

# Nichtlineare Filter und Neuronale Netzwerke

Autor(en): **Osuna, José A.**

Objektyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **82 (1991)**

Heft 13

PDF erstellt am: **12.07.2024**

Persistenter Link: <https://doi.org/10.5169/seals-902977>

## **Nutzungsbedingungen**

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

## **Haftungsausschluss**

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

# Nichtlineare Filter und Neuronale Netzwerke

José A. Osuna

*In der vorliegenden Einführung ins Gebiet der nichtlinearen Filter werden die Begriffe Systemidentifikation und -approximation präsentiert. Die klassische Approximation eines nichtlinearen Systems mit der abgebrochenen Volterrareihe weist grundlegende Nachteile auf. Diejenige mit Hilfe des Perceptrons, eines Neuronales Netzwerkes, ermöglicht einen neuen, praxisgerechten Ansatz für nichtlineare Filter.*

*Au cours de cette introduction dans le domaine des filtres non-linéaires, les notions d'identification et d'approximation de système sont présentés. L'approximation classique d'un système non-linéaire avec une série Volterra abrégée possède des désavantages fondamentaux; l'utilisation du Perceptron – un réseau de neurones – représente, par contre, une nouvelle approche pratique des filtres non-linéaires.*

## Adresse des Autors

José A. Osuna, Dipl. El.-Ing. ETH, Institut für Signal- und Informationsverarbeitung, ETH-Zentrum, 8092 Zürich

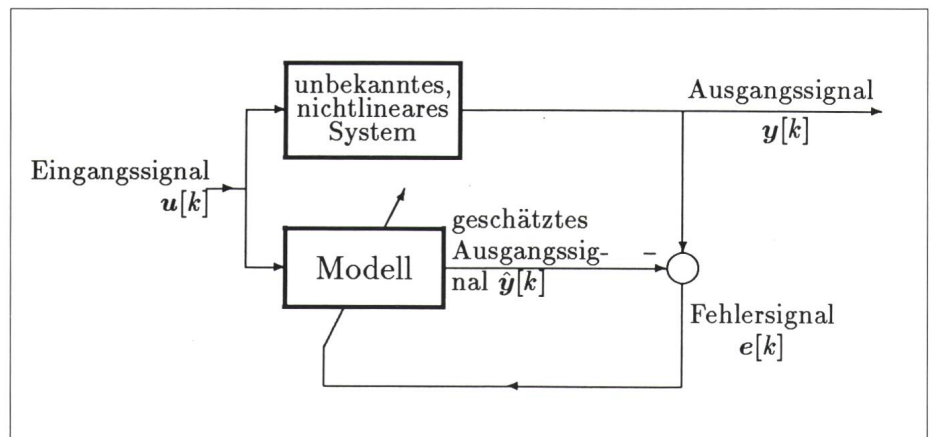


Bild 1 Systemidentifikation für zeitdiskrete Signale mit Vernachlässigung der Rauschquellen

Nichtlineare Filter lassen sich mit bisherigen Verfahren schwer realisieren. Neuronale Netzwerke bieten einen neuen Ansatz, um die nichtlineare Abbildung zwischen Filtereingang und -ausgang auf einfache Art und Weise zu verwirklichen. Im vorliegenden Artikel wollen wir die klassische Vorgehensweise und den neuen Ansatz kurz besprechen. Doch zunächst müssen wir das uns interessierende Problem definieren.

## Systemidentifikation

Dem Begriff Filter kommen in der Signalverarbeitung mehrere Bedeutungen zu. Unter einem Filter versteht man sowohl eine Blackbox, die zum Beispiel einen bestimmten Frequenzbereich aus einem elektrischen Signal herauszufiltern hat (Bandpass-Filter in der Telefonübertragung), als auch – im erweiterten Sinne – jedes System, das Information verarbeitet. Um dieses allgemeinere Konzept eines Filters zu erfassen, wollen wir den Begriff der Systemidentifikation einführen.

Im Hinblick auf die adaptive Filterung von Signalen in zeitvarianten,

kausalen Systemen beschränken wir uns auf zeitdiskrete Signale  $f[k]$  mit  $k > 0$ <sup>1</sup>.

Das Modell übernimmt die Funktion eines Filters, indem es das Eingangssignal verarbeitet, um das Ausgangssignal des unbekanntes, nichtlinearen Systems nachzubilden. Den oberen Teil von Bild 1 kann man sich auch abstrakt vorstellen, das heisst, das unbekanntes System muss nicht real existieren. Wichtig ist, dass seine nichtlineare Abbildung definiert ist. Die Filterfunktion des Modells besteht darin, die definierte Abbildung nachzuahmen. Um beim Beispiel des Bandpasses in der Telefonübertragung zu bleiben, kann man sich vorstellen, dass das Gerät mit der genannten Filterfunktion dasjenige Modell ist, welches die definierte, in diesem konkreten Falle bekannte Abbildung *Schneide alle Frequenzen unterhalb  $f_u$  und oberhalb  $f_o$  ab*, nachzuahmen hat.

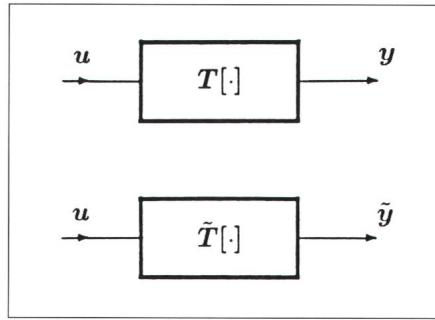
<sup>1</sup>  $f[k]$  folgt zum Beispiel aus der Abtastung des zeitkontinuierlichen Signals  $f(t)$  mit der Abtastperiode  $T = 1$ :  $f[k]$  ist die Auswertung der Funktion  $f(t)$  an den diskreten Zeitpunkten  $t = kT$ .

Wir sprechen von nichtlinearen, adaptiven Filtern, wenn die nichtlineare Abbildungsvorschrift mit der Zeit variieren kann. Das Modell muss dann seine Systemparameter in einer endlich langen Zeit wieder so einstellen, dass die neue Abbildungsvorschrift möglichst genau nachgeahmt wird. Das Modell adaptiert sich also auf die neue Umgebung, damit das sich verändernde, unbekannte System laufend identifiziert werden kann. Der sogenannte Adaptionsalgorithmus, der aufgrund der Beobachtung des Eingangs- und Ausgangssignals die Systemparameter des Modells einstellt, muss entsprechend schnell sein.

Wir setzen voraus, dass das unbekannte System nichtlinear ist, das heisst, wir betrachten nichtlineare Filter. Nichtlinear bezieht sich, wie bereits oben angetönt wurde, auf die nichtlineare Abbildung vom Eingang des Filters auf seinen Ausgang. Es hat keinen Sinn, mit den nichtlinearen neuronalen Netzwerken lineare Abbildungen realisieren zu wollen, da sich der lineare Ansatz eines linearen Systems dazu offensichtlich besser eignet. Ein lineares Modell genügt andererseits für die Approximation einer nichtlinearen Abbildungsvorschrift des unbekannten Systems natürlich nicht, wenn das Fehlersignal für einen begrenzten Bereich der Eingangsdaten beliebig klein gemacht werden soll. Der Nachteil des klassischen, nichtlinearen Ansatzes ist, dass der Rechenaufwand für die Adaptation der Modellparameter ins Unermessliche steigt, wenn die Systemidentifikation möglichst genau erfolgen soll. Wie gezeigt werden soll, können die neuronalen Netzwerke gegenüber den klassischen Lösungen ihre Vorteile voll ausspielen und sie ermöglichen eine praxisingerechte Realisierung von nichtlinearen Filtern.

### Systemapproximation

In Bild 2 ist das unbekannte System durch einen (unbekannten) nichtlinearen Operator  $T[\cdot]$  repräsentiert, seine «exakte»<sup>2</sup> Approximation durch den Operator  $\tilde{T}[\cdot]$ .  $\tilde{T}$  ist nicht mit dem Modell in Bild 1 gleichzusetzen, weil wir bei der praktischen Realisierung nur einen endlichen Aufwand betreiben können. Wir sagen, dass das System  $\tilde{T}[\cdot]$  das unbekannte System  $T[\cdot]$



**Bild 2** Unbekanntes System  $T[\cdot]$  und dessen Approximation  $\tilde{T}[\cdot]$  mit Eingangssignal  $u[k]$   $u$  in  $R^n$  und Ausgangssignale  $y[k]$ ,  $\tilde{y}[k]$   $y$ ,  $\tilde{y}$  in  $R^n$

beliebig genau approximiert, wenn für jedes Eingangssignal  $u[k]$  in einem  $n$ -dimensionalen Unterraum  $U$  von  $R^n$  die Ausgangssignale des unbekannten Systems und des approximierenden Systems für alle Zeitpunkte  $k$  bis auf einen wählbar kleinen absoluten Fehler identisch sind.

Das approximierende System  $\tilde{T}[\cdot]$  mit unendlich vielen Parametern wird also in einer zweiten Phase durch ein «endliches» Modell  $\hat{T}[\cdot]$  ersetzt, welches, mit Ausgangssignal  $\hat{y}$  in das Bild 1 eingesetzt, alle nichtlinearen Abbildungsvorschriften  $T[\cdot]$ , die mit  $\tilde{T}[\cdot]$  beliebig genau angenähert werden können, modelliert.

Fassen wir die Grundgedanken der Systemidentifikation und -approximation zusammen: Nach der Wahl der Approximation  $\tilde{T}[\cdot]$  und des damit verbundenen Modells  $\hat{T}[\cdot]$  stellen sich dessen freie Parameter online so ein, dass eine (im allgemeinen Fall mit einem Fehler behaftete) Identifikation des unbekannten Systems  $T[\cdot]$  resultiert.

### Das Volterrafilter

Damit wir uns die folgenden Ausführungen besser vergegenwärtigen können, betrachten wir den eindimensionalen Fall, nehmen also an, dass sich alle Signale von Bild 1 im eindimensionalen Raum  $R^1$  befinden und demzufolge nur eine Komponente besitzen. Wir wissen zu Beginn, dass das unbekannte System nichtlinear ist, wir haben aber sonst überhaupt keine A-priori-Information über seinen Aufbau. Es kann gezeigt werden, dass sowohl der klassische wie auch der neuronale nichtlineare Ansatz für das Modell den nichtlinearen Operator  $T[\cdot]$  beliebig genau approximieren. Mit einem grossen Implementations-

aufwand kann also der Restfehler beliebig klein gehalten werden.

Der klassische Ansatz für eine nichtlineare Approximation, die dem unbekanntem, nichtlinearen Operator  $T[\cdot]$  möglichst gut entsprechen soll, ist die abgebrochene<sup>3</sup> Volterrareihe, die weiter unten näher erklärt wird:

$$\tilde{y}[k] = h_0 +$$

$$\sum_{i=0}^{\infty} h_1[i]u[k-i] + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} h_2[i_1, i_2]u[k-i_1]u[k-i_2] + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \sum_{i_3=0}^{\infty} (h_3[i_1, i_2, i_3]u[k-i_1] \cdot u[k-i_2]u[k-i_3]) + \dots + \sum_{i_1=0}^{\infty} \dots \sum_{i_n=0}^{\infty} (h_n[i_1, \dots, i_n] \cdot u[k-i_1] \dots u[k-i_n]) \quad (1)$$

(Für mehrdimensionale Signale,  $m$ ,  $n > 1$ , sind die entsprechenden Änderungen vorzunehmen, das heisst zum Beispiel für die ersten zwei Summanden von Gl. 1, dass  $h_0$  durch den  $(m \times 1)$ -Vektor  $h_0$  zu ersetzen ist, und dass  $h_1[i]$  auf die  $(m \times n)$ -Matrix  $H_1[i]$  zu erweitern ist.)

In [1; 2] wird die auf der Volterrareihe basierende Wienertheorie präsentiert. Ihre Behandlung sprengt den Rahmen dieses Artikels. Wir wollen uns darauf beschränken, den Aufbau und die Bedeutung von Gl. 1 zu verstehen.

Lässt man in Gl. 1 alle Summanden weg bis auf den zweiten, dann folgt daraus die vertraute Formel für die zeitdiskrete Faltung des Eingangssignals  $u[k]$  mit der zeitdiskreten Impulsantwort  $h_1[k]$  eines linearen, zeitinvarianten und kausalen Systems:

$$y_1[k] = \sum_{i=0}^{\infty} h_1[i]u[k-i] = h_1[k] * u[k] \quad (2)$$

Aus Gl. 2 ist zu erkennen, dass zum Zeitpunkt  $k$  der vor  $\alpha$  Zeiteinheiten am Systemeingang vorhandene Si-

<sup>3</sup> Eine Reihe hat im allgemeinen unendlich viele nichtverschwindende Summanden. Brechen wir sie ab, indem wir die Glieder ab einer gewissen Stelle vernachlässigen, haben wir es mit einer Approximation dieser Reihe – also mit einem Polynom – zu tun.

<sup>2</sup> Die Approximation ist exakt, wenn wir unendlich viele Ressourcen zur Verfügung haben.

**Bedeutung der Systemidentifikation**

Ein unbekanntes System (Bild 1) bildet ein Eingangssignal auf sein Ausgangssignal ab. In unserem Falle handelt es sich um eine *nichtlineare* Abbildung von einem zeitdiskreten, im allgemeinen Fall mehrdimensionalen Eingangssignal  $\mathbf{u}[k] = [u_1[k] \ u_2[k] \ \dots \ u_n[k]]'$  ( $\mathbf{u}$  in  $\mathbb{R}^n$ ) auf das wiederum zeitdiskrete Ausgangssignal  $\mathbf{y}[k] = [y_1[k] \ y_2[k] \ \dots \ y_m[k]]'$  ( $\mathbf{y}$  in  $\mathbb{R}^m$ ) für alle  $k \in \mathbb{Z}^+$  (positive ganze Zahlen). Das Modell, das keinerlei Informationen über das unbekannte, nichtlineare System besitzt, basiert auf einer wie auch immer gewählten Approximation von nichtlinearen Systemen. Es versucht, durch bloße Beobachtung des Eingangs- und Ausgangssignals die Funktion des verhaltenen, nichtlinearen Systems zu identifizieren. Dabei adaptiert es sich nach einem gewissen Qualitätskriterium, indem es seine Systemparameter solange verändert, bis das aus dem Ausgangssignal und seiner Schätzung gebildete Fehlersignal  $\mathbf{e}$  ein minimales Mass erreicht hat. Das Fehlersignal strebt also für aufeinanderfolgende Signalzeitpunkte zu einem absolut möglichst kleinen Wert, das heisst, dass sich beide Systeme immer ähnlicher verhalten, und das Modell, zum Beispiel ein neuronales Netzwerk, das unbekanntes System nachbildet.

einzelnen Summanden der Volterrareihe entsprechen den Summanden der Taylorsche Potenzreihe, wenn wir die zeitdiskreten Faltungen mitberücksichtigen und  $u_0 = 0$  setzen. In Tab. I sind die aufgeführten Gedankenschritte zusammengefasst.

Der approximative Charakter des klassischen Ansatzes folgt aus der Tatsache, dass in Gl. 1 die Volterrareihe abgebrochen wird, was der Taylorsche Approximation entspricht. Wir können aus praktischen Gründen nicht unendlich viele Glieder der Reihe berücksichtigen und müssen uns mit einer endlichen Anzahl zufriedengeben, die das nichtlineare System nicht mehr exakt darstellt, sondern im Sinne von Ungleichung (8) nur noch approximiert.

Fassen wir kurz zusammen: Volterra-Filter sind eine spezielle Klasse von nichtlinearen Filtern, die dem auf den nichtlinearen Fall erweiterten Impulsantwort-Konzept entspringen.

**Vorteile und Nachteile der Volterrareihe**

Ein lineares, zeitinvariantes und zeitdiskretes System ist durch seine Impulsantwort  $h_1[i]$  vollständig charakterisiert. Systeme, bei denen die

Signalwert  $u[k - \alpha]$  mit  $h_1[\alpha]$  gewichtet wird. Die Impulsantwort  $h_1[k]$  beschreibt also das Gedächtnis des Systems; vergangene Eingangssignale  $u[j]$  ( $0 \leq j < k$ ) müssen bei nichtverschwindender Impulsantwort  $h_1[j]$  für die Ermittlung der Systemantwort  $y_1[k]$  mitberücksichtigt werden.

Ein lineares System ohne Gedächtnis hat die Impulsantwort

$$h_1[k] = c\delta_0[k] \tag{3}$$

wobei  $c$  eine reelle Konstante ist, und

$$\delta_0[k] := \begin{cases} 1 & \text{für } k = 0 \\ 0 & \text{sonst} \end{cases} \tag{4}$$

den Einheitsimpuls definiert.

Die zeitdiskrete Systemantwort für ein zeitinvariantes, lineares und kausales System ohne Gedächtnis lautet

$$y[k] = cu[k] \tag{5}$$

Wenn  $y(u[k])$  über  $u[k]$  aufgetragen wird, dann bestimmt die Gl. 5 Punkte  $(u_k, y_k)$  auf der Geraden

$$y(u) = cu \tag{6}$$

Für ein nichtlineares, zeitinvariantes und kausales System ohne Gedächtnis ist der Graph von  $y(u)$  über  $u$  nicht mehr eine Gerade, sondern eine Kurve  $y(u) = f(u)$  ( $f: \mathbb{R} \rightarrow \mathbb{R}$ , das heisst  $f(u)$  ist eine nichtlineare Abbildung von  $\mathbb{R}$  nach  $\mathbb{R}$ ), die unter bestimmten Einschränkungen durch die Taylorsche Potenzreihe

$$y(u) = f(u) = \sum_{n=0}^{\infty} c_n(u - u_0)^n$$

mit 
$$c_n = \frac{f^{(n)}(u_0)}{n!} \tag{7}$$

dargestellt werden kann<sup>4</sup>.

Falls die unendliche Potenzreihe in Gl. 7 abgebrochen wird, sprechen wir von der Taylorsche Approximation. Die Approximation ist so zu verstehen, dass sovieler Summanden mitber-

1. lineares System mit Gedächtnis	→	$y[k] = h_1[k] * u[k]$
2. lineares System ohne Gedächtnis	→	$y[k] = cu[k]$
3. nichtlineares System ohne Gedächtnis	→	$y[k] = \text{Taylorreihe}$
4. nichtlineares System mit Gedächtnis	→	$y[k] = \text{Volterrareihe}$

**Tabelle I Lineare und nichtlineare zeitdiskrete Systeme mit und ohne Gedächtnis und exakte Darstellungen der entsprechenden Systemantworten**

rücksichtigt werden, wie es für das Erfüllen der nächsten Ungleichung nötig ist:

$$\sup_{u \in D_y} |y(u) - \tilde{y}(u)| < \epsilon$$

mit 
$$\tilde{y}(u) = \sum_{n=0}^N c_n(u - u_0)^n \tag{8}$$

und  $D_y \subset \mathbb{R}$ : Definitionsbereich der Taylorreihe  $y(u)$  sowie  $\epsilon$ , eine kleine positive Zahl.

Die Volterrareihe ist anschaulich nichts anderes als die Erweiterung von Gl. 7 auf Systeme mit Gedächtnis: Die

Impulsantwort auf den Wert Null abklingt (FIR-Filter), werden durch die  $I$  Werte  $h_1[0], \dots, h_1[I - 1]$  bestimmt, wobei  $I - 1$  die Gedächtnislänge des linearen Systems darstellt. Analog ist ein nichtlineares, zeitinvariantes und zeitdiskretes System, das durch eine Volterrareihe dargestellt werden kann<sup>5</sup>, durch sie vollständig charakterisiert. Verschwinden deren Volterra Kernels  $h_n[i_1, \dots, i_n]$  für  $n > n_0$ , dann wird das nichtlineare System durch die ersten  $n_0 + 1$  Volterra Kernels  $h_0$  bis  $h_{n_0}[i_1, \dots, i_{n_0}]$  (vgl. Gl. 1 für  $n = n_0$ ) vollständig bestimmt. Wir betrachten nur symmetrische Kernels, da es sich

<sup>4</sup>  $u_0$  ist der Entwicklungspunkt der Reihe, und  $f^{(n)}(u_0)$  stellt die  $n$ -te Ableitung der nichtlinearen Funktion  $f(u)$  nach  $u$  an der Stelle  $u_0$  dar.

<sup>5</sup> Das Konvergenzproblem wird weiter unten besprochen.

zeigen lässt [1], dass diese Bedingung keine Einschränkung der Allgemeinheit ist. Der Wert eines symmetrischen Kerns  $h_n[i_1, \dots, i_n]$  bleibt für alle  $n!$  Permutationen seiner Argumente  $i_1, \dots, i_n$  unverändert, so dass zum Beispiel für  $n = 2$  folgendes gilt:

$$h_2[i_1, i_2] = h_2[i_2, i_1] \text{ für alle } i_1, i_2 \geq 0$$

Die Volterra Kernels  $h_n[i_1, \dots, i_n]$  sind die durch den Adaptationsalgorithmus einzustellenden Parameter des Modells (Bild 1). Die Volterrareihe hat den Vorteil, dass sie linear ist bezüglich der  $h_n[i_1, \dots, i_n]$ , das heisst, das Modell ist linear bezüglich der freien Parameter  $h_n[i_1, \dots, i_n]$ . Der Fehler

$$e[k] = y[k] - \hat{y}[k] \quad (9)$$

ist ebenfalls linear von den Volterra Kernels abhängig. Bei einem linearen Verhalten des Modells bezüglich der zu adaptierenden Parameter besitzt die durch den MSE<sup>6</sup> aufgespannte Fläche im  $(L+1)$ -dimensionalen Raum ( $L$  sei die Anzahl der zu adaptierenden Parameter) ein rein quadratisches Verhalten mit einem einzigen Minimum. Als Beispiel einer quadratischen MSE-Fläche für  $n = 1$  und  $I = 1$  sei der in Bild 3 dargestellte Paraboloid

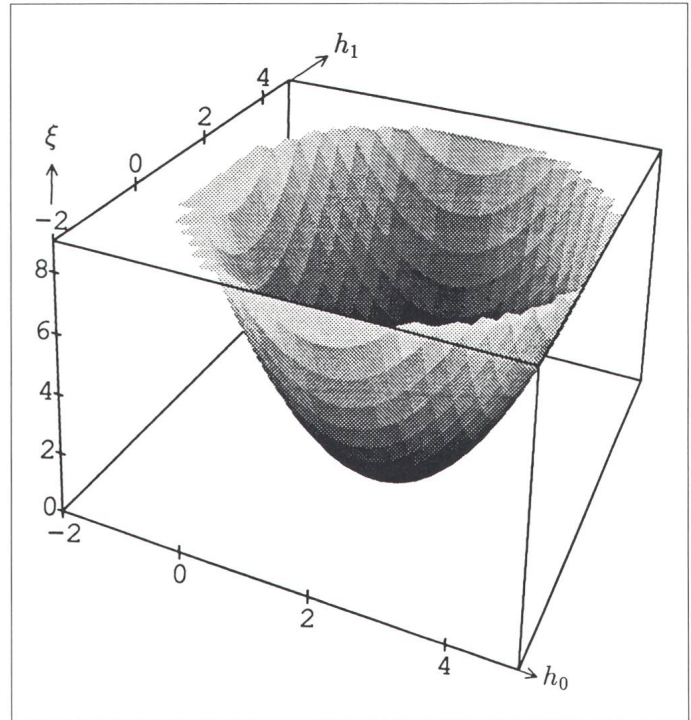
$$\xi = (h_0 - 2)^2 + (h_1 - 1)^2 \quad (10)$$

angegeben. Ein auf dem Gradientenverfahren basierender Adaptationsalgorithmus, der das globale Minimum  $\xi_{\min}$  der MSE-Fläche sucht, kann bei einer quadratischen Fläche nicht in einem lokalen Minimum verharren (weil es keines gibt), so dass das Erreichen von  $\xi_{\min}$  gewährleistet ist. Das Problem der lokalen Minima fällt beim Approximieren des unbekannt Systems durch die abgebrochene Volterrareihe somit weg.

Trotz dieser schönen Eigenschaft treten mit dem klassischen Ansatz Schwierigkeiten auf:

Eine erste Schwierigkeit bei der Approximation durch die abgebrochene Volterrareihe ist, dass unsere adaptiven Filter nicht wie oben angenommen zeitinvariante Systeme sind. Vielmehr soll der nichtlineare, sich zeitlich verändernde Operator  $T[\cdot]$  laufend identifiziert werden können. Der In-

**Bild 3**  
Dreidimensionale Skizze des durch Gl. 10 gegebenen Paraboloids ( $L = 2$ )



genieur meistert dieses Problem folgendermassen: Er trifft die Annahme, dass die Zeitkonstante, mit der die Änderung von  $T[\cdot]$  erfolgt, sehr viel grösser als die Zeitkonstante des Adaptationsalgorithmus ist. Das unbekannte System sieht für die Systemidentifikation dann so aus, wie wenn es zeitinvariant wäre. Wir betonen, dass dieses Vorgehen eine vereinfachende Annahme ist, um zeitveränderliche Systeme mathematisch in den Griff zu bekommen.

Um die nächste Schwierigkeit besser zu verstehen, wollen wir den Volterra Operator  $n$ -ter Ordnung  $H_n[\cdot]$  definieren:

$$H_n[u[k]] := \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} (h_n[i_1, \dots, i_n] \cdot u[k - i_1] \cdots u[k - i_n]) \quad (11)$$

Damit können wir die abgebrochene Volterrareihe (Gl. 1) abgekürzt wie folgt aufschreiben:

$$\hat{y}[k] = h_0 + H_1[u[k]] + H_2[u[k]] + \cdots + H_n[u[k]] \quad (12)$$

Eine der grössten Schwierigkeiten beim Approximieren des unbekannt Systems durch die abgebrochene Volterrareihe stellt sich in der Konvergenzanalyse. Die Volterrareihe ist

eine Potenzreihe mit Gedächtnis. Dies kann schnell eingesehen werden, wenn wir das Eingangssignal mit einem reellen Faktor  $c$  multiplizieren und  $cu[k]$  als neues Eingangssignal betrachten. Wir gehen von Gl. 12 aus und sehen aus der nächsten Gleichung, dass wir die Volterrareihe als Potenzreihe des Amplitudenfaktors  $c$  interpretieren können.

$$y[k] = h_0 + \sum_{n=1}^{\infty} H_n[cu[k]] = h_0 + \sum_{n=1}^{\infty} c^n H_n[u[k]] \quad (13)$$

Gl. 13 ist eine Potenzreihe mit Gedächtnis, weil die Volterra-Operatoren  $H_n[\cdot]$  Faltungen darstellen.

Aus der Analysis wissen wir, dass eine Potenzreihe nicht immer konvergieren muss, das heisst, gewisse nicht-lineare Funktionen  $f(u)$  können nicht für alle  $u$  durch die Taylorsche Potenzreihe in Gl. 7 exakt dargestellt werden. Als Beispiel sei hier die Signumfunktion

$$f(u) = |E| \operatorname{sgn}[u] \quad ; \quad E \in \mathbb{R} := \begin{cases} -|E| & \text{für } u < 0 \\ 0 & \text{für } u = 0 \\ |E| & \text{für } 0 < u \end{cases} \quad (14)$$

gegeben. Es existiert keine konvergierende Taylorreihe um  $u_0 = 0$  für diese

<sup>6</sup> MSE steht für den englischen Ausdruck Mean Squared Error, den Erwartungswert des Fehlers im Quadrat, und wird folgendermassen definiert:  $\xi = E[|e|^2]$  (z.B. [3]).

**Tabelle II**  
Auswertung von  $L$   
nach Gl. 15 und 16

$I$	$K$									
	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	1	3	6	10	15	21	28	36	45	55
3	1	4	10	20	35	56	84	120	165	220
4	1	5	15	35	70	126	210	330	495	715
5	1	6	21	56	126	252	462	792	1287	2002
6	1	7	28	84	210	462	924	1716	3003	5005
7	1	8	36	120	330	792	1716	3432	6435	11440
8	1	9	45	165	495	1287	3003	6435	12870	24310
9	1	10	55	220	715	2002	5005	11440	24310	48620
10	1	11	66	286	1001	3003	8008	19448	43758	92378

ideale Begrenzungsfunktion. Das heisst, hier existiert die Taylorsche Potenzreihe für keinen Wert von  $u$ .

Da die Volterrareihe eine Potenzreihe darstellt, treten wegen des Konvergenzproblems Einschränkungen auf beim Approximieren des nichtlinearen Operators  $T[\cdot]$ . Das Konvergenzproblem ist so zu deuten, dass die Volterrareihe bei gewissen unbekannt Systemen nicht für jedes Eingangssignal  $u[k]$  konvergiert. Der Ansatz in Gl. 1 kann also nicht vorbehaltlos gemacht werden.

Die dritte Schwierigkeit haben wir bereits oben erwähnt: Der Ansatz der abgebrochenen Volterrareihe hat die Anwendung der entsprechenden nichtlinearen Filter wegen der aufwendigen Berechnung der Systemparameter, das heisst der Volterra Kernels  $h_n[i_1, \dots, i_n]$ , stark beeinträchtigt.

Berücksichtigen wir die symmetrische Eigenschaft der Kernels, sehen wir, dass  $h_n[i_1, \dots, i_n]^7$  durch

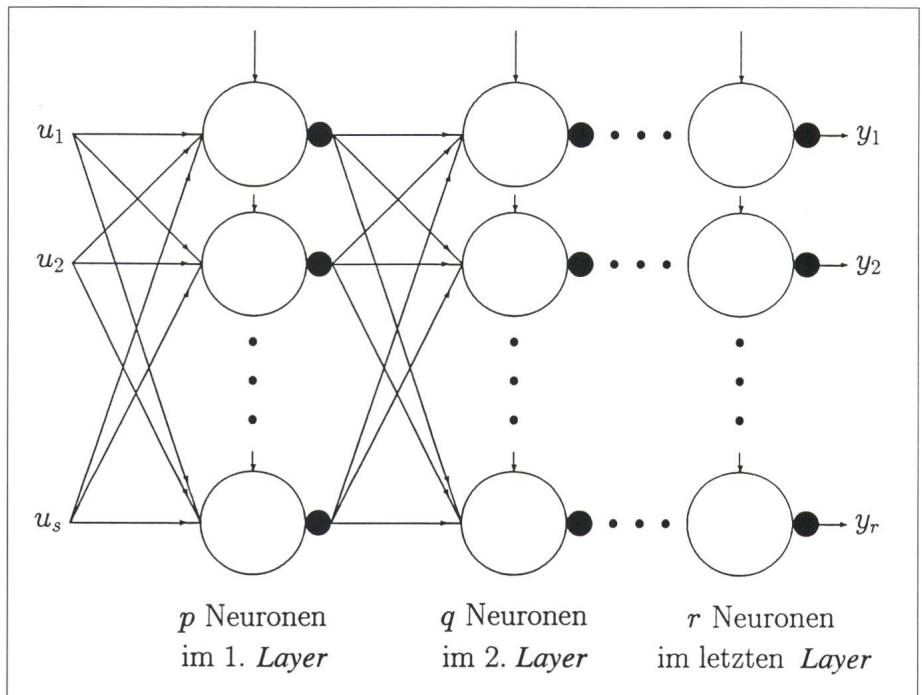
$$I_n = \binom{I+n-1}{n} \quad (15)$$

Werte bestimmt wird. Wir nehmen in allen Faltungen von Gl. 1 eine Gedächtnislänge von  $I-1$  an, das heisst, wir summieren von  $i_i = 0$  bis  $i_i = I-1$  auf für alle Laufvariablen  $i_i$ . Die Gl. 15 stellt die Anzahl der Kombinationen zu je  $n$  Elementen aus  $I$  verschiedenen Elementen mit Wiederholung der Elemente dar. (Beispiel:  $I = 4$ , Elemente  $a, b, c, d$ ;  $n = 2$ ; Kombinationen:  $aa, ab, ac, ad, bb, bc, bd, cc, cd, dd$ ). Wir müssen also  $I_n$  Terme berechnen, um  $h_n[i_1, \dots, i_n]$  zu bestimmen. Nehmen wir weiterhin an, dass wir die Volterrareihe nach  $K = n+1$  Summanden abrechnen, weil wir uns bei der Ap-

<sup>7</sup>  $h_n[i_1, \dots, i_n]$  erstreckt sich über die  $n$  Dimensionen  $i_i$  bis  $i_n$ .

proximation von  $T[\cdot]$  mit dem verbleibenden Fehler zufriedengeben. Dann ist

$$L = \sum_{j=1}^K I_{j-1} \quad (16)$$



**Bild 4** Das Multi-Layer-Perceptron mit Eingangssignalen  $u_1$  bis  $u_s$  und den Ausgangssignalen  $y_1$  bis  $y_r$ .

die Anzahl der zu adaptierenden Parameter im Modell.

$L$  explodiert mit wachsenden Parametern  $I$  und  $K$ . Die Tabelle II listet die Funktionswerte von  $L$  für  $I, K = 1, \dots, 10$  auf. Mit der rapid steigenden Anzahl freier Parameter in Funktion der Gedächtnislänge  $I-1$  und der  $K$  Summanden der abgebrochenen Volterrareihe steigt auch die benötigte Zeit, um die entsprechenden Rechenoperationen im Adaptionsalgorithmus durchzuführen. Aus praktischen Gründen ist im allgemeinen der Ansatz der abgebrochenen Volterrareihe

ungeeignet, wenn  $\xi_{\min}$  möglichst klein gehalten werden soll.

### Das Perceptron

In [4] findet sich eine allgemeine Einführung in das Gebiet der neuronalen Netzwerke. Hier wird nur in Zusammenhang mit nichtlinearen Filtern darauf eingegangen. Das Perceptron, ein neuronales Netzwerk ohne Rückkopplungen, gehört zu den sogenannten Feedforward-Netzwerken und besitzt eine Struktur entsprechend Bild 4. Die Eingangssignale  $u_1$  bis  $u_s$  sind hier der Einfachheit halber als Skalare angenommen. Dasselbe gilt für die Ausgangssignale  $y_1$  bis  $y_r$ . Der Aufbau der einzelnen Neuronen wird in Bild 5 aufgezeigt: Die Ein-

gangssignale  $v_i$  zu jedem einzelnen Neuron werden mit den Skalaren  $a_i$  gewichtet und aufsummiert. Dazu wird der konstante Term  $b$  aufaddiert. Die resultierende Summe  $x$  ist das Argument der nichtlinearen Funktion  $f(\cdot)$ <sup>8</sup>.

<sup>8</sup> Die Nichtlinearität macht die neuronalen Netzwerke überhaupt interessant. Wäre sie nicht vorhanden, könnte man z.B. komplizierte, aus vielen Neuronen aufgebaute Netzwerke auf ein äquivalentes mit einfacherem Aufbau zurückführen. Das heisst, dass ohne die nichtlineare Eigenschaft der neuronalen Netzwerke bestimmte Leistungsgrenzen nicht überschritten werden könnten.

Für die nichtlineare Aktivierungsfunktion  $f(\cdot)$  des neuronalen Netzwerkes setzen wir eine sigmoidale Nichtlinearität  $\Phi(\cdot)$  ein.  $\Phi(\cdot)$  wird in Bild 6 definiert und skizziert. Der Ausgang  $w$  des Neurons ist der Funktionswert von  $f(\cdot) = \Phi(\cdot)$  an der Stelle  $x$ . Die einzelnen Neuronen des Netzwerkes haben im allgemeinen verschiedene Werte für die Gewichte  $a_1$  bis  $a_p$  und für den konstanten Term  $b$ . Die einzelne «Nervenzelle» ist also durch ihre Gewichtungsfaktoren  $a_1$  bis  $a_p$  und durch ihren konstanten Term  $b$  charakterisiert.

Bei einer Approximation  $\hat{T}[\cdot]$  des unbekanntes, nichtlinearen Systems  $T[\cdot]$  mit einem Zweischichten-Perceptron sieht der Filterausgang  $\hat{y}$  folgendermassen aus [5; 6] (vgl. Bild 7):

$$\hat{y} = C\phi(AU + b) + d \quad (17)$$

mit

$$A = \begin{bmatrix} a'_{10} & a'_{11} & \dots & a'_{1I-1} \\ a'_{20} & a'_{21} & \dots & a'_{2I-1} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{K0} & a'_{K1} & \dots & a'_{KI-1} \end{bmatrix},$$

$$a'_{ij} = \begin{bmatrix} (a_{ij})_1 & (a_{ij})_2 & \dots & (a_{ij})_n \end{bmatrix},$$

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1K} \\ c_{21} & c_{22} & \dots & c_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mK} \end{bmatrix},$$

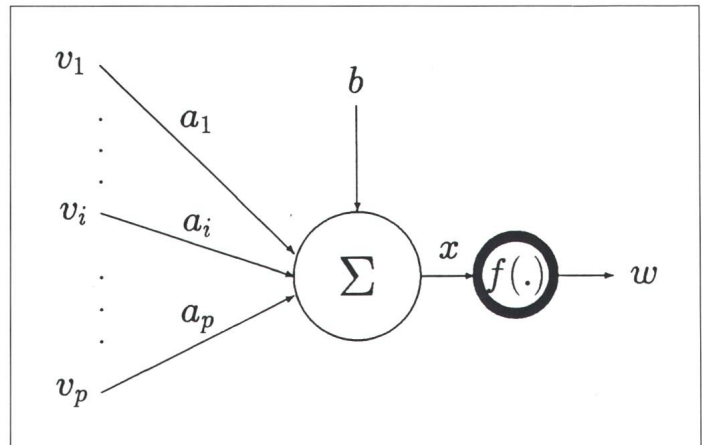
$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix},$$

$$U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{I-1} \end{bmatrix} \quad \text{und} \quad \phi(\cdot) = \begin{bmatrix} \phi(\cdot) \\ \phi(\cdot) \\ \vdots \\ \phi(\cdot) \end{bmatrix},$$

wobei  $\Phi(\mathbf{x}) = [\Phi(x_1) \Phi(x_2) \dots \Phi(x_K)]'$  für  $\mathbf{x} \in R^K$ ,  $u_0 = \mathbf{u}[k]$ ,  $u_{I-1} = \mathbf{u}[k-(I-1)]$  in  $R^n$  und  $\hat{y} = \hat{y}[k]$  in  $R^m$  ist. Die zweite Schicht (Layer) am Ausgang kommt ohne Nichtlinearität aus.

In [6] wird gezeigt, dass das Zweischichten-Perceptron jedes zeitinvariante System  $T[\cdot]$ , das vergangene

**Bild 5**  
**Das künstliche Neuron**  
 $x$  Summe der mit  $a_i$  gewichteten Eingangssignale  $v_i$ ,  $i = 1 \dots p$   
 $b$  konstanter Term  
 $f(\cdot)$  nichtlineare Funktion  
 $w$  Ausgang



Eingangswerte vergessen kann, beliebig genau approximieren kann. In (17) haben wir eine Gedächtnislänge von  $I-1$  angesetzt und  $K$  Neuronen («Summanden») der Approximation berücksichtigt. Um einen Vergleich mit dem vorhin besprochenen Volterrafilter machen zu können, setzen wir  $n = m = 1$  und betrachten wieder eindimensionale Signale.

### Eigenschaften des Zweischichten-Perceptrons

Gl. 17 sieht für den eindimensionalen Fall ( $n = m = 1$ ) folgendermassen aus:

$$\hat{y} = \begin{bmatrix} c_1 & c_2 & \dots & c_K \end{bmatrix} \phi(\mathbf{x}) + d$$

$$\mathbf{x} = \begin{bmatrix} a_{10} & a_{11} & \dots & a_{1I-1} \\ a_{20} & a_{21} & \dots & a_{2I-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K0} & a_{K1} & \dots & a_{KI-1} \end{bmatrix}$$

$$\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{I-1} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix} \quad (18)$$

Wir setzen diese neuronale Approximation als Modell  $\hat{T}[\cdot]$  in Bild 1 ein (für eindimensionale Signale). Die Anzahl  $L$  der zu adaptierenden Parameter des Perceptron-Modells  $\hat{T}[\cdot]$  beträgt

$$L = K(I + 2) + 1 \quad (19)$$

Wir berücksichtigen dabei alle freien Parameter des Zweischichten-Perceptrons (die  $K \cdot I$  Elemente der  $A$ -Matrix,  $c_1$  bis  $c_K$ ,  $b_1$  bis  $b_K$  und  $d$ ).

$L$  wächst linear mit  $K$  und  $I$ . In Tabelle III ist die Trennlinie eingezeichnet, ab welcher  $L$ (Perceptron) kleiner als  $L$ (Volterrafilter) ist. Wichtig ist, dass für genauere Approximationen des unbekanntes Systems im Falle des Zweischichten-Perceptrons die Anzahl freier Parameter mit grösseren  $K$  und  $I$  nicht explodiert.

Wenn wir es mit zeitvarianten Systemen zu tun haben, dann müssen wir beim Perceptron-Modell wie vorhin beim Volterrafilter annehmen, dass das unbekanntes System  $T[\cdot]$  sich langsam ändert. So hat der Adaptionsalgorithmus genügend Zeit, um die freien Parameter des neuronalen Netzwerkes auf die bestmögliche Identifikation einzustellen, bevor er bei jeder Änderung von  $T[\cdot]$  die Parameterwerte auf die neue unbekanntes Abbildung trimmt.

Als Adaptionsalgorithmus eignet sich ein Gradientenverfahren<sup>9</sup> wie der vom Perceptron her bestbekannte Backpropagation-Algorithmus [7]. In der Regel handelt es sich bei den Adaptionsalgorithmen um nichtlineare Verfahren, welche die Parameterwerte des Modells  $\hat{T}[\cdot]$  nach einem gewissen Qualitätskriterium, zum Beispiel nach dem minimalen Erwartungswert des Fehlers im Quadrat, einstellen. Die Algorithmen wurden für lineare Filter entwickelt ([3; 8]). Wir können sie für unser Problem prinzipiell übernehmen, da die

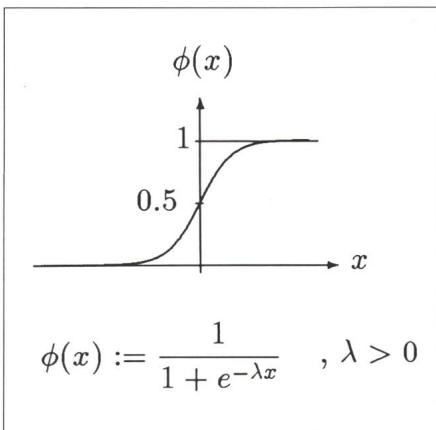
<sup>9</sup> Die Gradientenverfahren suchen den tiefsten Punkt der MSE-Fläche (Bild 3) entlang der grössten negativen Steilheit.

**Tabelle III**  
Anzahl der zu adaptierenden Parameter

I	K									
	1	2	3	4	5	6	7	8	9	10
1	4	7	10	13	16	19	22	25	28	31
2	5	9	13	17	21	25	29	33	37	41
3	6	11	16	21	26	31	36	41	46	51
4	7	13	19	25	31	37	43	49	55	61
5	8	15	22	29	36	43	50	57	64	71
6	9	17	25	33	41	49	57	65	73	81
7	10	19	28	37	46	55	64	73	82	91
8	11	21	31	41	51	61	71	81	91	101
9	12	23	34	45	56	67	78	89	100	111
10	13	25	37	49	61	73	85	97	109	121

Systemapproximation und die Adaptation der Modellparameter zwei getrennte Aufgaben sind. Für den einzelnen Fall muss überprüft werden, ob der Adaptationsalgorithmus beim nichtlinearen System auch konvergiert.

Zum Abschluss wollen wir betonen, dass die Approximation einer unbe-



**Bild 6** Sigmoidale Aktivierungsfunktion  $\Phi(x)$  des neuronalen Netzwerkes  
 $f(\cdot) = \Phi(\cdot)$  für alle Neuronen

kannten, nichtlinearen Abbildung mit dem Zweischichten-Perceptron garantiert für diejenigen nichtlinearen Systemen konvergiert, welche vergangene Eingangswerte mit der Zeit vergessen. Für diese weite Klasse von Systemen lässt sich mit dem vorgestellten neuronalen Netzwerk ein nichtlineares Filter realisieren, bei welchem bekannte Algorithmen die praxisgerechte Anzahl freier Parameter adaptieren.

**Zusammenfassung**

Ein Filterungsproblem kann als Systemidentifikationsproblem aufgefasst werden. Nichtlineare Abbildungen lassen sich bei der Systemidentifikation im allgemeinen nur approximie-

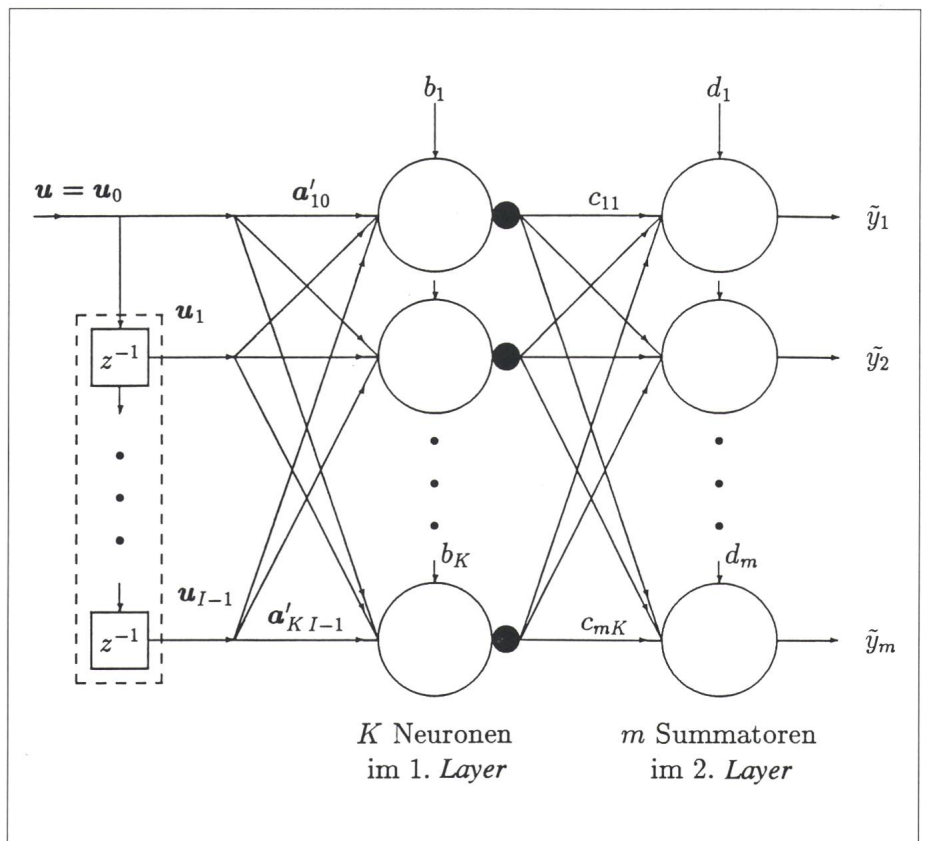
ren. Der Approximationsansatz mit Hilfe der abgebrochenen Volterrareihe weist die günstige Eigenschaft auf, dass die als Modell eingesetzte, abgebrochene Reihe linear von den freien Parametern abhängt. Das Problem der lokalen Minima fällt somit weg. Die Volterrareihe konvergiert aber nicht für jede nichtlineare Abbildung, und die Anzahl freier Parameter explodiert mit steigender Approximationsgenauigkeit.

Der Ansatz des Zweischichten-Perceptrons für die Approximation von

nichtlinearen Systemen, die vergangene Eingangswerte mit der Zeit vergessen, erweist sich als praxisgerecht: Die Anzahl freier Parameter wächst linear mit der gewünschten Genauigkeit. Als Adaptationsalgorithmus kann jedes bekannte Verfahren angewendet werden, das für nichtlineare Systeme mit beschränkter Gedächtnislänge konvergiert.

**Literaturverzeichnis**

- [1] Schetzen M.: Nonlinear System Modeling Based on the Wiener Theory. Proceedings of the IEEE, vol. 69, no. 12, december 1981.
- [2] Schetzen M.: The Volterra and Wiener Theories of Nonlinear Systems. Wiley, New York, 1980.
- [3] Widrow B., Stearns S.D.: Adaptive Signal Processing. Prentice-Hall, New Jersey, 1985.
- [4] Leber J.-F., Matthews M.B.: Neuronale Netzwerke: eine Übersicht. Bulletin SEV, Nr. 80 (1989)15, 5. August 1989.
- [5] Matthews M.B.: An Adaptive Nonlinear Filter Structure. Proc. IEEE, Int. Symp. Circuits and Systems, 1991.
- [6] Hornik K.: Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks, vol. 4, pp. 251...257, 1991.
- [7] Rumelhart D.E., Hinton G.E., Williams R.J.: Learning Internal Representations by Error Propagation. in Parallel distributed processing, vol. 1, pp. 318...362. MA: MIT Press, Cambridge, 1986.
- [8] Haykin S.: Adaptive Filter Theory. Prentice-Hall, New Jersey, 1986.



**Bild 7** Zweischichten-Perceptron mit den I Eingangsvektoren  $u_0 = u[k]$  bis  $u_{I-1} = u[k - (I - 1)]$  in  $R^n$ , dem Ausgangsvektor  $\tilde{y} = \tilde{y}[k]$  in  $R^m$  und mit den freien Parametern  $a'_{ij}$ ,  $b_i$  bis  $b_K$ ,  $c_{kl}$  und  $d_1$  bis  $d_m$