

DIPLOMARBEIT

Optimierung der Struktur Neuraler Netze zur Erkennung von Atemstillständen

ausgeführt am Institut für
Grundlagen und Theorie der Elektrotechnik
Abteilung für Bioelektrizität und Magnetismus
der Technischen Universität Wien

unter Anleitung von
Vertr. Ass. Dipl. Ing. Markus CLABIAN
und
ao. Univ. Prof. Dr. Helmut PFÜTZNER

durch
Thomas ZEMEN
Weidengasse 2
A-2345 Brunn am Gebirge
Matr.Nr. 8925585

Zusammenfassung

In dieser Arbeit werden grundlegende Untersuchungen zur Erkennung von Atemstillständen mit Neuralen Netzen (NNs) durchgeführt. Einführend wird die Messung der Sauerstoffsättigung des Blutes, der Herzfrequenz und des elektrischen Feldplethysmographie (EFPG)-Signals, mit dem Herz- und Lungentätigkeit erfaßt werden können, beschrieben. Darauf aufbauend werden verschiedene Vorverarbeitungsalgorithmen und ihre mathematischen Grundlagen im Detail vorgestellt. Dies sind für das EFPG-Signal die zeitabhängige Fourier-Transformation, die Cepstrum-Transformation (CT), das Verfahren des Linear Predictive Codings (LPC) und die Wavelet-Transformation. Weiters wird auch eine Kombination von LPC und CT vorgestellt (LPCC). Die resultierenden Vektoren werden in einem Zeitraster von einer Sekunde berechnet, standardisiert und mittels Hauptkomponentenanalyse in ihrer Dimensionalität reduziert.

Die charakteristischen Eigenschaften der verwendeten NNs und dafür geeignete Lernverfahren werden in einem weiteren Kapitel präsentiert. Die beschriebenen Topologien sind das Multi Layer Perceptron (MLP) und das Radial Basis Function Network (RBFN), dessen erste Schicht durch den Algorithmus der Self Organizing Map (SOM) bzw. der Learning Vector Quantization (LVQ) trainiert wird. Als Testverfahren für die NNs wird ein Kreuzvalidierungsalgorithmus verwendet und zur Verbesserung der Erkennungsrate die Bildung eines Komitees aus mehreren NNs untersucht.

Im empirischen Teil der Arbeit wird der am besten geeignete Vorverarbeitungsalgorithmus zur Berechnung der Vektoren für das nachfolgende NN ermittelt. Dazu wird in einem ersten Schritt eine Grobauswahl mittels Hauptkomponentenanalyse getroffen. Zur weiteren Selektion werden die Ergebnisse eines RBFN mit LVQ-Schicht herangezogen. Das LPCC-Verfahren wurde dabei als der am besten geeignete Vorverarbeitungsalgorithmus für das EFPG-Signal ausgewählt. Diverse Parameter dieses Algorithmus werden in Hinblick auf eine möglichst hohe Erkennungsrate optimiert.

Mit den nun festgelegten Eingangs- und Zielvektoren wird die Leistungsfähigkeit der verschiedenen NNs untersucht, und diverse Parameter, wie z.B. die Neuronenanzahl, werden optimiert. Weiters wird auch die für die Kreuzvalidierung notwendige Segmentierung des Datenmaterials im Detail untersucht.

Mit dem MLP wird für die Patientengruppe Säuglinge (Erwachsene) eine Erkennungsrate von $63.9\% \pm 4.1\%$ ($70.4\% \pm 2.1\%$) erzielt. Durch die Änderung der Segmentierung von Vektor- auf Ereignisbasis reduziert sich die Erkennungsrate auf $50.6\% \pm 8.1\%$ ($62.6\% \pm 4.4\%$). Die Werte für das RBFN mit LVQ-Schicht sind um ca. 4% reduziert. Durch die Verwendung eines Komitees läßt sich das Ergebnis um ca. 3% verbessern.

Am Schluß der Arbeit wird in einem kurzen Anhang ein Überblick über die für MATLAB 5 erstellte Software gegeben.

Summary

This thesis presents basic techniques to detect apnea events with neural networks (NNs). As an introduction, the measurement of the relevant vital parameters is discussed. These are the blood oxygen saturation, the heart rate and the electric field plethysmography (EFPG) signal, which is used to monitor both, the cardiac and the respiratory activity.

Various preprocessing algorithms for these signals and the necessary mathematical background are described in detail. The EFPG signal is processed by the time dependent Fourier transform, the cepstrum transform (CT), the linear predictive coding algorithm (LPC), the wavelet transform and by a combination of LPC and CT (LPCC). For the resulting feature vectors, a sampling rate of one second is used. Besides, statistic methods for standardisation and the principal component analysis for dimensionality reduction of vector sets are presented.

Different NNs and suitable training methods are described in a separate chapter. The discussed topologies are the multi layer perceptron (MLP) and the radial basis function network (RBFN). To train the first layer of the RBFN, the self organising map (SOM) and the learning vector quantization algorithm (LVQ) are used. To test the NNs, a cross validation algorithm is applied and committees of NNs are investigated to increase the recognition rate.

In the empirical part of this paper it is tried to choose an optimal subset of feature vectors resulting from the pre-processing. The feature vectors are selected by principal component analysis and by comparing the training results of a RBFN with LVQ layer. In this process the LPCC preprocessing algorithm is selected. Various parameters are optimised with regard to a high recognition rate.

The performance of different NN topologies trained with the chosen input and target vectors are investigated, and some parameters - e.g., the number of neurons - are optimised. The necessary segmentation of the data set for the cross validation algorithm is analysed in detail.

As the final result, the recognition rate achieved with the MLP for the infant (adult) patient group is $63.9\% \pm 4.1\%$ ($70.4\% \pm 2.1\%$). When the segmentation of the data set is changed from vector based to event based the recognition rate is reduced to $50.6\% \pm 8.1\%$ ($62.6\% \pm 4.4\%$). The results for the RBFN are approximately 4% lower. The use of a committee of NNs increases the recognition rate by approximately 3%.

A short appendix gives an overview over the written software for MATLAB 5.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Vitalparameter	3
1.1.1	EFPG-Signal	3
1.1.2	Herzfrequenz	4
1.1.3	Oxymetrie	4
1.2	Klassifikation der Meßdaten	4
1.3	Datenmaterial	7
1.3.1	Säuglinge	7
1.3.2	Erwachsene	7
2	Vorverarbeitung	9
2.1	EFPG-Signal	9
2.1.1	Zeitabhängige Fouriertransformation	9
2.1.2	Leistungsdichtespektrum	12
2.1.3	Cepstrum und Entfaltung	13
2.1.4	Linear Predictive Coding	16
2.1.5	Linear Predictive Coding Cepstrum	18
2.1.6	Wavelets	19
2.2	Oxymetriesignal	22
2.3	Herzfrequenzsignal	24
2.4	Standardisierung	24
2.5	Hauptkomponentenanalyse	25
2.6	Kodierung des Zielvektors	27
2.6.1	1-aus-C-Kodierung	27

2.6.2	Geglättete 1-aus-C-Kodierung	27
2.6.3	Anteil des Atemstillstandes am Zeitfenster	28
3	Neurales Netz	29
3.1	Netztopologie	29
3.1.1	Multi Layer Perceptron	29
3.1.2	Radial Basis Function Network	31
3.2	Testverfahren	36
3.2.1	Kreuzvalidierung	36
3.2.2	Mehrere Neurale Netze als Komitee	37
4	Empirische Auswertungen	38
4.1	Zusammenstellung des Datenmaterials	38
4.2	Anwendung der Hauptkomponentenanalyse	38
4.2.1	Selektion alternativer Feature-Vektoren	38
4.2.2	Lineare Separierbarkeit der Eingangsdaten	41
4.3	Segmentierung des Datenmaterials	42
4.4	Optimale Zusammensetzung des Eingangsvektors	44
4.5	Zusammensetzung der Zielvektoren	49
4.6	Struktur des Neuralen Netzes	50
4.6.1	Multi Layer Perceptron	50
4.6.2	Radial Basis Function Network	51
4.7	Verschiedene Kreuzvalidierungs-Strategien	54
	Literaturverzeichnis	62
	Anhang - Software	64
A.1	Installation	64
A.2	Typischer Ablauf eines Berechnungszyklus	64
A.3	Kommandos	65
A.4	Unterprogramme	65

Notation

Zur einheitlichen Darstellung mathematischer Ausdrücke wurde folgende Notation gewählt: Fettgedruckte Kleinbuchstaben, wie zum Beispiel \mathbf{x} , bezeichnen Vektoren, fettgedruckte Großbuchstaben, wie \mathbf{X} , bezeichnen Matrizen. Die Elemente des Vektors \mathbf{x} werden durch x_k dargestellt, für die Elemente (i, j) einer Matrix \mathbf{X} wird die Notation $(\mathbf{X})_{ij}$ eingeführt. Alle Vektoren sind Spaltenvektoren, Zeilenvektoren werden durch die Transposition \mathbf{x}^T markiert. Die euklidische Länge eines Vektors wird durch $\|\mathbf{x}\|$ ausgedrückt. Falls mehrere Samples ein- und desselben Vektors verwendet werden, wird dies durch den hochgestellten Index \mathbf{x}^m symbolisiert.

Tiefgestellte Indizes wie in d_k werden kursiv dargestellt, tiefgestellte Abkürzungen jedoch normal c_{ox} . Diesen Konventionen folgend (siehe auch [1]) werden im Rahmen dieser Arbeit die in Tab. 0.1 dargestellten Symbole verwendet. Die verwendeten Abkürzungen sind in Tab. 0.2 angeführt.

Da alle vorkommenden Signale digitaler Natur sind, wird bei der Notation nicht zwischen analogen Signalen $s(t)$ und digitalen Wertefolgen $s[n]$ unterschieden, da es zu keiner Verwechslung kommen kann.

Symbole

\mathbf{A}	Matrix der Eigenvektoren \mathbf{a}_k von \mathbf{S}
a_k	LPC-Koeffizienten
\mathbf{c}	Feature-Vektor
$c_{\text{ox}}(n)$	Oxymetriesignal
d_k	LPCC-Koeffizienten
$f_{\text{C}}(n)$	Herzfrequenzsignal
f_{R}	Atemfrequenz
L	Fensterlänge
l_k	Eigenwerte von \mathbf{S}
m	Index der Eingangs- und Zielvektoren
M	Anzahl der Eingangs- und Zielvektoren
N	Anzahl der Frequenzpunkte der FFT
$P_{xx}(n, k)$	Leistungsdichtespektrum
r	Zeitindex
R	Übereinstimmungs- bzw. Erkennungsrate
$s(n)$	EFPG-Signal
\mathbf{S}	Kovarianzmatrix
\mathbf{t}	Zielvektor
$\tilde{\mathbf{t}}(n)$	Klassifikationssignal (Transformierter Zielvektor)
T	Zeitraaster der Eingangsvektoren
\mathbf{v}	Hauptkomponentenvektor
ν_{s}	Abtastfrequenz des EFPG-Signals
ν_{ox}	Abtastfrequenz des Oxymetriesignals
ν_{C}	Abtastfrequenz des Herzfrequenzsignals
$w(n)$	Fensterfunktion im Zeitbereich
$W(k)$	Fensterfunktion im Frequenzbereich
\mathbf{x}	Eingangsvektor des Neuralen Netzes
$\mathbf{y}^{(k)}$	Aktivierungsvektor der verborgenen Schicht k
\mathbf{z}	Ausgangsvektor des Neuralen Netzes
$\tilde{\mathbf{z}}(n)$	Ausgangsklassifikation (Transformierter Ausgangsvektor)

Tab. 0.1: Im Rahmen dieser Arbeit verwendete mathematische Symbole.

Abkürzungen

CA	Zentrale Apnoe, <i>central apnea</i>
CH	Zentrale Hypopnoe, <i>central hypopnea</i>
CT	Cepstrum-Transformation, <i>cepstrum transform</i>
EFPG	Elektrische Feldplethysmographie, <i>electric field plethysmography</i>
EKG	Elektrokardiogramm, <i>electro cardiogramm</i>
FFT	Schnelle Fourier-Transformation, <i>fast fourier transform</i>
LPC	<i>Linear predictive coding</i>
LVQ	<i>Learning vector quantization</i>
MA	Gemischte Apnoe, <i>mixed apnea</i>
MH	Gemischte Hypopnoe, <i>mixed hypopnea</i>
MLP	Mehrschichtiges Perzeptron-Netz, <i>multi layer perceptron</i>
NB	Normalatmung, <i>normal breathing</i>
NN	Neurales Netz, <i>neural network</i>
OA	Obstruktive Apnoe, <i>obstructive apnea</i>
OH	Obstruktive Hypopnoe, <i>obstructive hypopnea</i>
PB	Periodische Atmung, <i>periodic breathing</i>
PC	Hauptkomponente, <i>principal component</i>
PCA	Hauptkomponentenanalyse, <i>principal component analysis</i>
PSD	Leistungsdichtespektrum, <i>power spectral density</i>
RBFN	<i>Radial basis function network</i>
SI	Seufzer, <i>sigh</i>
SOM	Selbstorganisierende Karte, <i>self organizing map</i>
TDFT	Zeitabhängige Fourier-Transformation, <i>time dependent fourier transform</i>
WT	Wavelet-Transformation, <i>wavelet transform</i>

Tab. 0.2: Im Rahmen dieser Arbeit verwendete Abkürzungen.

Kapitel 1

Einleitung

Zur Überwachung von Patienten mit Atemstörungen während des Schlafes werden in der medizinischen Praxis Verfahren verwendet, die einen großen apparativen Aufwand erfordern. Dies liegt in diesem Zusammenhang an der großen Anzahl an wichtigen Vitalparametern die registriert werden müssen, dazu gehören unter anderem Herz- und Atemfrequenz, Sauerstoffsättigung des Blutes sowie Airflow und Thorax- und Abdomenbewegung (Polysomnographie). Die Auswertung der Meßergebnisse ist wiederum äußerst personalaufwendig und wenig automatisiert. Dies schließt einen großflächigen Einsatz und eine Überwachung außerhalb des Krankenhauses aus.

Um ein Schlafmonitoring des Patienten auch zuhause durchführen zu können, bietet sich das Verfahren der elektrischen Feldplethysmographie (*electric field plethysmography*, EFPG) an. Ein elektrisches Feld, das auf den Thoraxraum einwirkt, ändert seine Konfiguration in Abhängigkeit von dessen Blut- und Luftinhalt. Mit nur zwei Doppelelektroden kann so ein EFPG-Signal registriert werden, das zugleich Information über die Atem- und Herztätigkeit beinhaltet. Zusätzlich kann über die Elektroden auch ein nicht standardisiertes EKG abgeleitet und daraus die Herzfrequenz bestimmt werden. Für dieses Verfahren wurde ein EFPG Home Monitoring System entwickelt, dessen genauer Aufbau in [2] nachzulesen ist. Dieses System registriert weiters die Sauerstoffsättigung des Blutes über einen Fingerclip.

Die drei somit verfügbaren Vitalparameter, das EFPG-Signal s , die Herzfrequenz f_C und das Oxymetriesignal c_{ox} müssen durch geeignete Algorithmen vorverarbeitet werden. Deren Ergebnisse, die so genannten Feature-Vektoren, dienen als Eingangsdatenmaterial für ein Neuronales Netz (*neural network*, NN). In Kapitel 1.1 werden die Details der einzelnen Signale erläutert, die Vorverarbeitungsalgorithmen und die damit berechneten Feature-Vektoren werden in Kapitel 2 vorgestellt. In Abb. 1.1 ist ein Überblick über die verwendete Signalverarbeitungs-Struktur zu sehen.

Der Vorverarbeitung folgt eine Hauptkomponentenanalyse (*principal component analysis*, PCA), um die Dimension der einzelnen Feature-Vektoren \mathbf{c}_i zu reduzieren (siehe Kapitel 2.5). So kann die relevante Information für das NN mit möglichst wenig Komponenten re-

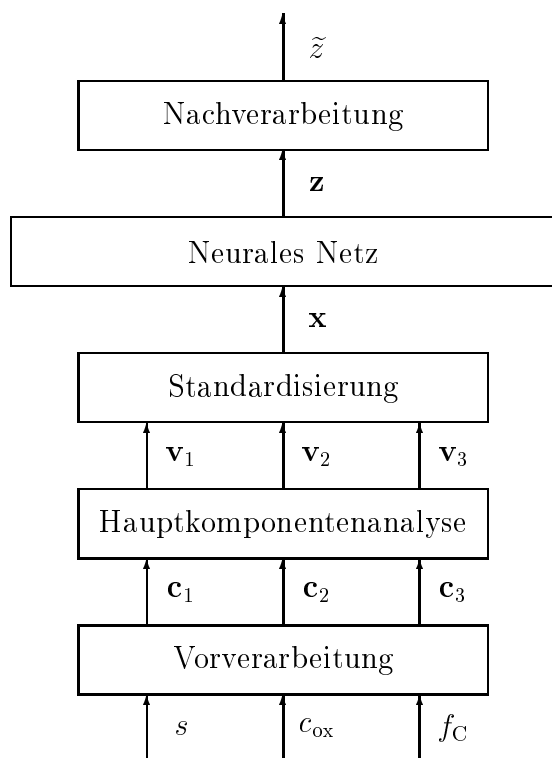


Abb. 1.1: Struktur der Verarbeitungsschritte und Bezeichnung der verwendeten Variablen. s EFPG-Signal, c_{ox} Oxymetriesignal, f_C Herzfrequenzsignal, \mathbf{c} Feature-Vektor, \mathbf{v} Hauptkomponentenvektor, \mathbf{x} Eingangsvektor, \mathbf{z} Ausgangsvektor, \tilde{z} Ausgangsklassifikation.

präsentiert werden. Die Hauptkomponenten \mathbf{v}_i werden weiters noch standardisiert (siehe Kapitel 2.4) und dann zum Eingangsvektor \mathbf{x} für das NN kombiniert.

Das NN soll diese Eingangsdaten den in Kapitel 1.2 beschriebenen Atemereignissen mit möglichst hoher Wahrscheinlichkeit richtig zuordnen. Das vorhandene Datenmaterial wurde von medizinischem Personal klassifiziert, um so anhand von Beispielen ein NN trainieren zu können. Diese Klassifikation ist aber leider aus verschiedenen Gründen, wie Zeitdruck, großer Datenmenge etc. oft fehlerhaft. Genauere Details über die Durchführung der Messungen und die Synchronisation des EFPG-Gerätes mit den bestehenden Polysomnographie-Systemen sind in [3] zu finden.

Der Gegenstand dieser Diplomarbeit ist die Auswahl einer geeigneten Vorverarbeitung der Meßdaten, sowie die Analyse verschiedener NN-Topologien (siehe Kapitel 4). Die dadurch zu maximierende Größe ist die Übereinstimmungsrate R der Klassifikation durch das NN mit der durch das medizinische Personal. Der Ausdruck Übereinstimmungsrate wurde gewählt, um zu verdeutlichen, daß die im Rahmen dieser Arbeit verwendete Zielklassifikation selbst nicht fehlerlos ist. Die durch das NN erzielte Übereinstimmungs- bzw. Erkennungsrate ist unter diesem Aspekt zu bewerten.

Alle Berechnungen im Rahmen dieser Arbeit wurden mit MATLAB 5 für Windows 95 durchgeführt. Die dafür erstellte Software ist im Anhang dokumentiert.

1.1 Vitalparameter

Folgende Signale des Patienten stehen für die Analyse zur Verfügung:

- Das Signal der elektrischen Feldplethysmographie s ,
- die Sauerstoffsättigung des Blutes c_{ox} und
- die Herzfrequenz f_C .

Im folgenden werden die Erfassung der Vitalparameter und deren charakteristische Eigenschaften genauer erläutert.

1.1.1 EFPG-Signal

Das elektrische Feldplethysmographie (EFPG)-Signal s erfaßt die Impedanzänderung des Thorax verursacht durch die Herz- und Lungentätigkeit und besteht somit aus zwei periodischen Komponenten mit zeitlich veränderlicher Amplitude und Frequenz. Die Impedanz vergrößert sich durch das Füllen der Lunge mit Luft (Inspiration) und durch die Kontraktion des Herzens. Die Herzkomponente wird durch die Lungenaktivität amplituden- und frequenzmoduliert (respiratorische Sinusarythmie, RSA).

Die beiden Komponenten des Signals s haben folgende grundlegende Eigenschaften:

- Die Grundfrequenz f_R der Atemkomponente liegt bei Normalatmung zwischen 0.15 Hz und 1.3 Hz ($9 \text{ min}^{-1} \dots 78 \text{ min}^{-1}$), bei Apnoe ist $f_R = 0$.
- Die Herzfrequenz f_C umfaßt einen Bereich von 0.75 Hz bis 2.8 Hz ($45 \text{ min}^{-1} \dots 168 \text{ min}^{-1}$).
- Das Verhältnis f_C/f_R bewegt sich von 2.5 bis 5.8, f_C ist also immer größer als f_R .
- Der Signalpegel der Herzkomponente liegt typischerweise 15 dB unterhalb des Pegels der Lungenkomponente.

Der Variationsbereich der angeführten Parameter resultiert aus den unterschiedlichen physiologischen Zuständen Schlaf, Ruhe und Belastung, sowie aus der Betrachtung von Säuglingen und Erwachsenen. Da sich die physiologischen Eigenschaften von Erwachsenen und Säuglingen stark unterscheiden, werden diese beiden Patientengruppen getrennt betrachtet. Genauere Details dazu sind in [4] zu finden.

1.1.2 Herzfrequenz

Über die Doppelelektroden zur Erfassung des EFPG-Signals wird gleichzeitig ein EKG abgeleitet. Aus diesem kann die Herzfrequenz f_C bestimmt werden, deren Registrierung aber vor allem in der Patientengruppe Säuglinge sehr unzuverlässig funktioniert. Das bedeutet, daß der registrierte Wert entweder falsch ist oder über längere Zeiträume fehlt. Daher wurde f_C aus dem EFPG-Signal s nachträglich für die fraglichen Signalabschnitte ermittelt (siehe [5]).

1.1.3 Oxymetrie

Die Messung der Sauerstoffsättigung des Blutes erfolgte nicht-invasiv mittels eines optischen Sensors am Finger. Bei Säuglingen wurde dieser Sensor am Fuß angebracht. Der Wertebereich für c_{ox} liegt zwischen 0 und 100 Prozent. Bei diesem Parameter ist durch den peripheren Meßort (am Finger, bzw. Unterschenkel) mit einer deutlichen Zeitverzögerung zu rechnen.

1.2 Klassifikation der Meßdaten

In Abb. 1.2 ist eine typische zentrale Apnoe (CA) mit den registrierten Signalen s , f_C , c_{ox} und dem dazu synchronen Klassifikationssignal \tilde{t} zu sehen. Der Zeitbereich der CA ist markiert, die Sauerstoffsättigung sinkt in diesem Beispiel stark verzögert erst nach dem Ende der CA ab.

Der Wert des Klassifikationssignals \tilde{t} gibt Auskunft über die Art des Atemstillstandes, die genaue Zuordnung ist in Tab. 1.1 angegeben.

In [3] ist eine ausführliche Definition aller angeführten Ereignistypen nachzulesen, hier soll nur ein kurzer Überblick über die im Rahmen dieser Arbeit interessanten Ereignisse

Wert	Klassifikation	Abkürzung
-2	Bewegung	–
0	Normalatmung	NB
1	Seufzer	SI
2	periodische Atmung	PB
4	zentrale Hypopnoe	CH
5	obstruktive Hypopnoe	OH
6	gemischte Hypopnoe	MH
8	zentrale Apnoe	CA
9	obstruktive Apnoe	OA
10	gemischte Apnoe	MA

Tab. 1.1: Bedeutung der Werte des Klassifikationssignals \tilde{t} .

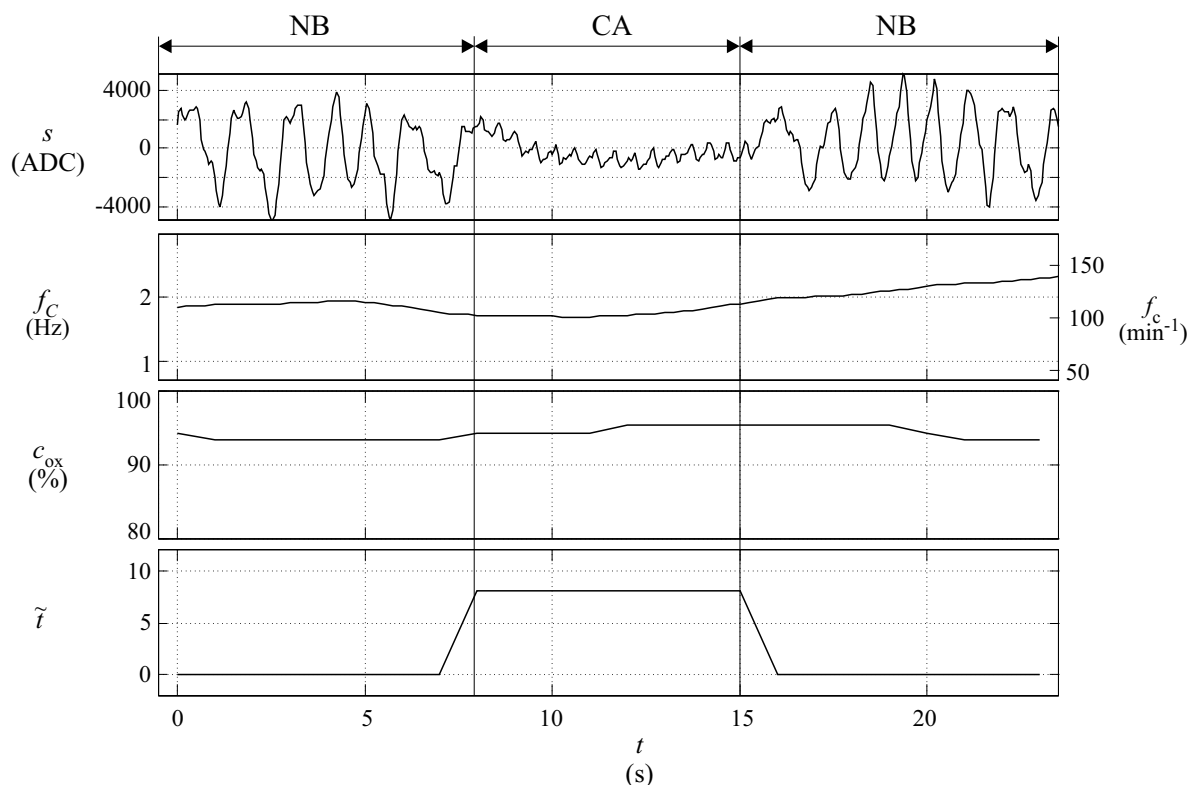


Abb. 1.2: Verlauf der registrierten Signale während einer zentralen Apnoe (CA), deren Zeitbereich markiert ist. Die Sauerstoffsättigung sinkt verzögert ab. s EFPG-Signal, f_C Herzfrequenzsignal, c_{ox} Oxymetriesignal, \tilde{t} Klassifikationssignal.

gegeben werden:

Seufzer (SI) - Spontaner tiefer Atemzug, gefolgt von einem Aussetzen der Atmung. Seufzer werden nur in der Patientengruppe Säuglinge registriert.

Zentrale Apnoe (CA) - Ausfall der Atmung, verursacht durch ein Aussetzen der Nervenimpulse aus dem Atemzentrum.

Obstruktive Apnoe (OA) - Ausfall der Atmung durch eine Blockierung der Atemwege, z.B. durch Erschlaffen der Rachenmuskulatur; die Impulse des Atemzentrums sind vorhanden. Thorax- und Abdomenbewegungen können weiterhin beobachtet werden, sie sind jedoch unrythmisch und meistens in ihrer Amplitude vermindert, es kann aber auch zu einer Verstärkung kommen.

Gemischte Apnoe (MA) - Kombination von CA und OA, wobei die Abfolge der Ereignisse beliebig sein kann, meist tritt aber zuerst OA und infolge dessen CA auf. MA wird im Rahmen dieser Arbeit nicht zum Trainieren des NN verwendet, sondern als Aneinanderreihung von OA und CA betrachtet.

Hypopnoe - Hypopnoen sind ebenfalls in drei Formen vertreten, als

- zentrale Hypopnoe (CH),
- obstruktive Hypopnoe (OH) und
- gemischte Hypopnoe (MH).

Im Gegensatz zur Apnoe ist der Airflow hier um mehr als 50 Prozent reduziert, verschwindet aber nicht völlig. Hypopnoen werden nur bei der Patientengruppe Erwachsene ausgewiesen und wegen des geringen Unterschiedes zu den Apnoen im Rahmen dieser Arbeit mit diesen zusammengefaßt. MH werden ebenfalls wie MA nicht zum Training des NN verwendet.

In Abb. 1.3 wird zur Illustration ein typisches Beispiel eines Seufzers einer zentralen Apnoe gegenübergestellt. Es ist zu erwarten, daß die Unterscheidung zwischen den verschiedenen Apnoetypen schwieriger sein wird als zwischen Atemstillstand und Normalatmung.

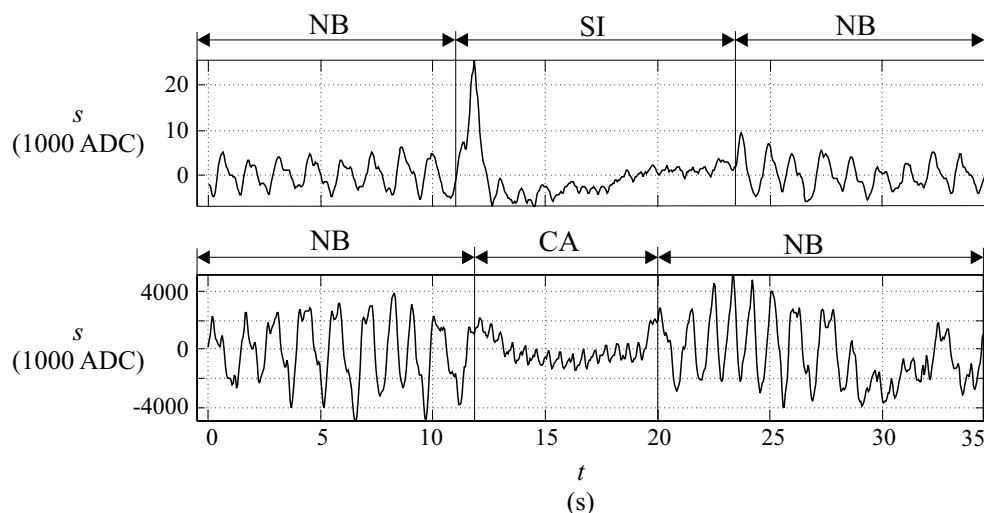


Abb. 1.3: Gegenüberstellung des EFPG-Signals eines Seufzers (SI) und einer zentralen Apnoe (CA).

1.3 Datenmaterial

Das zur Verfügung stehende Datenmaterial stammt von folgenden zwei Patientengruppen:

- 30 Säuglingen aus der Frühgeborenenstation des Landeskrankenhauses Mödling.
- 50 Erwachsenen aus dem Schlaflabor des Krankenhauses Lainz.

Die genauen Details der Meßdatenerfassung sind in [3] ausführlich dargestellt.

1.3.1 Säuglinge

Die Anzahl der registrierten Ereignisse pro Ereignistyp sind für die Patientengruppe Säuglinge in Tab. 1.2 dargestellt. Von diesen ist aber nur ein Teil verwendbar, da folgende Randbedingungen erfüllt werden müssen:

- Störungsfreie Messung der Oxymetrie c_{ox} und der Herzfrequenz f_C vom Anfang des Ereignisses bis 35 Sekunden danach.
- Vor und nach dem Ereignis dürfen in einem Zeitraum, der gleich lange wie das Ereignis selbst ist, weder Bewegung noch ein anderes Ereignis stattfinden.

Wie in Tab. 1.2 ersichtlich stehen nur sehr wenige OA zur Verfügung. Diese sind ungleichmäßig über die einzelnen Patienten verteilt, wie Abb. 1.4 illustriert: 76 Prozent der verwendbaren OA stammen aus dem Datenmaterial von nur drei Patienten. Um eine möglichst hohe Erkennungsrate durch ein NN zu erreichen, ist eine Gleichverteilung der Ereignisse pro Patient und pro Ereignistyp anzustreben. Dies ist mit dem zur Verfügung stehenden Datenmaterial jedoch nicht möglich. Die in dieser Arbeit präsentierten Ergebnisse können folglich durch weiteres Datenmaterial noch verbessert werden.

1.3.2 Erwachsene

Auf die in der Patientengruppe Erwachsene vorhandenen Ereignisse wurden die gleichen Selektionskriterien angewandt, wie schon im vorhergehenden Abschnitt für die Patientengruppe Säuglinge beschrieben. Die Gesamtanzahl und die davon verwendbaren Ereignisse sind in

Ereignistyp	Anzahl	davon verwendbar
Seufzer (SI)	481	282
zentrale Apnoe (CA)	1419	866
obstruktive Apnoe (OA)	161	70

Tab. 1.2: Anzahl der Ereignisse in der Patientengruppe Säuglinge.

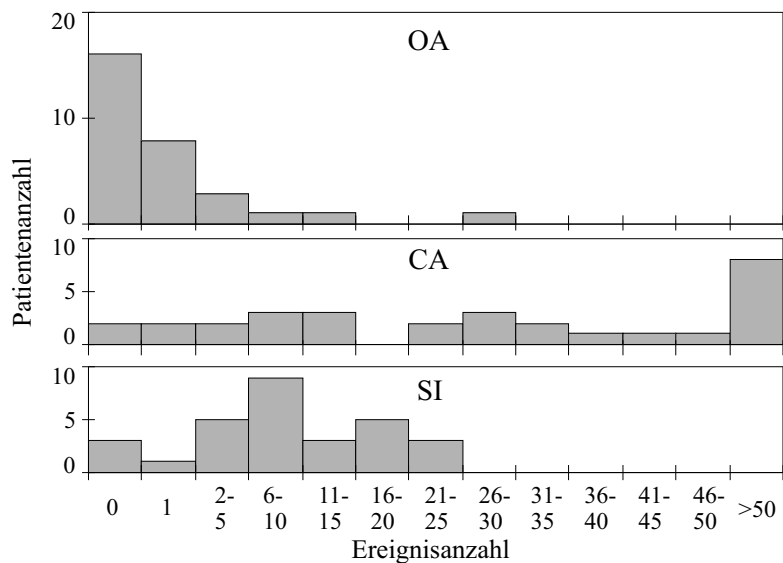


Abb. 1.4: Die Anzahl der Ereignisse pro Patient und Ereignistyp in der Patientengruppe Säuglinge ist sehr ungleichmäßig verteilt. Besonders die Verteilung der OA ist problematisch, da bei 16 Patienten keine OA auftritt, bei neun nur eine und bei einem Patienten 30 OA.

Tab. 1.3 angeführt. Auch hier sind die OA stark unterrepräsentiert, weiteres Datenmaterial wäre auch in dieser Patientengruppe von Vorteil. Die CH und OH werden im Rahmen dieser Arbeit nicht berücksichtigt.

Ereignistyp	Anzahl	davon verwendbar
zentrale Hypopnoe (CH)	104	54
obstruktive Hypopnoe (OH)	997	260
zentrale Apnoe (CA)	633	351
obstruktive Apnoe (OA)	697	123

Tab. 1.3: Anzahl der Ereignisse in der Patientengruppe Erwachsene.

Kapitel 2

Vorverarbeitung

Die zur Verfügung stehenden Vitalparameter s , c_{ox} und f_C beschreiben den aktuellen Zustand der Lungen- und Herztätigkeit des Patienten. Die daraus resultierende Diagnose des aktuellen Atmungsereignisses des Patienten wird durch \tilde{t} klassifiziert. Es können daher s , c_{ox} und f_C als die Eingangsvariablen einer Transformation und \tilde{t} als deren Ausgangsvariable betrachtet werden.

Es ist nun von großem Vorteil, diese Transformation nicht komplett von einem Neuronen Netz (NN) durchführen zu lassen, sondern die Meßdaten einer Vorverarbeitung bzw. Feature Extraction zu unterziehen. Dadurch kann eine Invarianz gegenüber bestimmten Parametern der Eingangsdaten, wie Zeit, Amplitude oder Frequenz, erreicht werden. Weiters ist es so auch möglich, schon bekanntes Wissen über den Mustererkennungsprozeß einzubringen.

Durch die Vorverarbeitung soll auch die Anzahl der Eingangsvariablen, die zur Verarbeitung durch ein NN als Vektor aufgefaßt werden, möglichst gering gehalten werden. Das ist notwendig, da die Menge an erforderlichen Trainingsvektoren linear mit der Dimension des Eingangsvektors steigt, wie in [1] ausführlich dargestellt ist.

Alle im folgenden vorgestellten Algorithmen stellen einen systematischen, mathematischen Zugang zur Berechnung verschiedener Feature-Vektoren aus den Meßsignalen und somit zur Erkennung der Atemstillstände dar.

2.1 EFPG-Signal

2.1.1 Zeitabhängige Fouriertransformation

Das Signalerzeugungsmodell des EFPG-Signals in Kapitel 1.1.1 führt zu dem Schluß, daß vor allem die kurzfristige Änderung der Spektralanteile von s interessant ist. Deswegen ist es notwendig, die FFT nur auf kurze Ausschnitte des Signals anzuwenden, um

so die zeitliche Veränderung der Frequenzkomponenten erfassen zu können. Dies führt zur zeitabhängigen Fouriertransformation (TDFT, time dependent fourier transform). Der Signalausschnitt wird durch eine Fensterfunktion w der Länge L bestimmt, eine einfache Form stellt die Rechteckfunktion

$$w_1(n) = \begin{cases} 1 & 0 \leq n \leq L, \\ 0 & \text{sonst} \end{cases} \quad (2.1)$$

dar. Diese besitzt die diskrete Frequenzdarstellung

$$W_1(k) = e^{-j\pi kL/N} \frac{\sin[\pi k(L+1)/N]}{\sin(\pi k/N)}, \quad (2.2)$$

in der N die Anzahl der Frequenzpunkte angibt.

Die Multiplikation eines allgemeinen Signals x mit der Fensterfunktion w im Zeitbereich korrespondiert im Frequenzbereich mit einer periodischen Faltung

$$X(n, k) = \sum_{m=0}^{L-1} x(n+m) w(m) e^{-j(2\pi/N)km}. \quad (2.3)$$

Um die dadurch entstehende Beeinflussung durch Frequency Leakage zu reduzieren, verwendet man statt (2.1) eine Hamming-Funktion

$$w(n) = \begin{cases} 0.54 - 0.46 \cos(2\pi n/L) & 0 \leq n \leq L, \\ 0 & \text{sonst} \end{cases}. \quad (2.4)$$

Eine ausführliche Darstellung der Problematik ist in [6] und [4] zu finden. In Abb. 2.1 ist eine hochaufgelöste TDFT einer zentralen Apnoe (CA) dargestellt. Die strichlierte Linie markiert die Herzfrequenz, die unterbrochene punktierte Linie kennzeichnet die Atemfrequenz.

Die Parameterwahl der TDFT für das EFPG-Signal s stellt einen Kompromiß zwischen einer hohen Zeit- und Frequenzauflösung sowie einer notwendigen Beschränkung der Datenmenge dar. Folgende Kriterien werden daher angewendet:

- Die Länge des mit der Hammingfunktion gewichteten Zeitfensters wird durch die untere Grenzfrequenz und die Abtastfrequenz bestimmt, in unserem Fall sind das $f_{R_{\min}} = 0.16 \text{ Hz} = 9.6 \text{ min}^{-1}$ und $\nu_s = 20 \text{ Hz}$. Daraus folgt für die Fensterlänge

$$L = \nu_s / f_{R_{\min}} \approx 120. \quad (2.5)$$

- Um die FFT-Vektoren in einem Zeitraster von $T = 1 \text{ s}$ zu erhalten, wird das Fenster jeweils um $T\nu_s = 20$ Abtastwerte verschoben.

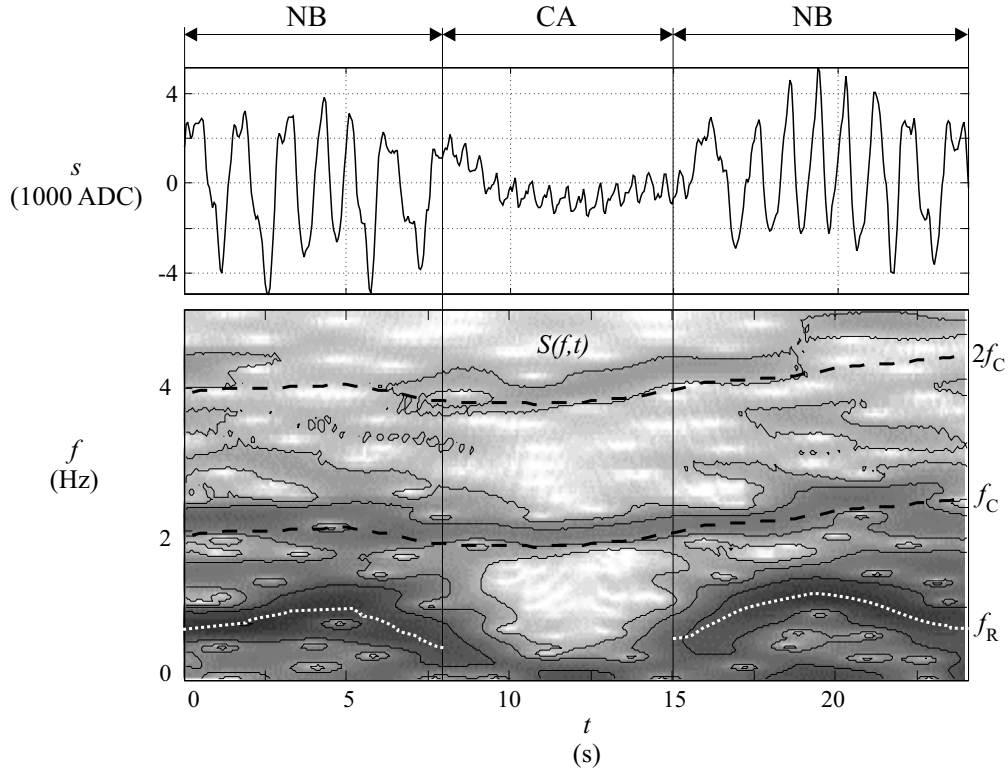


Abb. 2.1: Hochaufgelöste TDFT einer zentralen Apnoe; 1 Punkt Fensterverschiebung, 120 Punkte Fensterlänge, 256 Frequenzpunkte. Die während des Atemstillstandes verschwindende Atemkomponente ist deutlich zu erkennen. $f_C(t)$ ist zur Verdeutlichung strichliert, $f_R(t)$ punktiert eingezeichnet.

- Die aus der TDFT resultierenden Frequenzkomponenten werden zeitlich der Mitte der Fensterfunktion zugeordnet. Die Anzahl der Frequenzkomponenten wurde mit $N = 64$ gewählt, wobei aus dem Frequenzbereich von 0 Hz bis 10 Hz nur jene bis 5 Hz von Interesse sind, das sind die ersten $N' = 33$ Komponenten.

Die TDFT mit den angegebenen Eigenschaften ergibt sich zu

$$S(r\nu_s T, k) = \sum_{m=0}^{L-1} s(r\nu_s T - L/2 + m) w(m) e^{-j(2\pi/N)km} \quad (2.6)$$

für $-\infty < r < \infty$, $0 \leq k \leq N - 1$. r stellt dabei einen dimensionslosen ganzzahligen Zeitindex dar. Eine Vektordarstellung in dB wird durch den Feature-Vektor

$$\mathbf{c}_{\text{TDFT}}(r) = 20 * \log \begin{pmatrix} S(r\nu_s T, 1) \\ S(r\nu_s T, 2) \\ \vdots \\ S(r\nu_s T, N') \end{pmatrix} \quad (2.7)$$

definiert. Die Komponentenanzahl dieses Feature-Vektors ist für ein NN sehr hoch, durch eine Hauptkomponentenanalyse kann diese aber reduziert werden, wie in Kapitel 4.2.1 gezeigt werden wird.

Alle folgenden Vorverarbeitungsalgorithmen verwenden, falls nicht explizit angeführt, die selben Fensterparameter wie die TDFT und einen konstanten Zeitraster von einer Sekunde. In Abb. 2.2 ist die TDFT einer zentralen Apnoe mit den eben beschriebenen Parametern dargestellt. Die charakteristischen Eigenschaften sind zwar immer noch erkennbar, jedoch deutlich schlechter als in der hochauflösten TDFT in Abb. 2.1.

2.1.2 Leistungsdichtespektrum

Ein Atemstillstand stellt eine kurzfristige Veränderung eines stationären Zustandes des EFPG-Signales dar. Durch die Bildung des Verhältnisses zwischen zwei Schätzungen des Leistungsdichtespektrums (PSD, power spectral density) über verschieden lange Zeiträume kann diese Veränderung hervorgehoben werden.

Das Leistungsdichtespektrum kann durch die Berechnung des modifizierten Periodogrammes geschätzt werden. Es wird der gewichtete Mittelwert der FFT über mehrere mit der Fensterfunktion bewertete Signalabschnitte gebildet. Das PSD

$$P_{xx}(r\nu_s T, k) = \frac{\sum_{m=1}^{\frac{D\nu_s}{L}} S(r\nu_s T - mL, k)^2}{\left(\sum_{l=1}^L w(l)^2\right) \frac{D\nu_s}{L}} \quad (2.8)$$

wurde durch Mittelwertbildung über einen Zeitraum von $D = 300$ s ermittelt. Als Ausdruck

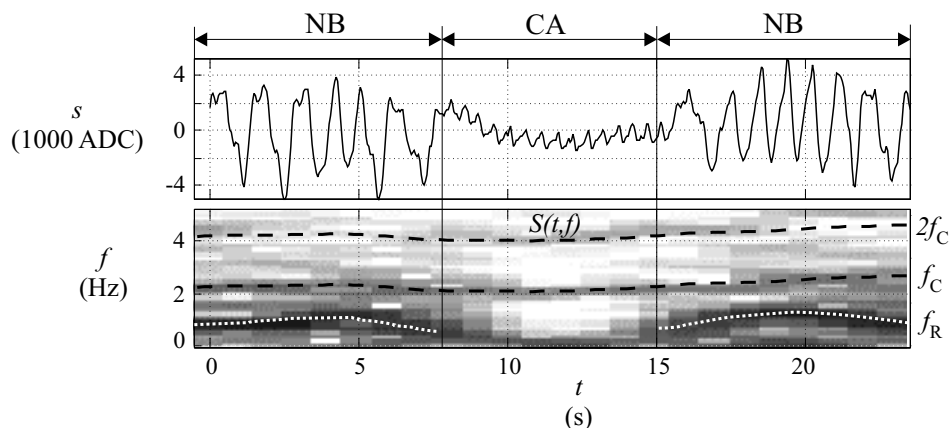


Abb. 2.2: TDFT einer zentralen Apnoe; Fensterverschiebung 20 Punkte, Fensterlänge 120 Punkte, 33 Frequenzpunkte. Die Auflösung ist im Vergleich zu Abb. 2.1 deutlich reduziert, die charakteristischen Details sind jedoch immer noch ausreichend zu erkennen.

für die kurzfristige Änderung des Signals und zum Vergleich mit (2.8) dient

$$P_{xx}^s(r\nu_s T, k) = \frac{S(r\nu_s T, k)^2}{\sum_{l=1}^L w(l)^2}. \quad (2.9)$$

Eine Darstellung in Vektorform des langfristigen PSD ergibt

$$\mathbf{p}_{xx}(r\nu_s T) = 10 * \log \begin{pmatrix} P_{xx}(r\nu_s T, 1) \\ P_{xx}(r\nu_s T, 2) \\ \vdots \\ P_{xx}(r\nu_s T, N^l) \end{pmatrix}, \quad (2.10)$$

das kurzfristige PSD P_{xx}^s wird analog in Vektorform dargestellt.

Als Feature-Vektoren wurden zwei Varianten definiert, und zwar

$$\mathbf{c}_{\text{PSD1}}(r) = \mathbf{p}_{xx}(r\nu_s T) - \mathbf{p}_{xx}^s(r\nu_s T) \quad (2.11)$$

sowie

$$\mathbf{c}_{\text{PSD2}}(r) = \begin{pmatrix} \mathbf{p}_{xx}(r\nu_s T) \\ \mathbf{p}_{xx}^s(r\nu_s T) \end{pmatrix}. \quad (2.12)$$

Bei \mathbf{c}_{PSD2} können die beiden Leistungsdichtespektren durch das NN kombiniert werden.

2.1.3 Cepstrum und Entfaltung

Die Cepstrum-Transformation (CT, cepstrum transform) bietet die Möglichkeit, zwei Signale, die durch eine Faltungsoperation kombiniert wurden, wieder zu trennen. Die Anwendung auf das EFPG-Signal kann durch folgendes vereinfachtes Modell motiviert werden: Herz und Lunge erzeugen durch ihre Tätigkeit zwei getrennte Signale, die als voneinander unabhängig angenommen werden. Der sie umgebende Thoraxraum wirkt als Filter, das von der Summe beider Signale durchlaufen wird und an dessen Ausgang, der Hautoberfläche, das endgültige EFPG-Signal s registriert werden kann. Durch die Cepstrum-Transformation kann die Faltung des Summensignals aus Herz- und Lungenanteil mit der Impulsantwort des Filters (dem Thoraxraum) rückgängig gemacht werden. Weiters besteht durch das Cepstrum auch die Möglichkeit der frequenzinvarianten Filterung eines Signals, da es im Zeitbereich ähnlich wie die Fourier-Transformation im Frequenzbereich behandelt werden kann.

Die Grundlage für die Entfaltung zweier Signale bildet die Theorie homomorpher Systeme, die im folgenden kurz vorgestellt wird. Weitere Details sind in [6] nachzulesen. Ein lineares System, das durch die lineare Transformation G dargestellt wird, erfüllt für zwei beliebige Eingangssignale $x_1(n)$ und $x_2(n)$ sowie für einen beliebigen Skalar c die Bedingungen

$$G\{x_1(n) + x_2(n)\} = G\{x_1(n)\} + G\{x_2(n)\} \quad (2.13)$$

und

$$G\{cx_1(n)\} = cG\{x_1(n)\}. \quad (2.14)$$

Eine Verallgemeinerung dieses Prinzips stellen homomorphe Systeme dar. Hier wird die allgemeine Kombination zweier Eingänge mit dem Operator \oplus und die Kombination eines Einganges mit einem Skalar durch \odot dargestellt. Die allgemeine Kombination zweier Ausgänge wird durch \square , die Kombination eines Ausganges mit einem Skalar durch \circ symbolisiert entsprechend

$$H\{x_1(n) \oplus x_2(n)\} = H\{x_1(n)\} \square H\{x_2(n)\} \quad (2.15)$$

und

$$H\{c \odot x_1(n)\} = c \circ H\{x_1(n)\}. \quad (2.16)$$

Im Fall eines linearen Systems wie in (2.13) und (2.14) entsprechen \oplus und \square der Addition, \odot und \circ der Multiplikation.

Ein allgemeines homomorphes System wie in Abb. 2.3 kann auch in kanonischer Form durch Kaskadierung dreier homomorpher Systeme dargestellt werden. Das erste ist nur von der Eingangsoperation und das letzte nur von der Ausgangsoperation abhängig. Das mittlere System ist linear. In Abb. 2.4 wird die kanonische Darstellung für den Faltungsoperator gezeigt.

Das komplexe Cepstrum einer Folge $x(n)$ ist definiert als die stabile Folge $\hat{x}(n)$, deren z -Transformation durch

$$\hat{X}(z) = \log(X(z)) \quad (2.17)$$

gegeben ist. Wenn für den Operator D_* das komplexe Cepstrum gewählt wird, geht die Faltung zweier Signale

$$x(n) = x_1(n) * x_2(n) \quad (2.18)$$

in die Summe

$$\hat{x}(n) = \hat{x}_1(n) + \hat{x}_2(n) \quad (2.19)$$

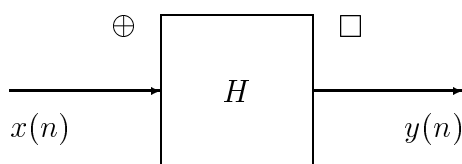


Abb. 2.3: Homomorphes System mit Eingangsverknüpfung \oplus , Ausgangsverknüpfung \square und Transformation H .

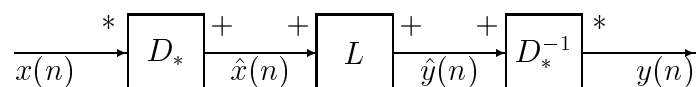


Abb. 2.4: Kanonische Form eines homomorphen Systems mit der Faltung $*$ als Eingangs- und Ausgangsoperator.

über. Diese Transformation erfolgt über folgende Zwischenschritte: Die Faltung zweier Signale entspricht im Frequenzbereich einer Multiplikation

$$X(z) = X_1(z)X_2(z). \tag{2.20}$$

Der Logarithmus eines Produktes läßt sich weiters als die Summe der Logarithmen der beiden Faktoren schreiben.

$$\hat{X}(z) = \log(X(z)) = \hat{X}_1(z) + \hat{X}_2(z) \tag{2.21}$$

Dies ist eine lineare Operation, die auch nach der Rücktransformation in den Zeitbereich durch

$$\hat{x}(n) = \frac{1}{2\pi j} \oint_C \log [X(z)] z^{n-1} dz \tag{2.22}$$

erhalten bleibt und zu (2.19) führt.

Die Voraussetzung für eine mögliche Entfaltung besteht in sich nicht überlappenden Bereichen des Trägers der Folgen $\hat{x}_1(n)$ und $\hat{x}_2(n)$. Der Träger einer Folge $\hat{x}(n)$ ist jener Bereich von n , in dem $\hat{x}(n)$ ungleich Null ist. Wenn zum Beispiel

$$\hat{x}_1(n) = 0 \quad \text{für } n \geq n_0 \tag{2.23}$$

und

$$\hat{x}_2(n) = 0 \quad \text{für } n < n_0 \tag{2.24}$$

gilt, ist $\hat{x}(n)$ durch Multiplikation mit einer Funktion

$$l(n) = \begin{cases} 1 & n < n_0, \\ 0 & n \geq n_0 \end{cases} \tag{2.25}$$

einer frequenzunabhängigen Filterung zugänglich. Für den Operator L in Abb. 2.4 wird in diesem Fall die Multiplikation mit der Funktion $l(n)$ gewählt, entsprechend

$$\hat{y}(n) = L[\hat{x}(n)] = \hat{x}(n)l(n). \tag{2.26}$$

Daraus folgt

$$\hat{y}(n) = \hat{x}_1(n) \tag{2.27}$$

und nach der Rücktransformation durch D_*^{-1}

$$y(n) = x_1(n). \quad (2.28)$$

Durch die Umkehrung der Faltungsoperation können aus dem Ausgangssignal eines Filters $x(n)$ das Eingangssignal $x_1(n)$ und die Impulsantwort des Filters $x_2(n)$ ermittelt werden.

Im konkreten Anwendungsfall der Erkennung von Atemstillständen wird die CT, ebenso wie die FFT, auf kurze, mit einer Fensterfunktion bewertete Signalabschnitte angewendet. Dies führt zur zeitabhängigen CT. Den Ausgangspunkt für deren Berechnung stellt die TDFT aus (2.6)

$$S(r\nu_s T, k) = \sum_{m=0}^{L-1} s(r\nu_s T - L/2 + m)w(m)e^{-j(2\pi/L)km}$$

dar. Die Berechnung der CT durch

$$\hat{s}(r\nu_s T, k) \simeq \frac{1}{L} \sum_{l=0}^{L-1} \hat{S}(r\nu_s T, l)e^{j(2\pi/L)l(r\nu_s T - L/2 + k)} \quad (2.29)$$

mit

$$\hat{S}(r\nu_s T, k) = \log(S(r\nu_s T, k)) \quad (2.30)$$

ist zwar nur eine Näherung, da \hat{s} noch durch Zeit-Aliasing beeinträchtigt ist. In der praktischen Anwendung mit genügend hoher Anzahl an Frequenzpunkten L (hier gleich der Fensterlänge) kann dies aber vernachlässigt werden, da das Cepstrum stärker als exponentiell abnimmt und somit im Überlappungsbereich schon sehr kleine Werte annimmt (siehe [6]).

Die Dimension eines damit definierbaren Feature-Vektors

$$\mathbf{c}_{CT}(r) = \begin{pmatrix} \hat{s}(r\nu_s T, 1) \\ \hat{s}(r\nu_s T, 2) \\ \vdots \\ \hat{s}(r\nu_s T, L) \end{pmatrix} \quad (2.31)$$

ist gleich der Fensterlänge L . Die große Komponentenanzahl dieses Feature-Vektors ist ein gravierender Nachteil im Zusammenhang mit NNs, er kann jedoch bei Kombination mit Linear Predictive Coding (LPC) entschärft werden (siehe Kapitel 2.1.5).

2.1.4 Linear Predictive Coding

Das Verfahren des Linear Predictive Coding (LPC) modelliert ein Signal als Impulsantwort eines Filters ohne Nullstellen. Dazu wird angenommen, daß jeder Abtastwert eines Signals

$x(n)$ als Linearkombination der vorhergegangenen p Abtastwerte dargestellt werden kann und die Koeffizienten im betrachteten Intervall konstant bleiben:

$$x(n) = \sum_{k=1}^p a_k x(n-k). \quad (2.32)$$

Die Übertragungsfunktion $H(z)$ eines Filters ohne Nullstellen läßt sich durch

$$H(z) = \frac{b_0}{\sum_{k=0}^p a_k z^{p-k}}; \quad a_0 = 1 \quad (2.33)$$

darstellen.

Die Nullstellen des Nennerpolynoms von (2.33) stellen die Polstellen der Übertragungsfunktion des Filters (2.33) dar. Die Koeffizienten a_k werden durch die Levinson-Durbin Rekursion berechnet, diese ist in [7], [8] und [9] ausführlich beschrieben. Dieser Algorithmus minimiert den quadratischen Fehler des durch die a_k approximierten Signals. Dadurch werden bei kleinem p nur die stärksten Komponenten des Signals durch die Koeffizienten a_k dargestellt, erst bei Erhöhung von p wird auch die Feinstruktur des Signals berücksichtigt. b_0 repräsentiert die Amplitude des Signals.

Das EFPG-Signal setzt sich, wie schon erwähnt, aus einem Lungen- und einem Herzanteil zusammen, wobei der Herzanteil typischerweise 15 dB schwächer als der Lungenanteil ist. Wie aus Abb. 2.5 ersichtlich, reicht schon eine Ordnung von $p = 6$ zur Approximation des Lungenanteils aus. Ab $p = 12$ wird der Herzanteil gut repräsentiert, ab $p = 18$ auch die erste Oberschwingung der Herzfrequenz.

Die erwähnte Einschränkung auf konstante Koeffizienten a_k für einen Signalabschnitt der Länge L stellt beim EFPG-Signal nur eine grobe Näherung dar. Versuche mit Adaptive Linear Predictive Coding (ADLPC) und diversen statistischen Abschätzungen der Modellparameter brachten aber keinen erkennbaren Vorteil und werden daher nicht näher ausgeführt, da bei diesen Algorithmen auch die benötigte Rechenleistung enorm anwächst.

Die Wahl der Ordnung p hat großen Einfluß auf die Erkennungsrate eines nachfolgenden NN. Ab einer bestimmten Ordnung p verschlechtert sich diese wieder, da dann auch die Oberwellen und diverse Artefakte des Signals in den Koeffizienten repräsentiert werden. Die optimale Ordnung p kann nur experimentell ermittelt werden (siehe Kapitel 4).

Durch die Verschiebung des Fensters um $r\nu_s T$ entstehen zeitabhängige LPC-Koeffizienten $a_k(r\nu_s T)$. Die p Koeffizienten $a_1(r\nu_s T), \dots, a_p(r\nu_s T)$ und $b_0(r\nu_s T)$ bilden den Vektor

$$\mathbf{c}_{\text{LPC}}(r) = \begin{pmatrix} a_1(r\nu_s T) \\ \vdots \\ a_p(r\nu_s T) \\ b_0(r\nu_s T) \end{pmatrix} \quad (2.34)$$

der Dimension $p+1$, der als Feature-Vektor für das NN verwendet wird (a_0 ist immer gleich eins und beinhaltet daher keine Information).

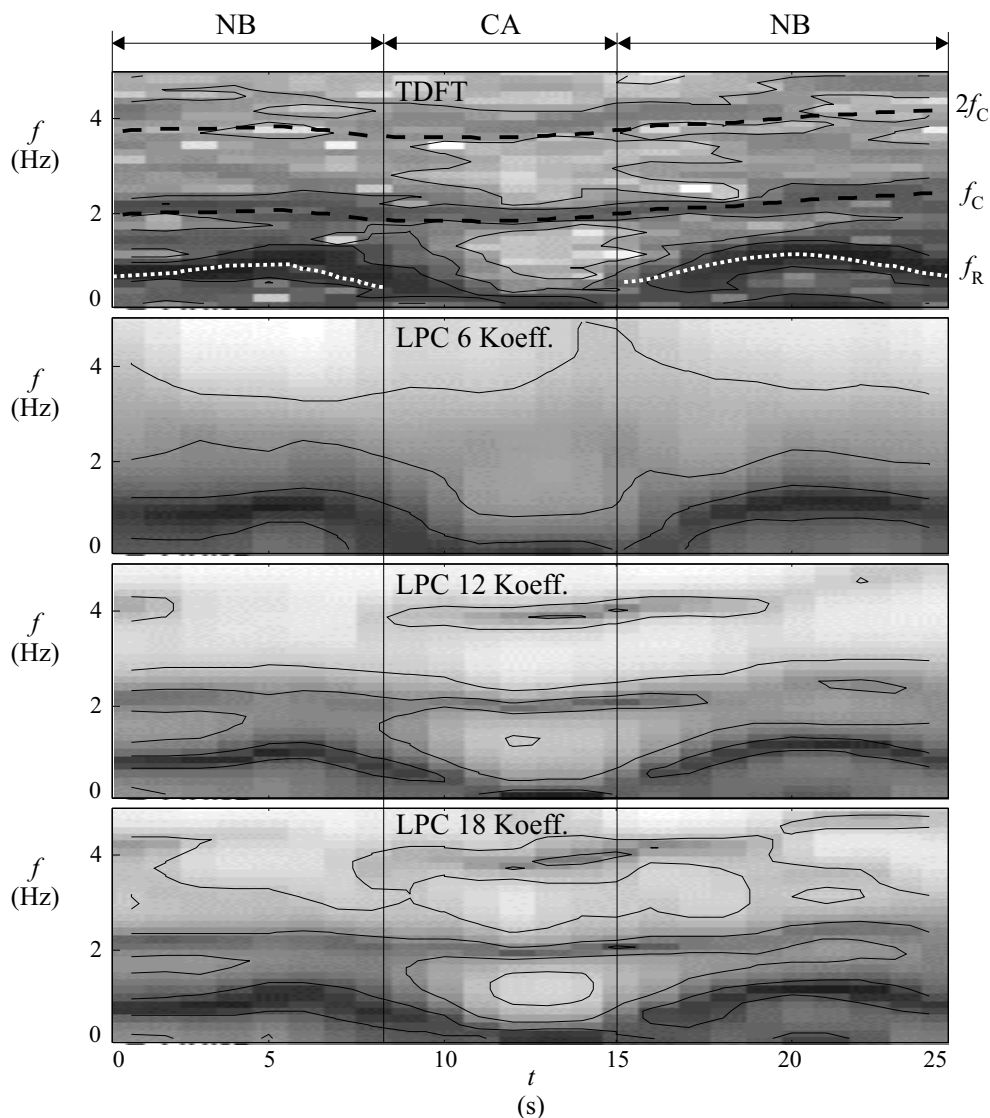


Abb. 2.5: Zeit-Frequenzdarstellungen eines Atemstillstandes repräsentiert durch 6, 12 bzw. 18 LPC-Koeffizienten. Die Approximation des Signals verbessert sich deutlich mit steigender Koeffizientenanzahl.

2.1.5 Linear Predictive Coding Cepstrum

Eine Kombination der Cepstrum-Transformation und des Linear Predictive Coding stellt das Linear Predictive Coding Cepstrum (LPCC) dar. Aus den LPC-Koeffizienten a_k können durch eine einfache rekursive Formel die Cepstrum-Koeffizienten d_k ermittelt werden. In [10] ist zusätzlich zu der rekursiven Berechnung auch eine direkte Berechnungsmethode angegeben, die aber im Rahmen dieser Arbeit nicht verwendet wird.

Die Cepstrum-Koeffizienten eines Filters ohne Nullstellen laut (2.33) werden definiert

als

$$\ln(H(z)) = \sum_{n=1}^{\infty} d_n z^{-n}. \quad (2.35)$$

Daraus ergibt sich der rekursive Zusammenhang:

$$d_1 = -a_1, \quad (2.36)$$

$$d_n = -a_n - \frac{1}{n} \sum_{k=1}^{n-1} k d_k a_{n-k} \quad \text{für } n = 2 \dots p. \quad (2.37)$$

Durch einfaches Umstellen der Gleichung (2.37) ergibt sich die Umkehrung:

$$a_1 = -d_1, \quad a_0 = 1, \quad (2.38)$$

$$a_n = -d_n - \frac{1}{n} \sum_{k=1}^{n-1} k d_k a_{n-k} \quad \text{für } n = 2 \dots p. \quad (2.39)$$

Aus den p zeitabhängigen LPC-Koeffizienten entstehen ebenfalls zeitabhängige LPCC-Koeffizienten $d_1(r\nu_s T), \dots, d_p(r\nu_s T)$. Diese liefern einen Feature-Vektor

$$\mathbf{c}_{\text{LPCC}}(r) = \begin{pmatrix} d_1(r\nu_s T) \\ \vdots \\ d_p(r\nu_s T) \\ b_0(r\nu_s T) \end{pmatrix} = \text{LPCC}\{s(r\nu_s T), p\} \quad (2.40)$$

für das NN. b_0 repräsentiert hier wie beim LPC ebenfalls die Amplitude des Signals.

Der Vorteil des LPCC besteht in der kleinen Anzahl der Koeffizienten des Feature-Vektors und den zusätzlichen Eigenschaften einer Cepstrumdarstellung. Diese sind die Möglichkeit der frequenzinvarianten Filterung und der Entfaltung bei entsprechendem Träger (siehe Kapitel 2.1.3) durch den Lernalgorithmus eines NN bzw. einer gewichteten linearen Kombination der Komponenten des Feature-Vektors durch die Hauptkomponentenanalyse.

2.1.6 Wavelets

Die TDFT (siehe Kapitel 2.1.1) stellt einen Kompromiß zwischen zeit- und frequenzbasierter Darstellung eines Signals dar. Es sind somit zugleich Informationen über den Zeitpunkt und über die Frequenzkomponenten eines Ereignisses verfügbar, wobei die Auflösung durch die für alle Frequenzkomponenten geltende Fensterlänge L bestimmt wird.

Ein in dieser Hinsicht flexibleres Verfahren stellt die Wavelet-Transformation (WT, wavelet transform) dar, die es gestattet, die Fensterlänge zu variieren. Dadurch kann ein längeres Fenster für tieffrequente Anteile und ein kürzeres für hochfrequente Anteile verwendet werden.

Die FT stellt ein gegebenes Signal als Summe von Sinusschwingungen unendlicher Dauer und unterschiedlicher Frequenz dar:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega, \quad (2.41)$$

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt. \quad (2.42)$$

Die WT verwendet nun im Gegensatz dazu sogenannte Wavelets. Wie in Abb. 2.6 zu sehen ist, hat ein Wavelet ψ eine begrenzte Dauer und ist nicht so regelmäßig geformt wie eine Sinusschwingung. Des weiteren kann ein Wavelet auch zeitlich komprimiert oder gedehnt werden.

Die kontinuierliche WT ist folgendermaßen definiert:

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} x(t) \psi(\text{scale}, \text{position}, t) dt. \quad (2.43)$$

Die Waveletkoeffizienten C sind eine Funktion der Skalierung und Position eines Wavelets, wobei C das Inprodukt zwischen dem modifizierten Wavelet und dem zu analysierenden Signal darstellt. Die Skalierung entspricht der Frequenzachse und die Position der Zeitachse.

Um eine effiziente Implementierung zur Verarbeitung digitalisierter Signale zu ermöglichen, wird statt der kontinuierlichen WT die diskrete WT verwendet. Bei der diskreten

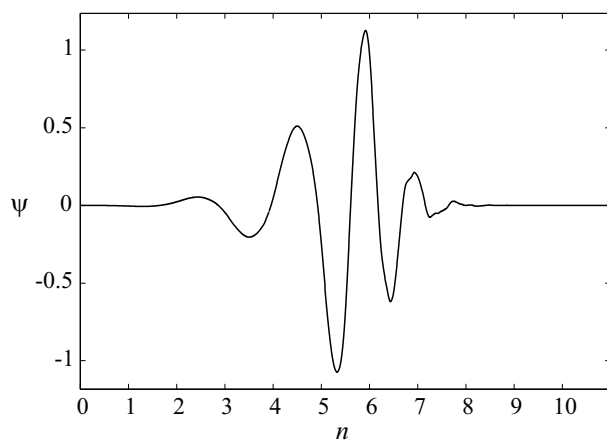


Abb. 2.6: Daubechies Wavelet der Ordnung 6.

WT werden die Skalierung und Position basierend auf Potenzen von zwei gewählt (dyadic scaling).

Die diskrete WT entspricht einer fortgesetzten Hoch- und Tiefpaßfilterung, wobei der Hochpaßanteil als Detail und der Tiefpaßanteil als Näherung des Signals bezeichnet werden. Da dies aber zu zwei Signalen und somit zur doppelten Datenmenge führen würde, werden, wie in Abb. 2.7 illustriert, das Hoch- und das Tiefpaßsignal einem Downsampling unterzogen und somit jeder zweite Datenpunkt verworfen.

Die Aufspaltung in Hoch- und Tiefpaßanteil kann beliebig fortgesetzt werden, wie im Wavelet Decomposition Tree in Abb. 2.8 veranschaulicht ist.

Aus den ermittelten Wavelet-Koeffizienten kann nun durch einfache Umkehrung des Vorganges (inverse diskrete WT) das Ursprungssignal rekonstruiert werden. Durch das Downsampling kommt es jedoch zu einer Verzerrung des Signals durch Aliasing, das durch geeignete Wahl des Hoch- und Tiefpaßfilters kompensiert werden muß. Durch kombinierte Wahl der Zerlegungs- und Rekonstruktionsfilter, die aus der Waveletfunktion ψ durch einen iterativen Prozeß (Quadrature Mirror Filter) berechnet werden, kann das durch Aliasing verzerrte Signal rekonstruiert werden. Die Funktion ψ ist somit keinesfalls frei wählbar, es existieren aber verschiedene Wavelet-Familien, aus denen eine Waveletfunktion ausgewählt werden kann. Eine davon sind die Daubechies Wavelets; sie ergeben orthogonale Zerlegungen, haben einen beschränkten Träger und sind nicht symmetrisch (siehe Abb. 2.6). Weiters existiert für sie auch ein schneller Berechnungsalgorithmus; genauere Details sind in [11] zu finden.

Das EFPG-Signal s wird in die Anteile s_{D1}, \dots, s_{D5} und s_{A5} aufgespaltet, wie in Abb. 2.9 dargestellt ist. Diese Signale wurden durch Upsampling der Waveletkoeffizienten c_{D1}, \dots, c_{D5} und c_{A5} sowie nachfolgender Filterung mit den Rekonstruktionsfiltern berechnet. Dem NN können nun jeweils drei LPCC-Koeffizienten jeder einzelnen Wavelet-

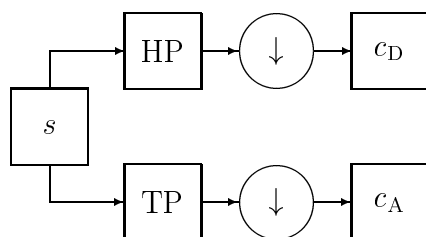


Abb. 2.7: Prinzipielle Bildung der einzelnen Wavelet-Zerlegungen (wavelet decompositions). s ist das ursprüngliche Signal, c_D das Detail und c_A die Näherung (approximation).

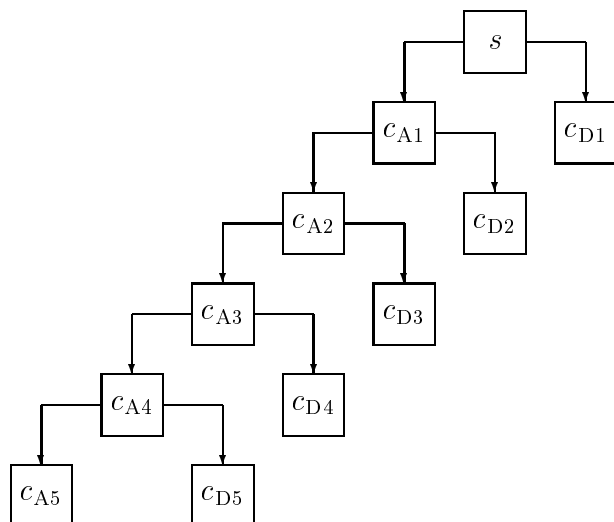


Abb. 2.8: Wavelet Decomposition Tree.

Dekomposition zugeführt werden. Dieser Vektor läßt sich symbolisch wie folgt schreiben:

$$\mathbf{c}_{WT}(r) = \begin{pmatrix} \text{LPCC}\{s_{D1}(r\nu_s T), 3\} \\ \text{LPCC}\{s_{D2}(r\nu_s T), 3\} \\ \text{LPCC}\{s_{D3}(r\nu_s T), 3\} \\ \text{LPCC}\{s_{D4}(r\nu_s T), 3\} \\ \text{LPCC}\{s_{D5}(r\nu_s T), 3\} \\ \text{LPCC}\{s_{A5}(r\nu_s T), 3\} \end{pmatrix}. \quad (2.44)$$

Durch die Beschränkung auf drei LPCC-Koeffizienten wird nur der dominante Frequenzanteil jeder Zerlegung in den Koeffizienten repräsentiert.

2.2 Oxymetriesignal

Das Oxymetriesignal wurde im Rahmen dieser Arbeit bezogen auf den aktuellen Wert als Zeitvektor dargestellt. Die Abtastfrequenz ν_{ox} beträgt 1 Hz, entspricht somit schon dem verwendeten Zeitraster:

$$\mathbf{c}_{\text{OXD}}(r) = \begin{pmatrix} c_{\text{ox}}(r+5) \\ c_{\text{ox}}(r+10) \\ c_{\text{ox}}(r+15) \\ c_{\text{ox}}(r+20) \\ c_{\text{ox}}(r+25) \\ c_{\text{ox}}(r+30) \end{pmatrix} - c_{\text{ox}}(r) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \quad (2.45)$$

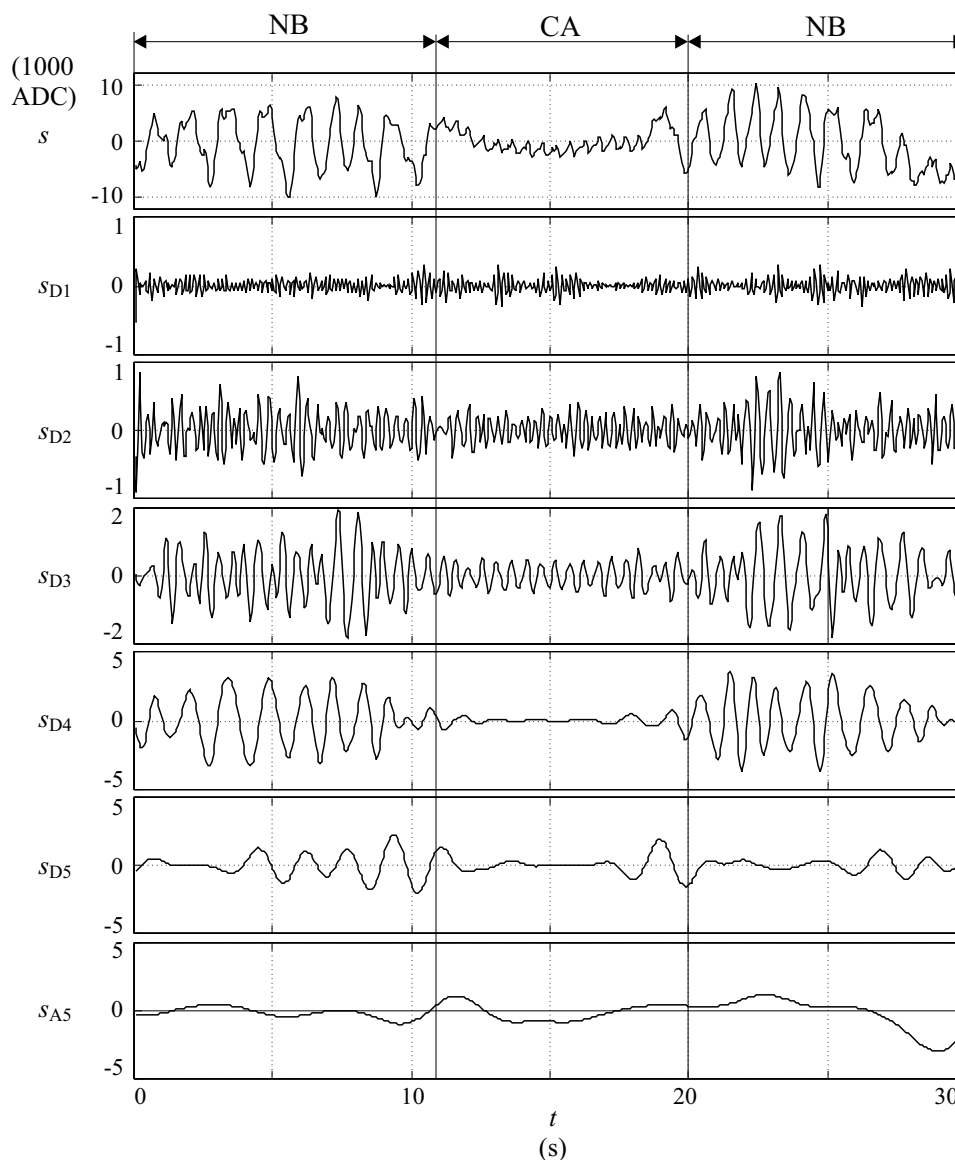


Abb. 2.9: Wavelet-Dekompositionen eines Atemstillstandes unter Verwendung eines Daubechies Wavelets 10. Ordnung. Die fortlaufende Separation in Hoch- und Tiefpaßanteil ist deutlich zu erkennen.

Als zweite Variante wurde zusätzlich zu (2.45) der Feature-Vektor

$$\mathbf{c}_{\text{OXQ}}(r) = \frac{1}{c_{\text{ox}}(r)} \begin{pmatrix} c_{\text{ox}}(r + 5) \\ c_{\text{ox}}(r + 10) \\ c_{\text{ox}}(r + 15) \\ c_{\text{ox}}(r + 20) \\ c_{\text{ox}}(r + 25) \\ c_{\text{ox}}(r + 30) \end{pmatrix} \quad (2.46)$$

durch Bildung des Quotienten berechnet. Eine Selektion zwischen beiden Darstellungsformen wird in Kapitel 4.2.1 durchgeführt.

2.3 Herzfrequenzsignal

Die Abtastfrequenz des Herzfrequenzsignals $\nu_C = 4$ Hz wird durch Downsampling entsprechend

$$\overline{f_C}(n) = \frac{1}{n} \sum_{k=\nu_C T(n-1)}^{\nu_C T n - 1} f_C(k) \quad (2.47)$$

auf $1/T = 1$ Hz reduziert. Das Herzfrequenzsignal wird wie das Oxymetriesignal ebenfalls als Zeitvektor

$$\mathbf{c}_{\text{HRD}}(r) = \begin{pmatrix} \overline{f_C}(r\nu_s T_C + 5) \\ \overline{f_C}(r\nu_s T_C + 10) \\ \overline{f_C}(r\nu_s T_C + 15) \\ \overline{f_C}(r\nu_s T_C + 20) \\ \overline{f_C}(r\nu_s T_C + 25) \\ \overline{f_C}(r\nu_s T_C + 30) \end{pmatrix} - \overline{f_C}(r\nu_s T_C) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (2.48)$$

dargestellt. Der Feature-Vektor $\mathbf{c}_{\text{HRQ}}(r)$ wird analog zu (2.46) definiert.

2.4 Standardisierung

Falls einem NN mehrere Variablen x_i unterschiedlicher Einheit und Größe zugeführt werden, ist es sinnvoll, deren Größenordnungen durch eine lineare Transformation aneinander anzupassen. Jede Variable wird auf Mittelwert $\overline{x}_i = 0$ und Varianz $\sigma_i^2 = 1$ standardisiert. Dadurch können die Konvergenzgeschwindigkeit und die numerische Stabilität der Lernalgorithmen für ein MLP und RBFN stark verbessert werden.

Die Berechnung des Mittelwertes \overline{x}_i und der Varianz σ_i^2 eines Datensets bestehend aus M Vektoren \mathbf{x}^m der Dimension p erfolgt durch:

$$\overline{x}_i = \frac{1}{M} \sum_{m=1}^M x_i^m \quad i = 1, 2, \dots, p \quad (2.49)$$

und

$$\sigma_i^2 = \frac{1}{M-1} \sum_{m=1}^M (x_i^m - \overline{x}_i)^2 \quad i = 1, 2, \dots, p \quad (2.50)$$

Die Standardisierung des Datensets wurde definiert als

$$\tilde{x}_i^m = \frac{x_i^m - \bar{x}_i}{\sigma_i}, \quad (2.51)$$

für eine Matrix \mathbf{X} bestehend aus Vektoren \mathbf{x}^m wird folgende Notation verwendet:

$$\widetilde{\mathbf{X}} = \text{STAND}(\mathbf{X}). \quad (2.52)$$

2.5 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (principal component analysis, PCA) stellt ein Verfahren zur Reduktion der Dimension p eines beliebigen Vektorsets \mathbf{x}^m dar. In dieser Arbeit wird die PCA speziell auf Eingangsvektoren eines NN angewendet. Die Hauptkomponenten (PCs) v_k sind Linearkombinationen

$$v_k = \mathbf{a}_k \mathbf{x} \quad (2.53)$$

der ursprünglichen Komponenten des Eingangsvektors. Bei geeigneter Wahl von \mathbf{a}_k sind diese zueinander orthogonal und somit unkorreliert. Die Varianz der ersten PC wird durch entsprechende Wahl von a_1 maximiert und repräsentiert somit den Großteil der Variation der Eingangsvektoren \mathbf{x}^m .

In Matrixschreibweise werden die M Vektoren $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^M$ der Dimension p zur $(p \times M)$ Matrix $\widetilde{\mathbf{X}}$ zusammengefaßt. Um die Notation zu vereinfachen, wird die Matrix \mathbf{X} mit den Elementen

$$(\mathbf{X})_{ij} = (\widetilde{\mathbf{X}})_{ij} - \bar{x}_i \quad (2.54)$$

definiert, deren Zeilenvektoren mittelwertfrei sind (Definition von \bar{x}_i siehe (2.49)).

Die Kovarianzmatrix \mathbf{S} folgt dann aus

$$\mathbf{S} = \frac{1}{M-1} \mathbf{X} \mathbf{X}^T. \quad (2.55)$$

Die Eigenvektoren \mathbf{a}_k der Kovarianzmatrix \mathbf{S} bilden die nach der Größe der Eigenwerte l_k geordneten Spalten der Matrix \mathbf{A} . Die PCs lassen sich somit folgendermaßen darstellen:

$$\mathbf{V} = \mathbf{X}^T \mathbf{A}. \quad (2.56)$$

Zur effektiven Berechnung der Eigenvektoren \mathbf{a}_k und Eigenwerte l_k der Kovarianzmatrix \mathbf{S} wird im Rahmen dieser Arbeit die Singular Value Decomposition (SVD) verwendet. Jede $(p \times M)$ Matrix \mathbf{X} kann durch

$$\mathbf{X}^T = \mathbf{U} \mathbf{L} \mathbf{A}^T \quad (2.57)$$

dargestellt werden. \mathbf{L} ist eine $(p \times p)$ Diagonalmatrix, deren Elemente $(\mathbf{L})_{ii}$ durch λ_i repräsentiert sind. Für die Eigenwerte der Kovarianzmatrix \mathbf{S} folgt

$$l_i = \text{var}[z_i] = \frac{\lambda_i^2}{M - 1}. \quad (2.58)$$

In der Matrix \mathbf{U} erhält man standardisierte PCs, es folgt

$$\mathbf{V} = \mathbf{U}\mathbf{L}. \quad (2.59)$$

Falls die Eingangsvariablen, das sind die Komponenten x_i des Eingangsvektors \mathbf{x} , stark unterschiedliche Varianz aufweisen, treten die mit der größten Varianz als dominante Anteile in den ersten PCs auf. Um dies zu vermeiden und auch Variablen unterschiedlicher Maßeinheiten miteinander kombinieren zu können, wird der Vektor \mathbf{x} durch den standardisierten Vektor \mathbf{x}^* mit den Komponenten x_i/σ_i ersetzt (Definition von σ_i siehe (2.50)). Die Kovarianzmatrix von \mathbf{X}^* ist dann gleich der Korrelationsmatrix von \mathbf{X} .

Um nun eine effektive Reduzierung der Dimension p des Vektors \mathbf{x} zu erreichen, wird dieser durch einen aus den ersten k PCs bestehenden Vektor ersetzt. Durch

$$h(k) = 100 \sum_{j=1}^k l_j / \sum_{i=1}^p l_i \quad (2.60)$$

kann der somit repräsentierbare Prozentsatz der Varianz (bzw. Information) des ursprünglichen Vektors \mathbf{x} in Abhängigkeit von der Hauptkomponentenanzahl k berechnet werden. Die Eigenwerte l_i sind gleich der Varianz der jeweiligen PCs.

Je nach Datenmaterial kann erwartet werden, daß $h(k)$ schon für $k \ll p$ einen Wert von z.B. 80 Prozent überschreitet. In Abb. 2.10 ist ein typischer Verlauf von $h(k)$ für die TDFT eines EFPG-Signals dargestellt. Weitere Details zur PCA sind in [12] nachzulesen.

Die Bildung der ersten k PCs einer Matrix \mathbf{X} bestehend aus Vektoren \mathbf{x}^m wird folgendermaßen definiert:

$$\mathbf{V} = \text{PCA}(\mathbf{X}, k). \quad (2.61)$$

Die Bildung der PCs eines Datensets bringt auch dann Vorteile, wenn keine oder nur eine minimale Dimensionsreduktion laut (2.60) möglich ist. Die Standardisierung eines

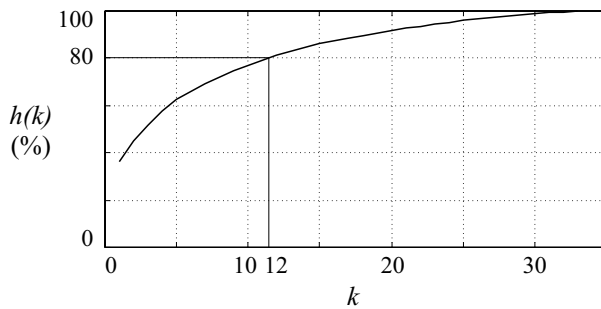


Abb. 2.10: $h(k)$ für die TDFT eines EFPG-Signals. Um 80 Prozent der Varianz der Eingangsdaten darzustellen, reichen 12 PCs aus. Die Anzahl der Komponenten kann somit in diesem Beispiel auf 1/3 reduziert werden.

Datensets laut (2.51) behandelt alle Komponenten des Datensets als unabhängig, die bestehenden Korrelationen werden nicht berücksichtigt. Bei der Bildung der PCs ist dies aber nicht der Fall, die Kovarianzmatrix der standardisierten PCs \mathbf{U} ist gleich der Einheitsmatrix. Dies entspricht einem sogenannten Whitening des ursprünglichen Datensets \mathbf{X} (siehe [1]).

2.6 Kodierung des Zielvektors

Das Klassifikationssignal \tilde{t} stellt einzelne einander gegenseitig ausschließende Kategorien dar (siehe Tab. 1.1). Deswegen bietet sich eine 1-aus-C-Kodierung für den Zielvektor \mathbf{t} des NN an.

2.6.1 1-aus-C-Kodierung

Bei der 1-aus-C-Kodierung wird jeder Kategorie ein Ausgangsneuron zugewiesen. Der Zielvektor wird aus dem Klassifikationssignal \tilde{t} formal folgendermaßen gebildet:

$$\mathbf{t}_1(r) = \begin{pmatrix} \delta(\tilde{t}(r) - c_1) \\ \delta(\tilde{t}(r) - c_2) \\ \vdots \\ \delta(\tilde{t}(r) - c_i) \end{pmatrix}. \quad (2.62)$$

Die c_i sind die Werte der im Zielvektor zu kodierenden Kategorien, entsprechend Tab. 1.1. Die Abtastfrequenz von \tilde{t} beträgt 1 Hz, eine Anpassung an den verwendeten Zeitraster von $T = 1$ s ist daher nicht notwendig.

2.6.2 Geglättete 1-aus-C-Kodierung

Ein Atemstillstand stellt kein abruptes Ausfallen der Atmung dar, sondern am Anfang und Ende kommt es zu einer Übergangsphase. Um dies im Zielvektor

$$\mathbf{t}_2(r-2) = \begin{pmatrix} \delta(\tilde{t}(r) - c_1) * w_2(r) \\ \delta(\tilde{t}(r) - c_2) * w_2(r) \\ \vdots \\ \delta(\tilde{t}(r) - c_i) * w_2(r) \end{pmatrix} \quad (2.63)$$

besser zu repräsentieren, wird jede Komponente des Zielvektors mit einem Hanning-Fenster w_2 mit einer Länge von vier Sekunden gefaltet. In Abb. 2.11 ist das Ergebnis dieser Operation zu sehen.

Da die Bewertung des aktuellen Atemzustandes eines Patienten im nachhinein durch eine Schwester oder einen Arzt oft in - zur Reduktion des Zeitaufwandes - nur überschlagsmäßiger Weise erfolgt, kann das Datenmaterial sowohl zeitlich falsch markiert als auch

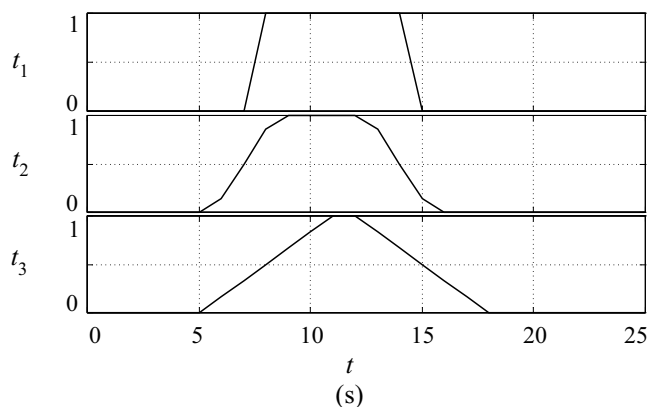


Abb. 2.11: Darstellung der diversen Kodierungsmöglichkeiten für den Zielvektor des NN. \mathbf{t}_1 1-aus-C-Kodierung des Zielvektors, \mathbf{t}_2 geglättete 1-aus-C-Kodierung, \mathbf{t}_3 Prozentanteil des Ereignisses am Zeitfenster.

fehlerhaft klassifiziert worden sein. Durch den geglätteten Übergang wird die zeitliche Position etwas toleranter behandelt.

2.6.3 Anteil des Atemstillstandes am Zeitfenster

Die dem NN zur Verfügung stehenden Eingangsvektoren stammen aus einem Zeitfenster des EFPG-Signals s mit einer Länge von 6 s. Daraus ergibt sich eine mögliche Kodierung des Zielvektors \mathbf{t} , bei welcher der Wert jeder Kategorie dem zeitlichen Prozentanteil des Atemstillstandes am gesamten Zeitfenster entspricht:

$$\mathbf{t}_3(r) = \begin{pmatrix} \frac{1}{6} \sum_{u=r-2}^{r+3} \delta(\tilde{t}(u) - c_1) \\ \vdots \\ \frac{1}{6} \sum_{u=r-2}^{r+3} \delta(\tilde{t}(u) - c_i) \end{pmatrix}. \quad (2.64)$$

Dies ergibt zum Beispiel für die Kategorie CA 0%, falls über einen Zeitraum von sechs Sekunden Normalatmung vorliegt und 100%, falls über den selben Zeitraum eine CA registriert worden ist.

Kapitel 3

Neurales Netz

Ein Neuronales Netz (NN) hat die Fähigkeit, anhand von Beispielen die Transformation der vorverarbeiteten Eingangssignale in das gewünschte Ausgangssignal zu erlernen. Durch die in Kapitel 2 beschriebenen Algorithmen werden aus den Eingangssignalen s , c_{ox} und f_C Feature-Vektoren errechnet. Durch geeignete Kombination dieser Feature-Vektoren wird der Eingangsvektor \mathbf{x} für das NN gebildet. Das Klassifikationssignal \tilde{t} bildet nach einer geeigneten Transformation den Zielvektor \mathbf{t} für das NN.

3.1 Netztopologie

Im Rahmen dieser Arbeit werden zwei NN-Topologien untersucht, die erste ist das Multi Layer Perceptron (MLP), die zweite das Radial Basis Function Network (RBFN).

3.1.1 Multi Layer Perceptron

Das MLP stellt die klassische Struktur eines NN dar. Die Neuronen sind in Schichten organisiert, jedes Neuron einer nachfolgenden Schicht ist mit jedem Neuron der vorhergehenden Schicht verbunden. Die Gewichte der Verbindungen von Neuron j bzw. Eingangsvariable x_j zu Neuron i der Schicht k werden mit $w_{ij}^{(k)}$ bezeichnet. Das NN in Abb. 3.1 besitzt drei Schichten.

Das innere Produkt des Gewichtsvektors $\mathbf{w}_i^{(1)}$ der ersten Schicht mit dem Eingangsvektor \mathbf{x} stellt die Aktivierung a_i des Neurons i dar. Die Aktivierung a_i wird mit einer Aktivierungsfunktion bewertet und ergibt die Ausgangsaktivierung des Neurons i . Als Aktivierungsfunktion eines Neurons wurde die logarithmisch-sigmoide (logsig) Funktion

$$g(a) = \frac{1}{1 + e^{-a}} \quad (3.1)$$

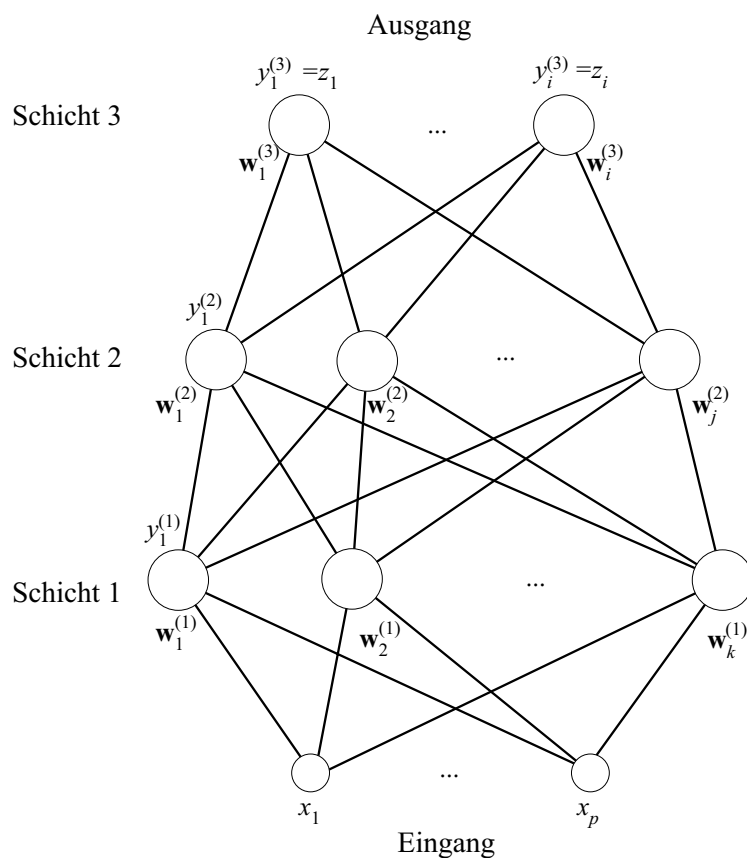


Abb. 3.1: MLP mit 3 Schichten.

verwendet. Die Ausgangsaktivierung des Neurons i der ersten Schicht kann also folgendermaßen berechnet werden:

$$y_i^{(1)}(\mathbf{x}) = g((\mathbf{w}_i^{(1)})^T \mathbf{x}). \quad (3.2)$$

Als Gesamtmodell für ein 3 schichtiges MLP ergibt sich

$$\mathbf{z}(\mathbf{x}) = \mathbf{y}^{(3)}(\mathbf{x}) = g\left((\mathbf{W}^{(3)})^T g\left((\mathbf{W}^{(2)})^T g\left((\mathbf{W}^{(1)})^T \mathbf{x}\right)\right)\right). \quad (3.3)$$

Lernvorgang

Als Lernalgorithmus für das MLP wurde Backpropagation mit Momentum verwendet; dieser minimiert hier die quadratische Fehlerfunktion

$$E = \frac{1}{2} \sum_m \|\mathbf{z}(\mathbf{x}^m; \mathbf{W}) - \mathbf{t}^m\|^2. \quad (3.4)$$

E ist ein Maß für die Abweichung der Ausgangsvektoren \mathbf{z}^m von den gewünschten Zielvektoren \mathbf{t}^m . Bei einer großen Anzahl an freien Parametern, entweder durch eine große Neuronenanzahl oder durch einen hochdimensionalen Eingangsvektor verursacht, führt

dieses Verfahren zu Overfitting. Der Fehler des MLP für das Trainingsset ist dabei zwar klein, für ein davon unabhängiges Datenset (Testset) ergibt sich aber ein überproportional großer Fehler. Um die Generalisierungsfähigkeit zu verbessern, wird der Lernvorgang abgebrochen, sobald der Fehler eines dritten unabhängigen Kontrollsets anzusteigen beginnt. Dadurch kann die Anzahl der effektiven freien Parameter im MLP reduziert werden. Dies wird als Early Stopping bezeichnet. In diesem Zusammenhang wird die normalisierte Fehlerfunktion

$$\tilde{E} = \sqrt{\frac{\sum_m \|\mathbf{z}(\mathbf{x}^m; \mathbf{W}) - \mathbf{t}^m\|^2}{\sum_m \|\mathbf{t}^m - \bar{\mathbf{t}}\|^2}}, \quad (3.5)$$

$$\bar{\mathbf{t}} = \frac{1}{M} \sum_{m=1}^M \mathbf{t}^m \quad (3.6)$$

verwendet. Sie ist unabhängig von der Anzahl der Eingangsvektoren und ergibt eins, wenn das NN die Zielvektoren im Mittel vorhersagen kann und null, wenn das Ergebnis exakt den Zielvektoren entspricht.

Vor dem Lernvorgang werden alle Gewichte des MLP zufällig initialisiert. Diese Initialisierung hat großen Einfluß auf die Konvergenzgeschwindigkeit des Lernalgorithmus und auf das endgültig gefundene relative Minimum der Fehlerfunktion. Deswegen werden 10 Anfangsinitialisierungen mit einem nachfolgenden kurzen Trainingszyklus durchgeführt. Die Anfangsinitialisierung mit dem kleinsten Fehler wird für den endgültigen Trainingsvorgang verwendet.

In diesem Zusammenhang ist noch erwähnenswert, daß laut [1] die Mindestanzahl der Trainingsvektoren M , die für eine gute Generalisierungsfähigkeit eines NN mit zwei Schichten benötigt wird, folgendermaßen abgeschätzt werden kann:

$$M_{\min} \simeq W/\epsilon. \quad (3.7)$$

W ist die Anzahl der Gewichte im NN, und $1 - \epsilon$ ist der Prozentanteil der korrekt zu klassifizierenden Eingangsvektoren. Falls nun $\epsilon = 0.1$ gewählt wird, bedeutet dies, daß M mindestens zehn mal so groß wie W sein muß.

Ein MLP bestehend aus 3 Schichten mit Sigmoid-Aktivierungsfunktion kann beliebige Entscheidungsgrenzen repräsentieren. Wie in [1] gezeigt wird, kann durch Kombination mehrerer Neuronen in einem MLP mit 3 Schichten eine lokalisierte Antwort erzeugt werden, aus der jede beliebige Funktion zusammengesetzt werden kann. Dies ist der Funktion des RBFN äquivalent, das im folgenden Kapitel beschrieben wird.

3.1.2 Radial Basis Function Network

Das in Kapitel 3.1.1 beschriebene MLP basiert auf Neuronen, die das innere Produkt aus Eingangsvektor und Gewichtsvektor mit einer nichtlinearen Funktion bewerten. Das

RBFN gehört zu einer Klasse von NN, bei der die Aktivierung der verborgenen Neuronen der ersten Schicht durch den Abstand des Eingangsvektors von einem Prototypenvektor bestimmt wird. Durch die Struktur des RBFN (siehe Abb. 3.2) kann dieses in zwei Stufen trainiert werden. In der ersten Phase werden die Prototypenvektoren der ersten Schicht bestimmt. Im Rahmen dieser Arbeit werden dafür eine selbstorganisierende Karte (SOM, self organizing map) und der Learning Vector Quantization (LVQ)-Algorithmus verwendet. Erst in der zweiten Phase werden die Gewichte der linearen Neuronen der zweiten Schicht durch den Levenberg-Marquardt Algorithmus (siehe [1] und [13]) bestimmt.

Dieser zweistufige Prozeß bringt einen Geschwindigkeitsvorteil, aber die Kopplung zwischen den beiden Schichten ist nicht sehr stark. Eine gemeinsame Optimierung ist zwar möglich (siehe [1]), aber sehr rechenintensiv.

Die radialen Basisfunktionen haben ihren Ursprung in der Theorie der exakten Interpolation von Datenpunkten in einem mehrdimensionalen Raum. Bei einem RBFN ergibt sich die gleiche Aufgabenstellung, die Interpolation muß jedoch nicht exakt sein. Jeder Prototypenvektor bildet das Zentrum einer radialen Funktion, die durch eine gewichtete Summenbildung die gewünschte Hyperfläche im mehrdimensionalen Raum ergibt.

Als radiale Basisfunktion wird die Gaußfunktion

$$\gamma(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3.8}$$

verwendet. Wenn nun für x der Abstand des Eingangsvektors \mathbf{x} zum Zentrum $\boldsymbol{\mu}_k$ des jeweiligen Neurons eingesetzt wird, ergibt sich für die Basisfunktion jedes Neurons:

$$\gamma_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right). \tag{3.9}$$

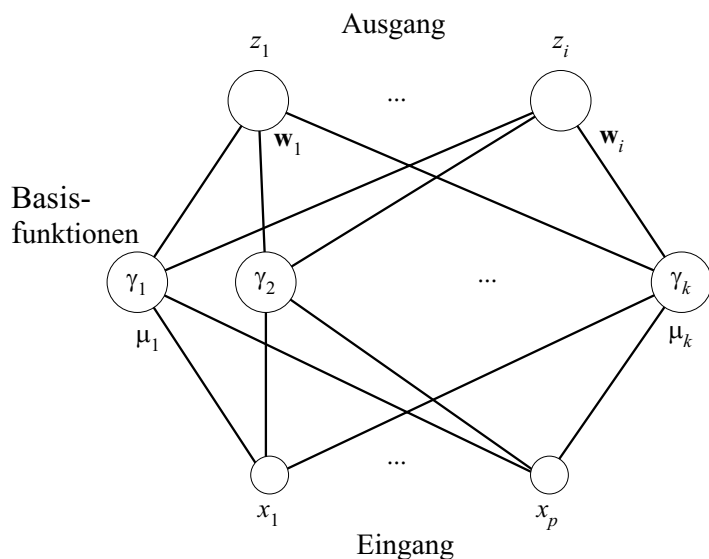


Abb. 3.2: Struktur eines RBFN. Jede Basisfunktion γ_k mit dem Zentrum $\boldsymbol{\mu}_k$ wird durch ein Neuron der verborgenen Schicht repräsentiert. Die Gewichte \mathbf{w}_i sind als Verbindungen der Basisfunktionen mit den linearen Ausgangsneuronen dargestellt.

Das RBFN kann also komplett durch

$$z_i(\mathbf{x}) = \sum_{k=1}^K w_{ik} \gamma_k(\mathbf{x}) + w_{i0} \tag{3.10}$$

beschrieben werden. Da die Eingangsdaten meistens verrauscht sind, erweist es sich aus theoretischen Überlegungen als sinnvoll, normierte Basisfunktionen zu verwenden:

$$\gamma_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right) / \sum_{\kappa=1}^K \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_\kappa\|^2}{2\sigma_\kappa^2}\right). \tag{3.11}$$

Dies entspricht lateralen Verbindungen zwischen den Neuronen der verborgenen Schicht. In dieser Arbeit werden die normierten Basisfunktionen jedoch nicht eingesetzt.

Self Organizing Map

Einer SOM wird während des Lernvorganges keine explizite Ausgabe vorgeschrieben, daher kann auch keine Fehlerfunktion definiert werden. Dennoch führt der Lernvorgang zur Festlegung der Gewichte jedes Neurons in einem selbstgesteuerten, stochastischen Prozeß (siehe [14] und [13]). Die Gewichte der K Neuronen werden durch die Vektoren $\boldsymbol{\mu}_k$ dargestellt und nach Abschluß des Lernvorganges als Zentrumsvektoren für das RBFN verwendet.

Wie aus der Neurobiologie bekannt ist, besitzen viele Verarbeitungsregionen des Gehirns eine lineare oder planare Struktur. Das bedeutet, daß die vieldimensionalen Eindrücke der Sinnesorgane im Gehirn auf ein- oder zweidimensionale Areale abgebildet werden. In so einem Areal besteht außerdem auch eine Nachbarschaftsbeziehung zwischen den einzelnen in diesem Bereich repräsentierten Erregungsmustern. Beim Modell der SOM von Teuvo Kohonen wird diese Topologie berücksichtigt, und die Nachbarschaft eines Neurons wird nicht erst beim freien Lauf des NN, sondern schon in der Lernphase beeinflusst.

Im eindimensionalen Fall stellen die Neuronen eine Kette dar (siehe Abb. 3.3), jedes erhält den Eingangsvektor \mathbf{x} und berechnet die zugehörige Erregung. Durch das Lernverfahren wird jedes Neuron auf einen Unterbereich des Definitionsbereiches spezialisiert.

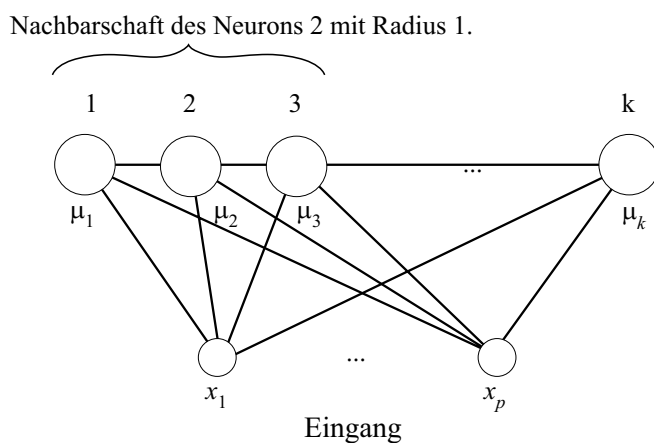


Abb. 3.3: Eindimensionale Neuronenanordnung einer SOM.

Zur Berücksichtigung der Nachbarschaft wird jedes Neuron numeriert. Somit kann im Radius r von einer Nachbarschaft gesprochen werden, in der die Neuronen während des Lernvorganges mitbeeinflusst werden. Diese Nachbarschaft wird durch die Funktion $\varphi(i, k)$ definiert. Sie gibt an, wie stark das Neuron i an das Zentrum k gebunden ist. Eine mögliche Definition wäre z.B. eine abnehmende Funktion des Abstandes $\|i - k\|$.

Das Lernverfahren belegt nun die Gewichtsvektoren $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k$ mit dem Mittelwert aus Maximal- und Minimalwert aller Eingangsvektoren, sowie den Nachbarschaftsradius r und die Lernrate η mit Anfangswerten. Der Lernvorgang wird in folgenden Schritten ausgeführt:

1. Von den Eingangsvektoren \mathbf{x}^m wird eine Eingabe $\boldsymbol{\xi} = \mathbf{x}^i$ zufällig ausgewählt.
2. Das Neuron k mit der maximalen Ausgangsaktivierung, das ist jenes mit dem minimalen euklidischen Abstand $\|\boldsymbol{\mu}_k - \boldsymbol{\xi}\|$, wird ermittelt.
3. Die Gewichtsvektoren der Neuronen i in der Nachbarschaft r des Neurons k werden nach der Vorschrift $\boldsymbol{\mu}_i \leftarrow \boldsymbol{\mu}_i + \eta\varphi(i, k)(\boldsymbol{\xi} - \boldsymbol{\mu}_i)$ aktualisiert.
4. Reduktion der Lernrate η und des Radius r . Fortsetzung mit Punkt 1 oder Abbruch nach festgelegter Schrittzahl.

Falls die SOM als eigenständiges NN verwendet wird, feuert nach dem Abschluß des Lernvorganges immer das Neuron, dessen Aktivierung maximal ist. Die Nummer des aktivierten Neurons reicht somit als Information aus. Obwohl der Lernvorgang selbst nicht überwacht ist, muß eine Zuordnung zu den entsprechenden Zielkategorien durchgeführt werden, um verwendbare Ergebnisse zu erhalten. Das Nachschalten einer Schicht mit linearen Neuronen stellt eine solche Möglichkeit dar.

Die Berechnung der euklidischen Distanz zwischen den Gewichtsvektoren und dem Eingangsvektor kann durch das innere Produkt ersetzt werden, falls die Vektoren auf die Länge eins normiert sind. Gegebenenfalls ist auch eine Pseudonormalisierung auf eine feste Länge l von Vorteil, in diesem Fall muß aber die maximal vorkommende Länge im vorhinein bekannt sein.

Ein unüberwachter Lernvorgang hat zwar den Vorteil, nicht von einer Zielklassifizierung abhängig zu sein, besitzt aber auch einige Schwachpunkte: Die zur ausreichenden Abdeckung eines hochdimensionalen Eingangsraumes benötigte Neuronenanzahl steigt exponentiell mit dessen Dimensionen an, da die Neuronen nicht nur auf den Teil des Eingangsraumes beschränkt werden, in dem auch Daten vorhanden sind. Ein weiterer Nachteil entsteht durch die zweidimensionale Struktur, auf die der Eingangsraum abgebildet wird, denn diese Abbildung muß nicht notwendigerweise optimal sein. Eine Variierung der Seitenverhältnisse oder der Dimension des Ausgangsraumes kann das Lernergebnis stark beeinflussen. Die optimale Geometrie kann aber nur experimentell ermittelt werden.

Learning Vector Quantization

Der LVQ-Algorithmus stellt im Gegensatz zur vorher besprochenen SOM einen überwachten Lernalgorithmus dar, ist mit der SOM aber eng verwandt (siehe [14] und [13]). Die K Neuronen des NN werden den Zielklassen entsprechend ihrer Häufigkeit zugeordnet und initialisiert. Der Lernvorgang verläuft in folgenden Schritten:

1. Von den Eingangsvektoren \mathbf{x}^m wird eine Eingabe $\boldsymbol{\xi} = \mathbf{x}^i$ zufällig ausgewählt, weiters wird aus dem zugehörigen Zielvektor \mathbf{t}^i die Klassenzugehörigkeit c_i bestimmt.
2. Das Neuron k mit der maximalen Erregung, das ist jenes mit dem minimalen euklidischen Abstand $\|\boldsymbol{\mu}_k - \boldsymbol{\xi}\|$, wird ermittelt.
3. Der Gewichtsvektor des Neurons k wird nach der Vorschrift $\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k + \eta(\boldsymbol{\xi} - \boldsymbol{\mu}_k)$ aktualisiert, falls das Neuron k ebenfalls der Klasse c_i zugeordnet ist. Andernfalls wird das Vorzeichen umgedreht: $\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k - \eta(\boldsymbol{\xi} - \boldsymbol{\mu}_k)$. Im Fall einer Fehlklassifikation bewegt sich der Gewichtsvektor dadurch in die entgegengesetzte Richtung.
4. Fortsetzung mit Punkt 1, oder Abbruch nach festgelegter Schrittzahl.

Durch den LVQ-Algorithmus können auch nichtlinear separierbare Eingangsvektoren korrekt klassifiziert werden, die Neuronenzahl muß nur groß genug sein. Die durch das LVQ-Verfahren ermittelten Gewichtsvektoren werden direkt für die Zentrumsvektoren eines RBFN verwendet. Die LVQ hat ein besseres Lernverhalten bei einem hochdimensionalen Eingangsvektor als die SOM, da hier ein überwachter Lernvorgang durchgeführt wird und somit der Eingangsvektorraum nur dort mit Neuronen abgedeckt wird, wo auch Daten vorhanden sind.

Wahl des Spread σ

Der Spread bestimmt die Breite der radialen Funktion γ . Für seine Wahl existiert kein optimaler Algorithmus, σ kann daher nur empirisch ermittelt werden. Im Rahmen dieser Arbeit werden dafür zwei Varianten untersucht. Die einfachere Variante besteht darin, den Spread σ_k für alle Basisfunktionen gleich zu wählen:

$$\sigma_k = \sigma. \quad (3.12)$$

Als zweite Variante wird σ_k für jedes Neuron individuell zweimal so groß wie der minimale Abstand von Neuron k zum nächsten Nachbarn gewählt:

$$\sigma_k = 2 * \min_l (\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|), \quad l \neq k. \quad (3.13)$$

Die Wahl von σ_k hat starken Einfluß auf die Generalisierungsfähigkeit des RBFN. Ein größerer Wert ist allgemein mit besserer Generalisierungsfähigkeit gleichzusetzen.

3.2 Testverfahren

3.2.1 Kreuzvalidierung

Um die Ergebnisse eines NN statistisch beurteilen zu können, ist es notwendig, das Datenmaterial aufzuteilen, wie in Abb. 3.4 schematisch dargestellt wird. Von den Eingangsvektoren wurden 20 Prozent für ein Kontrollset D_0 verwendet, der Rest wurde in S Segmente geteilt ($D_1 \dots D_S$), und je ein Segment wurde für das Testset und die restlichen $S - 1$ Segmente für das Trainingsset verwendet. Durch diese Segmentierung des Datenmaterials und die Durchführung von S Trainingszyklen können statistische Aussagen über die Eigenschaften des NN gemacht werden, wie die mittlere Erkennungsrate und ihre Varianz (siehe [15]). Diese Vorgangsweise wird als S -fache Kreuzvalidierung bezeichnet. Das Kontrollset kann zur Optimierung verschiedener Netzparameter wie der Neuronenanzahl oder der Anzahl an Trainingsschritten verwendet werden.

Die Aufteilung der Vektoren ohne Berücksichtigung ihrer Patienten- oder Ereigniszugehörigkeit stellt die einfachste Möglichkeit dar, eine Kreuzvalidierung durchzuführen. Diese Segmentierung entspricht aber nicht den in der reellen Anwendung gestellten Anforderungen. Zwischen den Feature-Vektoren eines Ereignisses besteht eine hohe Ähnlichkeit bzw. Korrelation. Falls sich nun verschiedene Vektoren aus ein und demselben Ereignis im Trainings- und Testset befinden, werden dadurch die Erkennungsrate eines NN über- und die zugehörige Varianz unterbewertet. Diese Form der Segmentierung ist aber wegen der geringen Varianz gut zur Parameteroptimierung geeignet.

Durch die Aufteilung der Vektoren auf Basis der Ereignisse bzw. in weiterer Folge auf Basis der Patienten erfolgt eine Annäherung an die im angestrebten Einsatz auftretenden Anforderungen. Dies geht mit einer Erhöhung der Varianz einher.

Die Klassifizierung der Meßdaten geschah nicht auf Basis der Ereignisse, sondern sie wurde pro Eingangsvektor und somit jede Sekunde durchgeführt, um eine Aussage über die Länge eines Ereignisses treffen zu können. Durch die unterschiedliche Länge der verschiedenen Ereignistypen kommt es zu einem starken Ungleichgewicht der Vektorenanzahl

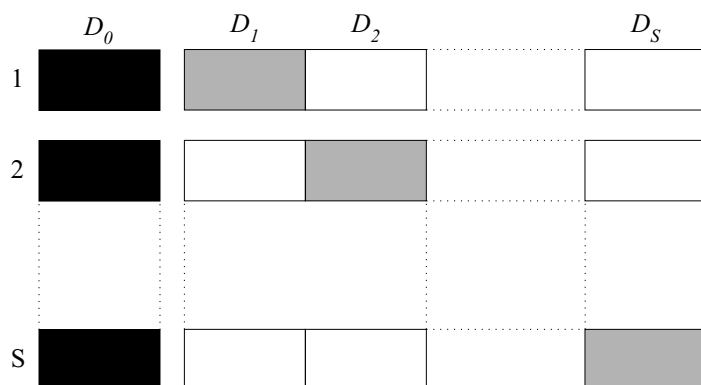


Abb. 3.4: Schematische Darstellung der Aufteilung des Datensets in die Segmente D_0, \dots, D_S . Das Segment D_0 stellt das Kontrollset dar, von den Segmenten D_1, \dots, D_S wird jeweils das grau schattiert dargestellte als Testset und die restlichen $S - 1$ Segmente als Trainingsset verwendet.

jeder Kategorie. Dies führt zu einer schlechten Erkennungsrate, da von einzelnen Kategorien weniger Eingangsvektoren als von anderen vorhanden sind. Eine Möglichkeit, dies zu verbessern, besteht darin, die Anzahl der Vektoren pro Kategorie zu bestimmen und von jeder dann durch eine Zufallsauswahl nur so viele zu verwenden, wie in der kleinsten Gruppe vorkommen.

Durch die beschriebene Vorgangsweise werden der Übergangsbereich und der Normalatmungsbereich schlechter repräsentiert als der Bereich, in dem das Ereignis vorliegt. Dies wird noch deutlicher, wenn statt einer einfachen 1-aus-C-Kodierung der Zielvektoren andere Formen der Darstellung des Zielvektors verwendet werden, wie die geglätteten Zielvektoren oder der Prozentanteil des Ereignisses an der Fensterlänge.

3.2.2 Mehrere Neurale Netze als Komitee

Aus V NNs kann ein Komitee entsprechend

$$\mathbf{z}_{\text{COM}}(\mathbf{x}) = \frac{1}{V} \sum_{i=1}^V \mathbf{z}_i(\mathbf{x}) \quad (3.14)$$

gebildet werden (zum Beispiel aus den verschiedenen Realisierungen einer V -fachen Kreuzvalidierung), siehe [1].

Bei statistischer Unabhängigkeit der einzelnen NN ist der Fehler des Komitees

$$E_{\text{COM}} = \frac{1}{V} E_{\text{AV}} \quad (3.15)$$

um den Faktor $1/V$ gegenüber dem Fehler des durchschnittlichen Netzes

$$E_{\text{AV}} = \frac{1}{V} \sum_{i=1}^V E_i \quad (3.16)$$

reduziert. E_{AV} ist der Mittelwert der Fehler der Einzelnetze E_i .

In der Praxis bestehen zwischen den einzelnen Netzen jedoch statistische Abhängigkeiten, E_{COM} ist aber sicher kleiner als E_{AV} :

$$E_{\text{COM}} < E_{\text{AV}}. \quad (3.17)$$

Kapitel 4

Empirische Auswertungen

4.1 Zusammenstellung des Datenmaterials

Die Vorselektion des Datenmaterials, das für den Lernvorgang eines Neuronen Netzes (*neural network*, NN) verwendet wird, hat großen Einfluß auf das erzielbare Ergebnis. Im Fall der Erkennung von Atemstillständen liegen einzelne Ereignisse unterschiedlicher Dauer vor. Die Vorverarbeitungsalgorithmen berechnen aus den gemessenen Vitalparametern Feature-Vektoren in einem einheitlichen Zeitraster von einer Sekunde. Das vorliegende Datenmaterial stammt von verschiedenen Patienten, wobei von jedem eine stark unterschiedliche Anzahl an Ereignissen zur Verfügung steht. Um eine gute Generalisierungsfähigkeit des NN für neue Patienten zu erreichen, wurde folgendermaßen vorgegangen:

Von jedem Ereignistyp werden von allen Patienten nach Möglichkeit gleich viele Ereignisse verwendet, um einen breiten Querschnitt aller Patienten in den Eingangsdaten des NN zu repräsentieren. Nach diesem automatisierten Vorgang (siehe Anhang) werden die Ereignisse optisch inspiziert und jene entfernt, die durch das Krankenhauspersonal offensichtlich fehlerhaft bewertet wurden. Die verbleibenden Ereignisse stellen dann die Ereignisliste dar, aus der die Eingangsvektoren berechnet werden, die allen Analysen zugrunde liegen. Dies wurde für die Patientengruppe Säuglinge durchgeführt, bei der Patientengruppe Erwachsene wurde keine Inspektion der zufällig ausgewählten Ereignisse vorgenommen.

4.2 Anwendung der Hauptkomponentenanalyse

4.2.1 Selektion alternativer Feature-Vektoren

In Kapitel 2 wurden verschiedene Möglichkeiten zur Darstellung der Meßergebnisse durch Feature-Vektoren vorgestellt. In diesem Kapitel soll daraus eine Vorauswahl mittels Hauptkomponentenanalyse (*principal component analysis*, PCA) getroffen werden; dazu wurden

die Daten der Patientengruppe Säuglinge verwendet.

Für das Oxymetriesignal c_{ox} und das Herzfrequenzsignal f_C existieren jeweils zwei alternative Feature-Vektoren, dies sind \mathbf{c}_{OXD} und \mathbf{c}_{OXQ} sowie \mathbf{c}_{HRD} und \mathbf{c}_{HRQ} .

Die aus der PCA von \mathbf{c}_{OXD} resultierenden ersten drei Eigenvektoren a_1, \dots, a_3 sind in Abb. 4.1 als Zeitverlauf dargestellt. Sie unterscheiden sich nur unwesentlich von denen, die sich bei der PCA von \mathbf{c}_{OXQ} ergeben. Das gleiche gilt auch für die alternativen Formen \mathbf{c}_{HRD} und \mathbf{c}_{HRQ} . Aus diesem Grund werden nur die Feature-Vektoren \mathbf{c}_{OXD} und \mathbf{c}_{HRD} für weitere Analysen verwendet.

Interessanterweise stimmen auch die Eigenvektoren von \mathbf{c}_{OXD} und \mathbf{c}_{HRD} überein, obwohl verschiedene Signale zugrunde liegen und nur deren Algorithmus für die Vektordarstellung übereinstimmt. Der Algorithmus hat also einen starken Einfluß auf das Ergebnis der PCA.

Mittels Gleichung (2.60) kann die Anzahl der Hauptkomponenten (*principal components*, PCs) abgeschätzt werden, die für die Darstellung eines bestimmten Prozentsatzes der Varianz des zugrunde liegenden Datenmaterials benötigt wird. Um 90 Prozent der Varianz des Oxymetriesignals abzudecken, reichen demzufolge die ersten 3 PCs aus. Die gleiche Anzahl an PCs ist auch für das Herzfrequenzsignal ausreichend.

Für die Kodierung des relativen Leistungsdichtespektrums sind in Kapitel 2.1.2 ebenfalls zwei alternative Darstellungsformen \mathbf{c}_{PSD1} und \mathbf{c}_{PSD2} angegeben, wobei \mathbf{c}_{PSD2} die doppelte Komponentenanzahl wie \mathbf{c}_{PSD1} aufweist. In Abb. 4.2 ist der Verlauf von $h(k)$ über $k' = k/k_{max}$ für die Datensets \mathbf{c}_{PSD1} , \mathbf{c}_{PSD2} und \mathbf{c}_{TDFT} dargestellt. Die Darstellungsform \mathbf{c}_{PSD2} benötigt die gleiche Anzahl an PCs wie \mathbf{c}_{PSD1} , obwohl die Komponentenanzahl doppelt so hoch ist. Daraus kann der Schluß abgeleitet werden, daß die enthaltenen Informationen gleichwertig sind. Aus diesem Grund wird der kompakteren Darstellung \mathbf{c}_{PSD1} der Vorzug gegeben.

Der Vergleich des relativen Leistungsdichtespektrums \mathbf{c}_{PSD1} mit dem Frequenzspektrum \mathbf{c}_{TDFT} in Abb. 4.2 zeigt weiters, daß \mathbf{c}_{PSD1} mehr PCs benötigt, obwohl nun aber die Anzahl der Komponenten in beiden Datensets die gleiche ist. Daraus folgt analog, daß \mathbf{c}_{PSD1} mehr Information beinhaltet als \mathbf{c}_{TDFT} .

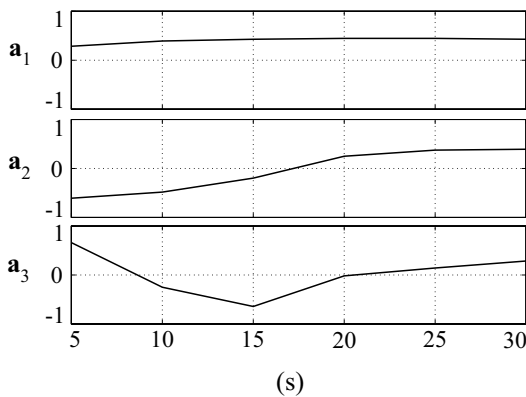


Abb. 4.1: Darstellung der ersten drei Eigenvektoren der Hauptkomponentenanalyse von \mathbf{c}_{OXD} als Zeitverlauf.

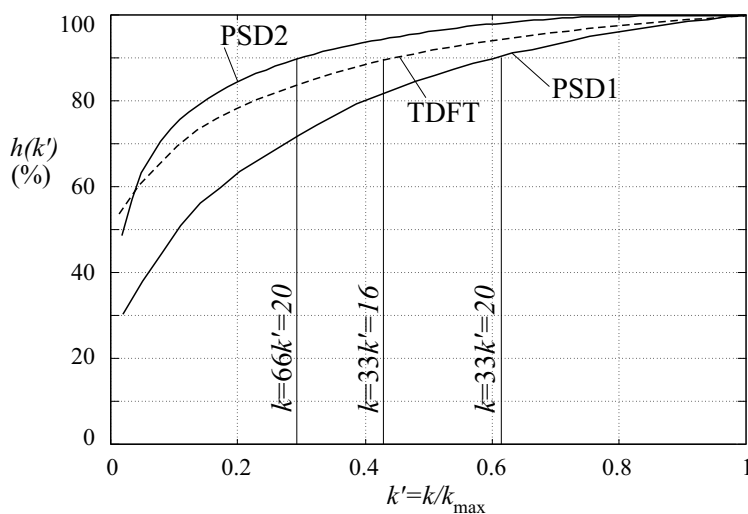


Abb. 4.2: Darstellung von $h(k')$ über $k' = k/k_{\max}$ für die Datensets \mathbf{c}_{PSD1} , \mathbf{c}_{PSD2} und \mathbf{c}_{TDFT} . Der Schnittpunkt von $h(k')$ mit 90 Prozent ist jeweils markiert und die daraus resultierende Komponentenanzahl angegeben. Der bezogene Index wurde zur Darstellung gewählt, da \mathbf{c}_{PSD1} und \mathbf{c}_{TDFT} aus 33 und \mathbf{c}_{PSD2} aus 66 Komponenten besteht.

Wie in Kapitel 3 erläutert, ist eine geringe Komponentenanzahl des Eingangsvektors ein wichtiger Grundstein für eine gute Generalisierungsfähigkeit eines NN. Die Auswahl der Vorverarbeitungsalgorithmen erfolgt also größtenteils nach diesem Kriterium. Im Falle der CT weist der Feature-Vektor 120 Komponenten auf. Durch eine PCA kann die Komponentenanzahl nur ca. um die Hälfte reduziert werden, wie in Abb. 4.3 dargestellt ist. Aus diesem Grund wird dieser Feature-Vektor nicht verwendet.

Die LPC-Darstellung des EFPG-Signals ist, wie in Kapitel 2.5 beschrieben, effizient in der benötigten Komponentenanzahl. Durch die Anwendung der CT auf die LPC-Darstellung ergibt sich die LPCC-Darstellung. Beide Darstellungen sind vom Informationsgehalt gleichwertig, da eine einfache Umrechnung in beide Richtungen möglich ist. Trotzdem ergibt die PCA aber für $\mathbf{c}_{\text{LPC}_{12}}$ und $\mathbf{c}_{\text{LPCC}_{12}}$ unterschiedliche Ergebnisse, wie in Abb. 4.3 dokumentiert ist. Daraus läßt sich der Schluß ableiten, daß die PCA beim

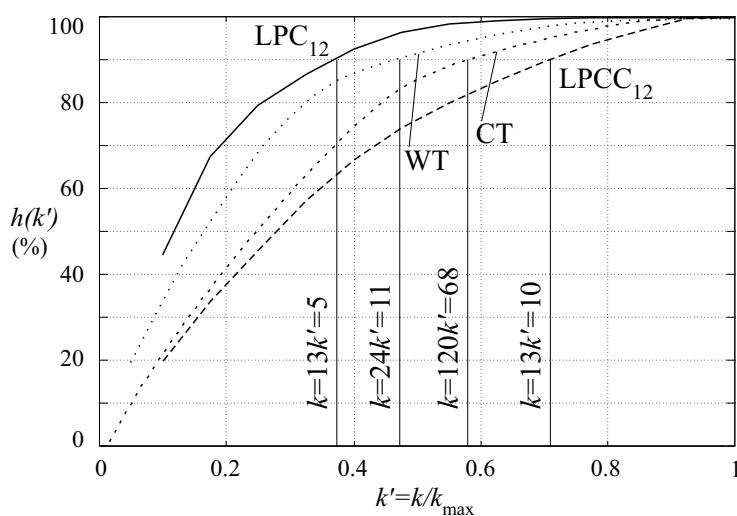


Abb. 4.3: Darstellung von $h(k')$ über $k' = k/k_{\max}$ für die Datensets \mathbf{c}_{CT} , \mathbf{c}_{LPC} , \mathbf{c}_{LPCC} und \mathbf{c}_{WT} .

Vergleich von unterschiedlichen Kodierungsformen ein und desselben Signals durch ihre lineare Natur nur bedingt zur Selektion eingesetzt werden kann.

4.2.2 Lineare Separierbarkeit der Eingangsdaten

Um die Möglichkeit einer linearen Separierung eines Datensets durch die ersten beiden Hauptkomponenten

$$\mathbf{v} = \text{PCA}(\mathbf{c}_{\text{LPCC}_{12}}, 2) \quad (4.1)$$

zu untersuchen, werden diese zweidimensional, wie in Abb. 4.4 dargestellt, aufgetragen. Das dargestellte Beispiel ist typisch für alle anderen Feature-Vektoren, eine einfache lineare Trennung der Vektoren ist auf diesem Weg nicht durchführbar. Durch die zusätzliche Verwendung der um +6s und -6s verzögerten Vektoren wie in

$$\mathbf{v} = \text{PCA} \left(\begin{pmatrix} \mathbf{c}_{\text{LPCC}_{12}}(r) \\ \mathbf{c}_{\text{LPCC}_{12}}(r+6) \\ \mathbf{c}_{\text{LPCC}_{12}}(r-6) \end{pmatrix}, 2 \right) \quad (4.2)$$

ergibt sich ebenfalls keine Verbesserung der linearen Separation.

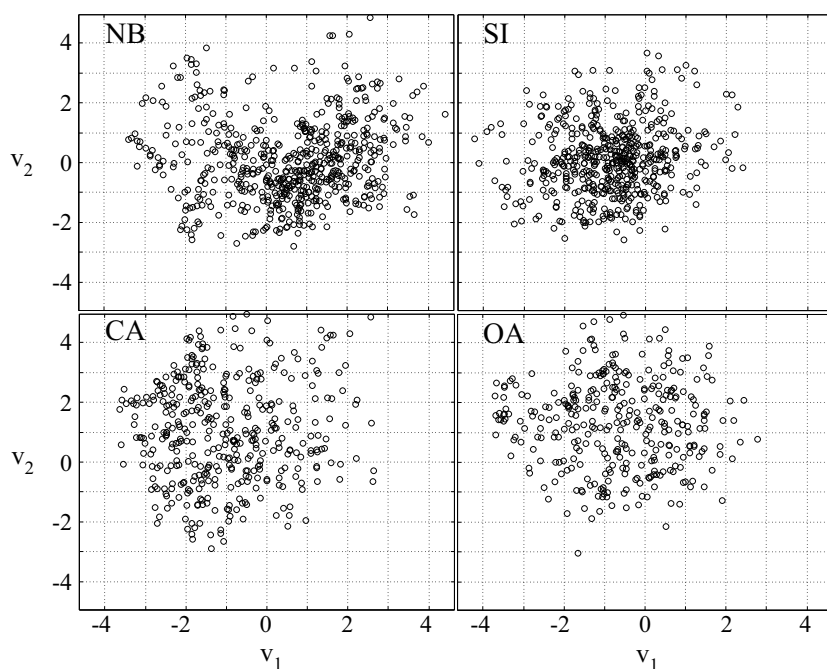


Abb. 4.4: Versuch der linearen Separation der verschiedenen Atemereignisse in zwei Dimensionen mittels Hauptkomponentenanalyse.

4.3 Segmentierung des Datenmaterials

Wie schon in Kapitel 3.2.1 erwähnt, hängt die Varianz der Erkennungsrate, die durch eine Kreuzvalidierung ermittelt wurde, von der gewählten Strategie zur Segmentierung der Eingangsvektoren ab. Um darüber einen besseren Überblick zu bekommen, wurde eine Segmentierung der Eingangs- und Zielvektoren basierend auf der Ereignis- bzw. Patientenzugehörigkeit nach der Leave-One-Out-Methode durchgeführt. Dies ist der Grenzfall, der sich ergibt, wenn die Segmentanzahl S gleich der Anzahl der Ereignisse bzw. der Patienten gewählt wird. Dadurch kann eine Aussage über jedes einzelne Ereignis bzw. jeden einzelnen Patienten getroffen werden.

Als Eingangsvektor wurde

$$\mathbf{x}(r) = \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{LPCC}}(r+6) \\ \mathbf{v}_{\text{LPCC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \quad (4.3)$$

mit

$$\mathbf{v}_{\text{LPCC}}(r) = \text{PCA}(\mathbf{c}_{\text{LPCC}_{12}}(r), 10), \quad (4.4)$$

$$\mathbf{v}_{\text{OX}}(r) = \text{PCA}(\mathbf{c}_{\text{OXD}}(r), 3), \quad (4.5)$$

$$\mathbf{v}_{\text{HR}}(r) = \text{PCA}(\mathbf{c}_{\text{HRD}}(r), 3) \quad (4.6)$$

verwendet (vergleiche Abb. 1.1), als Zielvektor wurde $\mathbf{t} = \mathbf{t}_1$ gewählt. Die hier getroffene Wahl für \mathbf{x} und \mathbf{t} ist suboptimal, die Ergebnisse sind aber dennoch repräsentativ. Als NN wird ein RFBN mit einer LVQ-Eingangsschicht bestehend aus acht Neuronen verwendet, da dafür ein schneller Lernalgorithmus vorhanden ist.

Das für die Analyse verwendete Datenmaterial stammt aus der Patientengruppe Säuglinge, bestehend aus 30 Patienten, von denen insgesamt 176 Ereignisse ausgewählt wurden. Die angegebenen Erkennungsrate R beziehen sich jeweils auf den Anteil an richtig klassifizierten Eingangsvektoren pro Patient bzw. pro Ereignis.

Analyse der Patienten

In Abb. 4.5 ist die Erkennungsrate R pro Ereignistyp für jeden Patienten dargestellt. Daraus ist ersichtlich, wie stark die Streuung über alle Patienten ist. Nur im Fall der Normalatmung (NB) ergibt sich ein konstantes Ergebnis.

Ein Histogramm der Erkennungsrate pro Patient ist in Abb. 4.6 dargestellt. Die mittlere Erkennungsrate beträgt 47.5 Prozent bei einer Standardabweichung von 13.7 Prozent.

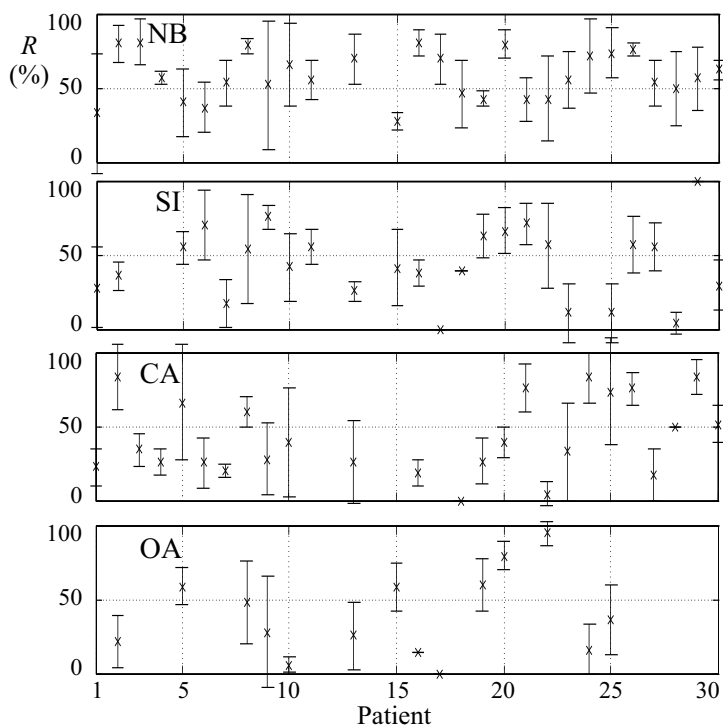


Abb. 4.5: Erkennungsrate R pro Ereignistyp bei Segmentierung nach Patienten (Suglinge).

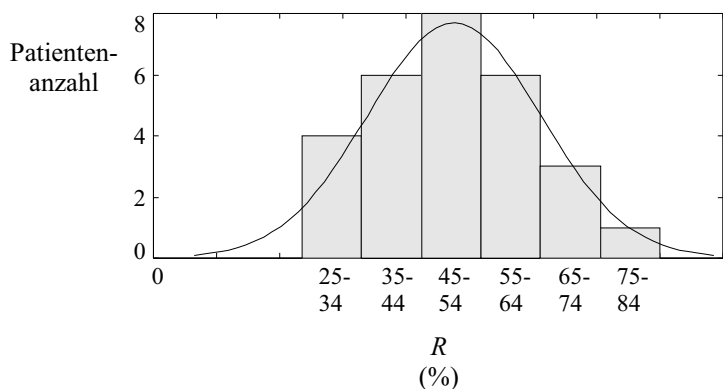


Abb. 4.6: Histogramm der Erkennungsrate R bei Segmentierung nach Patienten. Der Mittelwert betragt 47.5 Prozent und die Standardabweichung 13.7 Prozent.

Analyse der Ereignisse

In Abb. 4.7 ist ein Histogramm der Erkennungsrate pro Ereignis dargestellt. Die mittlere Erkennungsrate betragt hier 54.1 Prozent bei einer Standardabweichung von 20.7 Prozent. Der Vergleich mit der Patientenanalyse zeigt, da bei Segmentierung der Daten auf Patientenbasis, die dem Echtbetrieb entspricht, eine Reduktion der Erkennungsrate um ca. 6 Prozent gegenuber den Ergebnissen bei Segmentierung auf Ereignisbasis zu erwarten ist.

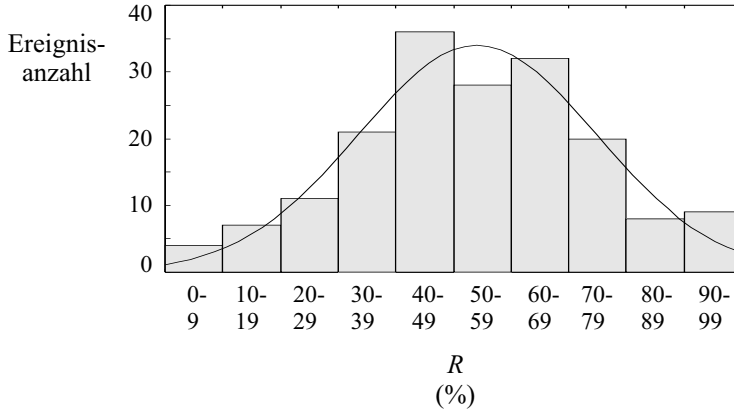


Abb. 4.7: Histogramm der Erkennungsrate R bei Segmentierung nach Ereignissen. Der Mittelwert beträgt 54.1 Prozent und die Standardabweichung 20.7 Prozent.

4.4 Optimale Zusammensetzung des Eingangsvektors

Durch die Untersuchungen mittels PCA im Kapitel 4.2.1 konnte eine grobe Vorauswahl der in Frage kommenden Feature-Vektoren für den Eingangsvektor des NN getroffen werden. Nun wird durch Training eines NN mit verschiedenen zusammengesetzten Eingangsvektoren die optimale Konfiguration ermittelt. Das verwendete Datenmaterial stammt aus der Patientengruppe Säuglinge. Als NN wird ein RFBN mit einer LVQ-Eingangsschicht bestehend aus acht Neuronen verwendet, da dafür ein schneller Lernalgorithmus vorhanden ist und die Varianz der Ergebnisse gut kontrolliert werden kann.

Hauptkomponenten der Feature-Vektoren

Die Dimension der Feature-Vektoren wird mittels PCA reduziert. Die Anzahl der PCs wurde für jeden Feature-Vektor mittels (2.60) so gewählt, daß durch die PCs jeweils 90 Prozent der Varianz der zugrunde liegenden Daten repräsentiert werden. Die verwendeten Parameter sind in Tab. 4.1 angegeben.

$$\begin{aligned}
 \mathbf{v}_{\text{TDFTE}}(r) &= \text{PCA}(\mathbf{c}_{\text{TDFTE}}(r), 17) & \mathbf{v}_{\text{PSD}}(r) &= \text{PCA}(\mathbf{c}_{\text{PSD1}}(r), 20) \\
 \mathbf{v}_{\text{LPCC}}(r) &= \text{PCA}(\mathbf{c}_{\text{LPCC}_{12}}(r), 10) & \mathbf{v}_{\text{LPC}}(r) &= \text{PCA}(\mathbf{c}_{\text{LPC}_{12}}(r), 5) \\
 \mathbf{v}_{\text{WT}}(r) &= \text{PCA}(\mathbf{c}_{\text{WT}}(r), 6) & \mathbf{v}_{\text{OX}}(r) &= \text{PCA}(\mathbf{c}_{\text{OXD}}(r), 3) \\
 \mathbf{v}_{\text{HR}}(r) &= \text{PCA}(\mathbf{c}_{\text{HRD}}(r), 3)
 \end{aligned}$$

Tab. 4.1: Hauptkomponenten der einzelnen Feature-Vektoren.

Zusammensetzung des Eingangsvektors

Die Zusammensetzungen der Eingangsvektoren $\mathbf{x}_1, \dots, \mathbf{x}_{12}$ sind in Tab. 4.2 aufgelistet, als Zielvektor wird \mathbf{t}_1 verwendet.

In Kapitel 2 wurden zwei verschiedene Möglichkeiten der Standardisierung eines Vektorsets beschrieben. Die erste Variante ist die der Standardisierung auf Mittelwert null und Varianz eins, wie in Kapitel 2.4 vorgestellt. Die zweite Variante besteht im sogenannten Whitening des Vektorsets durch Anwendung der Hauptkomponentenanalyse, dies ist in Kapitel 2.5 beschrieben. Zwischen den beiden Varianten konnte kein signifikanter Unterschied ermittelt werden. Da sich die Eingangsvektoren hauptsächlich aus Hauptkomponenten zusammensetzen, die unkorreliert sind, war dies durchaus zu erwarten. Alle Eingangsvektoren wurden daher durch

$$\mathbf{x} = \text{STAND}\{\mathbf{x}_i\} \quad (4.7)$$

standardisiert.

In Tab. 4.3 sind die Ergebnisse des Testsets angegeben. Die Ergebnisse wurden mittels zehnfacher Kreuzvalidierung ermittelt. Dieser Vorgang wurde viermal mit unterschiedlicher Anfangsinitialisierung wiederholt.

Wie aus Tab. 4.3 zu entnehmen ist, ergibt der LPCC Feature-Vektor die besten Ergebnisse, dies sind die höchste Gesamterkennungsrates mit der kleinsten Standardabweichung

$$\begin{aligned} \mathbf{x}_1(r) &= \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{LPCC}}(r+6) \\ \mathbf{v}_{\text{LPCC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} & \mathbf{x}_2(r) &= \begin{pmatrix} \mathbf{v}_{\text{LPC}}(r) \\ \mathbf{v}_{\text{LPC}}(r+6) \\ \mathbf{v}_{\text{LPC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} & \mathbf{x}_3(r) &= \begin{pmatrix} \mathbf{v}_{\text{WT}}(r) \\ \mathbf{v}_{\text{WT}}(r+6) \\ \mathbf{v}_{\text{WT}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \\ \mathbf{x}_4(r) &= \begin{pmatrix} \mathbf{v}_{\text{TDFT}}(r) \\ \mathbf{v}_{\text{TDFT}}(r+6) \\ \mathbf{v}_{\text{TDFT}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} & \mathbf{x}_5(r) &= \begin{pmatrix} \mathbf{v}_{\text{PSD}}(r) \\ \mathbf{v}_{\text{PSD}}(r+6) \\ \mathbf{v}_{\text{PSD}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} & \mathbf{x}_6(r) &= \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{LPCC}}(r+6) \\ \mathbf{v}_{\text{LPCC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \end{pmatrix} \\ \mathbf{x}_7(r) &= \begin{pmatrix} \mathbf{c}_{\text{LPCC12}}(r) \\ \mathbf{c}_{\text{LPCC12}}(r+6) \\ \mathbf{c}_{\text{LPCC12}}(r-6) \\ \mathbf{c}_{\text{OX}}(r) \end{pmatrix} & \mathbf{x}_8(r) &= \begin{pmatrix} \mathbf{v}_{\text{LPC}}(r) \\ \mathbf{v}_{\text{LPC}}(r+6) \\ \mathbf{v}_{\text{LPC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \end{pmatrix} & \mathbf{x}_9(r) &= \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{LPCC}}(r+6) \\ \mathbf{v}_{\text{OX}}(r) \end{pmatrix} \\ \mathbf{x}_{10}(r) &= \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{OX}}(r) \end{pmatrix} & \mathbf{x}_{11}(r) &= \begin{pmatrix} \mathbf{v}_{\text{OX}}(r) \end{pmatrix} & \mathbf{x}_{12}(r) &= \begin{pmatrix} \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \end{aligned}$$

Tab. 4.2: Zusammensetzung der verschiedenen Eingangsvektoren für das NN.

\mathbf{x}	Gesamt	σ	NB	SI	CA	OA
$\mathbf{x}_1(r)$	56.2 ± 3.4	1.9	58.5 ± 8.1	54.9 ± 6.4	54.5 ± 8.4	57.0 ± 9.5
$\mathbf{x}_2(r)$	47.7 ± 5.1	6.4	41.6 ± 9.6	42.7 ± 10.7	52.5 ± 9.5	53.9 ± 8.7
$\mathbf{x}_3(r)$	48.0 ± 5.1	6.8	56.7 ± 10.0	42.1 ± 10.4	43.2 ± 10.4	49.9 ± 10.3
$\mathbf{x}_4(r)$	52.7 ± 4.3	4.1	52.4 ± 11.4	48.2 ± 7.4	52.2 ± 8.3	58.1 ± 9.7
$\mathbf{x}_5(r)$	53.3 ± 4.7	3.0	50.4 ± 10.8	54.9 ± 10.6	56.7 ± 8.0	51.2 ± 10.8
$\mathbf{x}_6(r)$	55.3 ± 5.0	3.7	59.4 ± 8.5	55.2 ± 10.0	56.0 ± 10.6	50.5 ± 10.8
$\mathbf{x}_7(r)$	56.2 ± 4.4	3.0	59.7 ± 10.3	53.3 ± 8.6	57.8 ± 7.3	54.1 ± 11.6
$\mathbf{x}_8(r)$	45.5 ± 5.3	6.4	44.8 ± 10.0	44.5 ± 10.4	54.0 ± 9.1	38.6 ± 9.7
$\mathbf{x}_9(r)$	52.5 ± 4.4	6.8	60.7 ± 9.5	47.3 ± 9.8	46.6 ± 9.6	55.6 ± 9.2
$\mathbf{x}_{10}(r)$	50.1 ± 4.6	10.9	61.6 ± 8.0	44.8 ± 9.9	37.7 ± 9.1	56.5 ± 9.6
$\mathbf{x}_{11}(r)$	44.3 ± 3.8	22.8	17.1 ± 9.6	52.8 ± 9.6	36.8 ± 11.1	70.5 ± 8.3
$\mathbf{x}_{12}(r)$	33.9 ± 3.7	24.3	12.0 ± 7.7	37.5 ± 11.8	19.4 ± 11.5	66.6 ± 13.7

Tab. 4.3: Klassifikationsergebnisse eines RBFN mit einer LVQ Eingangsschicht bestehend aus 8 Neuronen. Die Ergebnisse wurden durch zehnfache Kreuzvalidierung und viermalige Neuinitialisierung des NN ermittelt. Die Spalte „ σ “ gibt die Standardabweichung der Erkennungsraten pro Ereignistyp an.

sowie die kleinste Standardabweichung der Erkennungsraten pro Ereignistyp (Spalte „ σ “). Die Ergebnisse von \mathbf{x}_1 und \mathbf{x}_7 unterscheiden sich nur geringfügig, bei \mathbf{x}_1 ist aber die Standardabweichung sowohl der Gesamterkennungsraten als auch der Erkennungsraten pro Ereignistyp geringer. Eine weitere Optimierung scheint hier zielführend zu sein.

Um weitere Details der LPCC-Darstellung zu ermitteln, werden im folgenden die Anzahl der LPCC-Koeffizienten, die Zeitverzögerung und die Anzahl der verwendeten Hauptkomponenten variiert. Das verwendete NN entspricht jenem aus dem vorhergehenden Kapitel.

Variation der LPCC-Koeffizientenanzahl

In Abb. 4.8 sind die Erkennungsrate und deren Standardabweichung für Test- und Trainingsset bei variiertem LPCC-Koeffizientenanzahl k für

$$\mathbf{x}(r) = \begin{pmatrix} \mathbf{c}_{\text{LPCC}_k}(r) \\ \mathbf{c}_{\text{LPCC}_k}(r+6) \\ \mathbf{c}_{\text{LPCC}_k}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \quad (4.8)$$

aufgetragen. Aus diesem Bild ist zu erkennen, daß die Erkennungsrate mit zunehmender Koeffizientenanzahl ansteigt, ab 12 Koeffizienten aber konstant bleibt.



Abb. 4.8: Erkennungsrate R des Test- und Trainingssets bei variiertem Anzahl k der LPCC-Koeffizienten. Die dicken Linien stellen den Mittelwert, die dünnen Linien die Standardabweichung dar.

Variation der Zeitverzögerung

Der Eingangsvektor setzt sich zusätzlich zum Vektor $\mathbf{c}_{\text{LPCC}_{12}}(r)$ aus zwei um $\pm d$ verzögerten Vektoren zusammen. In Abb. 4.9 ist die Erkennungsrate des Test- und Trainingssets

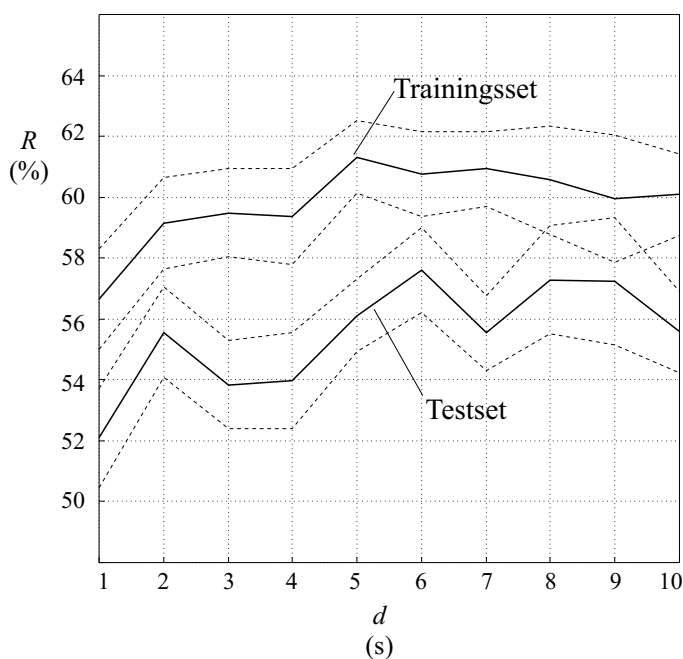


Abb. 4.9: Erkennungsrate R des Test- und Trainingssets bei variiertem Zeitverzögerung d . Die dicken Linien stellen den Mittelwert, die dünnen Linien die Standardabweichung dar.

über die Zeitverzögerung d für

$$\mathbf{x}(r) = \begin{pmatrix} \mathbf{c}_{\text{LPCC}_{12}}(r) \\ \mathbf{c}_{\text{LPCC}_{12}}(r+d) \\ \mathbf{c}_{\text{LPCC}_{12}}(r-d) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \quad (4.9)$$

aufgetragen.

Wie zu erwarten war, ist die optimale Zeitverzögerung gleich der Länge des verwendeten Zeitfensters in Sekunden ($L/(\nu_s T) = 6\text{s}$), da so Überlappungen oder Leerräume vermieden werden.

Variation der Hauptkomponentenanzahl

Die Erkennungsrate bei variiert Hauptkomponentenanzahl der LPCC-Transformation ist in Abb. 4.10 für

$$\mathbf{x}(r) = \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{LPCC}}(r+6) \\ \mathbf{v}_{\text{LPCC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \quad (4.10)$$

mit

$$\mathbf{v}_{\text{LPCC}}(r) = \text{PCA}(\mathbf{c}_{\text{LPCC}_{12}}(r), p) \quad (4.11)$$

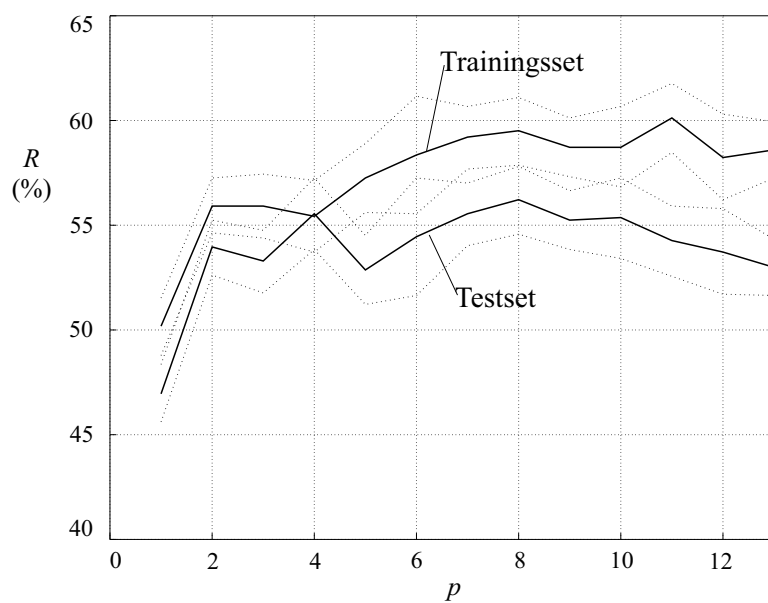


Abb. 4.10: Erkennungsrate R des Test- und Trainingssets bei variiert Hauptkomponentenanzahl p . Die dicken Linien stellen den Mittelwert, die dünnen Linien die Standardabweichung dar.

dargestellt.

Hier ist ein für ein NN typisches Ergebnis zu sehen. Die Erkennungsrate steigt zuerst mit zunehmender Hauptkomponentenanzahl an. Ab einem kritischen Wert sinkt sie jedoch wieder, da die Anzahl der Komponenten im Eingangsvektor zu groß wird und der Informationsgehalt dieser Hauptkomponenten zu gering ist. Der optimale Wert liegt bei etwa acht Hauptkomponenten.

Mit Tab. 4.3 ergibt sich die optimale Zusammensetzung des Eingangsvektors damit zu

$$\mathbf{x}(r) = \begin{pmatrix} \mathbf{v}_{\text{LPCC}}(r) \\ \mathbf{v}_{\text{LPCC}}(r+6) \\ \mathbf{v}_{\text{LPCC}}(r-6) \\ \mathbf{v}_{\text{OX}}(r) \\ \mathbf{v}_{\text{HR}}(r) \end{pmatrix} \quad (4.12)$$

mit

$$\mathbf{v}_{\text{LPCC}}(r) = \text{PCA}(\mathbf{c}_{\text{LPCC}_{12}}(r), 8). \quad (4.13)$$

Dieser Eingangsvektor wird für alle weiteren Auswertungen herangezogen.

4.5 Zusammensetzung der Zielvektoren

Um die optimale Darstellung des Zielvektors \mathbf{t} zu ermitteln, wurde ein NN mit allen drei in Kapitel 2.6 vorgestellten Möglichkeiten trainiert. Als Eingangsvektor wurde die optimale Konfiguration aus (4.12) verwendet. Die Vorgangsweise ist analog zu der in Kapitel 4.4, das NN besitzt jedoch 20 Neuronen. In Tab. 4.4 sind die ermittelten Werte für das Testset angeführt.

Aus diesen Ergebnissen geht klar hervor, daß die optimale Kodierung des Zielvektors durch

$$\mathbf{t} = \mathbf{t}_2 \quad (4.14)$$

entsprechend (2.63) dargestellt wird. Dieser Zielvektor wird nun für alle folgenden Analysen verwendet.

\mathbf{t}	Gesamt	NB	SI	CA	OA
\mathbf{t}_1	59.0 ± 4.4	58.7 ± 10.3	59.1 ± 9.9	58.2 ± 15.0	60.0 ± 9.5
\mathbf{t}_2	61.6 ± 5.1	65.3 ± 9.8	59.3 ± 11.3	62.9 ± 16.2	58.7 ± 13.1
\mathbf{t}_3	59.3 ± 5.5	76.6 ± 8.4	53.7 ± 10.8	50.7 ± 18.1	56.3 ± 10.4

Tab. 4.4: Klassifikationsergebnisse eines RBFN mit einer LVQ-Eingangsschicht bestehend aus 20 Neuronen. Die Ergebnisse wurden durch zehnfache Kreuzvalidierung und viermalige Neuinitialisierung des NN ermittelt.

4.6 Struktur des Neuralen Netzes

Nachdem in den vorhergehenden Kapiteln optimale Kodierungsformen für den Eingangsvektor und den Zielvektor gefunden wurden, soll nun unter deren Verwendung eine optimale Struktur des NN bestimmt werden. Dazu wird das Datenmaterial aus der Patientengruppe Säuglinge verwendet. Alle Ergebnisse werden mittels zehnfacher Kreuzvalidierung ermittelt, die Segmentierung wird auf Vektorbasis durchgeführt.

4.6.1 Multi Layer Perceptron

Das verwendete MLP besteht aus drei Schichten und wird entsprechend dem in Kapitel 3.1.1 beschriebenen durch Mehrfachinitialisierung und Early Stopping modifizierten Backpropagation Algorithmus trainiert. Für die Mehrfachinitialisierung werden 20 Trainingsepochen und für den endgültigen Lernvorgang maximal 500 Epochen verwendet.

Ab einem Anstieg der Fehlerrate des Kontrollsets um 10 Prozent über das Minimum wird der Lernvorgang abgebrochen. Die Gewichte dieses Minimums werden zur Bestimmung der Erkennungsraten des Test- und Trainingssets verwendet.

In Abb. 4.11 ist der Verlauf der Erkennungsrate bei variiertem Neuronenanzahl der ersten (k_1) und zweiten Schicht (k_2) zu sehen. Die dritte Schicht bestand aus $k_3 = 4$ Neuronen, die zweite Schicht aus $k_2 = (k_3 + k_1)/2$ Neuronen. Mit einer Konfiguration von 8-6-4 für k_1 - k_2 - k_3 wurden die besten Erkennungsraten erzielt, diese sind in Tab. 4.5 angegeben.

Der Verlauf des Fehlers \tilde{E} über die Anzahl der Trainingsintervalle bei der vorher als optimal ermittelten Konfiguration von 8-6-4 ist in Abb. 4.12 dargestellt. Das Minimum

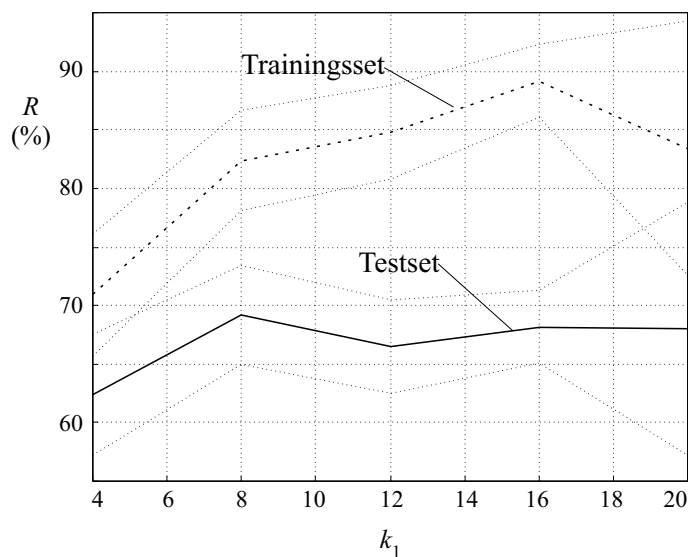


Abb. 4.11: Erkennungsrate R eines MLP bei variiertem Neuronenanzahl der ersten und zweiten Schicht. Die Eingangs- und Zielvektoren wurden entsprechend (4.12) und (4.14) gewählt. k_1 ist die Neuronenanzahl der ersten Schicht, $k_2 = (k_3 + k_1)/2$ und $k_3 = 4$.

Gesamt	NB	SI	CA	OA
69.2 ± 5.5	63.2 ± 11.1	70.7 ± 10.0	72.5 ± 7.8	70.4 ± 9.1

Tab. 4.5: Klassifikationsergebnisse eines MLP mit 3 Schichten und 8-6-4 Konfiguration. Die Eingangs- und Zielvektoren wurden entsprechend (4.12) und (4.14) gewählt.

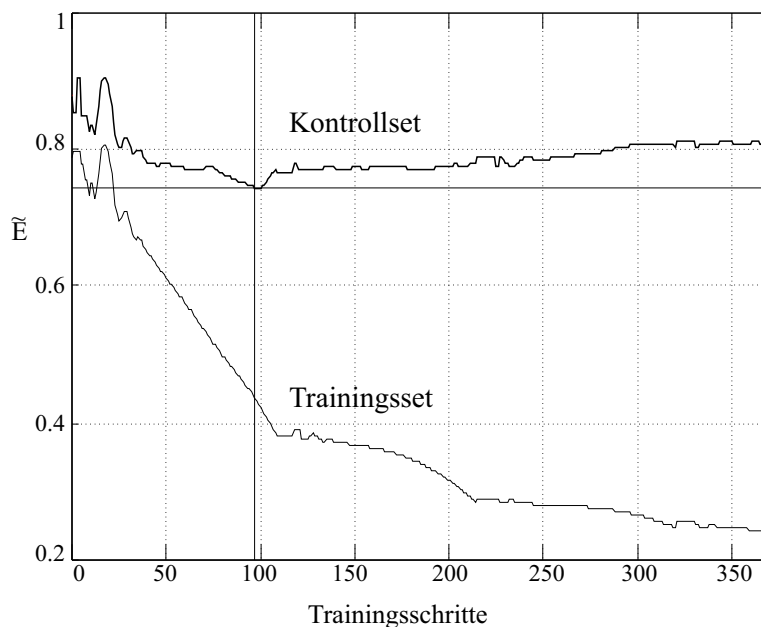


Abb. 4.12: Verlauf des normierten Fehlers \tilde{E} über die Anzahl der Trainingsschritte für das Trainingsset und das Kontrollset. Das Minimum ist hier schon nach ca. 100 Trainingsschritten erreicht.

ist schon nach ca. 100 Trainingsschritten erreicht und macht die Notwendigkeit des Early-Stoppings deutlich.

4.6.2 Radial Basis Function Network

In Kapitel 3.1.2 wurden die theoretischen Grundlagen eines RBFN vorgestellt. Hier sollen nun die beiden Varianten eines RBFN mit einer LVQ- bzw. einer SOM-Schicht verglichen werden. Am Beispiel des RBFN mit LVQ-Schicht wird auch die Auswirkung einer Variation des Spreads veranschaulicht.

RBFN mit LVQ-Schicht

Berechnung des Spreads - In Abb. 4.13 ist die Erkennungsrate eines RBFN mit LVQ-Schicht bestehend aus acht Neuronen und variiertem Spread σ dargestellt (sonst Standardauswertung wie bisher). Oberhalb eines Schwellwertes ist die Wahl von σ unkritisch für die Erkennungsrate. Als waagrechte Linie ist das Ergebnis für ein NN eingezeichnet, bei dem der Spread nach (3.13) für jedes Neuron getrennt bestimmt wurde. Wie auch in der detaillierten Auswertung der einzelnen Ereignistypen in Abb. 4.14 zu sehen ist, stellt die

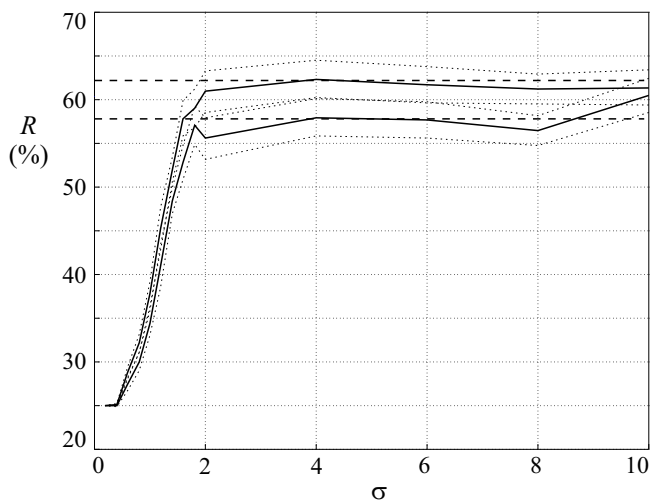


Abb. 4.13: Erkennungsrate R eines RBFN mit LVQ-Schicht bestehend aus acht Neuronen bei Variierung des Spread σ gleichzeitig für alle Neuronen. Die waagrechte Linie resultiert aus der Wahl des Spreads nach (3.13) individuell für jedes Neuron.

Wahl des Spreads durch (3.13) ein Optimum dar.

Variierung der Neuronenanzahl - In Abb. 4.15 ist die Erkennungsrate eines RBFN mit LVQ-Schicht bei variiertem Neuronenanzahl dargestellt. Die Zunahme der Erkennungsrate flacht ab einer bestimmten Grenze ab, und die Differenz zwischen den Erkennungsraten des Trainings- und Testsets nimmt zu. Ein deutliches Overfitting wie beim MLP ist nicht zu erkennen. Dies ist auch auf die individuelle Wahl des Spreads laut (3.13) für jedes Neuron zurückzuführen, da vor allem ein zu klein gewählter Spread Overfitting bei einem RBFN begünstigt.

Die Anzahl der Trainingsschritte wird proportional zur Neuronenanzahl entsprechend

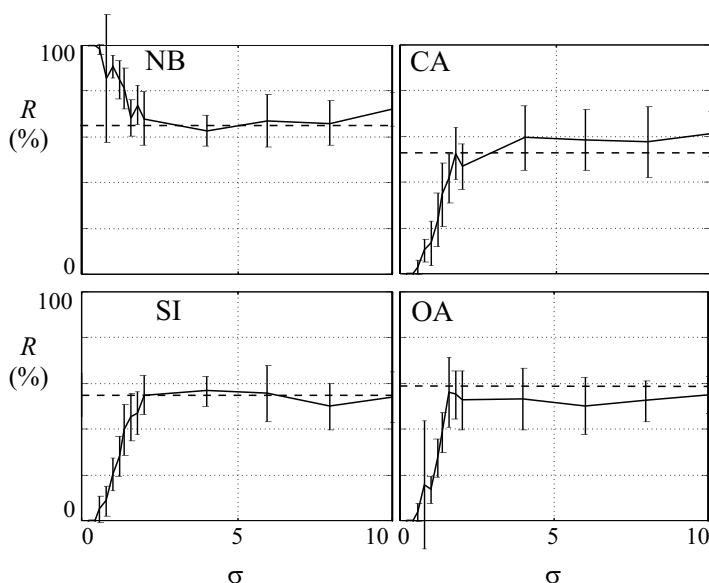


Abb. 4.14: Erkennungsrate R eines RBFN mit LVQ-Schicht bestehend aus acht Neuronen getrennt nach Ereignistyp bei Variierung des Spread σ gleichzeitig für alle Neuronen. Die waagrechte Linie resultiert aus der Wahl des Spreads nach (3.13) individuell für jedes Neuron.

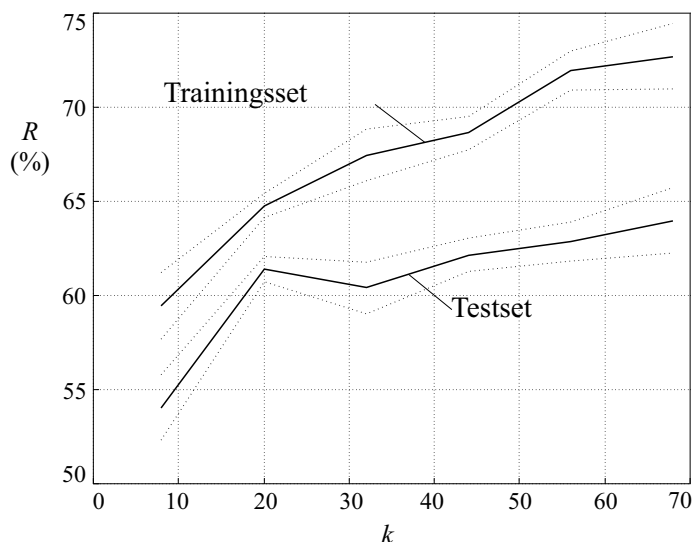


Abb. 4.15: Erkennungsrate R eines RBFN mit LVQ-Schicht bei variiertem Neuronenanzahl k .

$k * M/2$ erhöht, um die Varianz der Ergebnisse konstant zu halten. Hierbei bedeutet k die Anzahl der Neuronen und M die Anzahl der Eingangsvektoren. Als Kompromiß zwischen Rechenzeitbedarf und Erkennungsrate wurde die Neuronenanzahl $k = 20$ gewählt. Dies ergibt die in Tab. 4.6 angeführten Ergebnisse des Testsets.

RBFN mit SOM-Schicht

Ein RBFN mit SOM-Schicht ist dem vorher behandelten mit LVQ-Schicht sehr ähnlich. Wie in Kapitel 3.1.2 schon erwähnt wurde, ist aber bei diesem Typ auch die Geometrie der Neuronenanordnung von Bedeutung. Aus diesem Grund wurden eine zweidimensionale quadratische und eine eindimensionale Neuronenanordnung getestet. Wie aus Abb. 4.16 hervorgeht, ist die zweidimensionale Anordnung der eindimensionalen vorzuziehen. Die Erkennungsrate im Vergleich mit einer gleich großen LVQ-Schicht ist jedoch geringer. Dies kann durch den unüberwachten Lernvorgang bei der SOM erklärt werden.

In Tab. 4.7 sind die Ergebnisse eines RBFN mit SOM-Schicht bestehend aus 25 Neuronen angeführt. Obwohl die Neuronenanzahl höher ist als bei dem im vorhergehenden Kapitel beschriebenen RBFN mit LVQ-Schicht, ist die Erkennungsrate geringer.

Gesamt	NB	SI	CA	OA
60.2 ± 2.2	63.0 ± 12.5	59.8 ± 11.0	63.0 ± 17.1	55.0 ± 13.8

Tab. 4.6: Klassifikationsergebnisse eines RBFN mit LVQ-Schicht bestehend aus 20 Neuronen. Die Eingangs- und Zielvektoren wurden entsprechend (4.12) und (4.14) gewählt.

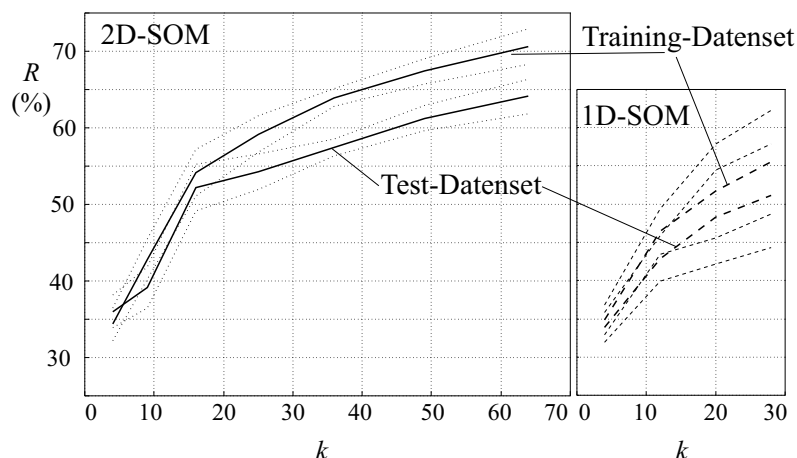


Abb. 4.16: Erkennungsrate R eines RBFN mit SOM-Schicht, bei ein- und zweidimensionaler Geometrie und variierter Neuronenzahl k .

Gesamt	NB	SI	CA	OA
54.2 ± 6.3	56.7 ± 6.8	52.4 ± 15.1	52.5 ± 12.1	55.5 ± 13.3

Tab. 4.7: Klassifikationsergebnisse eines RBFN mit quadratischer SOM-Schicht bestehend aus 25 Neuronen. Die Eingangs- und Zielvektoren wurden entsprechend (4.12) und (4.14) gewählt.

4.7 Verschiedene Kreuzvalidierungs-Strategien

Aus den Ergebnissen des vorhergehenden Kapitels wurden das MLP und das RBFN mit LVQ-Schicht als die beiden am besten geeigneten Netztopologien ausgewählt. In Tab. 4.8 sind die Ergebnisse dieser NNs bei der für die Kreuzvalidierung notwendigen Segmentierung des Datenmaterials der Patientengruppe Säuglinge auf Vektor- bzw. Ereignisbasis, angeführt. Die Eingangs- und Zielvektoren werden entsprechend (4.12) und (4.14) gewählt. Wie schon in Kapitel 3.2.1 ausgeführt, sinkt durch die Segmentierung auf Ereignisebene die Erkennungsrate, die Varianz steigt an. Dieses Verhalten ist bei beiden NNs zu beobachten.

Netz	Segm.	Gesamt	NB	SI	CA	OA
MLP	Vektor	63.9 ± 4.1	57.4 ± 13.4	64.6 ± 13.6	65.5 ± 10.3	68.0 ± 9.2
MLP	Ereignis	50.7 ± 8.1	48.9 ± 18.1	54.8 ± 17.1	52.7 ± 18.9	46.4 ± 13.8
RBFN-LVQ	Vektor	62.6 ± 6.8	65.8 ± 11.2	62.2 ± 12.7	65.2 ± 6.8	57.2 ± 8.3
RBFN-LVQ	Ereignis	47.7 ± 10.7	55.5 ± 17.4	45.4 ± 22.1	48.0 ± 21.0	41.7 ± 21.9

Tab. 4.8: Klassifikationsergebnisse des RBFN mit LVQ-Schicht und des MLP bei Segmentierung auf Vektor- bzw. Ereignisbasis. Die Eingangs- und Zielvektoren wurden entsprechend (4.12) und (4.14) gewählt (Säuglinge).

Eine Segmentierung auf Patientenbasis konnte wegen unzureichenden Datenmaterials nicht durchgeführt werden.

Patientengruppe Erwachsene

Alle bisher präsentierten Ergebnisse bezogen sich immer auf die Patientengruppe Säuglinge, hier werden nun die Ergebnisse für die Patientengruppe Erwachsene in Tab. 4.9 angeführt. Die Eingangs- und Zielvektoren werden entsprechend (4.12) und (4.14) gewählt. Die für die Patientengruppe Erwachsene verwendeten NN unterscheiden sich in der Neuronenzahl, da bei den Erwachsenen nur drei Ereignistypen untersucht werden. Als optimale Konfigurationen wurden für das RBFN mit LVQ-Schicht 18 Neuronen und für das MLP eine 8-6-3 Struktur ermittelt.

Komitee aus Neuralen Netzen

Während des Kreuzvalidierungsvorganges werden zehn verschiedene Netze trainiert. Diese einzelnen Realisierungen können als eigenständig betrachtet werden und somit ein Komitee aus NNs bilden (entsprechend (3.14)). Zum Test dieses Komitees wird ein weiteres unabhängiges Datenset verwendet. Beim RBFN mit LVQ-Schicht ist dies das Kontrollset, das 20 Prozent des Datenmaterials beinhaltet. Beim MLP wird ein viertes Datenset, das Early-Stoppingset, eingeführt, um das Kontrollset zum Testen des Komitees verwenden zu können. Das Early-Stoppingset und das Kontrollset beinhalten in diesem Fall jeweils 10 Prozent des Datenmaterials.

In Tab. 4.10 sind die Ergebnisse für die Patientengruppe Säuglinge angeführt, die durch ein Komitee von zehn NNs ermittelt wurden. In der Spalte „Durchschnitt“ ist der Mittelwert der Erkennungsrate über alle Einzelnetze angeführt, dieser Wert liegt in dem Schwankungsbereich, der durch den vorangegangenen Kreuzvalidierungsvorgang ermittelt worden ist. Die Erkennungsrate des Komitees ist in der Spalte „Komitee“ angeführt. Weiters sind auch die Erkennungsraten des Komitees für die einzelnen Kategorien angeführt. In Tab. 4.11 sind die entsprechenden Ergebnisse für die Patientengruppe Erwachsene dargestellt.

Die in Tab. 4.10 und 4.11 angeführten Ergebnisse zeigen, daß sich die Verbesserung der Erkennungsrate durch die durchgeführte Bildung eines Komitees auf maximal zwei

Netz	Segm.	Gesamt	NB	CA	OA
MLP	Vektor	70.4 ± 2.1	71.4 ± 4.6	75.2 ± 4.6	64.6 ± 5.0
MLP	Ereignis	62.6 ± 4.4	73.4 ± 8.0	63.7 ± 8.4	50.6 ± 14.4
RBFN-LVQ	Vektor	64.4 ± 3.4	69.1 ± 6.4	66.6 ± 7.4	57.5 ± 4.4
RBFN-LVQ	Ereignis	58.5 ± 4.5	62.5 ± 7.7	59.6 ± 8.3	53.5 ± 9.6

Tab. 4.9: Klassifikationsergebnisse der Patientengruppe Erwachsene bei unterschiedlicher Segmentierung des Datenmaterials.

Netz	Segm.	Durchschnitt	Komitee	NB	SI	CA	OA
MLP	Vektor	61.8 ± 4.6	65.5	58.6	72.4	69.0	62.1
MLP	Ereignis	42.5 ± 5.2	47.2	51.9	37.0	51.9	48.2
RBFN-LVQ	Vektor	59.1 ± 2.4	60.8	67.2	63.8	62.1	50.0
RBFN-LVQ	Ereignis	60.0 ± 2.0	64.3	64.3	53.6	75.0	64.3

Tab. 4.10: Klassifikationsergebnisse eines Komitees aus zehn RBFNs mit LVQ-Schicht bzw. zehn MLPs bei Segmentierung auf Vektor- bzw. Ereignisbasis für die Patientengruppe Säuglinge. In der Spalte Durchschnitt ist der Mittelwert der Erkennungsrate über die einzelnen Netze angeführt.

Netz	Segm.	Durchschnitt	Komitee	NB	CA	OA
MLP	Vektor	70.1 ± 1.7	73.4	73.4	79.7	67.2
MLP	Ereignis	60.2 ± 2.0	61.7	75.8	63.4	45.8
RBFN-LVQ	Vektor	62.8 ± 1.2	64.8	65.7	69.8	59.0
RBFN-LVQ	Ereignis	62.3 ± 2.2	65.1	68.7	65.5	61.0

Tab. 4.11: Klassifikationsergebnisse eines Komitees aus zehn RBFNs mit LVQ-Schicht bzw. zehn MLPs bei Segmentierung auf Vektor- bzw. Ereignisbasis für die Patientengruppe Erwachsene. In der Spalte Durchschnitt ist der Mittelwert der Erkennungsrate über die einzelnen Netze angeführt.

bis drei Prozent beschränkt. Auch eine ausführliche Analyse eines Komitees aus RBFNs mit LVQ-Schicht durch Kreuzvalidierung unterstützt dieses Ergebnis. Hierbei wurden zehn Netze zehnfach kreuzvalidiert und dabei jeweils auch das Kontrollset, mit dem das Komitee getestet wird, zyklisch verändert. Für das Komitee ergab sich eine Gesamterkennungsrate von $53.8\% \pm 6.2\%$ im Vergleich zu $51.9\% \pm 6.5\%$ für das durchschnittliche Netz.

In Abb. 4.17 bis 4.20 sind die typischen Zeitverläufe der Vitalparameter für die Ereignisse NB, SI, CA und OA abgebildet. Weiters ist die dafür ermittelte Klassifizierung durch ein Komitee aus zehn RBFNs mit LVQ-Schicht bestehend aus 20 Neuronen dargestellt. Die Segmentierung des Datenmaterials wurde auf Ereignisebene durchgeführt, die verwendeten Ereignisse stammten aus dem Kontrollset.

Die Gegenüberstellung der Ergebnisse in Tab. 4.8 und 4.9 für das MLP und das RBFN mit LVQ-Schicht zeigt, daß die Erkennungsraten des MLP um ca. 4 % besser sind als die des RBFN mit LVQ-Schicht. Beiden Netzen gemeinsam ist, daß die Erkennungsrate deutlich sinkt, wenn das Datenmaterial auf Ereignisbasis anstatt auf Vektorbasis segmentiert wird.

Unabhängig davon bietet das RBFN folgende Vorteile:

- Kein Overfitting des Trainingsalgorithmus, daher ist auch kein Early-Stopping notwendig.

- Die Varianz der Trainingsergebnisse kann durch Erhöhung der Trainingsschrittzahl gezielt gesenkt werden.
- Schnelles, zweistufiges Lernverfahren. Dadurch ist die Kombination mit einem unüberwachten Lernverfahren, z.B. einer SOM, durchführbar.
- Eine weitere Verbesserung der Erkennungsrate ist durch Erhöhung der Neuronenzahl möglich (siehe Abb. 4.15).

Für das MLP sprechen folgende Punkte:

- Höhere Erkennungsrate.
- Geringere Anzahl freier Parameter im Vergleich zu einem RBFN bei gleicher Erkennungsrate.
- Einstufiger Lernvorgang, daher enge Kopplung der einzelnen Schichten.

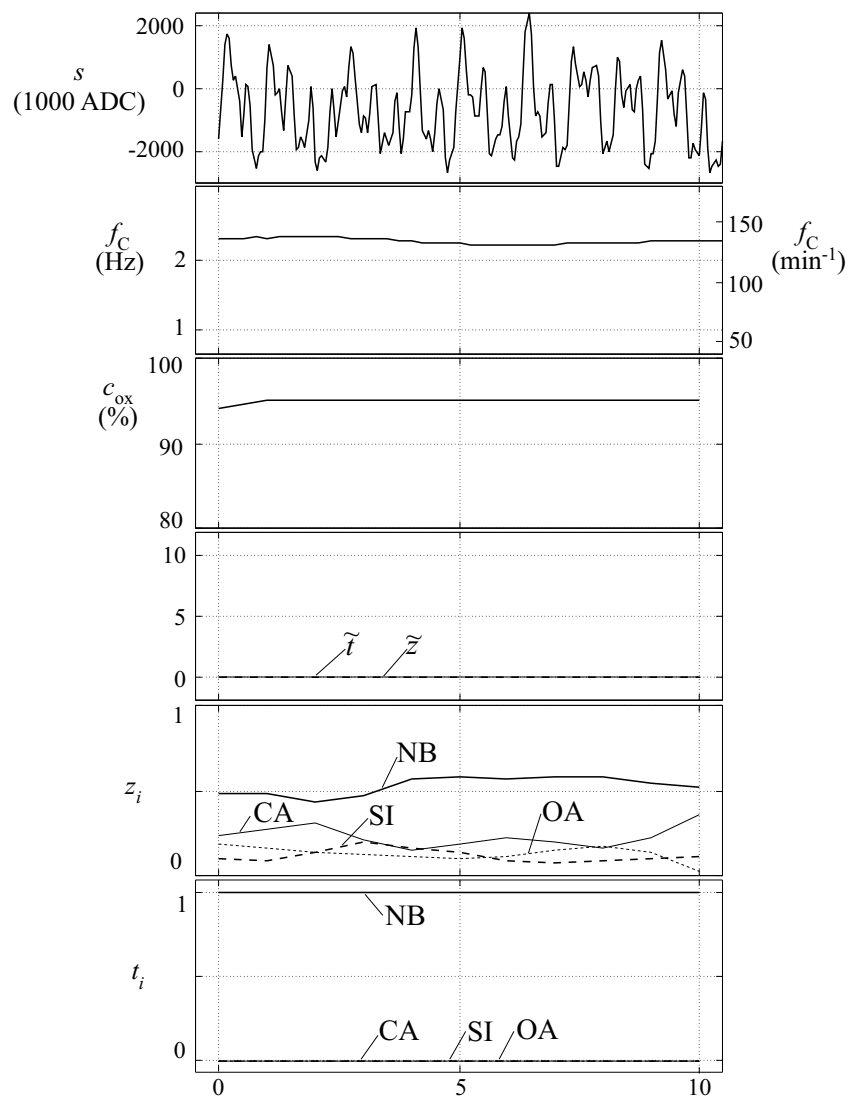


Abb. 4.17: Klassifikation eines NB-Ereignisses durch ein Komitee aus RBFNs mit LVQ-Schicht. s EFPG-Signal, f_C Herzfrequenzsignal, c_{ox} Oxymetriesignal, z_i Ausgänge des NN, t_i Zielwerte für das NN, \tilde{t} Klassifikationssignal (Zielklassifikation), \tilde{z} Ausgangsklassifikation.

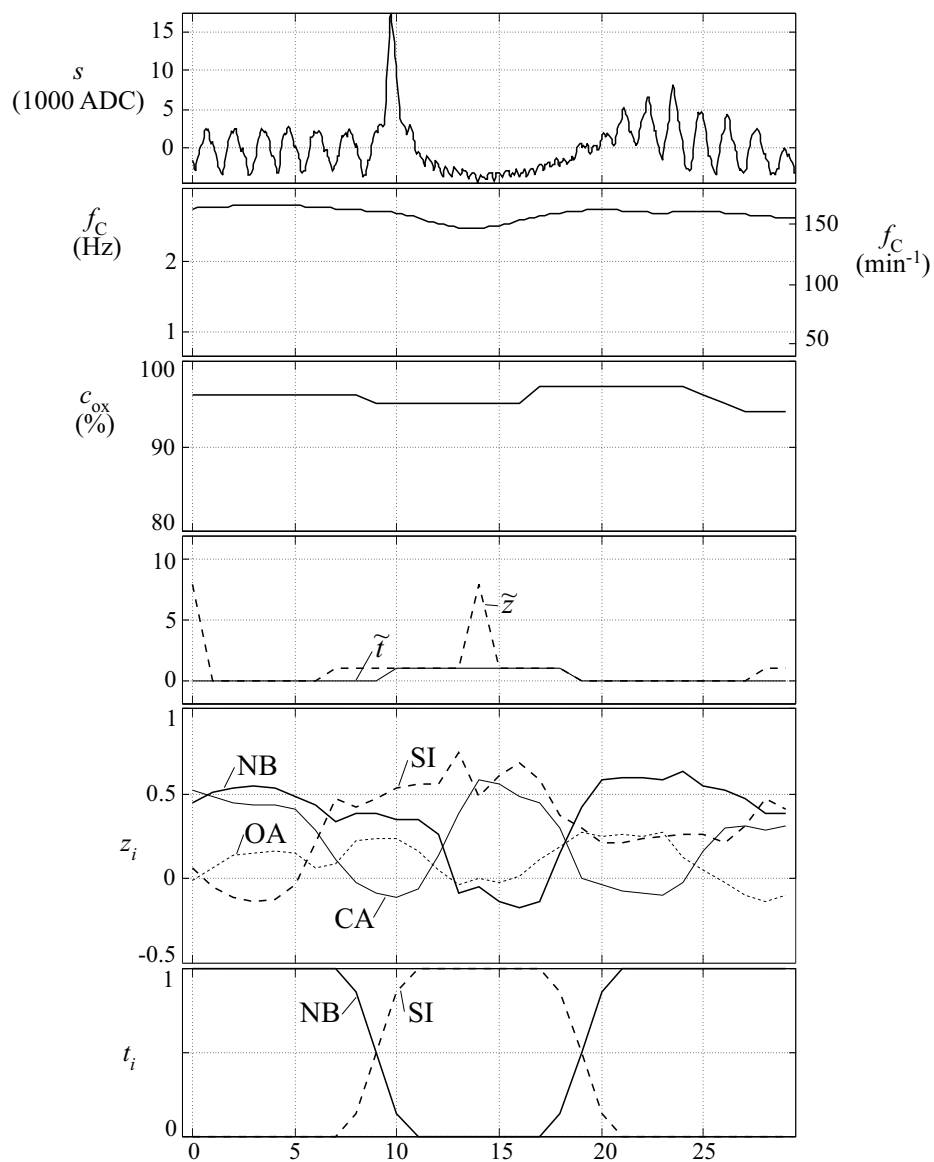


Abb. 4.18: Klassifikation eines SI-Ereignisses durch ein Komitee aus RBFNs mit LVQ-Schicht. s EFPG-Signal, f_C Herzfrequenzsignal, c_{ox} Oxymetriesignal, z_i Ausgänge des NN, t_i Zielwerte für das NN, \tilde{t} Klassifikationssignal (Zielklassifikation), \tilde{z} Ausgangsklassifikation.

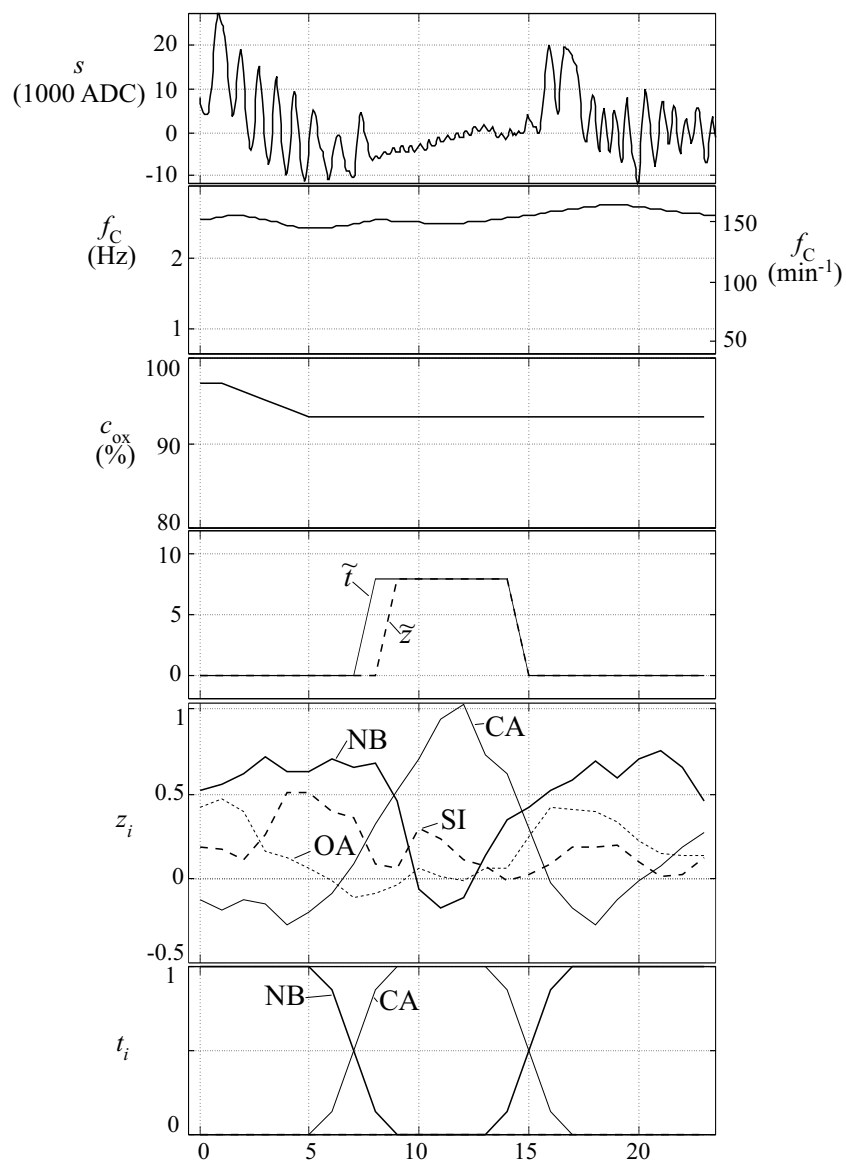


Abb. 4.19: Klassifikation eines CA-Ereignisses durch ein Komitee aus RBFNs mit LVQ-Schicht. s EFPG-Signal, f_C Herzfrequenzsignal, c_{ox} Oxymetriesignal, z_i Ausgänge des NN, t_i Zielwerte für das NN, \tilde{t} Klassifikationssignal (Zielklassifikation), \tilde{z} Ausgangsklassifikation.

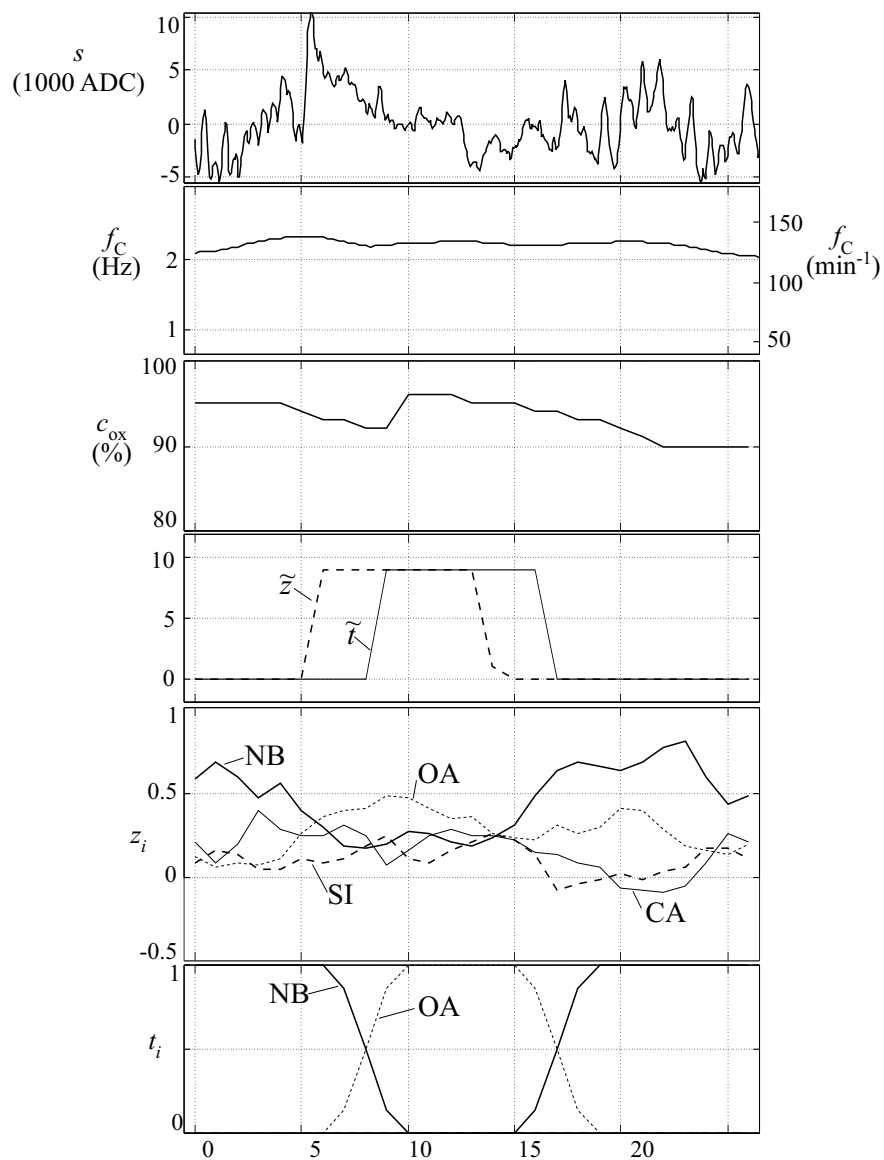


Abb. 4.20: Klassifikation eines OA-Ereignisses durch ein Komitee aus RBFNs mit LVQ-Schicht. s EFG-Signal, f_C Herzfrequenzsignal, c_{ox} Oxymetriesignal, z_i Ausgänge des NN, t_i Zielwerte für das NN, \tilde{t} Klassifikationssignal (Zielklassifikation), \tilde{z} Ausgangsklassifikation.

Literaturverzeichnis

- [1] Christopher M. Bishop. Neural Networks for Pattern Recognition. Clarendon Press, Oxford, 1995.
- [2] Johann Zauner, Josef Falkner, Rudolf Leitgeb, Bernhard Stöbich, Rüdiger Landgraf und Michael Mairinger. Heimmonitoringgerät zur Erfassung und Speicherung von Schlafapnoeparametern, Teile A-E. Diplomarbeiten an der TU-Wien, Wien, 1995.
- [3] Gernot Schmid. EFPG-Monitoring klinisch relevanter Apnoen. Diplomarbeit an der TU Wien, Wien, 1997.
- [4] Christoph Ruhsam. Digitale Verarbeitung feldplethysmographischer Signale zur simultanen Erfassung kardiorespiratorischer Funktion. Dissertation an der TU Wien, Wien, 1994.
- [5] Markus Clabian, H. Pfützner. MATLAB zur Analyse physiologischer Signale. MATLAB und SIMULINK-Entwicklung und Anwendung. Seminarunterlagen S40, TU-Wien, 1997.
- [6] Alan V. Oppenheim und Roland W. Schaffer. Discrete-Time Signal Processing. Prentice-Hall, 1989.
- [7] Thomas P. Krauss, Loren Shure, John N. Little. Signal Processing Toolbox User's Guide. The MathWorks, Inc., 1994.
- [8] Ananthram Swami, Jerry M. Mendel, Chrysostomos L. (Max) Nikias. Higher Order Spectral Analysis Toolbox User's Guide. The MathWorks, Inc., 1995.
- [9] Frank Fallside, William A. Woods. Computer Speech Processing. Prentice Hall, 1985.
- [10] Manfred R. Schroeder. Direct (Nonrecursive) Relation Between Cepstrum and Predictor Coefficients. IEEE Trans. Acoust., Speech, Signal Processing, pp. 297-301, 1981.
- [11] Michael Misiti, Yves Misiti, Georges Oppenheim, Jean-Michel Poggi. Wavelet Toolbox User's Guide. The MathWorks, Inc., 1996.
- [12] I. T. Jolliffe. Principal Component Analysis. Springer-Verlag, New-York, 1986.

- [13] Howard Demuth, Mark Beale. Neural Network Toolbox User's Guide. The MathWorks, Inc., 1994.
- [14] Teuvo Kohonen. Self-Organization and Associative Memory. Springer-Verlag, Berlin, 1989.
- [15] Arthur Flexer. Statistical Evaluation of Neuronal Network Experiments: Minimum Requirements and Current Practice, Wien, 1996.
- [16] Monika Köhle, Neurale Netze. Springer-Verlag, Wien - New York, 1990.
- [17] Raúl Rojas. Theorie der neuronalen Netze. Springer-Verlag, Berlin, 1993.

Anhang - Software

A.1 Installation

Alle Programmdateien für Matlab müssen im Verzeichnis `c:\diploma\apnoe\` installiert sein. Weiters muß das Verzeichnis `c:\diploma\versuche\` existieren.

A.2 Typischer Ablauf eines Berechnungszyklus

CrEvLst: Liest die Ereignislisten aus einem spezifizierbaren Verzeichnis und erstellt dafür die Ereignislisten, die im selben Verzeichnis gespeichert werden. Die Einstellungen für die Daten aus Mödling bzw. Lainz müssen im Quelltext durch Entfernen der entsprechenden Kommentare angepaßt werden.

CrDaSet: Die spezifizierten Ereignislisten werden eingelesen und eine daraus getroffene möglichst gleich verteilte Zufallsauswahl wieder gesichert. Die Parameter für die Daten aus Mödling bzw. Lainz müssen im Quelltext durch Entfernen der entsprechenden Kommentare angepaßt werden.

EvLst: Ereignisliste bearbeiten, laden und speichern.

CrSelFx: Berechnet ausgewählte Features zur geladenen Ereignisliste. Die verfügbaren Features und veränderbaren Parameter sind in `CalcFeat` dokumentiert.

Parx: Erstellt eine Parameter-Struktur für die Funktion `BatchRun`. Die Felder der Struktur und ihre Bedeutung sind in `CvNNTop` definiert.

BatchRun: Verwendet die durch `Parx` erstellte Struktur, um verschiedene Trainingsläufe eines Neuren Netzes (*neural network*, NN) durchzuführen. Das Ergebnis wird als Struktur gespeichert und kann mittels `BatchRes` ausgewertet werden. Die für die einzelnen NN-Topologien verfügbaren Felder sind ebenfalls in `CvNNTop` angeführt.

BatchRes: Ergebnis von `BatchRun` analysieren und auswerten.

A.3 Kommandos

Alle in diesem Kapitel aufgelisteten Programme werden direkt als Kommandos ausgeführt. Die veränderten Variablen und benützten Unterprogramme sind im Quelltext oder über das Matlab-Kommando `help` zu finden.

BatchRun Training und Test des NN mit verschiedenen Parameterkonfigurationen durch Kreuzvalidierung.

BatchRes Auswerten eines Batch-Runs.

CrCv Berechnet den Positionsvektor und daraus die Kreuzvalidierungs-Matrizen zur Erzeugung des Trainings-, Test- und Kontrollsets.

CrDaSet Liest die Ereignislisten aus einer vorgegebenen Liste an Patienten und erstellt daraus eine gleichverteilte Zufallsauswahl.

CrEvLst Erzeugt Ereignislisten aus den vorgegebenen Patientendaten.

CrSelFx Berechnet die ausgewählten Features für die geladene Ereignisliste und speichert sie im Workspace.

EvLst Ermöglicht das interaktive Bearbeiten, Laden und Speichern einer Ereignisliste.

Parx Erstellt Parameter für `BatchRun`.

TvFx Berechnet Ziel- und Eingangsvektoren für ein NN.

A.4 Unterprogramme

Zu allen Unterprogrammen folgt eine kurze Beschreibung, weitere Information ist im Quelltext selbst oder über das Matlab Kommando `help` zu finden.

CalcFeat Berechnet die einzelnen Features.

CcepsW Berechnet das komplexe Cepstrum einer Matrix von Signalabschnitten.

Cep2Lpc Wandelt LPCC-Koeffizienten in LPC-Koeffizienten um.

ClRate Ermittelt die Erkennungsrate eines NN.

CrEvPos Berechnet den Positionsvektor aus der Ereignisliste.

CrFeat Berechnet eine Liste von Features.

CrossValEvent Berechnet die Kreuzvalidierungs-Matrizen, Aufteilung der Vektoren nach Ereigniszugehörigkeit.

CrossValVector Berechnet die Kreuzvalidierungs-Matrizen, Aufteilung auf Vektorbasis.

CrSet Zufallsauswahl von Ereignissen aus mehreren Ereignislisten.

CrVec Zufallsauswahl von Vektoren aus 2 korrespondierenden Matrizen.

CrNNTop Kreuzvalidiert ein NN, Topologie und diverse andere Parameter wählbar.

DeStand Berechnet die Umkehrung von **Stand**.

EditEvLst Dient zum Bearbeiten und Anzeigen der Ereignisse aus der Ereignislist.

Ind2VecF Wandelt Index in Vektor mit fixer Dimension um.

LdEvent Lädt angegebenes Ereignis.

LiesHea2 Interface zu EFPG-Dateien.

Lpc2Cep Wandelt LPC-Koeffizienten in LPCC-Koeffizienten um.

Lpc2Fft Wandelt LPC-Koeffizienten in Fourier-Spektrum um.

LpcW Berechnet LPC-Koeffizienten (inklusive Amplitude).

Pca Berechnet die Hauptkomponenten (*principal components*, PCs).

PcaCorKd Berechnet die PCs bei bekannter Korrelationsmatrix.

PcaCorr Berechnet die PCs mit Hilfe der Korrelationsmatrix.

PcaEfpg Berechnet die PCs speziell für zeitverzögerte Feature-Vektoren des EFPG-Signals.

PcaEfpgJoined Berechnet die PCs speziell für zeitverzögerte Feature-Vektoren des EFPG-Signals, kombinierte Variante.

PlotEv Darstellen eines Ereignisses.

PlotPca Zweidimensionale Darstellung der ersten beiden PCs, sowie Ausgabe der Eigenvektoren.

RbNorm Berechnet normierte Radiale Basis Funktion.

ReadEv Einlesen einer Ereignisliste.

Stand Matrix auf $\sigma = 1$ und $\mu = 0$ standardisieren.

StandKd Matrix bei bekannter Varianz und bekanntem Mittelwert standardisieren.

TomBpx3 Training eines MLP mit 3 Schichten und Early Stopping.

TPlotTr Darstellen des Fehlers während des Lernvorganges.

Transf Berechnet die mit der Fensterfunktion bewerteten Signalabschnitte zur weiteren Verarbeitung.

TrLvq Training einer LVQ-Schicht.

TrRb Training eines RBFN mit gleichem Spread für alle Neuronen.

TrRbS Training eines RBFN mit individuellem Spread für jedes Neuronen.

TrSom Training einer SOM-Schicht.

ttrainlvq Training eines LVQ-Netzes ohne Grafikausgabe.

WriteEv Schreibt Ereignisliste in eine Datei.