# Künstliche neuronale Netzwerke zur adaptiven Geräuschreduktion für robuste Spracherkennung

Zur Erlangung des akademischen Grades eines

**DOKTOR-INGENIEURS**

von der Fakultät für
Elektrotechnik
der Universität Fridericiana Karlsruhe

genehmigte

**DISSERTATION**

von

**Dipl.-Ing. Michael Trompf**
aus St. Georgen i. Schw.

| | |
|---|---|
| Tag der mündlichen Prüfung: | 2. Mai 1995 |
| Hauptreferent: | Prof. Dr.-Ing. K. Kroschel |
| Korreferent: | Prof. Dr. Alexander Waibel |

# DANKSAGUNG

Die vorliegende Arbeit entstand während meiner Tätigkeit im Fachbereich für *Signal Processing Applications* im Forschungszentrum der Alcatel SEL AG in Stuttgart. Meiner Fachbereichsleiterin Frau Dr. Heidi Hackbarth danke ich für ihre großzügige Unterstützung mit einer flexiblen Arbeitszeitregelung, ohne die diese Arbeit nicht hätte durchgeführt werden können.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Kristian Kroschel für sein großes Interesse an diesem Thema, die engagierte Betreuung der Arbeit sowie die Übernahme des Hauptreferats.

Herrn Prof. Dr. Alexander Waibel danke ich für seine wertvollen Anregungen und für die Übernahme des Korreferats.

Weiterhin möchte ich meinen Kollegen sowie den zahlreichen Diplomanden und Praktikanten danken, die in vielfältiger Weise zum Gelingen dieser Arbeit beigetragen haben. Dabei gilt mein besonderer Dank Herrn Dipl.-Ing. Harald Eckhardt für die softwaretechnische Betreuung des Simulationssystems sowie die Wartung meiner Workstation.

Besonders verbunden fühle ich mich Herrn Dipl.-Ing. Wolfgang Bisch, der mit wertvollen Korrekturhinweisen dazu beitrug, die Verständlichkeit dieser Arbeit zu erhöhen.

Teile der vorliegenden Arbeit wurden im Rahmen des Verbundvorhabens *Verbmobil* vom Bundesministerium für Forschung und Technologie (BMFT) unter dem Förderkennzeichen 01 IV 102 I2 unterstützt. Die Verantwortung für den Inhalt dieser Arbeit liegt beim Autor.

Stuttgart, im Januar 1996

Michael Trompf

# INHALTSVERZEICHNIS

# FORMELZEICHEN UND SCHREIBWEISEN

**Formelzeichen** sind kursiv gedruckt und wurden nach folgenden Regeln gewählt:

- Vektoren sind in fetter, Koeffizienten von Vektoren und Variablen in normaler Schriftstärke gedruckt.
- Matrizen sind groß und fett und Elemente von Matrizen klein und in normaler Schriftstärke gedruckt.

| | |
|---|---|
| $a$, $a(l,n)$ | Überschätzfaktor |
| $a_j$ | $j$-ter Prädiktorkoeffizient |
| $\boldsymbol{a}$ | Eigenvektor |
| $A(l)$, $A(l,n)$ | Frequenzgang des (verallgemeinerten) Wiener Filters |
| $\boldsymbol{A}$ | Abbildungsmatrix, Gewichtsmatrix |
| $b_1$, $b_2$ | Konstanten |
| $\boldsymbol{B}$ | Gewichtsmatrix |
| $c$ | Spectral Floor |
| $\boldsymbol{c}$, $c_q$ | Koeffizienten des Cepstrums |
| $C_k$ | Kostenfaktor für den $k$-ten Ausgangskoeffizienten |
| $C(\boldsymbol{e})$ | Kostenfunktion des quadratischen Fehlers |
| $C(z)$ | $z$-Transformierte des Cepstrums |
| $\boldsymbol{C}$ | Gewichtsmatrix |
| $\boldsymbol{D}$ | Gewichtsmatrix |
| $e(i)$, $e(n)$ | Fehlersignal |
| $E$ | Energie |
| $E_{lg}$ | logarithmierte Energie |
| $f_a$ | Abtastrate |
| $f_g$ | Grenzfrequenz |
| $f_w$ | Wortfehlerrate |
| $f(\boldsymbol{x})$ | Aktivierungsfunktion |
| $f_{\boldsymbol{x}}(\boldsymbol{x})$ | Wahrscheinlichkeitsdichtefunktion von $\boldsymbol{x}$ |
| $f_{\boldsymbol{x}}(\boldsymbol{x}\,\vert\,\boldsymbol{y})$ | bedingte Wahrscheinlichkeitsdichtefunktion von $\boldsymbol{x}$ gegeben $\boldsymbol{y}$ |

| | |
|---|---|
| $f_{x,y}(x,y)$ | Verbundwahrscheinlichkeit von $x$ und $y$ |
| $F(x)$, $F(x)$ | Übertragungs- oder Abbildungsfunktion |
| $F_B(x)$ | Systemfunktion des Bayes-Schätzsystems |
| $g(x)$ | Aktivierungsfunktion |
| $G$ | Verstärkungsfaktor; Geräuschparameter; Zahl der Gewichte |
| $h$ | Zahl der verdeckten Knoten |
| $\boldsymbol{h}$ | Aktivierungen der verdeckten Knoten |
| $H(z)$ | Übertragungsfunktion |
| $i$ | diskrete Zeitvariable; Laufvariable; Knotenindex |
| $i_{kl}$ | Ausgangserregung des Knotens $kl$ |
| $\boldsymbol{I}$ | Einheitsmatrix |
| $j$ | Laufvariable; Index für lpc- und cepstrum-Koeffizienten; Knotenindex |
| $J_m$ | Zahl der Vektoren zur Berechnung der $m$-ten Ableitung |
| $k$ | Laufvariable; Wort- bzw. Pausenindex; Knotenindex |
| $K_i$ | $i$-ter Kandidatenknoten |
| $k1,k2$ | Indizes für Knoten 1 bzw. 2 |
| $l$ | diskrete Frequenz; Index für Trainingsvektor |
| $l_{ij}$ | Korrelation zwischen den Koeffizienten $i$ und $j$ |
| $lr$, $lr(n)$ | Lernrate |
| $L$ | Zahl der Trainingsvektorpaare |
| $L_N(l)$ | Leistungsdichte des Geräuschsignals |
| $L_R(l)$ | Leistungsdichte des gestörten Sprachsignals |
| $m$ | Zahl der Eingangsknoten; Ordnung der Ableitung; Koeffizientenzahl von $x$ und $v$ |
| $m_o(l)$ | oberer spektraler Moment |
| $m_S(l)$ | Schwerpunkt des komplexen Kurzeitspektrums |
| $m_t$ | Momentum Term |
| $m_u$ | unterer spektraler Moment |
| $M$ | akustisches Ereignis; Zahl der Spektrallinien |
| $MSE$ | mittlerer quadratischer Fehler |
| $MSE_{lin}$ | mittlerer quadratischer Fehler eines linearen Netzwerks |
| $MSE_{nl}$ | mittlerer quadratischer Fehler eines nichtlinearen Netzwerks |
| $MSE_{rel}$ | relativer quadratischer Restfehler |
| $n$ | Segmentindex; Iterationszahl; Koeffizientenzahl |

| | |
|---|---|
| $n(t)$, $n(i)$ | Geräuschsignal |
| $\boldsymbol{n}(k)$, $\boldsymbol{n}_k$ | Geräuschparameter im $k$-ten Segment |
| $net$ | Erregungspotential |
| $N$ | Zahl der Abtastwerte |
| $N_{i,k}$ | Eingangserregung durch Geräuschparameter |
| $N_K$ | Zahl der Kurzzeitspektren |
| $N_{Knoten}$ | Zahl der Knoten im Netzwerk |
| $N_{max}(l,n)$ | Schätzfunktion für das Geräuschspektrum |
| $o$ | Laufvariable; Analyseordnung; Zahl der Kandidatenknoten |
| $\boldsymbol{p}$, $p_i$ | Koeffizienten aus der Hauptachsentransformation |
| $p(i)$ | Pausensignal |
| $P$ | Zahl der Pausen |
| $P(k)$ | Zahl der Pausensegmente |
| $P_m(j,J)$ | $m$-tes orthogonales Polynom |
| $q$ | Koeffizientenzahl von $t$; Quefrency-Variable |
| $R$ | Risiko; Zahl der installierten Kandidatenknoten |
| $R(l)$ | Spektrum von $r(i)$ |
| $R_i$ | $i$-ter Regressionskoeffizient |
| $r$ | Laufvariable; Dimensionalität von $\boldsymbol{n}$; Index für Kandidatenzyklen |
| $r_w$ | Worterkennungsrate |
| $r(t)$, $r(i)$ | geräuschbehaftetes Sprachsignal |
| $SNR(k)$ | lokales Signal-zu-Rausch-Verhältnis, wortbezogen |
| $SNR_{Seg}(n)$ | lokales Signal-zu-Rausch-Verhältnis, segmentbezogen |
| $SNR_{soll}$ | Sollwert für das Signal-zu-Rausch-Verhältnis |
| $s(t)$, $s(i)$ | Sprachsignal |
| $\tilde{s}(i)$ | Ausgangssignal des LPC-Filters |
| $S(l)$ | Spektrum von $s(i)$ |
| $S(z)$ | $z$-Transformierte von $s(i)$ |
| $t$ | kontinuierliche Zeitvariable |
| $\boldsymbol{t}$ | Zielvektor |
| $T$ | Zeitdauer zwischen zwei Abtastwerten |
| $T_n$ | Zeitdauer zwischen zwei Merkmalsvektoren |
| $T_r$ | Zeitdauer des Regressionsfensters |

| | |
|---|---|
| $u(i)$ | Eingangserregung des Vocoders |
| $\boldsymbol{u}$ | Mittelwertsvektor |
| $u_{k1,k2}$ | Gewicht zwischen zwei Knoten $k1$ und $k2$ |
| $U(z)$ | $z$-Transformierte von $u(i)$ |
| $\boldsymbol{v}$ | erweiterter Eingangsvektor |
| $\boldsymbol{V}$, $v_{ij}$ | Gewichtsmatrix von der Zwischen- zur Ausgangsschicht |
| | Ausgangsaktivierungen der Kandidatenknoten |
| $w(i)$ | geräuschbehaftetes Sprachsignal während eines Wortes |
| $W$ | Gesamtzahl der Wörter |
| $\boldsymbol{W}$, $w_{ij}$ | Gewichte von der Eingangs- zur Zwischenschicht |
| $W(k)$ | Zahl der Segmente im $k$-ten Wort |
| $\boldsymbol{x}$, $x_i$ | geräuschbehafteter Merkmalsvektor; Eingangserregung |
| $x_0$ | Bias- oder Offseteingang |
| $\boldsymbol{y}$, $y$ | Ausgangserregung; Funktionswert |
| $z$ | Variable der $z$-Transformation |
| $\boldsymbol{z}$, $z_i$ | Ausgangserregungen der verdeckten Knoten |
| | |
| $\alpha$ | Gewichtungsfaktor, Parameter |
| $\beta$ | Glättungsfaktor; Quickprop-Lernrate |
| $\beta_N$, $\beta_R$ | Glättungsfaktoren |
| $\delta_{k2}$ | Fehler am Netzwerkausgang, zum Knoten $k2$ rückwärts gerechnet |
| $\varepsilon$ | Fehlerschwelle; Schwelle für den Abstand zweier Vektoren |
| $\gamma$ | Parameter |
| $\kappa$ | Parameter |
| $\lambda_i$ | $i$-ter Eigenwert |
| $\Lambda_{\boldsymbol{x}}$ | Kovarianzmatrix von $\boldsymbol{x}$ |
| $\phi(l)$ | nichtlineare Funktion |
| $\rho(k)$ | modifizierter lokaler SNR |
| $\rho(l,n)$ | segmentbezogener lokaler SNR |
| $\sigma$ | Standardabweichung |
| $\sigma^2$ | Varianz |
| $\sigma_{ie}$ | Kovarianz des $i$-ten Koeffizienten mit dem Betrag des Fehlers |
| $\omega$ | normierte Kreisfrequenz |

Es werden folgende **Schreibweisen** verwendet (am Beispiel von $x$ bzw. $\boldsymbol{x}$):

| | |
|---|---|
| $E[x]$ | Erwartungswert von $x$ |
| $E[t\|x]$ | a posteriori Mittelwert von $t$ gegeben $x$ |
| $lg(x)$ | Zehnerlogarithmus von $x$ |
| $N$ | Menge der natürlichen Zahlen |
| $R$ | Menge der reellen Zahlen |
| $[x]$ | ganzzahliger Anteil von $x$ (ohne Nachkommastellen) |
| $\hat{x}$ | Schätzwert von $x$ |
| $\bar{x}$ | Mittelwert von $x$ |
| $\boldsymbol{x}^T$ | transponierter Vektor |
| $x'$ | erste Ableitung von $x$ |
| $x''$ | zweite Ableitung von $x$ |
| $x^{(m)}$ | $m$-te Ableitung von $x$ |
| $x_c$ | durch Gewichtung mit Kostenfaktoren berechneter Wert |
| $\boldsymbol{x}_w$ | durch Gewichtung mit einer Fensterfunktion berechneter Wert |
| $x_{min}$ | Wert von $x$ im Minimum der Fehlerfunktion |
| $x_P$ | Wert von $x$ in der Pause |
| $\boldsymbol{x}_S$ | Wert von $\boldsymbol{x}$ während eines Sprachabschnitts |
| $x_{Schwelle}$ | Schwellwert für $x$ |
| $x_T$ | Werte von $x$ im Trainingsdatensatz |
| $x_V$ | Werte von $x$ im Verifikationsdatensatz |
| $x_W$ | Wert von $x$ während eines Wortes |
| $\boldsymbol{x}^l$ | $l$-ter Vektor $\boldsymbol{x}$ aus der Trainingsdatenmenge |
| $\Delta x$ | Unterschied zum vorhergehenden Wert von $x$ |
| $\nabla$ | Gradient |
| $\|\boldsymbol{x}\|$ | Norm von $\boldsymbol{x}$ |

# ABKÜRZUNGEN

| | |
|---|---|
| A/D | Analog/Digital-Wandlung |
| AKG Q400T | Kondensator-Mikrofon der Fa. AKG |
| CC | Cascade Correlation |
| CD | Compact Disk |
| CFM | Classification Figure of Merit |
| CPU | Central Processing Unit |
| DAT | Digital Audio Tape |
| DSP56ADC16 | Hardware-Baustein mit A/D-Wandler der Fa. Motorola |
| DTW | Dynamic Time Warping |
| EBP | Error Backpropagation |
| FFT | Schnelle Fouriertransformation |
| $FFT^{-1}$ | Inverse schnelle Fouriertransformation |
| HAT | Hauptachsentransformation |
| HCNN | Hidden Control Neural Network |
| IBM | International Business Machines Corporation |
| LDA | Lineare Diskriminanzanalyse |
| LMS | Least Mean Square |
| MC 723 | Studio-Kondensatormikrofon Mikrofon der Fa. Beyer |
| MLP | Multilayer Perzeptron |
| MSE | Mean Squared Error |
| NN | Künstliches Neuronales Netzwerk |
| NSS | Nichtlineare Spektralsubtraktion |
| PC | Personal Computer |
| PLP | Perceptually Based Linear Predictive Coding |
| QP | Quickprop |
| RAN | Resource Allocating Network |
| RBF | Radial Basis Function |
| SEL 1074 | Telefon der Firma Alcatel SEL AG |
| SNR | Signal-to-Noise Ratio |
| SPARC | Workstation-Serie der Fa. Sun Microsystems |
| SUNROM-1 | Geräuschdatenbasis auf CD-Datenträger |
| SUNSTAR | ESPRIT-Projekt 2094 |
| TCD-D10 | DAT-Recorder der Fa. Sony |
| VAX | Rechenanlage der Fa. Digital Equipment Corporation |

# 1. Einleitung

**Spracherkennung** erlebt derzeit einen Nachfrageboom. Hierfür sind vor allem drei Gründe zu nennen:

1. **Der technische Fortschritt**. Obwohl Spracherkennungsverfahren seit langem entwickelt und immer wieder verbessert werden, sind erst seit wenigen Jahren funktionsfähige Systeme im Handel. Neben den Verfahren selbst wurde vor allem die Gestaltung ergonomischer Mensch-Maschine-Schnittstellen verbessert.

2. **Der Nutzen**. Bei neuen Generationen technischer Geräte ist vielfach die Steuerung aller Funktionen über mechanische Bedienelemente unmöglich. Dies kann beispielsweise durch die Abmessungen oder die Komplexität des jeweiligen Geräts bedingt sein. Daher sind heute Sprachein- und -ausgabe vielfach fester Bestandteil eines **multimodalen Bedienkonzeptes**.

   **Sicherheitsgründe** können eine sprachgesteuerte Bedienung erforderlich machen, wenn die manuelle Bedienung eines Geräts den Benutzer von seiner eigentlichen Aufgabe ablenkt oder andere gefährdet. Dies kann bei Operationsmikroskopen oder bei der Benutzung eines Mobiltelefons im Straßenverkehr der Fall sein.

   **Anwendungen über Telefonleitung** lassen oft keine andere als die sprachgesteuerte Bedienung zu. Dies ist insbesondere in Telefonnetzen der Fall, in denen das Mehrfrequenzwählverfahren nicht oder nur unvollständig eingeführt ist. So werden inzwischen Auskunfts- und Bestellsysteme sowie Systeme für Home Banking mit Hilfe von Spracherkennungssystemen gesteuert.

   Schließlich enthalten automatische Übersetzungs- und Diktatsysteme Spracherkenner, da die **Wandlung von Sprache in Text oder Lautschrift** dabei elementarer Bestandteil der Aufgabe des Systems ist.

3. **Die Benutzerakzeptanz**. Da natürlichsprachliche Dialoge mit Maschinen derzeit noch nicht ohne Einschränkungen möglich sind, setzt die Bedienung sprachgesteuerter Systeme *kooperative* Benutzer voraus. Die Bereitschaft zur

Sprechdisziplin ist gewachsen, da die Benutzer mit zunehmenden technischen Möglichkeiten auch den Nutzen solcher Systeme erkennen.

Die genannten Anwendungen machen deutlich, daß **Spracherkennungssysteme** in realen Anwendungen **fehlertolerant** sein müssen. Insbesondere ihre Anfälligkeit gegenüber Störeinflüssen bei der Generierung und Wandlung des akustischen sowie der Übertragung und Verarbeitung des elektrischen Sprachsignals muß daher weiter verringert werden.

Im Rahmen dieser Arbeit werden auf **künstlichen neuronalen Netzen** basierende Verfahren untersucht, die die **Störanfälligkeit** von Spracherkennungssystemen unter realen Anwendungsbedingungen **reduzieren**. Die verfolgten Ansätze werden in die Signalvorverarbeitung integriert, die der eigentlichen Klassifikationsstufe vorgeschaltet ist. Für einen späteren Einsatz im Produkt müssen sie zumindest konzeptionell echtzeitfähig sein.

**Künstliche neuronale Netzwerke** - im folgenden kurz als *Neuronale Netze* (NN) bezeichnet - stellen **vereinfachte Modelle biologischer Systeme** aus vernetzten Nervenzellen dar. In der vorliegenden Arbeit wird ihre Eigenschaft als **nichtlineare Filter** zur Signalverarbeitung genutzt. Die Frage nach der Übereinstimmung dieser Modelle mit dem biologischen Vorbild wird dabei nicht untersucht [1].

---

[1] Zu diesem Thema sei der Leser auf die vielfältige Literatur verwiesen. Einführungen in Neuronale Netzwerke mit zahlreichen Literaturangaben werden beispielsweise von Zell (1994) und von Rojas (1993) gegeben.

# 2. Problembeschreibung und Aufgabenstellung

## 2.1. Szenario für Spracherkennung in Anwendungsumgebung

Spracherkennungssysteme arbeiten in Laborumgebung mit hoher Zuverlässigkeit. Beim praktischen Einsatz in geräuscherfüllter Umgebung wie z. B. in Büro- oder Mobilfunkanwendungen muß jedoch selbst bei kleinem oder mittlerem Vokabular mit einer drastischen Zunahme der Wortfehlerrate gerechnet werden. Bild 2.1 illustriert ein mögliches Szenario aus der Mobilkommunikation.

Aus Benutzersicht treten Fehler des Spracherkenners unerwartet und scheinbar zufällig auf, während sie aus Sicht des Entwicklers Störeinflüssen unterschiedlichen Typs zugeordnet werden können. Gegenmaßnahmen dienen entweder der *passiven Erhöhung der Störrobustheit* oder umfassen *aktive Verfahren zur Störreduktion*. Diese Begriffe werden hier auf die Spracherkennung bezogen wie folgt definiert:

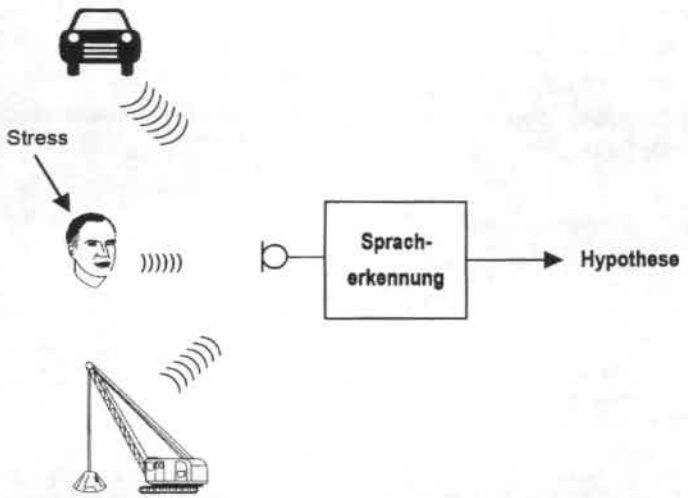



Bild 2.1: Anwendungsszenario in der Mobilkommunikation.

**Störrobustheit**: Fehlertolerante Informationsverarbeitung zur Begrenzung der Störanfälligkeit eines Spracherkennungssystems durch passive, nichtlernende Maßnahmen.

**Störreduktion:** Aktive Verfahren mit Hilfe von adaptiven oder lernenden Systemen unter Verwendung von Vorwissen über die Störung.

Eine strenge Trennung zwischen aktiven und passiven Maßnahmen ist nicht immer möglich. Oft wird in der Literatur der Begriff *Robustheit* auch für das gesamte Spracherkennungssystem ohne Unterscheidung zwischen aktiven und passiven Maßnahmen verwendet. In dieser Arbeit werden aktive Verfahren zur Störreduktion im Sinne der hier getroffenen Unterscheidung untersucht.

## 2.2. Störungsarten

Die Fehlerursachen können mit Hilfe des Modells in Bild 2.2 identifiziert und unterschiedlichen Klassen von Störungsarten zugeordnet werden. Hierbei werden folgende idealisierende Annahmen getroffen:

- Das Übertragungsverhalten von Systemen und die Kennlinien von Bauteilen sind linear.
- "Schmutzeffekte" wie die Übersteuerung eines Systems durch hohe Signalamplituden oder niedrige Signal-zu-Rausch-Verhältnisse aufgrund von Quantisierungsrauschen treten höchstens selten auf und können daher vernachlässigt werden.
- Die betrachteten Sprach- und Geräuschsignale können zumindest kurzzeitig als stationär angenommen werden.

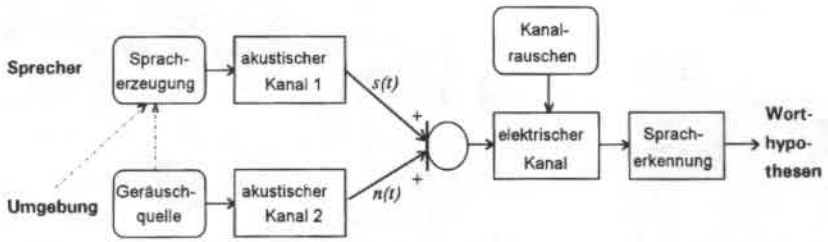



Bild 2.2: Störeinflüsse auf das Sprachsignal durch Hintergrundgeräusch, Sprecherstreß und Übertragungsfunktionen.

**Additive Hintergrundgeräusche**. An der Mikrophonmembran findet eine additive Überlagerung der Zeitsignale aus den in der Anwendungsumgebung vorhandenen Signalquellen statt. Hierzu gehören neben dem Sprachsignal $s(t)$ (Nutzsignal) die Signale aller Störgeräuschquellen, die zu einer *äquivalenten Geräuschquelle* mit dem Geräuschsignal $n(t)$ (Störsignal) zusammenfaßt werden können. Einen Sonderfall für additive Störsignale stellen Sprachsignale von Hintergrundsprechern dar, die ohne zusätzliche Information über den Sprecher (z. B. seine Position im Raum) nicht vom Nutzsignal unterschieden werden können und deshalb hier nicht näher betrachtet werden. Die Überlagerung der akustischen Signale $s(t)$ und $n(t)$ an der Mikrofonmembran ist in Bild 2.2 zu sehen.

**Sprachvariabilität**. Auch ohne Streßeinflüsse ist es einem Versuchssprecher unmöglich, eine Äußerung exakt zu reproduzieren. Veränderungen in der Sprechweise können aus Systemsicht als Störeinflüsse bezeichnet werden, da Abweichungen der Testäußerung von den Refenzmustern eine fehlerhafte Klassifikation verursachen können. Hierzu gehören:

- Abweichungen in der Sprechweise zwischen verschiedenen Äußerungen desselben Sprechers (intra-speaker variability), z. B. aufgrund des sog. *Lombardeffekts*.
- Abweichungen in der Sprechweise zwischen Äußerungen unterschiedlicher Sprecher (inter-speaker variability). Hierzu gehören auch Aussprachevarianten und akzentbehaftete Sprache.

Ursache für den Lombardeffekt sind vor allem die streßbedingte Veränderung der Sprechgewohnheiten durch mentale Belastung, z. B. bei hohen Störgeräuschpegeln oder Zusatzaufgaben wie das Steuern eines Kraftfahrzeugs. Die Auswirkungen auf das Sprachsignal drücken sich durch Änderungen von Signalparametern aus, zu denen u. a. Grundfrequenz, Formanten, Langzeitspektrum und Lautdauer gehören. Einige davon sind mit statistischen Methoden beschreibbar (Hansen and Bria 1992; Junqua 1992), wenn auch die bisherigen Ergebnisse einige Unsicherheiten aufweisen. Wegen der - meist nicht verfügbaren - hohen Zahl der benötigten Versuchspersonen und der Abhängigkeit der Ergebnisse vom Versuchsaufbau sind sie eher als Trends anzusehen. Die Einflüsse des Lombardeffekts auf die Spracherzeugung sind schematisch durch die strichpunktierten Linien in Bild 2.2 wiedergegeben. Wird Sprecherunabhängigkeit des Systems gefordert, treten zusätzlich Probleme durch sprecherübergreifende Unterschiede in der Sprechweise auf.

**Kanalverzerrungen**. Hierzu gehören lineare Verzerrungen des Signals, die bei der Ausbreitung des akustischen und der Übertragung des elektrischen Signals auftreten, vgl. Bild 2.2. Sie können sich auf Sprach- und Geräuschsignal auswirken und sind auf

- Akustische und elektrische Übertragungskanäle,
- Reflexionen des akustischen und elektrischen Signals (Echos) sowie auf
- Übertragungsfunktionen von elektronischen Bauteilen und Systemen

zurückzuführen. Bei Kanalverzerrungen auftretende Übertragungsfunktionen können zeitveränderlich sein, wobei Zeitkonstanten unterschiedlicher Größenordnung auftreten können. Beispiele hierfür sind akustische Übertragungskanäle bei bewegten Geräuschquellen oder variabler Abstand des Sprechers zum Mikrofon beim Freisprechen sowie die zeitveränderlichen Übertragungseigenschaften einer Funkstrecke.

**Rauschen**. Rauschen kann an unterschiedlichen Stellen des Gesamtsystems auftreten; stellvertretend hierfür ist in Bild 2.2 das Kanalrauschen symbolisiert. Zu den unterschiedlichen Rauschtypen gehören:

- Quantisierungsrauschen,
- Rauschen in Bauteilen,
- Kanalrauschen.

Da sich Rauschen in verschiedenen Verarbeitungsstufen des Systems auf die jeweils vorhandene Signalrepräsentation additiv auswirkt, kann sein Einfluß auf die Erkennungsleistung nicht ohne genauere Analyse angegeben werden.

**Andere Störungsarten**. Aus der Vielzahl anderer Störungsarten sollen noch die für die Spracherkennung wichtigen Hintergrundsprecher genannt werden; obwohl die Zeitsignale auch in diesem Falle additiv überlagert werden, tritt hier noch das Problem der Unterscheidbarkeit zwischen Nutz- und Störsignal auf.

### 2.3. Maßnahmen zur Behandlung der unterschiedlichen Störungsarten

Aus der Literatur ist eine Vielzahl von Arbeiten zur Erhöhung der Robustheit von Spracherkennungssystemen bekannt. Ein Überblick wird z. B. von Furui (1992) gegeben. In Tabelle 2.1 wird eine von vielen möglichen Aufteilungen vorgenommen, die sich an der getroffenen Unterscheidung der Störungstypen orientiert. Sie zeigt einen kleinen Ausschnitt der zahlreichen Literaturstellen zu diesem Thema.

Tabelle 2.1: Aktive und passive Verfahren zur Eliminierung unterschiedlicher Störungstypen.

| | Störungstyp | | |
|---|---|---|---|
| **Verfahren** | **Additive Geräusche** | **Lombardeffekt** | **Kanalverzerrungen** |
| **passiv** | **Gehörmodelle** in der Vorverarbeitung (Hermansky et al. 1991)<br>**Geräuschrobuste Merkmale** (Palival 1990, Hanson and Applebaum 1990)<br>**Störrobuste Abstandsmaße** (Furui 1992) | **Lombardbehaftete Referenzmuster** (Multi-Style Training; Lippmann et al. 1987) | **Verzerrungsrobuste Merkmale** (Hermansky et al. 1991) |
| **aktiv** | **Adaptive Filter zur Geräuschkompensation** (Kroschel 1988)<br>**Spektralsubtraktion** (Vary 1983; Reich 1985)<br>**Nichtlineare Spektralsubtraktion** (Lockwood et al. 1991, 1992)<br>**Neuronale Geräuschreduktion** (Tamura and Waibel 1988, 1989, 1990; Sorensen 1991) | **Musteradaption** (Dvorak and Hörmann 1991)<br>**Reduktion des Lombardeffekts** (Hansen and Bria 1992) | **Adaptive Filter zur Echokompensation** (Widrow 1975) |

### 2.4. Inhalt und Gliederung der Arbeit

Im weiteren Verlauf dieser Arbeit werden **aktive Verfahren** zur **Reduktion additiver Hintergrundgeräusche** auf Basis von neuronalen Netzwerken untersucht, wobei der zugrundeliegende Ansatz auch auf andere Störungsarten angewendet werden kann.

Eine **Einführung in neuronale Netzwerke** im 3. Kapitel wird im Zusammenhang mit ihren Einsatzmöglichkeiten für die vorliegende Aufgabe gegeben und stützt sich auf die Abbildungseigenschaften sog. *Multilayer Perzeptron*-Netzwerke (MLP). Darüber hinaus werden bekannte Ansätze zur automatischen Netzwerkgenerierung und zur Adaption vortrainierter Netzwerke diskutiert.

Reproduzierbare Versuchsbedingungen machen die Verwendung getrennter **Sprach- und Geräuschdatenbasen** erforderlich, aus denen die benötigten Trainings- und Testdaten auf dem Rechner generiert werden. Ihre Beschreibung erfolgt im 4. Kapitel zusammen mit dem verwendeten **Testbett** zur sprecherabhängigen Isoliertworterkennung, das zur Evaluierung der Geräuschreduktionsleistung sowie zur Parameteroptimierung mit Hilfe von Worterkennungsraten dient.

Die experimentellen Untersuchungen beschäftigen sich zunächst mit der Entwicklung der **Netzwerktopologie** und der Optimierung des **Trainingsverfahrens** für das Multilayer Perzeptron-Netzwerkmodell (Kapitel 5). Hierzu gehört auch die Untersuchung geeigneter Signalrepräsentationen und die Evaluierung der Robustheit der entwickelten Netzwerke.

Als Alternative zur experimentellen Vorgehensweise bei der Entwicklung einer geeigneten Topologie bieten sich **automatische Netzgenerierungsverfahren** an, die im 6. Kapitel beschrieben werden.

Für den Einsatz in instationären Signalumgebungen (z. B. wechselnde Geräuschquellen) wird die **Adaption vortrainierter Netzwerke** auf Basis von Signaleigenschaften in den Sprachpausen untersucht (Kapitel 7).

Schließlich werden im 8. Kapitel die auf **neuronalen Netzwerken** basierenden Verfahren mit einem **Spektralsubtraktionsverfahren** verglichen.

# 3. Lösungsansätze mit Neuronalen Netzwerken

In diesem Kapitel werden Lösungsansätze zur Behandlung verschiedener Störungsarten vorgestellt. Dabei wird das Störreduktionsproblem als eine nichtlineare Abbildungsaufgabe zwischen störbehafteten und störfreien Signalsegmenten betrachtet, die mit neuronalen Netzwerken realisiert wird. Zunächst wird der klassische Anwendungsfall einmaladaptiver Netzwerke betrachtet, deren Parameter vorab in einer Trainingsphase bestimmt werden und dann unverändert bleiben. Darauf aufbauend wird dann ein Konzept für eine fortlaufende Netzwerkadaption an zeitveränderliche Abbildungsaufgaben vorgestellt.

## 3.1. Einführung in Neuronale Netzwerke

Aufgrund der unterschiedlichen Aufgaben biologischer Nervenzellanordnungen wird eine Vielzahl verschiedener Netzwerkmodelle in der Literatur beschrieben. Eine Unterteilung kann beispielsweise nach ihrer Topologie, dem Trainingsverfahren oder anhand der Aufgaben vorgenommen werden, die mit ihrer Hilfe gelöst werden können [2].

Die in dieser Arbeit verwendeten Netzwerktypen werden mit Hilfe von überwachtem Lernen trainiert. Am Beispiel des Netzwerktyps *Multilayer Perzeptron* und des Trainingsverfahrens *Error Backpropagation* (EBP) werden nachfolgend die Grundbegriffe künstlicher neuronaler Netzwerke eingeführt, soweit sie zum Verständnis der vorliegenden Arbeit erforderlich sind.

### 3.1.1. Grundbausteine der Netztopologie

Künstliche neuronale Netze stellen meist vereinfachte Modelle ihres biologischen Vorbilds dar. Die Grundfunktionen der Nervenzellen (Neuronen) werden durch einzelne Rechenelemente, den sog. *Knoten* des Netzwerks, modelliert und mit

---

[2] Ein Überblick über diese Modelle ist beispielsweise in Lippmann (1987) oder Zell (1994) zu finden.

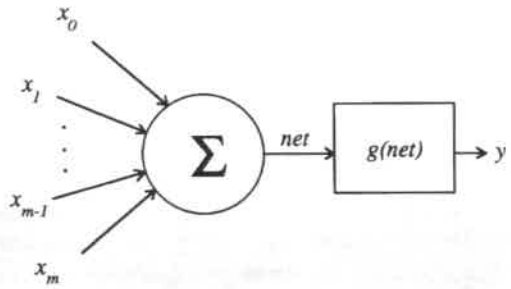



Bild 3.1: Vereinfachtes Modell eines Neurons

Hilfe von gerichteten *Verbindungen* zu komplexeren Netzwerkstrukturen verschaltet. Dabei findet wie bei biologischen Nervenzellanordnungen an der Synapse eine Steuerung des Signalflusses statt, dem im Modell eine multiplikative Bewertung der Verbindungen mit Gewichtsfaktoren (*Gewichten*) entspricht. Schichtweise zusammengefasste Neuronengruppen werden kurz als *Schichten* (*Layer*) des Netzwerks bezeichnet.

Mit den genannten Grundelementen der Netzstruktur findet eine Signalverarbeitung vom Netzwerkeingang zum -ausgang statt, die sich mit Hilfe einer Übertragungsfunktion beschreiben läßt. Diese setzt sich aus den Rechenvorschriften der Neuronenmodelle, ihrer Verbindungsstruktur sowie aus den aktuellen Gewichtswerten zusammen.

**Knoten des Netzwerks**. Ein vereinfachtes Modell eines Neurons ist in Bild 3.1 zu sehen. Es besteht aus der Summation der netzseitigen Eingangserregungen $x_j$, $j=0,\ldots,m$, und der Bewertung des zellinternen *Erregungspotentials net* mit einer *Aktivierungsfunktion* $g()$.

Am *Bias-* oder *Offseteingang* $x_0$ wird über eine konstant anliegende Erregung das Übertragungsverhalten des Knotens gesteuert. Wie bei biologischen Zellen, die erst nach Überschreiten des Schwellwerts durch das interne Erregungspotential impulsförmige Ausgangssignale weitergeben, steuert $x_0$ den Arbeitspunkt des Knotens. Bei $m+1$ Eingängen errechnet sich die Ausgangserregung $y$ aus

$$y(\boldsymbol{x}) = g(net) = g\left(\sum_{j=0}^{m} x_j\right) \quad . \tag{3.1}$$

In diesem Modell wird die Ausgangserregung innerhalb des betrachteten Zeitabschnitts als statischer Wert angenommen, während die Erregung in Nervenleitungen durch Impulsfolgen unterschiedlicher Zeitdauer und Frequenz kodiert ist.

Häufig verwendete Aktivierungsfunktionen (z. B. Zell 1994) sind die *sigmoid*-Funktion, die *lineare* Funktion oder die *gaußförmige* Aktivierungsfunktion. Sie sind durch die folgenden Berechnungsvorschriften gegeben:

Sigmoid-Funktion: $$g_{sig}(net) = \frac{1}{1+e^{-net}} \quad , \qquad (3.2)$$

Gaußförmige Funktion: $$g_{gauß}(net) = e^{-\frac{net^2}{2\sigma^2}} \quad , \qquad (3.3)$$

Lineare Funktion: $$g_{lin}(net) = \begin{cases} 1 & net > 1 \\ net & \text{für } -1 \le net \le 1 \\ -1 & net < -1 \end{cases} \quad , \qquad (3.4)$$

$net \in R$ Erregungspotential.

Die sigmoid-Funktion ist eine stetige, monoton steigende Approximation einer Schwellwertfunktion. $g_{gauß}$ ist bis auf einen Normierungsfaktor identisch mit einer Gaußfunktion mit Mittelwert 0 und Standardabweichung $\sigma$, während die lineare Funktion die Steigung 1 hat und - abgesehen von Begrenzungseffekten - die Summe der Eingangserregungen ausgangsseitig weitergibt. Letztere wird in den Eingangsknoten gebraucht, die die eingangsseitige Erregung ohne weitere Verarbeitung an die Knoten der nächsten Schicht durchreichen.

In den verdeckten Knoten [3] werden stetige nichtlineare Aktivierungsfunktionen gefordert, da in die Berechnung der Gewichtsmodifikation mit EBP die erste Ableitung der Aktivierungsfunktion eingeht (vgl. Abschnitt 3.1.2, Gln. (3.18) und (3.19)). Hier werden oft sigmoid-Funktionen eingesetzt, da ihre Ableitung

$$g'_{sig}(net) = net(1-net) \qquad (3.5)$$

einfach zu berechnen ist. Gaußförmige Funktionen von Eingangsvektoren werden z. B. beim *Resource Allocating Network* (RAN) verwendet (siehe Abschnitt 6.2).

**Topologie**. Der Begriff *Netzwerktopologie* schließt neben Zahl und Anordnung der Knoten auch die Verbindungsstruktur ein. Als Beispiel ist in Bild 3.2 ein Netz-

---

[3] Als *verdeckt* werden diejenigen Knoten bezeichnet, die weder der Eingangs- noch der Ausgangsschicht angehören; in Bild 3.2 sind dies die Knoten der Zwischenschicht.

werk mit MLP-Struktur abgebildet. Aus den unterschiedlichen Zählweisen in der Literatur soll diejenige übernommen werden, bei der die Eingangsschicht mitgezählt und das abgebildete Netzwerk somit als dreischichtiges Perzeptron bezeichnet wird. Im Bild sind alle Knoten schichtweise angeordnet und mit jedem Knoten der nächsten Schicht verbunden. Verbindungen zwischen den Schichten sind vorwärtsgerichtet. Knoten innerhalb einer Schicht haben keine Verbindung.

Das abgebildete Netz hat $m$ Eingangs-, $h$ verdeckte und aus Darstellungsgründen einen einzigen Ausgangsknoten. Die Aktivierungsfunktionen der verdeckten Knoten sind mit $g()$, die des Ausgangsknotens mit $f()$ bezeichnet. Dann ist die Antwort auf die Eingangserregung $\boldsymbol{x}$ durch die Übertragungsfunktion

$$F(\boldsymbol{x}) = f\left(\sum_{i=1}^{h} v_i \cdot g\left(\sum_{j=1}^{m} w_{ij}\, x_j + x_0^i\right) + z_0\right) = f\left(\sum_{i=1}^{h} v_i \cdot z_i + z_0\right) \tag{3.6}$$

gegeben, wobei $z_i$ die Erregung am Ausgang des $i$-ten Knotens der verdeckten Schicht und $v_i$ und $w_{ij}$ Elemente der Gewichtsmatrizen $\boldsymbol{V}$ und $\boldsymbol{W}$ von der verdeck-

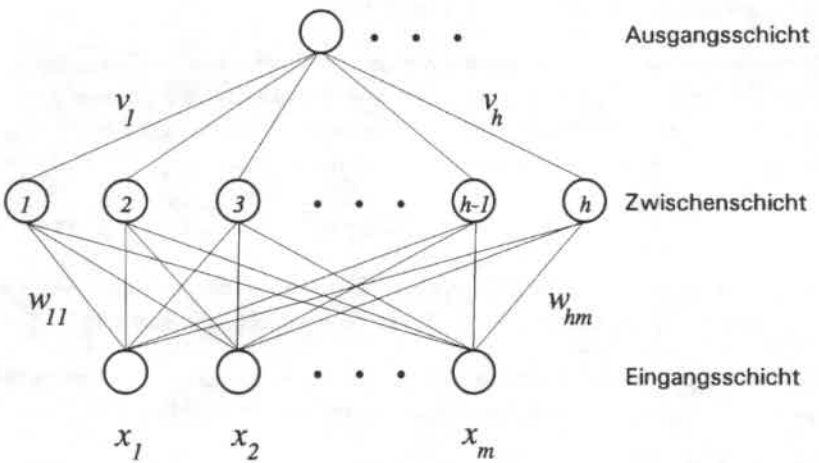



Bild 3.2: Dreischichtiges Perzeptron mit $m$ Eingangs-, $h$ verdeckten und einem Ausgangsknoten (Offseteingänge nicht abgebildet).

ten zur Ausgangs- bzw. von der Eingangs- zur verdeckten Schicht bedeuten. Die Offseteingänge am $i$-ten Knoten der verdeckten Schicht bzw. am Ausgangsknoten sind durch die zusätzlichen, nicht abgebildeten Eingänge mit den Erregungen $x_0^i$ bzw. $z_0$ berücksichtigt.

### 3.1.2. Training

Während des Trainings werden die Netzwerkparameter so eingestellt, daß die gestellte Aufgabe im Sinne eines Gütekriteriums optimal erfüllt wird. Mit Hilfe des *überwachten Lernens* wird iterativ eine *Ziel-* oder *Kostenfunktion* minimiert, die meist auf Basis des quadratischen Fehlers formuliert wird.

**Betriebsarten.** Bei einmaladaptiven Netzwerken unterscheidet man zwischen der anfänglichen *Lern-* oder *Trainings-* und der darauffolgenden *Testphase*. Außerdem ist für die Testphase der Begriff *Vorwärtsrechnung* im Gegensatz zur Rückrechnung des Fehlers während der Gewichtsmodifikation mit EBP gebräuchlich.

**Trainingsdaten.** Das Training erfolgt mit Hilfe eines repräsentativen Datensatzes [4]. Die Trainingsmuster bestehen aus Paaren von Eingabe- und gewünschten Ausgabemustern, die auch als *Zielmuster* (*Target Vector*) bezeichnet werden. Ihre Dimensionalität kann verschieden sein und entspricht der Zahl von Eingangs- bzw. Ausgangsknoten des Netzwerks. Für die weiteren Betrachtungen sei der Trainingsdatensatz durch $L$ Trainingsvektorpaare $(\boldsymbol{x}(l), \boldsymbol{t}(l))$ mit $l=1,\ldots,L$ gegeben, die Koeffizientenzahl des Eingabevektors $\boldsymbol{x}$ sei $m$ und die des Zielvektors $\boldsymbol{t}$ sei $q$. Für die folgenden Betrachtungen das Netzwerk in Bild 3.2 ebenfalls auf $q$ Ausgangsknoten erweitert. Zur Vereinfachung der Schreibweise wird das $l$-te Vektorpaar des Trainingsdatensatzes im folgenden mit $(\boldsymbol{x}^l, \boldsymbol{t}^l)$ bezeichnet.

**Zielfunktion.** Die am häufigsten verwendete Zielfunktion ist der mittlere quadratische Fehler (*Mean Squared Error*, MSE). Zu seiner Berechnung benötigt man den auf das $l$-te Vektorpaar bezogenen quadratischen Fehler

$$\boldsymbol{e}^{Tl}\boldsymbol{e}^{\,l} = \sum_{k=1}^{q}\left(F_k\left(\boldsymbol{x}^l\right) - t_k^l\right)^2 = \sum_{k=1}^{q}\left(y_k^l - t_k^l\right)^2 \quad , \tag{3.7}$$

wobei $q$ die Koeffizientenzahl ist. Dann kann der Ausgangsvektor als

---

[4] Eine *repräsentative* Trainingsstichprobe besitzt dieselben statistischen Eigenschaften wie die Testdaten.

$$\boldsymbol{y}^l = \boldsymbol{F}\left(\boldsymbol{x}^l\right) \qquad \text{mit} \qquad \boldsymbol{y}^l = \left(y_1^l, \ldots, y_q^l\right)^T \tag{3.8}$$

geschrieben werden. Der über alle Trainingsbeispiele gemittelte $MSE_T$ berechnet sich dann durch Summation aller Trainingsbeispiele und Normierung aus

$$\begin{aligned} MSE_T &= \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{q}\left(y_k^l - t_k^l\right)^2 \\ &= \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{e}^{T^l}\boldsymbol{e}^l \end{aligned} \tag{3.9}$$

Aus der Summation über $l$ läßt sich erkennen, daß der Gesamtfehler aus den Fehlerbeiträgen der einzelnen Vektorpaare besteht. Wegen $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{V}, \boldsymbol{W})$ ist $MSE_T$ ebenfalls eine Funktion der Gewichte in den Matrizen $\boldsymbol{V}$ und $\boldsymbol{W}$, d. h.

$$MSE_T = MSE_T(\boldsymbol{V}, \boldsymbol{W}) \tag{3.10}$$

**Kosten**. Wenn sich aus der Aufgabenstellung unterschiedliche Kosten $C_k$ für die Fehlerbeiträge der einzelnen Netzwerkausgänge angeben lassen, kann der auf den $l$-ten Vektor bezogene, gewichtete quadratische Fehler aus

$$\boldsymbol{e}_c^{T^l}\boldsymbol{e}_c^l = \sum_{k=1}^{q} C_k\left(y_k^l - t_k^l\right)^2 \tag{3.11}$$

berechnet werden. Der mit den Kosten gewichtete $MSE_c$ wird dann analog zu Gl. (3.9) durch Ersetzen von $\boldsymbol{e}^l$ durch $\boldsymbol{e}_c^l$ berechnet.

**Gradientenabstieg mit Error Backpropagation** [5] . EBP liefert eine Vorschrift zur Gewichtsmodifikation in Richtung des Minimums der Fehlerkurve, in dem bestmögliche Übereinstimmung zwischen Trainingszielen und tatsächlichen Ausgangsvektoren im Sinne des Fehlermaßes gegeben ist. Bei der Suche mit Gradientenabstiegsverfahren ist die Gewichtsänderung proportional zum negativen Gradienten der Fehlerfunktion $MSE_T$ nach Gl. (3.10) mit einer reellwertigen Lernrate $lr$ als Proportionalitätsfaktor. Man unterscheidet bei der Herleitung der Lernregeln zwischen der Modifikation der Gewichte $v_{ki}$ der Gewichtsmatrix $\boldsymbol{V}$ von

---

[5] Die folgende Beschreibung des Gradientenabstiegs mit Error Backpropagation gibt das Ergebnis der Herleitungen in Rumelhart et al. (1986) und Zell (1994) wieder. Die Bezeichnungen sind den Bildern 3.2 und 3.3 sowie den zugehörigen Erklärungen entnommen.

der verdeckten zur Ausgangsschicht und der Gewichte $w_{ij}$ der Gewichtsmatrix $\boldsymbol{W}$ von der Eingangs- zur verdeckten Schicht.

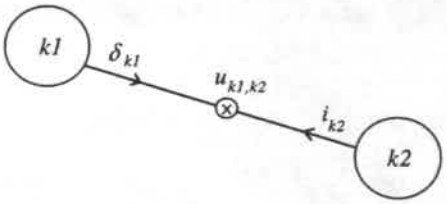



Bild 3.3: Gewichtsmodifikation nach der Delta-Regel.

Zum Zeitpunkt $n$ erfolgt die Aktualisierung des Gewichts vom verdeckten Knoten $i$ zum Ausgangsknoten $k$ ausgehend vom vergangenen Zeitpunkt $n$-1 gemäß

$$v_{ki}(n) = v_{ki}(n-1) + \Delta v_{ki}(n) \quad , \tag{3.12}$$

vom Eingangsknoten $j$ zum verdeckten Knoten $i$ gemäß

$$w_{ij}(n) = w_{ij}(n-1) + \Delta w_{ij}(n) \quad . \tag{3.13}$$

Der Lösungsansatz für die Berechnung der Gewichtsmodifikation basiert auf den partiellen Ableitungen des Fehlers nach den jeweiligen Gewichten:

$$\Delta v_{ki} = -lr \frac{\partial}{\partial v_{ki}} MSE_T(\boldsymbol{V},\boldsymbol{W}) \quad \text{bzw.} \quad \Delta w_{ij} = -lr \frac{\partial}{\partial w_{ij}} MSE_T(\boldsymbol{V},\boldsymbol{W}) \quad . \tag{3.14}$$

Die Berechnung des Gradienten erfolgt durch Rückrechnung (*Backpropagation*) von $MSE_T$ vom Ausgang des Netzwerks auf jedes einzelne Gewicht mit Hilfe der Kettenregel. Als Ergebnis erhält man eine Vorschrift zur Aktualisierung der Gewichte, die in der Literatur unter der Bezeichnung *generalisierte Delta-Regel* bekannt ist. Sie hat für alle Gewichte des Netzwerks eine gemeinsame Form und lautet für die Modifikation des Gewichts $u$ zwischen zwei Knoten [6] $k1$ und $k2$ als Antwort auf den Fehlerbeitrag des $l$-ten Vektorpaares (vgl. Bild 3.3)

$$\Delta u^l_{k1,k2} = lr\, \delta^l_{k1}\, i^l_{k2} \quad , \quad u \in \boldsymbol{W} \vee u \in \boldsymbol{V} \quad , \tag{3.15}$$

wobei die in der Gleichung auftretenden Faktoren folgende Bedeutung haben:

---

[6] Dabei wird angenommen, daß die Knoten aus unterschiedlichen Schichten stammen und daß $k1$ einer höheren Netzwerkschicht als $k2$ angehört.

$\delta_{k1}$ auf den Knoten $k1$ rückgerechneter Fehler vom Netzwerkausgang.

$i_{k2}$ Eingangserregung, auf den Ausgang des Knotens $k2$ vorwärtsgerechnet.

$\delta^l_{k1}$ und $i^l_{k2}$ sind bei der Modifikation der Gewichte $w_{ij}$ durch

$$i^l_{k2} = x^l_j \quad \text{und} \quad \delta^l_{k1} = \sum_{k=1}^{q} \left(y^l_k - t^l_k\right) f'\left(net^l_k\right) v_{ki}\, g'\left(net^l_i\right) \tag{3.16}$$

und bei Modifikation der Gewichte $v_{ki}$ durch

$$i^l_{k2} = z^l_i \quad \text{und} \quad \delta^l_{k1} = \left(y^l_k - t^l_k\right) f'\left(net^l_k\right) \tag{3.17}$$

gegeben. Dabei bedeuten (vgl. Gl. (3.6)):

| | |
|---|---|
| $x$ | Eingangsvektor, |
| $z$ | Aktivierungen an den Ausgängen der verdeckten Knoten, |
| $net$ | Erregungspotential eines Knotens, |
| $f()$ | Aktivierungsfunktionen der Ausgangsknoten, |
| $g()$ | Aktivierungsfunktionen der verdeckten Knoten. |

Aus der allgemeinen Form in Gl. (3.15) erhält man durch Einsetzen von Gl. (3.16) bzw. (3.17) unterschiedliche Ergebnisse für den Beitrag des $l$-ten Vektorpaars zur Gewichtsmodifikation:

$$\Delta w^l_{ij} = lr \sum_{k=1}^{q} \left(y^l_k - t^l_k\right) f'\left(net^l_k\right) v_{ki}\, g'\left(net^l_i\right) x^l_j \tag{3.18}$$

bzw.

$$\Delta v^l_{ki} = lr \left(y^l_k - t^l_k\right) f'\left(net^l_k\right) \cdot z^l_i \tag{3.19}$$

Für die Gesamtänderung mit den $L$ aufsummierten Fehlerbeiträgen gilt dann

$$\Delta w_{ij} = \sum_l \Delta w^l_{ij} \quad \text{bzw.} \quad \Delta v_{ki} = \sum_l \Delta v^l_{ki} \tag{3.20}$$

**Batch Learning und Stochastischer Gradientenabstieg.** Bei der Gewichtsmodifikation wird zwischen folgenden Vorgehensweisen unterschieden:

1. Addiert man alle Fehlerbeiträge nach Gl. (3.20) zuerst auf und führt dann die Gewichtsmodifikation nach (3.18) und (3.19) in einem Schritt durch (die Einzelbeiträge der Vektoren werden durch ihre Summen ersetzt), erfolgt die Ge-

wichtsmodifikation in Richtung des Gradienten der Fehlerkurve über den gesamten Trainingsdatensatz (*Batch* oder *Offline Learning*). Bei diesem Verfahren muß bei Trainingsbeginn der gesamte Datensatz zur Verfügung stehen; neue Muster können während des Trainings nicht hinzugefügt werden.

2. Werden die Gewichte wie in (3.15) bis (3.19) nach jedem eingangsseitig angelegten Vektorpaar modifiziert, folgen sie bei der Gewichtsmodifikation den lokalen Eigenschaften der Trainingsdaten (*stochastischer Gradientenabstieg, Online Learning*). Da der stochastische Gradientenabstieg in der Regel ein schnelleres Konvergenzverhalten aufweist, wird dieses Verfahren meist bevorzugt. Neue Trainingsmuster lassen sich leicht hinzufügen, weil das Netzwerk während des Trainings jeweils nur einen kleinen Ausschnitt der gesamten Datenmenge "sieht".

**Momentum-Term.** Um eine Spezialisierung des Netzwerks auf lokale Eigenschaften der Trainingsdaten zu verhindern, kann der vergangene Wert der Gewichtsänderung aus der vorhergehenden Iteration mit einem *Momentum-Term* $m_t$ gewichtet zur aktuellen Gewichtsänderung addiert werden:

$$\Delta u^l_{k1,k2}(n) = lr\,\delta^l_{k1}\,i^l_{k2} + m_t\,\Delta u^l_{k1,k2}(n-1) \quad , \quad 0 \le m_t \le 1 \; . \qquad (3.21)$$

Man erreicht so ein Tiefpaßverhalten, das die Modifikationsvorschrift weniger empfindlich gegenüber lokalen Minima der Fehlerkurve macht. Insbesondere für den stochastischen Gradientenabstieg wird die nach Gl. (3.21) modifizierte Deltaregel häufig verwendet. Eine ausführlichere Diskussion dieses Verfahrens ist in Rumelhart et al. (1986) zu finden. Dort wird u. a. darauf hingewiesen, daß die Einführung des Momentum-Terms größere Werte für die Schrittweite erlaubt, was in den meisten Fällen eine schnellere Konvergenz des Trainings zur Folge hat.

**Variable Lernrate.** Eine Möglichkeit zur Steuerung der Konvergenzgeschwindigkeit ist die Veränderung der Lernrate $lr$ entsprechend dem Trainingsfortschritt, vgl. z. B. Franzini (1987). Aus der Vielzahl bekannter Varianten wurde in der vorliegenden Arbeit folgende verwendet: von einem zufälligen Startpunkt aus wird zunächst mit großer Lernrate gesucht. Da in der Nähe des Minimums mit einem flachen Verlauf der Fehlerkurve zu rechnen ist, wird bei abnehmendem Gradienten $lr$ sukzessiv um einen experimentell bestimmten Faktor $b_l$ verringert. Dies hat eine höhere Auflösung der Suchschritte in Minimumnähe zur Folge. Die Neuberechnung von $lr$ zwischen den Iterationen $n$-1 und $n$ gemäß

$$lr(n) = \frac{lr(n-1)}{b_1} \qquad 1 \leq b_1 \qquad (3.22)$$

erfolgt dann, wenn ein flacher Verlauf des Fehlers durch Unterschreiten einer Schwelle für die Fehlerdifferenz $\Delta MSE_T$ zwischen zwei Iterationen gemäß

$$\Delta MSE_T(n) = MSE_T(n-1) - MSE_T(n) < \Delta MSE_{Schwelle}(n-1) \qquad (3.23)$$

detektiert wird. Gleichzeitig wird wegen der kleineren Schrittweite die Schwelle mit

$$\Delta MSE_{Schwelle}(n) = \frac{\Delta MSE_{Schwelle}(n-1)}{b_2} \quad , \quad b_2 \leq b_1 \quad . \qquad (3.24)$$

neu berechnet. Die Vorschrift $b_2 \leq b_1$ folgt aus der Überlegung, daß aufgrund des erwarteten flacheren Verlaufs der Fehlerkurve eine nochmalige Reduktion der Lernrate im darauffolgenden Schritt möglichst verhindert werden soll. Bei schrittweiser Verringerung von *lr* muß eine untere Grenze zum Trainingsabbruch führen, da sonst die Suche mit immer kleineren Schritten fortgesetzt wird.

**Generalisierungsfähigkeit und Cross Validation.** Während der Minimumsuche besteht die Gefahr der *Spezialisierung* der Gewichte auf die Trainingsdaten (*Übertraining*). Statt des Auswendiglernens von Beispieldaten soll das Lernziel vielmehr eine Abstraktion auf die eigentliche Trainingsaufgabe sein (*Generalisierungsfähigkeit*). Dadurch wird die erfolgreiche Anwendung einer einmal gelernten Zuordnung auf nicht zum Training verwendete Daten gleicher statistischer Eigenschaften ermöglicht.

Um ein Übertraining zu vermeiden, kann im Anschluß an jede Trainingsiteration die bis dahin erreichte Generalisierungsfähigkeit getestet werden. Hierzu wird ein getrennter *Verifikationsdatensatz* gleicher statistischer Eigenschaften verwendet, der nicht zur Gewichtsmodifikation herangezogen wird. Nach jeder Iteration $n$ wird dabei durch Vorwärtsrechnung aller Muster analog zu Gl. (3.9) der $MSE_V$ im Verifikationsdatensatz berechnet. Das Training wird fortgesetzt, solange

$$MSE_V(n) < MSE_V(n-1) \qquad (3.25)$$

gilt. Wird während des Gradientenabstiegs Gl. (3.25) trotz weiter abnehmendem $MSE_T$ verletzt, ist dies auf eine beginnende Spezialisierung des Netzwerks zurückzuführen, was durch rechtzeitigen Trainingsabbruch verhindert werden kann.

Dieses Verfahren wird als *Cross Validation* bezeichnet und wurde u. a. erfolgreich bei der Phonemerkennung eingesetzt (z. B. Morgan and Bourlard 1989). Eine Diskussion der Varianten dieses Verfahrens ist in Stone (1978) zu finden.

**Weight Decay.** Häufig strebt man die Einstellung betragsmäßig möglichst kleiner Gewichtswerte im Verlauf des Trainings an. Durch eine entsprechende Modifikation der Zielfunktion kann erreicht werden, daß gleichzeitig der $MSE$ und der Betrag der Gewichtswerte minimiert werden. Hierzu wird die Zielfunktion und daraus abgeleitet die Berechnungsvorschriften für $\Delta v_{ki}$ bzw. $\Delta w_{ki}$ in Gl. (3.14) um einen vom jeweils aktuellen Gewichtswert abhängigen Term ergänzt. Dieses mit *Weight Decay* bezeichnete Verfahren wird z. B. in Werbos (1988) beschrieben. Zusätzliche Vorschläge zur mathematischen Formulierung von Zielfunktionen und Update-Regeln mit Weight Decay findet man beispielsweise in Hertz et al. (1991).

**Weitere Varianten von Error Backpropagation.** Ein Überblick über weitere zahlreiche Varianten von Error Backpropagation ist in Hertz et al. (1991) oder in Zell (1994) enthalten. Zu diesen Varianten gehören u. a. Trainingsverfahren, die Ableitungen höherer Ordnung der Zielfunktion verwenden. Ein Vertreter dieser Verfahren ist der *Quickprop*-Trainingsalgorithmus, der im Zusammenhang mit dem *Cascade Correlation Lernalgorithmus* im 6. Kapitel beschrieben wird.

### 3.2. Abbildungseigenschaften von Multilayer Perzeptron-Netzwerken

Aufgrund ihrer Generalisierungsfähigkeit (vgl. Abschnitt 3.1) können mit Error Backpropagation trainierte Multilayer Perzeptron-Netzwerke aus Trainingsbeispielen eine Zuordnungs- oder Abbildungsaufgabe[7] lernen. Daher werden zunächst ihre Abbildungseigenschaften näher betrachtet mit dem Ziel, sie für die Formulierung eines neuronalen Ansatzes zur Störreduktion zu nutzen. Aus der Literatur (Hornik et al. (1989); Hecht-Nielsen (1990)) ist bekannt, daß MLP-Netzwerke folgende Eigenschaft besitzen:

> Ein dreistufiges Multilayer Perzeptron-Netzwerk mit $m$ Eingängen, $q$ Ausgängen und höchstens $2m+1$ Knoten mit nichtlinearen Aktivierungsfunktionen in der Zwischenschicht kann $q$ nichtlineare beschränkte und stetige Funktionen $F_o=\{F_1,...,F_q\}$ aus repräsentativem Trainingsmaterial beliebig genau approximieren, wenn das Training mit Error Backpropagation durchgeführt wird.

Die Aktivierungsfunktionen der Knoten in der verdeckten Schicht müssen stetig, monoton und beschränkt sein. Obiger Satz gibt lediglich die Existenz eines Netzwerks und eine obere Grenze für die Knotenzahl an, enthält jedoch keine genaue Vorschrift für den Entwurf der Topologie. Aus der Literatur sind daher zahlreiche empirische Untersuchungen zu diesem Thema bekannt (z. B. Fahlman 1988); sie werden im Zusammenhang mit den Experimenten (vgl. Kapitel 5, 6 und 7) beschrieben, soweit sie für die vorliegende Arbeit relevant sind.

Von *repräsentativen* Trainingsdaten wird gefordert, daß ihre Dichtefunktion $f_t(t)$ sowie die bedingte Dichtefunktion $f_t(t|x)$ mit denen der Testdaten übereinstimmen. Andernfalls liegen keine repräsentativen Trainingsdaten oder instationäre Signale vor, und das Netzwerk muß an die neue Statistik adaptiert werden.

In Gl. (3.9) wurde die Berechnung des mittleren quadratischen Fehlers $MSE_T$ aus den Vektorpaaren $(x^l, t^l)$ des Trainingsdatensatzes angegeben. Daraus folgt zusammen mit obigem Satz zur Approximation $F$ einer gesuchten Abbildungsfunktion $F_o$

---

[7] Der in der Nachrichtentechnik verwendete Begriff der *Übertragungsfunktion* wird bei neuronalen Netzwerken häufig als *Abbildungsfunktion* bezeichnet.

$$\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{q}\left(F_k(x^l)-F_{ok}(x^l)\right)^2<\varepsilon \quad , \qquad (3.26)$$

wobei $\varepsilon>0$ eine beliebig kleine und positive Zahl ist.

### 3.3. Schätzung der störfreien Spachsignalrepräsentation

Im folgenden wird der Zusammenhang zwischen dem Entwurf der optimalen Schätzeinrichtung für die störfreien Merkmalsvektoren mit dem Bayes-Kriterium und den Approximationseigenschaften des MLP-Netzwerks aufgezeigt.

Nach Abtastung mit der Frequenz $f_a$ und Digitalisierung können die Abtastwerte des störbehafteten Sprachsignals $r(iT)$, $T=1/f_a$, als Proben der Musterfunktion eines Zufallsprozesses zu den Zeitpunkten $iT$ aufgefaßt werden. Sie werden zu Segmenten gleicher Dauer zusammengefaßt und sollen innerhalb jedes Segments als quasistationär betrachtet werden können. Nach der Segmentierung und Vorverarbeitung wird das $n$-te störbehaftete Segment durch den $m$-dimensionalen Vektor

$$x(n) = (x_1(n),x_2(n),\ldots,x_m(n))^T \qquad n=0,1,2,\ldots \qquad (3.27)$$

repräsentiert, wobei $n$ für die Segmentnummer und $x^T$ für den transponierten Vektor $x$ stehen. Der $q$-dimensionale Vektor $t(n)$ des korrespondierenden störfreien Sprachsegments sei gegeben durch

$$t(n) = (t_1(n),t_2(n),\ldots,t_q(n))^T \quad . \qquad (3.28)$$

Gesucht sind $q$ Abbildungsfunktionen $F_k$, $k=1,2,\ldots,q$, zur Schätzung der Koeffizienten von $t$ aus den Koeffizienten von $x$:

$$\begin{aligned}\hat{t}(n) &= \left(F_1(x(n)),F_2(x(n)),\ldots,F_q(x(n))\right)^T \\ &= F(x(n))\end{aligned} \quad . \qquad (3.29)$$

Im folgenden soll angenommen werden, daß die Koeffizienten von $x$ und $t$ als kontinuierliche Werte betrachtet werden können, was einer Quantisierung mit unendlich vielen Amplitudenstufen entspricht. Ihre Wahrscheinlichkeitsdichten sind daher stetige Funktionen.

**Multiple Parameterestimation mit dem Bayes-Kriterium**. Die Schätzung von $t$ ist eine Aufgabe der multiplen Parameterestimation, deren Grundlagen z. B. in

Kroschel (1973) beschrieben sind. Sie läßt sich mit Hilfe von Bild 3.4 verdeutlichen: Sprachgenerierung und Signalkodierung (Merkmalsextraktion) finden im linken Verarbeitungsblock statt. Dabei wird ein akustisches Ereignis $M$ im betrachteten Signalsegment in einem (störfreien) Vektor $t$ kodiert, wobei hier Signalquelle (Sprecher) und -kodierung (Vorverarbeitung im Spracherkenner) vereinfacht als eine Einheit betrachtet werden. Aufgrund von Störeinflüssen emittiert dieser Verarbeitungsblock jedoch anstatt der $q$-dimensionalen störfreien Vektoren $t$ lediglich $m$-dimensionale störbehaftete Vektoren $x$, deren Dimensionalität $m \geq q$ sein muß, wenn kein Informationsverlust erfolgen soll. Mit Hilfe eines vorgegebenen Optimalitätskriteriums soll nun das optimale Schätzsystems $F_B()$ entworfen werden. Hierzu werden folgende Annahmen getroffen:

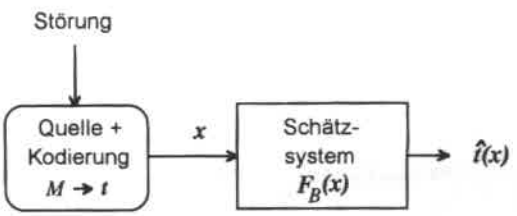



Bild 3.4: Schätzung des störfreien Sprachvektors $\hat{t}$.

1. Die a-priori-Wahrscheinlichkeitsdichte $f_t(t)$ ist bekannt.
2. Die gesuchte Schätzeinrichtung soll das Risiko $R$ minimieren, das durch den Mittelwert einer Kostenfunktion $C(e)$ mit

$$R = \mathrm{E}[C(e)] \tag{3.30}$$

gegeben ist. Die Erwartungswertbildung erfolgt über die Muster des Trainingsdatensatzes.

Der Fehlervektor $e$ zwischen den geschätzten und den tatsächlichen Werten ist durch

$$\begin{aligned} e &= \hat{t}(x) - t \\ &= F(x) - t \end{aligned} \tag{3.31}$$

gegeben. Mit Gl. (3.30) und (3.31) errechnet sich das Risiko $R$ aus

$$R = \int_{-\infty}^{\infty} \int \int_{-\infty}^{\infty} C\big(\hat{t}(x) - t\big) f_{t,x}(t,x)\, dx\, dt \tag{3.32}$$

Dabei stehen die doppelten Intergralzeichen für die $m$-fache bzw. $q$-fache Integration über $\boldsymbol{x}$ bzw. $\boldsymbol{t}$.

Zur Berechnung von $R$ benötigt man die Kostenfunktion $C(\boldsymbol{e})$. Für ihre Wahl bestehen unterschiedliche Möglichkeiten (Kroschel 1973). Wegen des Zusammenhangs mit EBP wird hier die Kostenfunktion des quadratischen Fehlers

$$\begin{aligned} C(\boldsymbol{e}) &= C\big(\hat{\boldsymbol{t}}(\boldsymbol{x}) - \boldsymbol{t}\big) \\ &= \boldsymbol{e}^T \boldsymbol{e} \\ &= \big(\hat{\boldsymbol{t}}(\boldsymbol{x}) - \boldsymbol{t}\big)^T \big(\hat{\boldsymbol{t}}(\boldsymbol{x}) - \boldsymbol{t}\big) \end{aligned} \tag{3.33}$$

verwendet. Meist interessiert man sich nicht für das Risiko in Gl. (3.32) selbst, sondern für das Schätzsystem, das $R$ minimiert. Dies kann bei bekannter Dichte $f_t(\boldsymbol{t})$ mit Hilfe des Bayes-Kriteriums entworfen werden. Dabei wird die Verbunddichte $f_{t,x}(\boldsymbol{t},\boldsymbol{x})$ in (3.32) durch

$$f_{t,x}(\boldsymbol{t},\boldsymbol{x}) = f_x(\boldsymbol{x}\,|\,\boldsymbol{t}) f_t(\boldsymbol{t}) \tag{3.34}$$

ersetzt, da $f_x(\boldsymbol{x}\,|\,\boldsymbol{t})$ gemessen werden kann und $f_t(\boldsymbol{t})$ voraussetzungsgemäß bekannt ist. Einsetzen von (3.33) und (3.34) in (3.32) sowie Minimumbestimmung von $R$ (Herleitung siehe Kroschel 1973) ergibt für $\hat{\boldsymbol{t}}$

$$\begin{aligned} \hat{\boldsymbol{t}}(\boldsymbol{x}) &= \int_{-\infty}^{\infty} \boldsymbol{t}\, f_t(\boldsymbol{t}\,|\,\boldsymbol{x})\, d\boldsymbol{t} \\ &= E\big[\boldsymbol{t}(\boldsymbol{x})\,|\,\boldsymbol{x}\big] \end{aligned} \tag{3.35}$$

$E[\boldsymbol{t}(\boldsymbol{x})\,|\,\boldsymbol{x}]$ ist der a-posteriori-Mittelwert von $\boldsymbol{t}(\boldsymbol{x})$ bei bekanntem $\boldsymbol{x}$. Die unbekannte Dichte $f_t(\boldsymbol{t}\,|\,\boldsymbol{x})$ in (3.35) kann aus den bekannten Dichten mit der Bayes-Formel

$$f_t(\boldsymbol{t}\,|\,\boldsymbol{x}) = \frac{f_x(\boldsymbol{x}\,|\,\boldsymbol{t}) f_t(\boldsymbol{t})}{f_x(\boldsymbol{x})} \tag{3.36}$$

berechnet werden. $f_x(\boldsymbol{x})$ im Nenner dient lediglich zur Normierung und hat keinen Einfluß auf die Lage des Minimums.

Der optimale Schätzwert $\hat{\boldsymbol{t}}\,(\boldsymbol{x})$ in Gl. (3.35) ist also durch einen Vektor von Funktionen $\boldsymbol{F_B}(\boldsymbol{x})$ gegeben,

$$\hat{t}(x) = E[t(x)|x] = \left(F_{B1}(x), F_{B2}(x), \ldots, F_{Bq}(x)\right)^T , \qquad (3.37)$$

der mit Hilfe des Bayes-Kriteriums berechnet wird und die Kostenfunktion des quadratischen Fehlers nach Gl. (3.33) minimiert.

Ein Vergleich mit dem Satz über die Approximationseigenschaften von Multilayer Perzeptron-Netzen (siehe Abschnitt 3.2) läßt den Schluß zu, daß die im Training gelernte Abbildungsfunktion $F()$ bei einer quadratischen Zielfunktion bis auf einen Restfehler $\varepsilon$ nach Gl. (3.26) mit der optimalen Schätzeinrichtung nach dem Bayes-Kriterium identisch ist. Mit

$$F_B(x) \equiv F_o(x) \qquad (3.38a)$$

gilt daher für die Abbildungsfunktion von MLP-Netzwerken

$$\frac{1}{L}\sum_{l=1}^{L}\left(F(x^l) - F_B(x^l)\right)^T\left(F(x^l) - F_B(x^l)\right) < \varepsilon \qquad (3.38b)$$

Daher können MLP-Netzwerke zur optimalen Schätzung der störfreien Merkmalsvektoren im Sinne des Bayes-Kriteriums eingesetzt werden. Diese Schlußfolgerung gilt mit folgenden Einschränkungen:

1. Die Netztopologie muß für das vorliegende Problem geeignet gewählt werden. Ein Verfahren zur Bestimmung der optimalen Topologie ist jedoch nicht bekannt.
2. Während des Trainings muß das globale Minimum der Fehlerkurve im Gewichtsraum gefunden werden. Diese stellt bei $G_{MLP}$ voneinander unabhängigen Gewichten anschaulich ein Gebirge im $G_{MLP}+1$-dimensionalen Raum dar [8]. Außer der vollständigen Suche ist jedoch kein Verfahren zur sicheren Bestimmung des globalen Minimums bekannt.

Trotz dieser Einschränkungen können zur Realisierung einer ausreichend guten Näherungslösung für das jeweilige Problem meist vernünftige Annahmen für Netztopologie und Trainingsparameter gemacht werden, wie aus den experimentellen Ergebnissen ersichtlich ist (vgl. 5. Kapitel).

---

[8] Zur Berechnung der Zahl der Gewichte siehe Gl. (6.16).

### 3.4. Konzepte zur neuronalen Störreduktion

Basierend auf den beschriebenen Abbildungseigenschaften von MLP-Netzwerken werden in diesem Abschnitt neuronale Ansätze zur Reduktion der unterschiedlichen Störungsarten vorgestellt.

#### 3.4.1. Einmaladaptive Netzwerke

Bei einmaladaptiven Netzwerken wird die Abbildungsfunktion vor Beginn der Testphase aus dem Trainingsmaterial gelernt. Instationaritäten, die nach Trai-

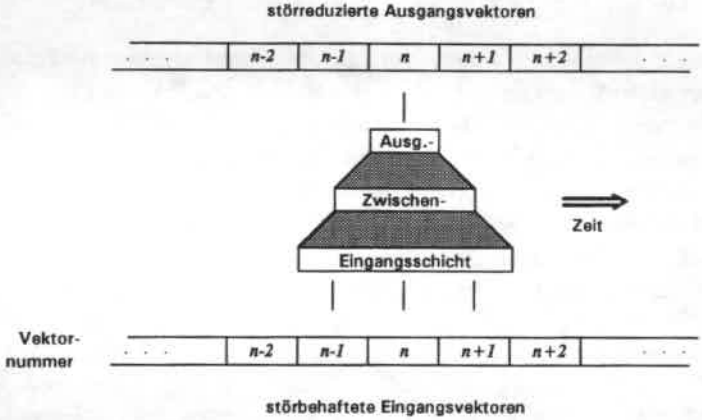



Bild 3.5: Prinzip der fortlaufenden Störreduktion aufeinanderfolgender Sprachsegmente nach vorausgehender Trainingsphase.

ningsabschluß auftreten, können damit nicht behandelt werden. Zu den Anwendungen gehört die Reduktion von **additiven Hintergrundgeräuschen** sowie von **Kanalverzerrungen**. Das Prinzip zeigt Bild 3.5 [9]: durch Anwendung der trainierten Abbildungsfunktion auf Vektoren aus dem kontinuierlichen Datenstrom am Eingang des Netzwerks werden vektorweise störreduzierte Ausgangsvektoren erzeugt. Nach Reduktion des $n$-ten Vektors wird im darauffolgenden Verarbei-

[9] Bild 3.5 zeigt die schematische Darstellung des MLP aus Bild 3.2, das für diese Illustration auf drei $m$-dimensionale Eingangsvektoren und einen $q$-dimensionalen Ausgangsvektor erweitert wurde. Diese Darstellungsweise wird in der weiteren Arbeit verwendet.

tungsschritt der $n+1$-te Vektor am Eingang störreduziert usw. Dieses einmaladaptive Verfahren hat den Vorteil, daß keine explizite Segmentierung in Wort- bzw. Pausenabschnitte des Signals erforderlich ist. Die im Bild schematisch gezeigte Berücksichtigung von Kontextinformation am Netzwerkeingang (gleichzeitiges Anlegen der Vektoren zum Zeitpunkt $n$-1, $n$ und $n$+1) führt zu besseren Ergebnissen und wird im Abschnitt 5.2 diskutiert.

Parallel zur Störreduktion könnten mit diesem Ansatz zusätzliche Verarbeitungsschritte aus der Signalvorverarbeitung vorgenommen werden, soweit sie sich als Abbildungsaufgaben formulieren lassen. Beispiele hierfür sind die Hauptachsentransformation (vgl. Abschnitt 4.2.3) und die Berechnung der zeitlichen Ableitungen von Merkmalskoeffizienten. Dies wird in Abschnitt 5.6 untersucht.

In Kapitel 2 wurden die Auswirkungen des **Lombardeffektes** aufgrund von statistischen Untersuchungen z. T. als systematische Veränderungen von Signalparametern beschrieben, für die derzeit kein Modell bekannt ist. Da zu den Parameterveränderungen auch zeitliche Schwankungen der Lautdauer gehören, müssen für ihre Korrektur korrespondierende Signalabschnitte zwischen lombardfreien und lombardbehafteten Daten gefunden werden. Gelingt dies, kann aus einer statistisch relevanten Menge von Trainingsdaten eine Abbildung trainiert werden, die eine "Entlombardisierung" der störbehafteten Eingangsdaten erlaubt. Dazu ist vor der Abbildung eine Zeitanpassung mit *Dynamic Time Warping* (DTW) notwendig (Bild 3.6), das in dieser Arbeit zur Worterkennung eingesetzt wird (vgl. 4. Kapitel).

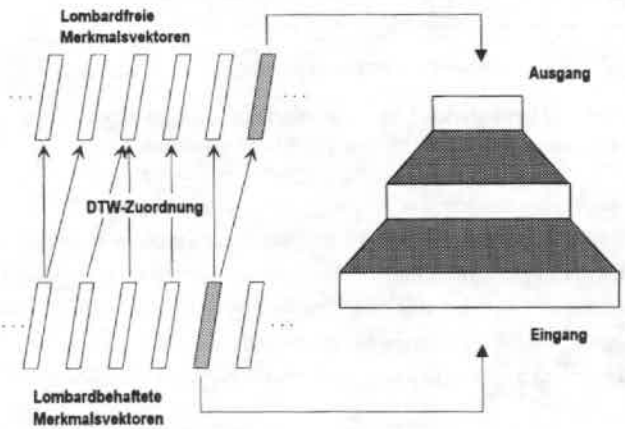



Bild 3.6: Reduktion des Lombardeffekts durch Abbildung der Eingangsvektoren nach vorausgehender dynamischer Zeitanpassung.

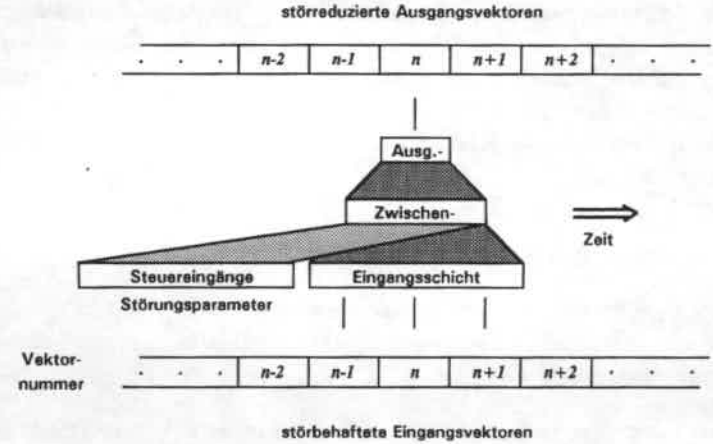



Bild 3.7: Adaptive vektorweise Geräuschreduktion mit Steuereingängen.

### 3.4.2. Kontinuierliche Netzwerkadaption

Zur kontinuierlichen Netzwerkadaption wird das Abbildungsverhalten des Netzwerks durch Steuerparameter beeinflußt, die an zusätzlichen Eingangsknoten anliegen und so die Modifikation der Abbildungsfunktion nach Trainingsabschluß ermöglichen (vgl. Bild 3.7). Dieser Ansatz zur Steuerung der Netzwerkeigenschaften wurde von Levin (1990) für die Unterscheidung der Zustände eines Hidden Markov Modells zur Phonemklassifikation untersucht. Eine experimentelle Untersuchung dieses Ansatzes für die Geräuschreduktion erfolgt in Kapitel 7.

Geeignete Steuerparameter sind getrennt nach Störungsart in Tabelle 3.1 angegeben. Beispiele sind Geräuschparameter zur Charakterisierung des Hintergrundgeräuschs, die Koeffizienten der Übertragungsfunktion zur Adaption an einen zeitveränderlichen Übertragungskanal oder auch an Streßparameter, die die

Tabelle 3.1: Steuerinformation am Netzwerkeingang in Abhängigkeit von der Störungsart.

| | Störungsart | | |
|---|---|---|---|
| | additives Geräusch | Kanalverzerrungen | Lombardeffekt |
| Steuerparam. | Geräuschparameter | Koeffizienten der Übertragungsfunktion | Streßparameter |

mentale Belastung des Sprechers charakterisieren. Die Änderungsgeschwindigkeit der Steuerparameter kann dabei andere Zeitkonstanten aufweisen als die des Nutzsignals. Die Aktualisierung der Abbildungsfunktion kann entweder kontinuierlich oder von Zeit zu Zeit, beispielsweise in den Sprachpausen, erfolgen. Aus der Literatur bekannte Ansätze zur Realisierung dieser Aufgabe werden in Abschnitt 3.6 diskutiert.

### 3.5. Bewertungskriterien

Im Folgenden werden aus der Literatur bekannte Kriterien zur Optimierung der Netzwerkparameter sowie zur Bewertung der Leistungsfähigkeit des Netzwerks nach Trainingsabschluß diskutiert.

**Mittlerer quadratischer Fehler**. Die zum Netztraining verwendete Kostenfunktion liefert im Minimum einen Restfehlerwert, der als Maß für die Wirksamkeit des Geräuschreduktionsverfahrens dienen kann. Für die Kostenfunktion des mittleren quadratischen Fehlers nach Gl. (3.9) kann der Vergleich zweier Netzwerke mit Hilfe des MSE durchgeführt werden, indem die Restfehlerwerte im Minimum verglichen werden. Gilt für die Restfehler zweier unterschiedlicher Netzwerke

$$MSE_{min1} < MSE_{min2} \quad , \tag{3.39}$$

so weist das zum $MSE_{min1}$ gehörige die bessere Geräuschreduktionsleistung auf.

Die relative Verbesserung gegenüber der Situation ohne Geräuschreduktion wird mit dem auf den ursprünglichen Wert bezogenen relativen quadratischen Restfehler angegeben:

$$MSE_{rel} = \frac{MSE_{min}}{MSE_{stör}} \quad . \tag{3.40}$$

$MSE_{stör}$ wird dabei zwischen den geräuschbehafteten und den geräuschfreien Vektoren des Datensatzes berechnet. Schließlich kann man den logarithmierten relativen quadratischen Restfehler

$$10\,lg\,MSE_{rel} = 10\,lg\,\frac{MSE_{min}}{MSE_{stör}} \tag{3.41}$$

in Dezibel messen. Die Fehlermaße nach Gln. (3.39), (3.40) und (3.41) können sowohl im Trainings- als auch im Verifikationsdatensatz verglichen werden, wobei die Generalisierungsfähigkeit sowie Übertrainingseffekte zu berücksichtigen sind (siehe Abschnitt 3.2).

**Worterkennungsrate.** Worterkennungstests stellen sicher, daß die Optimierung der Geräuschreduktionsstufe sich an der Anwendung orientiert. Obwohl dieses Maß vom verwendeten Erkennungssystem abhängig ist, wird es zur Evaluierung der Geräuschreduktionsstufe sowie des Gesamtsystems einschließlich der Merkmalsextraktionsstufe bevorzugt eingesetzt.

Grundsätzlich unterscheidet man in der Spracherkennung zwischen Verwechslungs-, Einfügungs- und Auslassungsfehlern. Da Einfügungen und Auslassungen auf Wortebene bei Isoliertworterkennung von der Qualität des Sprachpausendetektors (siehe Abschnitt 4.1) abhängen, sollen hier nur Verwechslungs- oder Substitutionsfehler betrachtet werden. Die Worterkennungsrate $r_W$ ist dann durch

$$r_W = \frac{W_{richtig}}{W_{gesamt}} \cdot 100 \ [\%] \tag{3.42}$$

gegeben, und die Wortfehlerrate $f_W$ berechnet sich mit $r_W$ aus

$$f_W = 100\% - r_W \tag{3.43}$$

Die eigentliche Aufgabe der Geräuschreduktionsstufe ist die Minimierung von $f_W$. Da eine entsprechende Formulierung der Zielfunktion für das Training auf Schwierigkeiten stößt, wird hier ein anderer Weg beschritten. Voruntersuchungen haben gezeigt, daß $f_W$ und $MSE_{min}$ stark korreliert sind. Daher wird zum Netzwerktraining zunächst $MSE_{min}$ als Gütemaß herangezogen; die quantitativen Verbesserungen für die Spracherkennung werden dann mit Hilfe von $f_W$ bzw. $r_W$ gemessen.

**Alternative Zielfunktionen.** Anstatt des $MSE$ nach Gl. (3.9) werden für manche Aufgaben andere Zielfunktionen mit speziellen Eigenschaften eingesetzt; hierzu gehören die *Cross Entropie* und der *McClelland Error*, die beide große Abweichungen der Ausgangsaktivierungen von den Trainingszielen stärker gewichten als die quadratische Fehlerfunktion. Ihre Eigenschaften sowie die Ergebnisse für eine Phonemklassifikationsaufgabe wurden von Hampshire und Waibel (1990)

beschrieben. Für binäre Trainingsziele (d. h. die $t_k$ nehmen entweder den Wert 0 oder 1 an) lautet die Cross Entropie-Zielfunktion

$$E_{CrossEntropie} = -\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{q}\left(t_k^l \log\left(y_k^l\right) - \left(1-t_k^l\right)\log\left(1-y_k^l\right)\right) \qquad (3.44)$$

und die McClelland-Fehlerfunktion

$$E_{McClelland} = -\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{q}\log\left(1-\left(t_k^l - y_k^l\right)^2\right) \qquad (3.45)$$

Für Klassifikationsaufgaben wurden mit diesen Zielfunktionen gute Ergebnisse berichtet (Hild and Waibel 1993c; Tebelskis and Waibel 1993). Eine weitere Zielfunktion ist *Softmax*; sie wird vor allem im Zusammenhang mit der Schätzung von Klassenwahrscheinlichkeiten verwendet, vgl. Manke et al. (1995).

Die Zielfunktion des *Classification Figure of Merit* (CFM, Hampshire and Waibel 1990) wurde ebenfalls für Klassifikationsaufgaben entwickelt und basiert auf der Maximierung von Differenzen zwischen der Aktivierung des korrekten und den Aktivierungen aller übrigen Ausgangsknoten. Sie weist gegenüber dem MSE ein verbessertes Monotonieverhalten auf. Dies bedeutet, daß für einzelne Musterpaare berechnete, diskrete Fehlerwerte zusammenhängenden (*monotonen*) Wertebereichen von korrekten bzw. fehlerhaften Klassifikationsergebnissen direkt zugeordnet werden können. Hampshire und Waibel (1990) zeigten, daß dies ist bei einer MSE-Zielfunktion nicht immer möglich ist. Von Hampshire und Waibel (1990) sowie Hild und Waibel (1993b und 1993c) wurden mit der CFM-Zielfunktion gute Klassifikationsergebnisse erreicht. Gegenüber dem MSE und der Cross Entropie-Funktion ist jedoch ein erhöhter Trainingsaufwand notwendig.

**Optimierung der Geräuschreduktionsleistung.** MLP-Netzwerke, die den mittleren quadratischen Fehler einer repräsentativen Trainingsstichprobe minimieren, stellen optimale Schätzsysteme im Sinne des Bayes-Kriteriums dar (vgl. Abschnitt 3.3). Daher werden hier für das Netzwerktraining *MSE*-basierte Zielfunktionen zur erkennerunabhängigen Optimierung und Bewertung der Geräuschreduktionsleistung verwendet. Da für die Worterkennungsexperimente ein nichtneuronales Verfahren verwendet wird (siehe Kapitel 4), erfolgt die Evaluierung des Gesamtsystems einschließlich Merkmalsextraktion mit Hilfe der Worterkennungs- bzw. Wortfehlerraten nach Gln. (3.42) und (3.43).

**Aufwandsbeurteilung.** Für eine Echtzeitimplementierung sind neben der Funktionalität vor allem Implementierungsaufwand, benötigte Rechenzeit sowie der Speicherplatzbedarf wichtig. Da aus den zur Simulation benötigten Systemressourcen nicht immer - oder nur mit großem Aufwand - exakte Angaben für eine Echtzeitumgebung gemacht werden können, werden neben den absoluten Rechenzeiten oft relative Vergleichszahlen zum Vergleich verschiedener Algorithmen benutzt. Dies sind beispielsweise die Zahl der verdeckten Knoten, die zum Training notwendige Zahl von Iterationen und die Zahl der durchgeführten Gewichtsmodifikationen. Da wegen der unterschiedlichen Netzmodelle Vorsicht beim direkten Vergleich dieser Zahlen geboten ist, werden sie in den jeweiligen Abschnitten zusammen mit den Eigenschaften der untersuchten Netzwerke diskutiert, vgl. z. B. Abschnitt 6.3.

## 3.6. Verwandte Ansätze

Nachfolgend wird ein Überblick über Literaturstellen gegeben, die den Stand der Technik beschreiben und in Beziehung zu den in dieser Arbeit untersuchten Ansätzen stehen.

### 3.6.1. Geräuschreduktion mit einmaladaptiven Netzen

Erste Arbeiten zur neuronalen Geräuschreduktion wurden von Tamura und Waibel (1988) und Tamura (1989) auf Basis der Abtastwerte des **Zeitsignals** mit Hilfe eines vierstufigen MLP-Netzes durchgeführt. Analysen des Netzwerkverhaltens ergaben, daß selbst nichttrainierte Geräuschanteile durch die gelernte Abbildungsfunktion unterdrückt wurden, während Sprachanteile weitgehend unverändert blieben. Aufgrund von Hörtests (*Auditory Preference Test*) und visueller Auswertung von Spektrogrammen wurde von Verbesserungen gegenüber Spektralsubtraktionsexperimenten berichtet.

Wegen des geringeren Trainingsaufwands aufgrund einer kompakteren Netztopologie basieren neuere Ansätze zur neuronalen Geräuschreduktion meist auf den **Merkmalsvektoren**: von Sorensen (1991) und Sorensen und Hartmann (1991) wurden erhebliche Steigerungen der sprecherunabhängigen Erkennungsraten für Ziffern (von 14 % auf 79 % bei 0 dB Signal-zu-Rausch-Verhältnis) bei rechneraddiertem Cockpitgeräusch berichtet; in Barbier und Chollet (1991) wird für Spracherkennung in Kraftfahrzeugumgebung auf Verbesserungen auch in sprecherübergreifenden Tests hingewiesen.

Die bisher genannten Verfahren sind einkanalig und bieten nach initialem Training der Abbildungsfunktion keine weitere Adaptionsmöglichkeit mehr. Bis auf eine Ausnahme (Sorensen and Hartmann 1991) erfolgte die Netzgenerierung experimentell, was zwei Entwicklungsschritte notwendig macht: erstens die Festlegung der Topologie und zweitens das Lernen der Gewichte. Eine wichtige Rolle spielt die Wahl der Ein- und Ausgangssignalrepräsentation. Mit Zusatzinformation am Netzwerkeingang kann das Abbildungsverhalten verbessert werden, wenn sie mit dem Fehlersignal korreliert ist. Hierzu gehören Verbesserungen durch Kontextinformation (Huang 1992, für Sprecheradaption; Trompf 1992a, für Geräuschreduktion) und mit zusätzlichen Merkmalskoeffizienten am Netzeingang (Trompf and Hackbarth 1993). Bei entsprechender Aufbereitung der Trainingsdaten können Eingangs- und Ausgangssignale unterschiedlichen Repräsentationen entstammen und so zusätzlich zur Geräuschreduktion Vorverarbeitungsschritte mitgelernt werden (Trompf et al. 1993).

### 3.6.2. Automatische Netzgenerierungsverfahren

Automatische Netzgenerierungsverfahren helfen, die zeitaufwendige, vorwiegend experimentell geprägte Suche nach problemangepaßten Netzwerkmodellen zu verkürzen, sind für zahlreiche Anwendungsgebiete in Untersuchung und lassen sich in Algorithmen mit konstruktiver und mit destruktiver Lernstrategie aufteilen. Wegen der Vielzahl der Systemparameter und der damit verbundenen Versuchswiederholungen ist auch bei der Entwicklung eines geeigneten Geräuschreduktionsnetzwerks der Einsatz eines automatischen Netzgenerierungsverfahrens wünschenswert.

Bei **konstruktiven Verfahren** werden Minimalnetze schrittweise durch automatisches Anfügen zusätzlicher Teilstrukturen erweitert und trainiert. Dies geschieht regelbasiert und iterativ solange, bis im Sinne einer vorgegeben Zielfunktion keine weitere Verbesserung eintritt oder ein anderes Abbruchkriterium erfüllt ist. Die Formulierung der Generierungsregeln ist meist abhängig von der betrachteten Netzwerkstruktur. Beispiele für konstruktive Netzgenerierungsverfahren sind:

- der *Cascade Correlation*-Lernalgorithmus (Fahlman and Lebière 1989), bei dem ausgehend von einer linearen Perzeptron-Struktur iterativ nichtlineare Knoten angefügt und trainiert werden, bis die vorgegebene Fehlerfunktion minimiert wurde oder keine weitere Verbesserung feststellbar ist. Die Verbindungsstruktur ist kompakter als beim Multilayer Perzeptron; abhängig von der

zeitlichen Reihenfolge ihrer Installation im Netzwerk existieren Verbindungen zwischen den Eingängen später angefügter verdeckter Knoten und den Ausgängen ihrer Vorgängerknoten. Sorensen und Hartmann (1992) haben gezeigt, daß dieser Netzwerktyp die Multilayer Perzeptron-Struktur als Teilmenge enthält. Eine Variante stellt *Recurrent Cascade Correlation* (Fahlman 1990) dar, dessen hauptsächlicher Unterschied zu Cascade Correlation in lokal rückgekoppelten Verbindungen von den Aus- zu den Eingängen der verdeckten Knoten besteht und das daher ein neuronales Modell mit Gedächtnis bildet.

- das *Resource Allocating Network* (Platt 1991), bei dessen Generierung am Netzwerkeingang auftretende Beobachtungen in den Mittelwertsvektoren neu anzufügender verdeckter Knoten mit gaußförmiger Aktivierungsfunktion gespeichert werden. Die Netzwerkgenerierung ist abgeschlossen, wenn sich alle Trainingsbeispiele ausreichend genau durch Kombinationen der gespeicherten Beobachtungen approximieren lassen. Bei der weiterentwickelten Version des *Enhanced Resource Allocating Network* wurde das ursprüngliche Least Mean Square-Trainingsverfahren durch ein bei Kalman-Filtern verwendetes Verfahren ersetzt, was für Funktionsapproximation eine kompaktere Topologie bei gleichzeitig schnellerer Konvergenz und kleinerem Restfehler zur Folge hat (Kadirkamanathan and Niranjan 1993).

- die beiden komplementären Verfahren *Automatic Structure Optimization* und *Automatic Validation Analyzing Control System*, die von Bodenhausen und Waibel (1993) sowie Bodenhausen (1994) zur Bestimmung der Systemparameter (Kontextfenster, Zahl der verdeckten Knoten und Modellzustände) eines *Multi State Time Delay Neural Network* vorgeschlagen wurden. Automatic Structure Optimization steuert die Netzwerkgenerierung auf Basis der Verwechslungsmatrix der Trainingsdaten; das Automatic Validation Analyzing Control System führt einen Generalisierungstest auf Basis der Differenzen zwischen den Verwechslungsmatrizen aus Trainings- und Validierungsdaten durch. Für die Erkennung gesprochener Buchstaben wurde mit diesen Verfahren eine Verbesserung der Erkennungsrate gegenüber einem handoptimierten System von 85 % auf über 92 % erreicht.

Bei **Verfahren mit destruktiver Strategie** wird eine vorläufige, eher überdimensionierte Netzstruktur festgelegt und trainiert. In einer späteren Trainingsphase werden aufgrund vorgegebener Regeln Verbindungen zusammengelegt bzw. eli-

miniert, um eine verbesserte Generalisierungsfähigkeit unter gleichzeitiger Optimierung des Netzwerks zu erreichen. Beispiele hierfür sind die Arbeiten zu *Optimal Brain Damage* von Le Cun et al. (1989) sowie *Soft Weight-Sharing* von Nowlan und Hinton (1991).

### 3.6.3. Adaption an instationäre Signalumgebungen

Eine Netzwerkadaption kann abhängig von instationären Geräuschkomponenten oder vom phonetischen Kontext der Sprachkomponente erfolgen. Dies kann entweder durch Umschaltung zwischen mehreren vortrainierten Abbildungen, durch schnelle Adaption vorhandener oder durch inkrementelles Anfügen neuer Teilstrukturen geschehen.

Eine **Adaption mit Hilfe von Steuerinformation** wurde für Geräuschreduktion erstmals von Tamura und Nakamura (1990) für die Modifikation der Ausgangsgewichte abhängig vom Phonemkontext untersucht, was zu einer Verringerung des Abbildungsfehlers bei additivem weißem Rauschen führte.

Das *Codeword-Dependent Neural Network* (Huang 1992) besteht aus mehreren vortrainierten Netzwerken zur Realisierung der Abbildungsfunktion eines fremden Sprechers auf einen Referenzsprecher; zwischen den unterschiedlichen Netzwerken wird abhängig vom Wert des Eingangsvektors umgeschaltet. Hierdurch konnte in Worterkennungsexperimenten die Fehlerrate im Vergleich zur Sprecheradaption mit einem einzigen Netzwerk von 6,8 % auf 5 % verringert werden. Eine Erweiterung dieses Ansatzes mit vortrainierten Abbildungen mehrerer Sprecher auf einen Referenzsprecher wurde von Hild und Waibel (1993a und 1993b) beschrieben. In einer mit *Tuning-In* bezeichneten Adaptionsphase werden dabei die Charakteristika eines neuen Sprechers aus den intern gespeicherten Abbildungen (*Internal Speaker Models)* mit Hilfe eines Codeworts zusammengesetzt.

Eine schnelle Adaption der Abbildungsfunktion durch Steuerparameter am Netzwerkeingang kann mit dem von Levin (1990) vorgeschlagenen *Hidden Control Neural Network* (HCNN) vorgenommen werden. Damit wurden für die Modellierung einzelner Zustände in Wortmodellen parallele nichtlineare Prädiktoren realisiert, zwischen denen mit Hilfe von Steuerinformation gewechselt werden kann. Mit diesem Ansatz wurde für die Erkennung zusammenhängend gesprochener

Ziffern mit Aufnahmen von elf männlichen Sprechern eine durchschnittliche Erkennungsrate von 99,3 % erreicht.

In einer ersten Veröffentlichung von Teilen der vorliegenden Arbeit zur geräuschparameterbasierten Netzwerkadaption in Sprachpausen (Trompf et al. 1994) wurde die Auswahl geeigneter Steuerparameter auf Basis eines Korrelationsmaßes beschrieben. In sprecherabhängigen Worterkennungstests mit rechneraddierten Störgeräuschen wurden mit zehn Steuerparametern Verbesserungen der Erkennungsrate von einigen Prozent im Vergleich zur Situation mit einem vorab trainierten Netzwerk ohne weitere Adaptionsmöglichkeit erreicht.

Das *HCNN-CDF* (Petek et al. 1992) stellt eine Erweiterung des HCNN zur Prädiktion von Signalsegmenten unter Berücksichtigung des Signalkontextes dar. Neben Steuereingängen für die einzelnen Phonem- bzw. Wortzustände wurden zusätzliche Eingänge für den phonetischen Kontext eingeführt. Hiermit konnten für eine Worterkennungsaufgabe Verbesserungen gegenüber der Verwendung von parallelen, phonemabhängig trainierten *Linked Predictive Neural Networks* (Tebelskis et al. 1991) und auch gegenüber HCNN-Netzwerken ohne Kontexteingänge erzielt werden.

**Adaption durch inkrementelles Lernen** kann u. a. mit Hilfe von konstruktiven Netzgenerierungsverfahren durchgeführt werden. Von Kadirkamanathan und Niranjan (1993) wurde die Prädiktion chaotischer Zeitreihen durch inkrementelles Erweitern der trainierten Abbildung mit Hilfe des *Enhanced Resource Allocating Network* durchgeführt. Je nach Anwendung kann diese Technik jedoch zu kontinuierlichem, unbegrenztem Anwachsen der Netztopologie führen.

In Arbeiten von Waibel et al. (1989) zum Thema *Modularität* in neuronalen Netzwerken wurden unterschiedliche Konzepte zur Integration spezialisierter Netzwerke untersucht. Am Beispiel der Phonemerkennung wurde gezeigt, wie mehrere auf bestimmte Phonemklassen spezialisierte Netzwerke mit Hilfe von frei trainierbaren Neuronen (*Connectionist Glue*) sowie inkrementellem Training der Gesamtstruktur (*All-Net Fine Tuning*) zusammengeschaltet werden können; dies führte zu einer schrittweise Steigerung der Erkennungsrate von 60,5 % auf 98,6 %.

Beim *Meta-Pi-Network* (Hampshire and Waibel 1989) werden die Ausgangsaktivierungen spezialisierter Teilstrukturen mit Hilfe einer übergeordneten Superstruktur verbunden, die aus multiplikativen Rechenelementen (*Meta-Pi Gate*

*Weights*) zur Gewichtung der einzelnen Anteile an der Gesamtaktivierung besteht. Die Teilstrukturen dienen zur Behandlung unterschiedlicher Eingangsquellen (z. B. Sprecher); die multiplikativen Gewichtungen werden in einer Adaptionsphase gelernt, in deren Verlauf das Abbildungsverhalten des Gesamtnetzwerks aus der Überlagerung der bereits vortrainierten Teilstrukturen zusammengesetzt wird. Dies wurde von Hampshire und Waibel am Beispiel der sprecheradaptiven Phonemklassifikation gezeigt, wobei Äußerungen neuer Sprecher durch Kombination sprecherabhängig vortrainierter Klassifikatoren erfolgreich erkannt wurden. Das Konzept von *Gate Weights* zur gezielten Aktivierung von Teilstrukturen eines übergeordneten Netzwerks wurde von Jain (1991) in einem neuronalen Parser zur Markierung von Satzteilen für die weitere Verarbeitung eingesetzt.

### 3.7. Ansätze in der vorliegenden Arbeit

Zur Lösung der Geräuschreduktionsaufgabe sind drei Schwerpunkte für die vorliegende Arbeit von besonderer Bedeutung:

1. Ein grundsätzlicher experimenteller Nachweis für die Funktionsfähigkeit des Ansatzes, die Aufgabe mit einer nichtlinearen Abbildungsfunktion auf Basis von neuronalen Netzwerken zu lösen (Kapitel 5).

2. Die Suche nach einer geeigneten Netztopologie. Hierzu wird ein Vergleich zwischen den automatischen Netzgenerierungsverfahren *Cascade Correlation* und *Resource Allocating Network* sowie einem handoptimierten Multilayer Perzeptron durchgeführt (Kapitel 6).

3. Schnelle Adaption an veränderliche Umgebungsgeräusche, die nicht notwendigerweise à priori bekannt sein müssen. Da für eine spätere Echtzeitrealisierung ein Neutraining bzw. eine (rechenaufwendige) Adaptionsphase des Netzwerks bei jeder Änderung der Umgebungsgeräusche zu aufwendig erscheint, wird hier ein HCNN-basierter Ansatz zur Netzwerkadaption bevorzugt (Kapitel 7).

# 4. Signalvorverarbeitung und Testbett

## 4.1. Datenbasis

In realer Anwendungsumgebung findet eine additive Überlagerung von Sprach- und Geräuschsignal statt, d. h. die Worterkennung wird auf Basis des gestörten Signals durchgeführt. Für die experimentellen Untersuchungen werden Trainings- und Testdaten durch Addition der digitalisierten Sprach- und Geräuschsignale auf dem Rechner erzeugt, was den Vorteil reproduzierbarer Versuchsbedingungen bietet.

Messungen unter realen Anwendungsbedingungen helfen bei der Auswahl der Trainings- und Testdaten: in einer Büroumgebung wurde ein Signal-zu-Rausch-Verhältnis (SNR) von 12 dB bei einem Telefongespräch mit Freisprechen und einem Nadeldrucker in ca. 2 m Entfernung gemessen. Bei Autofahrten wurden im Innenraum einer Reiselimousine bei geschlossenen Fenstern folgende SNR-Werte gemessen: im Stadtverkehr bei 50 km/h 8-10 dB, bei einer Überlandfahrt mit 100 km/h 4-7 dB und bei einer Autobahnfahrt mit 180 km/h 0-3 dB. Die Meßwerte wurden über einen Zeitabschnitt von ca. 60 s und über die Sprachaufnahmen von mehreren Sprechern gemittelt. Diese Meßergebnisse wurden als Anhaltspunkt für die Generierung der Trainings- und Testdaten aus getrennten Sprach- und Geräuschdatenbasen berücksichtigt.

**Sprachdaten**. Als Sprachdaten wurden vorhandene Aufnahmen (Angleys 1991) mit isoliert gesprochenen Wörtern von je fünf männlichen und fünf weiblichen Sprechern verwendet. Sie enthält pro Sprecher fünf Wiederholungen eines 30 Wörter umfassenden deutschen Vokabulars, das aus 20 Wörtern zur sprachgesteuerten Bedienung eines Textverarbeitungssystems und den zehn Ziffern besteht. Da solche Geräte häufig in Büroumgebung betrieben werden, wird dieser Wortschatz im folgenden als *Bürowortschatz* bezeichnet. Die Wortliste befindet sich im Anhang A.1.

Die Sprecher wurden angewiesen, Pausen von mindestens 0,5 s Dauer zwischen den Wörtern zu lassen. Dies ist aus folgenden Gründen notwendig:

Tabelle 4.1: Sprecherliste und Zuordnung der Wiederholungen pro Sprecher zu den Teildatensätzen.

| | | Teildatensätze und Aufnahmen | | |
|---|---|---|---|---|
| | Sprecherliste | Training | Verifikation | Test |
| männlich | DIE, KRA, MIC, THO, UWE | 1,2 | 3 | 4,5 |
| weiblich | CAR, EDD, HAC, SAB, SUS | 1,2 | 3 | 4,5 |

- Vermeidung von Koartikulationseffekten.
- Durchführung einer automatischen Sprachpausendetektion.
- Schätzung der Pausenenergie zur SNR-Bestimmung.
- Extraktion von Geräuschparametern zur Netzwerkadaption (vgl. 7. Kapitel).

Um das häufig beobachtbare Absinken der Betonung am Ende einer Äußerung zu vermeiden, wurde die Wortliste um das Wort *"dummy"* erweitert. Über die Steuerung der Sprechweise hinaus hat diese Maßnahme keine Bedeutung.

**Aufnahmebedingungen**. Die Sprachaufnahmen wurden in einem Büroraum von ca. 14 $m^2$ Grundfläche mit büroüblicher Einrichtung durchgeführt. Neben dem Schließen von Fenstern und Türen wurden keine Maßnahmen zur Dämpfung von Umweltgeräuschen (Verkehrsgeräusche sowie Geräusche aus benachbarten Büros und Fluren) getroffen. Die Sprecher saßen an einem Schreibtisch, das Aufnahmemikrofon befand sich in ca. 80 cm Entfernung vom Sprechermund.

Zur Wandlung des akustischen Signals wurde ein Kondensator-Backelektret Mikrofon (Datenblatt AKG Q400T, 1889) mit Hypercardioid-Charakteristik verwendet, das zusammen mit einem Telefongehäuse (SEL 1074) als Grenzfläche eine Freisprecheinrichtung realisiert. Der Übertragungsbereich des Mikrofons reicht von 200 Hz bis 8 kHz (3 dB-Grenzfrequenzen). Nach Verstärkung im integrierten Vorverstärker wurde das Signal mit einem DAT-Rekorder (Sony TCD-D10) aufgezeichnet. Das vom Rekorder wiedergegebene Analogsignal wurde mit einem PC-Board (Hörmann et al. 1993) mit Hardware-Filter tiefpaßgefiltert, abgetastet und digitalisiert (3 dB-Grenzfrequenz $f_g$=3,4 kHz, Auflösung 16 bit, Abtastfrequenz $f_a$=8 kHz), vgl. Datenblatt Motorola DSP56ADC16.

**Speicherung in Teildatensätzen.** Die digitalisierten Sprachdaten wurden als Dateien mit je einer Aufnahme der 30 Wörter auf einer Magnetspeicherplatte abgelegt. Die fünf Wiederholungen pro Sprecher wurden in drei Teildatensätze aufgeteilt (vgl. Tabelle 4.1). Dies ist zur Trennung von Trainings- und Testdaten so-

wie zur Durchführung der Generalisierungstests mit Verifikationsdaten (vgl. Abschnitt 3.2) notwendig. Die einzelnen Sprecher werden durch drei Buchstaben umfassende Abkürzungen identifiziert. Tabelle 4.1 zeigt die Sprecherliste nach Geschlecht getrennt sowie die Zuordung der Wiederholungen zu den Teildatensätzen. Die mittlere Dauer einer Wiederholung einschließlich Pausen beträgt 68 s, die längste Aufnahme dauert 97 s.

Das über jeweils eine Wiederholung der 30 Wörter gemittelte Signal-zu-Rausch-Verhältnis lag zwischen 27 und 32 dB. Diese Aufnahmen werden in der weiteren Arbeit als *geräuschfrei* oder auch als *Originalaufnahmen* bezeichnet. Anhand dieser Aufnahmen erfolgte die Wortgrenzendetektion halbautomatisch; die Hypothesen eines realen Sprachpausendetektors wurden manuell korrigiert, so daß für die Experimente ein idealer Sprachpausendetektor angenommen wird.

**Geräuschdaten**. In der Geräuschdatenbank befinden sich Aufnahmen aus zweierlei Herkunft:

1. Aufnahmen eines Nadeldruckergeräusches (Kurzbezeichnung: *Drucker*) sowie Aufnahmen im Betriebsraum eines Rechenzentrums; letztere setzt sich aus dem Summensignal einzelner überlagerter Geräuschsignale aus unterschiedlichen Quellen zusammen (Kurzbezeichnung: *Rechnerraum*). Hierzu gehören Betriebsgeräusche mehrerer VAX- und Sun SPARC-Rechenanlagen, Plattenspeicher, Protokolldrucker, Bandlaufwerke und Klimaanlage. Das Summensignal wurde mit einem Studio-Kondensatormikrofon Beyer MC 723 mit Cardioid-Charakteristik und eingebautem Vorverstärker mit einem Übertragungsbereich von 40 Hz bis 20 kHz (Datenblatt Beyer MC 723, 1990) aufgenommen. Aufzeichnung, Filterung und Digitalisierung der Aufnahmen erfolgte wie bei den Sprachaufnahmen und wurden von Eckhardt (1992) beschrieben.

2. Aufnahmen aus einer käuflichen Datenbasis auf CD-ROM-Datenträger aus dem ESPRIT-Projekt *SUNSTAR*. Aus verarbeitungstechnischen Gründen (Speicherplatz und Rechenzeit für die Simulationen) mußte aus der Vielzahl von Geräuschaufnahmen eine Auswahl getroffen werden. Wegen des Anwendungsszenarios für die zu entwickelnden Verfahren (siehe 2. Kapitel) wurden fünf Aufnahmen ausgewählt, die in einer Mobilfunkumgebung auftreten können. Diese umfassen Aufnahmen in einer Bahnhofshalle, einer Gaststätte, eines IBM-Matrixdruckers, einer Spülmaschine und von Straßenbauarbeiten. Der vollständige Inhalt ist aus dem Datenblatt (SUNROM-1) zu ersehen. Wegen der ursprünglichen Abtastrate von $f_a$=20 kHz mußten die Signale bandbe-

grenzt und die Abtastrate reduziert werden. Dies erfolgte durch digitale Tiefpassfilterung (3 dB-Grenzfrequenz $f_g$=3,4 kHz) und anschließender Reduktion auf $f_a$=8 kHz (Eckhardt 1993).

Eine grobe Beurteilung der Signaleigenschaften kann aufgrund der Zeitverläufe sowie der Leistungsdichtespektren getroffen werden. Hierzu wurde ein Zeitintervall von 1 s aus jeder Aufnahme extrahiert, mit einem Hammingfenster multipliziert und daraus das logarithmierte Leistungsdichtespektrum berechnet.

Tabelle 4.2: Aufteilung der Geräuschdaten in Trainings- und Testpool.

| Trainingspool | Testpool |
|---|---|
| Nadeldrucker | Rechnerraum |
| Bahnhofshalle | Straßenbauarbeiten |
| Gaststätte | |
| Spülmaschine | |
| IBM-Matrixdrucker | |

Wie aus den Zeitverläufen der Signale (Bilder A.2.1a bis A.2.7a im Anhang A.2) zu ersehen ist, besitzen die Aufnahmen der Geschirrspülmaschine und der beiden Drucker überwiegend periodische Anteile. Aus den Leistungsdichtespektren (Bilder A.2.1b bis A.2.7b) wird deutlich, daß die Rechnerraumaufnahmen und die Aufnahme der Straßenbauarbeiten zumindest im betrachteten Zeitintervall breitbandig sind. Insbesondere bei instationären Geräuschsignalen sind die Eigenschaften der Signale stark von der Auswahl des betrachteten Abschnitts abhängig. Dies ist aus der spektralen Leistungsdichte der Aufnahme in der Bahnhofshalle (A.2.1b) zu entnehmen, die bei ca. 3,2 kHz ein Maximum besitzt. Durch eine Hörprobe konnte festgestellt werden, daß die Ursache hierfür die Bremsgeräusche eines einfahrenden Zuges sind.

**Aufteilung in zwei getrennte Geräuschpools.** Die insgesamt sieben Geräuschaufnahmen aus unterschiedlichen Quellen wurden für die Experimente mit *Pooltraining* (siehe 5. Kapitel) wie in Tabelle 4.2 gezeigt in zwei Untermengen mit fünf bzw. zwei Aufnahmen aufgeteilt, die im weiteren Verlauf als *Trainingspool* bzw. als *Testpool* bezeichnet werden.

**Generierung geräuschbehafteter Sprachdaten**. Für die Simulationen werden geräuschbehaftete Sprachaufnahmen mit vorgegebenem mittlerem Signal-zu-Rausch-Verhältnis $SNR_{soll}$ [dB] benötigt. Hierzu wird zunächst eine Wiederholung der 30 Wörter zusammen mit einer Geräuschaufnahme gleicher Länge betrachtet. In die Berechnung des SNR gehen nur diejenigen Abtastwerte beider Aufnahmen

ein, deren Zeitindex $i$ zwischen den detektierten Wortanfängen und Wortenden liegen, die Abtastwerte in den Sprachpausen bleiben unberücksichtigt. Die Gesamtzahl der sprachbehafteten Abtastwerte in der folgenden Betrachtung sei $N$.

Für die Generierung des $i$-ten geräuschbehafteten Abtastwertes $r(i)$ wird der Abtastwert des Sprachsignals $s(i)$ mit dem des gewichteten Geräuschsignals $n_w(i)$ gemäß

$$r(i) = s(i) + n_w(i) \qquad \text{mit} \qquad i=1,2,\ldots,N \qquad (4.1)$$

additiv überlagert, wobei $n_w(i)$ aus dem ursprünglichen Wert $n(i)$ durch Gewichtung mit einem Faktor $\alpha \in R$ gemäß

$$n_w(i) = \alpha\, n(i) \qquad (4.2)$$

berechnet wird. Die Berechnung von $\alpha$ muß für jede Aufnahme getrennt vorgenommen werden, da aus folgenden Gründen keine feste Zuordnung zwischen Werten von $\alpha$ und $SNR_{soll}$ besteht:

- Abhängigkeit von der Aussteuerung der Geräusch- und Sprachaufnahme,
- Sprecherabhängige Faktoren (z. B. Sprechlautstärke),
- Abhängigkeit von der akustischen Strecke (z. B. Mikrofonabstand).

Das vorgegebene mittlere Signal-zu-Rausch-Verhältnis $SNR_{soll}$ über alle $r(i)$ einer Aufnahme ist durch

$$SNR_{soll} = 10lg\left(\frac{E_s}{E_{n_w}}\right) \qquad (4.3)$$

gegeben, wobei $E_s$ die Energie des Sprachsignals und $E_{n_w}$ die Energie des gewichteten Geräuschsignals jeweils innerhalb der Signalabschnitte zwischen Wortanfang und Wortende mit

$$E_s = \sum_{i=1}^{N} s^2(i) \qquad \text{bzw.} \qquad (4.4)$$

$$E_{n_w} = \sum_{i=1}^{N} n_w^2(i) \qquad (4.5)$$

bedeuten. Gesucht wird der Faktor $\alpha$, mit dem die Geräuschabtastwerte gewichtet werden müssen, um eine geräuschbehaftete Sprachaufnahme mit $SNR_{soll}$ zu erreichen. Einsetzen von Gl. (4.2) in (4.5) ergibt

$$E_{n_w} = \sum_{i=1}^{N} \alpha^2 n^2(i) = \alpha^2 E_n \quad , \tag{4.6}$$

wobei die Energie des geräuschfreien Sprachsignals $E_n$ analog zu Gl. (4.4) definiert ist. Nach Einsetzen von Gl. (4.6) in (4.3) und Auflösen nach $\alpha$ erhält man

$$\alpha = \sqrt{\frac{E_s}{E_n 10^{\frac{SNR_{soll}}{10}}}} \quad . \tag{4.7}$$

Durch gewichtete Addition der Abtastwerte beider Geräuschaufnahmen gemäß

$$r(i) = s(i) + \alpha\, n(i) \tag{4.8}$$

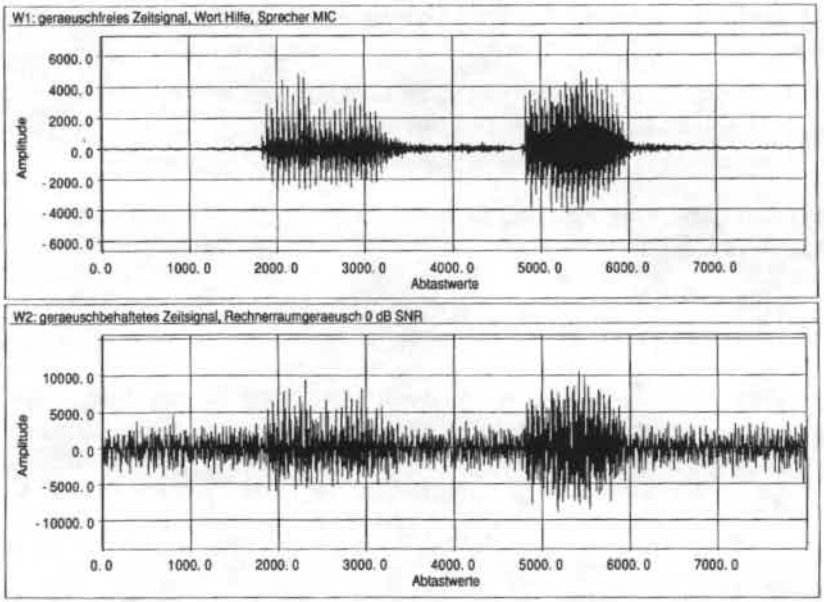



Bild 4.1a und b: Zeitsignal des Wortes *Hilfe*. Geräuschfreie (Bild 4.1a, oben) bzw. geräuschbehaftete (Bild 4.1b, unten) Aufnahme mit 0 dB additivem Rechnerraumgeräusch, Sprecher MIC.

wurde aus jeder der insgesamt 50 Sprachaufnahmen (10 Sprecher mal 5 Wiederholungen) jeweils eine geräuschbehaftete Version mit -5, 0, +5, +10, +15 und +20 dB SNR für jeden Geräuschtyp der Datenbasis generiert. Um getrennte Geräuschsignalproben für Training und Test zu verwenden, wurden die Sprachaufnahmen 1 und 2 (Training) sowie Sprachaufnahmen 3, 4 und 5 (Verifikation und Test) [10] mit unterschiedlichen Zeitabschnitten jeder Geräuschaufnahme aus der Datenbank überlagert. In der weiteren Beschreibung wird das gewichtete Geräuschsignal $n_w$ kurz mit $n$ bezeichnet.

Bild 4.1 zeigt das Zeitsignal des Wortes *Hilfe* des Sprechers MIC als geräuschfreie (Bild 4.1a) sowie mit 0 dB additivem Rechnerraumgeräusch überlagerte Aufnahme (Bild 4.1b). Der Geräuschanteil in Bild 4.1b erscheint zunächst - gemessen an der Vorgabe (0 dB SNR) - verhältnismäßig schwach; bei der Berechnung des SNR wird jedoch auch über Sprachsegmente mit kleiner Amplitude (z. B. Konsonanten) gemittelt, was sich auf die Amplitude der Geräuschkomponente insgesamt dämpfend auswirkt.

### 4.2. Merkmalsextraktion

Aufgabe der Merkmalsextraktion ist, mit Hilfe geeigneter Vorverarbeitungsmodelle eine Datenreduktion durchzuführen und dabei für die Spracherkennung relevante Information zu extrahieren. Aus der Literatur ist eine Vielzahl unterschiedlicher Verfahren bekannt. Insbesondere für gestörte Sprachsignale sind Verfahren zur Extraktion geräuschrobuster Merkmale von Hermansky (1990), Applebaum und Hanson (1990), Palival (1990) u. a. beschrieben worden.

Untersuchungen von Eckhardt et al. (1992) bestätigten die Geräuschrobustheit von *plp*-Koeffizienten (*Perceptually-Based Linear Predictive Coding*; Hermansky 1990) bei rechneraddierten Geräuschen, ergaben jedoch schlechtere Ergebnisse als *lpc-cepstrum*-Koeffizienten (*Linear Predictive Coding*, z. B. Furui 1989) in sprecherabhängigen Erkennungstests mit realen geräusch- und lombardbehafteten Sprachaufnahmen während einer Autofahrt mit mehreren Sprechern und Fahrsituationen. Daher werden in den folgenden Experimenten lpc-cepstrum-Koeffizienten verwendet; der neuronale Ansatz zur Geräuschreduktion ist jedoch konzeptionell unabhängig von der gewählten Signalrepräsentation.

---

[10] vgl. Tabelle 4.1

### 4.2.1 LPC-Cepstrum-Analyse

**LPC-Analyse** ist ein Standardverfahren zur Merkmalsextraktion und dient oft als Ausgangspunkt für weitere Verarbeitungsschritte, wie z. B. zur Berechnung der Cepstralkoeffizienten. Die Wahl des geeignetsten Verarbeitungsmodells ist problemabhängig und Gegenstand zahlreicher Arbeiten. Eine ausführliche Beschreibung des LPC-Verfahrens ist in Markel und Gray (1976) enthalten. Die folgenden Beschreibungen stammen aus Furui (1989) sowie Eppinger und Herter (1993).

Die Grundidee ist, daß sich in quasistationären Segmenten jeder Abtastwert des Sprachsignals $s(i)$ bis auf einen Restfehler durch eine Linearkombination $\tilde{s}(i)$ von $p$ vergangenen Werten approximieren läßt:

$$s(i) \approx \tilde{s}(i) = \sum_{j=1}^{p} a_j s(i-j) \qquad (4.9)$$

Die Koeffizienten $a_j$ heißen *Prädiktor-* oder *lpc*-Koeffizienten. Der Restfehler $e(i)$ bei der Prädiktion des $i$-ten Abtastwertes ist durch

$$e(i) = s(i) - \tilde{s}(i) = s(i) - \sum_{j=1}^{p} a_j s(i-j) \qquad (4.10)$$

gegeben. Die $a_j$ werden für jedes Segment so bestimmt, daß die Summe der quadrierten Prädiktionsfehler

$$\sum_{i=1}^{N} e^2(i) = \sum_{i=1}^{N} \left(s(i) - \tilde{s}(i)\right)^2 = \sum_{i=1}^{N} \left( s(i) - \sum_{j=1}^{p} a_j s(i-j) \right)^2 \qquad (4.11)$$

über alle $N$ Abtastwerte des Segments minimal wird, d. h.

$$\sum_{i=1}^{N} e^2(i) \rightarrow Min \qquad (4.12)$$

Minimumbestimmung von Gl. (4.11) führt über Nullsetzen der partiellen Ableitungen nach den $a_j$ auf das lineare Gleichungssystem

$$\sum_{j=1}^{p} a_j \sum_{i=1}^{N} \left(s(i-j)\, s(i-k)\right)^2 = \sum_{i=1}^{N} \left(s(i) s(i-k)\right) \qquad k = 1, \ldots, p \qquad (4.13)$$

mit $p$ unbekannten Koeffizienten $a_j$. Zu ihrer Bestimmung aus Gl. (4.13) existieren mehrere Lösungswege. Die in den Experimenten (Kapitel 5, 6 und 7) verwendeten Koeffizienten wurden mit der *Korrelationsmethode* berechnet, vgl. Thierer (1987). Nach Lösung des linearen Gleichungssystems in Gl. (4.13) erhält man für jedes Segment einen Vektor von $p$ Koeffizienten $a_j$.

Legt man der Spracherzeugung ein Vocoder-Modell zugrunde (z. B. Sickert 1983), so kann man $s(i)$ als Summe aus dem prädizierten Signal und der Eingangserregung $u(i)$ beschreiben:

$$\begin{aligned} s(i) &= \tilde{s}(i) + Gu(i) \\ &= \sum_{j=1}^{p} a_j s(i-j) + Gu(i) \end{aligned} \qquad (4.14)$$

wobei $G$ die Verstärkung des Systems ist. Transformation in den $z$-Bereich ergibt

$$S(z) = \frac{G}{1 - \sum_{j=1}^{p} a_j z^{-p}} U(z) \qquad (4.15)$$

wobei $U(z)$ die Anregung des Modells darstellt. Aus Gl. (4.15) läßt sich die Übertragungsfunktion $H(z)$ zwischen der Anregung $U(z)$ und dem Ausgangssignal $S(z)$ angeben:

$$H(z) = \frac{G}{1 - \sum_{j=1}^{p} a_j z^{-p}} \qquad (4.16)$$

$H(z)$ ist die Systemfunktion eines Allpolfilters mit den Koeffizienten $a_j$ und der Verstärkung $G$. Anschaulich kann man sich $H(z)$ als Übertragungsfunktion eines Röhrenmodells des menschlichen Vokaltrakts vorstellen, das durch einen Luftstrom angeregt wird. Der Frequenzgang $H(\omega)$ mit

$$H(\omega) = H(z)\big|_{z=e^{j\omega T}} \qquad (4.17a)$$

ist periodisch und kontinuierlich. Beschränkung auf eine Periode durch $0 \le \omega \le 2\pi f_a$ und Übergang von kontinuierlichen zu diskreten Frequenzwerten $l$ mit

$$\omega = \frac{2\pi f_a}{N_S} l \quad \text{bzw.} \quad l = \frac{N_S}{2\pi f_a} \omega \qquad 0 \le l \le N_S\text{-}1 \qquad (4.17b)$$

führt auf den Frequenzgang $H(l)$ bei diskreten Frequenzwerten, wobei $N_S$ die Segmentlänge bedeutet.

Da die für die Spracherkennung relevante Information in den Parametern des Modells und somit in den Koeffizienten $a_j$ der Übertragungsfunktion $H(z)$ enthalten ist, ist man an einer Trennung von Erregungs- und Modellanteilen in $S(z)$ interessiert. Hierzu erhält man den Frequenzgang von $S(l)$ durch Einsetzen von (4.17a) und (4.17b) in (4.15) aus

$$S(l) = H(l) \cdot U(l) \quad . \tag{4.18}$$

Anschließende Auswertung der Betragsinformation, Quadrieren und Logarithmierung ergibt (siehe Furui 1989)

$$log|S(l)|^2 = log|H(l)|^2 + log|U(l)|^2 \quad . \tag{4.19}$$

Durch die homomorphe Analyse in Gl. (4.19) sind nun die Anteile des Modells im ersten Term der rechten Seite mit den Anteilen der Anregung im zweiten Term additiv verknüpft. Diese Eigenschaft bleibt bei einer anschließenden inversen Fouriertransformation erhalten und wird zur Trennung beider Anteile benutzt.

**Cepstralanalyse**. Aus dem logarithmierten Spektrum $S(l)$ kann über die inverse schnelle Fouriertransformation (*Inverse East Fourier Transform*, FFT$^{-1}$) das komplexe Cepstrum $c_{FFT}(q,n)$ oder kurz $c_q$ eines Zeitsignals $s(i)$ im $n$-ten Segment berechnet werden, das durch

$$c_{FFT}(q,n) = FFT^{-1}\left(log\left(|S(l,n)|^2\right)\right) \qquad q \quad \text{Quefrency} \tag{4.20}$$

gegeben ist. Die Abszisse hat die Dimension Zeit; wegen des Übergangs auf die logarithmische Darstellung im Frequenzbereich und aufgrund der fehlenden Phaseninformation wurde für die Abszisse die Bezeichnung *Quefrency* eingeführt.

Koeffizienten mit niedrigem Index $q$ enthalten die Grobstruktur (die Modellanteile) des logarithmierten Betragsspektrums von $S(l)$ nach Gl. (4.19), während höher indizierte Koeffizienten die Feinstruktur (die Erregungsanteile) enthalten. Da man an den Parametern des Modells $H(l)$ interessiert ist, erhält man aus dem Vergleich des Cepstrums nach Gl. (4.20) mit der rechten Seite von (4.19) die Information über die Modelleigenschaften aus den ersten Koeffizienten des Cepstrums.

Der Ansatz zur Berechnung der Cepstralkoeffizienten $c_q$ aus den lpc-Koeffizienten $a_j$ basiert auf dem Vergleich der (bekannten) logarithmierten Systemfunktion $H(z)$ mit der zunächst noch unbekannten $z$-Transformierten $C(z)$ ihres Cepstrums $c(q)$ durch Gleichsetzen gemäß

$$C(z) = log(H(z)) \qquad . \qquad (4.21)$$

Ableiten nach $z^{-1}$, beidseitige Multiplikation mit $H(z)$ und anschließender Koeffizientenvergleich führen zur Lösung

$$c_1 = -a_1 \qquad \text{und}$$

$$c_q = -a_q - \sum_{j=1}^{q-1}\left(1-\frac{j}{q}\right)a_j\, c_{q-j} \quad , \quad 1<q\leq p \quad . \qquad (4.22)$$

Die Herleitung von Gl. (4.22) ist in Markel und Gray (1976) beschrieben. Die Zahl der zur Beschreibung von $H(z)$ notwendigen Koeffizienten schwankt je nach gewünschter Auflösung zwischen 8 und 15. Wegen ihrer Bestimmung aus den lpc-Koeffizienten werden die $c_q$ im Unterschied zu den *FFT-cepstrum-Koeffizienten* $c_{FFT}$ in Gl. (4.20) als *lpc-cepstrum-Koeffizienten* bezeichnet.

### 4.2.2 Zeitliche Ableitungen

Im vorigen Abschnitt wurde die Extraktion der lpc-cepstrum Koeffizienten beschrieben, mit denen sich in zahlreichen Arbeiten gute Erkennungsergebnisse erreichen ließen. Die Leistungsfähigkeit von Spracherkennungssystemen läßt sich noch steigern, wenn man die zeitliche Entwicklung der Koeffizienten im Merkmalsvektor berücksichtigt. Diese drückt sich z. B. in den Differenzen aufeinanderfolgender Werte oder in den zeitlichen Ableitungen ihres Kurvenverlaufs aus (*Temporal Derivative Features*). In Applebaum und Hanson (1990) sowie in Hanson und Applebaum (1990) sind experimentelle Ergebnisse sowie Hinweise auf den Zusammenhang zwischen Ableitungen und der darin enthaltenen Information über Wortuntereinheiten in Silbengröße beschrieben. Dort sind zwei Implementierungswege angegeben, die im folgenden skizziert werden.

1. **Zeitliche Differenzen** (*Difference Implementation*). Die rekursiven Berechnungsvorschriften für die ersten $m$ Ableitungen $c_q^{(m)}$ von $c_q$ lauten:

$$c'_q(n,J_1) = c_q(n+J_1) - c_q(n-J_1) \qquad n \text{ Segmentindex}$$

$$c_q''(n,J_2) = c_q'(n+J_2) - c_q'(n-J_2) \tag{4.23}$$

$$\vdots$$

$$c_q^{(m)}(n,J_m) = c_q^{(m-1)}(n+J_m) - c_q^{(m-1)}(n-J_m)$$

Die $J_m$ sind Vielfache des zeitlichen Abstands $T_n$ zweier aufeinanderfolgender Segmente (*Frame Rate*). Sie werden experimentell bestimmt und als *Fensterlänge* für die Berechnung der Ableitungskoeffizienten bezeichnet.

2. **Regression mit orthogonalen Polynomen** (*Regression Implementation*). Diese Implementierung beruht auf orthogonalen Polynomen, mit deren Hilfe der Wert der $m$-ten Ableitung des Koeffizienten $c_q$ wie folgt berechnet wird:

$$r_q^{(m)}(n,J_m) = \frac{\sum_{j=1}^{J_m} P_m(j,J_m)\, c_q\left(n+\left(j-\frac{J_m+1}{2}\right)T_n\right)}{\sum_{j=1}^{J_m} P_m^2(j,J_m)} \tag{4.24}$$

Die ersten drei orthogonalen Polynome lauten

$$\begin{aligned} P_1(j,J_1) &= j \\ P_2(j,J_2) &= j^2 - \frac{1}{12}\left(J_2^{\,2}-1\right) \\ P_3(j,J_3) &= j^3 - \frac{1}{20}\left(3J_3^{\,2}-7\right)j \\ &\vdots \end{aligned} \tag{4.25}$$

und sind aus Draper und Smith (1981) entnommen. $J_m$ ist die Anzahl der Vektoren, die zur Berechnung des jeweiligen Ableitungskoeffizienten berücksichtigt wurde und hängt von der Ordnung $m$ ab. Die Zeitdauer des *Regressionsfensters* $T_r$ kann daher mit

$$T_r(m) = J_m T_n \tag{4.26}$$

angegeben werden. Die Werte von $J_m$ werden experimentell bestimmt und steigen in der Regel mit zunehmender Ordnung an.

Aufgrund der besseren Ergebnisse in den zitierten Quellen wurde in der vorliegenden Arbeit mit der Regressionsimplementierung nach Gl. (4.24) experimentiert. Die Worterkennungsergebnisse mit Ableitungskoeffizienten bei geräuschbehafteten Testdaten sind in Eckhardt et al. (1992) sowie in Trompf (1993) beschrieben. Hiernach führte die erste Ableitung zu deutlichem und die zweite zu moderatem Anstieg der Erkennungsraten. Mit der dritten Ableitung konnten keine weiteren Verbesserungen mehr erreicht werden.

### 4.2.3 Hauptachsentransformation

**Transformationen zur Dimensionsreduktion der Merkmalsvektoren.** Aus Aufwandsgründen strebt man eine möglichst niedrigdimensionale, redundanzfreie Repräsentation des Sprachsignals ohne Verlust an Erkennungsleistung an. Zur Reduktion der Dimensionalität von Merkmalsvektoren sind aus der Literatur leistungsfähige Verfahren bekannt, vgl. z. B. Paliwal (1992). Insbesondere zwei Klassen von linearen Transformationen zur Orthogonalisierung der Merkmalskoeffizienten werden häufig verwendet, vgl. z. B. Ruske (1988):

1. Die *Hauptachsentransformation* (HAT). Bei dieser Abbildung wird die Varianz der Koeffizienten als ihr Informationsgehalt interpretiert. Ziel ist im ersten Schritt die Abbildung in ein neues, orthogonales Koordinatensystem gleicher Dimensionalität unter Beibehaltung der Gesamtvarianz. Es besitzt die Eigenschaft, daß seine Achsen in die Richtung mit der jeweils größten verbleibenden Restvarianz zeigen und nach absteigenden Varianzwerten sortiert sind. Die anschließende Dimensionsreduktion findet durch Abschneiden derjenigen Achsen mit niedrigen Varianzwerten statt, was lt. Annahme einer moderaten Reduktion des Informationsgehaltes entspricht. Bild 4.2 illustriert die Funktionsweise

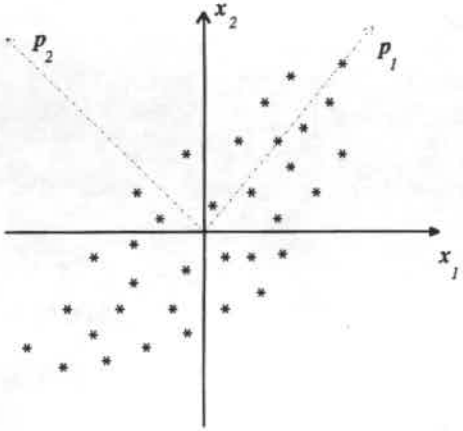



Bild 4.2: Abbildung des Merkmalsvektorraums $x$ in ein neues Koordinatensystem $p$ mit der Hauptachsentransformation.

für ein zweidimensionales Beispiel.

2. Die *lineare Diskriminanzanalyse* (*Linear Discriminant Analysis*, LDA). Bei diesem Verfahren wird ebenfalls eine lineare Abbildung in ein neues Koordinatensystem durchgeführt. Optimierungskriterium für die gesuchte Abbildung ist jedoch die Klassenunterscheidbarkeit, was beispielweise durch Maximierung des Abstandes zwischen den Zentren der zu unterscheidenden Klassen erfüllt werden kann.

In der vorliegenden Arbeit wird die gesuchte Abbildung in zweierlei Weise eingesetzt: erstens zur Dimensionsreduktion der Sprachkoeffizienten für die anschließende Klassifikation und zweitens zur Dimensionsreduktion der Geräuschkoeffizienten, um die Eigenschaften des Geräuschsignals für die Netzwerkadaption möglichst niedrigdimensional zu repräsentieren (vgl. Kapitel 7). Beides trägt zur Aufwandsbegrenzung für das Geräuschreduktionsnetz bei, da die Dimensionalität des kombinierten Vektors maßgeblich den Aufwand für die anschließende neuronale Geräuschreduktion bestimmt. Da sich die Hauptachsentransformation für beide Zwecke eignet, werden die weiteren Betrachtungen auf dieses Verfahren beschränkt.

**Berechnungsschritte für die Hauptachsentransformation.** Die zur Berechnung der Transformationsmatrix notwendigen Schritte sind in Anlehnung an Ruske (1988) beschrieben. Zunächst erfolgt die Berechnung der Kovarianzmatrix $\Lambda_x$ aus den $m$-dimensionalen Merkmalsvektoren $\boldsymbol{x}$ gemäß

$$\Lambda_x = E\left[(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^T\right] \quad . \tag{4.27}$$

$\Lambda_x$ ist eine $m \times m$-Matrix, deren Eigenvektoren $\boldsymbol{a}_i$ und Eigenwerte $\lambda_i$ aus den Gleichungen

$$\Lambda_x \boldsymbol{a}_i = \lambda_i \boldsymbol{a}_i \qquad \text{und} \tag{4.28}$$

$$\Lambda_x - \lambda_i \boldsymbol{I} = \boldsymbol{0} \tag{4.29}$$

berechnet werden. Nach ihrer Lösung werden die Paare $(\boldsymbol{a}_i, \lambda_i)$ nach absteigenden Eigenwerten

$$\lambda_1 > \lambda_2 > \ldots > \lambda_m \quad , \tag{4.30}$$

die gleichzeitig der Varianz der neuen Koeffizienten entsprechen, sortiert. Der neue Merkmalsvektor $\boldsymbol{p}$ mit den HAT-Koeffizienten wird dann aus der Abbil-

dungsmatrix $A$ berechnet, die aus den ersten $n$ Eigenvektoren $a_i$ mit den größten Eigenwerten $\lambda_i$ gebildet wird:

$$p = A^T x \quad \text{mit} \quad A = \left(a_1, \ldots, a_n\right) \quad , \quad n<m \quad . \quad (4.31)$$

Die Bestimmung von $n$ wird meist experimentell durch stufenweise Dimensionsreduktion und Evaluierung mit reduzierten Vektoren vorgenommen [11]. Für die Geräuschparameter zur Netzwerkadaption wird dies im Abschnitt 7.4.2 beschrieben.

### 4.3. Simulationssystem

In diesem Abschnitt wird ein Überblick über das Simulationssystem gegeben. Für die Worterkennung wurde ein vorhandenes Testbett eingesetzt (Eckhardt et al. 1992); die strichpunktierten Module wurden im Verlauf der vorliegenden Arbeit sowie in begleitenden Studien- und Diplomarbeiten hinzugefügt (Richter 1993, Rühle 1994, Chen 1994, Mekhaiel 1994).

**Kurzbeschreibung der einzelnen Module**. Bild 4.3 zeigt die sequentiell durchlaufenen Verarbeitungsblöcke des Gesamtsystems. Strichpunktiert gezeichnete Blöcke werden optional durchlaufen. Hierzu gehört auch die neuronale Geräuschreduktionsstufe, die grau unterlegt dargestellt wurde. Die einzelnen Module haben folgende Aufgaben:

- **SPR** und **GER**: Die Eingangsdaten werden den im vorigen Abschnitt beschriebenen **Sprach-** und **Geräuschdatenbanken** entnommen und durch entsprechende Gewichtung mit vorgegebenem Signal-zu-Rausch-Verhältnis $SNR_{soll}$ nach Gl. (4.8) additiv überlagert.
- **SS**: Dieses Modul realisiert eine Geräuschunterdrückung mit **Nichtlinearer Spektralsubtraktion** (Lockwood and Boudy 1991), wird alternativ zu neuronalen Verfahren eingesetzt und dient zu Vergleichszwecken (vgl. Kapitel 8).
- **ME**: In der **Merkmalsextraktionsstufe** wird eine für die Spracherkennung geeignete Signalrepräsentation generiert [12], vgl. Abschnitt 4.2. Außerdem werden in diesem Modul Geräuschparameter aus dem Pausensignal berechnet, die zur Netzwerkadaption benötigt werden (siehe 7. Kapitel).

---

[11] Dimensionsbestimmung der HAT-Koeffizienten für die Spracherkennung vgl. Richter (1993).

[12] Optimierungskriterium für die Merkmalsextraktionsverfahren ist die Worterkennungsrate.

- **HAT**: Die **Hauptachsentransformation** führt eine lineare Abbildung im Merkmalsraum so durch, daß die orthogonalen Achsen des neuen Koordinatensystems in die Richtungen mit den größten Varianzen der Merkmalskoeffizienten zeigen. Im Anschluß wird eine Dimensionsreduktion durchgeführt, indem nur die Koeffizienten mit größter Varianz weiterverarbeitet werden (vgl. Abschnitt 4.2.3).
- **NGR**: Die **Geräuschreduktion mit neuronalen Netzen** führt eine nichtlineare Abbildung vom Raum der geräuschbehafteten in den Raum der geräuschfreiem Merkmalskoeffizienten durch (Abschnitte 3.2 und 3.4) [13].
- **DTW**: Dieses Modul dient zum Vergleich zwischen Referenz- und Testmustern, indem mittels einer nichtlinearen Zeitanpassung (Ney 1984) Stellen größter Ähnlichkeit in beiden Mustern gesucht werden. Dieses Verfahren kann zur **Klassifikation isoliert gesprochener Wörter** verwendet werden, indem ein unbekanntes Testwort mit beispielsweise $i$ Referenzmustern im Trainingswortschatz verglichen wird. Als Ergebnis erhält man $i$ Distanzmaße, aus deren Werten durch Minimumsuche eine Hypothese für das unbekannte Wort generiert wird. Die Referenzmuster werden aus den Wiederholungen 1, 2 und 3 jedes Sprechers erzeugt (siehe Tabelle 4.1). Hierzu wird ebenfalls mit DTW eine nichtlineare Zeitanpassung zwischen den drei Trainingsmustern durchgeführt. Als Resultat erhält man für jede Wortklasse korrespondierende Vektoren der drei Repräsentanten, die dann zu einem einzigen Muster pro Klasse linear gemittelt werden. Das Trainingsverfahren ist in Krause (1991) beschrieben.
- **RES**: Dieses Modul führt eine **statistische Auswertung** der Geräuschreduktions- und Worterkennungsergebnisse durch. Aufgrund der Resultate werden Mittelwerte und Standardabweichungen der Worterkennungsraten sprecherabhängig oder gemittelt über alle Sprecher berechnet. Darüber hinaus kann mit diesem Modul eine statistische Auswertung der Ergebnisse für ausgewählte Geräuschtypen und SNR-Werte durchgeführt werden.

**Simulationsplattform**. Die Simulationen wurden mit Workstations vom Typ Sun SPARC10 sowie älteren Maschinen ähnlicher Bauart durchgeführt. Sie arbeiten unter dem Betriebssystem SunOS 4.1.3, das aus einem UNIX-Derivat als Basissystem und einer zugehörigenen Windows-Oberfläche besteht. Die Module des

---

[13] Weitere Aspekte der neuronalen Verfahren werden in den folgenden Abschnitten behandelt.

Simulationssystems wurden in der Programmiersprache C, die Ablaufsteuerung sowie verschiedene Hilfsprogramme in UNIX C-Shell implementiert.

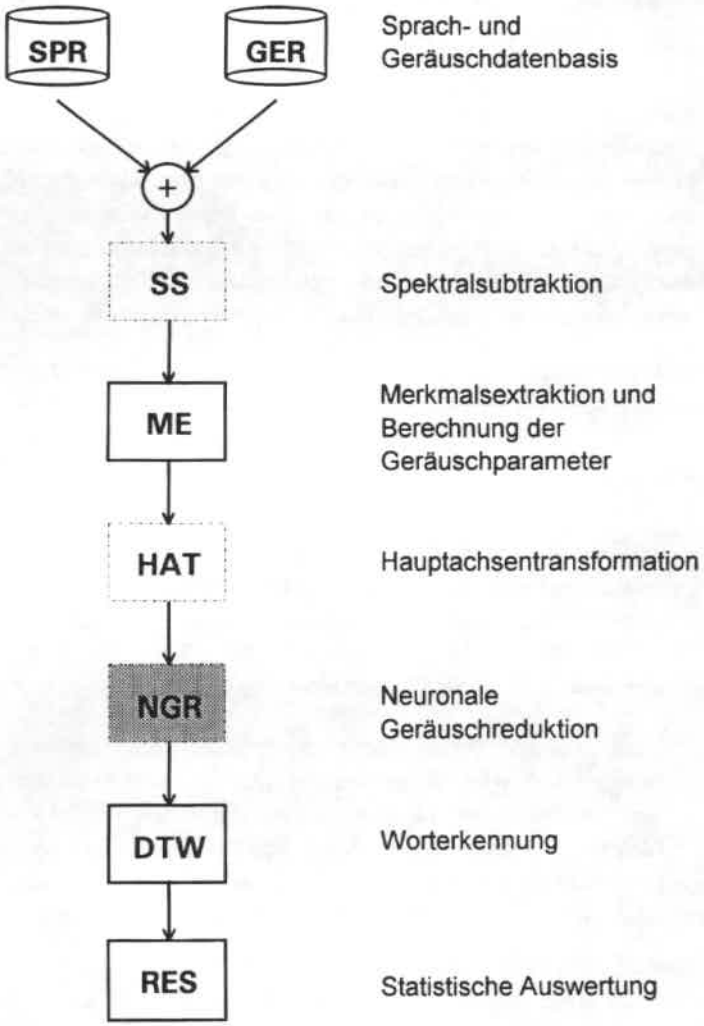



Bild 4.3: Blockbild des Simulationssystems.

# 5. GERÄUSCHREDUKTION MIT MULTILAYER PERZEPTRON-NETZWERKEN

Der merkmalsvektorbasierte Ansatz zur Reduktion additiver Hintergrundgeräusche wurde in Abschnitt 3.4 diskutiert; Bild 3.5 zeigt die Funktionsweise. Die Experimente zur Netzwerkentwicklung wurden mit dem Bürowortschatz und additivem Druckergeräusch bei unterschiedlichen Signal-zu-Rausch-Verhältnissen mit zehn Sprechern durchgeführt, vgl. Abschnitt 4.1. Die Evaluierung erfolgt mit Hilfe des MSE und der gemittelten Erkennungsraten, wobei die Angabe der Standardabweichungen aller Ergebnisse aus Platzgründen nicht immer möglich ist.

In Abschnitt 3.3 wurde als Voraussetzung für eine erfolgreiche Approximation der Regressionskurve das Erreichen eines globalen Minimums der Fehlerkurve genannt; in der Praxis läßt sich dies jedoch ohne vollständige Suche schwer nachprüfen. Dennoch sind unterschiedliche Tests zur Beurteilung des Trainingsergebnisses im Sinne einer Plausibilitätsprüfung möglich:

1. Vergleich der nichtlinearen mit linearen Netzen (Perzeptron), deren quadratische Fehlerfunktion nur ein einziges Minimum besitzt. Da der lineare als Spezialfall in nichtlinearen Lösungsansätzen enthalten ist, muß nach Gl. (3.39) im globalen Minimum für den Restfehler nichtlinearer Netze $MSE_{nl}$ im Vergleich zum linearen Fehler $MSE_{lin}$ die Beziehung $MSE_{nl} \leq MSE_{lin}$ gelten.

2. Ergebnisse von Trainingsläufen mit zufälliger Gewichtsinitialisierung unterscheiden sich in der Umgebung des (globalen) Minimums höchstens geringfügig; Abweichungen hängen von den Startwerten, der lokalen Gestalt der Fehlerkurve und der endlich kleinen Schrittweite bei der Minimumsuche ab. In Untersuchungen von Fahlman (1988) wird auf die Notwendigkeit von Versuchswiederholungen zum Erhalt signifikanter Ergebnisse hingewiesen.

Beide Punkte wurden in Voruntersuchungen überprüft, um gute Startwerte für die Experimente zu erhalten. Eine Wiederholung aller Simulationen mit unterschiedlichen Startwerten war jedoch wegen der Rechenzeiten von bis zu mehreren Tagen pro Experiment auf einer Sun SPARC10 Workstation nicht möglich. Wegen der großen Zahl von Versuchswiederholungen aufgrund der Sprecher-, Ge-

räusch- und SNR-Kombinationen (vgl. Kapitel 4) kann trotzdem davon ausgegangen werden, daß die Ergebnisse statistisch signifikant sind.

### 5.1. Experimente zur Netzwerkentwicklung

Die Netzwerkentwicklung erfolgt experimentell und gliedert sich in Simulationen zur Entwicklung der Topologie sowie des Trainingsalgorithmus. Obwohl beide nicht losgelöst voneinander bestimmt werden können, erfolgt ihre Beschreibung der besseren Übersichtlichkeit wegen dennoch getrennt. Ergänzend wurden Untersuchungen zur Auswahl und Vorverarbeitung der Trainingsdaten sowie zur Optimierung der Signalrepräsentation durchgeführt.

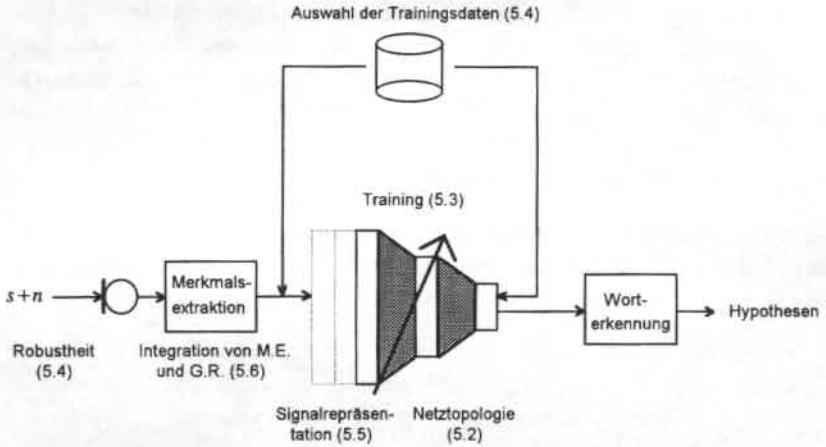



Bild 5.1: Zuordnung der Simulationsreihen zu den einzelnen Abschnitten

Bild 5.1 zeigt eine Übersicht der einzelnen Entwicklungsschritte. Zunächst wird die Netztopologie für einen Basis-Merkmalsvektor mit zehn lpc-cepstrum Koeffizienten bestimmt (Abschnitt 5.2). Zum Training (5.3) werden Varianten von EBP untersucht und die zugehörigen Parameter optimiert. Die Berücksichtigung von Parameterschwankungen in den Trainingsdaten dient zur Erhöhung der Robustheit gegenüber wechselnden Signalumgebungen (5.4). In Abschnitt 5.5 werden verbesserte Signalrepräsentationen sowie deren Einflüsse auf die Netzwerkentwicklung diskutiert, und schließlich wird die Integration von Merkmalsextraktion und Geräuschreduktion im selben Verarbeitungsschritt untersucht (5.6).

## 5.2. Topologie

**Bezeichnungsweise.** In der Versuchsbeschreibung wird die Netztopologie durch Zeichenfolgen abgekürzt, die beginnend mit der Eingangsschicht die Zahl der Neuronen in den einzelnen Schichten enthält. Sie sind jeweils getrennt durch einen Bindestrich zur Symbolisierung der Verbindungen von allen Knoten einer Schicht zur nächsten. Der Zusatz *lin* bzw. *nl* gibt an, ob es sich um ein lineares oder ein nichtlineares Netz handelt. Letzteres enthält Knoten mit sigmoider Aktivierungsfunktion nach Gl. (3.2) in der Zwischenschicht. So ist z. B. *50-20-10 nl* ein dreischichtiges Netzwerk mit 50 Eingangsknoten, 20 Knoten mit sigmoider Aktivierungsfunktion in der Zwischenschicht und 10 Ausgangsknoten.

**Testbedingungen.** Die Experimente wurden mit dem Bürodatensatz und additivem Druckergeräusch durchgeführt. Jedes Segment wird nach der Signalvorverarbeitung und Merkmalsextraktion durch zehn lpc-cepstrum-Koeffizienten repräsentiert. Zur Ermittlung der Worterkennungsraten wurde ein DTW-Erkenner verwendet (Abschnitt 4.3). Die Ergebnisse sind - wenn nicht anders angegeben - über die zehn Sprecher des Datensatzes gemittelt.

**Zahl der Schichten.** Zur Realisierung nichtlinearer Abbildungsfunktionen sind aus theoretischer Sicht dreischichtige Netzwerke notwendig und hinreichend, vgl. Abschnitt 3.2. In der Praxis werden folgende Topologieen eingesetzt:

- **zweischichtige** Perzeptrone realisieren **lineare Abbildungen.** Die Gewichte können als Koeffizienten eines (linearen) Gleichungssystems angesehen und zum Vergleich vom Trainingsalgorithmus bestimmt werden.
- **dreischichtige** MLP-Netzwerke mit nichtlinearen Aktivierungsfunktionen realisieren stetige **nichtlineare Abbildungen.** Der Aufbau eines solchen Netzwerks wurde in Bild 3.2 gezeigt.
- **vierschichtige** MLP-Netze können zur **Anpassung an die Repräsentation der Ausgangsdaten** dienen (Tamura und Waibel 1988); aufgrund der komplexeren Topologie ist dies jedoch mit erhöhtem Aufwand verbunden.

**Modifizierte Netzwerkstrukturen.** Oft werden aus der Literatur bekannte Varianten der MLP-Verbindungsstruktur verwendet. Ein Vertreter solcher Modifikationen sind z. B. *Shortcut Connections*. Sie stellen lineare Verbindungen von den Ein- zu den Ausgangsknoten dar, die parallel zur MLP-Struktur angefügt werden und den

linearen Anteil der gesamten Übertragungsfunktion realisieren (Huang et al. 1991). Sie entsprechen der Minimalstruktur, die beim *Cascade Correlation*-Netzwerk bei Trainingsstart verwendet wird (vgl. Kapitel 6).

Ziel der Experimente ist, ausgehend von einer Minimalstruktur eine problemangepaßte Topologie zu entwickeln. Zur Beurteilung der Komplexität der gesuchten Abbildung werden zunächst zwei- mit dreischichtigen Netzen verglichen. Das Trainingsverfahren, die Datenauswahl und die Bestimmung der Trainingsparameter werden in den nächsten Abschnitten gesondert beschrieben.

**Linearität.** Die Dimensionalität der Merkmalsvektoren bestimmt die Netztopologie des linearen Netzwerks mit je zehn Ein- bzw. Ausgangsknoten. Beim nichtlinearen Netzwerk wird zusätzlich eine Zwischenschicht mit 10 verdeckten Knoten und sigmoiden Aktivierungsfunktionen (siehe Gl. 3.2) eingefügt. Tabelle 5.1 zeigt den MSE sowie die Zahl der Trainingsiterationen für ein *10-10 lin* und ein *10-10-10 nl* Netzwerk. Die Lernrate und der Momentum-Term (siehe Abschnitt 5.3) wurden zu $lr$=0,01 bzw. $m_t$=0,5 gesetzt. Auffällig ist die um mehr als eine Größenordnung unterschiedliche Zahl der Iterationen. Dies hängt mit der Zahl der freien Parameter sowie der Gestalt der Fehlerfläche zusammen, die im linearen Fall eine Annäherung an das einzige Minimum in wenigen Schritten erlaubt. Der Restfehler nach Trainingsabschluß (Spalte "MSE") ist für das nichtlineare Netzwerk ca. 10 % niedriger als im linearen Fall. Die Zahl der Gewichte wird aus Gl. (6.16) berechnet.

Tabelle 5.1: Netzopologie, MSE und Zahl der Trainingsiterationen.

| Topologie | Gewichte | MSE | Iter. |
|---|---|---|---|
| *10-10 lin* | 110 | 0,434 | 6 |
| *10-10-10 nl* | 220 | 0,390 | 102 |

Ein Vergleich der Erkennungsraten in Abhängigkeit vom SNR (Bild 5.2a) zeigt, daß bei niedrigem SNR mit beiden Netzwerken Verbesserungen von mehr als 20 % erzielt werden. Hierbei weist das nichtlineare Netzwerk bei SNR-Werten von 0 und 5 dB Vorteile von einigen Prozent gegenüber dem linearen auf. Die Streuung der über zehn Sprecher gemittelten Ergebnisse ist für niedrige SNR-Werte beim nichtlinearen Netz etwas geringer als beim linearen (Bild 5.2b). Während sich hier der Aufwand für eine nichtlineare Struktur kaum zu lohnen scheint, wird im Vorgriff auf die Untersuchungen mit Signalkontext (vgl. Tabelle 5.2) auf die Notwendigkeit nichtlinearer Verarbeitungskapazität hingewiesen.

**Shortcut Connections.** Experimente mit Shortcut Connections ergaben qualitativ ähnliche Ergebnisse wie diejenigen mit der nichtlinearen *10-10-10* Basisstruktur

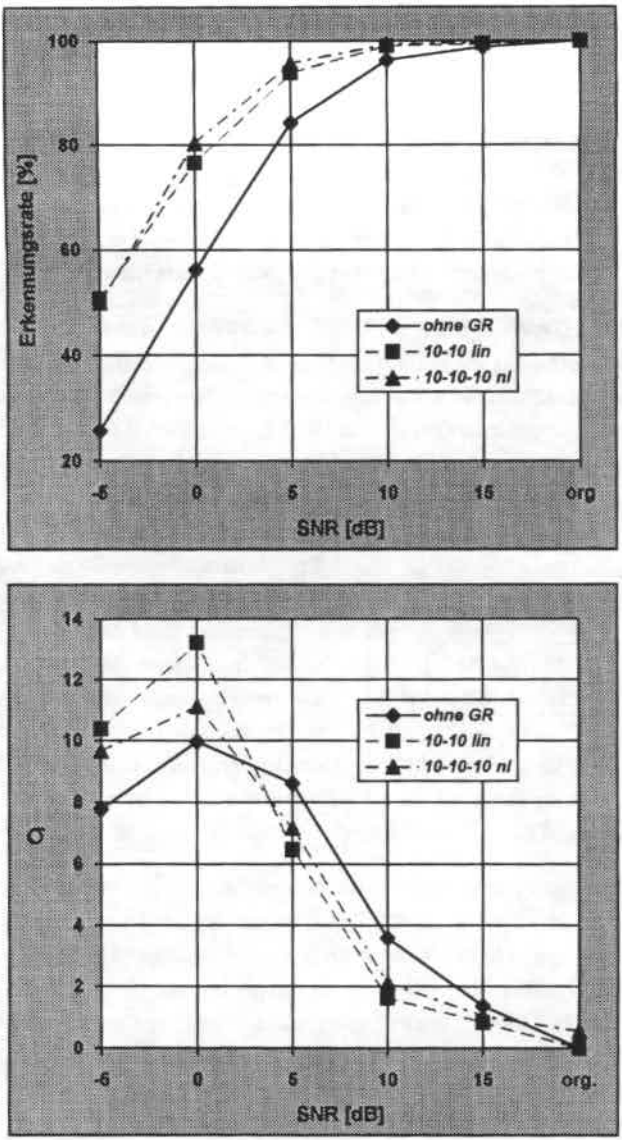



Bild 5.2a und b: Wortfehlerraten und Standardabweichung σ ohne und mit Geräuschreduktion mit linearem bzw. nichtlinearem Netzwerk.

in Tabelle 5.1, was die von Huang et al. (1991) erzielten Resultate bestätigt. Die MSE-Werte waren nahezu identisch; die Zahl der Trainingsiterationen für die nichtlineare Teilstruktur nahm dagegen nach vorherigem Training der Shortcut-Struktur zwischen 15 % und 20 % ab. Da der Aufwand für das initiale Training der Shortcut Connections zusätzlich anfällt, wurde dieser Ansatz nicht weiter verfolgt.

**Vierstufiges Netzwerk.** Das Training eines vierschichtigen *10-10-10-10* Netzwerks mit sigmoiden Aktivierungsfunktionen in beiden Zwischenschichten erwies sich bei vergleichbaren Geräuschreduktionsergebnissen als rechenaufwendiger als ein *10-10-10* Netzwerk und wurde daher ebenfalls nicht weiter untersucht.

**Zahl der verdeckten Knoten.** Bei zehn Eingangskoeffizienten ergab eine Erweiterung der Zwischenschicht um zusätzliche Knoten keine substantielle Veränderung der Ergebnisse. Im Gegensatz hierzu stehen Ergebnisse mit höherer Koeffizientenzahl, da hierbei auch in der verdeckten Schicht Kapazität zur Speicherung einer komplexeren Repräsentation des Eingangssignals geschaffen werden muß.

**Signalkontext in der Eingangsschicht.** Wegen der Korrelationen im Zeitsignal sowie der Vorverarbeitung auf Basis überlappender Segmente sind benachbarte Merkmalsvektoren in der Regel korreliert. Daher kann es vorteilhaft sein, bei der Geräuschreduktion des aktuellen Merkmalsvektors auch vergangene und zukünftige Vektoren mit zu verarbeiten, wobei ihre Zahl in den Experimenten symmetrisch zum aktuellen Vektor gewählt wurde. Dies führt zu einer erhöhten Zahl von Eingangsknoten und Gewichten. Da die Zahl der verdeckten Knoten und der Eingangsknoten nicht unabhängig voneinander optimiert werden können, wurden nach Vorversuchen 20 verdeckte Knoten im Netzwerk installiert, um zunächst die Optimierung des Eingangsfensters vornehmen zu können.

Wegen der Berücksichtigung zukünftiger Information erfordert die Schätzung des aktuellen geräuschfreien Vektors ein nichtkausales System, das sich durch Zwischenspeicherung von Eingangswerten in ein kausales überführen läßt. Die Zeitverzögerung bei seiner Berechnung hängt von der Zahl der Kontextvektoren und der Zeitdauer zwischen aufeinanderfolgenden Segmenten (*Frame Rate*) ab.

Bild 5.3a zeigt die Abhängigkeit zwischen dem MSE und der Zahl von Eingangsvektoren. Während der MSE im Trainingsdatensatz mit zunehmender Zahl von Eingangsvektoren monoton abnimmt, steigt der Verifikationsfehler nach Durchlaufen des Minimums bei neun Vektoren erneut an. Der Grund hierfür ist die erhöhte Speicherkapazität komplexer Netzwerkstrukturen, die sich mit einer größe

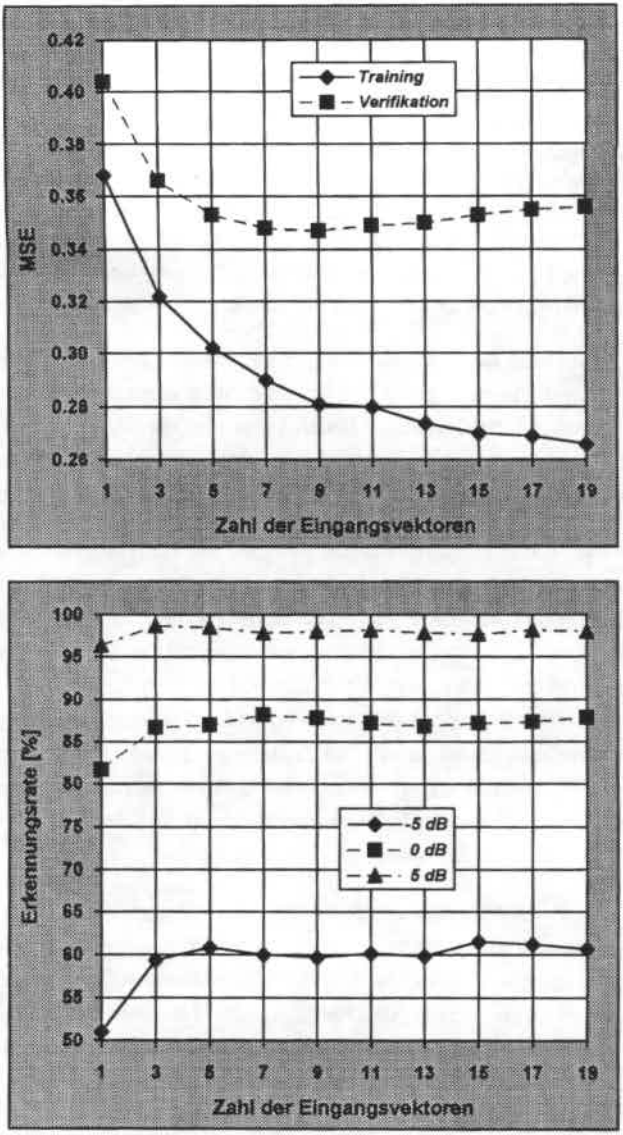



Bild 5.3 a und b: Einfluß der Zahl von Eingangsvektoren auf den MSE der Trainings- bzw. Verifikationsdaten (5.3a) und auf die Erkennungsraten bei -5, 0 und +5 dB (5.3b).

ren Zahl freier Parameter zu Lasten der Generalisierungsfähigkeit besser auf die Trainingsbeispiele spezialisieren. Aus den Erkennungsraten bei -5, 0 und 5 dB wird deutlich, daß fünf Eingangsvektoren für die weiteren Experimente ausreichen (Bild 5.3b), was zu einer Vergrößerung der Eingangsschicht auf 50 Knoten führt.

Dem Vergleich der quadratischen Restfehler des linearen und des nichtlinearen Netzwerks mit Kontextinformation (Tabelle 5.2) kann entnommen werden, daß nichtlineare Abhängigkeiten zwischen den Koeffizienten benachbarter Segmente bestehen. Bei ca. 28% höherem Restfehler für das lineare Netzwerk fällt im kontextabhängigen Fall die Ergebnisverbesserung durch das nichtlineare Netzwerk deutlich höher aus als ohne Kontext, vgl. Tabelle 5.1. Auch bei den in Tabelle 5.2 gezeigten Ergebnissen beträgt der Unterschied der Trainingsiterationen mehr als eine Größenordnung.

Tabelle 5.2: MSE und Iterationszahl bei linearer bzw. nichtlinearer Netztopologie mit jeweils fünf Eingangsvektoren.

| Topologie | Gewichte | MSE | Iter. |
|---|---|---|---|
| *50-10 lin* | 510 | 0,383 | 4 |
| *50-20-10 nl* | 1230 | 0,302 | 76 |

**Zahl der verdeckten Knoten.** Durch die Zahl der verdeckten Knoten wird die Dimensionalität der internen Signalrepräsentation bestimmt. Die tatsächlich erforderliche Zahl liegt zwischen der theoretischen Obergrenze von $2m+1$ (bei $m$ Eingangsknoten) und der Ausgangskoeffizientenzahl als Untergrenze. Eine noch kleinere Knotenzahl hätte bei unkorrelierten Ausgangskoeffizienten eine niedrigdimensionalere interne Repräsentation und somit Informationsverlust zur Folge.

Die Ober- bzw. Untergrenzen sind bei fünf Eingangsvektoren mit je zehn Koeffizienten ($m$=50) durch 101 bzw. 10 verdeckte Knoten gegeben. Da eine Annäherung an die Obergrenze somit einen starken Anstieg des Aufwands zur Folge hat, ist die Bestimmung der minimal notwendigen Knotenzahl unerläßlich.

Daher wurden Trainingsläufe mit *50-h-10* Topologieen durchgeführt. Bild 5.4 zeigt die Ergebnisse für den Restfehler nach Training mit den Daten der Sprecherin CAR. Die Verringerung der Knotenzahl unter die Dimensionalität des Ausgangsvektors hat einen drastischen Anstieg des MSE zur Folge. Höhere Knotenzahlen zwischen 10 und 20 bringen moderate Verbesserungen des MSE sowohl im Trainings- als auch im Verifikationsdatensatz. Eine weitere Vergrößerung der Zwischenschicht läßt den Trainingsfehler aufgrund einer Spezialisierung auf die Trainingsdaten weiter absinken, was aus dem flachen Verlauf des Verifikations-

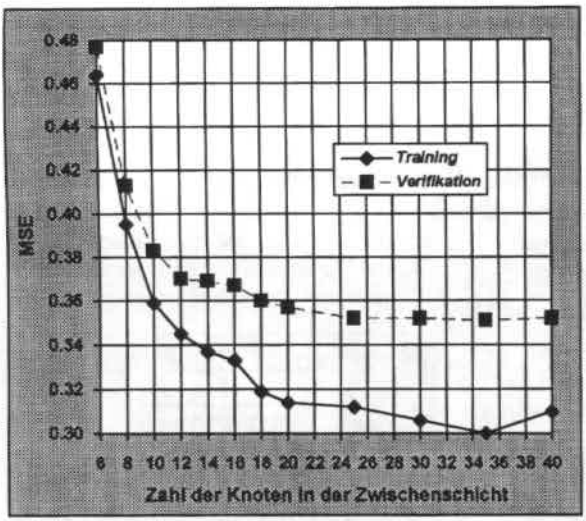



Bild 5.4: MSE versus Zahl der verdeckten Knoten (fünf Eingangsvektoren, Sprecherin CAR).

fehlers und der damit verbundenen Stagnation der Generalisierungsfähigkeit zu entnehmen ist. Eine Überprüfung mit den Daten des männlichen Sprechers MIC ergab vergleichbare Ergebnisse. Aus den bisher durchgeführten Experimenten scheint eine *50-20-10* Topologie für die vorliegende Aufgabe geeignet. Sie wird daher als Referenz für die weiteren Arbeiten verwendet.

Mit diesem Netzwerk wurde der zeitliche Verlauf aufeinanderfolgender lpc-cepstrum-Koeffizienten untersucht. Die Bilder 5.5 und 5.6 zeigen beispielhaft den Verlauf der Werte des ersten und zweiten lpc-cepstrum Koeffizienten während des Wortes *Ende*, das vom Sprecher MIC gesprochen wurde. Teilbilder 5.5a und 5.6a zeigen die geräuschfreien, 5.5b und 5.6b die geräuschbehafteten (mit 0 dB Druckergeräusch) und 5.5c und 5.6c die geräuschreduzierten Werte auf Basis des bisher entwickelten Netzwerks.

Aus den Kurvenverläufen kann zweierlei entnommen werden: erstens zeigen die geräuschreduzierten im Vergleich zu den geräuschbehafteten Werten in beiden Bildern einen glatteren Verlauf, und zweitens findet eine Angleichung der Mittelwerte an diejenigen der geräuschfreien Koeffizienten statt. Letzteres kann Bild 5.5 entnommen werden: durch den Geräuscheinfluß treten deutlich höhere Koeffizientenwerte auf, nach Geräuschreduktion (Bild 5.5c) besitzen sie jedoch wieder dieselbe Größenordnung wie die geräuschfreien Werte in Teilbild 5.5a. Insgesamt kann festgestellt werden, daß bei visueller Auswertung der Zeitverläufe neben den erwähnten Einflüssen nur wenig Ähnlichkeiten zwischen den geräuschreduzierten und den geräuschfreien Koeffizienten zu erkennen sind.

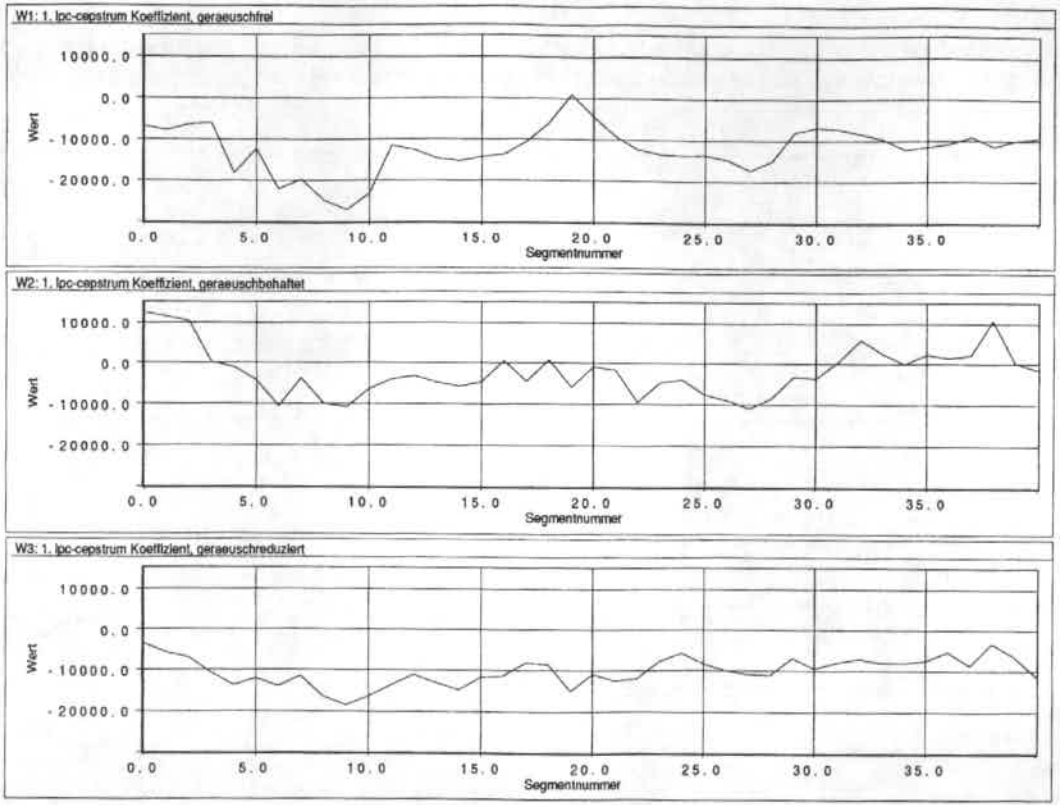



Bild 5.5 a-c: Zeitlicher Verlauf der Werte des 1. lpc-cepstrum Koeffizienten während des Wortes *Ende* (Sprecher MIC). 5.5a geräuschfreier Verlauf, 5.5b mit 0 dB additivem Druckergeräusch und 5.5c nach Geräuschreduktion.

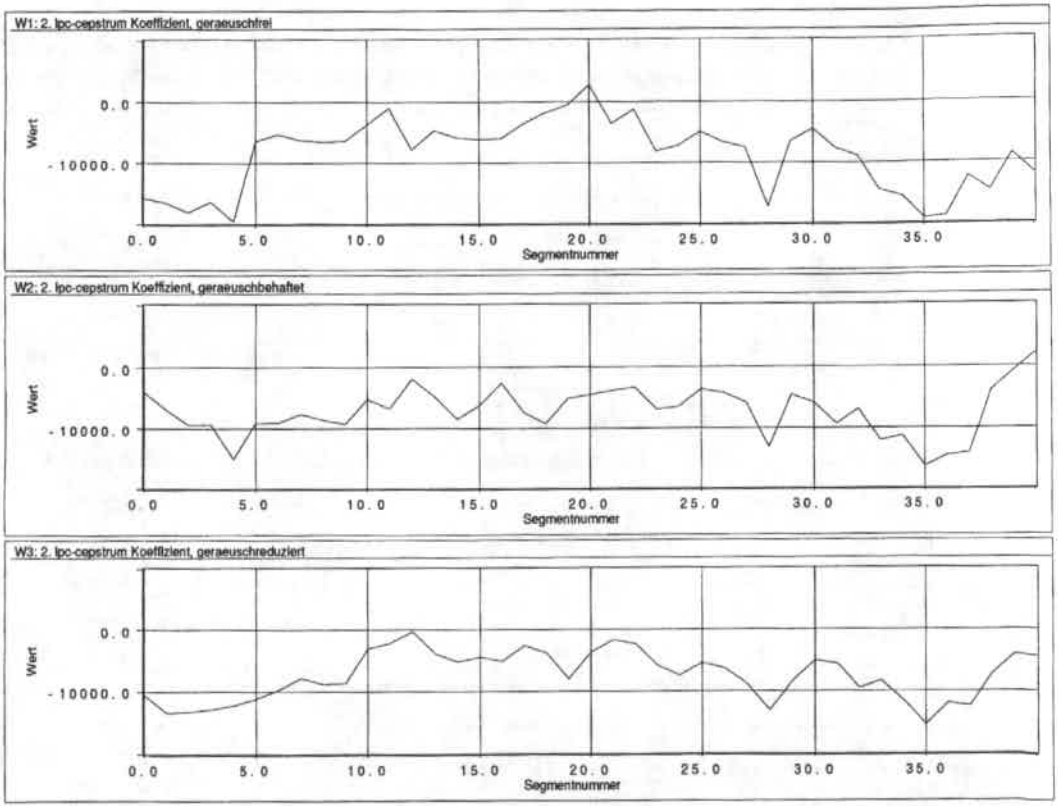



Bild 5.6a-c: Zeitlicher Verlauf der Werte des 2. lpc-cepstrum Koeffizienten während des Wortes *Ende* (Sprecher MIC). 5.6a geräuschfreier Verlauf, 5.6b mit 0 dB additivem Druckergeräusch und 5.6c nach Geräuschreduktion.

## 5.3. Training

In diesem Abschnitt werden die Parameter des Error Backpropagation-Algorithmus untersucht, Varianten des Algorithmus zur Konvergenzbeschleunigung verglichen sowie die Auswahl des Trainingsmaterials diskutiert.

**Rechenaufwand.** Ein geeignetes Maß für den Rechenaufwand während des Trainings ist die Zahl der erforderlichen Gewichtsmodifikationen. Sie errechnet sich aus dem Produkt aus der Zahl der Netzwerkgewichte, der Zahl der präsentierten Vektorpaare pro Iteration und den erforderlichen Iterationen über den gesamten Trainingsdatensatz. Modifikationen des Algorithmus zur Senkung der Iterationszahl dienen daher zur Beschleunigung der Konvergenz. Aus Rechenzeitgründen wurden die Gewichtsmodifikationen nach jedem Vektorpaar durchgeführt (stochastischer Gradientenabstieg, vgl. Abschnitt 3.1.2).

**Bestimmung der Lernparameter.** Die Lernparameter wurden für alle Gewichte gemeinsam optimiert; andere Vorgehensweisen, wie z. B. ihre schicht- oder knotenweise Festlegung, erhöhen den Aufwand beträchtlich. Die experimentelle Bestimmung geeigneter Lernparameter wurde ausführlich in Fahlman (1988) beschrieben; aus dieser Arbeit wurden auch die Startwerte übernommen. Nach Voruntersuchungen mit *10-10-10* sowie *50-20-10* Netzwerken auf Basis des Bürowortschatzes mit 10 dB additivem Druckergeräusch wurde die Lernrate auf $lr$=0,01 (fest) bzw. $lr$=0,02 (variabel) und der Momentum-Term aus Gl. (3.21) auf $m_t$=0,5 festgelegt. Die Initialisierung der Gewichte wurde mit kleinen, gleichverteilten Zufallszahlen zwischen +0,1 und -0,1 vorgenommen. Die folgenden Experimente basieren auf Merkmalsvektoren mit 10 lpc-cepstrum Koeffizienten.

**Fehlerverlauf und Generalisierungsfähigkeit.** Das Minimum des MSE im Verifikationsdatensatz wird meist vor dem Minimum im Trainingsdatensatz erreicht, da in letzterem nach dem Lernen der eigentlichen Approximationsaufgabe eine Spezialisierung auf die Trainingsbeispiele einsetzt (vgl. Abschnitt 3.1.2). Der dann beginnende erneute Anstieg des MSE führt zum Trainingsabbruch.

Bild 5.9 zeigt einen typischen Verlauf der MSE-Werte in Abhängigkeit von der Iterationszahl mit den Daten des Sprechers MIC. Die deutlich unterschiedlichen Fehlerwerte für die Trainings- bzw. Verifikationsdaten nach der ersten Iteration sind folgendermaßen zu erklären: der MSE im Trainingsdatensatz enthält die hohen Anfangsfehlerwerte nach zufälliger Initialisierung, während der MSE aus

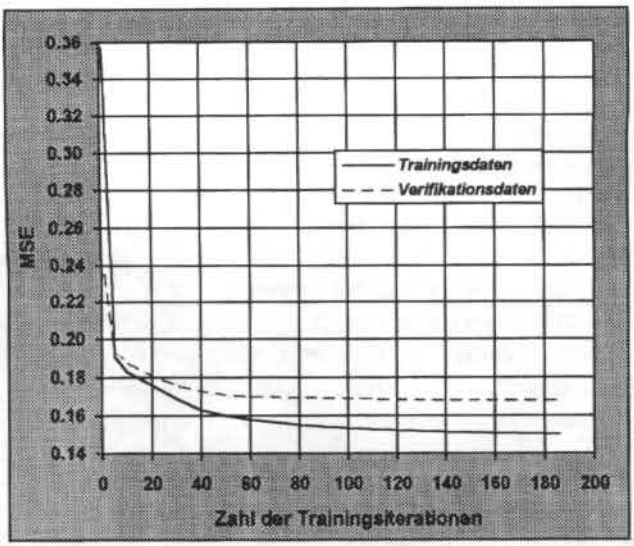



Bild 5.9: Entwicklung der MSE-Werte der Trainings- bzw. Verifikationsdaten, *50-20-10* Topologie, Daten des Sprechers MIC.

den Verifikationsdaten im Anschluß an die erste Iteration bei bereits begonnener Netzadaption gemessen wird. Nach großen anfänglichen Fehlerdifferenzen verläuft die Kurve mit zunehmender Iterationszahl flacher. Die niedrigeren absoluten Werte für den MSE im Trainingsdatensatz sind auf Spezialisierungseffekte zurückzuführen. Der erneute Anstieg im Verifikationsdatensatz nach der 186. Iteration führt zum Trainingsabbruch und ist aufgrund der Zeichenungenauigkeit im Bild nicht mehr zu erkennen.

**Kriterien zum Trainingsabbruch.** Cross Validation wird außer zur Beurteilung der Generalisierungsfähigkeit noch zur Steuerung des Trainingsverlaufs herangezogen, vgl. Abschnitt 3.1.2. Dies umfaßt die iterative Bestimmung der Lernrate sowie die Entscheidung über den Trainingsabbruch. Nach der $n$-ten Iteration gelten folgende Abbruchkriterien in der angegebenen Reihenfolge:

1. Fehlerminimum im Verifikationsdatensatz erreicht:

$$MSE_V(n) > MSE_V(n\text{-}1) \qquad (5.1a)$$

2. Fehlerminimum im Trainingsdatensatz erreicht:

$$MSE_T(n) > MSE_T(n\text{-}1) \quad . \tag{5.1b}$$

3. Flacher Verlauf der Fehlerkurve im Verifikationsdatensatz:

$$MSE_V(n\text{-}1) - MSE_V(n) < \Delta MSE_{Schwelle} \quad . \tag{5.1c}$$

4. Maximale Zahl von Trainingsiterationen überschritten:

$$n > n_{max} \quad . \tag{5.1d}$$

5. Minimale Lernrate unterschritten:

$$lr < lr_{min} \quad . \tag{5.1e}$$

Die in Gln. (5.1a-5.1e) auftretenden Parameterwerte wurden experimentell bestimmt und auf $\Delta MSE_{Schwelle} = 0{,}002$, $n_{max} = 200$ und $lr_{min} = 0{,}002$ gesetzt.

**Variable Lernrate.** In Abschnitt 3.1.2 wurde die Wirkungsweise des Gradientenabstiegs mit variabler Lernrate beschrieben, wobei $lr = 0{,}02$ als Startwert und $b_1 = 2$ sowie $b_2 = 3$ als Parameter der Reduktionsvorschrift von $lr$ nach Gln. (3.22)

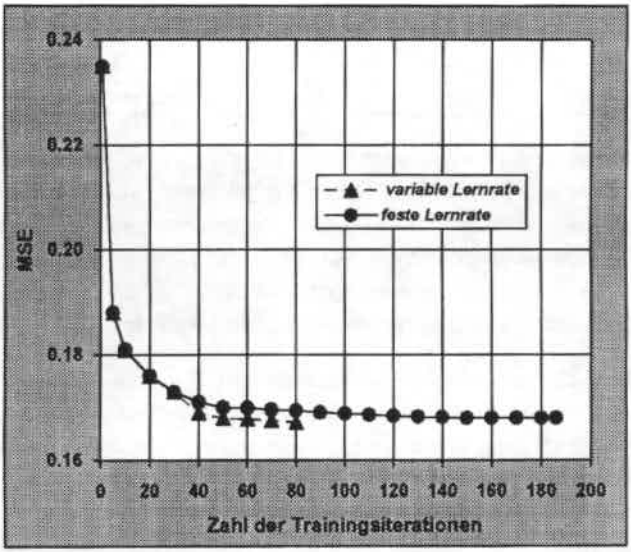



Bild 5.10: Verlauf des MSE der Verfikationsdaten bei fester bzw. variabler Lernrate, Daten des Sprechers MJC.

und (3.24) gewählt wurden. Bei den Kriterien zur Reduktion von Lernrate und Fehlerdifferenzschwelle wurden die MSE-Werte des Trainingsdatensatzes in Gln. (3.23) und (3.24) durch die entsprechenden Werte des Verifikationsdatensatzes ersetzt.

Bild 5.10 zeigt den Vergleich zweier Trainingsläufe für feste bzw. variable Lernrate mit den Daten des Sprechers MIC. Bei variabler Lernrate wird so lange mit dem Startwert für $lr$ gerechnet, bis die empirisch ermittelte Fehlerdifferenzschwelle $\Delta MSE_{Schwelle}$ unterschritten wird, was einen *flachen Verlauf* der Fehlerkurve anzeigt. Nach Verringerung von $lr$ wird $\Delta MSE_{Schwelle}$ nach Gl. (3.23) durch Division durch $b_2$ für den nächsten Schritt neu festgelegt. Dieser Vorgang wird bis zur Erfüllung eines der Abbruchkriterien fortgesetzt, bei den gewählten Startwerten erfolgte dieses nach viermaliger Reduktion der Lernrate wegen $lr < lr_{min}$.

Wie der MSE-Verlauf im Bild zeigt, bewirkt die variable Lernrate nach anfänglich identischer Fehlerentwicklung eine schnellere Konvergenz durch Reduktion der Iterationszahl von 186 auf 87. Auffällig ist, daß der Trainingsabbruch bei fester Lernrate wegen eines Minimums und bei variabler Lernrate wegen des flachen Verlaufs der Fehlerkurve im Verifikationsdatensatz erfolgte, siehe Tabelle 5.3. Dies zeigt, daß bei fester Lernrate das Minimum wegen der geringeren Auflösung überschritten und bei variabler Lernrate durch immer kleinere Schritte angenähert wurde. Die Geräuschreduktionsleistung beider Netze ist gleichwertig, was aus den MSE-Werten der Verifikationsdaten in Tabelle 5.3 zu sehen ist.

Tabelle 5.3: Training mit fester bzw. variabler Lernrate, Daten des Sprechers MIC (TD=Trainings-, VD=Verifikationsdaten).

| Lernrate | MSE im TD | MSE im VD | Iter. | Abbruchkriterium |
|---|---|---|---|---|
| fest | 0,150 | 0,168 | 186 | Minimum im VD erreicht |
| variabel | 0,154 | 0,167 | 87 | Flacher MSE-Verlauf im VD |

**Modifizierte Ableitung der Aktivierungsfunktion.** Die erste Ableitung der Aktivierungsfunktion geht multiplikativ in die Berechnung der Gewichtsmodifikation ein, vgl. Gln. (3.18) bzw. (3.19). Im flach verlaufenden Bereich der Aktivierungsfunktionen (z. B. bei sehr kleinen und großen Werten des Zellpotentials $net$) kann der bei Fahlman (1988) beschriebene, mit *Flat Spots* bezeichnete Effekt die schnelle Konvergenz des Netzes beeinträchtigen. Wenn die Aktivierungen $f'(net)$ oder $g'(net)$ einzelner Knoten sehr kleine Funktionswerte annehmen, liefern sie nur einen geringen Beitrag zur Gewichtsänderung und können damit die Konver-

genz des Trainingsverfahrens verzögern. Dies kann vermieden werden, wenn in Gl. (3.5) ein konstanter Term zum Ableitungswert hinzuaddiert wird, der unabhängig vom diesem einen zusätzlichen Beitrag zur Gewichtsmodifikation liefert (Fahlman 1988, *Modified Sigmoid Prime*). Hierzu wurden Simulationen mit unterschiedlichen additiven Konstanten zwischen 0,1 und 0,3 durchgeführt. Die Ergebnisse zeigten im Mittel eine Abnahme der Iterationszahl von ca. 5-10 % bei unveränderten Werten des MSE.

**Präsentation der Trainingsdaten.** Bei der Auswahl der Vektorpaare sollte auch bei stochastischem Gradientenabstieg jedes Vektorpaar genau einmal pro Trainingsiteration für die Fehlerberechnung und anschließende Gewichtsmodifikation berücksichtigt werden. Dies entspricht bei zufälliger Auswahl der Daten dem Verfahren *Ziehen ohne Zurücklegen*.

Wesentlich schneller und programmtechnisch weniger aufwendig ist jedoch das Verfahren *Ziehen mit Zurücklegen* zu realisieren. Dabei können innerhalb einer Iteration Vektorpaare mehrfach aus der Trainingsdatenmenge "gezogen" werden, während andere unberücksichtigt bleiben können. Geht man bei ihrer Auswahl von einer Gleichverteilung aus, so kann ihr Beitrag im Mittel über alle Iterationen als gleich groß angesehen werden. Aus Vergleichsgründen gehören wie bei sequentieller Präsentation ebenfalls $L$ Vektorpaare zu einer Iteration. An der Fehlerbestimmung im Verifikationsdatensatz nimmt jedes Vektorpaar genau einmal teil. Da die Reihenfolge der Präsentation hier keine Rolle spielt, erfolgt sie sequentiell.

Ein Vergleich der Iterationszahlen mit sequentieller und zufälliger Auswahl der Vektorpaare ergab eine um Faktor 2,3 schnellere Konvergenz bei zufälliger Präsentation (82 bzw. 192 Iterationen, Mittel über zehn Sprecher). Wie erwartet hatte dabei die Auswahl der Vektoren keinen Einfluß auf den Restfehler im Verifikationsdatensatz, der bei Erreichen des Minimums vom Suchverfahren unabhängig ist.

### 5.4. Datenauswahl und Robustheit

Gegen zweierlei Einflüsse wird die Robustheit vortrainierter Netze angestrebt:

- Abhängigkeit von Topologie- und Trainingsparametern vom jeweiligen Datensatz bei erneutem Training. Die bisherigen Ergebnisse konnten ohne Veränderung dieser Parameter mit anderen Geräuschsignalen qualitativ bestätigt werden, vgl. Abschnitte 5.4.2 und 5.5.
- Empfindlichkeit der Geräuschreduktionleistung eines trainierten Netzwerks bei instationären Signalverhältnissen.

Da der letzte Punkt mitentscheidend für eine reale Anwendung ist, wird im folgenden die Robustheit des Verfahrens gegenüber Instationaritäten des Geräuschsignals untersucht. Dabei sind zwei Fälle zu unterscheiden:

- Streuungen der Signalparameter einer einzigen Geräuschquelle.
- Einfluß unterschiedlicher Signalstatistiken bei mehreren Geräuschquellen.

Ziel der folgenden Experimente ist, die Robustheit des Geräuschreduktionsnetzwerks gegenüber beiden Arten von Fehladaption zu optimieren.

#### 5.4.1. Veränderlichkeit von Signalparametern

Zu den streuenden Parameterwerten gehören beispielsweise zeitvariante Schallpegel oder Signalspektren. Diese können durch Instationaritäten der Geräuschquelle oder veränderliche akustische Übertragungsstrecken bedingt sein und hängen von der jeweiligen Anwendungsumgebung ab.

Aufgrund der Interpolationseigenschaften neuronaler Netze wird erwartet, daß sich ihre Robustheit bei Aufnahme bekannter Parameterschwankungen in die Trainingsdaten erhöht. Dabei muß ein Kompromiß zwischen dem Wertebereich der Parameter und dem Anstieg des Restfehlers wegen ihrer Schwankungsbreite gefunden werden.

Tabelle 5.4: MSE im Testdatensatz nach Geräuschreduktion mit und ohne Multi-SNR-Training.

| | Trainings-SNR | |
|---|---|---|
| Test-SNR | +10 dB | Multi-SNR |
| Original | 0,19 | 0,20 |
| +10 dB | 0,31 | 0,28 |
| 0 dB | 0,43 | 0,36 |

**Multi-SNR-Training.** Der Einfluß streuender Parameterwerte wurde am Beispiel des SNR evaluiert. Im realen Betrieb umfaßt sein Werte-

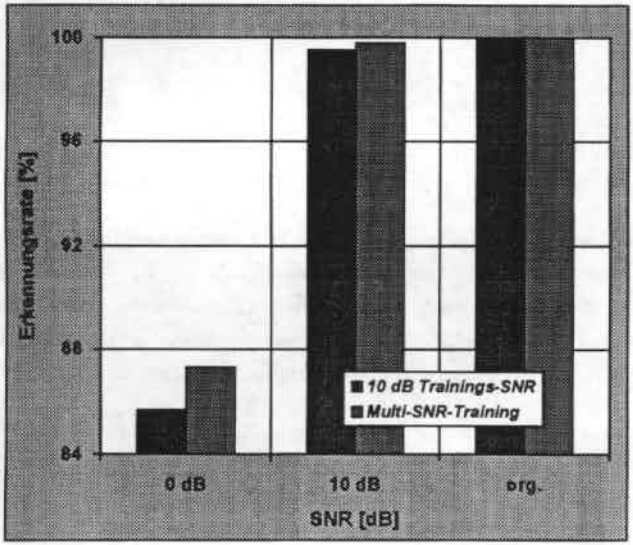



Bild 5.11: Erkennungsraten mit und ohne Multi-SNR-Training.

bereich ca. -5 bis +30 dB, was bei einer vollständigen Abdeckung im Trainingsmaterial zu einer riesigen Datenmenge und langen Trainingszeiten führt. Tatsächlich sollten wenige diskrete (mittlere) SNR-Stufen ausreichen. Dies wurde durch einen Vergleich bei Training mit 10 dB einerseits sowie bei 20, 10, 15 und 5 dB SNR (*Multi-SNR-Training*) andererseits überprüft.

Die gemittelten Ergebnisse sind in Tabelle 5.4 (MSE zwischen geräuschreduzierten und geräuschfreien Testdaten) und im Bild 5.11 (gemittelte Worterkennungsraten) gezeigt. Dabei wurden selbst bei einem Test-SNR von 10 dB (Trainings-SNR des Vergleichsnetzwerks) noch etwas bessere Ergebnisse mit Multi-SNR-Training erreicht, lediglich der MSE der geräuschreduzierten Originalaufnahmen ist geringfügig höher (erste Zeile der Tabelle). Die Worterkennungsraten für geräuschbehaftete Testdaten sind bei Multi-SNR-Training insgesamt besser (vgl. Bild 5.11). Ein Vergleich der Standardabweichung der Erkennungsraten in Bild 5.11 ergibt bei 0 dB SNR mit 8,7 gegenüber 10,9 und bei 10 dB mit 0,5 gegenüber 1,1 kleinere Werte für Multi-SNR-Training.

Bei versuchsweiser Aufnahme von Trainingsdaten mit 0 dB sowie negativen SNR-Werten verbesserten sich zwar die Erkennungsergebnisse für SNR-Werte $\leq 0$ dB weiter, doch die Fehlerrate für geräuschfreie Testdaten wurde dadurch erhöht.

### 5.4.2. Wechselnde Geräuschquellen

Experimente mit nichttrainierten Geräuschsignalen zeigten, daß fehlangepaßte Netzwerke sogar zu einer Verschlechterung der Erkennungsraten *im Vergleich zu* Systemen ohne Geräuschreduktion führen können. Ein komplettes Neutraining zur Adaption an neue Geräusche ist jedoch im realen Betrieb zu zeitaufwendig.

**Pooltraining.** Ein Ansatz zur Steigerung der Robustheit gegenüber nichttrainierten Signalen ist die Integration möglichst vieler Geräuschaufnahmen mit unterschiedlichen Eigenschaften in das Trainingsmaterial. Dies kann zudem als Ausgangspunkt für eine Adaption des so vortrainierten Netzwerks an die aktuelle Geräuschumgebung in den Sprachpausen dienen (siehe 7. Kapitel).

Für die Simulationen wurde die erweiterte Geräuschdatenbasis mit fünf Aufnahmen im Trainings- und zwei im Testpool benutzt, vgl. Abschnitt 4.1. Alternative Aufteilungen der Geräuschdatenbasis konnten aus Aufwandsgründen nicht untersucht werden. Die Worterkennungsexperimente mit *Pooltraining* wurden in zwei Simulationsreihen unterteilt:

1. die Geräuschkomponente des störbehafteten Testsignals stammt aus einer Geräuschquelle, von der eine andere Aufnahme auch im Trainingspool enthalten ist (geräusch*abhängige* Tests).
2. die Geräuschkomponente des Testsignals stammt aus einer anderen, nicht zum Training verwendeten Geräuschquelle, deren Aufnahme ausschließlich im Testpool enthalten ist (geräusch*übergreifende* Tests).

Die Begriffsdefinitionen für geräusch*abhängige* bzw. *-übergreifende* Tests im Zusammenhang mit den jeweils verwendeten Daten werden durch die in Tabelle 5.5 gezeigte Zuordnung der Trainings- bzw. Testgeräusche verdeutlicht. Zu Vergleichszwecken werden neben dem Pooltraining auch Experimente mit Einzelgeräuschen aus beiden Pools berücksichtigt, die bei gleichen Quellen in der Trainings- und Testphase ebenfalls als *geräuschabhängig* bezeichnet werden.

Tabelle 5.5: Begriffsdefinitionen für geräusch*abhängige* bzw. *-übergreifende* Simulationsreihen.

| | **Test** mit Einzelgeräuschen aus: | |
|---|---|---|
| **Training** mit: | Trainingspool | Testpool |
| Einzelgeräuschen | *abhängig* | *abhängig* |
| Trainingspool | *abhängig* | *übergreifend* |

Das Netzwerktraining erfolgte mit Error Backpropagation, variabler Lernrate, zufälliger Auswahl der Trainingsvektoren, Cross Validation und Multi-SNR-Training; die Netztopologie war *50-20-10*. Die Erkennungsergebnisse wurden jeweils aus den Mittelwerten beider Aufnahmen der 30 Wörter von zehn Sprechern gebildet, die von fünf (Trainingspool) bzw. zwei (Testpool) Geräuschaufnahmen additiv überlagert waren. Die geräuschabhängigen (geräuschübergreifenden) Ergebnisse basieren also auf 2*30*10*5=3000 (2*30*10*2=1200) Testwörtern pro mittlerem SNR-Wert (vgl. Abschnitt 4.1).

**Ergebnisse mit Geräuschpooltraining.** Die gemittelten **Worterkennungsraten** sind in den Bildern 5.12a und 5.12b (Testgeräusche aus dem Trainings- bzw. Testpool) in Abhängigkeit vom SNR der Testdaten aufgetragen. In beiden Fällen mit Pooltraining (*pool*) sind Verbesserungen für SNR-Werte ≤ 15 dB (mittlere Kurven) gegenüber der Situation ohne Geräuschreduktion (*ohne GR*) zu erkennen (untere Kurven). Erwartungsgemäß fällt der Gewinn niedriger als bei geräuschabhängigem Training und Test mit Einzelgeräuschen aus (obere Kurven). Die mit Pooltraining erzielte Verbesserung ist im geräuschabhängigen Fall (Testgeräusch im Trainingspool enthalten) größer. Aufgrund der unterschiedlichen Signaleigenschaften in beiden Pools weichen die absoluten Erkennungsraten für beide Testreihen voneinander ab. Dies ist auf die willkürliche Unterteilung der Datenbasis zurückzuführen.

Bilder 5.13a und 5.13b zeigen die **Streuung** σ der Erkennungsergebnisse in Bild 5.12a bzw. 5.12b. Beiden Diagrammen ist - mit Ausnahme des -5 dB-Wertes in Bild 5.13a sowie der geräuschfreien Daten (org) - gemeinsam, daß die Ergebnisse bei geräuschabhängiger Geräuschreduktion nach Training mit Einzelgeräuschen weniger stark streuen als diejenigen ohne Geräuschreduktion. Bei Pooltraining sind unterschiedliche Trends abzulesen: die Werte für σ im SNR-Bereich ≥ 0 dB unterscheiden sich nur unwesentlich von denen bei Tests ohne Geräuschreduktion (Bild 5.13a, *pool*, geräuschabhängiger Fall). Die geräuschübergreifenden Experimente mit Pooltraining (Bild 5.13b, *pool*) führen - gemessen an den Ergebnissen ohne Geräuschreduktion - dagegen eher zu einem Anstieg der Streuungswerte.

Für die Ergebnisse mit Pooltraining gilt, daß der durch Geräuschreduktion erzielte Gewinn hinter den mit Einzelgeräuschtraining erreichbaren Werten zurückbleibt. Ein Risiko durch Fehladaption ist im Vergleich zur Situation ohne Geräuschreduktion zumindest für die untersuchten Geräuschsignale nicht mehr gegeben.

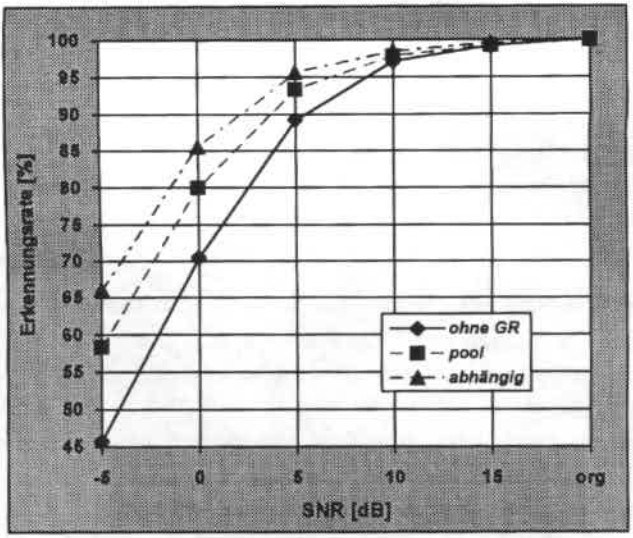



Bild 5.12a: Worterkennungsraten nach Geräuschreduktion mit Pooltraining, **geräuschabhängige** Tests.

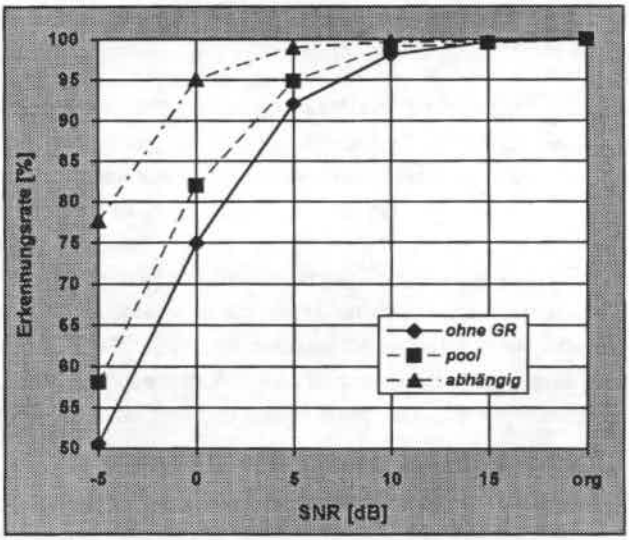



Bild 5.12b: Worterkennungsraten nach Geräuschreduktion mit Pooltraining, **geräuschübergreifende** Tests.

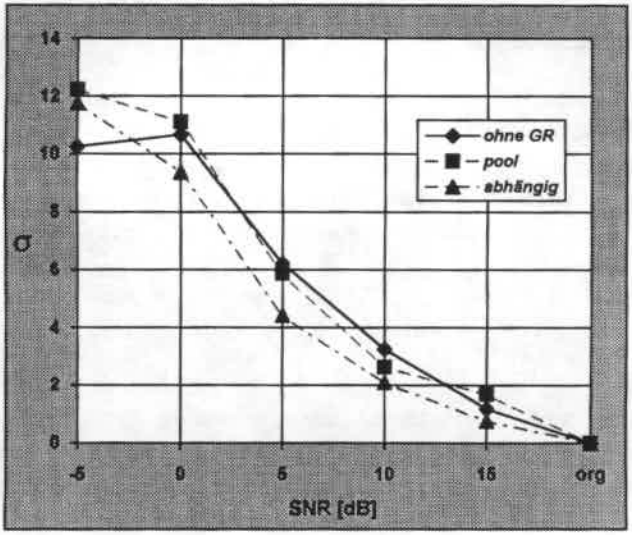



Bild 5.13a: Standardabweichung der gemittelten Worterkennungsraten nach Bild 5.12a.

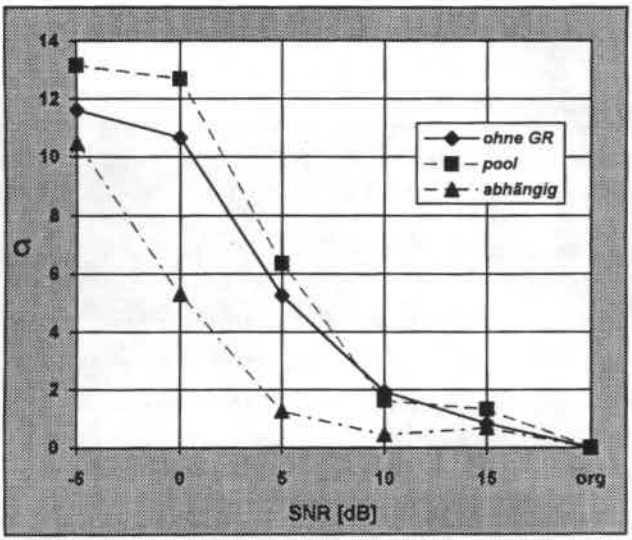



Bild 5.13b: Standardabweichung der gemittelten Worterkennungsraten nach Bild 5.12b.

## 5.5. Signalrepräsentation

Mit Hilfe der in Abschnitt 4.2.2 diskutierten zeitlichen Ableitungen der lpc-cepstrum Koeffizienten wird vor allem in geräuschbehafteter Umgebung eine Steigerung der Erkennungsrate erwartet. Dies führt zunächst zu einer Erhöhung der Koeffizientenzahl in den Merkmalsvektoren. Im nächsten Schritt kann dann eine Dimensionsreduktion mit Hilfe der Hauptachsentransformation (siehe Abschnitt 4.2.3) untersucht werden.

### 5.5.1. Erweiterte Merkmalsvektoren

Aufgrund der erhöhten Koeffizientenzahl muß die Netztopologie erweitert werden. Geht man von fünf Kontextvektoren aus, so enthält die Eingangsschicht bei $m$ Koeffizienten $5m$ Knoten. Aufgrund der Ergebnisse aus Abschnitt 5.2 wurden für die Experimente mit Ableitungskoeffizienten die in Tabelle 5.6 gezeigten Netztopologieen verwendet. Wie aus der Tabelle zu entnehmen ist, wurde wegen der höherdimensionalen internen Signalrepräsentation mit steigender Koeffizientenzahl die verdeckte Schicht um zusätzliche Knoten ergänzt. Dies führt zu starkem Anstieg der Gewichtszahl in Abhängigkeit von der Koeffizientenzahl.

Tabelle 5.6: Signalrepräsentation und Netztopologie.

| Repräsentation | Koeff. | Topologie | Gewichte |
|---|---|---|---|
| 10 lpc-cepstrum | 10 | *50-20-10* | 1230 |
| + 1. Abl. | 20 | *100-30-20* | 3650 |
| +1.+2. Abl. | 30 | *150-40-30* | 7270 |
| Hauptachsentrf. | 20 | *100-30-20* | 3650 |

Bild 5.14 zeigt den prinzipiellen Aufbau der in Tabelle 5.6 aufgeführten Netztopologieen. Wie daraus zu sehen ist, liegen in der Eingangsschicht zwei verschiedene Arten von Zusatzinformation über die Koeffizienten des aktuellen Vektors an:

- Kontextinformation durch vergangene und zukünftige Vektoren.
- Koeffizienten aus nachgeschalteten Vorverarbeitungsschritten (z. B. Berechnung der Ableitungen), die zusammen mit den lpc-cepstrum-Koeffizienten einen erweiterten Merkmalsvektor bilden.

Aus Netzwerksicht werden die beiden genannten Arten zusätzlicher Koeffizienten am Netzeingang nicht unterschieden; beide führen zu einer höheren Zahl linear

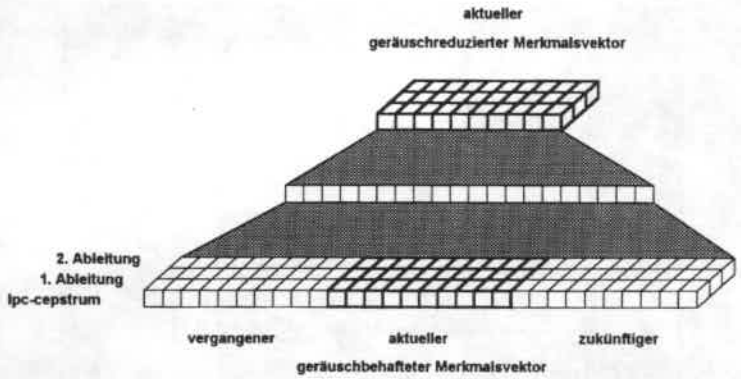



Bild 5.14: Topologie des Netzwerks mit erweiterter Eingangsinformation.

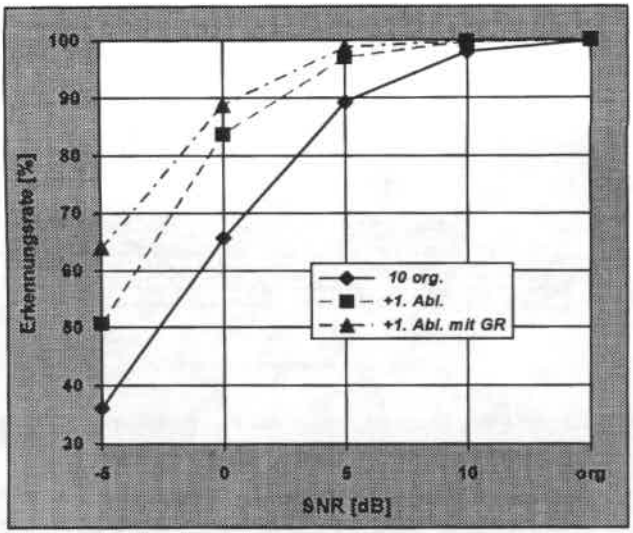
angeordneter Knoten in der Eingangsschicht. Die zusätzlichen Koeffizienten können in zweierlei Weise zur Steigerung der Erkennungsleistung beitragen:

- Durch bessere Geräuschreduktion der Basiskoeffizienten, was zu moderaten Verbesserungen sowohl des MSE nach Trainingsabschluß als auch der Erkennungsraten führt, vgl.Trompf und Hackbarth (1993).
- Durch Bildung eines erweiterten Merkmalsvektors für zuverlässigere Erkennung, d. h. die zusätzlichen Koeffizienten werden selbst geräuschreduziert und tragen anschließend zum Mustervergleich im Spracherkenner bei.

Im ersten Fall beschränkt sich die Zahl der Ausgangsknoten auf die Koeffizienten des Basisvektors. Im zweiten Fall enthält die Ausgangsschicht für jeden der $m$ Koeffizienten des erweiterten Vektors einen Knoten, was den hier durchgeführten Untersuchungen entspricht (vgl. Bild 5.14). Diese Betrachtungen gelten auch für die Koeffizienten der Hauptachsentransformation (Abschnitt 5.5.3).

### 5.5.2. Zeitliche Ableitungen

Mit den lpc-cepstrum Koeffizienten sowie ihrer ersten und zweiten Ableitung nach Gl. (4.24) wurden Erkennungstests mit additivem Rechnerraumgeräusch (vgl. Abschnitt 4.1) durchgeführt (vgl. Trompf et al. 1993). Die Ergebnisse ohne und mit Geräuschreduktion sind in den Bildern 5.15a und b zu sehen.



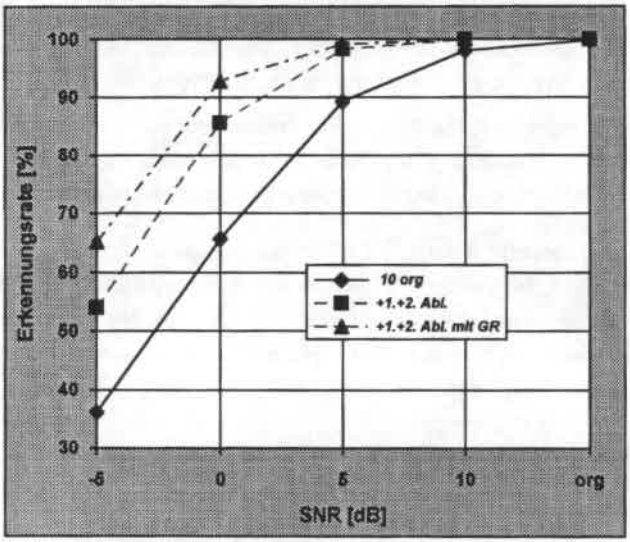




Bild 5.15a und b: Worterkennungsraten mit 20 Koeffizienten (lpc-cepstrum und 1. Abl., Teilbild (a)) bzw. 30 Koeffizienten (lpc-cepstrum, 1. und 2. Abl., Teilbild (b)), jeweils mit und ohne Geräuschreduktion.

Aus den Ergebnissen ist zu erkennen, daß durch die Ableitungskoeffizienten und die Geräuschreduktion des erweiterten Vektors insbesondere bei niedrigem SNR jeweils erhebliche Verbesserungen zu erzielen sind. Ein Vergleich beider Teilbilder zeigt, daß der Hauptanteil des Gewinns auf die erste Ableitung zurückzuführen ist. Experimente mit der dritten Ableitung führten zu keiner weiteren Verbesserung.

### 5.5.3. Hauptachsentransformation

Auf Basis der erweiterten, 30-dimensionalen Merkmalsvektoren wurde die in Abschnitt 4.2.3 beschriebene Hauptachsentransformation durchgeführt. Anschliessende Erkennungstests mit 5 bis 30 HAT-Koeffizienten ergaben, daß eine Dimensionsreduktion von 30 ursprünglichen auf 20 HAT-Koeffizienten bei gleichbleibender Erkennungsrate mit ungestörten Sprachdaten möglich ist.

Anschließend wurden die Simulationen mit geräuschbehafteten Sprachdaten wiederholt. Ein Vergleich der Ergebnisse mit 20 HAT-Koeffizienten mit denen der ursprünglichen Koeffizienten in Bild 5.16 zeigt, daß bei SNR-Werten ≥5 dB die Ergebnisse mit 30 ursprünglichen Koeffizienten bzw. 20 HAT-Koeffizienten nahezu identisch sind. Dieses Ergebnis verschiebt sich bei stärker gestörten Signalen zugunsten der HAT-Koeffizienten. Insbesondere bei SNR-Werten ≤ 0 dB erreicht

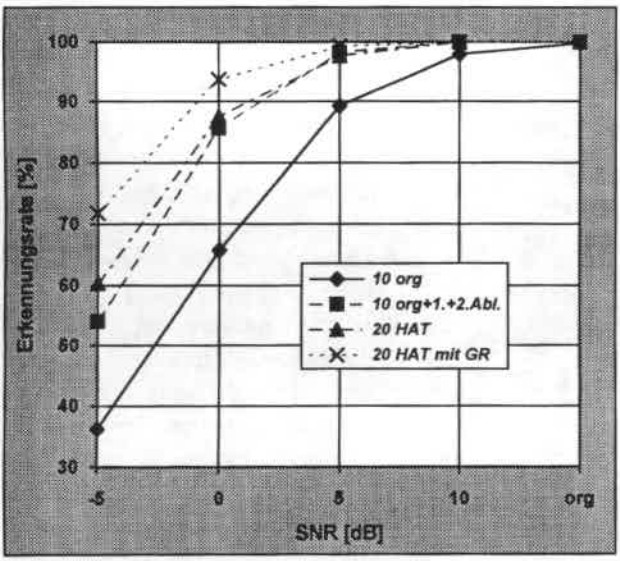



Bild 5.16: Erkennungsergebnisse mit HAT-Koeffizienten mit und ohne Geräuschreduktion bei unterschiedlichen SNR-Werten.

man mit anschließender Geräuschreduktion einen Gewinn; nach deren Anwendung auf die HAT-Koeffizienten wurden bei 0 dB SNR noch ca. 94 % Erkennungsrate erzielt.

Die aus geräuschfreier Sprache extrahierten HAT-Koeffizienten sind im Gegensatz zum ursprünglichen Merkmalsvektor unkorreliert. Eine Untersuchung ihrer Korrelationsmatrix auf Basis geräuschbehafteter Sprachdaten zeigt jedoch Korrelationen zwischen Koeffizientenpaaren, die ausschließlich auf Störeinflüsse zurückzuführen sind. Da die Zielmuster beim Training geräuschfrei sind, lernt das Netzwerk also eine Dekorrelation der HAT-Koeffizienten.

## 5.6. Vorverarbeitung und Geräuschreduktion im selben Schritt

Neben der Geräuschreduktion beinhaltet auch die Vorverarbeitung der Merkmalskoeffizienten lineare bzw. nichtlineare Zuordnungen (z. B. Berechnung der Ableitungen, Hauptachsentransformation), die prinzipiell von einem MLP-Netzwerk gelernt werden können (vgl. Abschnitt 3.2). Voraussetzung hierzu ist eine entsprechende Aufbereitung der Trainingsdaten. Dies ist in Bild 5.17 dargestellt: von den identischen Vorverarbeitungsschritten im linken (geräuschfreie Trainingsziele) und im rechten Verarbeitungszweig (geräuschbehafteter Eingang) werden die schraffiert gezeichneten Vorverarbeitungsstufen *Hauptachsentransformation* und *Ableitungsberechnung* bei schrittweisem Entfernen der schraffierten Module mittrainiert. Dann liegen am Eingang statt der 30-dimensionalen HAT-Vektoren entweder 30 lpc-cepstrum- und Ableitungskoeffizienten oder 10 lpc-cepstrum-Koeffizienten ohne Ableitungen an.

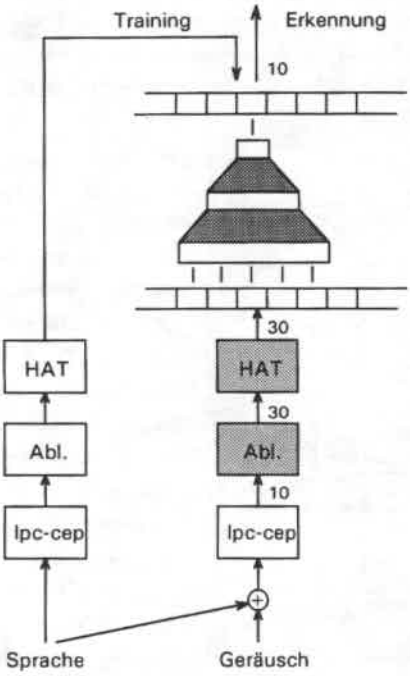



Bild 5.17: Vorverarbeitung und Geräuschreduktion in einem Schritt.

Somit ergeben sich drei Simulationsreihen mit unterschiedlichen Abbildungsaufgaben, die in Tabelle 5.7 gezeigt werden: neben der Geräuschreduktion (1.) werden vom Netzwerk zusätzlich die Hauptachsentransformation (2.) und die Berechnung der Ableitungen (3.) im gleichen Verarbeitungsschritt durchgeführt.

Tabelle 5.7: Experimente zur schrittweisen Integration von Vorverarbeitung (Abl., HAT) und Geräuschreduktion (GR).

| | | | Abbildung | | |
|---|---|---|---|---|---|
| Sim | Eingang | Ausgang | Abl. | HAT | GR |
| 1. | 30 HAT | 10 HAT | | | X |
| 2. | org+1.+2. | 10 HAT | | X | X |
| 3. | 10 org | 10 HAT | X | X | X |

Aus Rechenzeitgründen wurden alle Simulationen mit reduzierter Koeffizientenzahl und geringer Netzgröße (10 HAT-Koeffizienten als Trainingsziel) durchgeführt. Bild 5.18 zeigt die Ergebnisse mit additivem Druckergeräusch: in allen drei Experimenten mit Geräuschreduktion bzw. mit Geräuschreduktion und zusätzlich trainierten Verarbeitungsschritten wurden nahezu identische Verbesserungen gegenüber den Erkennungstests ohne Geräuschreduktion (untere Kurve im Bild) erreicht. Dies bestätigt, daß die Vorverarbeitungsschritte ohne Verlust an Erkennungsleistung zusammen mit der Geräuschreduktion durchgeführt werden können. Der Rechenaufwand wurde somit durch entsprechende Aufbereitung des Datenmaterials in die Trainingsphase vorverlagert.

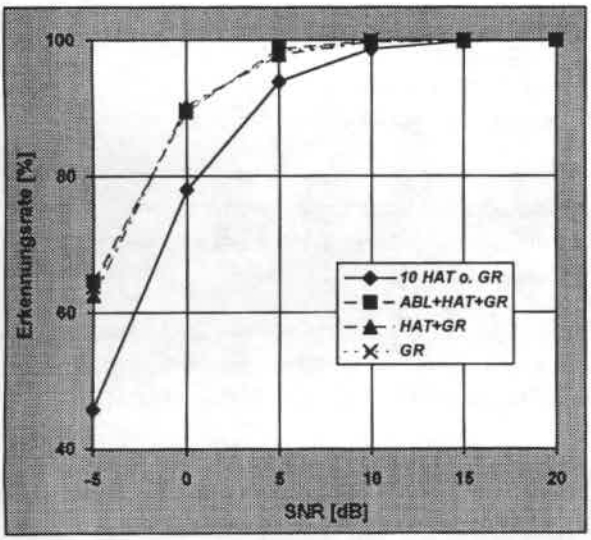



Bild 5.18: Worterkennungsraten bei Durchführung mehrerer Abbildungsaufgaben in einem Schritt.

Das 1. und 2. Experiment (vgl. Tabelle) wurde mit je fünf Kontextvektoren und 30 Eingangskoeffizienten durchgeführt. Bei impliziter

Berechnung der Ableitungskoeffizienten durch das Netzwerk (3. Experiment) wurde das Kontextfenster vergrößert, um vergleichbare Ergebnisse zu erhalten. Gute Resultate wurden mit 19 bis 23 lpc-cepstrum-Eingangsvektoren erreicht, was zunächst viel erscheint. Dies entspricht üblichen Fensterlängen zur Berechnung der 2. Ableitung (Hanson und Applebaum 1990). Mit Gl. (4.26) und $\Delta T=20$ ms beträgt das zugehörige Zeitfenster 380 bis 460 ms. Mit 190 Eingangskoeffizienten ist das Netzwerk nur wenig größer als bei 150 Knoten für fünf Kontextvektoren mit je 30 Koeffizienten.

### 5.7. Zusammenfassung der Ergebnisse mit Multilayer Perzeptron-Netzwerken

Die Entwicklung der einmaladaptiven Geräuschreduktionsnetzwerke erfolgte experimentell gesteuert und ist daher zeit- und arbeitsaufwendig. Mit dem verfolgten Ansatz lassen sich gute Geräuschreduktionsergebnisse erzielen. Das Training ist rechenaufwendig, die Vorwärtsrechnung dagegen schnell und somit echtzeitfähig.

Aufgrund ihrer Abbildungseigenschaften wurden mit Error Backpropagation trainierte Multilayer-Perzeptron-Netzwerke eingesetzt. Durch Varianten des Standardtrainings (Generalisierungstests mit Cross Valdiation, zufällige Auswahl der Trainingsmuster, Multi-SNR-Training) konnten Konvergenzgeschwindigkeit und Robustheit des Netzwerks gesteigert werden. Die Topologie hängt von der Signalrepräsentation ab und muß in der Ein- und Ausgangsschicht an die Dimensionalität der Merkmalsvektoren angepaßt werden.

Die Robustheit des Netzwerks wird durch die Auswahl der Trainingsdaten mitbestimmt und läßt sich durch Aufnahme variierender Parameterwerte in die Trainingsdaten erhöhen. Im Falle unterschiedlicher Geräuschquellen vermindert dies allerdings die Geräuschreduktionsleistung.

Abbildungsaufgaben in der Signalvorverarbeitung können durch entsprechende Aufbereitung der Trainingsdaten parallel zur Geräuschreduktion im gleichen Schritt durchgeführt werden. Hierdurch verringert sich der Rechenaufwand.

Hieraus lassen sich folgende Entwicklungsziele ableiten (vgl. Kapitel 6 und 7):

- Aufgrund der arbeitsaufwendigen, experimentell gesteuerten Entwicklung ist eine automatische Netzwerkgenerierung erforderlich.
- Lange Trainingszeiten und Leistungseinbußen bei Pooltraining machen eine Verfahren zur schnellen Adaption an neue Geräuschsignale erforderlich.

# 6. Automatische Netzwerkgenerierung

Als Alternative zur aufwendigen, experimentell gesteuerten Entwicklung von MLP-Netzwerken werden in diesem Kapitel automatische Verfahren zur Netzwerkgenerierung untersucht. Aus Aufwandsgründen und wegen der bereits guten Ergebnisse mit kleinen linearen Netzwerken (vgl. Abschnitt 5.2) werden hier konstruktive Netzgenerierungsverfahren eingesetzt, bei denen ausgehend von Minimalstrukturen nach vorgegebenen Regeln komplexere und leistungsfähigere Netze generiert werden [14]. Folgende zwei Vertreter der konstruktiven Netzgenerierungsverfahren werden in diesem Abschnitt untersucht:

- Der *Cascade Correlation*-Lernalgorithmus (CC, Fahlman and Lebière 1991) wurde ursprünglich für Aufgaben mit binären Ausgangswerten entwickelt. Sein Name steht sowohl für die Netzstruktur als auch für das Trainingsverfahren. Die Verschaltungsweise der beim Training angefügten Knoten wird als *Kaskadierung* bezeichnet, die Optimierung der zugehörigen Gewichte erfolgt über ein Kovarianzmaß [15]. Zur Minimumsuche wird der Algorithmus *Quickprop* (QP, Fahlman 1988) verwendet, der mit einer quadratischen Approximation der Fehlerkurve arbeitet und meist schnellere Konvergenz als EBP aufweist.

- Das *Resource Allocating Network* (RAN, Platt 1991; Kadirkamanathan and Niranjan 1993). Der Grundgedanke bei diesem Verfahren ist die explizite Speicherung von Trainingsbeispielen in neu angefügten verdeckten Knoten mit gaußförmigen Aktivierungsfunktionen. Anders als bei CC existieren hier keine Kurzschlußverbindungen zwischen Ein- und Ausgangsknoten.

Beiden Verfahren ist die schrittweise Approximation einer unbekannten Übertragungsfunktion durch iterative Veränderung der Netzstruktur und Modifikation der Gewichte gemeinsam. Ziel dieses Kapitels ist der Vergleich der Leistungsfähigkeit sowie des Implementierungs- und Rechenaufwands beider Netzwerkmodelle mit dem MLP für die Geräuschreduktionsaufgabe.

---

[14] siehe hierzu auch Abschnitte 3.6 und 3.7

[15] Die Autoren ersetzten das anfänglich benutzte Korrelationsmaß durch die Kovarianz, behielten jedoch den Namen des Verfahrens unverändert bei.

## 6.1. Cascade Correlation

### 6.1.1. Funktionsweise des Cascade Correlation-Lernverfahrens

Im folgenden wird das CC-Lernverfahren beschrieben, soweit es zum Verständnis der Erweiterungen für die Geräuschreduktion notwendig ist.

**Strategie.** Bei der Netzgenerierung werden ausgehend von einer Minimalstruktur solange Knoten hinzugefügt und deren Gewichte trainiert, bis keine weitere Reduktion des Ausgangsfehlers mehr eintritt. Strukturerweiterung und Gewichtstraining erfolgen schrittweise im Wechsel. Die Gewichtsmodifikationen werden mit Hilfe von Quickprop durchgeführt, das auf Basis einer quadratischen Approximation der Fehlerkurve arbeitet und im Vergleich zu anderen Gradientenabstiegsverfahren in der Regel schneller konvergiert (Fahlman 1988). Das Training läßt sich in vier Schritte aufteilen, die nachfolgend beschrieben werden.

**Schritt 1: Start mit einer linearen Minimalstruktur.** Die Anfangsstruktur besteht aus Ein- und Ausgangsschicht, deren Knotenzahl wie beim MLP (siehe Abschnitt 5.2) von der Signalrepräsentation bestimmt wird. Für die folgenden Betrachtungen sei die Zahl der Eingangsknoten $m$ und die der Ausgangsknoten $q$. Zusammen mit dem zusätzlichen Offset- oder Biaseingang $x_0$ ergeben sich $m$+1 Knoten in der Eingangsschicht.

Die Darstellung der Netztopologie (Bild 6.1a) erfolgt aus Gründen der Übersichtlichkeit in Anlehnung an die von Fahlman and Lebière (1991) verwendete. Während die Eingangsknoten nicht dargestellt sind, werden die Ausgangsknoten durch das Symbol ihrer (im Bild linearen) Aktivierungsfunktion gekennzeichnet. Ferner sind die Gewichte mit Hilfe von Quadraten dargestellt, die im Signalfluß eine Multiplikation mit den jeweiligen Gewichtswerten an den Kreuzungspunkten zweier Verbindungslinien bedeuten.

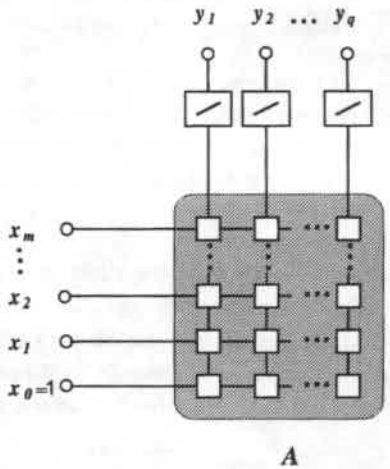



Bild 6.1a: Zweistufige Minimalstruktur aus Ein- und Ausgangsschicht.

Der Signalfluß erfolgt vom Eingang (links im Bild) zum Ausgang (oben).

Zur Berechnung der Ausgangserregungen $y$ werden die Skalarprodukte zwischen dem Eingangsvektor $x=\{x_o, x_1, \ldots, x_m\}$ und den $q$ Zeilen der $m+1 \times q$ -dimensionalen Gewichtsmatrix $A$ gebildet (zur Indizierung der Matrixelemente siehe Bild 6.1e). $x_o$ wird der Wert +1 zugewiesen und dient zur Einstellung der Biaseingänge an den Ausgangsknoten und nach Trainingsfortschritt auch der verdeckten Knoten über die zugehörigen Gewichte. Bei linearen Aktivierungsfunktionen in den Ausgangsknoten läßt sich die Übertragungsfunktion des Minimalnetzwerks zum Trainingsstart (Bild 6.1a) als Multiplikation des Eingangsvektors mit der Abbildungsmatrix $A$ angeben:

$$y^T = A\,x \qquad \text{mit} \qquad A=\{a_1^T, a_2^T, \ldots, a_q^T\}\,. \tag{6.1}$$

Die Elemente von $A$ sind durch die Gewichte des Minimalnetzwerks gegeben, die in der Gewichtsmatrix in den Zeilenvektoren $a_j$ zusammengefasst werden. Zusammen mit den Gewichten zwischen verdeckten Knoten und Ausgangsknoten werden sie als *Ausgangsgewichte* bezeichnet.

**Schritt 2: Erweiterung durch Kandidatenknoten.** Während des *Kandidatentrainings* werden versuchsweise einzelne oder mehrere konkurrierende Knoten (*Kandidatenknoten* bzw. *Kandidaten*) mit nichtlinearen Aktivierungsfunktionen zum Netzwerk hinzugefügt. Bild 6.1b zeigt $o$ parallele Knoten $K_1$ bis $K_o$, die im *Kandidatenpool* zusammengefasst werden. Derjenige Kandidat, der den größten Beitrag zur weiteren Reduktion des Restfehlers verspricht, wird fest im Netzwerk installiert (z. B. der fett gezeichnete Knoten im Bild). Die übrigen Kandidaten werden am Ende jedes *Kandidatenzyklus* (Training und Installation eines Kandidaten) wieder entfernt. Das Training innerhalb jedes Kandidatenzyklus erfolgt in zwei Durchgängen: zunächst werden die Eingangsgewichte jedes Kandidaten (Bild 6.1b), dann alle Ausgangsgewichte im Netzwerk einschließlich der des neuen Kandidaten trainiert (Bild 6.1c). Am aktuellen Trainingschritt teilnehmende Gewichte sind dunkelgrau markiert.

**Training der Eingangsgewichte**. Zunächst werden die Kandidaten eingangsseitig mit dem Netzwerk verbunden (siehe Bild 6.1b). Nach Initialisierung mit kleinen Zufallszahlen werden ihre Eingangsgewichte iterativ so trainiert, daß der Betrag der Kovarianz zwischen der Ausgangsaktivierung $V_i$ des $i$-ten Kandidaten und den Fehlerbeiträgen an jedem Ausgangsknoten maximal wird. Sie errechnet sich durch Summation über die Beiträge der $L$ Trainingsvektorpaare aus

$$\sigma_{Ve}(j) = \sum_{k=1}^{q} \left| \sum_{l=1}^{L} \left( V^l(j) - \overline{V}(j) \right) \left( e_k^{2^l} - E\left[ e_k^2 \right] \right) \right| \quad , \; j = 1, \ldots, o \; , \quad \text{mit} \quad (6.2)$$

| | |
|---|---|
| $j$ | Index des Kandidatenknotens |
| $k$ | Index des Ausgangsknotens, $1 \le k \le q$ |
| $l$ | Index des Trainingsvektorpaars, $1 \le l \le L$ |
| $o$ | Zahl der Kandidaten im Pool |
| $e_k$ | Fehler am $k$-ten Ausgang |
| $V_j$ | Aktivierung des $j$-ten Kandidatenknotens |
| $\sigma_{Ve}$ | Kovarianz zwischen $V$ und $e^2$. |

Die Maximierung der Kovarianz in Abhängigkeit von den Eingangsgewichten geschieht durch Maximumsuche mit Hilfe eines Gradientenaufstiegsverfahrens, wobei aus Rechenzeitgründen ebenfalls QP eingesetzt wird. Zu diesem Zeitpunkt besteht noch keine Verbindung zwischen Kandidaten- und Ausgangsknoten, wie aus Bild 6.1b zu sehen ist. Die optimierten Eingangsgewichte des besten Kandidaten werden als Elemente einer Zeile in die Matrix $\boldsymbol{B}$ übernommen (Bild 6.1c, Indizierung siehe Bild 6.1e). Sie enthält die Gewichte der Verbindungen zwischen dem Netzwerkeingang und den verdeckten Knoten.

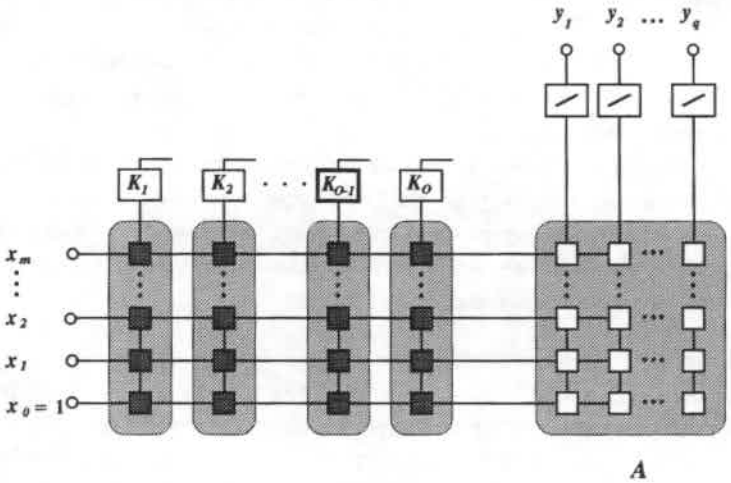



Bild 6.1b: Training der Eingangsgewichte konkurrierender Kandidaten.

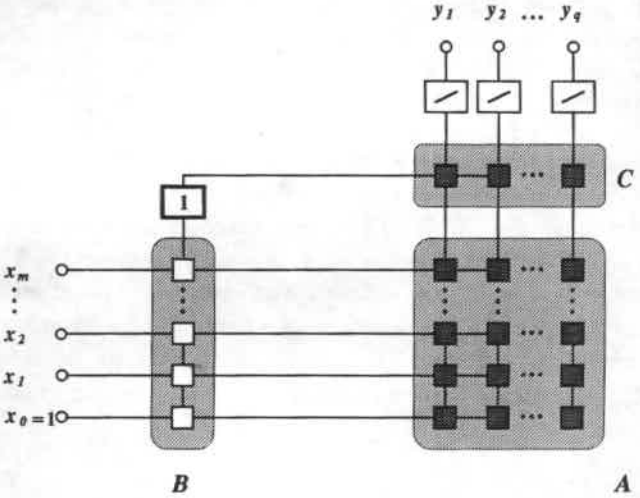



Bild 6.1c: Installation eines Kandidatenknotens und Training der zugehörigen Ausgangsgewichte.

**Training der Ausgangsgewichte.** Nach Abschluß der eingangsseitigen Gewichtsoptimierung werden die Verbindungen zwischen den Ausgängen des Kandidaten (bei mehreren Kandidaten im Pool derjenige mit der größtem Kovarianz nach Gl. (6.2)) und den Knoten der Ausgangsschicht hergestellt. Nach der Initialisierung mit $-\sigma_{V_e}(k)$ werden die ausgangsseitigen Gewichte trainiert, während die Eingangsgewichte festgehalten werden (Bild 6.1c). Nach Abschluß der Minimumsuche ist der Kandidatenzyklus beendet. Dies wird entweder durch ansteigenden oder flachen Verlauf der Fehlerkurve [16] angezeigt, der sich im Unterschreiten einer dynamisch berechneten Fehlerdifferenz in $N_i$ aufeinanderfolgenden Trainingsschritten äußert:

$$MSE_T(k+n)-MSE_T(k+n+1) < \alpha MSE_T(k+n) \quad \text{für} \quad n=0,1,\ldots,N_i . \qquad (6.3)$$

Häufig verwendete Werte für die Parameter in Gl. (6.3) sind $\alpha=0{,}01$ und $N_i=10$. Eine weitere Abbruchbedingung für einen Kandidatenzyklus ist das Erreichen ei-

---

[16] Als Zielfunktion zur Fehlerberechnung dient in der ursprünglichen Version des Verfahrens der *Error Index*, der aus der Wurzel des normierten MSE berechnet wird. Wegen der Vergleichbarkeit mit den übrigen Netzwerkmodellen wurde der *Error Index* durch den MSE nach Gl. (3.9) ersetzt.

ner vorgebbaren maximalen Zahl von Trainingsiterationen $N_{out}$. Nach Abschluß des Kandidatenzyklus wird der Kandidat fester Bestandteil des Netzwerks und als *verdeckter Knoten* bezeichnet. Die Ausgangsgewichte der verdeckten Knoten werden in der Matrix $C$ zusammengefaßt (Bilder 6.1c bzw. 6.1d).

**Schritt 3: Kaskadierung neuer Knoten.** Jeder neu generierte Knoten bildet aufgrund der in CC verwendeten Verbindungsstruktur eine eigene Zwischenschicht, wie in Bild 6.1d zu sehen ist. Seine Eingänge sind sowohl mit den Ausgängen der vorhergehenden verdeckten Knoten als auch mit den Netzwerkeingängen verbunden. Daher sind alle in den vorigen $r$ Kandidatenzyklen gelernten Signalrepräsentationen $h_r$ nach Multiplikation mit den Elementen der Matrix $D$ als Eingangssignale für die nachfolgenden Knoten verfügbar. Durch die hierarchische Anordnung neuer Knoten wird mit zunehmendem Trainingsfortschritt das Lernen

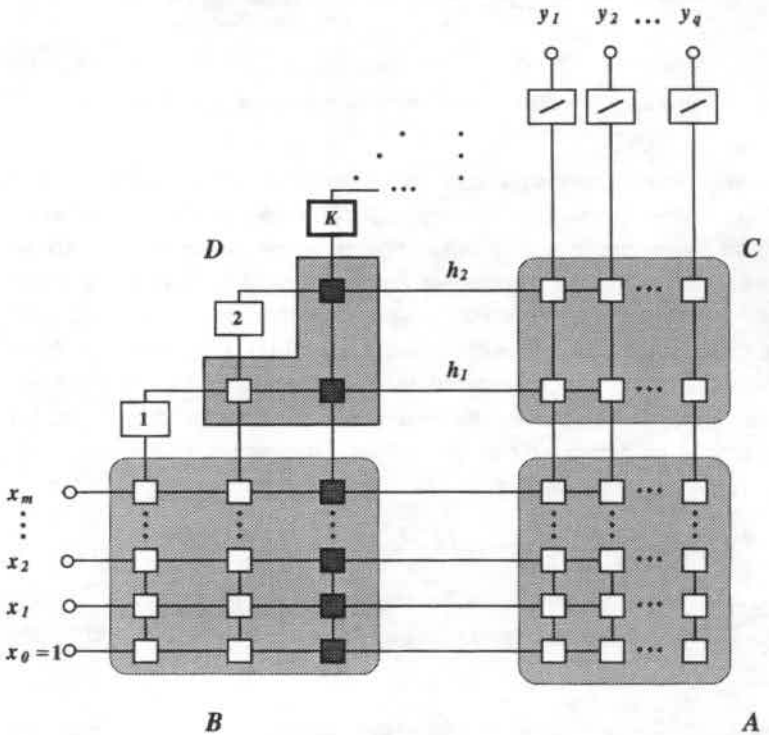



Bild 6.1d: Kaskadierung neuer Knoten und Training eines Kandidaten $K$ nach zwei erfolgreich verlaufenen Kandidatenzyklen.

komplexerer Repräsentationen des Eingangssignals an den Ausgängen der verdeckten Knoten (*Feature Detectors*, Fahlman and Lebière 1991) möglich. Daneben bleiben die Kurzschlußverbindungen zwischen Eingangs- und Ausgangsknoten der anfänglichen Minimalstruktur erhalten.

Die Anordnung der verdeckten Knoten wird als *Kaskadierung* bezeichnet. In einer Momentaufnahme ist zu sehen (Bild 6.1d), wie die Eingangsgewichte eines Kandidatenknotens nach zwei erfolgreich verlaufenen Kandidatenzyklen (installierte Knoten 1 und 2) trainiert werden. Das Anfügen von Kandidaten und deren Installation im Netzwerk wird fortgesetzt, bis eines der Abbruchkriterien erfüllt ist.

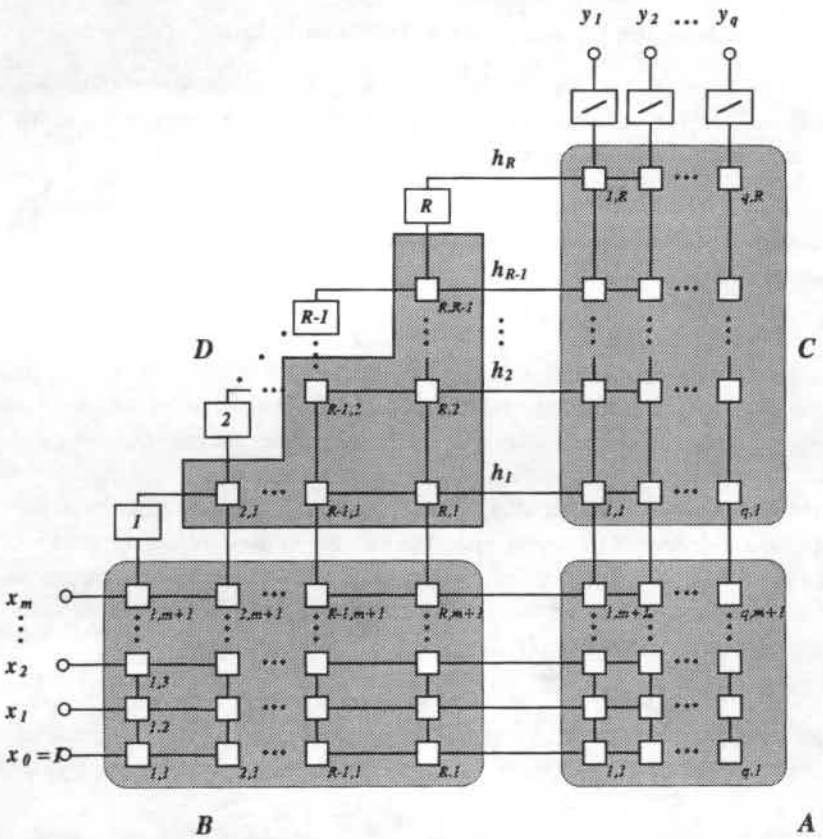



Bild 6.1e: Netztopologie mit $R$ verdeckten Knoten nach Trainingsabschluß.

**Schritt 4: Trainingsabbruch.** Jedes der beiden folgenden Kriterien ist eine hinreichende Bedingung für den Trainingsabbruch. Sie sind auf Basis des MSE [17] bzw. der maximalen Zahl verdeckter Knoten formuliert:

1. Unterschreitung einer vorgegebenen unteren Schwelle $MSE_{min}$ für den Trainingsfehler, d.h. $MSE_T < MSE_{min}$.
2. Erreichen einer maximalen Zahl von Kandidatenzyklen, d.h. $r = R_{max}$.

Die Schwellwerte $MSE_{min}$ und $R_{max}$ müssen abhängig vom aktuellen Datensatz bestimmt werden. Bild 6.1e zeigt die vollständige Topologie mit $R$ verdeckten Knoten nach Trainingsabschluß.

### 6.1.2. Vergleich der Topologie mit dem Multilayer Perzeptron-Netzwerk

Sorensen und Hartmann (1992) haben gezeigt, daß die CC-Netzstruktur dreischichtige MLP-Netzwerke als Untermenge enthält. Dies wird anhand der in Bild 6.1e abgebildeten CC-Netzstruktur erläutert.

**Ausgangsaktivierung des CC-Netzwerks**. Für die Ausgangsaktivierung des CC Netzwerks in Bild 6.1e mit $m$+1 Netzwerkeingängen, $R$ verdeckten Knoten und $q$ Ausgängen gilt

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{C}\boldsymbol{h} \quad , \qquad (6.4)$$

wobei $\boldsymbol{A}$ und $\boldsymbol{C}$ die Dimensionalitäten $q \times m$+1 bzw. $q \times R$ besitzen. Dabei wird der lineare Anteil von $\boldsymbol{y}$ durch den Term $\boldsymbol{A}\boldsymbol{x}$ beschrieben und ist durch die gewichteten *Kurzschlußverbindungen* von den Eingangs- zu den Ausgangsknoten gegeben. Die in $\boldsymbol{h}=\{h_1,\ldots,h_R\}$ enthaltene interne Signalrepräsentation setzt sich aus den Aktivierungen $h_r$ an den Ausgängen der verdeckten Knoten zusammen. Wegen der nichtlinearen Aktivierungsfunktionen der verdeckten Knoten stellt der zweite Term in Gl. (6.4) den nichtlinearen Anteil an der Übertragungsfunktion des Netzwerks dar. Die Berechnung von $\boldsymbol{h}$ aus dem Eingangssignal geschieht mit Hilfe der Gewichtsmatrizen $\boldsymbol{B}$ und $\boldsymbol{D}$.

$\boldsymbol{B}$ enthält die Gewichte der Verbindungen von den $m$+1 Eingangsknoten zu den $R$ verdeckten Knoten und hat daher die Dimensionalität $R \times m+1$. Die Elemente von $\boldsymbol{D}$ sind durch die Verbindungen der verdeckten Knoten untereinander gegeben.

---

[17] Der ursprünglich verwendete Error Index wurde durch den MSE nach Gl. (3.9) ersetzt.

Eine Besonderheit der CC-Struktur ist, daß die verdeckten Knoten mit höherer Ordnungszahl eingangsseitig mit den Ausgängen der davorliegenden Knoten mit niedrigerer Ordnungszahl verbunden sind. Die Reihenfolge der Numerierung wird durch den Kandidatenzyklus, in dem der jeweilige Knoten installiert wurde, bestimmt. Die Ausgangsaktivierungen der ersten beiden Knoten mit der Aktivierungsfunktion $g()$ berechnen sich aus

$$h_1 = g\left(\sum_{j=0}^{m} b_{1,j+1} x_j\right) \qquad \text{und}$$

$$h_2 = g\left(\sum_{j=0}^{m} b_{2,j+1} x_j + d_{2,1} h_1\right) \qquad (6.5)$$

Für die Aktivierung des $r$-ten verdeckten Knotens gilt dann

$$h_r = g\left(\sum_{j=0}^{m} b_{r,j+1} x_j + d_{r,1} h_1 + d_{r,2} h_2 + \ldots + d_{r,r-1} h_{r-1}\right) \qquad (6.6)$$

Fasst man in Gl. (6.6) die von $\boldsymbol{D}$ abhängigen Glieder zusammen, so errechnet sich $h_r$ aus den Eingangswerten und den Aktivierungen der $r$-1 eingangsseitig vorgelagerten verdeckten Knoten aus

$$h_r = g\left(\sum_{j=0}^{m} b_{r,j+1} x_j + \sum_{i=1}^{r-1} d_{r,i} h_i\right) \qquad (6.7)$$

Dabei repräsentiert der erste Term im Argument von $g()$ wie $\boldsymbol{W}$ in der Eingangsmatrix des MLP (vgl. Abschnitt 3.1) die Abbildung von den Eingangs- zu den verdeckten Knoten. Der zweite Term enthält die Gewichte der Verbindungen der verdeckten Knoten untereinander. Voraussetzung zur Berechnung der Aktivierung des $r$-ten verdeckten Knotens ist, daß alle vorhergehenden $r$-1 Aktivierungen bekannt sind. Wegen der Nichtlinearität von $g()$ enthält dieser Teil der Übertragungsfunktion die nichtlinearen Eigenschaften des gesamten Netzwerks.

**Verbindungsstruktur von MLP- und CC-Topologie**. Vergleicht man die in Bild 6.1e gezeigte CC- mit einer MLP-Struktur, so stellt man folgende Unterschiede fest:

1. Das MLP enthält keine Kurzschlußverbindungen von der Ein- zur Ausgangsschicht, d.h. der erste Summand $\boldsymbol{Ax}$ in Gl. (6.4) verschwindet.

2. In der MLP-Struktur sind keine Verbindungen der verdeckten Knoten untereinander enthalten, d.h. der zweite Term im Argument der Funktion $g()$ in Gl. (6.7) verschwindet ebenfalls. Gl. (6.7) lautet dann in Matrixschreibweise

$$\boldsymbol{h} = \boldsymbol{g}(\boldsymbol{Bx}) \qquad (6.8a)$$

Setzt man Gl. (6.4) in (6.8a) ein, vereinfacht sich die Übertragungsfunktion zu

$$\boldsymbol{y} = \boldsymbol{C}\,\boldsymbol{g}(\boldsymbol{Bx}) \qquad (6.8b)$$

Gl. (6.8b) zeigt, daß die MLP- als Sonderfall in der CC-Struktur enthalten ist, wenn nach der Bezeichnungsweise in Bild 6.1e die Bedingungen

$$\underbrace{\boldsymbol{A} \equiv \boldsymbol{0}}_{\text{Ein-/Ausgangsverbindungen}} \quad \text{und} \quad \underbrace{\boldsymbol{D} \equiv \boldsymbol{0}}_{\text{Verbindungen in verdeckter Schicht}} \qquad (6.9)$$

erfüllt sind. Beim MLP sind lediglich die Elemente von $\boldsymbol{B}$ und $\boldsymbol{C}$ von Null verschieden. Da die MLP-Struktur eine Teilmenge der CC-Struktur darstellt, besitzen CC-Netzwerke ebenfalls die Eigenschaft, nichtlineare und stetige Funktionen aus repräsentativen Trainingsbeispielen beliebig genau approximieren zu können (vgl. Abschnitt 3.2).

### 6.1.3. Erweiterungen für die Geräuschreduktion

Folgende **Unterschiede** zu MLP-Netzwerken sind für die experimentellen Untersuchungen mit Cascade Correlation von Bedeutung:

- **Fehlerkurve und Gradientenabstieg**. Während bei EBP die tatsächliche Fehlerkurve zur Berechnung des Gradienten benutzt wird, wird bei QP mit einer quadratischen Approximation gerechnet (vgl. Fahlman 1988). Statt des stochastischen Gradientenabstiegs bei EBP wird bei QP der Fehlergradient nicht aus einzelnen Trainingsvektorpaaren (stochastischer Gradientenabstieg), sondern aus dem Mittelwert über das gesamte Trainingsmaterial berechnet (*Batch Learning*, vgl. Gl. (3.20)). Daher erfolgt die Gewichtsmodifikation in Richtung des Gradienten des Gesamtfehlers und das Training konvergiert in weniger Schritten; andererseits sind zur Gradientenberechung erheblich mehr Fehlerbeiträge zu berücksichtigen.

- **Vernetzungsstruktur**. Die Minimalstruktur zum Trainingsstart realisiert bereits eine vollständige lineare Abbildung zwischen den Ein- und Ausgangsdaten, was aufgrund der Ergebnisse aus Abschnitt 5.2 eine suboptimale, aber im-

merhin bereits funktionsfähige Lösung der Geräuschreduktionsaufgabe bedeutet. Daher liegt die Vermutung nahe, daß die CC-Struktur zur Realisierung der nichtlinearen Abbildungsfunktion wenige zusätzliche Knoten benötigt. Außerdem führen die Unterschiede in der Vernetzungsstruktur bei gleicher Knotenzahl zu erheblich mehr Gewichten als bei der MLP-Struktur [18].

Aufgrund der Anforderungen für die Geräuschreduktionsaufgabe wurden gegenüber der ursprünglichen Version des CC-Algorithmus [19] folgende **Erweiterungen** vorgenommen:

- **Modifikation der Abbruchkriterien** nach Abschnitt 6.1.1, Schritt 4. Die absolute Fehlerschwelle als Abbruchkriterium ist vom Datensatz sowie von der Signalvorverarbeitung abhängig und kann daher für die vorliegende Problemstellung nicht verwendet werden. Das Kriterium $MSE_T < MSE_{min}$ wurde in den Geräuschreduktionsexperimenten nie erreicht, da $MSE_{min} = 0$ gesetzt wurde.

  Als zusätzliches Abbruchkriterium wurde wie beim MLP-Netzwerk ein Generalisierungstest mit Cross Validation nach Gl. (3.25) zwischen den Kandidatenzyklen eingeführt. Dadurch läßt sich die unerwünschte Spezialisierung des Netzwerks auf die Trainingsbeispiele verhindern. Die neuen Abbruchkriterien lassen sich damit wie folgt formulieren (vgl. MLP-Training, Gln. (5.1a-e)):

  1. Fehlerminimum im Verifikationsdatensatz erreicht:

  $$MSE_V(r) > MSE_V(r\text{-}1) \,. \tag{6.10a}$$

  2. Fehlerminimum im Trainingsdatensatz erreicht:

  $$MSE_T(r) > MSE_T(r\text{-}1) \,. \tag{6.10b}$$

  3. Maximale Zahl von Kandidatenzyklen überschritten:

  $$r > R_{max} \,. \tag{6.10c}$$

  Im Unterschied zum MLP-Training wurde der Iterationszähler $n$ durch den Zähler für die Kandidatenzyklen $r$ ersetzt. Der in Gl. (6.10c) auftretende Para-

---

[18] Dies wird beim Aufwandsvergleich der unterschiedlichen Netzwerke in Abschnitt 6.3.2 noch diskutiert, vgl. Gln. (6.16) und (6.20).

[19] Für die experimentellen Untersuchungen wurde ein Software-Simulationssystem (Crowder and Fahlman 1991) benutzt, das interessierten Benutzern über elektronische Mail als C-Sourcecode zur Verfügung steht.

meter $R_{max}$ wurde auf 50 gesetzt. Da dieser Wert in den Experimenten nie erreicht wurde, erfolgte der Trainingsabbruch stets aufgrund einer der Bedingungen in Gln. (6.10a) oder (6.10b). Um die Vergleichbarkeit mit den MLP-Ergebnissen zu gewährleisten wurden die Abbruchbedingungen auf Basis des MSE formuliert (vgl. hierzu auch Gl. (6.3) und Fußnote 16).

- **Kontinuierliche Ausgangswerte.** Da die ursprüngliche Version des Algorithmus für binäre Ausgangswerte ausgelegt war, wurde das System für Abbildungsaufgaben mit reellen Ausgangswerten modifiziert.

Die bisher beschriebenen Modifikationen wurden notwendig, um das Verfahren an die Geräuschreduktionsaufgabe anzupassen. Die bei der MLP-Entwicklung vorgenommenen problemspezifischen Erweiterungen (z. B. Multi-SNR-Training und Kontextinformation am Netzwerkeingang) wurden für das CC-Lernverfahren nachgerüstet und führten zu ähnlichen Ergebnisverbesserungen wie beim MLP.

Der Trainingsverlauf des CC-Lernalgorithmus wird durch eine Reihe von Parametern gesteuert (siehe Crowder and Fahlman 1991). Sie werden dem Simulationsprogramm durch eine Steuerdatei übergeben, wenn die voreingestellten Werte überschrieben werden sollen. Die Parameterwerte sind im Anhang A.3 aufgelistet.

Einige dieser Parameter behandeln Sonderfälle für bestimmte Verläufe der Fehlerkurve während des QP-Trainings und werden abhängig vom jeweiligen Datensatz bestimmt. Nach Optimierung der Trainingsparameter auf additives Druckergeräusch konnten mit MLP-Resultaten vergleichbare Ergebnisse erzielt werden. Tests mit anderen Daten ohne weitere Parameteroptimierung führten jedoch zu Verschlechterungen, was beim Vergleich mit den Worterkennungsraten nach MLP-basierter Geräuschreduktion deutlich wird (siehe Abschnitt 6.3).

Weitere Modifikationen umfassen die Einführung einer *Quickprop-Lernrate* und das Entfernen niedrig gewichteter Verbindungen (*Modellreduktion*):

- **Quickprop-Lernrate.** Der (bekannte) Verlauf der quadratischen Approximation der Fehlerkurve erlaubt einen direkten Sprung ins Minimum. Dies ist bei einer fehlerhaften Approximation mit der aus QP berechneten Gewichtsmodifikation risikobehaftet; eine vorsichtigere Strategie könnte die Annäherung an das Minimum durch Multiplikation der berechneten Gewichtsmodifikation mit einem Faktor $\beta<1$ darstellen. In Experimenten konnte kein Vorteil festgestellt werden; im Minimum erhielt man nahezu gleiche MSE-Werte für $0{,}6<\beta<1{,}5$;

allerdings sank die Zahl der benötigten Iterationen zum Training der Ausgangsgewichte mit zunehmendem $\beta$ von 27 für $\beta$=1,0 auf 19 für $\beta$=1,4. Dies bedeutet eine Reduktion der Iterationszahl auf ca. 70% des ursprünglichen Wertes. Die Abhängigkeit zwischen $\beta$ und der mittleren Zahl der hinzugefügten Knoten verhielt sich gegenläufig: im Minimum bei $\beta$=0,7 erreichte sie einen geringfügig niedrigeren Wert als für $\beta$=1,0.

- **Modellreduktion**. Da die Vernetzungsstruktur beim CC- erheblich komplexer als beim MLP-Netzwerk ist, kann das Entfernen von Verbindungen mit betragsmäßig kleinen Gewichtswerten zu Aufwandseinsparungen ohne Ergebniseinbußen führen. Dies kann aufgrund des geringen Signalflusses über solche Verbindungen angenommen werden. Der Algorithmus wurde so modifiziert, daß nach Abschluß jedes Kandidatenzyklus Gewichte der Matrizen $\boldsymbol{B}$ und $\boldsymbol{D}$ (vgl. Bild 6.1e) mit Beträgen kleiner als 0,25 aus dem Netzwerk entfernt wurden. Damit wurde eine Reduktion der Gewichtszahl um ca. 15-20 % bei vergleichbaren MSE-Werten erreicht. Die Reduktion betragsmäßig größerer Gewichte führte zu einer erhöhten Zahl neu hinzugefügter Knoten.

Ein Vergleich der Worterkennungsraten sowie des Aufwands für CC-, MLP- und RAN-basierte Geräuschreduktion wird in Abschnitt 6.3 diskutiert.

### 6.2. Resource Allocating Network

**Grundgedanke.** Beim *Resource Allocating Network* (Platt 1991) wird die Übertragungsfunktion durch Überlagerung gewichteter Summen von Radial Basis Functions approximiert; sie stellen eine Erweiterung der gaußförmigen Aktivierungsfunktionen nach Gl. (3.3) auf vektorielle Eingangsgrößen und nichtverschwindende Mittelwertsvektoren dar. Der Funktionswert des $i$-ten Knotens ist durch

$$g_{RBF_i}(\boldsymbol{x}) = exp\left\{-\frac{1}{\sigma_i^2}\left\|\boldsymbol{u}_i - \boldsymbol{x}\right\|\right\} \qquad (6.11a)$$

gegeben, wobei $\sigma_i^2$ die Varianz und $\boldsymbol{u}_i$ der Mittelwertsvektor (*Center Vector*) des $i$-ten Knotens bedeuten; dabei steht die Norm ||...|| in Gleichung (6.11a) für den quadratischen Fehler zwischen dem Eingangsvektor $\boldsymbol{x}$ und $\boldsymbol{u}_i$ nach Gl. (3.7).

**Vorwärtsrechnung**. Bei der im Bild 6.2 gezeigten Topologie des RAN sind die RBF-Funktionen kurz als $g_i(\boldsymbol{x})$, $i = 1,\ldots,h$, bezeichnet (ohne Knoten $h+1$); aus Gründen der Übersichtlichkeit wurde nur ein einziger Ausgangsknoten angenommen. Während der Vorwärtsrechnung wird der eingangsseitig anliegende Vektor an die verdeckten Knoten weitergeleitet, durch die Aktivierungsfunktionen bewertet und über die mit $v_i$ gewichteten Verbindungen zum Ausgangsknoten weitergeben und aufsummiert. Daher gilt für die Abbildungsfunktion des Netzwerks in Bild 6.2

$$F(\boldsymbol{x}) = v_0 + \sum_{i=1}^{h} v_i g_i(\boldsymbol{x}) \quad , \qquad (6.11b)$$

wobei $v_0$ für den im Bild nicht gezeichneten Offseteingang des Ausgangsknotens steht, dessen Aktivierungsfunktion linear ist und die Steigung 1 besitzt. Aus Gln. (6.11a) und (6.11b) wird deutlich, daß die Abbildungsfunktion durch Linearkombination der Ausgangswerte der RBF-Funktionen zusammengesetzt wird.

Ein Vergleich der Bilder 6.2 und 3.2 zeigt nahezu identische Topologieen für MLP und RAN; Unterschiede zum MLP sind neben der dynamischen Allokierung neuer Knoten (z. B. Knoten $h+1$ in Bild 6.2) vor allem das Fehlen von Gewichten bei den Verbindungen von der Eingangs- zur Zwischenschicht sowie die Radial Basis Functions in den verdeckten Knoten.

**Speicherung neuer Beobachtungen und Gewichtsmodifikation.** Die Strategie beim Netzaufbau besteht in der expliziten Speicherung von Eingangsvektoren $\boldsymbol{x}$

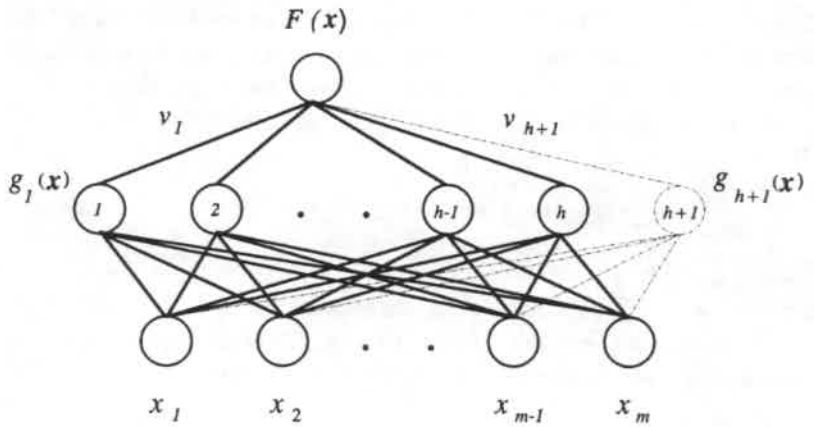



Bild 6.2: Topologie des *Resource Allocating Network*.

als Mittelwertsvektoren $u_i$ in den verdeckten Knoten des Netzwerks. Durch die zugehörigen Gaußfunktionen wird der Merkmalsvektorraum in gaußförmige Bereiche aufgeteilt, deren Lage im Raum durch die Trainingsvektoren und deren Einflußradius durch ihre Standardabweichung bestimmt werden. Im Test liefern eingangsseitig anliegende Vektoren hohe Ausgangsaktivierungen an den ihnen zugeordneten und höchstens geringe an den übrigen verdeckten Knoten.

Im folgenden wird der Trainingsalgorithmus des Netzwerks mit den Bezeichnungen nach Bild 6.2 im Hinblick auf die Anwendung zur Geräuschreduktion kurz beschrieben. Eine detailliertere Herleitung des Lernverfahrens ist bei Kadirkamanathan und Niranjan (1993) zu finden. Die Erweiterung auf $q$ Ausgänge erfolgt durch Ersetzen der zugehörigen skalaren Größen durch Vektoren.

Im Gegensatz zum MLP wird die RAN-Struktur in der Trainingsphase dynamisch erweitert. Über die Installation neuer Knoten in der Zwischenschicht entscheidet der Neuheitsgrad anliegender Eingangsmuster. Dieser wird mit dem *Novelty Criterion* geprüft; nach Anlegen des $l$-ten Trainingsmusters in der $n$-ten Trainingsiteration wird ein neuer Knoten installiert, wenn die beiden Bedingungen

$$\left(e^l\right)^2 = \left(t^l - F\left(x^l\right)\right)^2 > e_{min} \qquad \text{und} \qquad (6.12a)$$

$$\min_i \left\| u_i - x^l \right\| > \varepsilon^n \qquad i = 1,\ldots,h \qquad (6.12b)$$

gleichzeitig erfüllt sind. Dabei ist $e^l$ der quadratische Ausgangsfehler, $e_{min}$ eine vorgegebene Fehlerschwelle und $\varepsilon^n$ eine Schwelle für den Abstand des Eingangsvektors zum nächstgelegenen Mittelwertsvektor. In Gl. (6.12a) wird geprüft, ob der Ausgangsfehler die Schwelle $e_{min}$ übersteigt, und bei Erfüllung von (6.12b) fällt der $l$-te Eingangsvektor in keinen der bisher im Netzwerk gespeicherten Bereiche. Dabei bestimmt der Wert von $\varepsilon^n$ die Auflösung bei der Bereichsbildung (*Input Resolution*) während der $n$-ten Iteration.

Bei Erfüllung der Bedingungen wird ein neuer Knoten mit dem Mittelwertsvektor $u_{h+1} = x$ und dem Ausgangsgewicht $v_{h+1}$ angefügt. Seine Parameter werden mit

$$v_{h+1} = e^l \qquad , \qquad (6.13a)$$

$$u_{h+1} = x^l \qquad \text{und} \qquad (6.13b)$$

$$\sigma_{h+1} = \kappa \min_i \left\{ \left\| x^l - u_i \right\| \right\} \qquad (6.13c)$$

initialisiert, wobei Gl. (6.13c) den Einflußradius des neuen Bereichs angibt.

$\varepsilon^n$ in Gl. (6.12b) nimmt mit der Zahl der Trainingsiterationen bis zum Erreichen einer unteren Schwelle $\varepsilon_{min}$ ab und wird aus

$$\varepsilon^n = max\left\{\varepsilon_{max}\gamma^n, \varepsilon_{min}\right\}$$

bestimmt. Zusammen mit Gl. (6.13c) bedeutet dies ausgehend vom Anfangswert $\varepsilon_{max}$ eine mit $\gamma^n$ abnehmende Standardabweichung und somit eine durch kleinerwerdende Radien bedingte zunehmende Auflösung der neu hinzugefügten Bereiche bis zur unteren Schwelle $\varepsilon_{min}$.

Wenn kein neuer Knoten angefügt wird, erfolgt eine Adaption der Netzwerkparameter $v_i$ und $\boldsymbol{u}_i$ durch Gradientenabstieg gemäß

$$v_0^{neu} = v_0^{alt} + lr\left[t^l - F\left(\boldsymbol{x}^l\right)\right]\frac{\partial F\left(\boldsymbol{x}^l\right)}{\partial v_0} \quad \text{mit} \quad \frac{\partial F\left(\boldsymbol{x}^l\right)}{\partial v_0} = 1 \quad , \tag{6.14a}$$

$$v_i^{neu} = v_i^{alt} + lr\left[t^l - F\left(\boldsymbol{x}^l\right)\right]\frac{\partial F\left(\boldsymbol{x}^l\right)}{\partial v_i} \quad \text{mit} \quad \frac{\partial F\left(\boldsymbol{x}^l\right)}{\partial v_i} = g_i\left(\boldsymbol{x}^l\right) \; , \; i = 1,\ldots,h \tag{6.14b}$$

$$\boldsymbol{u}_i^{neu} = \boldsymbol{u}_i^{alt} + lr\left[t^l - F\left(\boldsymbol{x}^l\right)\right]\frac{\partial F\left(\boldsymbol{x}^l\right)}{\partial \boldsymbol{u}_i} \quad \text{mit} \quad \frac{\partial F\left(\boldsymbol{x}^l\right)}{\partial \boldsymbol{u}_i} = g_i\left(\boldsymbol{x}^l\right)\frac{2v_i}{\sigma_i^2}\left(\boldsymbol{x}^l - \boldsymbol{u}_i\right)^T \; . \tag{6.14c}$$

In den Gln. (6.14a-c) treten die partiellen Ableitungen des Ausgangsfehlers nach den Netzwerkparametern auf. Sie entsprechen dem von den adaptiven Filtern bekannten LMS-Algorithmus (Widrow et al. 1975). Die Wahl der Trainingsparameter in Gln. (6.12a) bis (6.14c) wird im folgenden Abschnitt beschrieben.

**Anpassung an die Geräuschreduktion**. Die Anpassung des RAN an die Geräuschreduktionsaufgabe sowie die Optimierung der Trainingsparameter für die verwendeten Datenbasen wurde von Chen (1994) untersucht. Hierzu war eine Erweiterung des Netzwerks auf $q$ Ausgangsdimensionen notwendig. Experimentelle Untersuchungen zeigten vergleichbare und in einigen Fällen auch bessere Ergebnisse als mit dem MLP. Die Parameteroptimierung ergab folgende Werte: $\varepsilon_{min}$=0,5 , $\varepsilon_{max}$=1,9, $\varepsilon^n$=0,1 , $\kappa$=0,87 , $\gamma$=0,995 und $lr$=0,005. Tests mit nicht zur Parameteroptimierung verwendeten Daten sowie ein Vergleich mit CC- und MLP-Netzen werden im nächsten Abschnitt beschrieben.

## 6.3 Vergleich automatisch generierter Netzwerke mit dem Multilayer Perzeptron

In diesem Abschnitt werden die automatisch generierten Netzwerke mit dem MLP bezüglich ihrer Geräuschreduktionsleistung sowie dem Aufwand für die Trainings- und Testphase verglichen.

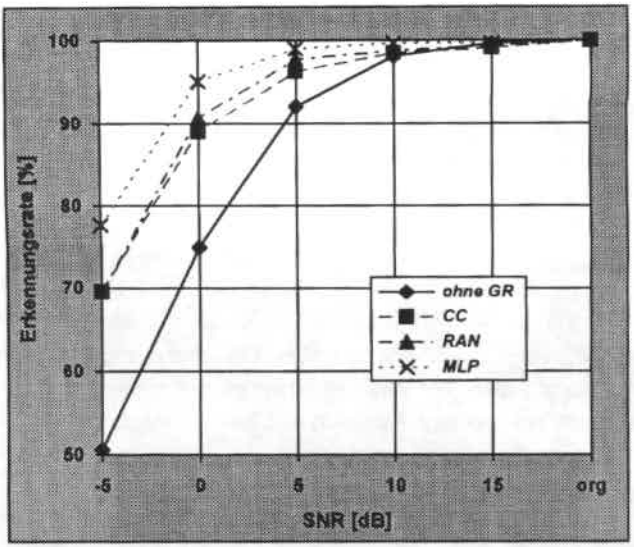
### 6.3.1. Worterkennungsergebnisse

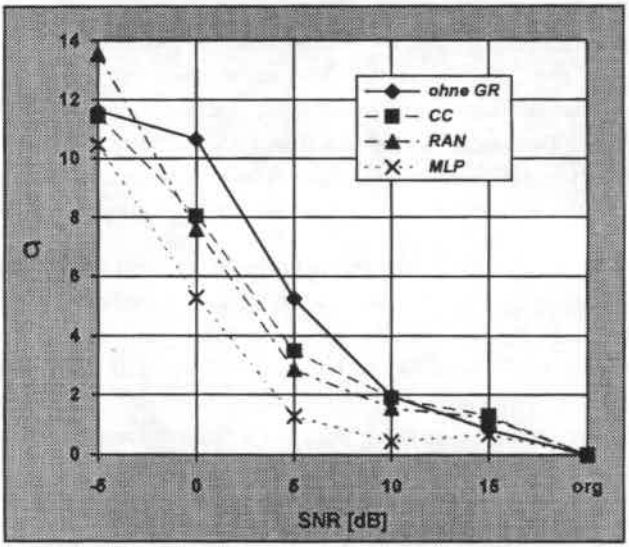
Aus Aufwandsgründen wurden ausschließlich geräuschabhängige Experimente mit Trainingsdaten aus jeweils einer einzelnen Geräuschquelle (im Gegensatz zu Pooltraining) durchgeführt. Die Versuchsreihen wurden mit beiden Geräuschaufnahmen aus dem Testpool (vgl. Kapitel 3) und jeweils zehn Sprechern wiederholt und die Ergebnisse anschließend gemittelt. Als Vergleichswerte dienten die Erkennungsraten ohne Geräuschreduktion.

Alle Netzwerke hatten 50 Eingangs- und 10 Ausgangsknoten, was bei zehndimensionalen Merkmalsvektoren fünf Kontextvektoren am Eingang entspricht. Die MLP-Ergebnisse sind aus Abschnitt 5.4.2 entnommen (obere Kurve, "tra=abh", in Bild 5.12b), die beiden fehlenden Simulationsreihen mit den automatisch generierten Netzwerken wurden unter gleichen Versuchsbedingungen nachgeholt. Die Parameter für das CC- bzw. RAN-Netzwerk wurden wie in den Abschnitten 6.1 und 6.2 beschrieben gewählt, ohne eine zusätzliche, datensatzabhängige Optimierung der Trainingsparameter durchzuführen.

Bild 6.3a zeigt den Vergleich der **Worterkennungsergebnisse** in Abhängigkeit vom SNR. Obwohl mit allen drei Netzwerken erhebliche Verbesserungen der Erkennungsrate erzielt werden, erreicht keines der beiden automatischen Verfahren die MLP-Ergebnisse (obere Kurve). Insbesondere bei stark gestörten Sprachsignalen (SNR < 5 dB) werden mit MLP-basierter Geräuschreduktion ca. 5-8 % bessere Erkennungsraten erreicht. Von den beiden automatischen Verfahren liefert das RAN bei 0 und 5 dB SNR etwas bessere Ergebnisse.

Bild 6.3b zeigt die **Standardabweichungen** der Resultate aus Bild 6.3a. Daraus kann entnommen werden, daß die mit dem MLP erzielten Erkennungsraten die niedrigsten Streuungswerte aufweisen. Zwischen dem RAN und dem CC-Netzwerk sind mit Ausnahme des Ergebnisses bei -5 dB (höhere Streuung der mit dem RAN erzielten Werte) nur geringfügige Unterschiede festzustellen.




Bilder 6.3a und b: Erkennungsraten (6.3a) und deren Streuung (6.3b) nach Geräuschreduktion mit CC-, RAN- und MLP-Netzwerken. Training mit Einzelgeräuschen, Mittel über zwei Aufnahmen des Testpools und zehn Sprecher.

### 6.3.2. Aufwandsvergleich der unterschiedlichen Netzwerkmodelle

Ein Problem der Aufwandsabschätzung liegt im Übergang von Simulations- zu Echtzeitsystemen[20]. Während bei Simulationssystemen Wert auf die notwendige Flexibilität durch zahlreiche Parameter zur Steuerung des Programmablaufs gelegt wird, steht bei Echtzeitsystemen neben der Funktionalität vor allem die Aufwandsminimierung im Vordergrund.

Wegen der unterschiedlichen Architektur von Echtzeit- und Simulationssystemen ist ein objektives Vergleichsmaß (wie z. B. Rechenzeit) zwischen beiden oft nicht zugänglich. Experimentiersysteme werden häufig von mehreren Anwendern gleichzeitig benutzt; daher spielt die Maschinenauslastung bei manchen Vergleichsgrößen eine Rolle. Darüber hinaus sind maschinen- und programmspezifische Besonderheiten von Bedeutung. Insbesondere der Speicherbedarf unterschiedlicher Programme oder Programmteile sowie die Bestückung der Rechenanlage mit Haupt- und Cachespeicher beeinflussen die Rechenzeit über die Zahl von Auslagerungen eines Prozesses auf ein nichtflüchtiges Speichermedium während seiner Gesamtlaufzeit.

Beim Auswandsvergleich wird zwischen den Betriebsarten *Training* und *Test* (oder *Vorwärtsrechnung*) unterschieden. Hierzu werden zunächst Vergleichskriterien diskutiert und anschließend anhand der Experimente im vorigen Abschnitt eine Aufwandsabschätzung für die unterschiedlichen Netzwerke durchgeführt.

Bei kostenrelevanten Entscheidungen interessiert zusätzlich der Implementierungs- und Optimierungsaufwand, der hier über die bereits genannten Unterschiede von experimenteller Bestimmung der Topologie versus automatischer Netzgenerierung hinaus nicht untersucht wird.

**Aufwandsabschätzung für die Trainingsphase:**

- **Iterationszahl.** Die Iterationszahl eignet sich vor allem zum Vergleich unterschiedlicher Trainingsläufe bei gleichem Datensatz und identischem Netzwerktyp. Bei unterschiedlichen Trainingsverfahren oder unterschiedlichen Datensätzen sind Iterationszahlen wenig aussagefähig. Bei automatisch generierten Netzen ändert sich oft die Zahl der Gewichtsmodifikationen, die in

---

[20] Siehe auch Anmerkungen zur Aufwandsabschätzung in Abschnitt 3.5.

eine Iteration eingehen; außerdem können die Netzwerke unterschiedliche Zahl von Knoten sowie unterschiedliche Vernetzungsstrukturen in den einzelnen Iterationen aufweisen.

- **Zahl der neu hinzugefügten Knoten.** Dieses Kriterium eignet sich nur für automatische Netzwerkgenerierungsverfahren mit konstruktiver Strategie; aufgrund der großen Unterschiede der Algorithmen[21] ist dieses Kriterium nur zum Vergleich von Netzwerken, die mit dem gleichen Trainingsalgorithmus erzeugt werden, von Bedeutung.

- **Rechenzeit**. Die benötigte Rechenzeit ist u. a. maschinen-, auslastungs- und implementierungsabhängig. Außerdem muß im Hinblick auf die Echtzeitimplementierung berücksichtigt werden, daß in den Rechenzeiten aus der Simulationsumgebung die Zeit für Initialisierungsoperationen, die nur einmal bei Programmstart anfallen, enthalten ist. Berücksichtigt man dies, so sind unterschiedliche Netzwerkmodelle höchstens bei gleichen Rechnern mit gleicher Auslastung vergleichbar.

**Aufwandsabschätzung für die Erkennungsphase:**

- **Zahl der Knoten.** Die Knotenzahl ohne Eingangsknoten gibt die erforderlichen Summationen über die am jeweiligen Knoten eingangsseitig eintreffenden Erregungen an sowie die Zahl der zu berechnenden Ausgangsaktivierungen (oder der Leseoperationen für abgespeicherte Funktionswerte, *Table Lookup*). Dieses Kriterium vernachlässigt unterschiedliche Vernetzungsstrukturen, wie beispielsweise die komplexe Vernetzung höherer Knoten beim CC-Netzwerk und die geringere Vernetzungskomplexität beim RAN. Da die Zahl der Ein- und Ausgangsknoten aufgabenabhängig und daher für alle Netzmodelle gleich ist, kann stattdessen auch die Zahl der verdeckten Knoten verglichen werden, die entweder beim Entwurf der Netztopologie vorab festgelegt wird (MLP) oder im Verlauf des Trainings gelernt wird (CC,RAN).

---

21) Bei Cascade Correlation: komplette Minimumsuche mit Gradientenabstieg pro Kandidatenknoten; beim Resource Allocating Network: einfache Entscheidung über Neuinstallation sowie Bestimmung der Parameter des Knotens aus dem aktuellen Trainingsvektor.

- **Zahl der Gewichte.** Die Zahl der Gewichte gibt gleichzeitig die erforderliche Zahl der Multiplikationen und Akkumulationen bei einer Echtzeitimplementierung an. Darüber hinaus bestimmt sie den benötigten Speicherplatz für die Gewichtsfaktoren. Dieses Maß ist unabhängig von der Vernetzungsstruktur. Nicht berücksichtigt werden dabei die knotenbezogenen Operationen wie die Summation der Eingangserregungen und die Berechnung der Aktivierungsfunktion.

- **Rechenzeit.** Die Rechenzeit ist prinzipiell zur Aufwandsabschätzung geeignet. Bei ihrer Bestimmung ergeben sich in der Praxis jedoch Ungenauigkeiten, da wegen der kurzen Rechenzeiten für die Vorwärtsrechnung der Verwaltungsaufwand für Programminitialisierung und Umschaltung zwischen unterschiedlichen Prozessen bei Multitasking Systemen stark ins Gewicht fällt.

Aus den genannten Gründen wird zum überschlägigen Aufwandsvergleich für die unterschiedlichen Netzmodelle die Rechenzeit für die Trainings- sowie die Zahl der Knoten und Gewichte für die Erkennungsphase herangezogen. Hierzu werden im folgenden die Berechnungsvorschriften für die Zahl der Knoten und Gewichte der MLP-, CC- und RAN-Netztopologien angegeben.

**Zahl der Knoten und Gewichte.** Bei ihrer Berechnung steht $m$ für die Zahl der Eingangsknoten, $h$ für die der verdeckten und $q$ für die Zahl der Ausgangsknoten.

**Zahl der Knoten**. Bei allen drei Netzwerken ergibt sich die Knotenzahl aus der Summe der Eingangs-, der verdeckten und der Ausgangsknoten. Die Gesamtknotenzahl $N_{Knoten}$ ist daher

$$N_{Knoten} = m + h + q \tag{6.15}$$

**Zahl der Gewichte.** Hierfür ergeben sich je nach Netztyp unterschiedliche Berechnungsvorschriften. Für ein dreischichtiges MLP ist die Zahl der Gewichte $G_{MLP}$ durch

**MLP:**

$$\begin{aligned} G_{MLP} &= mh + hq + h + q \\ &= h(m + q + 1) + q \end{aligned} \tag{6.16}$$

gegeben. Die ersten beiden Terme der rechten Seite in der oberen Zeile von Gl. (6.16) ergeben sich durch vollständige Verbindung der $m$ Eingangsknoten mit den $h$ verdeckten Knoten sowie der $h$ verdeckten mit $q$ Ausgangsknoten. Die

beiden letzten Summanden repräsentieren die Gewichte des Biaseingänge der verdeckten und der Ausgangsknoten.

Für das CC-Netzwerk müssen zur Gewichtsberechnung die Gewichtsmatrizen $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ und $\boldsymbol{D}$ nach Bild 6.1e gesondert betrachtet werden. Sie werden mit den Gewichtszahlen $G_A$, $G_B$, $G_C$ und $G_D$ aus

$$\begin{aligned} G_A &= mq \quad , \quad G_B = mh \quad , \\ G_C &= hq \quad , \quad G_D = \frac{h(h-1)}{2} \end{aligned} \tag{6.17}$$

bestimmt. Hierzu müssen noch

$$G_{Bias} = h + q \tag{6.18}$$

Bias-Gewichte für jeden verdeckten bzw. für jeden Ausgangsknoten hinzuaddiert werden. Die Gesamtzahl der Gewichte beim CC-Netzwerk beträgt daher

$$G_{CC} = G_A + G_B + G_C + G_D + G_{Bias} \quad . \tag{6.19}$$

Einsetzen von (6.17) und (6.18) in (6.19) ergibt

**CC:**
$$G_{CC} = h\left(m + \frac{h+1}{2} + q\right) + q(m+1) \quad . \tag{6.20}$$

Die Zahl der Gewichte beim RAN-Netzwerk bestimmt sich aus den Verbindungen zwischen der verdeckten und der Ausgangsschicht zu

**RAN:**
$$G_{RAN} = hq \quad . \tag{6.21}$$

Ein Vergleich der Gln. (6.16), (6.20) und (6.21) zeigt, daß bei gleicher Knotenzahl für die Zahl der Gewichte

$$G_{RAN} < G_{MLP} < G_{CC} \tag{6.22}$$

gilt. Gemessen an der Gewichtszahl ist daher für das RAN der geringste Aufwand zu erwarten; die CC-Struktur weist bei gleicher Knotenzahl stets mehr Verbindungen als die MLP-Struktur auf.

**Experimenteller Aufwandsvergleich.** Dem Vergleich liegen die Daten der Netzwerke zugrunde, mit denen die Ergebnisse in den Bildern 6.3a und 6.3b erzielt wurden. Aufgrund der gleichen Ein- und Ausgangsdimensionalität für alle Netzwerke enthalten die Eingangsschichten für fünf Kontextvektoren mit je 10 Koeffizienten fünfzig Knoten; die Ausgangsschichten enthalten jeweils zehn Knoten.

Die Vergleichszahlen in der ersten Spalte von Tabelle 6.1 geben die für das Training benötigte **mittlere Rechenzeit pro Sprecher** für alle drei Netzwerkmodelle auf einer Sun SPARC 10 Workstation an. Aus den genannten Gründen stellen diese Angaben lediglich einen Anhaltspunkt dar, da sie von der Maschinen- und Netzwerkbelastung abhängen. Aus den Vergleichszahlen kann festgestellt werden, daß die Trainingszeit für MLP-Netzwerke ca. eine halbe Größenordnung über denen der automatischen Netzgenerierungsverfahren liegt.

Die mittlere **Zahl der verdeckten Knoten** zeigt deutliche Unterschiede zwischen den drei Netzwerktypen. Das CC-Netzwerk ist mit 4,4 verdeckten Knoten im Mittel das kompakteste Netzwerk, da die verdeckten Knoten nur die nichtlinearen Anteile der Übertragungsfunktion repräsentieren (vgl. Abschnitt 6.1); die linearen Anteile werden parallel dazu über die Kurzschlußverbindungen der Abbildungsmatrix $A$ in Gl. (6.4) realisiert. Die RAN-Struktur benötigt für dieselbe Aufgabe mehr als doppelt soviele verdeckte Knoten wie das MLP, und zwischen den Zahlen für die RAN- und CC-Netzwerke liegt etwa ein Faktor von zehn.

Bei der **Zahl der Gewichte** spiegelt sich die komplexere Vernetzungsstruktur des CC-Netzwerks wieder, das trotz der geringen Zahl von verdeckten Knoten noch 64 % der Gewichte des MLP enthält und ca. 175 % der Gewichte des RAN.

Anhand der Programmlaufzeiten für die **Vorwärtsrechnung** ließ sich verifizieren, daß das RAN die kürzesten Rechenzeiten aufweist.

Tabelle 6.1: Vergleich zwischen MLP-, CC- und RAN-Netzwerken. Mittel über zwei Simulationsreihen mit jeweils zwei Geräuscharten und zehn Sprechern.

| | Training | Erkennung | |
|---|---|---|---|
| | Rechenzeit [min] | verd. Knoten | Gewichte |
| MLP | 110,0 | 20,0 | 1230 |
| CC | 18,5 | 4,4 | 793 |
| RAN | 24,0 | 45,2 | 452 |

## 6.4 Zusammenfassung der automatischen Netzgenerierung

Sowohl das *Cascade Correlation*-Lernverfahren als auch das *Resource Allocating Network* wenden eine konstruktive Strategie für die Netzwerkgenerierung an, bei der ausgehend von einer Minimalstruktur iterativ verdeckte Knoten angefügt werden. Dies wird solange fortgesetzt, bis ein Abbruchkriterium erfüllt ist.

Anfängliche Geräuschreduktionsexperimente mit beiden automatisch generierten Netzwerktypen zeigten mit MLP-Resultaten vergleichbare Ergebnisse. Tests mit neuen Daten, die nicht zur Parameteroptimierung benutzt wurden, zeigten jedoch die Empfindlichkeit der Lernparameter bei den automatischen Verfahren gegenüber einem Wechsel der Geräuschquelle. Ein Vergleich der Netzwerktypen MLP, CC und RAN ergab daher suboptimale Erkennungsergebnisse nach Geräuschreduktion mit beiden automatisch generierten Netzwerken.

Netzwerkparameter und Trainingsstatistik automatisch generierter Netzwerke eignen sich nur bedingt für einen direkten Aufwandsvergleich, da die meisten Größen modellabhängig sind und daher unterschiedliche Bedeutung haben. Zum Vergleich des Trainingsaufwands wurde die Rechenzeit gemessen, für die Testphase wurden die Zahlen der verdeckten Knoten sowie der Gewichte als Vergleichsparameter herangezogen. Die kompaktesten Netzwerke bezüglich der Zahl der verdeckten Knoten erhält man mit Cascade Correlation, die wenigsten Gewichte enthält das RAN. Beide Netzwerke benötigen ca. fünfmal weniger Trainingszeit als das MLP.

Wünschenswert wäre eine Kombination der Geräuschreduktionsleistung des MLP mit den Rechenzeiten der automatisch generierten Netzwerke. Im 7. Kapitel wird daher ein Ansatz zur schnellen Adaption der MLP-Netzwerke an instationäre Geräuschsignale in den Sprachpausen ohne zeitaufwendiges Neutraining untersucht.

# 7. NETZWERKADAPTION MIT GERÄUSCHPARAMETERN

## 7.1. Funktionsweise der Netzwerkadaption mit Geräuschparametern

Zur Netzwerkadaption[22] bieten sich Parameter an, die die momentanen statistischen Eigenschaften des Geräuschsignals repräsentieren (z. B. Information aus dem Zeit-, Spektral- oder Cepstralbereich). Bild 7.1 illustriert die Arbeitsweise: während die geräuschbehafteten Merkmalsvektoren fortlaufend im Segmentraster abgebildet werden, wird die Abbildungsfunktion des Netzwerks in den Pausen durch Aktualisierung der Steuerinformation adaptiert. Dies geschieht durch ihre Neuberechnung und somit ohne zusätzliches Netzwerktraining in der Testphase.

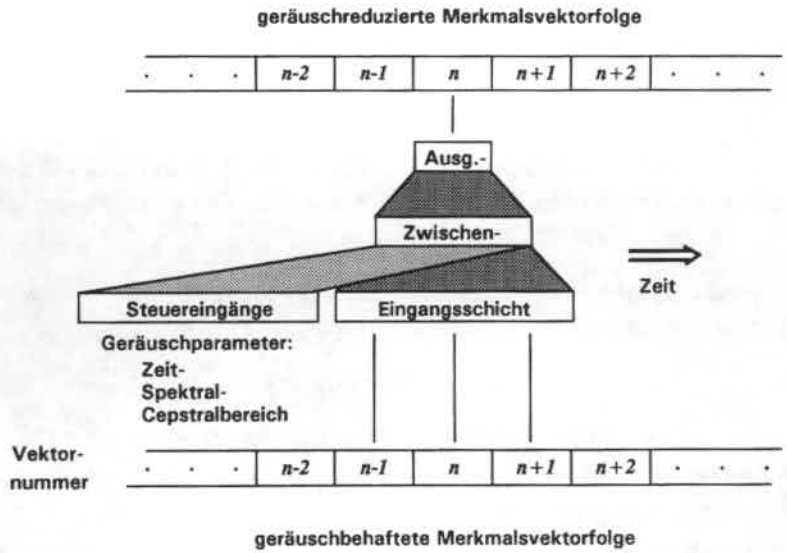



Bild 7.1: Adaptive vektorweise Geräuschreduktion mit Steuereingängen.

---

[22] vgl. Abschnitt 3.4.2

Während der auf eine Pause folgenden Sprachäußerung bleiben die Parameterwerte an den Steuereingängen[23] unverändert. Für die Aufnahme der Parameter ist eine Erweiterung der Eingangsschicht nach Bild 7.1 erforderlich. Den Betrachtungen zur Netzwerkadaption liegen folgende Annahmen zugrunde:

- Die Zahl der Sprachinformation tragenden Koeffizienten am Netzwerkeingang und -ausgang ist gleich groß. Eingangsvektor $\boldsymbol{x}$, Ausgangsvektor $\boldsymbol{y}$ und Zielvektor $\boldsymbol{t}$ haben daher gleiche Dimensionalität $q$, d. h. $\boldsymbol{x}\in R^q$, $\boldsymbol{y}\in R^q$ und $\boldsymbol{t}\in R^q$. Eine Erweiterung Kontextinformation ist durch Erhöhung der Dimensionalität in der Eingangsschicht leicht möglich.
- Die Steuerparameter mit den Geräuschkoeffizienten werden im $r$-dimensionalen Vektor $\boldsymbol{n}$ zusammengefasst, $\boldsymbol{n}\in R^r$.
- Die Speicherung von Sprach- und ggf. der Geräuschinformation erfolgt in erweiterten Merkmalsvektoren $\boldsymbol{v}$ mit der Dimensionalität $m$.

Dann können für den Eingangsvektor $\boldsymbol{v}$ zwei Betriebsarten unterschieden werden:

1. Ohne Netzwerkadaption in den Sprachpausen sind $\boldsymbol{v}$ und $\boldsymbol{x}$ identisch:
$$\boldsymbol{v}=\boldsymbol{x}\in R^m \text{ und } m=q.$$
2. Mit Netzwerkadaption setzt sich $\boldsymbol{v}$ aus Sprach- und Geräuschkoeffizienten zusammen:
$$\boldsymbol{v}^T=\{\boldsymbol{x}^T,\boldsymbol{n}^T\}\in R^m \text{ mit } m=q+r.$$

In den folgenden Betrachtungen soll der 2. Fall betrachtet werden. Dabei wird die Adaption der Übertragungsfunktion in die beiden Schritte *initiales Training* und *Netzwerkadaption in den Sprachpausen* aufgeteilt.

Die Abbildung eines Eingangsvektors $\boldsymbol{v}$ auf einen Ausgangsvektor $\boldsymbol{y}$ im $n$-ten Segment kann dann mit der Abbildungsfunktion $\boldsymbol{F}()$ durch

$$\boldsymbol{y}(n) = \boldsymbol{F}(\boldsymbol{w}(n),\boldsymbol{v}(n)) \qquad (7.1)$$

ausgedrückt werden. Dabei steht $\boldsymbol{w}(n)$ für den aktuellen Gewichtsvektor. Für die Netzwerkadaption mit Geräuschparametern können die Argumente von $\boldsymbol{F}()$ nach Gl. (7.1) nach Sprach- und Geräuschanteilen getrennt angegeben werden:

---

[23] Eine Ausnahme bildet das segmentbezogene Signal-zu-Rausch-Verhältnis nach Gl. (7.19), das vektorweise neu berechnet wird.

$$y(n) = F\big(w(n), n(n), x(n)\big) \quad . \tag{7.2}$$

**Initiales Training**. Während des initialen Trainings wird durch Optimierung der Netzparameter der mittlere quadratische Fehler $MSE_T$ (im Trainingsdatensatz) nach Gleichung (3.9) bzw. der $MSE_V$ (im Verifikationsdatensatz) minimiert. Dabei besitzen der Eingangsvektor $v \in R^m$ und der geräuschfreie Zielvektor $t \in R^q$ unterschiedliche Dimensionalität. Mit Gln. (3.9) und (7.2) errechnet sich der $MSE_T$ aus den $L$ Trainingsvektorpaaren $\{v^l, t^l\}$, $l=1,\ldots,L$, aus

$$MSE_T = \frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{q}\left(t_j^l - F_j^l(w,n,x)\right)^2 \quad . \tag{7.3}$$

Optimierung der Netzwerkparameter durch Minimumsuche liefert bei gegebenen Trainingsdaten den Gewichtsvektor $w_{min}$:

$$w_{min} = \underset{w}{argmin}\left\{MSE_T(w)\right\} \quad . \tag{7.4}$$

Durch Einfrieren der Gewichte nach Trainingsabschluß hängt $F$ nach Gl. (7.2) nur noch von den Eingangsdaten ab und nimmt die Form

$$F(w,v) \xrightarrow{\arg\min\left(MSE_T(w)\right)} F_{w_{min}}(v) \tag{7.5}$$

an. $F_{w_{min}}$ wird künftig kurz mit $F_w$ bezeichnet. Da nach Trainingsabschluß die Gewichtskoeffizienten $w$ unverändert bleiben, können die Ausgangsvektoren im $n$-ten Segment $y(n)$ als Funktion der Eingangsvektoren ausgedrückt werden:

$$y(n) = F_w(v(n)) \quad . \tag{7.6}$$

**Netzwerkadaption in den Sprachpausen**. Während die Berechnung von $x(n)$ und die anschließende Geräuschreduktion im Segmentraster erfolgen, findet die Aktualisierung der Geräuschparameter $n$ in größeren Zeitabständen statt. Diese sind durch die Zeitdauer zwischen zwei Adaptionsintervallen gegeben, die bei Isoliertworterkennung durch die jeweilige Wortlänge bestimmt wird.

Erweitert man die eindimensionale Abbildungsfunktion in Gl. (3.6) auf $n$ Ausgänge und setzt anstelle des Eingangsvektors $x$ den Vektor $v$ mit Sprach- und Geräuschkoeffizienten ein, so errechnet sich der Netzwerkausgang aus

$$F(v) = f\left(\sum_{i=0}^{h} v_{oi} \cdot z_i\right) = f\left(\sum_{i=0}^{h} v_{oi} \cdot g\left(\sum_{j=0}^{m} w_{ij} v_j\right)\right) \quad , \quad o=1,\ldots,n \quad . \qquad (7.7)$$

$z_i$ ist die Ausgangserregung des $i$-ten von insgesamt $h$ verdeckten Knoten mit den Aktivierungsfunktionen $g()$. Die untere Summationsgrenze 0 für $i$ und $j$ schließt die Offseteingänge der verdeckten und der Ausgangsknoten ein. $z_i$ ist durch

$$\begin{aligned} z_i(v) &= g\left(\sum_{j=0}^{m} w_{ij} v_j\right) \\ &= g\left(\sum_{j=0}^{q} w_{ij} x_j + \sum_{j=q+1}^{m} w_{ij} n_{j-q}\right) \end{aligned} \qquad (7.8)$$

gegeben. Berücksichtigt man, daß der letzte Term der zweiten Zeile in Gl. (7.8) für die Dauer eines Wortes konstant bleibt, kann $z_i$ im $k$-ten Wort aus

$$z_{i,k}(x,n,k) = g\left(\sum_{j=0}^{q} w_{ij} x_j + N_i(k)\right) \quad \text{mit} \qquad (7.9a)$$

$$N_i(k) = \sum_{j=q+1}^{m} w_{ij} n_{j-q}(k) \quad , \; n_{j-q}(k) \in n_k \qquad (7.9b)$$

berechnet werden, wobei $n_k$ der Pausenvektor mit den Geräuschkoeffizienten vor dem $k$-ten Wort ist. Aus Gln. (7.9a) und (7.9b) kann man erkennen, daß die an den Steuereingängen anliegende Information für die Dauer eines Wortes als konstanter additiver Beitrag $N_i(k)$ zum Zellpotential des $i$-ten verdeckten Knotens aufgefasst werden kann. Mit Gln. (7.7), (7.9a) und (7.9b) kann nun $F_w(x,n,k)$ für die Koeffizienten des $k$-ten Wortes angegeben werden:

$$F_w(x,n,k) = F_{w,n_k}(x) = f\left(\sum_{i=0}^{h} v_{oi} \cdot g\left(\sum_{j=0}^{q} w_{ij} x_j + N_i(k)\right)\right) . \qquad (7.10)$$

Die Adaption der Geräuschreduktionsabbildung wird also bei $W$ zu erkennenden Wörtern durch eine Folge $\{n_1, n_2, \ldots, n_W\}$ von Vektoren aus den Sprachpausen vorgenommen. Der $n$-te Ausgangsvektor $y(n)$ des $k$-ten Wortes berechnet sich dann mit der adaptiven Abbildung $F_{w,n_k}$ aus

$$y(n) = F_{w,n_k}(x(n)) \quad . \qquad (7.11)$$

## 7.2. Extraktion der Geräuschparameter

Zur Beschreibung des Geräuschsignals werden mit möglichst geringem Rechenaufwand extrahierbare Parameter gesucht. Voruntersuchungen (Trompf et al. 1994) mit einer kleinen Geräuschdatenbasis und einigen wenigen Parametern zeigten, daß dieser Ansatz prinzipiell funktionsfähig ist. Mit dem SNR und drei spektralen Momenten als Steuerparameter wurden Verbesserungen der Worterkennungsrate von einigen Prozent bei 0 dB SNR erreicht. Im vorliegenden Abschnitt wird die Erweiterung des Geräuschparametersatzes beschrieben.

Zur Generierung der Geräuschkoeffizienten wurde die Merkmalsextraktion (vgl. Abschnitt 4.2) wie in Bild 7.2 gezeigt erweitert. Die Berechnung der Sprachkoeffizienten und deren Ableitungen sowie die anschließende Hauptachsentransformation finden im oberen Verarbeitungszweig statt. Um eine geeignete Repräsentation des Pausensignals zu untersuchen, wurde mit konkurrierenden Vorverarbeitungsmodellen eine Vielzahl von Geräuschparametern erzeugt (unterer Verarbeitungszweig). Da die Dimensionalität des Merkmalsvektors den Aufwand für das nachfolgende Geräuschreduktionsnetz bestimmt, werden hieraus möglichst wenige Parameter aufgrund objektiver Kriterien ausgewählt (Abschnitt 7.3). Die nachgeschaltete HAT dient der Dimensionsreduktion durch Eliminierung redundanter Information. Ziel der Untersuchungen ist die Bestimmung von Art und Zahl der benötigten Parameter sowie die Evaluierung des Verfahrens mit Hilfe der erweiterten Geräuschdatenbasis (siehe Abschnitt 4.1).

**Indizierung der Wort- und Pauseninformation**. Zur Berechnung von Geräuschparametern wird die Kenntnis der Wort- bzw. Pausengrenzen im Sprachsignal

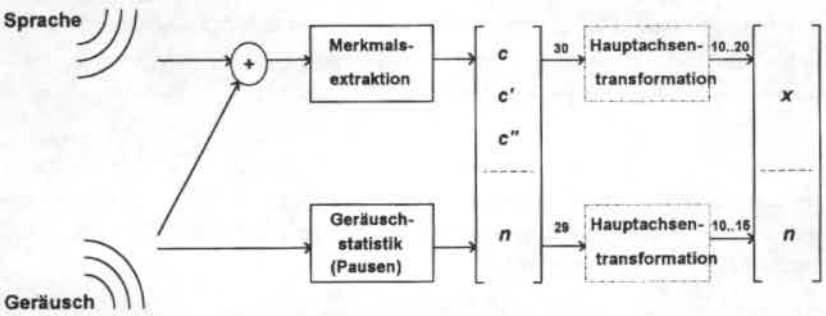



Bild 7.2: Extraktion des erweiterten Merkmalsvektors mit Geräuschparametern.

vorausgesetzt[24]. Dann können die Abtastwerte des geräuschbehafteten Sprachsignals $r(i)$ in Paare von Pausen- und Sprachabschnitte $p$ bzw. $w$ segmentiert werden, die mit dem Pausen- bzw. Wortzähler $k$, dem Segmentindex $n$ sowie der diskreten Zeitvariablen $i$ indiziert werden. Die neue Indizierung wird bei der Beschreibung der Geräuschparameterextraktion wie folgt verwendet:

$$r(i) \xrightarrow[\text{Wortgrenzen}]{} \{p(k,n,i), w(k,n,i)\} \quad . \qquad (7.12)$$

Dabei bedeuten:

| | |
|---|---|
| $k = 1,2,\ldots,L$ | Wort- bzw. Pausenindex, mit insgesamt $L$ Wörtern (Pausen). |
| $n = 1,2,\ldots,W(k)$ | im Wort, mit $W(k)$ Segmenten im $k$-ten Wort, bzw. |
| $n = 1,2,\ldots,P(k)$ | in der Pause, mit $P(k)$ Segmenten in der $k$-ten Pause. |
| $i = 1,2,\ldots,N$ | diskrete Zeitvariable, mit der Segmentlänge $N$. |

**Berechnungsintervalle**. Nach Berechnung der Parameter zur Netzwerkadaption innerhalb kurzer, quasistationärer Pausensegmente werden daraus Schätzwerte für größere Zeitabschnitte ermittelt. Um eine Adaption an lokale Signaleigenschaften zu vermeiden, erfolgt dabei entweder eine Glättung durch Tiefpaßfilterung aufeinanderfolgender Werte oder eine Mittelwertbildung über den gesamten Wort- bzw. Pausenabschnitt.

**Glättung**. Bei der Berechnung eines Schätzwertes für den Geräuschparameter $G$ im $n$-ten Segment der $k$-ten Pause werden seine Werte gemäß

$$\hat{G}_{TP}(k,n) = \beta\, G(k,n) + (1-\beta)\, \hat{G}_{TP}(k,n-1) \quad \text{mit} \qquad (7.13)$$

$$\beta \quad \text{Glättungsparameter, } \beta \in R \text{ und } \beta \in [0,1]$$

durch Tiefpaßfilterung geglättet, wobei der Wert von $\beta$ den Glättungsgrad bestimmt und von der Stationariät des Geräuschsignals abhängt. Der Schätzwert im letzten der $P(k)$ Segmente wird dann als Schätzwert $\hat{G}(k)$ für den gesamten Pausenabschnitt angenommen:

$$\hat{G}(k) = \hat{G}_{TP}(P(k)) \quad . \qquad (7.14a)$$

Bei dieser Berechnungsart werden weiter zurückliegende Werte schwächer gewichtet, was beispielsweise bei der Schätzung des Störspektrums für die Spektralsubtraktion benutzt wird, vgl. Gl. (8.5) und Reich (1985).

---

[24] Für die folgenden Betrachtungen wird ein idealer Sprachpausendetektor vorausgesetzt, vgl. hierzu Abschnitt 4.1.

**Mittelung.** Hierbei wird über die Parameterwerte aller Pausensegmente gemittelt. Der Schätzwert $\hat{G}$ für $G$ in der $k$-ten Pause berechnet sich daher direkt aus

$$\hat{G}(k) = \overline{G}(k) = \frac{1}{P(k)} \sum_{n=1}^{P(k)} G(k,n) \qquad (7.14b)$$

Bei dieser Berechnungsart tragen die Werte in den einzelnen Pausensegmenten gleichberechtigt zum Schätzwert bei.

**Verarbeitungsbereich.** In den Experimenten zur Netzwerkadaption wurden insgesamt 29 Geräuschparameter untersucht, die entweder dem Zeit-, dem Spektral- oder dem Cepstralbereich zugeordnet werden können. Wie bei der Merkmalsextraktion aus dem Sprachsignal ist die Auswahl geeigneter Vorverarbeitungsmodelle nicht in erster Linie physikalisch motiviert, sondern orientiert sich an ihrer Effizienz für die Nachadaptionsaufgabe sowie an ihrer Eignung für eine Echtzeitimplementierung. Um eine möglichst zuverlässige Schätzung der Parameter zu erreichen, erfolgt ihre Berechnung hier durch Mittelung über alle Segmente der jeweiligen Pause nach Gleichung (7.14b). Eine Übersicht der im folgenden beschriebenen Parameter zeigt Tabelle 7.1. Die Implementierung wurde von Eckhardt (1994) vorgenommen [25].

Tabelle 7.1: Übersicht der Geräuschparameter.

| Verarb. im | Bedeutung | Zahl |
|---|---|---|
| Zeitbereich | mittlere Energie des Pausensignals | 1 |
| | log. mittlere Energie des Pausensignals | 1 |
| | lokaler SNR, wortbezogen | 2 |
| | lokaler SNR, segmentbezogen | 1 |
| Spektralbereich | Schwerpunkt des Spektrums, gemittelt | 2 |
| | oberer und unterer spektraler Moment, gemittelt | 2 |
| Cepstralbereich | lpc-cepstrum Koeffizienten, gemittelt | 10 |
| | Varianz der lpc-cepstrum Koeffizienten | 10 |
| | Summe | 29 |

**Parameter im Zeitbereich**. Hierzu gehören die lineare und die logarithmierte mittlere Energie des Pausensignals sowie verschiedene Schätzwerte für das Signal-zu-Rausch-Verhältnis. Sie werden folgendermaßen berechnet:

---

[25] Einige der unten angegebenen Rechenvorschriften gehen auf Vorschläge von Eckhardt (1994) zurück. Dies betrifft die in Gln. (7.19), (7.20) und (7.25) aufgeführten Parameter.

1. Die auf einen Abtastwert normierte **mittlere Energie** $\overline{E}_P$ des Pausensignals $p$ vor dem $k$-ten Wort:

$$\overline{E}_P(k) = \frac{1}{P(k)N} \sum_{n=1}^{P(k)} \sum_{i=1}^{N} p^2(k,n,i) \quad . \quad (7.15)$$

2. Die **logarithmierte mittlere Pausenenergie** $\overline{E}_{lgP}$ vor dem $k$-ten Wort:

$$\overline{E}_{lgP}(k) = lg\, \overline{E}_P(k) \quad . \quad (7.16)$$

3. Der Schätzwert $\hat{S}NR$ für das **lokale wortbezogene Signal-zu-Rausch-Verhältnis** in dem auf die $k$-te Pause folgenden Wort:

$$\hat{S}NR(k) = 10\, lg \left( \frac{\overline{E}_W(k) - \overline{E}_P(k)}{\overline{E}_P(k)} \right) \quad , \quad (7.17)$$

mit $\overline{E}_P$ nach Gl. (7.15) und der mittleren Energie $\overline{E}_W$ des geräuschbehafteten Sprachsignals $w$ im $k$-ten Wort:

$$\overline{E}_W(k) = \frac{1}{W(k)N} \sum_{n=1}^{W(k)} \sum_{i=1}^{N} w^2(k,n,i) \quad . \quad (7.18)$$

4. Der Schätzwert für das **lokale, segmentbezogene Signal-zu-Rausch-Verhältnis** im $n$-ten Segment des $k$-ten Wortes:

$$\hat{S}NR_{Seg}(k,n) = 10\, lg \left( \frac{\frac{1}{N} \sum_{i=1}^{N} w^2(k,n,i) - \overline{E}_P(k)}{\overline{E}_P(k)} \right) \quad . \quad (7.19)$$

$\hat{S}NR_{Seg}$ bildet wegen seiner segmentweisen Neuberechnung eine Ausnahme.

5. Ein **modifiziertes**, durch lineare Mittelung der logarithmierten Energiewerte berechnetes Maß $\rho$ für das **Signal-zu-Rausch-Verhältnis** im $k$-ten Wort:

$$\rho(k) = \frac{1}{W(k)} \sum_{n=1}^{W(k)} lg \left( \frac{1}{N} \sum_{i=1}^{N} w^2(k,n,i) \right) - \frac{1}{P(k)} \sum_{n=1}^{P(k)} lg \left( \frac{1}{N} \sum_{i=1}^{N} p^2(k,n,i) \right) \quad . \quad (7.20)$$

**Parameter im Spektralbereich**. Zur groben Beschreibung des Pausenspektrums mit wenigen Parametern kann die Energieverteilung auf Basis der spektralen Momente aufeinanderfolgender Kurzzeitspektren nach Schotola (1984) berechnet werden. Der einzige Unterschied der vorliegenden Implementierung liegt in der Verwendung von $M$ Spektrallinien anstatt der von Schotola verwendeten 24 Lautheitskanäle. Der Schwerpunkt $m_S$ des $n$-ten komplexen Kurzeitspektrums $S(n)$ mit $M$ Spektrallinien ist gegeben durch:

$$m_S(n) = \frac{\sum_{j=1}^{M} j\left|S(n,j)\right|^2}{\sum_{j=1}^{M} \left|S(n,j)\right|^2} \qquad (7.21)$$

Die unteren bzw. oberen spektralen Momente $m_u$ und $m_o$ werden dann aus den spektralen Energieanteilen unterhalb bzw. oberhalb von $m_S$ aus

$$m_u(n) = \frac{\sum_{j=1}^{\left[m_S(n)+0{,}5\right]} j\left|S(n,j)\right|^2}{\sum_{j=1}^{\left[m_S(n)+0{,}5\right]} \left|S(n,j)\right|^2} \quad \text{und} \quad m_o(n) = \frac{\sum_{j=\left[m_S(n)+1{,}5\right]}^{M} j\left|S(n,j)\right|^2}{\sum_{j=\left[m_S(n)+1{,}5\right]}^{M} \left|S(n,j)\right|^2} \qquad (7.22)$$

berechnet [26]. Aus den in Gleichungen (7.21) und (7.22) berechneten, segmentbezogenen Momenten werden in der $k$-ten Pause folgende Parameter generiert:

6. Der **Mittelwert des Schwerpunkts** $\overline{m}_S$ aus $P(k)$ Segmenten der $k$-ten Pause:

$$\overline{m}_S(k) = \frac{1}{P(k)} \sum_{n=1}^{P(k)} m_S(k,n) \qquad (7.23)$$

7. Die **Mittelwerte** $\overline{m}_u$ und $\overline{m}_o$ der **oberen** bzw. **unteren spektralen Momente**:

$$\overline{m}_u(k) = \frac{1}{P(k)} \sum_{n=1}^{P(k)} m_u(k,n) \quad \text{und} \quad \overline{m}_o(k) = \frac{1}{P(k)} \sum_{n=1}^{P(k)} m_o(k,n) \qquad (7.24)$$

---

[26] Die mit [] bezeichnete Abschneidefunktion dient zur Konvertierung der reellen Werte von $m_S$ zu ganzzahligen Summationsgrenzen. Der Funktionswert $[x] \in N$ einer reellen Zahl $x \in R$ wird durch Weglassen ihrer Nachkommastellen (Abschneiden) berechnet.

8. Bei einer von den Gleichungen (7.22) und (7.23) abweichenden Berechnungsweise des Schwerpunkts wird zunächst eine Mittelung über die Pausensegmente und erst im nächsten Schritt die Berechnung des **modifizierten Schwerpunkts** durchgeführt:

$$\overline{m}_{\bar{S}}(k) = \frac{\sum_{j=1}^{M} j\,\overline{|S(k,j)|^2}}{\sum_{j=1}^{M} \overline{|S(k,j)|^2}} \quad \text{mit} \quad \overline{|S(k,j)|^2} = \frac{1}{P(k)} \sum_{n=1}^{P(k)} |S(k,n,j)|^2 \,. \quad (7.25)$$

Analog zu Gl. (7.25) könnten ebenfalls die unteren und oberen Pausenmomente der gemittelten Kurzzeitspektren berechnet werden. Aus Aufwandsgründen wurde jedoch auf ihre Untersuchung verzichtet.

**Parameter im Cepstralbereich**. Die Extraktion der lpc-cepstrum Koeffizienten erfolgt wie in Abschnitt 4.2.1 beschrieben. Folgende Parameter wurden daraus zur Beschreibung des Geräuschsignals in der $k$-ten Pause abgeleitet:

9. Der **Mittelwert** $\bar{c}_q$ des $q$-ten **lpc-cepstrum Koeffizienten** über $P(k)$ Segmente der $k$-ten Pause:

$$\bar{c}_q(k) = \frac{1}{P(k)} \sum_{n=1}^{P(k)} c_q(k,n) \quad . \quad (7.26)$$

10. Die **Standardabweichung** $\sigma_q$ des $q$-ten **lpc-cepstrum Koeffizienten** $\bar{c}_q$ der $k$-ten Pause:

$$\sigma_q(k) = \frac{1}{P(k)} \sqrt{\sum_{n=1}^{P(k)} \left(c_q(k,n) - \bar{c}_q(k)\right)^2} \quad . \quad (7.27)$$

### 7.3. Auswahl geeigneter Parameter

Zur Aufwandsbeschränkung ist eine Begrenzung der Parameterzahl durch Auswahl der wichtigsten Koeffizienten aufgrund eines objektiven Gütemaßes notwendig. Nachfolgend werden zwei aufgabenbezogene Auswahlkriterien untersucht. In den Merkmalsvektoren des Folgewortes werden lediglich die ausgewählten Geräuschparameter gespeichert und vom Netzwerk weiterverarbeitet.

**Regressionskoeffizient**. Ein potentieller Beitrag des $i$-ten Geräuschparameters $G_i$ zur Verminderung des Abbildungsfehlers im Trainingsdatensatz mit $L$ Vektorpaaren ist durch die Kovarianz $\sigma_{ie}$ mit dem quadratischen Fehler $e^T e$ zwischen geräuschfreien und -behafteten Vektoren im Trainingsdatensatz nach Gl. (3.7) zu erwarten. $\sigma_{ie}$ berechnet sich aus

$$\sigma_{ie} = \frac{1}{L-1}\sum_{l=1}^{L}\left(G_i^l - \overline{G}_i\right)\left(e^{T^l} e^l - MSE_T\right) \quad (7.28)$$

Normierung von $\sigma_{ie}$ auf die Varianz $\sigma_i^2$ des Parameters erleichtert die Vergleichbarkeit unterschiedlicher Parameter $G_i$ und führt auf den Regressionskoeffizienten $R_i$ (z.B. Lechner und Lohl 1990), der durch

$$R_i = \frac{\sigma_{ie}}{\sigma_i^2} \quad (7.29)$$

gegeben ist. Experimentelle Ergebnisse zur Parameterauswahl mit Hilfe der Werte des Regressionskoeffizienten sind in Abschnitt 7.4.1 beschrieben.

**Varianz der Koeffizienten aus der Hauptachsentransformation**. Die Eigenwerte der Kovarianzmatrix der Merkmalsvektoren (vgl. Abschnitt 4.2.3) können als Informationsinhalt der neuen Koeffizienten aufgefaßt werden. Sortiert man sie nach absteigenden Eigenwerten nach Gl. (4.30), liefern die ersten Koeffizienten auch den größten Beitrag zur Gesamtinformation. Diese Anwendung der HAT zur Dimensionsreduktion der Geräuschparameter wird in Abschnitt 7.4.2 beschrieben.

### 7.4. Experimentelle Ergebnisse

Die Erhöhung der Robustheit durch Erweiterung des Trainingsmaterials wurde in Abschnitt 5.4 diskutiert. Dabei ergaben Ergebnisse mit Geräuschpooltraining dann nahezu keinen Unterschied zu denjenigen bei Training mit Einzelgeräuschen, wenn das Testgeräusch im Trainingspool enthalten war. Verbesserungswürdig ist daher der praxisnahe Fall der Adaption des Netzwerks an ein fremdes, nichttrainiertes Geräusch nach initialem Vortraining. Dies wird im vorliegenden Abschnitt untersucht; dabei dienen die Ergebnisse mit Pooltraining als Referenzwerte, an denen der Adaptionserfolg gemessen wird.

**Trainingsdaten.** Wie bei den Geräuschpoolexperimenten wurde auch hier die erweiterte Geräuschdatenbasis mit fünf Aufnahmen im Trainings- und zwei im Testpool eingesetzt. Die einzige Modifikation bestand in der Erweiterung des SNR-Wertebereichs im Trainingsmaterial unter Beibehaltung der Gesamtdatenmenge. In den Trainingspool wurden daher störbehaftete Sprachaufnahmen mit je -5, +5 und +15 dB SNR sowie zusätzlich die unverrauschten Originalaufnahmen aufgenommen. Aufgrund der Netzwerkadaption war dies ohne Einbußen an Erkennungsleistung bei ungestörten Testdaten möglich. Die zusätzlich generierten Parameter wurden in den Merkmalsvektoren gespeichert.

Tabelle 7.2: Topologie bei 5 Kontextvektoren mit 10 Koeffizienten und unterschiedlicher Geräuschparameterzahl.

| Pausenadaption | Topologie |
|---|---|
| ohne | *50-20-10* |
| mit 5 Parametern | *75-30-10* |
| mit 10 Parametern | *100-30-10* |
| mit 15 Parametern | *125-30-10* |

**Netzwerkparameter**. Die Trainings- und Netzwerkparameter wurden wie bei den Geräuschpoolexperimenten gewählt, vgl. Abschnitt 5.4.2. Für die Netzwerkadaption wurde die Eingangsschicht bei $r$ Geräuschparametern und fünf Kontextvektoren um $5r$ Knoten erweitert. Die Erweiterung der Zwischenschicht von 20 auf 30 Knoten erfolgte aufgrund von Erfahrungswerten, die für komplexere Lernaufgaben eine höhere Zahl verdeckter Knoten vorsehen. Tabelle 7.2 zeigt die verwendeten Topologieen in Abhängigkeit von der Parameterzahl, wobei das in Kapitel 5 entwickelte Basisnetzwerk mit *50-20-10* Topologie und ohne Pausenadaption als Referenz dient.

### 7.4.1. Adaption mit Geräuschkoeffizienten

Tabelle 7.3: Nach dem Regressionskoeffizienten $R$ sortierte Geräuschparameter.

| | Para. | $R$ |
|---|---|---|
| 1 | $\overline{c}_2$ | 0,1125 |
| 2 | $\overline{c}_7$ | 0,1051 |
| 3 | $\overline{m}_o$ | 0,1035 |
| 4 | $\overline{c}_{10}$ | 0,1021 |
| 5 | $\overline{m}_{\bar{s}}$ | 0,0957 |
| 6 | $\overline{m}_s$ | 0,0949 |
| 7 | $\overline{c}_1$ | 0,0904 |
| 8 | $\overline{E}_P$ | 0,0903 |
| 9 | $\overline{c}_3$ | 0,0893 |
| 10 | $\overline{c}_8$ | 0,0890 |
| 11 | $\overline{c}_9$ | 0,0880 |
| 12 | $\overline{c}_5$ | 0,0860 |
| 13 | $\overline{E}_{lgP}$ | 0,0821 |
| 14 | $\hat{SNR}$ | 0,0788 |
| 15 | $\rho$ | 0,0781 |
| 16 | $\sigma_7$ | 0,0772 |
| 17 | $\overline{m}_u$ | 0,0716 |
| 18 | $\overline{c}_6$ | 0,0714 |
| 19 | $\sigma_1$ | 0,0538 |
| 20 | $\sigma_3$ | 0,0525 |
| 21 | $\sigma_5$ | 0,0523 |
| 22 | $\overline{c}_4$ | 0,0468 |
| 23 | $\hat{SNR}_{\bar{s}_{eq}}$ | 0,0445 |
| 24 | $\sigma_2$ | 0,0437 |
| 25 | $\sigma_6$ | 0,0432 |
| 26 | $\sigma_4$ | 0,0356 |
| 27 | $\sigma_{10}$ | 0,0330 |
| 28 | $\sigma_8$ | 0,0303 |
| 29 | $\sigma_9$ | 0,0264 |

Aufgrund der Ergebnisse der Voruntersuchungen wurde der Geräuschparametersatz von 4 auf 29 Parameter erweitert, vgl. Gln. (7.15) bis (7.27) sowie Tabelle 7.1. Ihre Sortierung erfolgte aufgrund der Werte des Regressionskoeffizienten $R$ nach Gl. (7.28), siehe Tabelle 7.3. Die Werte für $R$ wurden dabei aus der gesamten Sprach- und Geräuschdatenbasis mit SNR-Werten von -5 dB bis 20 dB in 5 dB-Schritten sowie den Originalaufnahmen berechnet. Wie aus den ersten 10 Zeilen zu sehen ist, sind darin lpc-cepstrum- und momentbasierte Parameter sowie die Pausenenergie enthalten.

Die Parameterzahl $r$ wurde auf Basis dieser Reihenfolge in zwei Versuchsreihen auf 10 bzw. 15 beschränkt. Bilder 7.3a und 7.3b zeigen die Ergebnisse bei geräuschabhängigen und geräuschübergreifenden Tests mit Pooltraining. Durch die Netzwerkadaption (*MLP+10* bzw. *MLP+15*) werden bei geräuschabhängigen Experimenten (Bild 7.3a) die Ergebnisse bei Training mit Einzelgeräuschen (*abhängig*) fast erreicht, während bei geräuschübergreifenden Experimenten (Bild 7.3b) trotz Verbesserungen von bis zu 7 % gegenüber der Situation ohne Adaption und mit Pooltraining (*pool*) noch deutliche Unterschiede zur geräuschabhängigen Situation und Training mit Einzelgeräuschen (*abhängig*) erkennbar sind. Bei nichttrainierten Testgeräuschen werden durch Erhöhung der Parameterzahl von 10 auf 15 Verbesserungen von einigen Prozent bei niedrigen SNR-Werten erreicht (Bild 7.3b), während dies auf die geräuschabhängigen Ergebnisse nahezu keinen Einfluß hat. Aus diesen Ergebnissen kann geschlossen werden, daß bei Optimierung der Geräuschsignalrepräsentation die Adaption an nicht vortrainierte Signale noch verbessert werden kann.

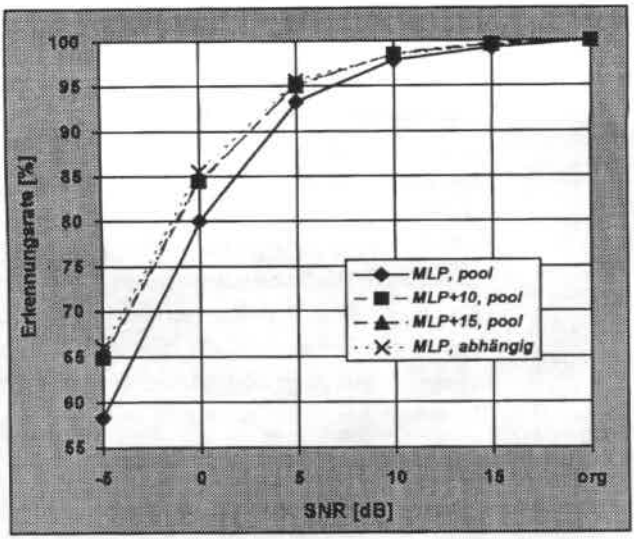



Bild 7.3a: Worterkennungsraten nach Geräuschreduktion mit Pooltraining und Geräuschparameteradaption, **geräuschabhängige** Tests.

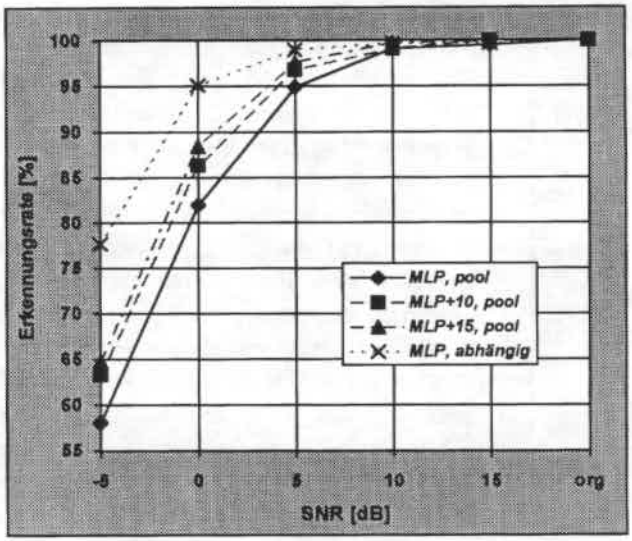



Bild 7.3b: Worterkennungsraten nach Geräuschreduktion mit Pooltraining und Geräuschparameteradaption, **geräuschübergreifende** Tests.

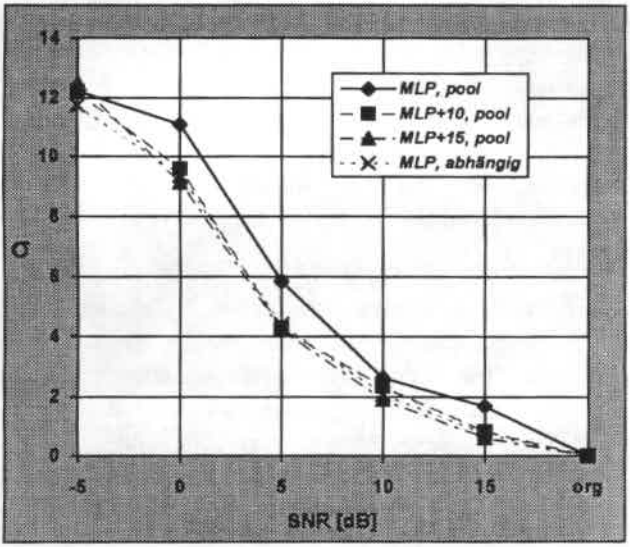



Bild 7.4a: Standardabweichung σ der Worterkennungsraten nach Bild 7.3a.

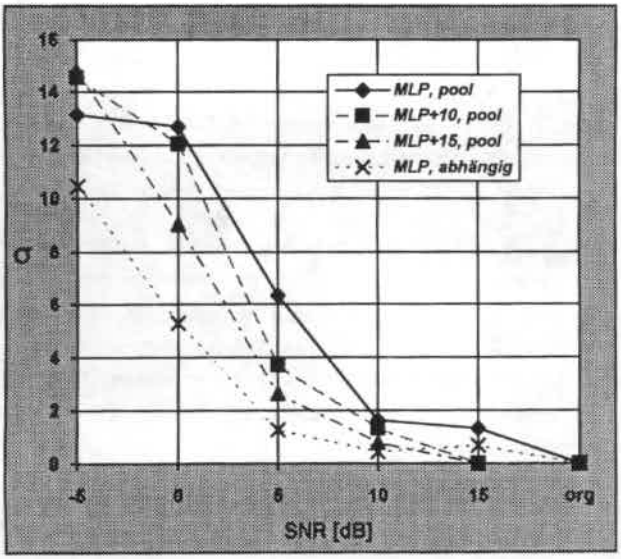



Bild 7.4b: Standardabweichung σ der Worterkennungsraten nach Bild 7.3b.

Die Streuung der Erkennungsraten ist in den Bildern 7.4a und 7.4b zu sehen. In den geräuschabhängigen Ergebnissen mit Pooltraining und Adaption ist $\sigma$ nur wenig höher als bei denen mit geräuschabhängigem Training (Bild 7.4a). Die geräuschübergreifenden Ergebnisse mit Adaption (Bild 7.3b) streuen für kleine SNR-Werte erheblich stärker als im geräuschabhängigen Fall. Insgesamt sind bei SNR-Werten > -5 dB die $\sigma$-Werte für 15 Parameter niedriger als bei 10. Bei SNR-Werten > 15 dB gilt $\sigma$=0, und die geräuschabhängigen Werte werden damit sogar noch unterschritten.

**Korrelation der Geräuschparameter**. Um die im Parametersatz enthaltenen Redundanzen festzustellen, wurde die Korrelationsmatrix für die ersten 10 bzw. 15 Geräuschparameter berechnet. Der Korrelationskoeffizient $l_{ij}$ zwischen dem $i$-ten und dem $j$-ten Parameter $G_i$ bzw. $G_j$ ist ein Maß für die lineare Abhängigkeit zweier Meßgrößen und wird aus (z.B. Lechner und Lohl 1990)

$$l_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \tag{7.30}$$

berechnet, wobei die Kovarianz zweier Zufallsvariablen $\sigma_{ij}$ in Gl. (7.28) definiert wurde und $\sigma_i$ bzw. $\sigma_j$ ihre Standardabweichungen sind. Die Werte der Korrelationskoeffizienten zwischen den ersten 15 Geräuschparametern nach Tabelle 7.3 können der Tabelle 7.4 entnommen werden. Da die Korrelationsmatrix symmetrisch ist, wurde nur die obere Hälfte dargestellt. Die Beträge der Korrelationswerte $|l_{ij}|$ bewegen sich zwischen 0 (unkorrelierte Parameter) und 1 (maximale Korrelation); Werte von $|l_{ij}|>0{,}5$ sind durch Fettdruck hervorgehoben.

Wie insbesondere aus den Korrelationswerten zwischen den Momenten des Geräuschspektrums nach Gln. (7.23) bis (7.25) untereinander und mit den ersten beiden lpc-cepstrum-Koeffizienten ersichtlich ist, enthält der Parametersatz redundante Information. Auffällig ist außerdem die starke Korrelation zwischen dem modifizierten Signal-zu-Rausch-Verhältnis $\rho$ und den beiden Energieparametern nach Gln. (7.15) und (7.16). Der Grund dafür liegt darin, daß die Pausenenergie in die Berechnung von $\rho$ einfließt, vgl. Gl. (7.20).

Die starken Korrelationen im Geräuschparametersatz sind nicht überraschend; sie sind in den Berechnungsvorschriften für einige der Parameter begründet. Um Redundanzfreiheit zu erhalten, wird zur anschließenden Optimierung des Parametersatzes eine Hauptachsentransformation durchgeführt.

Tabelle 7.4: Korrelationsmatrix der ersten 10 bzw. 15 Geräuschparameter.

| | $\bar{c}_2$ | $\bar{c}_7$ | $\bar{m}_o$ | $\bar{c}_{10}$ | $\bar{m}_{\bar{S}}$ | $\bar{m}_S$ | $\bar{c}_1$ | $\bar{E}_P$ | $\bar{c}_3$ | $\bar{c}_8$ | $\bar{c}_9$ | $\bar{c}_5$ | $\bar{E}_{lgP}$ | $\hat{SNR}$ | $\rho$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\bar{c}_2$ | **1,000** | -0,084 | **0,514** | -0,485 | 0,219 | 0,223 | -0,116 | -0,040 | -0,043 | -0,230 | **0,606** | 0,354 | 0,244 | 0,097 | -0,224 |
| $\bar{c}_7$ | | **1,000** | 0,071 | 0,158 | -0,035 | 0,055 | -0,054 | 0,229 | -0,474 | **0,595** | -0,307 | 0,171 | 0,291 | -0,137 | -0,251 |
| $\bar{m}_o$ | | | **1,000** | -0,402 | **0,771** | **0,928** | **-0,827** | -0,010 | -0,143 | -0,032 | **0,541** | **0,639** | -0,026 | -0,010 | 0,020 |
| $\bar{c}_{10}$ | | | | **1,000** | -0,303 | -0,312 | 0,323 | 0,089 | -0,153 | 0,140 | -0,375 | -0,104 | 0,004 | -0,008 | -0,003 |
| $\bar{m}_{\bar{S}}$ | | | | | **1,000** | **0,862** | **-0,864** | 0,006 | 0,133 | -0,171 | 0,243 | 0,417 | -0,115 | -0,010 | 0,112 |
| $\bar{m}_S$ | | | | | | **1,000** | **-0,942** | 0,011 | -0,012 | -0,038 | 0,361 | **0,539** | -0,088 | 0,004 | 0,074 |
| $\bar{c}_1$ | | | | | | | **1,000** | -0,039 | -0,073 | 0,148 | -0,213 | -0,390 | 0,116 | -0,013 | -0,104 |
| $\bar{E}_P$ | | | | | | | | **1,000** | -0,177 | 0,101 | -0,035 | -0,025 | **0,670** | -0,405 | **-0,613** |
| $\bar{c}_3$ | | | | | | | | | **1,000** | -0,457 | -0,206 | -0,281 | -0,134 | 0,069 | 0,119 |
| $\bar{c}_8$ | | | | | | | | | | **1,000** | -0,141 | 0,268 | 0,076 | -0,037 | -0,058 |
| $\bar{c}_9$ | | | | | | | | | | | **1,000** | 0,410 | 0,091 | -0,054 | -0,103 |
| $\bar{c}_5$ | | | | | | | | | | | | **1,000** | -0,037 | 0,019 | 0,062 |
| $\bar{E}_{lgP}$ | | | | | | | | | | | | | **1,000** | **-0,544** | **-0,961** |
| $\hat{SNR}$ | | | | | | | | | | | | | | **1,000** | **0,572** |
| $\rho$ | | | | | | | | | | | | | | | **1,000** |

### 7.4.2. Adaption mit den Koeffizienten der Hauptachsentransformation

**Bestimmung der Koeffizientenzahl.** Als Ergebnis der Hauptachsentransformation erhält man neue, unkorrelierte Koeffizienten, die nach ihren Varianzwerten absteigend sortiert sind (siehe Abschnitt 4.2.3). Da die Varianz als Informationsinhalt eines Koeffizienten interpretiert werden kann (vgl. Paliwal 1992), geht bei der anschließenden Dimensionsreduktion auf die $r$ "besten" Koeffizienten Information verloren. Die erhalten gebliebene Information kann aus dem Verhältnis der $r$ akkumulierten Varianzen zur Gesamtvarianz berechnet werden.

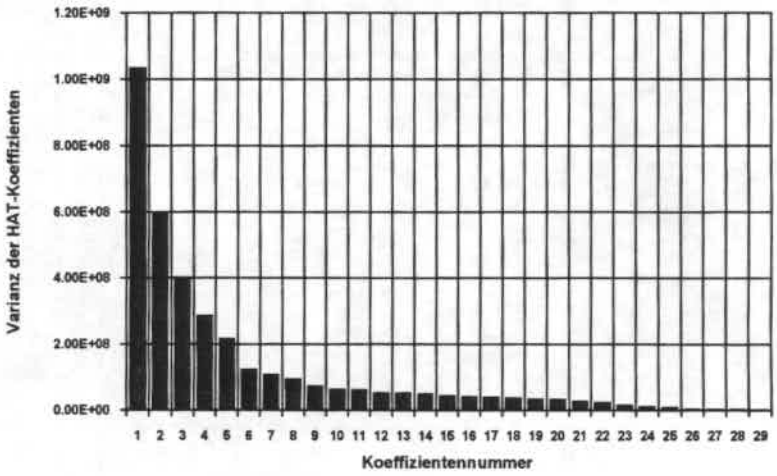



Bild 7.5: Varianzwerte der 29 HAT-Koeffizienten.

Zur Bestimmung der Varianzanteile wurden die Geräuschparameter aus der gesamten Trainings- und Testdatenbasis berechnet und im Anschluß einer Hauptachsentransformation unterzogen. Bild 7.5 zeigt die Varianzwerte der sortierten HAT-Koeffizienten nach Gln. (4.28) und (4.30), und Tabelle 7.5 zeigt die Abhängigkeit zwischen der Koeffizientenzahl und ihrem Anteil an der Gesamtvarianz. Daraus wird deutlich, daß für $r$=10 ($r$=15) 85 % (92,3 %) der Gesamtinformation erhalten bleibt. Zur Netzwerkadaption scheinen daher 10-15 Koeffizienten ausreichend.

Tabelle 7.5: Anteile der akkumulierten Varianzen an der Gesamtvarianz für unterschiedliche Koeffizientenzahlen $n$.

| $r$ | 3 | 5 | 10 | 15 | 29 |
|---|---|---|---|---|---|
| % | 57,7 | 72,0 | 85,0 | 92,3 | 100 |

Aus Aufwandsgründen wird zunächst mit wenigen Experimenten der Einfluß der Koeffizientenzahl auf die Adaptionsergebnisse bestimmt. Ausgehend von den Erfahrungen mit Sprachkoeffizienten (vgl. Abschnitt 5.5.3) wurden Untersuchungen mit 5, 10 und 15 HAT-Koeffizienten zur Beschreibung des Pausensignals durchgeführt. Das Netzwerktraining erfolgte mit den Geräuschaufnahmen des Trainingspools (Pooltraining, vgl. Abschnitt 5.4.2). Alle im folgenden beschriebenen Ergebnisse sind über 10 Sprecher gemittelt.

Tabelle 7.6: Erkennungsraten [%] mit unterschiedlicher Zahl von HAT-Koeffizienten zur Netzwerkadaption.

| | -5 dB | 0 dB | +5 dB |
|---|---|---|---|
| o. GR | 50,6 | 75,0 | 92,1 |
| m. GR | 58,1 | 82,0 | 94,9 |
| 5 Koeff. | 64,2 | 87,8 | 97,6 |
| 10 Koeff. | 68,7 | 88,5 | 97,9 |
| 15 Koeff. | 70,0 | 89,2 | 98,0 |

Tabelle 7.6 zeigt die Erkennungsraten mit nichttrainierten Geräuschsignalen des Testpools bei -5, 0 und +5 dB SNR ohne (o. GR) und mit (m. GR) einmaladaptiver Geräuschreduktion sowie mit Pausenadaption mit den Koeffizienten aus der HAT (3. bis 5. Zeile). Wie aus dem Vergleich der Zeilen 2 und 3 zu sehen ist, führt die Pausenadaption mit 5 Koeffizienten zu einigen Prozent Gewinn im gesamten untersuchten SNR-Wertebereich. Durch schrittweise Erhöhung der Koeffizientenzahl über 10 auf 15 erhält man eine weitere Verbesserung der Erkennungsraten (Zeilen 4 und 5). Diese fällt für -5 dB SNR deutlich (5,8 % bei Erhöhung von 5 auf 15 Koeffizienten) und für +5 dB gering (0,4 %) aus, wobei nur ein geringer Unterschied zwischen den Ergebnissen für 10 bzw. 15 Koeffizienten besteht. Daher werden in den weiteren Experimenten 10 HAT-Koeffizienten für die Pausenadaption verwendet, was einen guten Kompromiß zwischen Aufwand und Adaptionsgewinn darstellt.

Tabelle 7.7: MSE im Trainings- (Tra) und Verifikationsdatensatz (Ver) sowie Zahl der Trainingsiterationen bei Pausenadaption des Netzwerks.

| | MSE | | |
|---|---|---|---|
| | Tra | Ver | Iter. |
| o. GR | 0,873 | 0,888 | - |
| m. GR | 0,547 | 0,597 | 55 |
| 15 GP | 0,487 | 0,554 | 31 |
| 10 HAT | 0,482 | 0,544 | 35 |

In Tabelle 7.7 ist ein Vergleich der Iterationszahlen (letzte Spalte) sowie der MSE-Werte nach Abschluß des Pooltrainings mit und ohne Pausenadaption zu sehen. Ein Vergleich der Ergebnisse mit und ohne Geräuschreduktion zeigt eine Verringerung der MSE-Werte im Trainingsdatensatz (Verifikationsdatensatz) um 37,3 % (32,8 %) durch die einmaladaptive Geräuschreduktion (Zeilen 1 und 2). Eine weitere Verringerung um 11 % (7,2 %) erhält man durch Pausenadaption mit 15 Geräusch-

parametern (15 GP, Zeile 3). Trotz Reduktion der Parameterzahl nach der Hauptachsentransformation erhält man mit 10 HAT-Koeffizienten geringfügig niedrigere MSE-Werte (4. Zeile) als in Zeile 3. Durch den Einfluß der Pausenadaption verringert sich die Zahl der Trainingsiterationen in beiden Experimenten mit Steuerparametern um 37 bzw. 43 %.

**Vergleich der Koeffizientensätze zur Netzwerkadaption.** Bilder 7.6a und 7.6b zeigen die **Worterkennungsraten** bei Geräuschreduktion mit Pausenadaption und Pooltraining (*pool*) auf Basis von 15 Geräuschparametern (*MLP+15*) bzw. 10 HAT-Koeffizienten (*MLP+10HAT*). Bei geräuschabhängigen Tests (Bild 7.6a, Stichprobe des Testgeräuschs ist im Trainingspool enthalten) sind die Ergebnisse mit denjenigen bei Training mit Einzelgeräuschen (*abhängig*) und ohne Pausenadaption nahezu identisch. Auch gegenüber den Ergebnissen mit 15 ursprünglichen Geräuschparametern sind keine signifikanten Unterschiede festzustellen. Bei geräuschübergreifenden Tests (Bild 7.6b) erhält man bei SNR-Werten $\geq 0$ dB für die HAT-Koeffizienten ähnliche Adaptionsgewinne wie mit den 15 Geräuschparametern, während man bei sehr niedrigen SNR-Werten (-5 dB) mit 10 HAT-Koeffizienten einige Prozent gewinnt. Insgesamt kann festgestellt werden, daß durch die HAT mindestens vergleichbare Ergebnisse bei geringerer Koeffizientenzahl erreicht werden, was den Aufwand für die Geräuschreduktion verringert.

Ein Vergleich der **Standardabweichungen** $\sigma$ zeigt unterschiedliche Werte für geräuschabhängige (Bild 7.7a) und geräuschübergreifende (Bild 7.7b) Experimente: im geräuschabhängigen Fall besteht nahezu kein Unterschied zwischen der einmaladaptiven Geräuschreduktion mit Einzelgeräuschtraining einerseits und beiden Experimenten mit Pausenadaption nach Pooltraining andererseits. Im geräuschübergreifenden Fall liegen die Streuungsverläufe beider Experimente mit Pausenadaption und Pooltraining für SNR-Werte $\leq 10$ dB über denen mit Einzelgeräuschtraining; das geräuschabhängige Vortraining (*MLP, abhängig*) führt zu geringeren $\sigma$-Werten als die nachträgliche Adaption an nichttrainierte Testgeräusche. Bild 7.7b zeigt, daß hier die mit HAT-Koeffizienten erzielten Ergebnisse für niedrigere SNR-Werte (-5 dB) eine kleinere Streuung aufweisen als diejenigen, die mit den ursprünglichen Geräuschparametern erzielt wurden. Außerdem liegen die Streuungen der Adaptionsergebnisse mit HAT-Koeffizienten im gesamten SNR-Wertebereich unter denjenigen mit initialem Pooltrainingtraining ohne spätere Adaption.

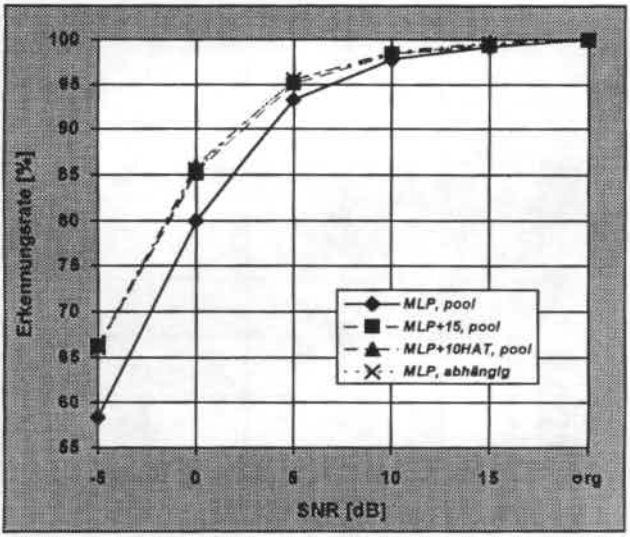



Bild 7.6a: Worterkennungsraten nach Geräuschreduktion mit Pooltraining und HAT-basierter Pausenadaption, **geräuschabhängige** Tests.

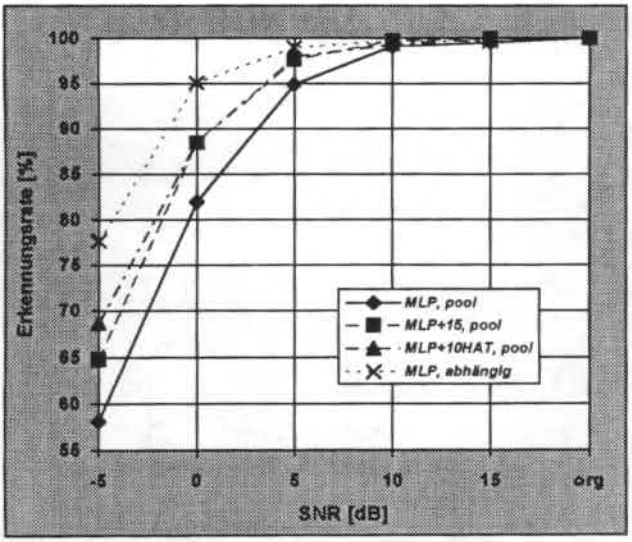



Bild 7.6b: Worterkennungsraten nach Geräuschreduktion mit Pooltraining und HAT-basierter Pausenadaption, **geräuschübergreifende** Tests.

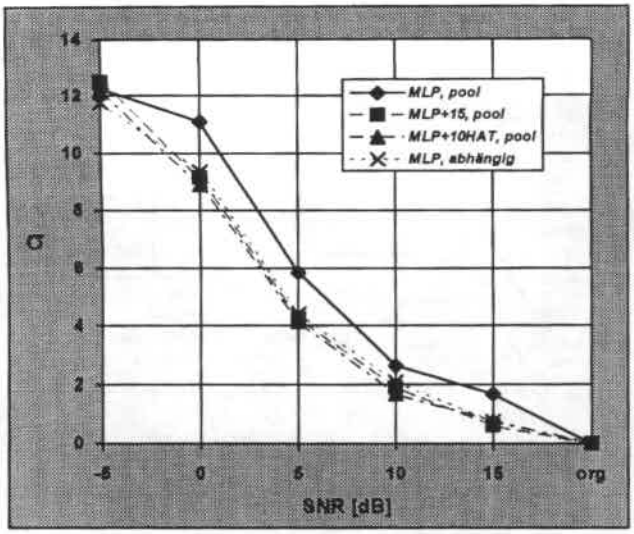



Bild 7.7a: Standardabweichung σ der Worterkennungsraten nach Bild 7.6a.

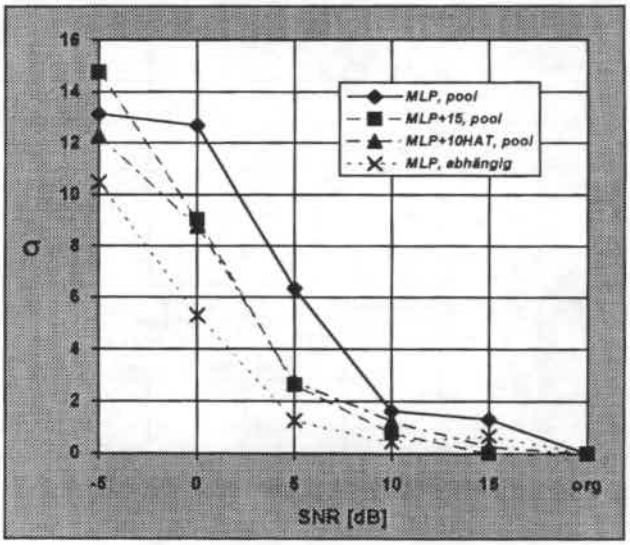



Bild 7.7b: Standardabweichung σ der Worterkennungsraten nach Bild 7.6b.

## 7.5. Zusammenfassung der Netzwerkadaption mit Geräuschparametern

Beim geräuschparameterbasierten Ansatz zur Netzwerkadaption wird die Abbildungsfunktion des Netzwerks über zusätzliche Knoten der erweiterten Eingangsschicht an das Geräuschsignal in der davorliegenden Sprachpause adaptiert. Die Geräuschparameter stammen aus dem Zeit-, Spektral- und Cepstralbereich und repräsentieren die Eigenschaften des Pausensignals mit möglichst wenig Parametern. Die im Parametersatz enthaltene redundante Information kann durch eine anschließende Hauptachsentransformation eliminiert werden. Seine Berechnung erfolgt in den Sprachpausen durch Mittelung der segmentbasierten Werte.

Die Auswahl geeigneter Geräuschparameter erfolgt auf Basis ihres Regressionskoeffizienten bzw. der Varianzwerte bei den Koeffizienten aus der Hauptachsentransformation. Experimentelle Untersuchungen mit Pausenadaption ergaben, daß 15 Geräuschparameter bzw. 10 HAT-Koeffizienten aus dem ursprünglichen Parametersatz mit 29 Koeffizienten ausreichen.

Pooltraining dient neben der Erhöhung der Robustheit zum Vortraining des Netzwerks für die nachfolgende Pausenadaption an das unbekannte Testgeräusch. Im geräuschabhängigen Fall mit Pooltraining zeigten die Worterkennungsraten nahezu keinen Unterschied zu den Ergebnissen bei Training mit (vorab bekannten) Einzelgeräuschen. Im geräuschübergreifenden Fall konnten durch Pausenadaption erhebliche Verbesserungen erreicht werden; die geräuschabhängigen Erkennungsraten bei Training mit Einzelgeräuschen wurden jedoch nicht erreicht. Gleichzeitig verringerte sich die Zahl der Iterationen durch Geräuschparameteradaption um ca. 40 %. Mit der HAT wird eine geringfügige Ergebnisverbesserung vor allem bei niedrigen SNR-Werten erreicht.

Die Optimierung der Geräuschrepräsentation erfolgte ausschließlich auf Basis der 29 bisher untersuchten Parameter. Dies führt zur Annahme, daß eine weitere Verbesserung der Geräuschparameterextraktion möglich ist. Insbesondere zur Beschreibung des Signalspektrums wurden bisher nur wenige, momentbasierte Parameter untersucht.

# 8. Vergleich mit Nichtlinearer Spektralsubtraktion

Zur Einordnung der mit neuronalen Netzen erzielten Ergebnisse ist ein Vergleich mit konkurrierenden, nichtneuronalen Verfahren notwendig. Hierzu bieten sich aus folgenden Gründen Spektralsubtraktionsverfahren an:

- Beide Ansätze sind konzeptionell ähnlich: sie gehören zur Klasse der einkanaligen Verfahren, bei denen in Sprachpausen Schätzwerte für die statistischen Eigenschaften des Geräuschsignals berechnet werden. Mit ihrer Hilfe wird die Übertragungsfunktion des Systems an instationäre Signale adaptiert. Die Abstände zwischen den Adaptionsintervallen sind durch die jeweiligen Wortlängen gegeben.

- Mit der *Nichtlinearen Spektralsubtraktion* [27] (*NSS*, Lockwood and Boudy 1992; Lockwood et al. 1992) steht ein leistungsfähiger Repräsentant dieser Klasse von Verfahren zur Verfügung, der den aktuellen Stand der Technik wiederspiegelt. In den genannten Literaturstellen werden gute Ergebnisse mit NSS für geräuschbehaftete Spracherkennung berichtet.

Im folgenden Abschnitt wird die prinzipielle Funktionsweise der Spektralsubtraktion kurz beschrieben; eine ausführlichere Diskussion des Verfahrens mit Ergebnissen ist in Reich (1985) zu finden. Anschließend werden Erweiterungen des Basisalgorithmus erläutert, die auf die *Nichtlineare Spektralsubtraktion* nach Lockwood führen. Schließlich wird die NSS experimentell evaluiert und mit den neuronalen Verfahren bezüglich Leistungsfähigkeit und Aufwand verglichen.

---

[27] Strenggenommen besitzen alle praktisch eingesetzten Spektralsubtraktionsverfahren nichtlineare Eigenschaften, vgl. Gl. (8.8). Dennoch soll im folgenden die von Lockwood vorgeschlagene Erweiterung des verallgemeinerten Wiener Filters als *Nichtlineare Spektralsubtraktion* bezeichnet werden.

## 8.1. Funktionsweise einkanaliger Spektralsubtraktionsverfahren

**Systemfunktion des Wiener Filters.** Gesucht ist ein System, das aus einem geräuschbehafteten Sprachsignal $r(i)$ einen möglichst guten Schätzwert für den geräuschfreien Sprachanteil $s(i)$ liefert, wobei die zugehörigen Spektren durch $S(l)$ bzw. $R(l)$ mit $l$ als Index für die diskrete Frequenz gegeben seien, vgl. Gl. (4.17b).

Aus der Theorie der Optimalfilter ist bekannt, daß das im Sinne des minimalen quadratischen Fehlers optimale System mit linearer und zeitinvarianter Struktur durch ein Wiener Filter gegeben ist (Kroschel 1974). Durch Anwendung dieses Ansatzes auf die Verbesserung geräuschbehafteter Sprachsignale erhält man für die Frequenzgang $A(l)$ des Wiener Filters (Reich 1985)

$$A(l) = 1 - \frac{L_N(l)}{L_R(l)} \qquad (8.1)$$

wobei $L_N$ und $L_R$ die Leistungsdichten des Geräuschsignals bzw. des gestörten Sprachsignals bedeuten. Mit diesem zeitinvarianten System kann die prinzipielle Funktionsweise veranschaulicht werden; für eine reale Anwendung wird der Ansatz wegen der i. a. instationären Signaleigenschaften auf zeitvariante Systeme erweitert.

**Einbettung ins Gesamtsystem.** Bild 8.1 zeigt ein vereinfachtes Blockbild des Erkennungssystems mit Spektralsubtraktionsstufe (schraffierter Block). Betrachtet man sie als zeitinvariantes Filter mit dem Frequenzgang $A(l)$ nach Gl. (8.1), so erhält man aus

$$\hat{S}(l) = A(l)\, R(l) \qquad (8.2)$$

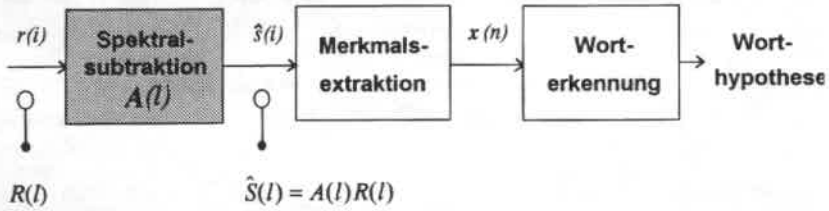



Bild 8.1: Blockbild des Erkennungssystems mit Spektralsubtraktionsstufe.

einen Schätzwert für das geräuschfreie Kurzzeitspektrum $S(l)$. Das geräuschreduzierte Sprachsignal $\hat{s}(i)$ wird dann durch Rücktransformation mit Hilfe der inversen schnellen Fouriertransformation $FFT^{-1}$ aus $\hat{S}(l)$ gemäß

$$\hat{s}(i) = FFT^{-1}\left\{\hat{S}(l)\right\} \qquad (8.3)$$

berechnet. $\hat{s}(i)$ ist das Eingangssignal für die nachfolgende Merkmalsextraktion.

**Erweiterung auf adaptive Systeme.** Um das Wiener Filter an instationäre Signale zu adaptieren, werden die Langzeit-Leistungsdichtespektren in Gl. (8.1) durch Schätzwerte auf Basis der Kurzzeitspektren des Sprach- bzw. Geräuschsignals ersetzt (Reich 1985):

$$L_N(l) \approx \left|\hat{N}(l,n)\right|^2 \qquad \text{und} \qquad (8.4a)$$

$$L_R(l) \approx \left|\hat{R}(l,n)\right|^2 \qquad . \qquad (8.4b)$$

Die Schätzwertberechnung kann wie bei den Geräuschparametern durch Mittelwertwertbildung oder Glättung vorgenommen werden, vgl. Gln. (7.13) und (7.14). Da zur Schätzung der quadrierten Betragsspektren in den Sprachpausen meist Glättungsverfahren eingesetzt werden, erfolgt ihre Berechnung gemäß

$$\left|\hat{N}(l,n)\right|^2 = \beta_N \cdot \left|\hat{N}(l,n-1)\right|^2 + \left(1-\beta_N\right)\cdot\left|N(l,n)\right|^2 \quad . \qquad (8.5)$$

Das Geräuschsignal ist nur in den Sprachpausen verfügbar; daher kann die Schätzwertberechnung ausschließlich mit Hilfe der Pausensegmente erfolgen. Gl. (8.5) entspricht einer Tiefpaßfilterung 1. Ordnung des quadrierten Betragsspektrums, wobei $\beta_N$ die Grenzfrequenz des Filters und somit den Glättungsgrad bei der Schätzung des Störspektrums bestimmt. In Sprachsegmenten wird keine Adaption des Störspektrums vorgenommen und daher der gespeicherte Schätzwert des letzten Pausensegments verwendet.

Um seine instationären Eigenschaften zu erhalten, ist eine Glättung des Sprachsignals meist unerwünscht; daher wird beim klassischen Spektralsubtraktionsansatz der Schätzwert $\hat{R}(l,n)$ in Gl. (8.4b) durch seinen tatsächlichen Wert im $n$-ten Segment approximiert, d. h.

$$\hat{R}(l,n) = R(l,n) \qquad . \qquad (8.6)$$

Einsetzen von Gln. (8.4a), (8.4b) und (8.6) in den Wiener Filter-Ansatz (8.1) ergibt für die Systemfunktion des (jetzt zeitvarianten) Wiener Filters

$$A(l,n) = 1 - \frac{\left|\hat{N}(l,n)\right|^2}{\left|R(l,n)\right|^2} \qquad (8.7)$$

**Verallgemeinertes Wiener Filter.** In praktischen Anwendungen werden häufig verallgemeinerte Wiener Filter eingesetzt, die mit Hilfe folgender Modifikationen aus Gl. (8.7) hervorgehen:

- Bei instationären Geräuschsignalen können nach Anwendung von Gl. (8.7) auf das gestörte Eingangssignal negative Werte für das geräuschreduzierte Betragsspektrum auftreten. Daher müssen die auftretenden Werte auf nichtnegative Zahlen begrenzt werden. Da niedrige Werte ohnehin fehleranfällig sind, wird häufig eine untere Grenze $c \geq 0$ mit $c \in \Re$ für die Werte von $A(l)$ festgelegt, die als *Spectral Floor* bezeichnet wird.

- Der zweite Term der rechten Seite in Gl. (8.7) bestimmt die Korrektur des gestörten Betragsspektrums mit Hilfe des beschriebenen Schätzwerte; je nach Anwendung und Geräuschsignal können unterschiedliche Gewichtungen dieses Korrekturterms die Leistungsfähigkeit des Systems günstig beeinflussen. Hierzu wird ein Faktor $a \in \Re$, der als *Überschätzfaktor* (*Overestimation Factor*) bezeichnet wird, als neuer Parameter eingeführt.

- Potenzierung der Systemfunktion mit den reellen Parametern $\alpha$ und $\gamma$ erlaubt die Realisierung unterschiedlicher Geräuschreduktionscharakteristika (Reich 1985).

Damit lautet die Systemfunktion des verallgemeinerten Wiener Filters:

$$A(l,n) = \begin{cases} \left[1 - a\left(\dfrac{\left|\hat{N}(l,n)\right|}{\left|R(l,n)\right|}\right)^{2\alpha}\right]^{\gamma} & \text{für } A(l,n) > c \\ c & \text{sonst} \end{cases} \qquad (8.8)$$

Wegen seiner guten Ergebnisse dient das *Verfahren der Teilbandbeträge*, bei dem $\alpha$=1/2 und $\gamma$=1 gesetzt werden, als Basis für die Modifikationen zur NSS. Andere Geräuschreduktionscharakteristika sind für die Erweiterung auf das NSS-

Verfahren ohne Bedeutung und werden deshalb hier nicht betrachtet. Die Systemfunktion für das Verfahren der Teilbandbeträge hat die Form

$$A(l,n) = \begin{cases} 1 - a\dfrac{|\hat{N}(l,n)|}{|R(l,n)|} & \text{für } A(l,n) > c \\ c & \text{sonst} \end{cases} \qquad (8.9)$$

**Verarbeitungsschritte**. Bild 8.2 zeigt die einzelnen Verarbeitungsschritte des Verfahrens: das Eingangssignal bilden die Abtastwerte des geräuschbehafteten Zeitsignals $r(i)$. Nach Segmentierung in überlappende Segmente mit Index $n$ und Multiplikation mit einer Fensterfunktion wird das segmentierte und gefensterte Signal $r_w(i,n)$ fouriertransformiert. Als Ergebnis erhält man das komplexe Kurzzeitspektrum $R_w(l,n)$, aus dem das Betragsspektrum gebildet wird. Der Reduktionsalgorithmus ist durch die Systemfunktion $A(l,n)$ nach Gl. (8.9) gegeben, mit deren Hilfe der Betrag des geräuschfreien Kurzzeitspektrums $S_w(l,n)$ geschätzt wird. Die Rekonstruktion des geräuschreduzierten Zeitsignals $\hat{s}(i)$ erfolgt schließlich durch Zuführung der ursprünglichen Phase, Rücktransformation und

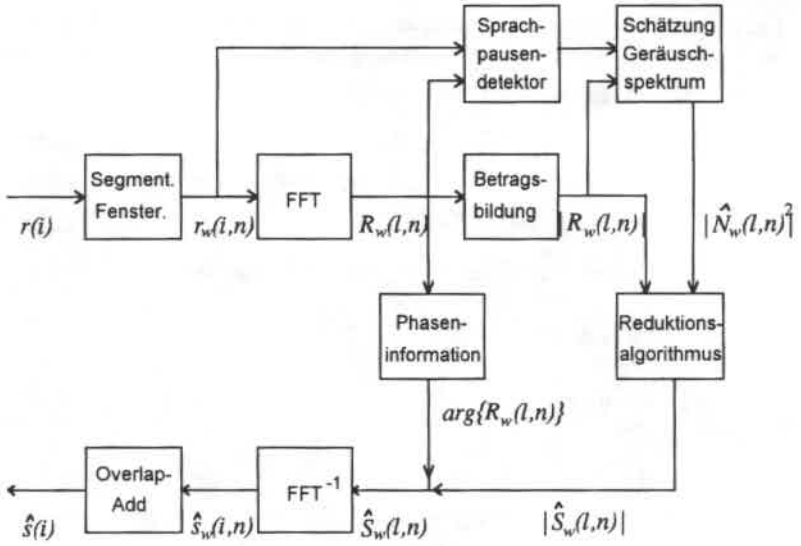



Bild 8.2: Struktur des Spektralsubtraktionssystems.

Addition zeitlich überlappender Werte aus den einzelnen Segmenten (*Overlap-Add*). Die Schätzung des Geräuschspektrums erfolgt wie beschrieben nach Gl. (8.5) in den Pausen, deren Grenzen mit Hilfe des (hier idealen) Sprachpausendetektors festgelegt werden.

### 8.2. Nichtlineare Spektralsubtraktion

Nichtlineare Spektralsubtraktion besteht aus einer Reihe von Erweiterungen des Verfahrens der Teilbandbeträge nach Gl. (8.9). Diese beziehen sich vor allem auf eine verbesserte Schätzung des Pausenspektrums sowie die SNR- und frequenzabhängige Bestimmung des Überschätzfaktors $a$. Die modifizierte Systemfunktion der NSS hat die allgemeine Form

$$A(l) = 1 - \frac{\phi(l)}{\left|\overline{R}(l)\right|} \quad , \tag{8.10}$$

wobei bei der Wahl von $\phi()$ die genannten Modifikationen realisiert werden. Außerdem wird das geräuschbehaftete Betragsspektrum $|R|$ in Gl. (8.9) durch seinen geglätteten Schätzwert ersetzt, der durch gewichtete Addition vergangener Werte mit $\beta_R \in \Re$ bestimmt wird. Seine rekursive Berechnung lautet:

$$\left|\overline{R}(l,n)\right|^2 = \beta_R \cdot \left|\overline{R}(l,n-1)\right|^2 + \left(1-\beta_R\right) \cdot \left|R(l,n)\right|^2 \quad , \tag{8.11}$$

wobei $\beta_R$ verglichen mit $\beta_N$ in Gl. (8.5) kleine Werte annimmt. Für die Funktion $\phi()$ in Gl. (8.10) wurden in Mekhaiel (1994) verschiedene Ansätze untersucht. Die besten Ergebnisse wurden mit

$$\phi\left(N_{max}, \rho\right) = e^{\frac{1-\rho(l,n)}{\kappa}} \cdot N_{max}(l,n) \tag{8.12}$$

erzielt, wobei $\rho()$ und $N_{max}()$ noch definiert werden. Ein Vergleich mit dem Verfahren der Teilbandbeträge nach Gl. (8.9) ergibt folgende Unterschiede:

1. Eine gegenüber Gl. (8.5) modifizierte Berechnung von $\left|\hat{N}(l,n)\right|$ durch Maximumsuche über $N_k$ vergangene Kurzzeitspektren für jede Frequenz $l$:

$$\left|\hat{N}(l,n)\right| = N_{max}(l,n) = \max_{n-N_k \le \tau \le n} \left\{\left|N(l,\tau)\right|\right\} . \tag{8.13}$$

Bei der Realisierung von Gl. (8.13) muß die Vorgeschichte der $N_k$ letzten Kurzzeitspektren gespeichert werden.

2. Eine SNR- und frequenzabhängige Bestimmung der Überschätzfaktors $a$ mit Hilfe der Berechnungsvorschrift

$$a(l,n) = e^{\frac{1-\rho(l,n)}{\kappa}} \qquad \text{mit } \kappa \in R \quad , \tag{8.14}$$

wobei $\rho()$ die Bedeutung eines frequenzabhängigen lokalen SNR hat, der sich aus

$$\rho(l,n) = \frac{\left|\overline{R}(l,n)\right|}{\left|\hat{N}(l,n)\right|} \tag{8.15}$$

berechnet. $\left|\overline{R}(l,n)\right|$ und $\left|\hat{N}(l,n)\right|$ sind durch Gln. (8.11) und (8.13) gegeben. Durch Gl. (8.14) wird ein "weicher" Übergang des Überschätzfaktors von $a=1$ für niedrige SNR-Werte auf $a=0$ bei ungestörten Signalen realisiert. Der vom lokalen SNR $\rho$ abhängige Verlauf dieses Übergangs ist von der Wahl des Parameters $\kappa$ abhängig [28].

3. Das Ersetzen des geräuschbehafteten Betragsspektrums $|R(l,n)|$ durch den geglätteten Schätzwert $\left|\overline{R}(l,n)\right|$ nach Gl. (8.11), wobei der Glättungsparameter $\beta_R$ in der Regel kleiner als in Gl. (8.15) gewählt wird.

Einsetzen von Gln. (8.13) und (8.14) in (8.9) ergibt die Systemfunktion der Nichtlinearen Spektralsubtraktion $A_{NSS}$:

$$A_{NSS}(l,n) = \begin{cases} 1 - e^{\frac{1-\rho(l,n)}{\kappa}} \cdot \dfrac{N_{max}(l,n)}{\left|\overline{R}(l,n)\right|} & \text{für } A(l,n) > c \\ c & \text{sonst} \end{cases} \quad . \tag{8.16}$$

Dieser Formulierung der NSS nach Gl. (8.16) liegen die im folgenden Abschnitt beschriebenen Experimente zugrunde.

---

[28] siehe Mekhaiel (1994)

### 8.3. Experimentelle Ergebnisse

Die Parameterwerte für die NSS wurden wie in Tabelle 8.1 gezeigt gewählt. Dabei erfolgte gegenüber den in Mekhaiel (1994) beschrieben Experimenten keine erneute Optimierung auf die erweiterte Datenbasis.

Als Vergleich dienen die in Kapitel 7 beschrieben Experimente mit Pooltraining und Pausenadaption. Sie basieren auf dem *100-20-10* Netzwerk mit je 10 lpc-cepstrum Koeffizienten und HAT-Steuerparametern pro Segment bei 5 Kontextvektoren.

Tabelle 8.1: Parameter für die Experimente mit Nichtlinearer Spektralsubtraktion.

| Par. | Bedeutung | Wert |
|---|---|---|
| $\beta_N$ | Glättungsparameter für $N$ | 0,8 |
| $\beta_R$ | Glättungsparameter für $R$ in Gl. (8.15) | 0,5 |
| $\beta_R$ | Glättungsparameter für $R$ in Gl. (8.16) | 0,1 |
| $c$ | Spectral Floor | 0,1 |
| $\kappa$ | Parameter für lokalen SNR $\rho$ in Gl. (8.14) | 1,1 |
| $N_k$ | Zahl der gespeicherten Spektren | 20 |

Die Auswertung der Ergebnisse erfolgte wie bei den Simulationsreihen mit Pooltraining und Pausenadaption (Bilder 7.6a und 7.6b) getrennt nach geräuschabhängigen bzw. geräuschübergreifenden Experimenten. Im Gegensatz zu den neuronalen Verfahren besteht bei NSS kein prinzipieller Unterschied zwischen diesen beiden Fällen, da hier kein Vortraining erforderlich ist. Zum Vergleich sind in beiden Ergebnisdiagrammen (Bilder 8.3a und 8.3b) die Ergebnisse ohne Geräuschreduktion aufgetragen (untere Kurven).

Bild 8.3a zeigt die **geräuschabhängigen** Erkennungsraten mit 10 lpc-cepstrum-Koeffizienten: die NN-basierten Ergebnisse sind für SNR-Werte $\leq 10$ dB einige Prozent besser als die mit NSS. Wie bereits beschrieben, bestehen bei den neuronalen Ergebnissen mit Pooltraining und Pausenadaption im Vergleich zu denjenigen mit Einzelgeräuschtraining nahezu keine Unterschiede.

Bei **geräuschübergreifenden** Experimenten (Bild 8.3b) unterscheiden sich die Ergebnisse deutlicher: während hier Pooltraining mit anschließender Pausenadaption aufgrund der in Abschnitt 7.4.2 getroffenen Feststellungen deutlich schlechter abschneidet als die Experimente mit Einzelgeräuschtraining, erhält man mit NSS bessere Ergebnisse als mit pooltrainiertem Netzwerk und Pausenadaption. Bei initialem Training mit Einzelgeräuschen fallen die NN-basierten Ergebnisse in beiden Fällen besser aus.

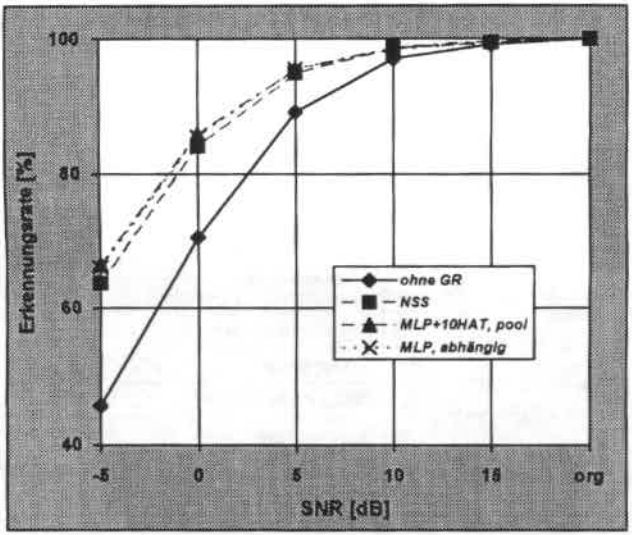



Bild 8.3a: Vergleich der Worterkennungsraten bei Geräuschreduktion mit NSS und neuronalen Netzwerken (**geräuschabhängiger** Fall bei Pooltraining).

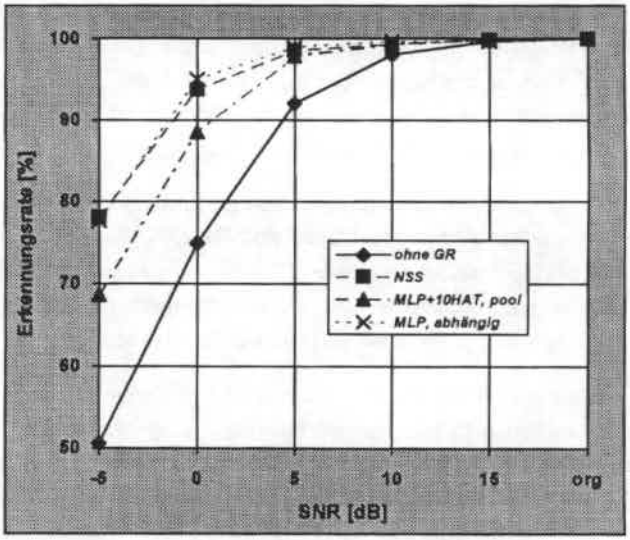



Bild 8.3b: Vergleich der Worterkennungsraten bei Geräuschreduktion mit NSS und neuronalen Netzwerken (**geräuschübergreifender** Fall bei Pooltraining).

## 8.4. Zusammenfassung des Vergleichs mit Nichtlinearer Spektralsubtraktion

Zur Einstufung der mit neuronalen Netzen erzielten Ergebnisse wurde ein Vergleich mit den Ergebnissen der Nichtlinearen Spektralsubtraktion nach Lockwood und Boudy (1992) sowie Lockwood et al. (1992) durchgeführt. Die NSS ist ein Vertreter der einkanaligen Spektralsubtraktionsverfahren, der auf Erweiterungen des verallgemeinerten Wiener Filter-Ansatzes beruht und aufgrund der in der Literatur berichteten guten Ergebnisse dem derzeitigen Stand der Technik entspricht.

Die Erweiterungen umfassen im wesentlichen zwei Punkte: erstens eine modifizierte Schätzung des Geräuschspektrums in den Sprachpausen, die auf vergangenen Maximalwerten beruht, und zweitens die frequenz- und SNR-abhängige Bestimmung des Überschätzfaktors. Darüber hinaus wird in der Systemfunktion das Betragsspektrum des geräuschbehafteten Sprachsignals im aktuellen Segment durch seinen geglätteten Wert ersetzt.

Ein Ergebnisvergleich führt zu folgenden Schlußfolgerungen: wenn eine Aufnahme der Testgeräuschquelle im Trainingspool enthalten ist (geräuschabhängiger Fall), schneidet die NSS einige Prozent schlechter als die neuronalen Verfahren mit Pausenadaption ab. Bei geräuschübergreifenden Experimenten (Testquelle nicht im Trainingspool) sind die NSS-Ergebnisse trotz Pausenadaption des Netzwerks bei stark geräuschbehafteter Sprache deutlich besser. Bei initialem Training mit Einzelgeräuschen fallen die Ergebnisse mit neuronalen Netzen in jedem Fall besser aus.

# 9. Zusammenfassung und Ausblick

**Neuronaler Ansatz zur Störreduktion.** Neuronale Ansätze zur Störreduktion zeigen im Vergleich zu bisherigen, aus der Literatur bekannten nichtneuronalen Verfahren u. a. folgende Unterschiede:

- Wegen ihrer Approximationseigenschaften sind neuronale Netzwerke in der Lage, ein inverses Modell der gesamten Störungseinflüsse zu lernen. Insbesondere können hiermit auch Kombinationen aus mehreren Störungsarten behandelt werden.
- Im Gegensatz zu den *lernfähigen Verfahren* stellen künstliche neuronale Netzwerke *lernfähige Modelle* dar, die ihre Struktur und Abbildungseigenschaften an die Erfordernisse der vorliegenden Aufgabe adaptieren.

Von den vielfältigen Einsatzmöglichkeiten wurde für den experimentellen Teil der vorliegenden Arbeit eine Beschränkung auf den Einsatz neuronaler Netzwerke als nichtlineare Filter zur einkanaligen Reduktion additiver Störgeräusche getroffen. Ziel war dabei, die Leistung von Spracherkennungssystemen in geräuscherfüllter Umgebung zu verbessern.

**Ergebnisse zur Reduktion additiver Störgeräusche.** Der einmaladaptive Ansatz auf Basis des *Multilayer Perzeptron*-Netzwerks stellt nach Abschluß des Trainings mit *Error Backpropagation* ein optimales (zeitinvariantes) Schätzsystem im Sinne des Bayes-Kriteriums dar, wenn die Topologie geeignet gewählt und im Training das globale Minimum gefunden wurde.

Die experimentellen Untersuchungen auf Basis der Merkmalsvektorfolge wurden in einem Testbett zur Isoliertworterkennung mit rechneraddierten Geräuschdaten aus unterschiedlichen Quellen eines Mobilfunkszenarios durchgeführt. Mit dem einmaladaptiven Geräuschreduktionsnetzwerk wurde eine Verbesserung der mittleren sprecherabhängigen Erkennungsrate von ca. 75 % auf 95 % bei 0 dB SNR erreicht. Die Untersuchungen zeigten, daß hierzu die nichtlineare Verarbeitungskapazität und die Berücksichtigung von Signalkontext am Netzwerkeingang beitragen.

Das Standardtrainingsverfahren wurde um einen Generalisierungstest mit *Cross Validation*, eine variable Lernrate sowie die zufällige Auswahl der Traingsbeispiele ergänzt. Die Robustheit gegenüber moderaten Veränderungen der Signalparameter kann durch Berücksichtigung von Parameterstreuungen bei der Auswahl des Trainingsmaterials erhöht werden (z. B. *Multi-SNR-Training*).

Nachteile der MLP-Netzwerke sind lange Trainingszeiten sowie die zeitaufwendige Entwicklung einer problemangepaßten Topologie. Zur Vermeidung des Rechenaufwands in der Anwendungsumgebung wurde das beim Hersteller durchführbare *Geräuschpooltraining* untersucht. Es führte im Vergleich zum geräuschabhängigen Training zu schlechteren Ergebnissen; dennoch ergaben geräuschübergreifende Tests trotz Fehlanpassung noch Verbesserungen im Vergleich zur Situation ohne Geräuschreduktion. Dies ist auf die Generalisierungsfähigkeit der pooltrainierten Netze auf neue Geräuschquellen zurückzuführen. Die Integration von zusätzlichen Vorverarbeitungsschritten in das Netzwerk ist möglich und dient zur Aufwandsreduktion in Echtzeitrealisierungen.

Zur automatischen Netzgenerierung wurden zwei verschiedene Verfahren untersucht: das *Resource Allocating Network* und der *Cascade Correlation*-Lernalgorithmus. Beide benötigten deutlich kürzere Trainingszeiten (ca. Faktor 5) und lieferten kompaktere Topologieen; die erzielten Erkennungsraten waren jedoch einige Prozent schlechter als nach der Geräuschreduktion mit dem MLP.

Zur schnellen Adaption an instationäre Geräuschsignale kann die Abbildungsfunktion des Netzwerks mit Hilfe von Geräuschparametern gesteuert werden. Diese wurden in den Sprachpausen aus dem Zeit-, Spektral- und Cepstralbereich extrahiert. Pooltrainierte Netzwerke mit anschließender *geräuschparameterbasierter Adaption* zeigten ähnliche Ergebnisse wie mit Einzelgeräuschen trainierte Netzwerke, wenn eine Stichprobe des Testgeräuschs im Trainingspool enthalten war. Bei nichttrainierten Testgeräuschen fielen die Verbesserungen geringer aus, wobei der Gewinn von der Repräsentation des Pausensignals abhing.

Die *Hauptachsentransformation der Geräuschparameter* hatte trotz reduzierter Dimensionalität von 15 auf 10 Koeffizienten Ergebnisverbesserungen zur Folge. Es kann angenommen werden, daß eine weitere Steigerung der Geräuschreduktionsleistung durch Optimierung der Signalrepräsentation möglich ist. Dies betrifft sowohl die Extraktionsverfahren als auch die Verfahren zur Schätzung der Pausenwerte.

Ein Vergleich der neuronalen Verfahren mit *Nichtlinearer Spektralsubtraktion* zeigt im geräuschabhängigen Fall etwas höhere Erkennungsraten bei Geräuschreduktion mit neuronalen Netzwerken. Bei nicht vortrainierten Geräuschquellen werden mit Hilfe der Nichtlinearen Spektralsubtraktion bessere Ergebnisse erreicht.

**Ausblick.** Eine Fortführung der bisherigen Arbeiten könnte die beiden Schwerpunkte *Verfahrensoptimierung* und *Erweiterungen des Ansatzes* umfassen. Zur Verfahrensoptimierung sind neben den bereits genannten Punkten folgende Untersuchungen aussichtsreich:

- Einsatz rekurrenter Netzwerkstrukturen zur Generierung kompakterer Topologieen insbesondere für die Verarbeitung von Kontextinformation.
- Untersuchung modularer Netzwerkstrukturen mit störungsabhängig spezialisierten Teilnetzen zur Behandlung komplexer Geräusch- bzw. Störungsarten.
- Vergrößerung der Geräuschdatenbasis zur Adaption an neue Anwendungsumgebungen.
- Einsatz der entwickelten Netzwerke in leistungsfähigeren Erkennungssystemen zur Verarbeitung kontinuierlicher Sprache. Dabei ist die Frage nach den Adaptionsintervallen für die Geräuschparameter zu klären.

Zukünftige Erweiterungen des Ansatzes könnten folgende Punkte umfassen:

- Zweikanalige Verarbeitung zur kontinuierlichen Netzwerkadaption bei Verfügbarkeit eines Referenzkanals für die Geräuschinformation.
- Gleichzeitige Reduktion mehrerer Störungsarten (z. B. Kanalverzerrungen und Hintergrundgeräusche).
- Neuronaler Ansatz zur Verbesserung der Sprachverständlichkeit (z. B. im Mobilfunk).

# ANHANG

## A.1. Wortliste zur Steuerung des Textverarbeitungssystems

1. Null
2. Eins
3. Zwei
4. Drei
5. Vier
6. Fünf
7. Sechs
8. Sieben
9. Acht
10. Neun
11. Hilfe
12. Wiederholen
13. Anfang
14. Ende
15. Richtig
16. Löschen
17. Ändern
18. Modifizieren
19. Einfügen
20. Streichen
21. Dokument
22. Inhalt
23. Schreiben
24. Lesen
25. Verbinden
26. Links
27. Rechts
28. Nächster
29. Transfer
30. Speicher

## A.2. Zeitsignale und Leistungsdichtespektren der Geräuschaufnahmen

Zeitsignale und Leistungsdichtespektren der Geräuschaufnahmen mit folgender Kurzbezeichnung (Beschreibung der Datenbasis siehe Abschnitt 4.1):

- Bahnhofshalle (Bilder A2.1a und b)
- Nadeldrucker (Bilder A2.2a und b)
- Gaststätte (Bilder A2.3a und b)
- IBM-Matrixdrucker (Bilder A2.4a und b)
- Rechnerraum (Bilder A2.5a und b)
- Spülmaschine (Bilder A2.6a und b)
- Straßenbauarbeiten (Bilder A2.7a und b)

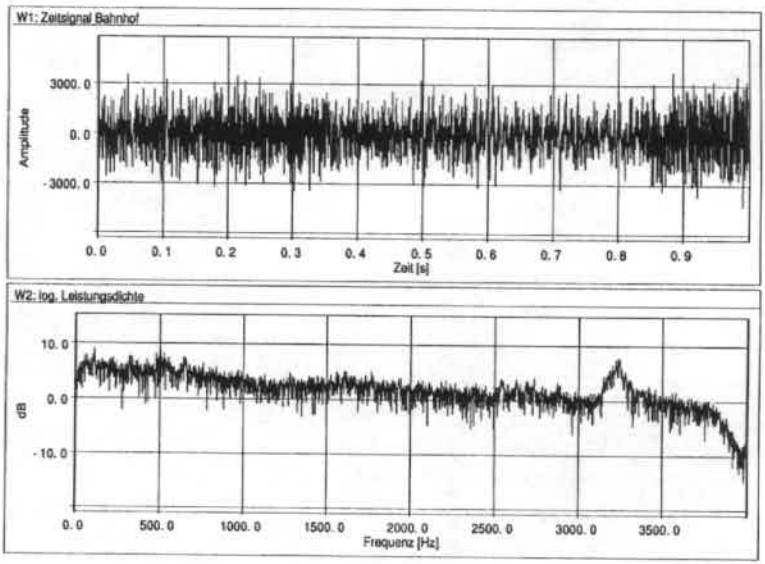



Bild A2.1a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum der Geräuschaufnahme in einer **Bahnhofshalle**.

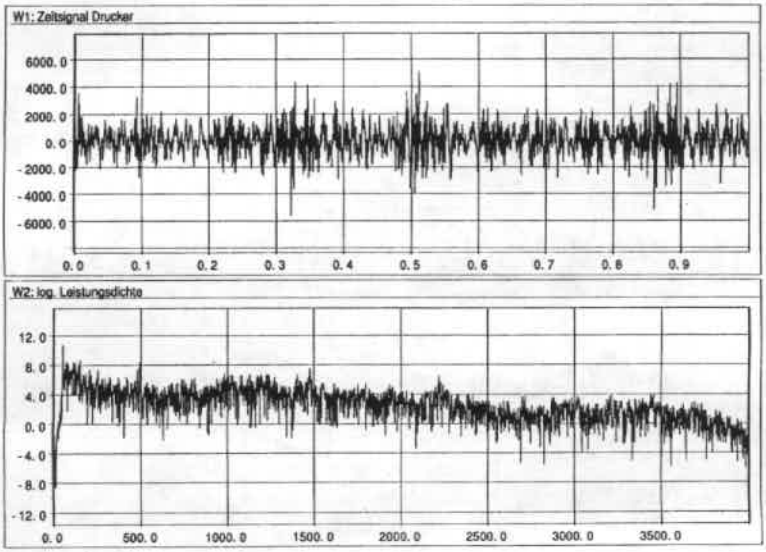



Bild A2.2a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum der Geräuschaufnahme eines **Nadeldruckers**.

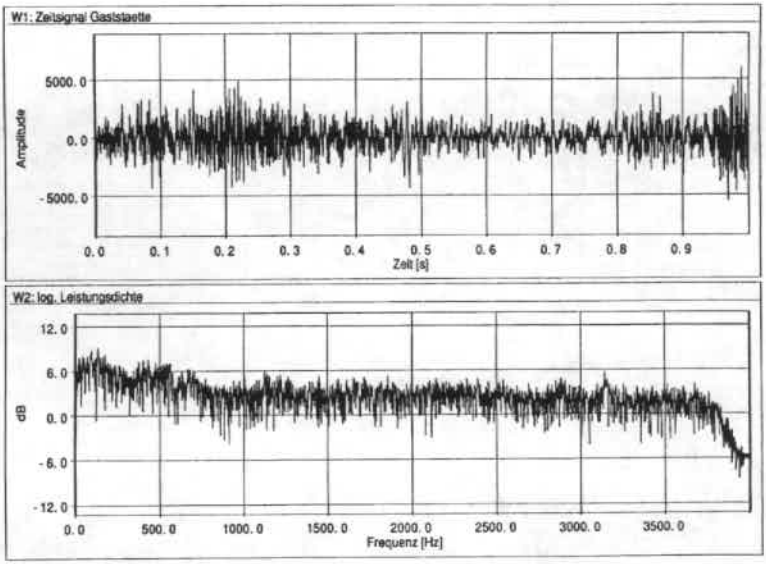



Bild A2.3a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum der Geräuschaufnahme in einer **Gaststätte**.

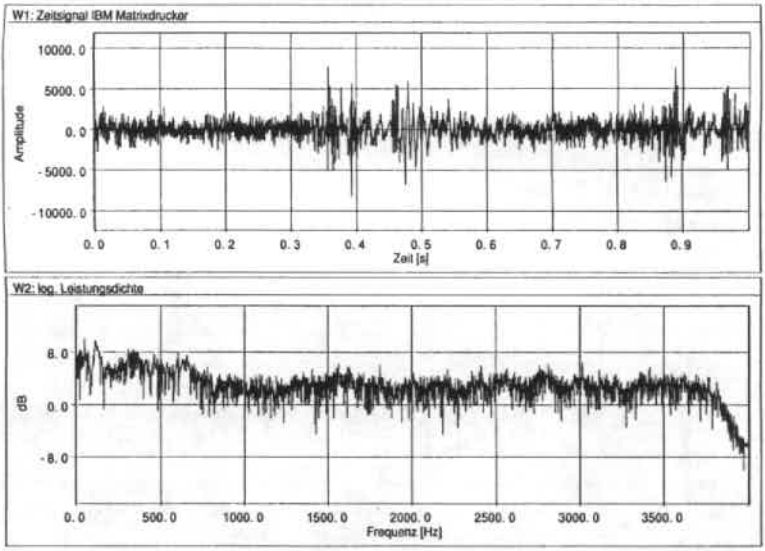



Bild A2.4a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum der Geräuschaufnahme eines **IBM-Matrixdruckers**.

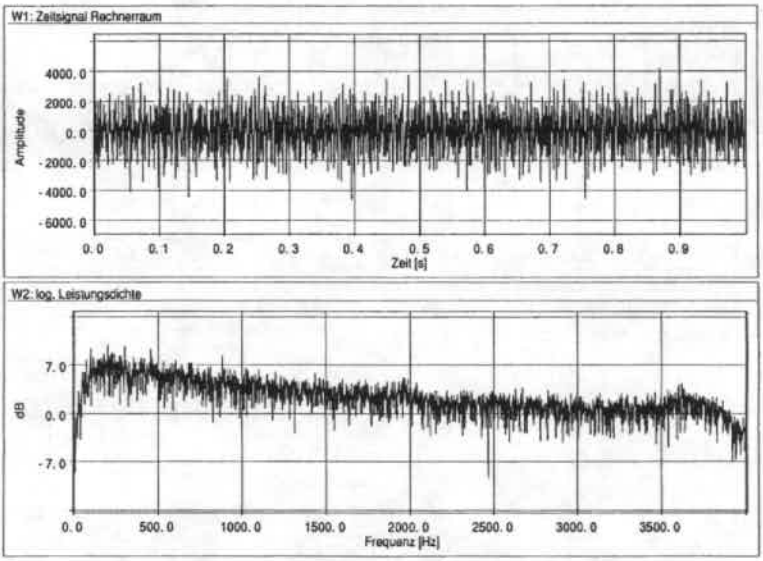



Bild A2.5a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum der Geräuschaufnahme in einem **Rechnerraum**.

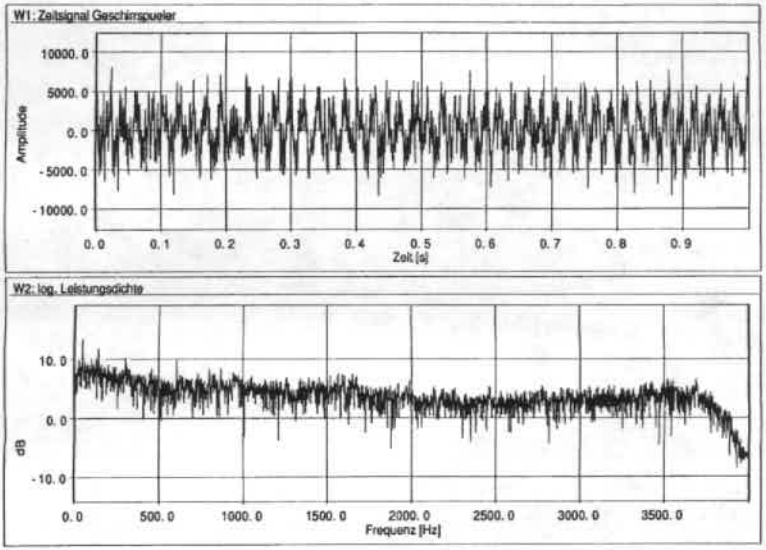



Bild A2.6a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum der Geräuschaufnahme einer **Spülmaschine**.

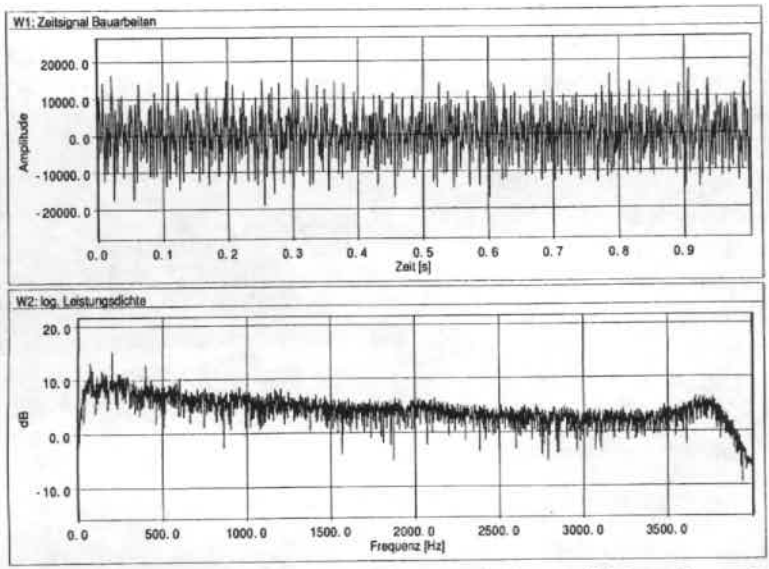



Bild A2.7a und b: Zeitsignal und logarithmiertes Leistungsdichtespektrum einer Geräuschaufnahme von **Straßenbauarbeiten**.

### A.3. Trainingsparameter in der Steuerdatei des Cascade Correlation-Lernalgorithmus

```
#
# Training parameters for a 10-r-10 network for noise reduction
#
# Input and output units:
#
Ninputs             10
Noutputs            10
#
OutputType          LINEAR
UnitType            SIGMOID
#
ErrorMeasure        INDEX
Test                TRUE
UseCache            TRUE
#
# Quickprop parameters
#
MaxUnits            50
Outputmu            1.75
OutputEspilon       0.35
OutputDecay         0.0001
NonRandomSeed       TRUE
ErrorIndexTreshold  0.1
OutputPatience      10
InputPatience       10
WeightRange         0.2
NCandidates         8
#
# Start with noise reduction:
#
Mapping
#
```

# LITERATURVERZEICHNIS


**Acero A, Stern R** (1992) Cepstral Normalization for Robust Speech Recognition. ESCA Workshop on Speech Processing in Adverse Conditions, Cannes, 89-92

**AKG Q400T**, Datenblatt der Firma AKG, 1989

**Angleys G** (1991) Sprachdatensammlung zur Steuerung eines Textverarbeitungssystems. Interne Dokumentation, Alcatel SEL AG

**Applebaum T, Hanson B** (1990): Robust Speaker-Independent Word Recognition Using Spectral Smoothing and Temporal Derivatives. EUSIPCO, 1183-1186

**Barbier L, Chollet G** (1991) Robust Speech Parameters Extraction for Word Recognition in Noise using Neural Networks. IEEE ICASSP, 145-148

**Battiti R** (1992) First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method. Neural Computation, Vol 4, 141-166

**Beyer MC 723**, Datenblatt der Firma Beyer, 1990

**Bodenhausen U, Waibel A** (1993) Tuning by Doing: Flexibility through Automatic Structure Optimization. EUROSPEECH, 1485-1488

**Bodenhausen U** (1994) Automatic Structuring of Neural Networks for Spatio-Temporal Real-World Applications. Dissertation, Universität Karlsruhe

**Chen H-Q** (1994) Untersuchung sequentieller Lernverfahren zur Adaption neuronaler Geräuschreduktionsnetze an instationäre Signalumgebungen. Diplomarbeit, Alcatel SEL Forschungszentrum Stuttgart und Institut für Nachrichtentechnik der Universität Karlsruhe

**Crowder S, Fahlman S** (1991) C Implementation of the Cascade-Correlation Learning Algorithm. Version 1.31, Release Date March 21, Carnegie Mellon University, USA

**DSP56ADC16**, Datenblatt der Firma Motorola, 1989, Phoenix, Arizona

**Dvorak S, Hörmann T** (1991) Noise-Robust Speech Recognition by Template Adaptation. DAGA, 1037-1040

**Eckhardt H** (1992) Geräuschdatensammlung in Industrie- und Büroumgebung, Interner Bericht, Alcatel SEL Forschungszentrum Stuttgart

**Eckhardt H** (1993) Digitale Filterung und Abtastratenreduktion von Geräuschsignalen. Interner Bericht, Alcatel SEL Forschungszentrum Stuttgart

**Eckhardt H** (1994) Extraktion von statistischen Parametern zur Beschreibung des Geräuschsignals in Sprachpausen. Interner Bericht, Alcatel SEL Forschungszentrum Stuttgart

**Eckhardt H, Trompf M, Angleys G, Hackbarth H** (1992) Robust Signal Preprocessing for Word Recognition in Noisy Environment. ESCA Workshop on Speech Processing in Adverse Conditions. Cannes, Frankreich, 85-88

**Eppinger B, Herter E** (1993) Sprachverarbeitung, Reihe Informationstechnik/ Nachrichtentechnik, Hanser-Verlag, München, Wien

**Fahlman S** (1988) Faster-Learning Variations on Back-Propagation: An Emperical Study. Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufmann

**Fahlman S** (1990) The Recurrent Cascade-Correlation Architecture, NIPS 1990, 190-196

**Fahlman S, Lebiere C** (1989) The Cascade-Correlation Learning Architecture, NIPS 1989, 524-532

**Fahlman S, Lebiere C** (1991) The Cascade-Correlation Learning Architecture. Carnegie Mellon University, TR CMU-CS-90-100

**Franzini M** (1987) Speech Recognition with Back Propagation. Ninth Annual Conference of the IEEE Engineering in Medicine and Biology Society, Boston, 1702-1703.

**Furui S** (1989) Digital Speech Processing, Synthesis, and Recognition. Marcel Dekker, New York

**Furui S** (1992) Toward Speech Recognition under Adverse Conditions. ESCA Workshop on Speech Processing in Adverse Conditions, Cannes, 31-42

**Hampshire J, Waibel A** (1989) The Meta-Pi Network: Building Distributed Knowledge Representations for Robust Pattern Recognition. Tech. Report CMU-CS-89-166-R, Carnegie Mellon University, School of Computer Science

**Hampshire J, Waibel A** (1990) A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks. IEEE Trans. on Neural Networks, Vol. 1, No. 2, 216-228

**Hanson B, Applebaum T** (1990) Robust Speaker-Independent Word Recognition Using Static, Dynamic and Acceleration Features: Experiments with Lombard and Noisy Speech. IEEE ICASSP, 857-860

**Hanson J, Bria O** (1992) Improved Automatic Recognition of Speech in Noise and Lombard Effect. EUSIPCO, 403-406

**Hecht-Nielsen R** Neurocomputing, Addison-Wesley Publishing Company, 1990

**Hermansky H** (1990) Perceptual Linear Predictive (PLP) Analysis of Speech. JAcoustSocAm 87(4), 1738-1752

**Hermansky H, Morgan N, Bayya A, Kohn P** (1991) Compensation for the Effect of the Communication Channel in Auditory-Like Analyses of Speech, EUROSPEECH, 1367-1370

**Hertz J, Krogh A, Palmer R** (1991) Introduction to the Theory of Neural Computation. Addison-Wesley

**Hild H, Waibel A** (1993a) Speaker-Independent Connected Letter Recognition with a Multi-State Time Delay Neural Network. EUROSPEECH, 1481-1484

**Hild H, Waibel A** (1993b) Multi-Speaker/Speaker-Independent Architectures for the Multi-State Time Delay Neural Network. IEEE ICASSP, Vol. 2, 255-258

**Hild H, Waibel A** (1993c) Connected Letter Recognition with a Multi-State Time Delay Neural Network. Advances in Neural Network Information Processing Systems (NIPS-5), Morgan Kaufmann

**Hinton G** (1987) Connectionist Learning Procedures. Carnegie Mellon University, Technical Report CMU-CS-87-115

**Hörmann T, Eckhardt H, Trompf M, Hackbarth H** (1993) A Noise-Robust Real-Time Word Recognition Hardware Module. ESCA EUROSPEECH, 1833-1836

**Hornik K, Stinchcombe M and White H** (1989) Multilayer Feed-Forward Networks are Universal Approximators. Neural Networks, Vol 2, 359-366

**Huang X** (1992) Speaker Normalization for Speech Recognition, IEEE ICASSP 1992, Vol 1, 465-469

**Huang X, Lee K, Waibel A** (1991) Connectionist Speaker Normalization and its Applications to Speech Recognition, IEEE Workshop on Neural Networks for Signal Processing, 357-366

**Jain A** (1991) PARSEC: A Connectionist Learning Architecture for Parsing Spoken Language. Doctoral Thesis, CMU-CS-91-208, School of Computer Science, Carnegie Mellon University

**Junqua JC** (1992) The Variability of Speech Produced in Noise. ESCA Workshop on Speech Processing in Adverse Conditions, Cannes, 43-52

**Kadirkamanathan V, Niranjan M** (1993) A Function Estimation Approach to Sequential Learning with Neural Networks. Neural Computation 5, 554-575

**Krause A** (1991) Clustering-Verfahren zur Generierung von Referenzmustern für die Worterkennung mit Dynamic Time Warping. Interner Bericht, Alcatel SEL Forschungszentrum Stuttgart

**Kroschel K** (1973) Statistische Nachrichtentheorie (Erster Teil). Springer-Verlag, Berlin-Heidelberg-New York

**Kroschel K** (1974) Statistische Nachrichtentheorie (Zweiter Teil). Springer-Verlag, Berlin-Heidelberg-New York

**Kroschel K** (1988) Umgebungsgeräuschreduktion bei Sprachkommunikationssystemen. Frequenz, 42(1988), 79-84

**Lechner W, Lohl N** (1990) Analyse Digitaler Signale, Vieweg&Sohn, Braunschweig, 112-113

**Le Cun Y, Denker J, Solla S** (1989) Optimal Brain Damage, NIPS, 598-605

**Levin E** (1990) Word Recognition Using the Hidden Control Neural Architecture, IEEE ICASSP, 433-436

**Lippmann RP** (1987) An Introduction to Computing with Neural Nets. IEEE ASSP magazine, April 1987, 4-22

**Lippmann RP, Martin EA, Paul DB** (1987) Multi-Style Training for Robust Isolated Word Speech Recognition, IEEE ICASSP, 705-708

**Lockwood P, Boudy J** (1991) Experiments with a Nonlinear Spectral Subtractor (NSS), Hidden Markov Models and the Projection, for Robust Speech Recognition in Cars. EUROSPEECH, 79-83

**Lockwood P, Boudy J** (1992) Experiments with a Nonlinear Spectral Subtractor (NSS), Hidden Markov Models and the Projection, for Robust Speech Recognition in Cars. Speech Communication 11, North Holland, 215-228

**Lockwood P, Boudy J, Blanchet M** (1992) Non-Linear Spectral Subtraction (NSS) and Hidden Markov Models for Robust Speech Recognition in Car Noise Environments. IEEE ICASSP, 265-268

**Manke S, Finke M, Waibel A** (1995) The Use of Dynamic Writing Information in a Connectionist On-Line Cursive Handwriting Recognition System. Advances in Neural Network Information Processing Systems (NIPS-7), Morgan Kaufmann

**Markel D, Gray H** (1976) Linear Prediction of Speech. Springer-Verlag, New York

**Mekhaiel M** (1994) Untersuchung und Vergleich nichtlinearer Geräuschreduktionsverfahren für robuste Spracherkennung. Diplomarbeit, Alcatel SEL Forschungszentrum Stuttgart und Institut für Nachrichtentechnik der Universität Karlsruhe

**Morgan N, Bourlard H** (1989) Generalization and Parameter Estimation in Feedforward Nets: Some Experiments. Technical Report TR-89-017, International Computer Science Institute, Berkeley

**Ney H** (1984) The Use of a One-Stage Dynamic Processing Algorithm for Connected Word Recognition. IEEE Trans. ASSP-32, No. 2, 263-271

**Nowlan S, Hinton G** (1991) Simplifying Neural Networks by Soft Weight-Sharing, Proceedings of Neural Information Processing Systems (NIPS), Denver

**Paliwal K** (1990) Neural Net Classifiers for Robust Speech Recognition under Noisy Environments. IEEE ICASSP, 429-432

**Paliwal K** (1992) Dimensionality Reduction of the Enhanced Feature Set for the HMM Speech Recognizer. Digital Signal Processing 2, 1992(92), 157-173

**Petek B, Waibel A, Tebelskis J** (1992) Integrated Phoneme and Function Word Architecture of Hidden Control Neural Networks for Continuous Speech Recognition. Speech Communication 11, 273-282, North-Holland

**Platt J** (1991) A Resource Allocating Network for Function Interpolation. Neural Computation 3(2), 213-225

**Reich W** (1985) Adaptive Systeme zur Reduktion von Umgebungsgeräuschen bei Sprachübertragung. Dissertation, Universität Karlsruhe

**Rojas R** (1993) Theorie der Neuronalen Netze: eine systematische Einführung. Springer-Lehrbuch

**Richter R** (1993) Datenreduktionsverfahren für die Signalvorverarbeitung zur automatischen Spracherkennung. Praktikumsbericht, Alcatel SEL Forschungszentrum Stuttgart und Universität Dresden

**Rühle A** (1994) Einsatz neuronaler Netzwerke zur Geräuschreduktion für robuste Spracherkennung. Praktikumsbericht zum 2. Industriesemester, Alcatel SEL Forschungszentrum Stuttgart und Fachhochschule Esslingen

**Rumelhart D, McClelland J, and The PDP Research Group** (1986) Parallel Distributed Processing. Volume 1: Foundations, MIT Press

**Ruske G** (1988) Automatische Spracherkennung. R. Oldenbourg Verlag, München, Wien

**Schotola T** (1984) On the Use of Demisyllables in Automatic Speech Recognition. Speech Communication 3, Elsevier Science Publishers B.V., North-Holland, 63-87

**Sickert K** (1983) Automatische Spracheingabe und Sprachausgabe. Markt&Technik

**Sorensen H** (1991) A Cepstral Noise Reduction Multi-Layer Neural Network. IEEE ICASSP, 933-936

**Sorensen H, Hartmann U** (1991) A Self-Structuring Neural Noise Reduction Model. EUROSPEECH, 567-570

**Sorensen H, Hartmann U** (1992) Self-Structuring Hidden Control Neural Model for Speech Recognition. IEEE ICASSP, Vol2, 353-356

**Stone M** (1978) Cross-Validation: A Review. Math. Operationsforsch. Statist., Ser. Statistics, 9(1)

**SUNROM-1**, Datenblatt der Noise CD-ROM aus ESPRIT Project 2094 *SUNSTAR*

**Tamura S, Waibel A** (1988) Noise Reduction Using Connectionist Models. IEEE ICASSP, 53-56

**Tamura S** (1989) An Analysis of an Noise Reduction Neural Network. IEEE ICASSP, 2001-2004

**Tamura S, Nakamura M** (1990) Improvements to the Noise Reduction Neural Network. IEEE ICASSP, 825-828

**Tebelskis J, Waibel A, Petek B, Schmidbauer O** (1991) Continuous Speech Recognition Using Linked Predictive Neural Networks. IEEE ICASSP, 61-64

**Tebelskis J, Waibel A** (1993) Performance Through Consistency: MS-TDNN's for Large Vocabulary Continuous Speech Recognition. Advances in Neural Network Information Processing Systems (NIPS-5), Morgan Kaufmann

**Thierer G** (1987) LPC-Analyse zur Merkmalsextraktion für die automatische Spracherkennung. Interner Bericht, Alcatel SEL Forschungszentrum Stuttgart

**Trompf M** (1992a) Building Blocks for a Neural Noise Reduction Network for Robust Speech Recognition. EUSIPCO, Vol 1, 431-434

**Trompf M** (1992b) Experiments with Noise Reduction Neural Networks for Robust Speech Recognition. ICSI Technical Report TR-92-035

**Trompf M** (1993) 1. Halbjahresbericht 1993 zum Alcatel SEL Teilvorhaben *Störrobustheit* des BMFT-geförderten Verbundprojekts Verbmobil.

**Trompf M, Eckhardt H** (1991) Simulationssystem für geräuschrobuste Isoliertworterkennung. Abschlußbericht des Alcatel SEL Teilvorhabens zum BMFT-Projekt *Architekturen von Speech und Language*

**Trompf M, Eckhardt H, Mekhaiel M** (1994) An Environment-Adaptive Noise Reduction Neural Network for Reliable Speech Recognition. EUSIPCO, 1202-1205

**Trompf M, Hackbarth H** (1993) Neural Noise Reduction Using Distortion-Robust Speech Signal Representations. DAGA, Frankfurt, 1020-1023

**Trompf M, Richter R, Eckhardt H, Hackbarth H** (1993) Combination of Distortion-Robust Feature Extraction and Neural Noise Reduction for ASR. EUROSPEECH, 1039-1042

**Vary P** (1983) Verfahren zur digitalen Verbesserung gestörter Sprache. TEKADE Technische Mitteilungen, 70-76

**Waibel A, Sawai H, Shikano K** (1989) Modularity and Scaling in Large Phonemic Neural Networks. IEEE Trans. on ASSP, Vol 37, No 12, 1888-1898

**Werbos P** (1988) Backpropagation: Past and Future. IEEE ICNN, Vol. 1, 343-353

**White H** (1990) Connectionists Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings. Neural Networks 3, 535-549

**Widrow B, Glover R, McCool M, Kaunitz J, Williams S, Hearn H, Zeidler S, Dong E, Goodlin C** (1975) Adaptive Noise Cancelling: Principles and Applications. Proc. IEEE, Vol 63, No 12, 1692-1716

**Zell A** (1994) Simulation Neuronaler Netze. Addison-Wesley

# LEBENSLAUF

| | |
|---|---|
| 28. November 1957 | geboren in St. Georgen i. Schw., Schwarzwald-Baar-Kreis |
| Apr. 1964 bis Aug. 1967 | Grundschule in St. Georgen |
| Sept. 1967 bis Juni 1976 | Gymnasium in St. Georgen, Abitur |
| Juli 1976 bis Sept. 1977 | Grundwehrdienst |
| Okt. 1977 bis Juni 1978 | Praktika für das Elektrotechnikstudium T. Baeuerle&Söhne, St. Georgen, und Kienzle Apparate GmbH, Villingen |
| Okt. 1978 bis März 1985 | Elektrotechnikstudium an der Universität Karlsruhe (TH), Diplom |
| Juli 1985 bis Juni 1987 | Entwicklungsingenieur im Forschungszentrum der Daimler-Benz AG, Sprachverarbeitung im Kraftfahrzeug |
| Juli 1987 bis Mai 1991 | Entwicklungsingenieur im Forschungszentrum der Alcatel SEL AG, Software- und Verfahrensentwicklung für automatische Spracherkennung |
| Juni 1991 bis Juni 1992 | Forschungsaufenthalt am International Computer Science Institute in Berkeley, Kalifornien, neuronale Netzwerke zur Geräuschreduktion |
| seit Juli 1992 | Entwicklungsingenieur und Projektleiter im Forschungszentrum der Alcatel SEL AG, Algorithmen und Software zur intelligenten Signalverarbeitung mit neuronalen Netzwerken in unterschiedlichen Anwendungsgebieten |

Michael Trompf, im Januar 1996