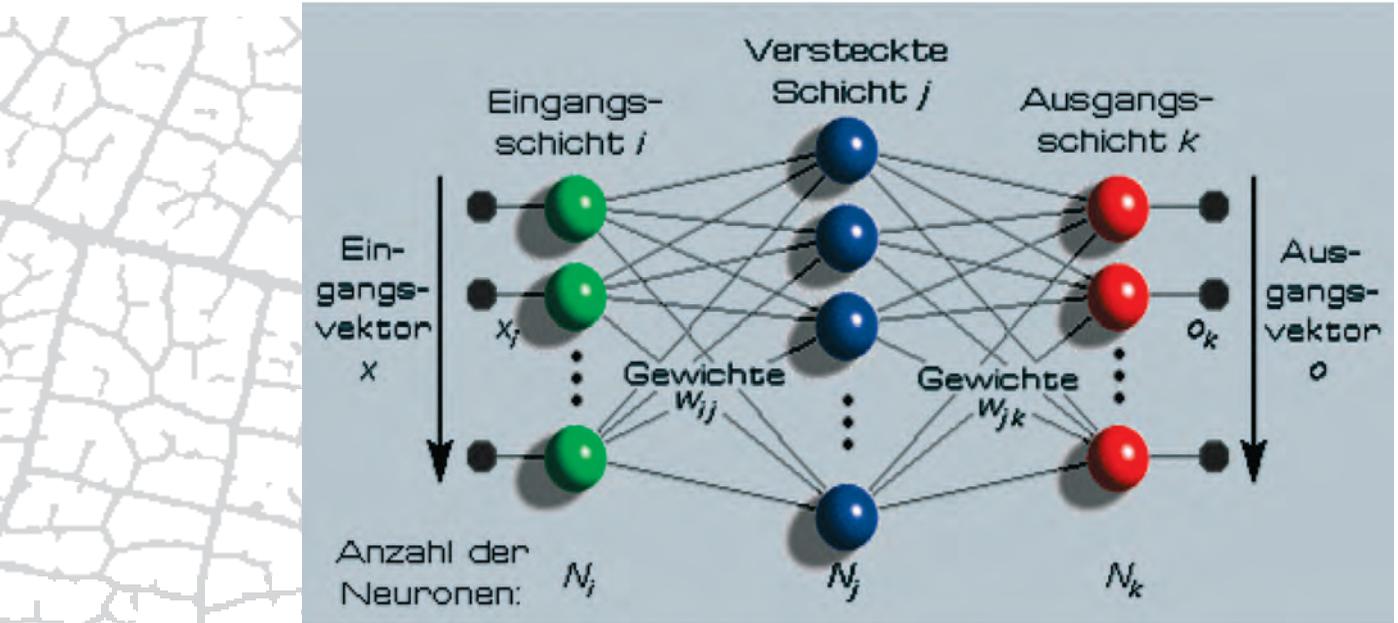


Marta Neunteufel
Josef Binder

Einsatz künstlicher neuronaler Netze zur Extra- und Interpolation von Zeitreihen







Marta Neunteufel
Josef Binder

Einsatz künstlicher neuronaler Netze zur Extra- und Interpolation von Zeitreihen

Wien, 2007

Inhalt

Vorwort	7
1 Zeitreihen, ihre Extra- und Interpolation mit künstlichen neuronalen Netzen.	9
1.1 Zeitreihenanalyse und Prognosen	9
1.2 Was sind künstliche neuronale Netze?	10
1.3 Elemente und Strukturen von KNN	13
1.3.1 Die Elemente von KNN	13
1.3.2 Die Strukturen von KNN	16
1.3.2.1 Perzeptronen	16
1.3.2.2 Multi-Layer Perzeptronen	17
1.3.2.3 Rekurrente Netze	18
1.3.2.4 Time-delay Netze	19
1.4 Zusammenfassung	20
2 Prognose der Schweineproduktion mit künstlichen neuronalen Netzen	21
2.1 Variante 1	22
2.2 Variante 2	22
2.3 Variante 3	23
2.4 Variante 4	23
2.5 Zusammenfassung	24
Literaturverzeichnis	25
Fußnoten	26

Vorwort

In der Forschungstätigkeit der Bundesanstalt für Agrarwirtschaft nehmen die Methoden der Zeitreihenanalyse einen wichtigen Platz ein. Da die Wahl einer geeigneten Methode von der konkreten Fragestellung (Prognose, Interpolation usw.) abhängig ist, ist es immer notwendig, die Wahlmöglichkeiten zu erweitern. Daher werden neue Methoden in der Bundesanstalt für Agrarwirtschaft auf deren Anwendbarkeit für diese konkreten Fragestellungen getestet und die Ergebnisse und Erfahrungen für die Mitarbeiter zu Verfügung gestellt.

Die Methode der künstlichen neuronalen Netze (KNN) wird immer häufiger zur Extrapolation und Interpolation von Zeitreihen eingesetzt. Ein wesentlicher Vorteil der KNN ist es, dass sie ermöglichen, auch Zusammenhänge in den (bzw. zwischen den) Zeitreihen aufzuzeigen und für die konkrete Anwendung zu benutzen, die sich einer expliziten Modellierung entziehen. Dadurch wird der Anwendungsbereich der Methode erweitert.

Der vorliegende Agrarpolitische Arbeitsbehelf erschließt die Methode der KNN zur Extrapolation und Interpolation von Zeitreihen für künftige Anwendungen in der Bundesanstalt für Agrarwirtschaft. Die Methode der KNN wurde kurz beschrieben, anhand einiger Beispiele dargestellt und die Testergebnisse hinsichtlich ihrer speziellen Anwendbarkeit analysiert.

Für die Lektorierung des Textes danken wir Herrn Hubert Schlieber.

Hubert Pfingstner
Direktor

1 Zeitreihen, ihre Extra- und Interpolation mit künstlichen neuronalen Netzen.

1.1 Zeitreihenanalyse und Prognosen

Die meisten Beobachtungen von demographischen, ökonomischen, physikalischen usw. Prozessen werden durch numerische Größen festgehalten, die als Ergebnis von Messungen entstehen. Aus den Messergebnissen mehrerer Beobachtungen können über die zeitliche Entwicklung des beobachteten Prozesses wertvolle Informationen gewonnen werden.¹

Diesen Prozessen wird ein dynamischer Vorgang mit Zufallscharakter unterstellt.

Die zeitlich geordnete unendliche Folge von den Zufallsvariablen

$$\{X\} = X_1, X_2, X_3, \dots, X_t, \dots$$

wird als stochastischer Prozess bezeichnet.

Die beobachteten Werte des stochastischen Prozesses in den Zeitpunkten $t=1,2,\dots,T$ x_1, x_2, \dots, x_T bilden eine Zeitreihe und werden als eine Reihe von möglichen (zufälligen) Realisierungen – als ein Sample – betrachtet. Die Zeitpunkte² können äquidistant sein oder unregelmäßige Abstände haben.

Wegen des Zufallscharakters des Prozesses $\{X\}$ und seiner Beobachtungen x_t bedarf es spezieller Methoden, um Rückschlüsse auf Regelmäßigkeiten und auf die rein zufällige Komponente ziehen zu können. Als regelmäßige Komponenten werden der Trend und die Periodizität, als rein zufällige Komponente das „weiße Rauschen“ betrachtet.

Üblicherweise werden Zeitreihen zur Analyse und zur Extrapolation eines stochastischen Prozesses verwendet. Die Extrapolationen, d.h. die Schätzungen von zukünftigen Werten, dienen zur Prognose der weiteren Entwicklung des stochastischen Prozesses. Eine exakte Prognose ist zwar nicht möglich, durch die Anwendung von verschiedenen Prognosemethoden können jedoch Ergebnisse erreicht werden, deren Fehler tolerierbar sind. Solche Prognosen können in der Praxis (z.B. als Entscheidungsgrundlagen) nützlich sein.

Das Gleiche gilt für die Schätzung von fehlenden Werten innerhalb der beobachteten Zeitspanne, für Interpolationen. Daher verwenden wir im Folgenden nur den Ausdruck der Prognose.

In der Ökonomie werden für Prognosen verschiedene Methoden verwendet:

Ökonometrische Modelle liefern Vorhersagen mithilfe von Regressionsgleichungen, deren Variablen und mathematische Form auf ökonomischen Theorien basieren. Die Parameter der Gleichungen werden aufgrund von statistischen Daten geschätzt.

Eine andere häufig verwendete Methode ist die Zeitreihenanalyse, die die Prognosen für einzelne Variablen aus ihrer Entwicklung in der Vergangenheit ableitet. Die bekanntesten Methoden der Zeitreihenanalyse sind das autoregressiv-moving-average (ARIMA) Modell und die Spektralanalyse. Im letzteren wird die Zeitreihe in den Frequenzbereich transponiert und analysiert.

Der Grund für den Verzicht auf theoretische Information über die ökonomische Struktur – also die Anwendung von Zeitreihenanalyse anstatt eines ökonometrischen Modells – besteht darin, dass ökonomische Theorien immer auf Annahmen basieren, die nicht notwendigerweise zutreffen: Die beobachteten Daten können den theoretischen Annahmen widersprechen und dadurch kann die Parameterschätzung zu unrealistischen Ergebnissen führen.

Künstliche neuronale Netze (KNN) bauen ebenfalls auf den Beobachtungen von stochastischen Prozessen (ähnlich der Zeitreihenanalyse) auf, und werden besonders in den letzten zwanzig Jahren immer häufiger für Prognosen verwendet.

Die Prognosemethoden mit KNN werden immer weiterentwickelt und verfeinert, um die Vorteile dieser Methode gegenüber den oben erwähnten Prognoseverfahren nutzen zu können. Aber auch die Prognosen mithilfe von KNN sind nicht in jeder Situation den anderen Methoden überlegen.

Bei den Anwendungen von KNN – ähnlich zu den ARIMA-Modellen und zur Spektralanalyse – geht es darum, jenes konkrete Verfahren zu finden, das für den gegebenen Fall am besten geeignet ist. Dies kann anhand von verschiedenen Kriterien beurteilt werden. Die Wahl des „besten Verfahrens“ ist nicht notwendigerweise eindeutig, weil die Gewichtung der verschiedenen Kriterien gegenüber einander nicht a priori gegeben ist.

Im Folgenden diskutieren wir, was KNN sind und welche Vor- und Nachteile ihre Anwendung für Prognosezwecke bieten. Es wird auf den Aufbau von KNN näher eingegangen und darauf, welche Strukturen sie haben können. Wir beschreiben, welche Lernverfahren es gibt, um ein Netz zu trainieren und welche Kriterien zur Beurteilung der Ergebnisse verwendet werden können. Schließlich werden wir im zweiten Teil dieser Arbeit das Prognoseverfahren anhand eines Beispiels illustrieren.

1.2 Was sind künstliche neuronale Netze?

Die Entwicklung von künstlichen neuronalen Netzen ist auf die Ergebnisse der Gehirnforschung zurückzuführen, auf die Erkenntnis, dass das menschliche Gehirn, seine Rechenfähigkeit (Wahrnehmungs- und Lernfähigkeit usw.) auf einem völlig anderen Aufbau und einer anderen Funktionsweise basiert als die der üblichen digitalen Computer. Vergleichsweise entspricht das Gehirn einem hoch komplexen, nicht-linearen Parallelcomputer. Als Informationsverarbeitungssystem entwickelt sich das Gehirn aus dem bei der Geburt gegebenen Gehirnzellen und deren Vernetzung (Verkabelung), die sich besonders schnell in den ersten zwei Lebensjahren durch Wahrnehmen, Probieren und Lernen herausbildet. Dies bedeutet, dass Regeln, an denen sich die Funktionsweise des Gehirns lebenslang anlehnt, auf die in dieser Lebensphase gemachten Erfahrungen zurückzuführen sind. (Man denke beispielsweise daran, wie im Gehirn aus den wahrgenommenen Lichtsignalen Farben oder Bilder von äußeren Gegenständen entstehen, und wie verschiedene andere Wahrnehmungen im Gehirn „Spuren“, d.h. Erinnerungen hinterlassen, usw.)

In der folgenden Abbildung wird das Vorbild von KNN, der Aufbau einer Zelle des menschlichen Gehirns, skizziert.

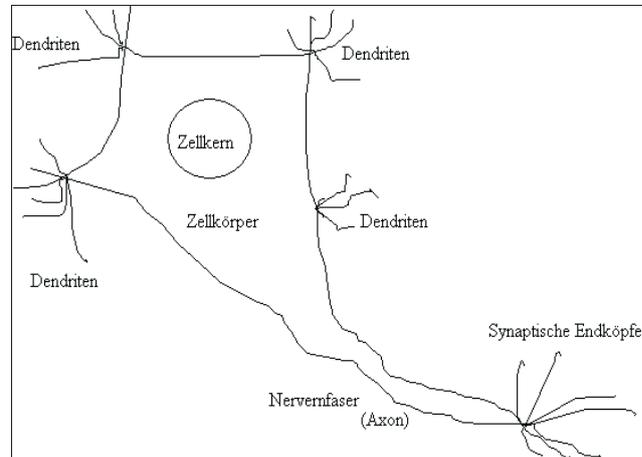


Abbildung 1:
Aufbau einer Zelle des menschlichen Gehirns

Quelle: Sauer 1991-2000

Der *Zellkörper* ist ein Informationsträger. Im einfachsten Fall wird zwischen zwei Zuständen unterschieden: erregt oder nicht erregt.

Das *Axon* dient zur Weitervermittlung des Erregungszustandes.

Die Funktion der *Synapsen* besteht darin, zu bestimmen, wie sich die über ein Axon vermittelte Erregung auf andere Zellen auswirken soll.

Unter einem KNN versteht man daher eine Maschine bzw. ein Computerprogramm, deren Plan und Aufbau sich nach dem „Modell menschliches Gehirn“ richtet. Um eine gute Leistung zu erreichen, verfügen KNN über starke Vernetzungen von einfachen Elementen („Neuronen“, „processing units“).

Definition: KNN sind massiv parallel verbundene Netzwerke aus einfachen Elementen, die die Fähigkeit haben, durch Experimente gewonnenes Wissen zu speichern und für spätere Verwendungen bereitzuhalten. Dem menschlichen Gehirn sind sie in zweifacher Weise ähnlich:

- Das Netz erlangt sein Wissen durch Lernen aus seiner Umgebung.
- Das erlernte Wissen wird als Stärke der zwischenneuronalen Verbindungen (Gewichte der Synapsen) gespeichert.

Der Lernprozess, in dem die Stärken der zwischenneuronalen Verbindungen modifiziert werden, wird durch die Anwendung eines Lernalgorithmus durchgeführt. Mit Lernalgorithmen können unterschiedliche Lernverhalten simuliert werden. Das Ziel eines Lernprozesses ist es, die zwischenneuronalen Verbindungen so zu modifizieren, dass das KNN jene Fähigkeiten erlangt, die für die Lösung der Aufgaben, die dem Netz gestellt werden, erforderlich sind.

KNN werden für die Lösungen von Aufgaben in mehreren Bereichen verwendet: Klassifizierung von Datensätzen, Modellierung und Simulation biologischer neuronaler Netze, Funktionsapproximation, Speicherung von Informationen, Prozesssteuerung, Schrifterkennung, Zeitreihenanalyse und Prognose, Robotsteuerung usw.

Die zwei wichtigsten Vorteile von KNN sind, dass sie durch die massive Vernetzung eine sehr hohe Rechenkapazität erlangen und durch das Training imstande sind, das Erlernete zu verallgemeinern. Diese Fähigkeiten ermöglichen dem Netz, komplexe und rechenintensive Aufgaben zu lösen.

KNN verfügen über mehrere **nützliche Eigenschaften**, von denen hier nur drei erwähnt werden sollen, die für die Erstellung von Prognosen wesentlich sind:

- **Nicht-Linearität:** Ein künstliches Neuron kann linear oder nicht-linear sein, ebenso wie seine Vernetzungen. Diese Eigenschaft ist von besonderer Bedeutung, wenn die zu modellierenden stochastischen Prozesse nicht-linear sind.
- **Abbildung des Zusammenhanges zwischen Input- und Outputsignalen:** Das vielleicht meistverwendete Lernparadigma, das überwachte Lernen (supervised learning) präsentiert dem Netz wiederholt Inputsignale sowie die erwarteten Outputsignale eines Lernbeispiels. Dabei werden die jeweiligen Input- und die dazu gehörigen Outputsignale aus der Menge der Beobachtungsdaten zufällig ausgewählt. Die Stärken der zwischenneuronalen Verbindungen werden so modifiziert, dass die Abweichung zwischen dem von Netz produzierten und dem erwarteten Output entsprechend einem statistischen Kriterium minimiert wird. Dabei verhält sich das Netz wie eine Blackbox, in der keine Hypothesen (a priori Modellannahmen des Netzplaners) über das Verhältnis zwischen Input- und Outputsignalen verwendet werden. Das Training wird abgebrochen, wenn die Stärken der zwischenneuronalen Verbindungen einen stabilen Bereich erreichen, d.h. ihre Änderungen zu keiner weiteren Abnahme der Differenz zwischen dem aktuellen und dem erwarteten Output führen.
- **Anpassungsfähigkeit:** KNN besitzen die Fähigkeit, die Stärken ihrer zwischenneuronalen Verbindungen an die Veränderungen ihrer Umgebung anzupassen. Dies ist besonders wichtig, wenn das Netz sich in einer nicht-stationären Umgebung befindet, wenn beispielsweise das Verhältnis zwischen dem Input- und Outputsignal sich im Laufe der Zeit sich verändert (einen Trend aufweist), oder eventuelle Periodizitäten trendabhängig sind.

Die **Nachteile** der Anwendung von KNN in der Prognoseerstellung:

- Der **Lernerfolg kann nicht garantiert werden**, weil kein allgemeingültiges Verfahren zum Design der Struktur, zur Wahl des Lernalgorithmus usw. existiert. Etwas vereinfachend könnte man von einem „trial and error Verfahren“ sprechen.
- Es besteht die **Gefahr von „Übertrainieren“** (overfitting), weil kein allgemeingültiges Kriterium für den Abbruch des Trainings existiert. Dies verbirgt die Gefahr des Verlustes an Anpassungsfähigkeit: Übertrainierte Netze werden rigide, ihre Fähigkeit zur Verallgemeinerung des Erlerneten wird eingeschränkt.
- Es gibt **keine Möglichkeit**, für die gefundenen Stärken der zwischenneuronalen Verbindungen eine **Begründung zu finden**, d.h. das Ergebnis kann nicht interpretiert werden. Das ist eine negative Folge des oben erwähnten Blackbox-Charakters von KNN.

1.3 Elemente und Strukturen von KNN

Ein neuronales Netz besteht aus den folgenden Elementen:

- ■ ■ Menge von Neuronen
- ■ ■ Zwischenneuronale Verbindungen (Synapsen)
- ■ ■ Gewichte der Verbindungen
- ■ ■ Aktivierungsfunktion
- ■ ■ Output-Funktionen der Neuronen
- ■ ■ Lernregel
- ■ ■ Umgebung, in der das Netz arbeitet

1.3.1 Die Elemente von KNN

Neuronen sind die Grundelemente von KNN. Sie sind Informationsverarbeitungseinheiten, in denen Inputsignale verarbeitet und als Outputsignale in das Netz weitergegeben werden.

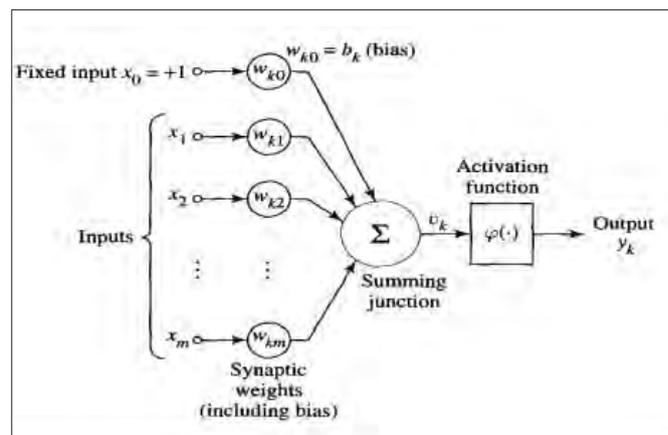


Abbildung 2:
Synaptische
Verbindungen

Quelle: Haykin, 1999, S. 11

Wie Abbildung 2 zeigt, besteht ein Neuron aus **Verbindungen (Synapsen)** die durch ihre **Gewichte (Stärke)** charakterisiert werden. In der Abbildung wird das Neuron k dargestellt, wobei w_{kj} die Stärke (das Gewicht) der Verbindung mit dem Inputsignal x_j bezeichnet. In KNN können die Gewichte sowohl positive als auch negative Werte annehmen.

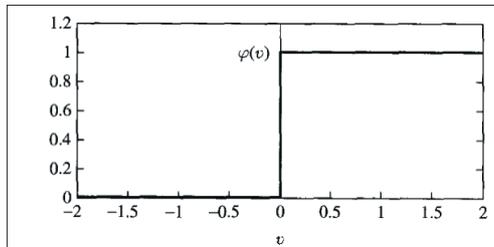
Die **Inputsignale** werden durch Summenbildung zusammengefasst, wobei die Gewichte der verschiedenen Synapsen berücksichtigt werden. Dies wird daher als eine lineare Kombination der Inputsignale bezeichnet.

Die **Aktivierungsfunktion** $\varphi(\cdot)$ dient zur Begrenzung der Amplitude des Outputsignals y_k , das vom Neuron k an weitere Neuronen des Netzes weitergegeben wird.

Die am häufigsten verwendeten Aktivierungsfunktionen sind die Schwellenfunktion, die lineare Beschränkung und die Sigmoid-Funktion:

Schwellenfunktion:

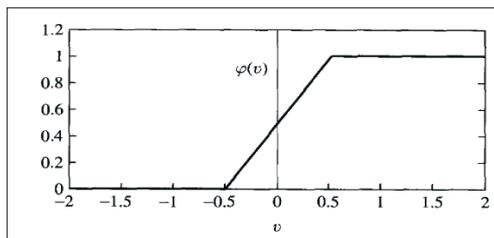
$$\varphi(v) = \begin{cases} 1 & \text{für } v \geq 0 \\ 0 & \text{für } v < 0 \end{cases}$$



Quelle: Haykin, S. 13 Abbildung 3: Die Schwellenfunktion

Lineare Beschränkung:

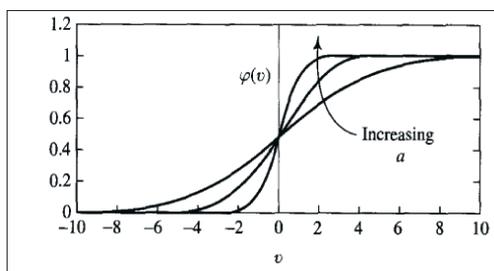
$$\varphi(v) = \begin{cases} \frac{1}{2} & \text{für } v \geq 0 \\ v & \text{für } \frac{1}{2} > v > -\frac{1}{2} \\ -\frac{1}{2} & \text{für } v \leq -\frac{1}{2} \end{cases}$$



Quelle: Haykin, S. 13 Abbildung 4: Lineare Beschränkung

Sigmoid-Funktion:

$$\varphi(v) = \frac{1}{1 + \exp(-\alpha v)} \quad \text{wobei } \alpha \text{ der Steigungsparameter der Funktion ist.}$$



Quelle: Haykin, S. 13 Abbildung 5: Sigmoid-Funktion

Die Sigmoid-Funktion ermöglicht eine beliebige, kontinuierliche Approximation einer Schwellenfunktion durch die entsprechende Wahl des Parameters α .

Der **Output des Neurons** k wird wie folgt berechnet:

$$y_k = \Phi(v_k), \quad \text{wobei} \quad v_k = \sum w_{kj} x_j$$

Die Lernregeln, die im Training des Netzes angewendet werden, beschreiben verschiedene Lernverhalten. Lernen wird als ein Prozess definiert, in dem die Parameter des Netzes durch Simulation von Signalen aus der Netzumgebung adaptiert werden. Dieser Lernprozess besteht aus folgenden Phasen: Das neuronale Netzwerk wird durch die Umgebung stimuliert, die Werte der Parameter werden verändert, dadurch ändert sich die Reaktion des Netzes auf den nächsten äußeren Impuls. Eine präzise definierte Menge der Lernregeln wird **Lernalgorithmus** genannt. Die Lernalgorithmen unterscheiden sich voneinander je nachdem, wie die Parameter (die Gewichte der zwischenneuronalen Verbindungen) verändert werden. Schließlich wird auch die Einbettung des Netzes in seiner Umgebung berücksichtigt, dies wird als **Lernparadigma** bezeichnet.

Die am häufigsten verwendeten Lernregeln:

- *Error-correction learning*: Diese Lernregel bewirkt die Approximation an die vorgegebenen (erwarteten) Outputwerte. Der Fehler wird in einer geschlossenen Feedback-Schleife durch die Veränderung der Gewichte der Synapsen minimiert. Wenn die erlaubten Veränderungen der Gewichte sehr klein sind, ist die Lerngeschwindigkeit ebenfalls klein, und das Training kann eine längere Zeit in Anspruch nehmen. Die Zulassung von zu großen Veränderungen kann aber dazu führen, dass die berechneten Output-Werte nicht konvergieren und der erwartete Output-Wert mit einem tolerierbaren Fehler nicht erreicht wird.
- *Hebbsches Lernen*: Bei der Anwendung dieser Regel wird das Gewicht einer Synapse erhöht, wenn die Neuronen, die durch eine Synapse verbunden sind, gleichzeitig (synchron) aktiviert werden. Wenn aber die Aktivierung der Neuronen nicht simultan (asynchron) erfolgt, wird das Gewicht der Synapse vermindert oder sogar eliminiert.
- *Kompetitives Lernen*: Entsprechend dieser Lernregel kann nur ein einziges von den Output-Neuronen einer Neuronen-Gruppe zu einem Zeitpunkt aktiviert werden, d.h. die Output-Neuronen des Netzes in einer Neuronen-Gruppe stehen miteinander im Wettbewerb, um aktiviert zu werden.

Wie schon erwähnt, beziehen sich die **Lernparadigmen** auf die Verbindung des Netzes mit seiner Umgebung. Die zwei wichtigsten Gruppen sind das überwachte (supervised) und das unüberwachte (unsupervised) Lernen.

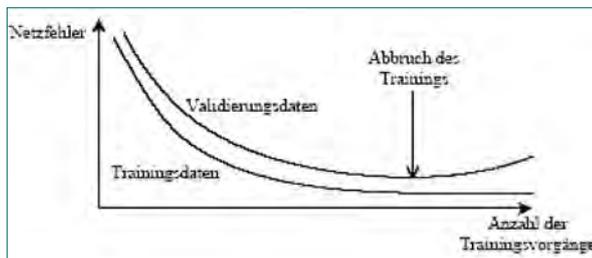
- Beim überwachten Lernen werden dem Netz im Training nicht nur die Inputs sondern auch der erwartete Output präsentiert und im Laufe des Trainings soll der Outputwert annähernd erreicht werden. (Siehe bspw. error-correction learning.)
- Beim unüberwachten Lernen werden dem Netz nur die Inputsignale präsentiert und das Netz muss die Muster in den Daten erkennen. (Siehe bspw. das kompetitive Lernen.) In diesem Fall kann man auch von selbstorganisierenden Netzen sprechen. Dieses Lernparadigma wird dann verwendet, wenn das Netz signifikante Muster oder Eigenschaften in den Inputdaten entdecken soll (bspw. Klassifizierungsaufgaben). Für Prognosezwecke ist es nicht geeignet.

Beim überwachten Lernen wird bei jedem Trainingsschritt der durchschnittliche quadratische Fehler (mean squared error) berechnet:

$$\text{MSE} = 1/n \sum_{i=1}^n (z_i - y_i)^2$$

wobei z_i den erwarteten Wert des Outputsignals im Zeitpunkt i bezeichnet,
 y_i den vom Netz berechneten Wert, und
 n die Länge des Trainingsatzes.

Abbildung 6:
Ablauf des Trainings eines KNN



Quelle: Eisenbach, 2005

Falls der Fehler im Bereich der tolerierten Abweichung liegt, kann das Training abgebrochen werden, anderenfalls wird es fortgesetzt. Um ein „Übertrainieren“ (overfitting) zu vermeiden, ist empfohlen, die Datensätze in Trainingsätze und in Validierungssätze zu trennen, und als Stopp-Kriterium nicht den MSE von dem Trainingsatz, sondern den von dem Validierungssatz zu verwenden.

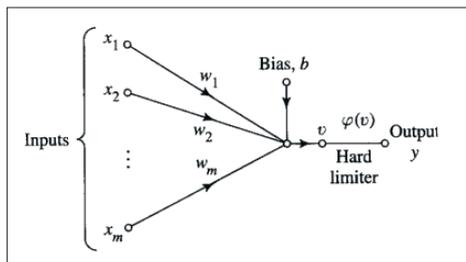
1.3.2 Die Strukturen von KNN

Die Struktur eines KNN steht im engen Zusammenhang mit dem Lernverhalten, das für das Training des Netzes verwendet wird.

1.3.2.1 Perzeptronen

Das einfachste neuronale Netz, *Perzeptron* genannt, wurde 1958 von dem Psychologen F. Rosenblatt entwickelt. Es besteht aus einer Inputschicht und einem Ausgabeneuron.

Abbildung 7:
Perzeptron



Quelle: Haykin 1999, S. 136

Dieses Netz lässt sich ohne prinzipielle Probleme erweitern, wenn mehrere Outputneuronen, d.h. eine Outputschicht berücksichtigt wird, in der jedes Outputneuron über einen eigenen unabhängigen Netzbereich verfügt.

Die Outputwerte werden mithilfe der Aktivierungsfunktion berechnet:

$$y_k = \varphi(v_k), \quad \text{wobei} \quad v_k = \sum w_{kj} x_j$$

Die Anwendbarkeit eines solchen Netzes ist jedoch beschränkt: Die Gewichte der Aktivierungsfunktionen v_k können nur dann bestimmt werden, wenn eine Reaktion (Verhaltensmuster) des Outputsignals auf ein Inputsignal unabhängig von seinen Reaktionen auf andere Inputsignale ist, d.h. die Aktivierungsfunktionen sind auf verschiedene Inputsignale linear separierbar.

Mit Perzeptronen kann das Error-correction learning simuliert werden. Entsprechend dieser Lernregel werden die Gewichte der zwischenneuronalen Verbindungen nach der so genannten Delta-Regel geändert:

$$\Delta w_{kj} = \epsilon (z_k - y_k) x_j, \quad \text{wobei } z_k \text{ der erwartete Wert des Outputsignals } k \text{ ist und der Parameter } \epsilon \text{ die Lerngeschwindigkeit bestimmt.}$$

1.3.2.2 Multi-Layer Perzeptronen

Eine Erweiterung der Netzstruktur von Perzeptronen wird erreicht, indem eine oder mehrere verborgene Schichten (hidden layers) von Neuronen berücksichtigt werden. Mithilfe der verborgenen Schichten kann die starke Einschränkung der Anwendbarkeit von einfachen Perzeptronen, welche die lineare Separierbarkeit der Aktivierungsfunktionen bedeutet, aufgehoben werden. Dadurch können simultane Abhängigkeiten der Verhaltensmuster der Outputsignale auf verschiedene Inputsignale berücksichtigt werden. Solche KNN werden als *Multi-Layer Perzeptronen (MLP)* bezeichnet. In der folgenden Abbildung ist ein Netz mit zwei verborgenen Schichten dargestellt.

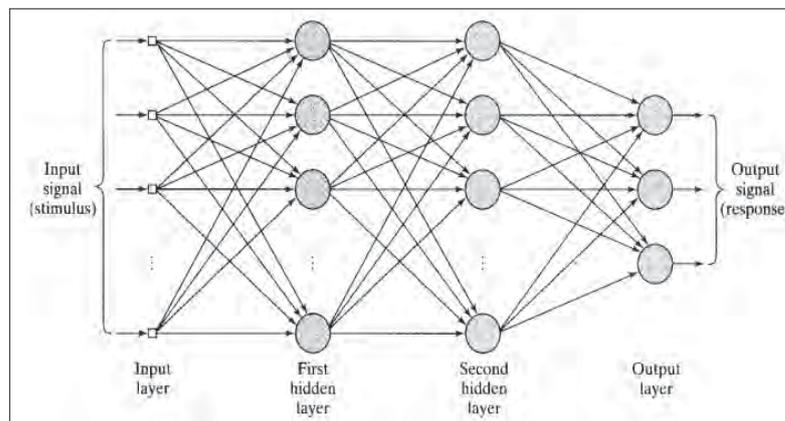


Abbildung 8: Multi-Layer Perzeptron

Quelle: Haykin, 1999 S. 159

Die Multi-Layer Perzeptronen (MLP) sind vorwärts gerichtete (feedforward) Netze, d.h. Rückkoppelungen werden nicht berücksichtigt. Dieser Netz-Typ wird häufig als das Standardnetz angesehen und wird bei Prognosen vielfach verwendet. Durch die ausschließlich nach vorne

gerichtete Verbindungsstruktur kann das Netz lediglich auf aktuelle Daten zurückgreifen, da es keine Möglichkeit gibt, Datensätze von früheren Zeitpunkten im Netz zu speichern. Da es aber für die Prognose von Zeitreihen oft notwendig ist, Informationen aus zeitverzögerten Datensätzen zu integrieren, müssen diese als separate Inputsignale ins Netz aufgenommen werden.

Die Multi-Layer Perzeptronen haben gegenüber den einfachen Perzeptronen einige wichtige Vorteile:

- ■ ■ Sie können jede logische Funktion (Boolsche Operatoren) darstellen, da die Unabhängigkeit der zu einem Outputneuron gehörenden Netzteile aufgehoben wird.
- ■ ■ Beliebige Funktionen können beliebig genau approximiert werden, wobei die zu approximierenden Funktionen nicht angegeben werden müssen.
- ■ ■ Es ist durchaus möglich, dass das Netz mit wenigen Verbindungen aber mit einer hohen Anzahl von Neuronen im Lernverhalten sich als besser erweist als ein Netz mit wenig Neuronen und vielen Verbindungen.

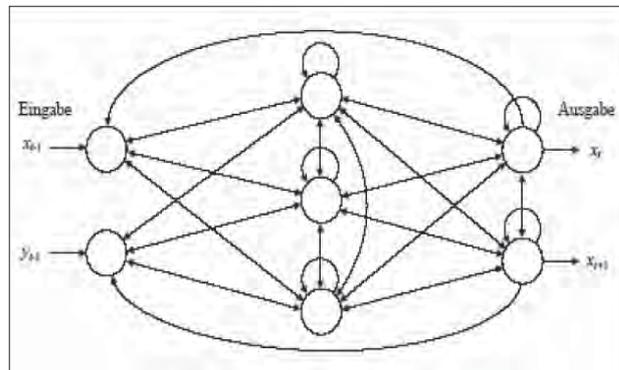
Bei den Anwendungen von Multi-Layer Perzeptronen wird häufig das so genannte *Backpropagation*-Lernverfahren eingesetzt. Um die Logik dieses Lernverfahrens beschreiben zu können, müssen wir zwischen funktionalen Signalen und Fehlersignalen unterscheiden.

Unter funktionalen Signalen versteht man die Inputsignale, die nach ihrem vorwärts gerichteten Weg durch das Netz (bei jedem Schritt mit einem unterschiedlichen Gewicht) die Outputneuronen erreichen. Die Fehlersignale, die sich bei einem überwachten Lernen als Abweichungen zwischen den berechneten und erwarteten Werten der Outputsignale ergeben, können im Netz rückwärts (also Schicht nach Schicht) auf „frühere“ Fehler der berechneten Werte von den Gewichten der zwischenneuronalen Verbindungen zurückgeführt werden. Daher der Name Backpropagation.

Der Rückverfolgungsalgorithmus ist eine Verallgemeinerung der Delta-Regel, um dieses Prinzip auch für mehrschichtige Netze anwenden zu können und zu ermitteln, wie die Gewichte der zwischenneuronalen Verbindungen in den verschiedenen verborgenen Schichten geändert werden sollen. Der Algorithmus wird für jeden Trainingslauf solange eingesetzt, bis die Ausgabefehler (die Abweichungen der berechneten von den erwarteten Outputsignalen) innerhalb eines Toleranzbereichs liegen. Wird dieser Toleranzbereich erreicht, wird das Training beendet.³

1.3.2.3 Rekurrente Netze

Rekurrente Netze (RNN) zeichnen sich zum Unterschied zu den Feedforward-Netzen dadurch aus, dass sie Abweichungen von der allgemeinen Verarbeitungsrichtung zulassen. Durch solche Verbindungen können Zyklen abgebildet werden, d.h. auch Rückkoppelungen können dargestellt werden. Die Rückkoppelungen können verschiedenartig sein. In der folgenden Abbildung ist ein RNN gezeigt, das direkte, indirekte und laterale Rückkoppelungen beinhaltet.



Quelle: Eisenbach, 2005, S. 32.

Abbildung 9:
Rekurrentes Netz

In der Abbildung 9 sind sowohl direkte als auch indirekte Rückkoppelungen dargestellt: Die beiden Signale x und y weisen mit den eigenen vergangenen Werten direkte Rückkoppelungen auf und sind mit den vergangenen Werten des jeweils anderen Signals (x mit y bzw. y mit x) - durch die verborgene Schicht - durch indirekte Rückkoppelungen verbunden.

Durch die Rückkoppelungen ist das Netz in der Lage, auch Informationen zu speichern, die über die Verarbeitung eines Datensatzes, der sich auf einen gewissen Zeitpunkt bezieht, hinausgeht. Dadurch können bei den Prognosen von Zeitreihen auch die im Netz gespeicherten Signale von vorhergehenden Zeitreihenwerten berücksichtigt werden, ohne dass diese explizit in das Netz eingegeben werden müssen. Wie diese Informationen im Netz bearbeitet werden, hängt davon ab, welche Art von Rekurrenz vom Netzdesigner definiert wird. Um eine sinnvolle Repräsentation von Rückkoppelungen erreichen, müssen bei der Anwendung von RNN die Datensätze in das Netz immer in chronologischer Reihenfolge eingegeben werden.

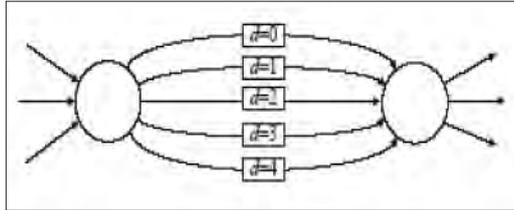
Durch die interne Speicherung von bereits verarbeiteten Werten sind RRN mächtiger als Feedforward-Netze, allerdings wird dieser Vorteil mit Schwierigkeiten im Training erkauft. Daraus folgt, dass man nicht darauf schließen darf, dass RRN den Feedforward-Netzen grundsätzlich überlegen wären.

1.3.2.4 Time-delay Netze

Time-delay Netze (TDNN) können im Gegensatz zu den Feedforward-Netzen ebenfalls Informationen über den Zeitablauf speichern. Das wird dadurch erreicht, dass Verbindungen zwischen den einzelnen Neuronen mit Zeitverzögerungen d_{kj} versehen werden, und dadurch diese zwischenneuronalen Verbindungen erst nach einer Zeit d_{kji} aktiv werden: Die jeweils ankommenden Informationen im Neuron k entstammen aus den mit d_{kji} verzögerten Werten von Neuron j , verzögert um einen Zeitabstand i .

Die folgende Abbildung zeigt zwei Neuronen, die mit verschiedenen Zeitverzögerungen verbunden werden.

Abbildung 10:
Time-delay Netz



Quelle: Eisenbach, 2005 S. 33.

In der Abbildung 10 sind fünf Zeitverzögerungen dargestellt: Während $d = 0$ eine Verbindung darstellt, die ohne Zeitverzögerung entsteht, werden die gleichen Signale vom ersten Neuron auch mit den Zeitverzögerungen 1, 2, 3, 4 zum anderen Neuron übertragen.

Da die zeitlich verzögerten Werte im Netz gespeichert werden, müssen diese in TDNN, ähnlich zu RNN, nicht bei jedem Trainingsschritt extra eingegeben werden. Wie die Abbildung zeigt, ist es möglich, zu den einzelnen Verbindungen mehrere unterschiedliche Zeitverzögerungen zuzuordnen. Diese können dann im Training mit verschiedenen Gewichten versehen werden, und jene, deren Gewicht „klein“ ist, weggelassen werden. Bei TDNN müssen die Datensätze ebenfalls in chronologischer Reihenfolge eingegeben werden.

Der Vorteil von TDNN gegenüber Feedforward-Netze ist ebenfalls relativ: Die Flexibilität und Adaptationsfähigkeit bei der Wahl der Zeitverzögerungen kann zu langwierigen und komplexen Trainingsabläufen führen.

1.4 Zusammenfassung

Im Folgenden fassen wir die Vor- und Nachteile von KNN für die Anwendung von Prognosen zusammen.

Es gibt mehrere Argumente, die für die Anwendung von KNN zu Prognosezwecken sprechen. Diese Argumente beziehen sich vor allem auf grundsätzliche Eigenschaften von neuronalen Netzen, die sich im Prognoseverfahren als nützlich erweisen. Es gibt aber auch mehrere Argumente, die gegen den Einsatz, bzw. nur für einen eingeschränkten Einsatz von KNN sprechen. Diese Argumente beziehen sich auf die Schwierigkeiten des Trainings bzw. auf den „trial and error Charakter“ der Methode.

Die Vorteile von KNN bei der Prognose von Zeitreihen sind:

- KNN sind universelle Funktions-Approximatoren, die jede stetige Funktion mit jedem gewünschten Grad an Genauigkeit berechnen können.
- Durch nichtlineare Aktivierungsfunktionen können KNN auch nichtlineare Zusammenhänge abbilden, ein Vorteil, der im Prognoseverfahren von Zeitreihen häufig notwendig ist.
- KNN können flexibel eingesetzt werden, weil keine speziellen Informationen über Wahrscheinlichkeitsverteilungen der stochastischen Prozesse und auch keine a priori Annahmen zur formalen Modellspezifikation benötigt werden.
- KNN sind relativ robust und reagieren deshalb nur wenig empfindlich auf das Rauschen in den Inputdaten.

- ■ ■ Da das Training des Netzes auch außerhalb der Trainingphase fortgesetzt werden kann, kann das Netz auch auf die sich über den weiteren Zeitablauf ändernden Zusammenhänge des stochastischen Prozesses angepasst werden.

Die Nachteile von KNN bei der Prognose von Zeitreihen sind:

- ■ ■ Der Lernvorgang eines KNN ist meistens ein sehr zeitaufwändiges Verfahren.
- ■ ■ Der Wissenserwerb von KNN ist nur durch Lernen möglich. Zusammenhänge, die schon bekannt sind, können nicht im Voraus in das Netz eingegeben werden.
- ■ ■ Es ist nicht möglich, aus einem trainierten Netz die zu Grunde liegenden kausalen Zusammenhänge, auf denen die Ergebnisse begründet sind, abzuleiten. Dies kann besonders dann, wenn die Prognosen als Entscheidungshilfe erstellt werden, ein besonderer Nachteil sein.
- ■ ■ Zwar sind viele Parameter eines KNN bereits durch die jeweilige Anwendung festgelegt, jedoch gibt es auch eine Vielzahl von weiteren Parametern, die noch spezifiziert werden müssen. In welcher Weise diese Spezifizierung erfolgen soll, dafür gibt es keine allgemeinen Regeln.
- ■ ■ KNN können lineare und nichtlineare Zusammenhänge gleichzeitig kaum berücksichtigen. Aus dem hier Gesagten folgt, dass die konkrete Entscheidung für oder gegen die Anwendung der KNN für Prognosen immer von der jeweiligen Situation der Anwendung abhängig ist.

2 Prognose der Schweineproduktion mit künstlichen neuronalen Netzen

Um die Möglichkeiten der Methode der KNN auch empirisch zu testen, wurden Prognosen der Schweineproduktion mithilfe von Multi-Layer-Perzeptronen (MLP) in verschiedenen Varianten berechnet. Dazu bedienten wir uns der Computersprache S-PLUS. Als Datenbasis verwendeten wir die Zeitreihen der monatlichen Schweineproduktion (in Tonnen) und der Ferkelpreise (als Inputvariablen) zwischen Jänner 1981 und Oktober 1997, bzw. zwischen Jänner 1980 und Dezember 2005. Als Kontrollperiode wurden jeweils die letzten 36 Daten (drei Jahre) der Zeitreihen berücksichtigt, d.h. das Training des Netzes basierte auf den Daten der vorangehenden Trainingsperiode.

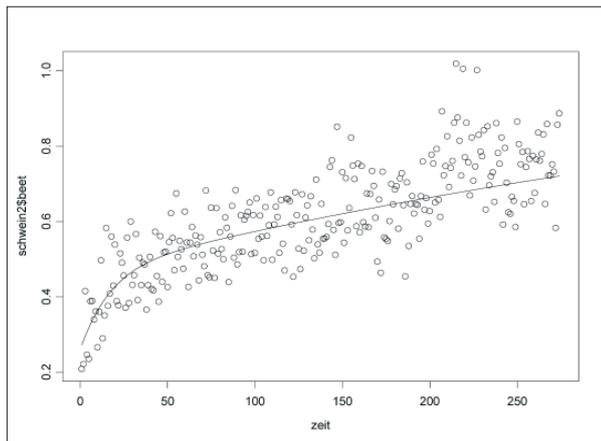
Die Struktur der berechneten Varianten enthält eine verborgene Schicht. Die Anzahl der Knotenpunkte wurde jedoch innerhalb der Schicht unterschiedlich gewählt. Ebenfalls wurden die Inputsignale variiert: Wir berücksichtigten neben einem Trend verschiedene Zeitverzögerungen der Outputvariable (der Schweineproduktion) und der Ferkelpreise.

Im Folgenden stellen wir vier ausgewählte Varianten von einer Vielzahl berechneter Prognosen vor. Sie veranschaulichen, wie schrittweise die Prognosen durch Veränderungen in einem KNN-Modell verbessert werden können.

2.1 Variante 1

Die erste Variante beruht auf den Daten zwischen Jänner 1981 und Oktober 1997 und verwendet als Inputsignal nur **einen** Trend. Die verborgene Schicht beinhaltet **drei** Knotenpunkte: d.h. es wurde für **ein** Outputsignal (Schweineproduktion) ein **MLP 1-3-1** berechnet.

Abbildung 11:
Ergebnis des
MLP 1-3-1.
Inputsignal:
Trend

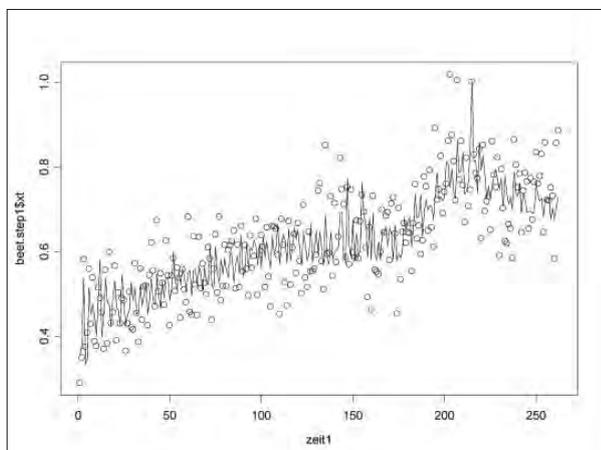


Wie aus der Abbildung 11 ersichtlich ist, kann mithilfe einer Trendkomponente als Inputsignal die Schweineproduktion sowohl für die Trainingsperiode als auch für die Kontrollperiode mit ähnlich großen Fehlern geschätzt werden. Diese Variante ist jedoch für Prognosezwecke ungeeignet.

2.2 Variante 2

In der zweiten Variante wurden neben dem Trend auch die zwölfmal und die neunmal verzögerten Werte der Outputvariable als Inputsignale und in der verborgenen Schicht fünf Knotenpunkte berücksichtigt, d.h. es wurde ein 3-5-1 MLP berechnet. Die Beobachtungsperiode war ebenfalls vom Jänner 1981 bis Oktober 1997.

Abbildung 12:
Ergebnis des MLP
3-5-1.
Inputsignale:
Trend, zwölfmal
und neunmal
verzögerte Werte
der Schweinepro-
duktion.



Wie Abbildung 12 zeigt, ist diese Variante imstande, sogar einzelne Ausreißer (etwa bei dem 220-ten Datenpunkt) zu erfassen, die Wiedergabe der Werte in der Kontrollperiode ist aber unbefriedigend.

2.3 Variante 3

In den nächsten zwei Schritten haben wir versucht, die Ferkelpreise als zusätzliche Inputsignale zu verwenden. Diese Varianten basieren auf den Zeitreihen von Jänner 1980 bis Dezember 2005.

In Abbildung 13 werden die Ergebnisse eines MLP 3-5-1 gezeigt. Die Inputsignale sind der Trend, die zwölfmal verzögerten Werte der Schweineproduktion und die neunmal verzögerten Ferkelpreise. In der verborgenen Schicht sind fünf Knotenpunkte.

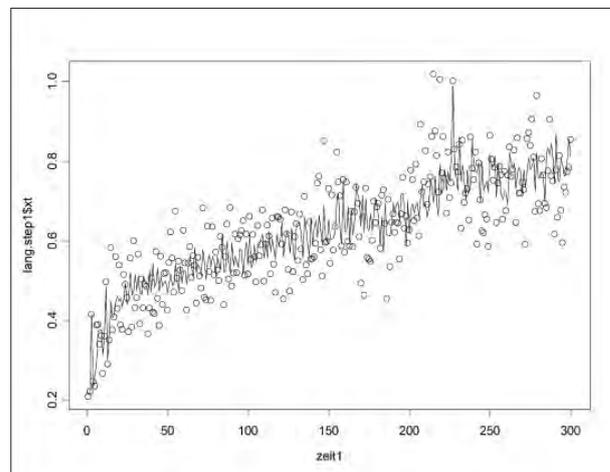


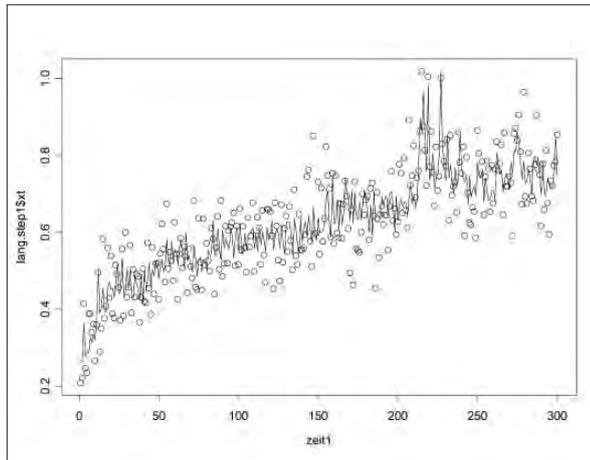
Abbildung 13:
Ergebnis des MLP
3-5-1. Inputsi-
gnale: Trend,
zwölfmal verzö-
gerte Werte der
Schweineproduk-
tion und neun-
mal verzögerte
Ferkelpreise

Wie die Abbildung zeigt, konnte durch die Einbeziehung der Ferkelpreise eine sichtbare Verbesserung der Prognoseergebnisse erreicht werden.

2.4 Variante 4

Die letzte Variante unterscheidet sich von der vorherigen dadurch, dass die Ferkelpreise nur einmal verzögert als Inputsignal berücksichtigt werden. Der Grund für die Wahl des einfach verzögerten Wertes war, dass diese die stärkste (negative) Korrelation mit der Schweineproduktion aufweist, nämlich -0.1738 . Es wurde wieder ein MLP 3-5-1 berechnet.

Abbildung 14:
Ergebnis des MLP
3-5-1. Inputsig-
nale: Trend,
zwölfmal verzö-
gerte Werte der
Schweineproduk-
tion und einmal
verzögerte Fer-
kelpreise.



Wie aus Abbildung 14 hervorgeht, hat die einfache Zeitverzögerung der Ferkelpreise die erhoffte Verbesserung der Prognose gebracht. In der Trainingsperiode konnten mehr Ausreißer erfasst werden als in der vorangegangenen Variante, und auch die geschätzten Werte in der Kontrollperiode liegen näher bei den Datenpunkten.

2.5 Zusammenfassung

Die Erfahrungen mit den verschiedenen Varianten der Prognose der Schweineproduktion zeigen, dass man mit den oben diskutierten, relativ einfachen KNN schon zufrieden stellende Prognosen berechnen kann. Der durchschnittliche quadratische Fehler in der Prognoseperiode konnte von der Variante 2 zur Variante 4 durch Änderung der Inputsignale nahezu halbiert werden. (Der Wert verminderte sich von 0.0260 auf 0.0152.)

Weitere Entwicklungsmöglichkeiten würden sich anbieten: Einerseits könnten zusätzliche Inputsignale (beispielsweise Futterpreise) eingeführt werden. Eine Voraussetzung dafür ist, dass genügend lange Zeitreihen zur Verfügung stehen.

Andererseits wäre es auch ein interessanter Versuch, ein KNN aufzubauen, in dem nicht nur die Schweineproduktion, sondern auch andere Produktionsbereiche des Schweinesektors (beispielsweise Zuchtsauen als Outputsignal) miteingefasst werden. Als Inputsignale könnten in einem derartigen Netz neben den verzögerten Werten der Outputsignale auch andere Variablen - Preise, Importe und Exporte usw. - eingeführt werden. Wenn man gleichzeitig auch die Preise prognostizieren möchte, bietet eher ein rekurrentes Netz (siehe unter 1.3.2.3) die besseren Möglichkeiten. Durch die Spezifikation der Netzstruktur und durch die Auswahl der relevanten Variablen (und deren wirksamen Zeitverzögerungen) könnten die simultanen Zusammenhänge im Schweinesektor erfasst werden. Allerdings soll hier wieder betont werden, dass solche Modelle und die mit ihrer Hilfe erstellten Prognosen kausal nicht interpretiert werden können. Diese Tatsache begrenzt die Anwendbarkeit der KNN-Methode.

Literaturverzeichnis

Callan, Robert (2003): Neuronale Netze . Pearson Studium, München

Eisenbach, Dominik (2005): Künstliche Neuronale Netze zur Prognose von Zeitreihen.

<http://cs.uni-muenster.de/Professoren/Lippe/diplomarbeiten/html/eisenbach/index.htm> [Stand: 17.01.2006]

Haykin, Simon (1999): Neural networks. A comprehensive foundation. Prentice-Hall, New Jersey, USA

Sauer, Jürgen (1991-2004): Neuronale Netze und Fuzzy Control-Systeme.

<http://fbim.fh-regensburg.de/~saj39122/vhb/NN-Script/script/gen/>
[Stand: 04.02.2004]

Venables, W.N. and B.D. Ripley (2002): Modern Applied Statistics with S. Springer, New York

Fußnoten

- ¹ Der Informationswert einer Messung hängt u. a. davon ab, inwieweit die gemessene Eigenschaft des Prozesses relevant für seine Entwicklung ist und wie genau die Messung durchgeführt wird.
- ² Es kann sich auch hier um Zeiträume handeln.
- ³ Eine detaillierte Beschreibung ist z.B. in Callan, S. 53. zu finden.

Impressum:

Agrarpolitischer Arbeitsbehelf Nr. 24

Eigentümer, Herausgeber, Verlag:

AWI - Bundesanstalt für Agrarwirtschaft

1030 Wien, Marxergasse 2

E-mail: office@awi.bmlfuw.gv.at

Web: www.awi.bmlfuw.gv.at

Gestaltung: [frey:grafik](http://www.freygrafik.at), Wien. www.freygrafik.at

Für den Inhalt verantwortlich: Marta Neunteufel, Josef Binder

Titelbild: http://www.et.fh-koeln.de/ia/dv/projekt_knn.html [Stand 13.12.2006]

Lektorat: Hubert Schlieber

Layout: Martina Wimmer

Druck: Bundesanstalt für Agrarwirtschaft

Copyright © 2007 by AWI - Bundesanstalt für Agrarwirtschaft, Wien

Alle Rechte vorbehalten.

Nachdruck, Vervielfältigung - auch auszugsweise - nur nach
Zustimmung und mit Quellenangabe





In letzter Zeit werden künstliche neuronale Netze (KNN) immer häufiger zur Extra- und Interpolation von Zeitreihen eingesetzt. Sie eignen sich besonders zur Bearbeitung unbekannter Ursache-Wirkungs-Zusammenhänge, die sich einer expliziten Modellierung entziehen, da sie imstande sind, Analogien und Ähnlichkeiten zu finden sowie Beschreibungen von unklaren Zusammenhängen zwischen Systemeingang und -ausgang zu liefern. In diesem Arbeitsbehelf wird die Methode der KNN für künftige Anwendungen in der Bundesanstalt für Agrarwirtschaft erläutert, ihre Potenziale und das Verfahren anhand einiger Beispiele illustriert. Der Bericht besteht aus zwei Teilen: Im ersten wird die Methode beschrieben, und im zweiten werden die Ergebnisse eines Beispiels (Extrapolation der Zeitreihe Schweineproduktion) diskutiert.