweise strukturiert, wobei die durch Konsonantengruppen voneinander getrennten Vokale als Silbenkerne und Fixpunkte dienen. Deshalb muß die phonetische Klassifikation als erstes eine zuverlässige Vokal-Konsonant-Unterscheidung liefern. Kommt es hierbei zu Fehlern, so kann leicht die richtige Silbenzuordnung gestört werden. Maßnahmen zur Unterdrückung solcher Fehler durch Spreizung (z. B. /11/) und andere gegenseitige Verschiebungen sind bisher nicht eingesetzt worden.

Das phonetische Klassifikationssystem liefert für jedes Segment nur ein einziges Ergebnis im Gegensatz zu den bewerteten Alternativen anderer Systeme. Es zeigte sich aber, daß Fehler in der Phonemklassifikation fast immer ein Phonem lieferten das der gleichen Laut-Oberklasse angehörte (s. Tab. 2).

Phoneme	Lautoberklasse
a, O, E	1
e, I, ä, ö, i	2
o, u	3
f, s, x, S	4
m, n, l	5
p, k, t, b, g, d	6

Tab. 2: Zuordnung der Phoneme zu Lautoberklassen

Deshalb wird die Satzerkennung in zwei Durchläufen durchgeführt. Im ersten Durchlauf wird die zuvor in Silben aufgeteilte Phonemfolge in die entsprechenden Lautoberklassen transformiert und mit den im gegenwärtig bearbeiteten Knoten gegebenen Worten verglichen und bewertet. Alle 2 Silben werden die Varianten, deren Bewertungszähler zu gering ist, eliminiert. Dadurch wird nach Art eines beam-search-Verfahrens die Zahl der jeweils bearbeiteten Kandidaten begrenzt. Die verbleibenden Sätze werden weiter vervollständigt bis ihr jeweiliges Ende erreicht wird.

Nur die so vorab ausgewählten Sätze gelangen in den zweiten Durchlauf und werden nun mit den Phonemen selbst verglichen und verifiziert.

4. Ergebnisse

Zur ersten Erprobung des DL-Systems wurde die Spracheingabe nur durch einen Sprecher vorgenommen. Erst bei Hinzunahme weiterer Vorverarbeitungsoperationen bzw. Merkmale in einem nächsten Entwicklungsschritt erschien eine Lösung des Mehrsprecherproblems sinnvoll zu sein.

Mittels 40 automatisch segmentierbaren Trainingsworten und etwa 20 weiteren handsegmentierten Worten wurde zunächst die Information für die Phonemklassifikation eingebracht. Es zeigte sich, daß hierbei für jedes Phonem im Durchschnitt 4 bis 5 Abfragen auf die Größe einer Komponente des Spektralvektors genügte, deutlich weniger also als die 16 vorhandenen Komponenten, die in einem klassischen Klassifikationsverfahren ja alle einbezogen werden. Einzelne Fehler beim Lernen ließen sich nachträglich durch zusätzliche Lernschritte und Regeln überdecken.

Für die Satzerkennung wurden etwa 30 Sätze mit zusammen ungefähr 150 Worten eingegeben. Um sicherzustellen, daß auch wirklich der

ganze Satz in den Erkennungsprozeß einbezogen wurde, wurden nicht nur Sätze, wie:

"informier mich über"
"hier Knoten eins"

etc. als Konkurrenten im gleichen Knoten gegenübergestellt, sondern auch

"fahr nach rechts"
"fahr nach links"

mit unterschiedlichen Phonemen erst am Satzende.

Unter diesen Bedingungen wurden im Durchschnitt 72 % der Sätze richtig erkannt. Ursachen für Fehlklassifikationen waren:

- Fehlende Zeitanpassung (Spreizung) bei der Satzerkennung (Schwächen des Satzerkennungsalgorithmus 64 %
- Fehler in der phonetischen Klassifikation - Lautstärkeabhängigkeit 28 %
- Sonstige 8 %

Der erste Punkt betrifft nicht das DL-Verfahren, sondern den einfachen Worterkennungsalgorithmus. Das Problem der Lautstärkeabhängigkeit der Phonemklassifikation ist ein allgemeines Problem aller Klassifikatoren und kann solange nicht ganz vermieden werden wie die Amplituden der Spektralkanäle direkt und unbearbeitet in die Klassifikation eingehen. Wie im nächsten Abschnitt dargestellt, bietet gerade das DL-Verfahren die Möglichkeit, rationell Transformationen der Rohdaten vorzunehmen, die die Klassifikation verbessern.

Die Rechenzeiten je Teilsatz liegen bei 0,6 bis 1 Sekunde, worin eine Wartezeit von 300 ms enthalten ist, die nach Absinken des

Signals unter einen Grenzwert noch die Möglichkeit für ein eventuell auftretendes Plosiv lassen.

Diese sonst wohl nicht mit Mikrorechnern erreichten Geschwindigkeiten - also praktisch eine Echtzeitverarbeitung - sind eines der wichtigsten Ergebnisse der Untersuchung. Dazu sei nochmals darauf hingewiesen, daß alle Programme in PASCAL geschrieben sind.

5. Weitere Entwicklung

Das DL-Verfahren arbeitet im Prinzip als ein System, das im Lernmodus sich jeweils die bestgeeigneten Parameter für die Klassifikation heraussucht und alle anderen nicht verwendet. Kriterium ist dabei die Größe der Differenz $P_i - P_j$ zum entsprechenden Parameter des falschen Klassifikationsergebnisses.

Man kann diese Vorgehensweise verallgemeinern und weitere abgeleitete Eingangsparameter erzeugen, wie etwa die Lage der Formaten, zeitliche Veränderungen der einzelnen Spektralkanäle und anderes mehr. Da man nun aber Parameter bekommt, die nicht mehr notwendig kommensurabel sind, muß ein verbessertes Auswahlkriterium gefunden werden. Eine Möglichkeit ist hier die Einführung einer bezogenen Parameterdifferenz $\frac{P_i - P_j}{P_i}$ (wobei $P_i = 0$ ausgeschlossen wird, praktisch auch nicht auftritt).

Das System würde sich dann unter einer größeren Zahl von Parametern und Parameteroperationen selbständig die mit dem größten Abstand und ggf. der größten Invarianz heraussuchen.

Am deutlichsten ist die Notwendigkeit neuer Klassifikationsregeln für die Plosivlaute zu erkennen, aber auch Nasale und Glissale mit ihren charakteristischen Übergängen sind noch weit von einer zufriedenstellenden Erkennbarkeit entfernt. Es gibt interessante Erkenntnisse aus Untersuchungen an Kleinkindern /12/, wo das Erkennen kurzer Verzögerungen im Einsetzen der

tiefen Frequenzen als charakteristisch für die Unterscheidung stimmhafter und stimmloser Plosive entscheidend ist, und weitere Erkenntnisse sind noch zu erwarten. Daraus müssen, möglichst automatisch, neue Regeln abgeleitet und einbezogen werden.

Es wird sich hier vermutlich auch zeigen, daß die üblicherweise 10 ms und mehr andauernden Taktzeiten zu lang sind. Bei kürzeren Taktzeiten wird die Plosivklassifikation verbessert.

Das DL-Verfahren, das im Klassifikationsprozeß immer nur die relevanten Musterkomponenten verwendet, bietet gute Möglichkeiten, die phonetische Klassifikation auch unter solchen Bedingungen noch im Echtzeitbereich zu bleiben.

Erst damit wäre auch eine sinnvolle Basis gegeben, das Mehrsprecherproblem zu lösen.

6. Schlußfolgerungen und Ausblick

Das DL-Verfahren hat sich für die phonetische Klassifikation aus 16 Spektralkomponenten als arbeitsfähig und schnell erwiesen. Die in Abschnitt 2.2 gestellten Fragen lassen sich zufriedenstellend beantworten. Es ist nun möglich, die Erweiterbarkeit auf zusätzliche abgeleitete Merkmale zu untersuchen und ggf. die phonetische Klassifikation zu verbessern. Das DL-Verfahren arbeitet nicht mit den Mustern als solchen, sondern mit den Unterschieden zwischen ihnen und ist damit verschieden von anderen Klassifikatoren. Der Prozeß, in dem die Unterscheidungsfähigkeit zwischen gegebenen Mustern aufgebaut wird, entspricht einer zunehmenden Differenzierung. In der dadurch entstehenden Baumstruktur stehen die generellen Muster näher an der Wurzel als die mehr differenzierten. Wenn ein bestimmtes Muster klassifiziert wird, werden im Verarbeitungsprozeß seine Generalisierungen durchlaufen. In der Psychologie ist die Fähigkeit und besonders bei Kindern auch die Tendenz zur Generalisierung und Differenzierung in der Mustererkennung vielfach belegt. Sie spielt für die jeder

Maschine überlegene Fähigkeit des Menschen (und vieler anderer Lebewesen) zur Mustererkennung und sicher auch für die Entwicklung intelligenter Ordnungskriterien in der Speicherung der Information eine wesentliche Rolle. Sicherlich sind die Differenzierungsregeln dabei komplexer als im DL-Verfahren. Aber wenn schon mit so einfachen Methoden eine wirksame Klassifikation erreichbar ist, sollte eine Weiterentwicklung interessante Möglichkeiten eröffnen.

Literatur

/ 1/ B. Lowerre, R. Reddy, The HARPY Speech Understanding System, in Trends in Speech Recognition, W.A. Lea, (ed.), Prentice Hall, Englewood Cliffs, N.J., 1979, S. 340-360

/ 2/ G. Gill, H. Goldberg, R. Reddy, B. Yegnanarayana, A Recursive Segmentation Procedure for Continuous Speech Carnegie-Mellon-University CMU-dCS-78-134, 1978

/ 3/ S.E. Levinson, A.E. Rosenberg, Some Experiments with a Syntax Directed Speech Recognition System, 1978 IEEE Int. Conf. on Acoustics, Speech & Signal Procession, Tulsa, OKL., 1978, S. 700-703

/ 4/ L.B. Bahl, R. Bakis, P.S. Cohen, A.G. Cole, F. Jelinek, B.L. Lewis, R.L. Mercev, Recognition Results with Several Experimental Acoustic Processors, 1979 IEEE Int. Conf. on Acoustics, Speech & Signal Processing Washington, 1979, S. 249-251

/ 5/ H. Niemann, A. Brietzmann, H.W. Hein, H.R. Mühlfeld, P. Regel, G. Schukat, A System for Understanding Continuous German Speech, Information Sciences, Vol. 3, p. 87 (1984)

/ 6/ Bildung von Lauthypothesen mit Polynomklassifikatoren und Markovmodellen in einem System zur Erkennung kontinuierlicher Sprache, 7. DAGM-Symposium, S. 229, Erlangen, Sept. 1985, Springer 1985

/ 7/ A. Noll, Explizite Segmentierung kontinuierlicher Sprache, 7. DAGM-Symposium, S. 234, Erlangen 1985, Springer, 1985

/ 8/ D. Smidt, unveröffentlichter Bericht (1985)

/ 9/ F. Jelinek, Continuous Speech Recognition by Statistical Methods, Proc. of the IEEE, Vol. 64, pp. 532-556, 1976

/10/ A. Brietzmann, Eine Netzwerk-Grammatik des Deutschen für die Automatische Spracherkennung. Sprache und Datenverarbeitung, W. Lenders u.a. Hrsg., 1-2/84, S. 54, Saarbrücker Druckerei u. Verlag, 1984

/11/ E.G. Schukat-Talamazzini, S. Heunisch, Schnelle Präselektion von Wörtern aus kontinuierlich gesprochener Sprache, 7. DAGM-Symposium, Erlangen, Sept. 1985, Springer, 1985, S. 170-174