Kapitel 5

Neuronale Netze als Modelle von Zeitreihen

5.1 Allgemeines

Zeitreihen spielen in Natur und Technik eine außerordentlich große Rolle. Jede Meßreihe eines zeitlichen Vorganges ist eine Zeitreihe. Formal nehmen wir an, daß an den diskreten Zeitpunkten $t = \mu h, \quad \mu = 0, 1, 2, 3, \ldots$ ein Datenvektor $\vec{x}(t)$ (z.B. als Ergebnis einer Messung) gegeben ist. Wenn die Zeitreihe von einem natürlichen oder technischen Prozeß erzeugt wird, dann gibt es im allgemeinen einen gesetzmäßigen Zusammenhang zwischen den Daten zu verschiedenen Zeitpunkten und damit prinzipiell die Möglichkeit der Vorhersage zukünftiger Werte der Zeitreihe aus den bereits bekannten Werten (gilt leider nicht immer, s. Lottozahlen).

Konkret heißt das für die **Einschrittprognose,** einen Zusammenhang zwischen dem Vorhersagewert zur Zeit $t + h$ und einer bestimmten Menge seiner Vorgängerwerte herzustellen, d. h. einen funktionellen Zusammenhang der Form [1]

$$\vec{x}_{\mu+1} = \vec{\Phi}\left(\vec{x}_\mu, \vec{x}_{\mu-1}, \ldots, \vec{x}_{\mu-k}\right) \tag{5.1}$$

zu finden, wobei hier und im folgenden

$$\vec{x}_\mu = \vec{x}\left(\mu h\right)$$

gesetzt ist.

Die unterliegenden Gesetzmäßigkeiten sind aber im allgemeinen nicht oder nur unvollständig bekannt. Gleichzeitig sind die Messungen oder Beobachtungen selten exakt, sondern im allgemeinen durch verschiedene zufällige Störgrößen verfälscht (verrauscht).

[1] Wir betrachten der Einfachheit halber einen stationären Zusammenhang, so daß die rechte Seite explizit nicht von der Zeit abhängt.

Das Zusammenspiel von (unbekanntem) Gesetz und Zufall bildet die Herausforderung an die Vorhersage einer Zeitreihe. Neuronale Netze mit ihrer Fähigkeit, aus Beispielen zu lernen und trotz Störeinflüssen die den Daten zugrundeliegende Gesetzmäßigkeit "erahnen" zu können, haben sich deshalb hier zu ernsthaften Konkurrenten der oft aufwendigen klassischen Verfahren entwickelt. Insbesondere liefern die neuronalen Netze oft nichtlineare Verallgemeinerungen bekannter klassischer Verfahren.

5.1.1 Beispiele für Zeitreihen:

Deterministische Zeitreihen aus mathematischen Gesetzmäßigkeiten:

Jede beliebige Funktion der Zeit $\vec{x}(t)$, genommen an den diskreten Zeitpunkten $t = \mu h, \quad \mu = 0, 1, 2, \ldots$,kann als eine Zeitreihe betrachtet werden. Zum Beispiel bilden die $x_\mu, \quad x_\mu \in \Re^1$ aus

$$x_\mu = \alpha \sin(\omega \mu h - \beta) \tag{5.2}$$

eine Zeitreihe.

Verrauschung: Um den Einfluß von zufälligen Störungen bei der Messung zu studieren, führen wir eine Verrauschung der deterministischen Zeitreihe durch, d. h. wir ersetzen

$$x_\mu \to x_\mu + \gamma z$$

wobei z eine standardverteilte Zufallszahl $z \in [-1, 1]$ ist.

Chaotische Zeitreihen:

Viele Prozesse in der Natur sind durch das Phänomen des deterministischen Chaos geprägt. Diese Prozesse produzieren sog. chaotische Zeitreihen.Mathematische Modelle chaotischer Zeitreihen können aus gewissen iterierten Abbildungen, wie z. B. der *logistischen Abbildung,* gewonnen werden. Die Zeitreihe ergibt sich dabei aus der wiederholten Iteration der Abbildung vergangener Werte des Datenvektors $\vec{x}$ auf den aktuellen Zeitpunkt. Verschiedene Zeitreihen ergeben sich aus der Iteration verschiedener Abbildungen. Beispiele:

Logistische Abbildung: Ist die Abbildung die logistische, dann lautet die Bildungsvorschrift für die Zeitreihe

$$x_{\mu+1} = a\, x_\mu\, (1 - x_\mu)$$

mit dem Kontrollparameter a, $0 \leq a \leq 4$ und $x_0 \in [-1, 1]$ womit auch $x_\mu \in [-1, 1] \quad \forall \mu$. Chaos tritt auf, sobald a seinen kritischen Wert $a_c \approx 3.8$ überschreitet.

Mackey-Glass Abbildung: Eine weitere chaotische Zeitreihe mit einer Verzögerungszeit $\tau > 1$ ist die sog. Mackey-Glass Reihe, die sich aus der folgenden, die Leukozytenbildung modellierenden Delay–Differentialgleichung ergibt,

$$\dot{x}(t) = \frac{ax\,(t-\tau)}{1 - x\,(t-\tau)^c} - bx\,(t) \tag{5.3}$$

mit der Verzögerungszeit τ und den frei wählbaren Parametern a, b, c. Ein "gängiger" Parametersatz ist $a = 0,2$; $b = 0,1$; $c = 10$ und $\tau = 17$ oder $\tau = 30$.

Henon-Abbildung: Die Henon-Abbildung ist ein Spezialfall einer allgemeineren Abbildung, die zur Beschreibung eines periodisch angestoßenen Rotators dient. Geschrieben in zwei Variablen lautet die Abbildung

$$x_{\mu+1} = 1 - ax_\mu^2 + y_\mu$$

$$y_{\mu+1} = bx_\mu,$$

was offensichtlich äquivalent ist zu

$$x_{\mu+1} = 1 - ax_\mu^2 + b\,x_{\mu-1}.$$

"Gängige" Parameter: $a = 1,4$ und $b = 0,3$.

Natürliche Zeitreihen:

Im Rahmen des Praktikums "Angewandte Neuroinformatik" kann auch auf eine Reihe von "Real-World" Daten zurückgegriffen werde, die von Prozessen in Natur, der Ökonomie oder der Technik generiert wurden. Im Unterschied zu obigen sind nun die Bildungsgesetze der Zeitreihen weitgehend unbekannt. Hier können die spezifischen Eigenschaften neuronaler Netze zur Erfassung und Nachbildung verborgener Gesetzmäßigkeiten besonders zur Wirkung gelangen. Es stehen folgende Zeitreihen zur Verfügung:

Sonnenfleckentätigkeit: Zahl und Größe der Sonnenflecken (Gebiete auf der Sonnenoberfläche, in denen die Temperatur um ca. 1000 Grad reduziert ist) sind starken Schwankungen unterworfen, wobei wir eine 11 bzw. 12 jährige und eine ca. 100-jährige Periodizität beobachten. Die für die Bildung der Sonnenflecken verantwortlichen Prozesse sind nur unvollkommen verstanden, so daß eine sichere Prognose nicht möglich ist.

Die mittlere jährliche Sonnenfleckentätigkeit wird seit 1700 quantitativ registriert. Diese Daten liegen in einem File vor.

Volkswirtschaftliche Daten: Besonders wichtig und interessant sind die monatlich ermittelten Daten für eine Reihe ökonomischer Kenngrößen, die die wirtschaftliche Entwicklung eines Landes widerspiegeln. Für das Praktikum stehen uns hier, beginnend mit Januar 1953, die Werte für den Diskont- und Lombardsatz und ab Januar 1980 für eine Reihe weiterer Größen zur Verfügung. Diese beinhalten die Preisentwicklung (Erzeuger-, Verbraucher- und Einfuhrpreise), die Auftragslage der Wirtschaft, Kreditvergabe, Staatsverschuldung, Kapitalbildung u. a., die ein ideales Testfeld für die prognostische Leistungsfähigkeit neuronaler Netze darstellen. Die Daten liegen im EXCEL-Format vor (Beschreibung des Dateiformats vorhanden) und sind für die Anwendung geeignet zu konvertieren.

Medizinische Daten Umfangreiche EEG–Daten liegen aus Untersuchungen vor, die am Max–Planck–Institut für Neuropsychologie Leipzig gewonnen wurden. Diese Daten wurden schon einer Waveletanalyse unterworfen, s. [3].

5.2 Prognose von Zeitreihen mit neuronalen Netzen

5.2.1 Ein *ad hoc* Ansatz

Ist das Bildungsgesetz der Zeitreihe bekannt, so ist eine Prognose der zukünftigen Werte von $\vec{x}\,(t)$ aus einer bestimmten Mindestzahl k von Vorgängerwerten zum aktuellen t exakt möglich. Zum Beispiel gilt $k = 2$ für die o. g. Zeitreihe aus der sin-Funktion (Beweis?) bzw. der Henon-Abbildung und $k = 1$ für die logistische Abbildung.

Hier erscheint es ziemlich trivial, ein neuronales Netz zur Prognose dieser Zeitreihen zu konstruieren. Bei geeigneter Konstruktion ist hier schon ein einzelnes (Ausgabe-) Neuron in der Lage, die Zeitreihe exakt vorauszusagen. Betrachten wir als Beispiel eine harmonische Schwingung der Frequenz ω an den Zeitpunkten $t = 0, h, 2h, 3h, \ldots$, mit h beliebig. Mit $t = \mu h, \quad x_\mu = x\,(t)$ gelte

$$x_\mu = \alpha \sin\left(\omega \mu h + \beta\right) \tag{5.4}$$

Aus den Additionstheoremen folgt

$$x_{\mu+1} = c\, x_\mu + d\, x_{\mu-1}, \tag{5.5}$$

wobei

$$c = 2\, \cos\left(\omega h\right), \quad d = -1.$$

Aufgrund von 5.5 kann ein einzelnes **lineares** Neuron (d. h. die Ausgabefunktion $S\,(z) = z$) mit zwei Inputeinheiten, d. h. mit der Transferfunktion

$$y = F\left(\vec{x}, \{\vec{w}\}\right) = \sum_{i=1,2} w_i \, X_i \tag{5.6}$$

schon die Zeitreihe der x_μ exakt voraussagen. In der Tat, mit der Wahl $w_1 = c$ und $w_2 = d$ für die synaptischen Gewichte gibt das Neuron gerade $x_{\mu+1}$ aus, wenn inputseitig die Vorgängerwerte anliegen, d. h. $X_1 = x_\mu$ und $X_2 = x_{\mu-1}$. Das Neuron führt damit eine Einschrittprognose aus. Durch Iteration kann das Neuron dann auch die gegebene Zeitreihe 5.4 für alle Zeiten exakt voraussagen.

Die Einstellung der synaptischen Gewichte w_i kann durch einen Lernalgorithmus anhand von Trainingsbeispielen aus der Zeitreihe erfolgen. Diese bestehen aus Input-Output Paaren der Form

$$y_{\mu\ soll} = x_{\mu+1}, \quad X_{\mu\,1} = x_\mu, \; X_{\mu\,2} = x_{\mu-1}$$

mit zufällig gewähltem μ. Als Lernalgorithmus eignet sich der Backpropagation–Algorithmus (s. Kap. 3).

Das gewählte Beispiel zeichnet sich durch den linearen Zusammenhang zwischen den Werten der Zeitreihe zu verschiedenen Zeitpunkten aus. Eine Übertragung des Zuganges etwa auf die Henon-Abbildung erfordert wegen des quadratischen Termes die Einbeziehung von Synapsen zweiter Ordnung, d. h. einen Ansatz für die Transferfunktion der Art

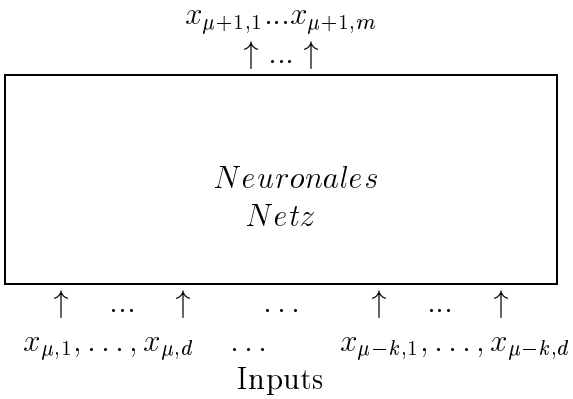
$$f\left(X\right) = \sum w_i X_i + \sum_{i,j} w^{(2)}_{i,j} X_i X_j. \tag{5.7}$$

5.2.2 Feed-Forward-Netze als universelle Modelle von Zeitreihen

Bei unbekannten oder komplexeren Bildungsgesetzen ist dieser direkte Zugang weniger geeignet (Mehrdeutigkeiten, Instabilitäten, lange Trainingszeiten, keine Standardhardware). Universell einsetzbare Lösungsansätze nutzen aus, daß neuronale Netze als universelle Funktionsapproximatoren dienen können, d. h. daß ausreichend komplex strukturierte neuronale Netze in der Lage sind, aus einer hinreichenden Zahl von Beispielen jede beliebige Abbildung der Form 5.1 zu inferieren, s. Kap. 3.

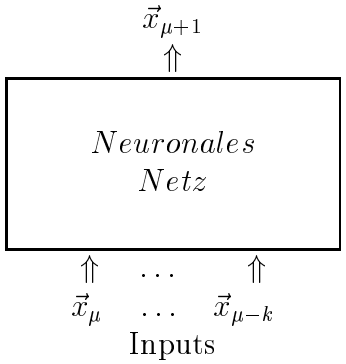
Dieser gedankliche Ansatz kann in folgender Weise formalisiert werden: Für die Zeitreihe $\vec{x}_\mu = \left(x_{\mu,1}, \ldots, x_{\mu,d}\right), \;\; \vec{x} \in \Re^d, \;\; \mu = 0, 1, 2, \ldots$ konstruieren wir ein

Feed-Forward-Netz mit $D = k \cdot d$ Input- und d Output-Einheiten.



(5.8)

oder kompakter



(5.9)

Das neuronale Netz ist durch seine Transferfunktion

$$\vec{y} = F\left(\vec{X}, \{w\}\right) \tag{5.10}$$

gegeben, wobei $\vec{X}$ =Inputvektor und $\{w\}$ = Menge der synaptischen Gewichte.Als Inputvektor $\vec{X}$ dient der $D = kd$-dimensionale Vektor, der aus k Vorgängerwerten des vorherzusagenden $\vec{x}_{\mu+1}$ gebildet wird (*time-delay* Koordinaten)[2]:

$$\vec{X}_\mu = (\vec{x}_\mu, \vec{x}_{\mu-1}, \ldots, \vec{x}_{\mu-k}) = (x_{\mu,1}, \ldots, x_{\mu-k,d}) \tag{5.11}$$

Als Trainingsbeispiele wählen wir für hinreichend viele μ Werte die Input-Output-Paare $\mathcal{Z}_\mu$

$$\mathcal{Z}_\mu = \left[\vec{y}_{\mu\, soll}, \vec{X}_\mu\right] = \left[\vec{x}_{\mu+1}, \vec{X}_\mu\right] \tag{5.12}$$

Der Output y des neuronalen Netzes 5.10 liefert den Schätzwert

$$\boxed{\tilde{x}_{\mu+1} = F\left(\vec{X}_\mu, \{w\}\right)} \tag{5.13}$$

[2]Im allgemeinen können die Vorgängerwerte auch zu beliebigen Zeitpunkten in der Vergangenheit gewählt werden.

Aufgabe des Backprop-Lernalgorithmus ist es dann, die Gewichte so einzustellen, daß

$$\vec{y}_\mu \Rightarrow \vec{y}_{\mu\ soll} \quad \text{, d. h. } \tilde{x}_{\mu+1} \Rightarrow x_{\mu+1} \quad \forall\mu$$

so daß im ideal trainierten Netz gilt

$$\vec{\Phi}\left(\vec{X}_\mu\right) = F\left(\vec{X}_\mu, \{w\}\right) \qquad \forall\mu$$

Zu beachten ist dabei, daß in den Fällen mit bekanntem Bildungsgesetz der Zeitreihe die Zahl k der benötigten Vorgängerwerte für eine sichere Vorhersage der Zeitreihe bekannt ist. Die Bestimmung dieser Zahl bildet jedoch bei den natürlichen Zeitreihen eine oft außerordentlich schwierige Aufgabe, deren effektive Lösung die Leistungen des Netzes in entscheidender Weise beeinflußt. Relevante Punkte sind dabei:

1. Der Einfluß des Rauschens läßt sich durch Mitnahme einer, im Vergleich zum deterministischen Bildungsgesetz, größeren Zahl von Vorgängerzeitpunkten im Inputvektor eliminieren.

2. Einfluß auf die Prognoseleistungen des Netzwerkes hat nicht nur die Zahl k der Vorgängerzeitpunkte, sondern auch die gewählte Schrittweite und damit die Länge des Intervalls in dem diese liegen (Zeithorizont). Eine elegante Lösung zur Rauschreduktion und flexiblen Schrittweitenwahl besteht in der Glättung der Inputs vor der Einspeisung in das neuronale Netz. Dazu sind gleitende Mittelungen besonders geeignet, d. h. wir bestimmen in jedem Zeitpunkt μ einen Mittelwert $\bar{x}_\mu$ über ein Zeitintervall in der unmittelbaren Vergangenheit, wobei $\bar{x}_\mu$ gemäß

   $$\bar{x}_\mu = \bar{x}_{\mu-1} - \epsilon\left(x_{\mu-1} - \bar{x}_{\mu-1}\right)$$

   in jedem Schritt aktualisiert wird und ϵ^{-1} den gewählten Zeithorizont in der Vergangenheit angibt (Länge des Mittelungsintervalls in Einheiten von h). Werden mehrere Vorgänger benötigt, so sind diese durch Mittelwerte $\bar{x}^\epsilon_\mu, \bar{x}^\eta_\mu, ...$ zu verschiedenen Zeithorizonten $\epsilon, \eta, ...$ gegeben, wobei $\vec{X}_\mu = \left(\bar{x}^\epsilon_\mu, \bar{x}^\eta_\mu, ...\right)$.

Bei der freien Iteration des Netzwerkes wird ein Startvektor $\vec{X}_\mu$ aus bekannten Werten der Zeitreihe als Input in das Netzwerk eingefüttert und der entsprechende Output $\vec{y}_\mu$ als neuer Wert $\vec{x}_{\mu+1}$ des Datenvektors interpretiert. Damit wird ein neuer Inputvektor generiert, indem durch eine Zeitverschiebung um eine Einheit alle Datenvektoren im Inputvektor $\vec{X}_{\mu+1}$ um einen Platz nach rechts rücken (Schieberegister). Dabei fällt der älteste Wert heraus und der jüngste Wert im Inputvektor ist jetzt $\vec{x}_{\mu+1}$, d. h.

$$\vec{X}_\mu = (\vec{x}_\mu, \vec{x}_{\mu-1}, \ldots, \vec{x}_{\mu-k}) \Rightarrow \vec{X}_{\mu+1} = (\tilde{x}_{\mu+1}, \vec{x}_\mu, \vec{x}_{\mu-1}, \ldots, \vec{x}_{\mu-k+1})$$

wobei $\vec{y}_\mu = F\left(\vec{X}_\mu, \{w\}\right)$. Diese Prozedur wird iteriert, so daß nach $k+1$ Schritten alle Komponenten des Inputvektors nur noch aus den vom Netzwerk selbst erzeugten Daten bestehen und damit kein Rückgriff auf die beobachteten Daten der Zeitreihe mehr erfolgt (*freie Iteration).* Wenn das neuronale Netz die Zeitreihe exakt repräsentiert, kann damit die Zeitreihe für alle Zeiten aus den k Startwerten vorausgesagt werden.

Interessante Anwendungen zur Systemidentifikation und Zeitreihenvorhersage finden sich in [7, 1, 6] u. a.

5.2.3 Qualitätskriterien

Im allgemeinen wird schon die Einschrittprognose mit Fehlern behaftet sein. Als quantitatives Maß dieses Fehlers kann die mittlere quadratische Abweichung

$$E = \frac{1}{P} \sum_{\mu \in \mathcal{M}} \|\vec{y}_{\mu\,\mathrm{soll}} - \vec{y}_\mu\|^2 = \frac{1}{P} \sum_{\mu \in \mathcal{M}} \|\vec{x}_{\mu+1} - \vec{y}_\mu\|^2 \tag{5.14}$$

für P Prognoseschritte dienen, wobei $\mathcal{M}$ die Menge der μ-Werte ist, für die das Netz angelernt wurde. Allerdings tritt auch hier wieder die Gefahr des Overfitting auf, so daß weitere Maße einbezogen werden müssen, die insbesondere die Generalisierungsfähigkeit des Netzwerkes beurteilen, s. dazu das Kap. 3 zu Feed-Forward-Netzwerken (Kreuzvalidierung, Bootstrap-Verfahren).

Für die Mehrschrittprognose sind, insbesondere bei chaotischen Zeitreihen, diese Maße nicht besonders aussagekräftig. Geeigneter ist hier eine Darstellung der Trajektorien im *Phasenraum* der Zeitreihe im Vergleich mit ihrer aus der freien Iteration gewonnenen Approximation.

5.3 Rekurrente Netze

Ein wesentlicher Nachteil der Modellierung von Zeitreihen mit feed–forward Netzen ist die Tatsache, daß die Zahl der Vorgängerzeitpunkte die Architektur (Anzahl der Inputunits) beeinflußt. Das macht die probeweise Veränderung des Zeithorizontes schwierig. Eine sehr interessante und effektive Alternative bieten hierzu die rekurrenten Netze, die ja über innere Zustände verfügen und damit in der Lage sind, intern eine Erinnerung über die vergangenen Werte der Zeitreihe aufzubauen. Diese Netze brauchen folglich keine time–delay Koordinaten als Inputs und sind deshalb für Zeitreihen mit unbekanntem Bildungsgesetz besondes gut geeignet.

Bekannte Varianten sind die Netze von Elman und Jordan, vgl. [4, 5, 7], die noch mit dem Backpropagation–Algorithmus trainierbar sind. Für das Training allgemeinerer rekurrenter Netze stehen verschiedene Verallgemeinerungen des Backpropagation zur Verfügung, wie z. B. rekurrentes Backpropagation oder *backpropagation through time*, s. [6].

5.4 Zeitreihenprognose mit selbstorganisierenden Merkmalskarten (Kohonen-Netze)

Ein alternatives Verfahren beruht auf der Verwendung der auf Kohonen zurückgehenden selbstorganisierenden Merkmalskarten [2].

5.4.1 Das Funktionsprinzip

Wie im Abschnitt "Zeitreihen" beschrieben, führen wir die sog. *time-delay* Koordinaten ein, d. h. wir definieren einen $D-$dimensionalen Vektor $\vec{X}_\mu$

$$\vec{X}_\mu = (\vec{x}_\mu, \vec{x}_{\mu-1}, \ldots, \vec{x}_{\mu-k})$$

der mit dem Vorhersagewert $\vec{x}_{\mu+1}$ zu einem $D+d-$dimensionalen Vektor

$$\vec{\mathcal{Z}}_\mu = \left(\vec{x}_{\mu+1}, \vec{X}_\mu\right) \tag{5.15}$$

zusammengeschlossen wird. Jeder Inputvektor $\vec{\mathcal{Z}}_\mu$ repräsentiert jeweils ein Trajektorienstück über $k+2$ Zeitschritte, so daß jedem Trajektorienstück damit ein Punkt im $D+d$ -dimensionalen Phasenraum entspricht. Im Unterschied zu den Prognoseverfahren mit anderen neuronalen Netzen (z. B. backpropagation) wird beim Modellieren der Zeitreihe mittels Kohonennenetzen der Vektor $\vec{\mathcal{Z}}_\mu$ nicht als Input-Output-Paar interpretiert sondern in der Lernphase insgesamt als **Input** in das Kohonennetz eingegeben.

5.4.2 Der Kohonenalgorithmus zur Zeitreihenvorhersage

Wir betrachten ein neuronales Netz, bei dem die N Neuronen im sog. Neuronenraum an Orten $\vec{r}_i,\ i = 1, \ldots, N$ sitzen. Üblich ist die Anordnung der Neuronen in einem Gitter der Dimension d_K. Jedes Neuron ist über seine synaptischen Gewichte mit den Inputunits verbunden. Wir schreiben den Vektor der synaptischen Gewichte des Neurons i hier als

$$\vec{\mathcal{W}}_i = \left(\vec{w}_i^{(1)}, \vec{w}_i^{(0)}, \ldots, \vec{w}_i^{(-k)}\right) = \left(w_{i,1}^{(1)}, \ldots, w_{i,d}^{(-k)}\right)$$

wobei $\vec{w} \in \Re^d$. Der Kohonen-Algorithmus besteht aus den Schritten

- **Bestimmung des Siegerneurons**: Bei Präsentation des Inputs $\vec{\mathcal{Z}}_\mu$ wird das Neuron Sieger, dessen synaptischer Verktor dem Neuron am nächsten liegt = best match neuron (im Sinne des Euklidischen Abstandes), d. h. Neuron s ist Sieger, falls gilt

$$\left(\vec{\mathcal{Z}}_\mu - \vec{\mathcal{W}}_s\right)^2 \leq \left(\vec{\mathcal{Z}}_\mu - \vec{\mathcal{W}}_i\right)^2 \quad \forall i \neq s \tag{5.16}$$

- **Lernschritt:** Im anschließenden Lernschritt werden die synaptischen Gewichte der Neuronen geändert,

$$\Delta\vec{\mathcal{W}}_i = \epsilon h_{i,s} \cdot \left(\vec{\mathcal{Z}}_\mu - \vec{\mathcal{W}}_i\right) \tag{5.17}$$

  mit der Nachbarschaftsfunktion

$$h_{i,s} = e^{-\frac{(\vec{r}_s - \vec{r}_i)^2}{2\sigma^2}} \tag{5.18}$$

$h_{i,s}$ mißt die Stärke mit der das Neuron i mitlernen darf, wenn Neuron s Sieger geworden ist. Diese Stärke klingt also mit dem Abstand zwischen den Neuronen mit der Halbwertsbreite σ ab.

Das Netz in der Lernphase:

Das Netz wird mit einer hinreichend großen Anzahl von Inputvektoren $\vec{\mathcal{Z}}_\mu$ (für verschiedene μ) trainiert. Hierbei wird also kein Unterschied zwischen bekannten Daten ($\vec{X}_\mu$) und den Vorhersagewerten ($\vec{x}_{\mu+1}$) gemacht. In der Trainingsphase lernen die Neuronen des Netzes ganz bestimmte Trajektorienstücke zu repräsentieren, da ja genau das Neuron s Sieger wird, dessen synaptischer Vektor $\vec{\mathcal{W}}_s$ am besten mit dem Trajektorienstück $\vec{\mathcal{Z}}_\mu$ übereinstimmt, s. Gleichung 5.16.

Das Netz in der Kannphase:

Nachdem das Netz angelernt ist, kann es zur Vorhersage der Zeitreihe verwendet werden. Das Prinzip zur Reproduktion der Zeitreihe besteht darin, passende Trajektorienstücke aneinanderzufügen. Bei der Einschrittprognose - Vorhersage von $\vec{x}_{\mu+1}$ aus $\vec{X}_\mu$ - wird dabei folgendermaßen vorgegangen:

  - **Partielles Matching zur Siegerbestimmung:** Sei $\vec{X}_\mu$ zur Zeit t_μ bekannt. Suche das Neuron, dessen synaptischer Vektor

$$\vec{\mathcal{W}}_i = \left(\vec{w}_i^{(1)}, \vec{w}_i^{(0)}, \ldots, \vec{w}_i^{(-k)}\right) = \left(w_{i,1}^{(1)}, \ldots, w_{i,d}^{(-k)}\right)$$

    bezüglich der Komponenten $\vec{w}_i^{(0)}, \ldots, \vec{w}_i^{(-k)}$ am besten mit den entsprechenden Komponenten des Inputvektors, d. h. mit den Komponenten $\vec{x}_\mu, \vec{x}_{\mu-1}, \ldots, \vec{x}_{\mu-k}$ von $\vec{X}_\mu$ übereinstimmt, d. h. Neuron s ist Sieger, wenn

$$\sum_{j=0}^{k} \left(\vec{w}_s^{(-j)} - \vec{x}_{\mu+j}\right)^2 \leq \sum_{j=0}^{k} \left(\vec{w}_i^{(-j)} - \vec{x}_{\mu+j}\right)^2 \quad \forall i \neq s$$

- **Der Vektor $\vec{w}_s^{(1)}$ ist der Vorhersagewert für $\vec{x}_{\mu+1}$.**

Bemerkung: In den Kohonenalgorithmus geht in Form der Nachbarschaftsfunktion wesentlich die Topologie des Neuronenraumes ein, i. a. werden die Neuronen auf ein Gitter der Dimension d_K gesetzt. Die Topologie des Neuronenraumes muß dem Problem in gewisser Weise angepaßt werden. Dazu überlege man sich, daß die Leistung des Kohonenalgorithmus in der Herstellung einer **nachbarschaftstreuen** Abbildung des Inputraumes auf den Neuronenraum besteht. Dabei wird nicht der gesamte Inputraum, sondern im wesentlichen nur das Gebiet, in dem die Wolke der Datenpunkte liegt, abgebildet.

Die Inputvektoren $\vec{\mathcal{Z}}_\mu$ repräsentieren jeweils ein Trajektorienstück über $k+2$ Zeitschritte. Jedem solchen $\vec{\mathcal{Z}}_\mu$, d. h. jedem Trajektorienstück, entspricht damit ein Punkt im $D+d$ dimensionalen Phasenraum. Wenn die Zeitreihe etwa durch eine harmonische Funktion erzeugt wird,

$$x_\mu = \alpha \, \sin\left(\omega \mu h - \beta\right)$$

dann liegen diese Punkte - unabhängig vom Wert von k und damit von der Dimension des Inputraumes - auf einer **eindimensionalen** geschlossenen Mannigfaltigkeit. Dementsprechend ist es hier sinnvoll, die Neuronen im Neuronenraum auf einem Kreis anzuordnen.

Analoge Überlegungen sind für alle betrachteten Zeitreihen anzustellen.

5.5 Übungen

Aufgabe 1: Zeitreihenvorhersage mit Feed-Forward-Netzwerken

Untersuchen Sie mit dem SNNS (Stuttgarter Neuronaler Netzwerksimulator) [8] die Leistungsfähigkeit von Feed-Forward-Netzwerken für die Zeitreihenvorhersage sowohl in der Einschritt- als auch in der Mehrschrittprognose. Aufgabenstellungen sind:

- Realisieren Sie mit geeigneten Feed-Forward-Netzen ein Progosesystem sowohl für Zeitreihen mit bekannten Bildungsgesetzen (sin-Funktion, chaotische Zeitreihen) als auch für Beobachtungsdaten (Sonnenflecken, ökonomische Zeitreihen).
- Untersuchen Sie den Einfluß der richtigen Wahl der Vorgängerzeitpunkte (s. o.), insbesondere zur Rauschreduktion und den
- Einfluß der Architektur des Netzes auf Prognosegenauigkeit und Verallgemeinerungsfähigkeit des Netzwerkes.
- Varianten des Backpropagation-Algorithmus und Lerngeschwindigkeit.
- Stellen Sie die Ergebnisse geeignet dar, z. B. durch eine Lernkurve und/oder den direkten Vergleich der Zeitreihen bzw. der entsprechenden Phasenraumdarstellungen.

Aufgabe 2: Vorhersage von Zeitreihen mit Kohonennetzwerken

Alternativ (oder zusätzlich) zu **Aufgabe 1** (Vorhersage mit Feed-Forward-Netzwerken) sollen sowohl für eine künstliche (sin-Funktion) und eine natürliche Zeitreihe (Sonnenfleckentätigkeit) mit der dargestellten Methode die Zeitreihenvorhersage durchgeführt werden. Dabei sind folgende Aufgabenstellungen zu beachten:

- Untersuchung der Prognosegenauigkeit und Generalisierungsfähigkeit als Funktion der Zahl der Neuronen und der Reichweite der Nachbarschaftsfunktion.
- Stellen Sie die Ergebnisse geeignet dar (Lernkurve, direkter Vergleich der vorhergesagten mit der vorgegebenen Zeitreihe). Diskutieren Sie die Ergebnisse, speziell im Vergleich mit Feed-Forward-Netzwerken und zum unter 5.2.1 beschriebenen *ad hoc* Ansatz (z. B. für die sin-Funktion)

Literaturverzeichnis

[1] A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing.* WILEY, 1993.

[2] R. Der, H. Englisch, M. Funke, and M. Herrmann. Time series prediction using hierarchical self-organized feature maps. *Neural Networks World*, 3(6):699–704, 1993.

[3] R. Der and U. Steinmetz. Wavelet analysis of eeg signals as a tool for the investigation of the time architecture of cognitive processes. Technical Report 4/97, Institut für Informatik, Universität Leipzig, 1997. Internet address: http://www.informatik.uni-leipzig.de/ der/Veroeff/eegwave.ps.gz.

[4] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179 – 211, 1990.

[5] J. L. Elman. Language as a dynamical system. In R. F. Port and T. V. Gelder, editors, *Mind as Motion: Explorations in the Dynamics of Cognition*, chapter 8, pages 195 – 226. MIT, 1995.

[6] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the theory of neural computation.* Addison Wesley, 1991.

[7] A. Zell. *Simulation neuronaler Netze.* Addison–Wesley, 1994.

[8] A. Zell. SNNS - Der Stuttgarter Neuronaler Netz Simulator. http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html, 1996.