Generalisierte Radiale-Basisfunktionen-(GRBF-) Netzwerke — Biologische Motivation und mathematische Modellierung

Axel Wismüller

Institut für Radiologische Diagnostik,
Ludwig-Maximilians-Universität München,
Klinikum Innenstadt, Ziemssenstr. 1, 80336 München
e-mail: axel@wismueller.de

Inhaltsverzeichnis

Vorwort

Das vorliegende Skriptum skizziert wichtige theoretische Prinzipien der generalisierten Radiale-Basisfunktionen- (GRBF-) Netzwerke. Hierbei wird sowohl die biologische Motivation als auch die mathematische Modellierung dieses Architekturkonzepts beleuchtet. Zur Diskussion von verschiedenen Detailproblemen sei auf die Arbeiten [2], [9], [3], [4], [10] verwiesen.

1 Einführung

Mit künstlichen neuronalen Netzen wird versucht, biologische neuronale Netze als informationsverarbeitende Systeme nachzuahmen. Es sollen Automaten gebaut werden, die gewisse Aspekte der Informationsverarbeitung beim Menschen, wie z. B. Adaptivität, Fehlertoleranz und Lernfähigkeit, imitieren. Auch wenn die Funktionsweise des Gehirns und der Nervensysteme von Lebewesen trotz intensiver Forschungsanstrengungen bislang nicht annähernd vollständig erforscht werden konnte und sich die einzelnen biologischen Modelle in vielen Aspekten unterscheiden, herrscht allgemeine Übereinstimmung darüber, daß das Grundprinzip der Funktion des Nervensystems in der Kontrolle durch Kommunikation besteht.

Nervensysteme bestehen aus vielen Millionen — das menschliche Gehirn grob geschätzt aus mindestens zehn Milliarden — von miteinander vernetzten Nervenzellen, den sogenannten *Neuronen*. Jede dieser Zellen kann eintreffende Signale empfangen, auf vielfältige Weise verarbeiten und neu erzeugte Signale weitergeben. Der vielfach äußerst komplexe Aufbau des Einzelneurons ist in Abbildung 1 links dargestellt. Die Fortsätze des Neurons, die die Eingangsinformation sammeln, werden Dendriten genannt. Die mitunter mehreren zehntausend Dendriten eines Neurons sind baumartig angeordnet. Sie übernehmen an spezifischen Kontaktstellen, den Synapsen, Signale aus anderen Nervenzellen. Der Zellkörper (Soma) reagiert auf diesen Stimulus und beginnt, ein Ausgabesignal zu übertragen. Dieses Signal wird über das Axon und mehrere zehntausend Synapsen an ebensoviele Nervenzellen weitergeleitet.

Diese vier Elemente (Dendriten, Synapsen, Soma und Axon) bilden die minimale Struktur, die aus der Biologie in die Neuroinformatik übernommen wird, so daß auch künstliche Neuronen als informationsverarbeitende Elemente

- gerichtete, gewichtete Eingabeleitungen,
- einen Berechnungskörper und
- eine Ausgabeleitung

besitzen. Abbildung 1 rechts zeigt ein Beispiel für die Struktur eines typischen abstrakten Neurons. Jede Eingabeleitung überträgt einen reellen Wert x_i. Die Eingabeleitungen sind gewichtet, d. h. jeder Eingabewert x_i wird mit dem entsprechenden Gewicht w_{ji} multipliziert. Die im Berechnungskörper ankommende Information wird addiert, und das Resultat wird als Argument für die Aktivierungsfunktion a_j verwendet.

Die abstrakten Einzelneuronen lassen sich auf vielfältige Weise miteinander zu künstlichen neuronalen Netzwerken verknüpfen. Sie unterscheiden sich je nach Anzahl der verwendeten Neuronen, Verknüpfungsschema, gewählter Aktivitätsfunktion und in den Regeln für die Wahl und Modifikation der Gewichte. Auf diese Weise entstehen neuronale Netzwerke mit verschiedenen Eigenschaften.

In der aktuellen neurobiologischen Forschung entpuppt sich zunehmend die Komplexität des Einzelneurons — der Neuroinformatiker macht daraus eine *Karikatur*, Abb. 1. (Zu elementaren neurobiologischen und -physiologischen Grundlagen sei z. B. auf [30] oder [31] verwiesen, eine Zusammenfassung wichtiger Organisationsprinzipien des Zentralnervensystems findet sich in [34]. Eine umfassende weiterführende Darstellung ist [18].) Hier seien nur einige der üblichen Vereinfachungen aufgeführt:

- In biologischen Neuronen erfolgt die Informationsübertragung durch Sequenzen von Aktionspotentialen — im formalen Neuron des Neuroinformatikers herrscht dagegen das Vorurteil der **Frequenzcodierung**: Die Aktionspotentialfolge wird als Analogwert modelliert, der die momentane „Aktivität“ des Neurons widerspiegeln soll. Erst in jüngster Zeit wird die **Zeitstruktur** des Spikeverhaltens von Neuronen deutlicher in theoretische Modellbildungen einbezogen (siehe z. B. [12], [7], [8]).

- Der Dendritenbaum des Neuroinformatikers ist ein linearer raum-zeitlicher Integrator. Rückwirkungen der Zellaktivität in den Dendritenbaum werden meist nur im Sinne von Hebbschem Korrelationslernen [15] mit der vorgeschalteten Nervenzelle interpretiert. Denkbare komplizierte Nichtlinearitäten bei der Informationsverarbeitung — so bedeutsam sie biologisch vielleicht auch sein mögen — entziehen sich dagegen weitgehend der theoretischen Modellierung. Die Rolle des Dendriten

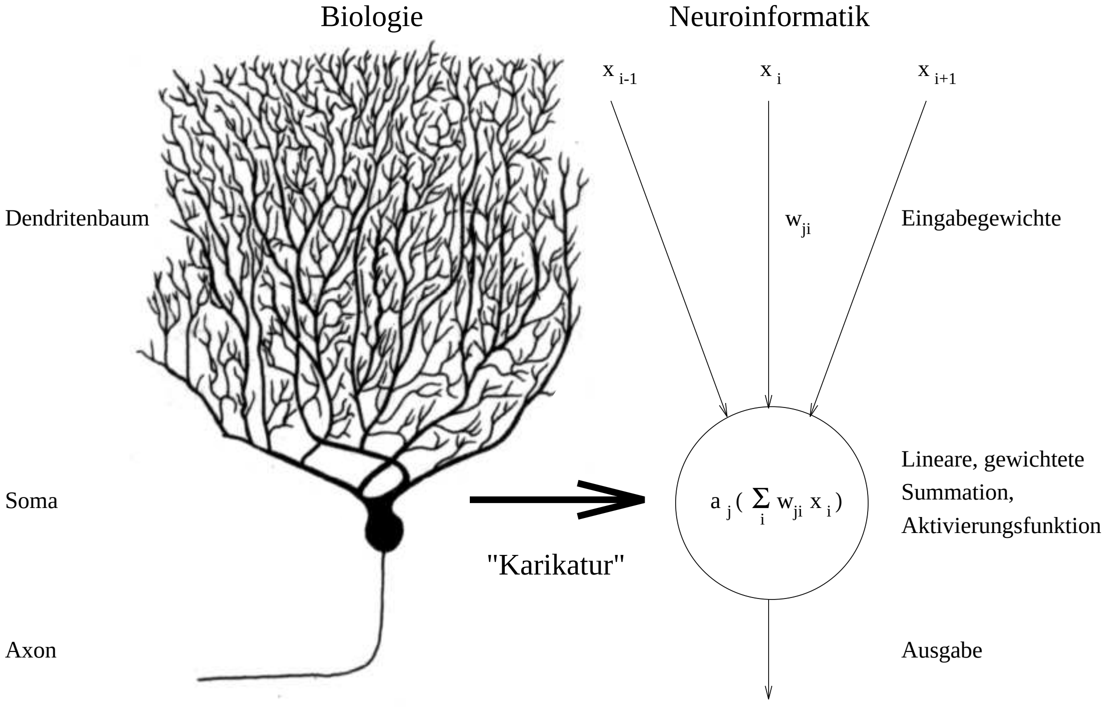


Abbildung 1: Das Neuron des Neuroinformatikers als „Karikatur" der biologischen Verhältnisse.

im Rahmen neuronaler Erregungsvorgänge zu erhellen, ist das Ziel aktueller Forschungsbemühungen, siehe z. B. [28], [32].

Trotz dieser Unzulänglichkeiten erweist sich die in Abb. 1 skizzierte „Karikatur" eines Neurons häufig als erste nützliche Näherung, um wenigstens grobe Modellvorstellungen zu Struktur-Funktionsbeziehungen in biologischen Nervensystemen zu entwickeln. Der überwältigende theoretische Erkenntnisgewinn der Neuroinformatik in den letzten Jahren sowie daraus resultierende Fortschritte in den Bereichen Mustererkennung und Datenanalyse mögen als nachträgliche Rechtfertigung für dieses Vorgehen dienen.

2 Biologische Motivation

Im folgenden wird die biologische Motivation von GRBF-Netzwerken im Detail erläutert. Der nicht im Hinblick auf biologische Modellierung interessierte Leser kann diesen Ab-

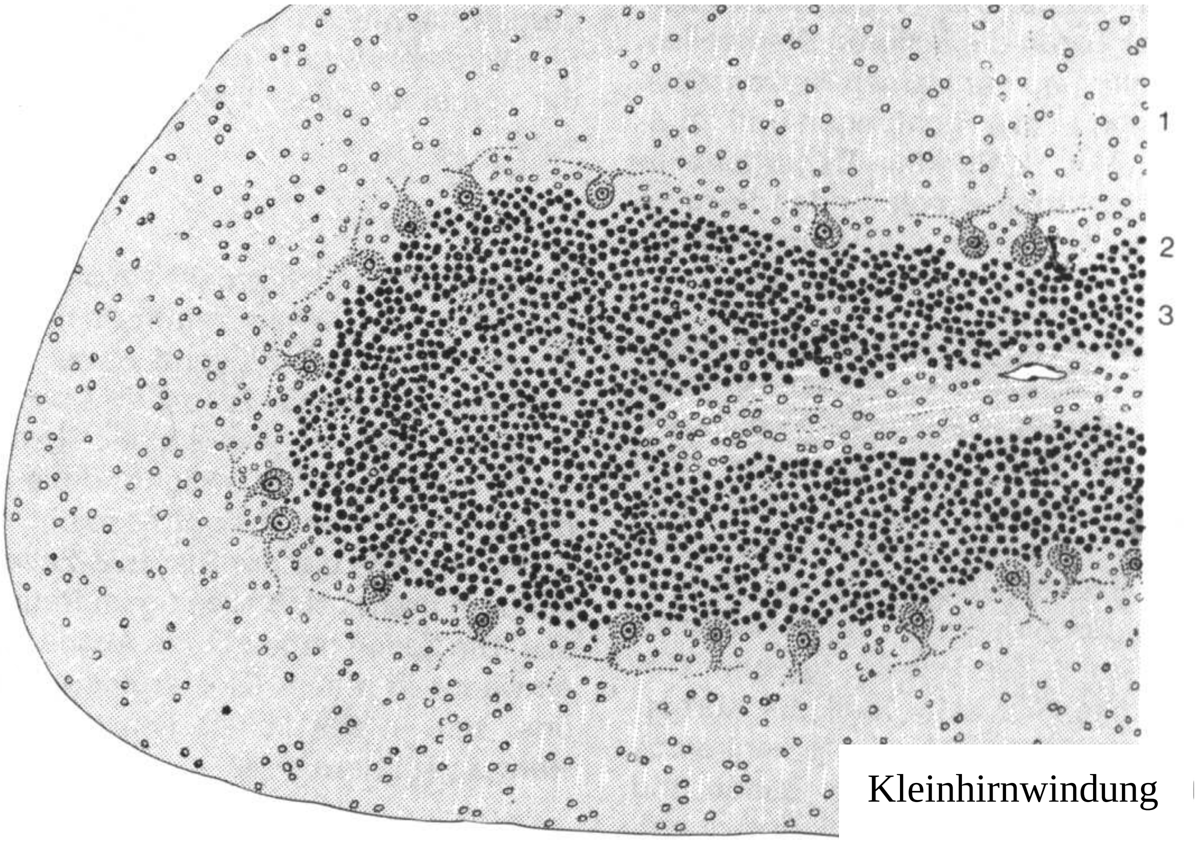


Abbildung 2: Mikroanatomischer Schnitt durch eine Kleinhirnwindung: 1 Stratum moleculare, 2 Stratum ganglionare, 3 Stratum granulare, siehe Text. Aus [17].

schnitt überspringen.

Ein charakteristisches Beispiel für den oben skizzierten Denkansatz zur Modellierung biologischer Nervensysteme ist der Versuch, neuroanatomische Erkenntnisse über den Aufbau der Kleinhirnrinde auf ein abstraktes Modell abzubilden — ein Netz von formalen Neuronen im Sinne der oben genannten „Neuronkarikaturen“:

Das Kleinhirn spielt eine wichtige Rolle bei der Feinabstimmung motorischer Aktionen, was bei Schädigungen (z. B. alkoholtoxisch, degenerativ, traumatisch etc.) eindrucksvoll als klinische Ausfallssymptomatik zu Tage tritt. (Nähere Erläuterungen müssen hier unterbleiben, zur Einführung siehe z. B. [1] oder [23].)

Das Kleinhirn kommuniziert mit zahlreichen anderen Hirnarealen. Hierbei läßt sich eine Reihe **„neuroanatomischer Grundregeln“** beobachten:

(i) Alle afferenten (zuführenden) Nervenfasern besitzen Endigungen im cerebellaren Cortex.

(ii) Afferente Fasern entsenden Kollateralen zu einem oder mehreren Kleinhirnkernen.

(iii) Die Efferenzen (wegführenden Nervenfasern) des cerebellaren Cortex werden ausschließlich von Axonen der sog. **Purkinje-Zellen** gebildet.

(iv) All diese Axone der Purkinje-Zellen enden in den Kleinhirnkernen (Ausnahme: cerebellovestibuläre Efferenzen).

(v) Alle Efferenzen des Kleinhirns werden von Axonen der Nervenzellen in den Kleinhirnkernen gebildet (Ausnahme: cerebellovestibuläre Efferenzen).

Die Kleinhirnrinde ist durch zahlreiche makroskopisch sichtbare Windungen und entsprechende Furchen gegliedert. Histologisch ist sie in allen Regionen gleichmäßig aufgebaut: In Abb. 2 erkennt man deutlich eine Dreischichtenstruktur (Diese ist nicht mit der Dreischichtenarchitektur der später beschriebenen abstrakten Modellierung zu verwechseln!). Eine äußere zellärmere Schicht **(Stratum moleculare)** besteht vorwiegend aus marklosen Axonen (der unten beschriebenen Körnerzellen). Sie enthält nur wenige weiter außen gelegene **äußere Sternzellen** und weiter innen gelegene **innere Sternzellen** (sog. **Korbzellen**). Das **Stratum ganglionare** trennt als dünne Schicht der großen sog. **Purkinje-Zellen** das Stratum moleculare von der inneren Körnerzellschicht, **Stratum granulare**. Sie ist außerordentlich zellreich und besteht aus dichtgepackten kleinen Nervenzellen, den sog. **Körnerzellen**. Vereinzelt finden sich dort die größeren, sog. **Golgi-Zellen**.

Den prinzipiellen mikroanatomischen Aufbau dieser Schichten zeigt Abb. 3: Der afferente Input der Kleinhirnrinde erfolgt durch zwei unterschiedliche Fasersysteme: **Moosfasern** und **Kletterfasern**.

Die Moosfasern entstammen einer Vielzahl anderer Kerngebiete, wie in Abb. 4 zu sehen ist (Abb. 5 zeigt die entsprechenden topographischen Verhältnisse.): Neben Afferenzen aus den Brückenkernen (5 und 6 in Abb. 4) gelangen Fasern aus der Medulla oblongata (14 in Abb. 4) und verschiedenen Kerngebieten des Rückenmarks als Moosfaser-Input zum Kleinhirn. Unter letzteren sind die spinocerebellären Bahnen (17 und 18 in Abb. 4) sowie die Hinterstrangafferenzen des Rückenmarks (19, 16, 15 in Abb. 4) bedeutsam. Die Moosfasern entsenden gemäß obiger Regel (ii) Kollateralen zu den Kleinhirnkernen (9 in Abb. 4, z. B. 11 in Abb. 5). Sie enden nach ausgeprägter Verzweigung mit excitatorischen Synapsen an den Körnerzellen, wobei eine Moosfaser mit ca. 450 Körnerzellen kommuniziert (1 in Abb. 3).

Das Kletterfasersystem ist in Abb. 6 dargestellt, entsprechende topographische Verhältnisse zeigt Abb. 7. Die Kletterfasern (14 in Abb. 6) entstammen Kerngebieten der unteren Olive (15, 16, 17 in Abb. 6, 16, 17, 18 in Abb. 7). Auch sie entsenden gemäß der Regel (ii) Kollateralen zu den Kleinhirnkernen (11, 12, 13 in Abb. 6, 9 in Abb. 7) sowie zu einigen Stern- (4 in Abb. 3) und Korbzellen in der Molekularschicht. Eine Kletterfaser endet jeweils an genau einer Purkinje-Zelle, indem sie sich aufsplittert und rankenartig den Verzweigungen des Dendritenbaumes anliegt.

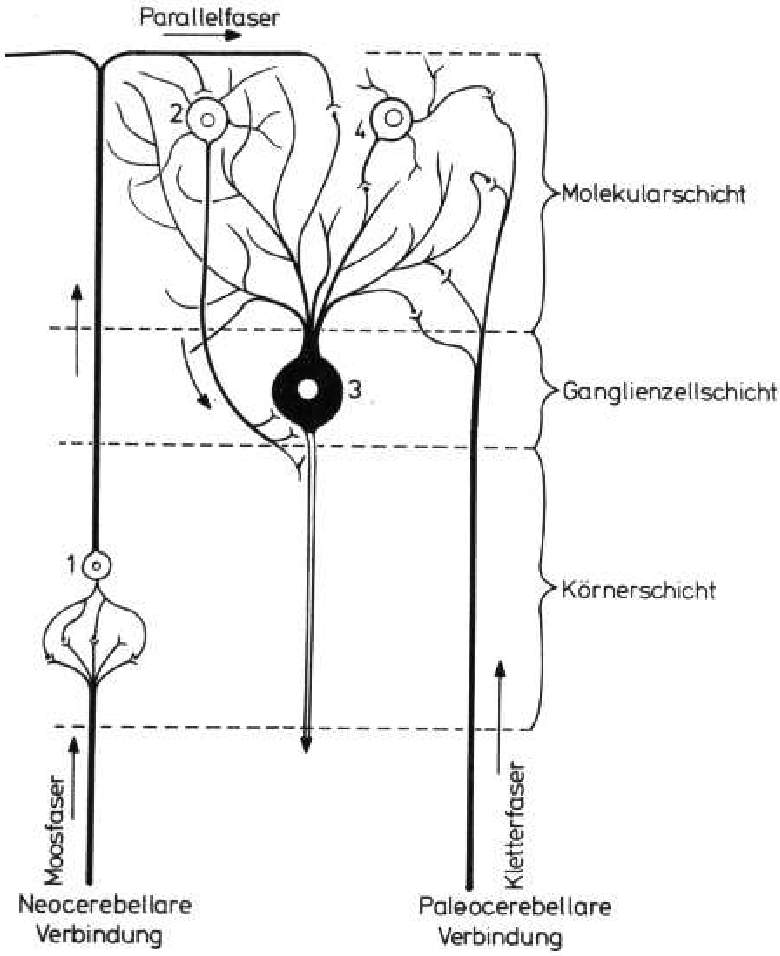
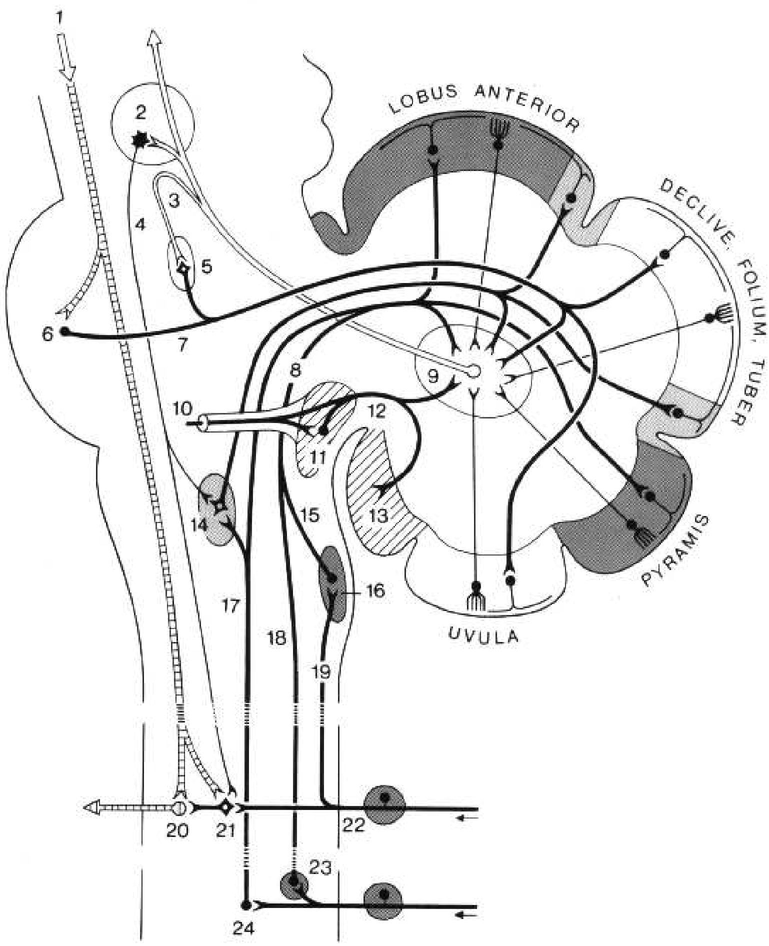
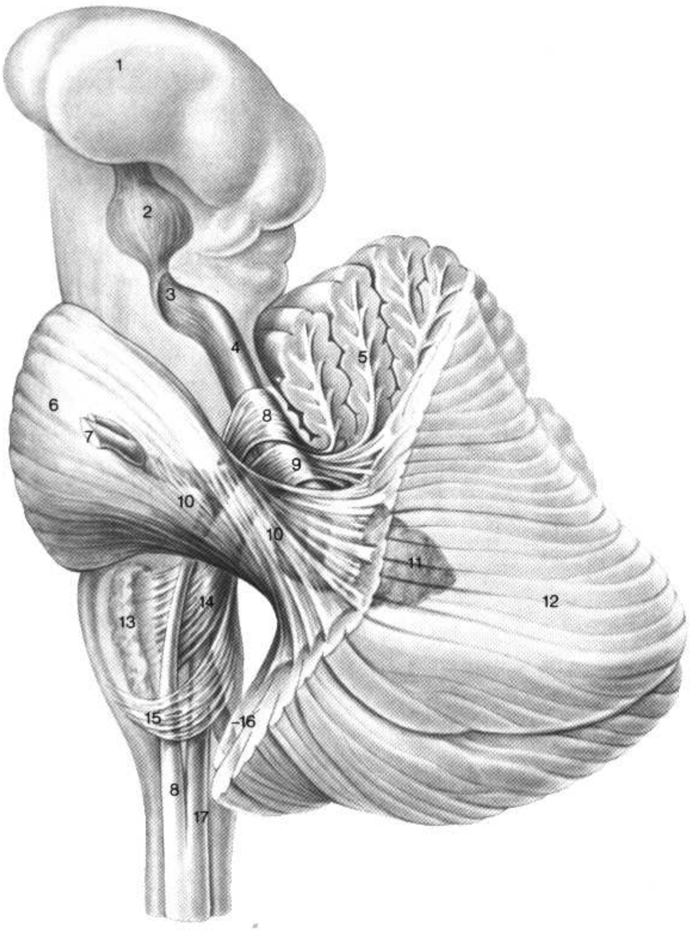


Abbildung 3: Die wichtigsten neuronalen Schaltmuster in der Kleinhirnrinde. Links eine indirekte Verbindung: die aufsteigende Nervenfaser (Moosfaser) endet vielfach aufgesplittert an den Dendriten einer Körnerzelle (1). Die Erregung wird zu den *Purkinje*-Zelldendriten und zu Dendriten einer Korbzelle (2) weitergeleitet. Deren Axon endet korbförmig am Perikaryon der *Purkinje*-Zelle (3). Rechts eine direkte Verschaltung zur *Purkinje*-Zelle durch eine Kletterfaser, die Axonkollateralen zu einer Sternzelle (4) schickt. Aus [11].



1 Tractus pyramidalis
2 Nucleus ruber
3 Pedunculus cerebellaris superior, ramus descendens
4 Tractus rubrospinalis
5 Formatio reticularis pontis
6 Nuclei pontis
7 Pedunculus cerebellaris medius
8 Pedunculus cerebellaris inferior
9 Nucleus centrales cerebelli
10 Nervus vestibularis (VIII)
11 Nuclei vestibulares
12 Fasciculus uncinatus cerebelli
13 Lobus flocculonodularis
14 Formatio reticularis myelencephali
15 Tractus cuneocerebellaris
16 Nucleus cuneatus lateralis
17 Tractus spinocerebellaris anterior
18 Tractus spinocerebellaris posterior
19 Funiculus posterior
20 Cellulae motoriae cornus anterioris
21 Substantia intermedia
22 Radix dorsalis nervi spinalis
23 Nucleus thoracicus
24 "Randzelle" (des Vorderhornes)

Abbildung 4: Afferente Verbindungen des Kleinhirns I: Verteilung der Moosfasersysteme Aus [22].



1 Thalamus
2 Nucleus ruber dexter
3 Decussatio pedunculorum
cerebellarium superiorum
4 Pedunculus cerebellaris superior
5 Vermis cerebelli
6 Pons
7 Nervus trigeminus
8 Tractus spinocerebellaris anterior
9 Pedunculus cerebellaris inferior
10 Pedunculus cerebellaris medius
11 Nucleus dentatus
12 Hemisphaerium cerebelli
13 Nucleus olivaris inferior dexter
14 Tractus olivocerebellaris
15 Fibrae arcuatae externae
16 Tractus cuneocerebellaris
17 Tractus spinocerebellaris posterior

Abbildung 5: Stiele des Kleinhirns in Lateralansicht. Aus [22].

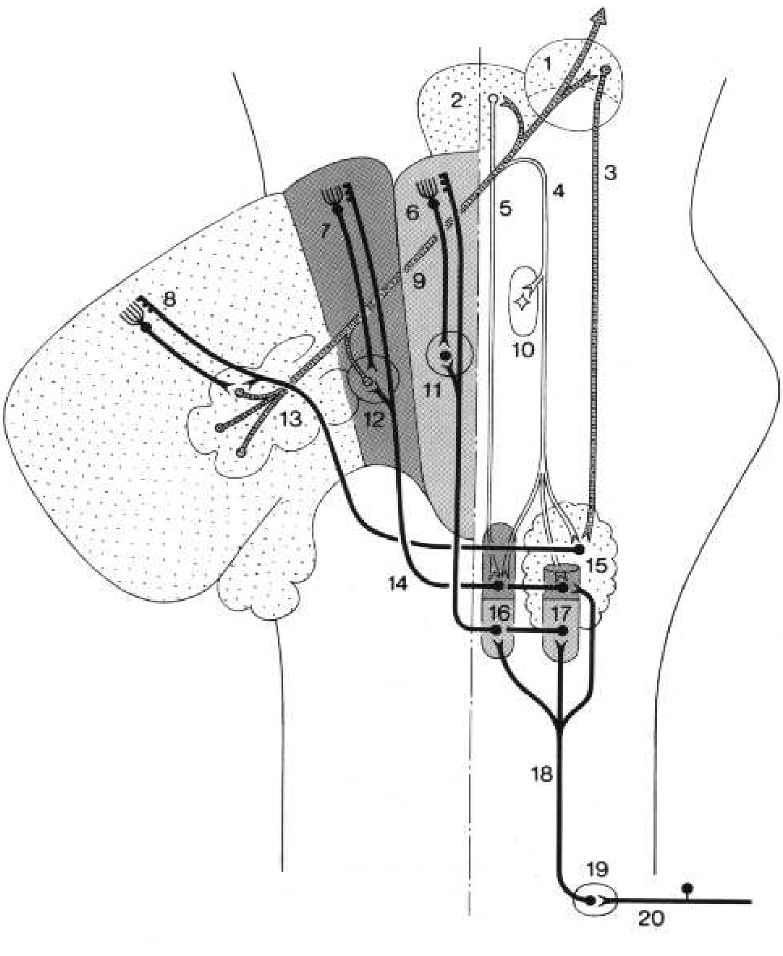


Abbildung 6: Afferente Verbindungen des Kleinhirns II: Kletterfasern aus der unteren Olive und dazugehörige Leitungsbögen. Aus [22].

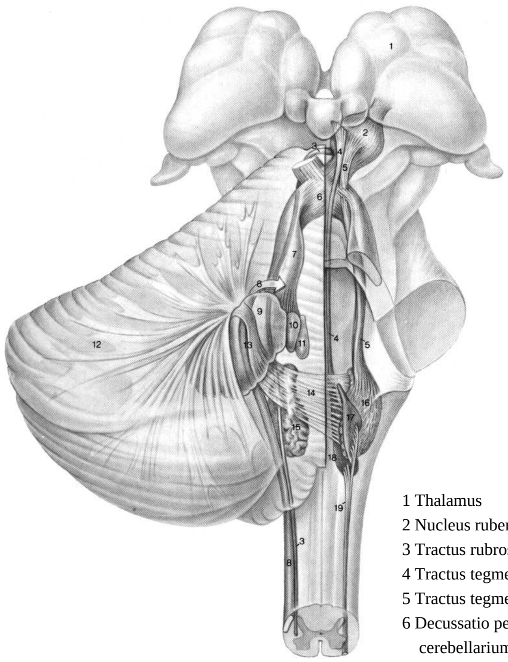


Abbildung 7: Olivocerebellarer Leitungsbogen. Die Lage der Bahnen und Kerne in Dorsalansicht. Das Kleinhirn wurde in der Mittellinie durchgeschnitten und die rechte Hälfte entfernt. Aus [22].

Innerhalb der Kleinhirnrinde ziehen die Axone der Körnerzellen (**Parallelfasern**) in die Molekularschicht, spalten sich dort T-förmig und verlaufen dann parallel zur Oberfläche entlang den Cortexwindungen durch die senkrecht dazu ausgerichteten Dendritenbäume der Purkinje-Zellen (3 in Abb. 3), mit denen sie excitatorische Synapsen bilden. Die Geometrie dieser mikroanatomischen Anordnung verdeutlicht Abb. 8. Man erkennt deutlich, daß eine Parallelfaser (5 in Abb. 8) zahlreiche flache Dendritenbäume von Purkinjezellen (4 in Abb. 8) in etwa senkrecht durchdringt. Zusätzlich bilden die Parallelfasern Synapsen mit inhibitorischen Interneuronen (Korbzellen und Sternzellen, 2 und 4 in Abb. 3 sowie 8 und 7 in Abb. 8). Ebenfalls inhibitorisch auf die Purkinjezellen wirken die Golgi-Zellen, die ihren Input aus Parallelfasern bzw. direkt aus Axonkollateralen der Moosfasern beziehen (9 in Abb. 8).

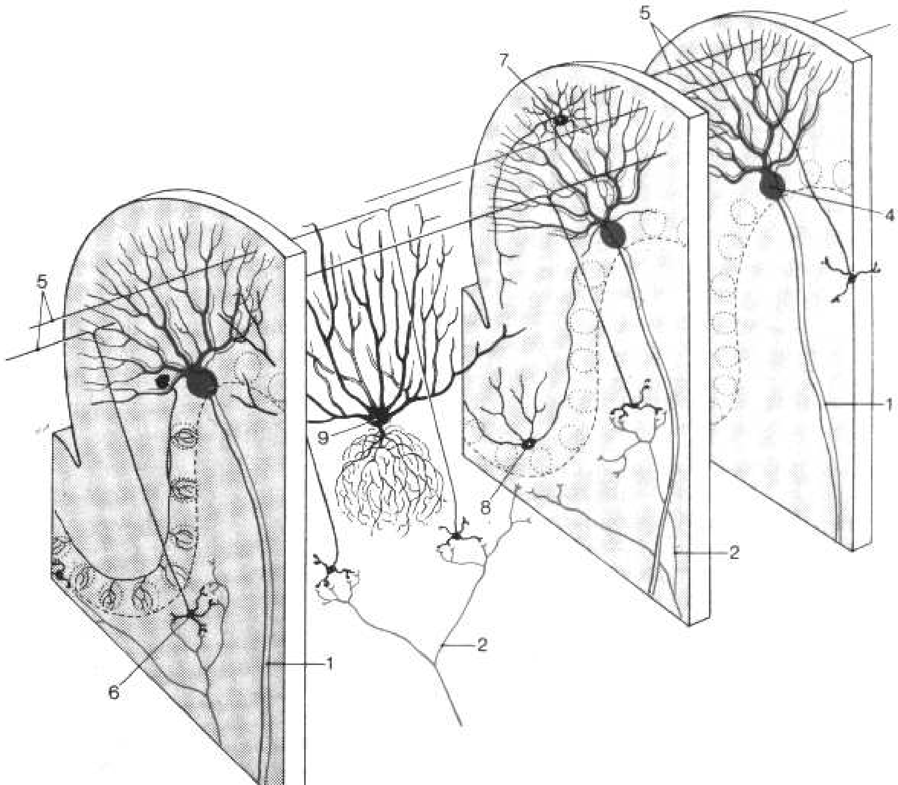


Abbildung 8: Schema zur Geometrie der mikroanatomischen Anordnung von Neuronenverbindungen der Kleinhirnrinde. Aus [17].

Der weit verzweigte Dendritenbaum einer Purkinje-Zelle (Abb. 9) empfängt also excitatorischen Input aus zwei Quellen: einerseits über die Parallelfasern von ca. 80000 Körnerzellen, andererseits über genau eine Kletterfaser von Kerngebieten der unteren Olive. Die Axone der Purkinje-Zellen enden gemäß obiger Regeln (iii) und (iv) in den Kleinhirnkernen. Interessanterweise üben sie dort einen inhibitorischen Einfluß aus. Von den Kleinhirnkernen ziehen übrigens auch efferente Fasern (Regel (v)) wieder zurück in die Quellgebiete sowohl der Moosfaseraktivität (pontine Formatio reticularis, 5 in Abb. 4) als auch der Kletterfaseraktivität (untere Olive, 15, 16, 17 in Abb. 6).

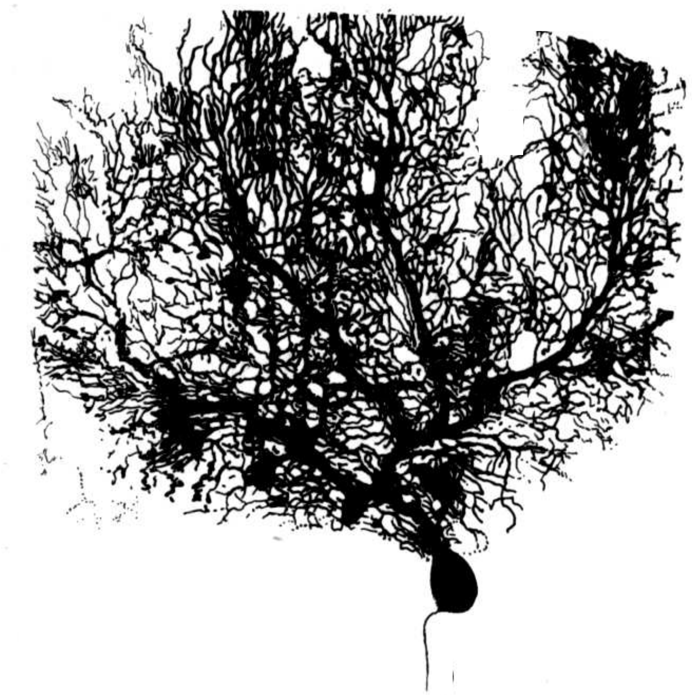


Abbildung 9: *Purkinje-Zelle*, Silberimprägnation. Aus [17].

Die Plastizität der Synapsen zwischen Parallelfasern und Purkinje-Zellen unter dem Einfluß der Kletterfaseraktivität ist nun die Kernnannahme zahlreicher theoretischer Modellbildungen zur Kleinhirnfunktion ([20], [5], [24]). Bei gleichzeitiger Erregung beider Fasersysteme ist eine verstärkte Erregungsübertragung von den Parallelfasern auf die Purkinje-Zellen experimentell nachgewiesen worden [16]. (Derartige Annahmen synaptischer Plastizität bilden übrigens auch die Grundlage zahlreicher Modellansätze zur Erklärung klinisch relevanter Phänomene in anderen funktionellen Systemen des ZNS, vgl. z. B. [33].)

Im folgenden wird eine von Marr und Albus begründete und durch Poggio [24] weiterentwickelte Interpretation einiger hier vorgestellter neuroanatomischer Strukturen skizziert.

Ausgangspunkt ist die Abstraktion im Hinblick auf eine Dreischichtenstruktur von Nervenzellen. Die **„Eingabeschicht“** wird hierbei mit den Quellgebieten der Moosfaseraktivität (Abb. 4) identifiziert. Dieser Input wird auf eine **„Zwischenschicht“** abgebildet, die den Körnerzellen (6 in Abb. 8, 1 in Abb. 3) entsprechen soll. Deren Axone bilden über die Parallelfasern (5 in Abb. 8) plastische Synapsen mit den Purkinje-Zellen (4 in Abb. 8, 3 in Abb. 3), die als Zellen einer **„Ausgabeschicht“** interpretiert werden. Die Stärke dieser plastischen Synapsen wird dabei durch die gleichzeitige Aktivierung des Kletterfasersystems (Abb. 6) beeinflußt, was in der Abstraktion als „überwachtes Lernen“ der Ausgabegewichte zwischen Körner- und Purkinje-Zellen modelliert wird. Insgesamt soll das biologische Netzwerk wohl dazu dienen, den mit motorischen Aktionen verknüpften sensiblen Input aus den Quellgebieten der Moosfaseraktivität mit „abstrakten Bewegungsentwürfen“ aus den Quellgebieten der Kletterfaseraktivität in Beziehung zu setzen. In der modellhaften Abstraktion wird daraus die Fähigkeit zur hochdimensionalen Funktionenapproximation, mit der sich u. a. auch komplexe Klassifkationsprobleme lösen lassen.

Die mathematische Formulierung dieses Modells wird weiter unten durchgeführt. Vorher sei jedoch noch auf einige neuroanatomische Aspekte hingewiesen, die in der abstrakten Modellierung bislang keine oder eine nur untergeordnete Berücksichtigung finden und die nach Meinung des Verfassers im Hinblick auf eine Modellerweiterung anhand der folgenden Fragen zumindest kritisch diskutiert werden sollten:

(i) Welche Rolle spielen die oben vorgestellten inhibitorischen Interneuronen, also die Golgi-, Stern- und Korbzellen? Für die Golgi-Zellen wurde eine regulatorische Funktion für die Größe eines sog. „Codons“ postuliert [20]. Dieses definiert die Anzahl der Moosfasern, die jeweils eine Körnerzelle beeinflussen. Könnte die Funktion der Golgi-Zellen mit der Einstellung der „ρ-Breiten“ (s. u.) in Beziehung gesetzt werden? Ist die Rolle der Stern- und Korbzellen im Sinne der „lokalen, lateralen Inhibition“ erschöpfend charakterisiert oder ist dies nur ein Vorurteil? Welche Rolle spielen in diesem Zusmammenhang insbesondere die Synapsen von Korb- und Sternzellen zum Kletterfasersystem?

(ii) Welche Rolle spielen die excitatorischen Kollateralen der Moosfasern und Kletterfasern (vgl. obige Regel (ii)) zu den Kleinhirnkernen, die ja den eigentlichen Output des Kleinhirns generieren (vgl. obige Regel (v)). Wie muß man den inhibitorischen Einfluß der Purkinje-Zellen auf die Kleinhirnkerne in eine sinnvolle Erweiterung des Modells einbeziehen?

(iii) Kommt der geometrischen Anordnung der beteiligten neuroanatomischen Strukturen (z. B. Parallelfasern und Dendritenbäume der Purkinje-Zellen) eine tiefere funktionelle Bedeutung zu?

(iv) Spielen Zeitverzögerungen im System eine wesentliche Rolle? Warum sind Parallelfasern marklos, Axone von Purkinjezellen hingegen markhaltig?

(v) Wie ist die oben beschriebene Rückkopplung der Efferenzen aus den Kleinhirnkernen auf die Quellgebiete sowohl der Moosfaser- als auch der Kletterfaseraktivität sinnvoll zu modellieren?

Diese Fragen machen deutlich, daß die im folgenden skizzierte Architektur des GRBF-Netzwerks wohl bestenfalls auch nur eine „Karikatur“ der Kleinhirnfunktion liefern kann — ebenso wie die eingangs erwähnte „Karikatur“ eines biologischen Neurons durch das formale Neuron des Neuroinformatikers. Dennoch erweist sich dieses aus der Kleinhirnanatomie motivierte Modell als leistungsfähiges Instrument für eine Reihe von Mustererkennungs- und Datenanalyseaufgaben. Dieser Erfolg mag nachträglich ein entsprechendes Vorgehen rechtfertigen.

3 Mathematische Modellierung

3.1 Aufbau und Funktionsweise

Den allgemeinen Aufbau eines GRBF-Netzwerkes verdeutlicht Abbildung 10. Es besteht aus drei Schichten: einer Eingabeschicht, einer Zwischenschicht und einer Ausgabeschicht. Legt man n Eingabezellen mit den Aktivitäten x_i, $i \in \{1, \ldots, n\}$ zugrunde, so repräsentiert das Aktivitätsmuster auf der *Eingabeschicht* einen Datenpunkt $\mathbf{x}$ im sogenannten *Merkmalsraum* $\mathbb{R}^n$. Durch gerichtete, mit den Gewichten w_{ji} gewichtete Verbindungen wird diese Aktivität auf die N Zellen einer *Zwischenschicht* propagiert. Die Synapsenstärken $\mathbf{w}_j \in \mathbb{R}^n$, $j \in \{1, \ldots, N\}$ sind hierbei so gewählt, daß die Aktivität a_j eines Zwischenschichtneurons j in geeigneter Weise vom Abstand $d = \|\mathbf{x} - \mathbf{w}_j\|$ des Datenpunkts $\mathbf{x}$ vom *virtuellen Ort* $\mathbf{w}_j$ des Zwischenschichtneurons j abhängt. d definiert dabei eine beliebige Metrik im Merkmalsraum, z. B. die euklidische Metrik. Der Begriff „virtueller Ort" basiert hierbei auf der Vorstellung, daß die Aktivität a_j des Zwischenschichtneurons für $\mathbf{x}_{\max} \stackrel{!}{=} \mathbf{w}_j$ ihren Maximalwert annehmen soll, was als „Spezialisierung" des Neurons j auf den Ort $\mathbf{x}_{\max}$ interpretiert werden kann.

Offensichtlich ist es sinnvoll, a_j mit wachsendem d monoton fallend zu wählen, d. h. mit zunehmendem Abstand zwischen $\mathbf{x}$ und dem virtuellen Ort $\mathbf{w}_j$ möge die Aktivität a_j des Zwischenschichtneurons abnehmen. Im einfachsten Fall wählt man hierbei das entsprechende Abklingverhalten unabhängig von der Raumrichtung der Abweichung $(\mathbf{x} - \mathbf{w}_j)$. Hieraus definiert sich die Rotationssymmetrie bzw. Radialität von $a_j(\mathbf{x})$ um $\mathbf{w}_j$: Man spricht von einer *radialen Basisfunktion* (RBF). Aufgrund günstiger Eigenschaften werden hierzu von F. Girosi und T. Poggio gaußförmige Aktivitätsfunktionen vorgeschlagen [13]:

$$\tilde{a}_j(\mathbf{x}) = g(\|\mathbf{x} - \mathbf{w}_j\|, \rho_j) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{w}_j\|^2}{2\rho_j^2}\right) \tag{1}$$

Moody und Darken [21] schlagen eine globale Normierung der Zwischenschichtaktivität mit

$$A(\mathbf{x}) = \sum_{i=1}^{N} \tilde{a}_i(\mathbf{x}) = \sum_{i=1}^{N} g(\|\mathbf{x} - \mathbf{w}_i\|, \rho_i) \tag{2}$$

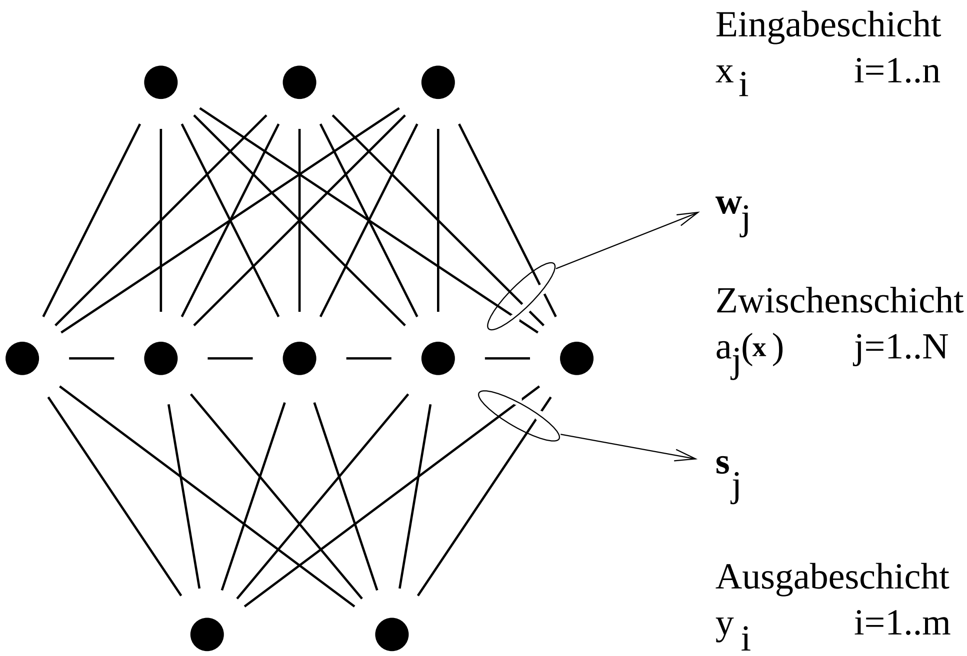


Abbildung 10: Struktur eines GRBF-Netzwerkes

vor, so daß sich für die Zwischenschichtaktivität

$$
\begin{aligned}
a_j(\mathbf{x}) &= \frac{\tilde{a}_j(\mathbf{x})}{A(\mathbf{x})} \\
&= \frac{g(\|\mathbf{x}-\mathbf{w}_j\|, \rho_j)}{\sum_{i=1}^{N} g(\|\mathbf{x}-\mathbf{w}_i\|, \rho_i)} \\
&= \frac{\exp\left(-\frac{\|\mathbf{x}-\mathbf{w}_j\|^2}{2\rho_j^2}\right)}{\sum_{i=1}^{N} \exp\left(-\frac{\|\mathbf{x}-\mathbf{w}_i\|^2}{2\rho_i^2}\right)}
\end{aligned} \tag{3}
$$

ergibt. Hierdurch wird eine Kompetition der Zwischenschichtneuronen eingeführt, die sich im Hinblick auf eine spätere Wahrscheinlichkeitsinterpretation von Klassifikationsergebnissen als nützlich erweist. Durch Gleichung 3 geht offensichtlich die Radialität der Aktivitätsfunktionen $\tilde{a}_j(\mathbf{x})$ (Gleichung 1) verloren. Im folgenden wird deshalb von *generalisierten* radialen Basisfunktionen (GRBF) gesprochen. In [13] wird ein derartiges System „Hyper-BF-Netzwerk“ genannt.

In einem letzten Schritt erfolgt nun eine lineare Signalpropagation der Zwischenschichtaktivität auf die m Zellen einer *Ausgabeschicht* durch gewichtete Summation:

$$\mathbf{y}(\mathbf{x}) = \sum_{j=1}^{N} \mathbf{s}_j a_j(\mathbf{x}), \tag{4}$$

also komponentenweise

$$y_i(\mathbf{x}) = \sum_{j=1}^{N} s_{ij} a_j(\mathbf{x}), \quad i \in \{1, \ldots, m\}. \tag{5}$$

Dieser Abbildung auf die Ausgabeschicht entspricht einer in der Neuroinformatik als *Perzeptron* bekannte Struktur (siehe F. Rosenblatt [27]).

3.2 Lernverfahren

Das im Netz gespeicherte „Wissen“ wird durch die freien Parameter $P = \{(\mathbf{w}_j, \mathbf{s}_j, \rho_j)\}$ des Netzwerkes, d. h. durch die Gewichte $\mathbf{w}_j$ und $\mathbf{s}_j$ sowie durch die Gaußbreiten ρ_j der Zwischenschicht repräsentiert. Es dient der Approximation einer Funktion

$$\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m, \quad \mathbf{x} \mapsto \mathbf{y} = \mathbf{F}(\mathbf{x}), \tag{6}$$

die anhand eines Beispieldatensatzes $T = \{(\mathbf{x}^\nu, \mathbf{y}^\nu) \mid \nu \in \{1, \ldots, p\}\}$ von Trainingsdaten gelernt werden soll. Hierbei werden zwei Forderungen gestellt: Einerseits soll das Netz diese vorgegebenen Trainingsdaten möglichst exakt reproduzieren, andererseits jedoch auch außerhalb dieser Punktmenge sinnvoll generalisieren.

Eine naheliegende Lernstrategie besteht in der Minimierung des Fehlers

$$E(P) = \sum_{\nu=1}^{p} \|\mathbf{y}^\nu - \mathbf{F}(\mathbf{x}^\nu)\|^2. \tag{7}$$

Beim Versuch der gleichzeitigen Optimierung aller freien Parameter mittels Gradientenabstieg auf der Kostenfunktion (7) treten zwei Probleme auf: Zunächst kann das Verfahren in lokalen Minima gefangen werden und somit suboptimale Lösungen liefern. Dieser Schwierigkeit kann durch Stochastifizierung (sog. *simulated annealing*) begegnet werden, indem man durch einen Rauschterm den Ausbruch aus lokalen Minima ermöglicht. Ein zweites Problem liegt in der langsamen Konvergenz einer derartigen globalen Optimierung. In dieser Hinsicht bietet dieses von Poggio in [24] vorgeschlagene Verfahren keinen

Vorteil gegenüber der klassischen Optimierungsstrategie des *error backpropagation* für das Mehrschichtperzeptron (siehe z. B. [29]). Eine solche globale, d. h. gleichzeitige Optimierung aller Parameter widerspricht auch der Vorstellung von der biologischen Informationsverarbeitung durch lokale Selbstorganisation. Dies motiviert den Vorschlag von Moody und Darken (siehe [21]), den Parameterraum $P = \{(\mathbf{w}_j, \mathbf{s}_j, \rho_j)\}$ in drei kleinere Teilräume $\{\mathbf{w}_j\}, \{\mathbf{s}_j\}, \{\rho_j\}$ aufzugliedern und auf diesen getrennt zu optimieren. Die Dimensionsreduktion führt zu einer erheblich höheren Konvergenzgeschwindigkeit für die Optimierung auf den einzelnen Teilräumen. Allerdings bedingt dieses Vorgehen die Gefahr, nur suboptimale Näherungslösungen für die Minimierung von (7) zu erhalten, da nun nicht mehr der gesamte Parameterraum P systematisch abgesucht wird.

Dieser Ansatz zur getrennten Optimierung auf den drei genannten Teilräumen von P bildet die Grundlage der GRBF-Architektur. Er erfolgt in drei Schritten:

1. Zunächst werden die virtuellen Orte $\mathbf{w}_j$ der Zwischenschichtneuronen durch ein Verfahren zur sogenannten *Vektorquantisierung* (VQ) ermittelt. Dabei wird ein unüberwachtes Lernverfahren (*unsupervised clustering*) verwendet, d. h. die Zielinformation $\mathbf{y}^\nu$ des Trainingsdatensatzes $T = \{(\mathbf{x}^\nu, \mathbf{y}^\nu) | \nu \in \{1, \ldots, p\}\}$ wird für den Lernvorgang nicht genutzt.

2. Die Breite der „rezeptiven Felder“ ρ_j der Zwischenschichtneuronen werden bei bekannten $\mathbf{w}_j$ heuristisch festgelegt oder im Hinblick auf die Güte des Approximationsergebnisses optimiert.

3. Letztlich werden die Ausgabegewichte $\mathbf{s}_j$ bestimmt. Hierzu kann gewählt werden zwischen einer globalen Gradientenabstiegsmethode und einem auf Hebbschem Korrelationslernen beruhenden lokalen Lernverfahren.

Die einzelnen Lernschritte werden im folgenden erläutert.

3.2.1 Optimierung der $\mathbf{w}_j$: Vektorquantisierung

Soll ein Datensatz $X = \{\mathbf{x}\}$ mit $\mathbf{x} \in \mathbb{R}^n$ durch eine Menge C von sogenannten *Codebuchvektoren* $\mathbf{w}_j$ mit $C = \{\mathbf{w}_j \in \mathbb{R}^n \mid j \in \{1, \ldots, N\}\}$ charakterisiert werden, so spricht man vom Problem der Vektorquantisierung. Hierbei soll das *Codebuch* C die Wahrscheinlichkeitsdichte

$$f : \mathbb{R}^n \to [0,1], \quad \mathbf{x} \mapsto f(\mathbf{x})$$

des Datensatzes X geeignet repräsentieren, wobei die Anzahl N der Codebuchvektoren wesentlich kleiner als die Anzahl $\sharp C$ der Datenpunkte sein soll. Zu den vielfältigen Anwendungsfeldern der VQ, wie Analyse und Kompression großer Datenmengen siehe z. B. [14].

Man unterscheidet bei der VQ zwischen einem *hard clustering*, bei dem jeder Datenpunkt $\mathbf{x}$ genau einem Codebuchvektor $\mathbf{w}_j$ zugeordnet wird, und einem *fuzzy clustering*, bei dem ein Datenpunkt $\mathbf{x}$ in geeigneter Weise auf mehrere Codebuchvektoren $\mathbf{w}_j$ abgebildet werden kann.

Der Prozeß der Vektorquantisierung läßt sich allgemein als iterativer Lernvorgang charakterisieren: Zunächst wird die Anzahl N der Codebuchvektoren $\mathbf{w}_j$ festgelegt, und diese werden initialisiert. Im weiteren wird jeweils ein Datenpunkt $\mathbf{x} \in X$ zufällig ausgewählt und alle Codebuchvektoren nach der *allgemeinen, sequentiellen VQ-Lernregel*

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \epsilon(t)\, \psi(t, \mathbf{x}, C)\, (\mathbf{x}(t) - \mathbf{w}_j(t)) \tag{8}$$

aktualisiert. t bezeichnet den Aktualisierungsschritt, ϵ einen frei wählbaren Lernparameter und ψ die sogenannte Kooperationsfunktion. Der Lernparameter ϵ wird mit fortschreitendem Aktualisierungsschritt monoton fallend gewählt. Aufgrund von Analogien zu Systemen der statistischen Physik wird dies als „Abkühlung“ bezeichnet. Häufig wird eine exponentielle Abkühlungsstrategie verwendet:

$$\epsilon(t) = \epsilon(0) \left(\frac{\epsilon(t_{\max})}{\epsilon(0)} \right)^{\frac{t}{t_{\max}}}, \quad t \in [0, t_{\max}]. \tag{9}$$

Neben der jeweils gewählten Heuristik zur Festlegung der Zeitabhängigkeit von ϵ und ψ unterscheiden sich die einzelnen VQ-Methoden nur durch die Definition der Kooperationsfunktion ψ.

Eine einfache Methode für das *hard clustering* bietet beispielsweise der LBG-Algorithmus von Y. Linde, A. Buzo und R. Gray [19]. Hierbei wählt ψ in jedem Lernschritt genau einen zu aktualisierenden Codebuchvektor $\mathbf{w}_j$ gemäß

$$\psi(t, \mathbf{x}^\nu, C) := \delta_{i(\mathbf{x}),j} \tag{10}$$

aus, wobei $i(\mathbf{x})$ aus dem minimalen Abstand

$$\|\mathbf{x} - \mathbf{w}_i\| = \min_j \|\mathbf{x} - \mathbf{w}_j\|$$

definiert wird. Da jeweils nur genau ein Codebuchvektor an jedem Lernschritt teilnimmt, spricht man auch von einer *winner-takes-all*-Lernregel.

Wird ψ hingegen so gewählt, daß in jedem Lernschritt mehrere Codebuchvektoren an der Aktualisierung beteiligt sind, so definiert Gleichung 8 eine *winner-takes-most*-Lernregel. Je nach Definition von ψ resultieren hieraus unterschiedliche Methoden für ein sogenanntes *fuzzy clustering.*

In dem hier angewendeten Verfahren erfolgt die Vektorquantisierung durch die sogenannte *minimal free energy VQ* [25, 26, 9]. Die Kooperationsfunktion ist bereits durch Gleichung 3 gegeben, wobei die ρ_j für alle Neuronen gleich gewählt werden:

$$\psi(\mathbf{x}) = a_j(\mathbf{x}) = \frac{\exp(-\frac{\|\mathbf{x}-\mathbf{w}_j\|^2}{2\rho^2})}{\sum_{i=1}^{N} \exp(-\frac{\|\mathbf{x}-\mathbf{w}_i\|^2}{2\rho^2})}. \tag{11}$$

Aufgrund ihrer Normierung können die Aktivitäten $a_j(\mathbf{x})$ als Zuordnungswahrscheinlichkeiten der Codebuchvektoren $\mathbf{w}_j$ zum jeweiligen Datenpunkt $\mathbf{x}$ interpretiert werden. Aus Gleichung 8 wird somit

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \epsilon(t)\, a_j(\mathbf{x}(t))\, (\mathbf{x}(t) - \mathbf{w}_j(t)) + R(\rho), \tag{12}$$

wobei ein Rauschterm $R(\rho)$ hinzugefügt wurde, der weiter unten erläutert wird.

Definiert man den Erwartungswert $\langle f \rangle_X$ einer Zufallsgröße f

$$\langle f \rangle_X := \int_X f(\mathbf{x})\, p(\mathbf{x}) d\mathbf{x} \tag{13}$$

mit der Wahrscheinlichkeitsdichte $p(\mathbf{x})$ und die klassenlokalen Erwartungswerte $\langle f \rangle_j$

$$\langle f \rangle_j := \int_X f(\mathbf{x})\, p(\mathbf{x}|j) d\mathbf{x}, \tag{14}$$

so berechnen sich diese mit dem *Satz von Bayes*

$$p(\mathbf{x}|j) = \frac{p(\mathbf{x})\, p(j|\mathbf{x})}{p(j)} \tag{15}$$

zu

$$\langle f \rangle_j = \int_X f(\mathbf{x}) \frac{p(\mathbf{x})\, p(j|\mathbf{x})}{p(j)} d\mathbf{x} = \frac{\langle a_j(\mathbf{x})\, f \rangle_X}{\langle a_j(\mathbf{x}) \rangle_X}, \tag{16}$$

wenn man aufgrund der Normierung der Zwischenschichtaktivitäten (Gleichung 3)

$$\sum_{j=1}^{N} a_j(\mathbf{x}) = 1 \tag{17}$$

diese als a-posteriori-Wahrscheinlichkeiten $p(j|\mathbf{x})$ für die Zuordnung des Datenpunkts $\mathbf{x}$ zum Zwischenschichtneuron j interpretiert und somit

$$p(j) = \int_X p(j \cap \mathbf{x}) d\mathbf{x} = \int_X p(\mathbf{x})\, p(j|\mathbf{x}) d\mathbf{x} = \langle a_j(\mathbf{x}) \rangle_X \tag{18}$$

als durchschnittliche Aktivierung (sog. *load*) des Neurons j erhält.
Aus der Stationaritätsbedingung

$$\langle a_j(\mathbf{x})\, (\mathbf{x} - \mathbf{w}_j) \rangle_X \stackrel{!}{=} 0 \tag{19}$$

der Lernregel (12) folgt unmittelbar

$$\mathbf{w}_j = \frac{\langle a_j(\mathbf{x})\, \mathbf{x} \rangle_X}{\langle a_j(\mathbf{x}) \rangle_X}. \tag{20}$$

Die Iteration des Verfahrens (12) bis zur Stationarität führt also zu einer unscharfen Tesselierung des Merkmalsraums

$$\int_X p(\mathbf{x}) d\mathbf{x} = \sum_j \int_X p(\mathbf{x})\, p(j|\mathbf{x}) d\mathbf{x}, \tag{21}$$

wobei die Codebuchvektoren $\mathbf{w}_j$ gemäß (16) die klassenlokalen Mittel $\langle \mathbf{x} \rangle_j$ der Datenverteilung repräsentieren.

Der Parameter ρ entspricht der „Auflösung“, mit der das Codebuch die Datenverteilung X sieht. Während der Iteration nach Gleichung 12 wird ρ langsam exponentiell verringert. Dadurch werden in X immer feinere Strukturen sichtbar, bis für $\rho \to 0$ die volle „Sehschärfe“ erreicht wird. Die Umstrukturierung des Codebuchs während der Verringerung von ρ wird von einer Folge von *Phasenübergängen* begleitet. Die Phasenübergänge trennen jeweils verschiedene Ordnungszustände des Codebuchs, die durch Häufungen von Neuronen an verschiedenen Orten im Datenraum gekennzeichnet sind. Diese „Ballungsgebiete“ werden auch als natürliche Cluster bezeichnet. Die Vektorquantisierung mit Hilfe der minimal free energy VQ wird in der Literatur auch als ein *Clustering*-Verfahren bezeichnet. Abbildung 11 zeigt den Vorgang der Vektorquantisierung am Beispiel einer Datenverteilung bestehend aus mehreren Gaußhaufen.

Zwischen den Phasenübergängen sind Teile des Codebuchs entartet, d. h. einige Codebuchvektoren haben numerisch den exakt gleichen Wert. Dies ist aber nicht erwünscht, weil bei exakter Entartung die Codebuchvektoren in jedem Lernschritt exakt gleich verändert werden und so ein Phasenübergang verhindert wird. Um dies zu umgehen, dient der in Gleichung 12 eingeführte Rauschterm $R(\rho)$. Dieser kann z. B. als eine normalverteilte Zufallsgröße, deren Streuung proportional zu ρ ist, gewählt werden. Die Proportionalitätskonstante liegt üblicherweise im Bereich $[10^{-3}, 10^{-2}]$.

Die Beziehung 20 liefert unmittelbar eine *batch-Version* der minimal free energy VQ: Die rechte Seite von Gleichung 20 hängt über die Aktivitäten $a_j(\mathbf{x})$ von den Codebuchpositionen $\mathbf{w}_j$ ab. Die Gewichtsberechnung wird so lange iterativ fortgeführt, bis für die $\mathbf{w}_j$ Konvergenz erreicht wird. Dieses Verfahren eignet sich gut zur Implementierung auf einem Parallelrechner (siehe [3]).

3.2.2 Optimierung der ρ_j

Zu verschiedenen Heuristiken für die Bestimmung der Breiten ρ_j der rezeptiven Felder bei bekannten Codebuchvektoren $\mathbf{w}_j$ und deren Optimierung sei auf die Arbeit [2] verwiesen.

3.2.3 Optimierung der $\mathbf{s}_j$: Überwachtes Lernen

Die Ausgabegewichte $\mathbf{s}_j$ (siehe Abbildung 10) werden zur Matrix $\mathbf{S} := (s_{ij}) \in \mathbb{R}^{(m,N)}$ zusammengefaßt. Ihre Bestimmung kann auf zwei verschiedene Arten erfolgen:

- Einerseits kann mittels eines *globalen Lernverfahrens* bei gegebenem Trainingsdatensatz $T = \{(\mathbf{x}^\nu, \mathbf{y}^\nu) \mid \nu \in \{1, \dots, p\}\}$ der Fehler

$$E = \frac{1}{2p} \sum_{\nu=1}^{p} \|\mathbf{y}^\nu - \mathbf{S}\mathbf{a}(\mathbf{x}^\nu)\|^2 \tag{22}$$

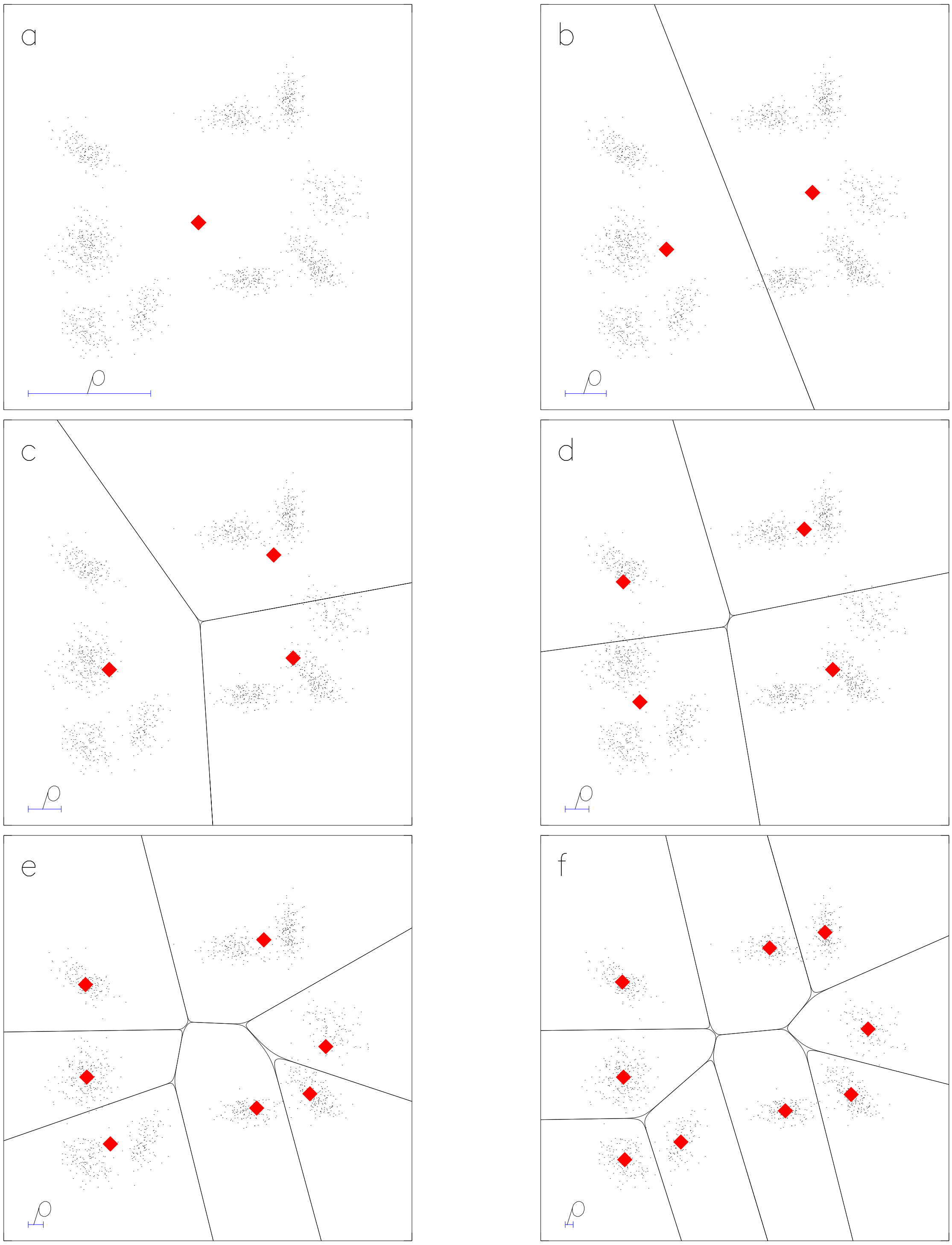


Abbildung 11: Ein Beispiel für eine *minimal free energy VQ*. Mit abnehmendem ρ spalten sich die zunächst entarteten Prototypen immer weiter auf und unterteilen so den Datensatz in immer feinere Klassen.

durch ein Gradientenabstiegsverfahren minimiert werden. Die Gewichtsänderung Δs_{ij} in einem Lernschritt mit der Lernrate ϵ definiert sich also aus

$$\Delta s_{ij} = -\epsilon \nabla_{s_{ij}} E \tag{23}$$

zu

$$\Delta s_{ij} = \frac{\epsilon}{p} \sum_{\nu=1}^{p} \left[y_i^\nu - \sum_{k=1}^{N} s_{ik} a_k(\mathbf{x}^\nu) \right] a_j(\mathbf{x}^\nu). \tag{24}$$

Unter Konvergenzannahme, d. h. $\Delta s_{ij} = 0$, folgt hieraus

$$\sum_{\nu=1}^{p} y_i^\nu a_j(\mathbf{x}^\nu) = \sum_{\nu=1}^{p} \sum_{k=1}^{N} s_{ik} a_k(\mathbf{x}^\nu) a_j(\mathbf{x}^\nu). \tag{25}$$

Das Einbringen der Klasseninformation geschieht wie folgt: Gehört $\mathbf{y}^\nu$ zur Klasse $\lambda \in \{1, \ldots, m\}$, so gilt:

$$y_i^\nu = \delta_{i,\lambda(\mathbf{x}^\nu)}, \quad i \in \{1, \ldots, m\}, \tag{26}$$

d. h. die Komponente y_λ^ν von $\mathbf{y}^\nu$ wird Eins gesetzt, alle anderen Komponenten sind Null ($\delta_{i,\lambda}$ bezeichnet das Kronecker Delta). Bildet man jeweils die Summe über die Aktivitäten aller Datenpunkte einer Klasse λ, so entsteht aus Gleichung 25 ein System aus m linearen Gleichungen für die Gewichte $s_{\lambda k}$:

$$\langle a_j(\mathbf{x}) \rangle_{X_\lambda} = \sum_{k=1}^{N} s_{\lambda k} \langle a_k(\mathbf{x}) a_j(\mathbf{x}) \rangle_X, \quad \lambda \in \{1, \ldots, m\}. \tag{27}$$

- Eine Alternative hierzu bietet ein *lokales Lernverfahren* auf der Basis einer Hebbschen Lernregel:

$$\Delta s_{ij} = \epsilon \left(y_i - s_{ij} \right) a_j(\mathbf{x}) \tag{28}$$

Sie stützt sich gemäß der Hypothese von Donald Hebb [15] auf einen Wachstumsterm $\Delta s_{ij} \sim y_i a_j(\mathbf{x})$, der korrelierte prä- und postsynaptische Aktivität belohnt. Der Zerfallsterm $\Delta s_{ij} \sim s_{ij} a_j(\mathbf{x})$ verhindert auf einfache Weise unbeschränktes Wachstum. Fordert man wiederum die Konvergenz des Verfahrens, also $\Delta s_{ij} = 0$, so folgt bei Summation über alle $\mathbf{x} \in X$:

$$\sum_{\nu=1}^{p} y_i^\nu a_j(\mathbf{x}^\nu) = \sum_{\nu=1}^{p} s_{ij} a_j(\mathbf{x}^\nu). \tag{29}$$

Nimmt man wiederum die Klasseninformation durch $y_i^\nu = \delta_{i,\lambda(\mathbf{x}^\nu)}$ hinzu, so errechnen sich die endgültigen Gewichte zu

$$s_{\lambda j} = \frac{\langle a_j(\mathbf{x}) \rangle_{X_\lambda}}{\langle a_j(\mathbf{x}) \rangle_X}. \tag{30}$$

Neben der Lokalität des Lernverfahrens bieten die nach Gleichung 30 errechneten Gewichte den Vorteil der Normierung:

$$\sum_{\lambda=1}^{m} s_{\lambda j} = 1. \tag{31}$$

Dies führt zusammen mit der Normierung der Zwischenschichtaktivitäten nach Gleichung 17 ebenfalls zu einer Normierung der Aktivitäten $y_\lambda(\mathbf{x})$ auf der Ausgabeschicht

$$\sum_{\lambda=1}^{m} y_\lambda(\mathbf{x}) = \sum_{\lambda=1}^{m} \sum_{j=1}^{N} s_{\lambda j} a_j(\mathbf{x}) = 1 \tag{32}$$

und ermöglicht somit (auch wegen $a_j(\mathbf{x}) \geq 0$ und $s_{\lambda j} \geq 0$) eine probabilistische Interpretation der Klassifikationseigenschaften des GRBF-Netzwerks. Interpretiert man infolge von (17) die Zwischenschichtaktivitäten $a_j(\mathbf{x})$ nämlich als bedingte Zuordnungswahrscheinlichkeiten $p(j|\mathbf{x})$ des Datenpunkts $\mathbf{x}$ zum Codebuchvektor j und infolge von (31) ebenso $s_{\lambda j}$ als die bedingte Zuordnungswahrscheinlichkeit $p(\lambda|j)$ der Klasse λ zum Codebuchvektor j, so folgt aus der linearen Signalpropagation

$$y_\lambda(\mathbf{x}) = \sum_{j=1}^{N} s_{\lambda j} a_j(\mathbf{x}) \tag{33}$$

die *a-posteriori*-Wahrscheinlichkeit für die Klasse λ bei gegebenem Datenpunkt $\mathbf{x} \in X$:

$$p(\lambda|\mathbf{x}) = \sum_{j=1}^{N} p(\lambda|j) p(j|\mathbf{x}). \tag{34}$$

Das GRBF-Perzeptron mit der lokalen Lernregel (28) für die Ausgabegewichte stellt somit einen *Bayes-Klassifikator* dar. Die entsprechenden a-posteriori-Wahrscheinlichkeiten müssen hierbei nicht aus bekannten a-priori-Wahrscheinlichkeiten mittels des Satzes von Bayes (15) bestimmt werden.

Literatur

[1] R.D. Adams and M. Victor. *Principles of neurology.* McGraw-Hill Book Company, New York, 1985. 3rd ed.

[2] U. Adolphs. Selbstorganisierende Perzeptronstrukturen mit radialen Basisfunktionen. Diplomarbeit, Physik-Department, Technische Universität München, München, 1992.

[3] S. Albrecht. Ein neuronaler Klassifikator für Einzelphoneme. Diplomarbeit, Technische Universität München, Physik-Department, 1995.

[4] S. Albrecht, A. Wismüller, D.R. Dersch, and P. Tavan. Ein neuronaler Klassifikator für Einzelphoneme. In A. Wismüller and D.R. Dersch, editors, *Symposion über biologische Informationsverarbeitung und Neuronale Netze – SINN '95, Konfernzband.* Hanns-Seidel-Stiftung, München, 1995. In Vorbereitung.

[5] J.S. Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10:25–61, 1971.

[6] J.A. Anderson and E. Rosenfeld, editors. *Neurocomputing: Foundations of Research.* MIT Press, Cambridge, 1988.

[7] W. Bialek, F. Rieke, R. de Ruyter van Steveninck, and D. Warland. Reading a neural code. *Science*, 252:1854–1857, 1991.

[8] R. de Ruyter van Steveninck and F. Bialek. Real-time performance of and movement-sensitive neuron in the blowfly visual system: coding and information transfer in short spike sequences. *Proc. R. Soc. London, Ser. B234*, pages 379–414, 1988.

[9] D.R. Dersch. Eigenschaften neuronaler Vektorquantisierer. Doktorarbeit, Sektion Physik, Ludwig-Maximilians-Universität, München, 1995.

[10] D.R. Dersch, S. Albrecht, and P. Tavan. Hierarchical fuzzy clustering by a self-controlled annealing scheme. In *Proceedings of EUFIT*, 1995.

[11] W.G. Forssmann and C. Heym. *Neuroanatomie, 3. Auflage*. Springer-Verlag, Berlin, Heidelberg, New York, 1982.

[12] W. Gerstner and J.L. van Hemmen. Neuronale Netze mit pulskodierten Neuronen. In A. Wismüller and D.R. Dersch, editors, *Symposion über biologische Informationsverarbeitung und Neuronale Netze – SINN '95, Konferenzband*. Hanns-Seidel-Stiftung, München, 1995.

[13] F. Girosi and T. Poggio. Networks and the best approximation property. A. I. Memo 1164, Massachusetts Institute of Technology, 10 1989.

[14] R.M. Gray. Vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 4–29, April 1984.

[15] D.O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949. Teilweise abgedruckt in [6].

[16] M. Ito. *The cerebellum and neural control*. Raven Press, New York, 1984.

[17] W. Kahle. *Taschenatlas der Anatomie, Bd. 3: Nervensystem und Sinnesorgane, 3. Aufl.* Georg-Thieme-Verlag, Stuttgart, 1979.

[18] E.R. Kandel and J.H. Schwartz. *Principles of neural science, 3rd ed.* Elsevier, 1993.

[19] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[20] D. Marr. A theory of cerebellar cortex. *Journal of Physiology*, 202:437–470, 1969.

[21] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[22] R. Nieuwenhuys, J. Voogd, and C. van Huijzen. *Das Zentralnervensystem des Menschen.* Springer-Verlag, Berlin, Heidelberg, New York, 1980.

[23] K. Poeck. *Neurologie.* Springer-Verlag, Berlin, Heidelberg etc., 1987.

[24] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

[25] K. Rose, E. Gurewitz, and G.C. Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(11):589–594, 1990.

[26] K. Rose, E. Gurewitz, and G.C. Fox. Vector quantization by deterministic annealing. *IEEE Transactions on Information Theory*, 38(4):1249–1257, 1992.

[27] F. Rosenblatt. *Principles of Neurodynamics.* Spartans Books, 1962.

[28] A. Roth and B. Sakmann. Programmbausteine zur Simulation exzitatorischer synaptischer Übertragung zwischen detailliert rekonstruierten Nervenzellen. In A. Wismüller and D.R. Dersch, editors, *Symposion über biologische Informationsverarbeitung und Neuronale Netze – SINN '95, Konfernzband.* Hanns-Seidel-Stiftung, München, 1995. In Vorbereitung.

[29] D.E. Rumelhart and J.L. McClelland. Learning internal representations by error propagation. In *Parallel Distributed Processing*, volume I. M.I.T. Press, Cambridge, MA, 1986.

[30] R.F. Schmidt and G. Thews (Hrsg.). *Physiologie des Menschen.* Springer-Verlag, Berlin, Heidelberg, New York, 1983. 21. Auflage.

[31] G.M. Shepherd. *Neurobiologie.* Springer-Verlag, Berlin, Heidelberg etc., 1993.

[32] G.J. Stuart and B. Sakmann. Active propagation of somatic action potentials into neocortical cell dendrites. *Nature*, 367:69–72, 1994.

[33] A. Wismüller. Phänomenologie der Schmerzsymptomatik bei traumatischen Armplexusläsionen. Dissertation, Technische Universität München, München, 1992.

[34] A. Wismüller. Organisation des Zentralnervensystems. Skriptum, Selbstverlag, 1995.