Einführung in Neuronale Netze
Backpropagation Learning

Motivation

Durch das Kriterium der linearen Separierbarkeit wird die Leistungsfähigkeit des Perzeptrons stark eingeschränkt. Diese kann durch die Einführung verborgener Schichten erhöht werden. Um ein Netz mit verborgenen Schichten sinnvoll trainieren zu können, wird jedoch die erwünschte Ausgabe für die verborgene Schicht benötigt. Allerdings ist zu jedem Trainingsmuster lediglich die gewünschte Ausgabe bekannt. Erst als das Backpropagation-Verfahren entdeckt wurde, war eine Rechenvorschrift bekannt, mit der die Verbindungen zu den verborgenen Schichten modifiziert werden können.

Das Backpropagation-Verfahren wurde in den 70ger Jahren von mehren Autoren vorgeschlagen, u.a. von Paul Werbos 1974. Allerdings geriet das Verfahren für über 10 Jahre in Vergessenheit, bis es von verschiedenen Autoren wiederentdeckt wurde. Am bekanntesten wurde die verallgemeinerte Delta-Regel, die von Rumelhart, Hinton und Williams, 1986, veröffentlicht wurde.

Das Backpropagation-Verfahren ist ein iteratives Verfahren. Es ermittelt eine Konfiguration der Gewichte im Netz, bei der die Fehlersumme über alle Trainingsmuster minimal ist. Künstliche Neuronale Netze auf der Basis von Backpropagation werden zum jetzigen Zeitpunkt für praktische Anwendungen am häufigsten verwandt.

Das Backpropagation-Netz

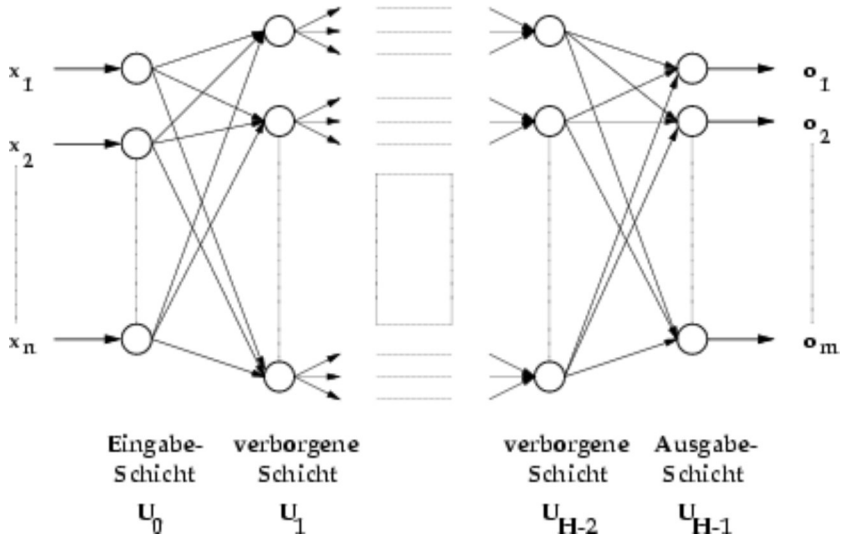
Backpropagation-Netze werden oft als multilayer perceptron (MLP) oder multilayer MADALINE bezeichnet, obwohl diese Begriffe nicht ganz korrekt sind. Wie bei allen anderen Arten von Künstlichen Neuronalen Netzen, gibt es nicht das Backpropagation-Netz, sondern eine Menge von unterschiedlichen Netzen, die die im folgenden beschriebenen Gemeinsamkeiten aufweisen:

Die Basis für Backpropagation-Netze bilden Backpropagation-Netze können sich also in der

mehrschichtige Feedforward-Netze. Meistens werden total verbundene Topologien benutzt, aber auch Topologien mit Shortcuts werden verwandt.

Topologie unterscheiden. Die folgenden Eigenschaften treffen jedoch für jedes Backpropagation-Netz zu:

Backpropagation-Netze bestehen aus einer Eingabeschicht, einer Ausgabeschicht und mindestens einer verborgenen Schicht.



Weitere Gemeinsamkeiten sind:

Die Neuronen sind in H Schichten angeordnet mit $H \geq 3$.
Die erste Schicht U_0 besteht aus n Eingabe- oder auch Verteilerneuronen. Diese Neuronen besitzen keine Eingangs-Gewichte und sie reichen die Eingabe unverändert an die zweite Schicht weiter.
Die folgenden $H-2$ Schichten U_1 bis U_{H-2} sind verborgene Schichten.
Die $H-1$-te Schicht U_{H-1} besitzt m Ausgabeneuronen.
Die ersten $H-1$ Schichten können zusätzlich ein konstantes Bias-Neuron besitzen.
Als Aktivierungsfunktion dient in allen nicht-konstanten Neuronen außerhalb der Eingabeschicht im Regelfall die gewichtete Summe der Eingaben.

Für die meisten Backpropagation-Netze gilt:

Die Ausgabefunktion aller nicht-konstanten Neuronen der verborgenen Schichten ist

sigmoid.

Die Ausgabefunktion der Ein- und Ausgabeschicht ist die Identität.

Im weiteren wird die folgende Notation verwandt:

Ein $\#U_0$ - $\#U_1$ - ... - $\#U_{H-2}$ - $\#U_{H-1}$ - **Netz** ist ein Backpropagation-Netz, indem $\#U_i$ die Anzahl der nicht-konstanten Neuronen in der *i-ten* Schicht bezeichnet.

$N_{h,i}$ bezeichnet das *i*-te Neuron der *h*-ten Schicht, dessen Ausgabe mit $o_{h,i}$ bezeichnet wird. Das Gewicht für die Verbindung zwischen den Neuronen $N_{h-1,j}$ und $N_{h,i}$ wird mit $w_{h,i,j}$ bezeichnet.

Das Bias-Neuron besitzt den Index *i=0*.

Das Bias-Neuron

Der Lernalgorithmus trennt die Trainingsdaten durch eine Hyperebene. Der Abstand zwischen dem Ursprung und der Hyperebene wird durch den Schwellenwert beschrieben. Die Trainingsdaten können somit leichter separiert werden, wenn der Schwellenwert variabel gehalten wird. Erreicht wird dies durch die Einführung konstanter Bias-Neuronen in jeder Schicht bis auf die Ausgabeschicht.

Da die Ausgabe jedes Bias-Neurons konstant eins ist, stellten die Verbindungen des Bias-Neurons zu den Neuronen der folgenden Schicht einen zusätzlichen Bias-Eingang dar. Wird der Schwellenwert der Neuronen im Netz wird auf null gesetzt, kommt der Bias-Eingang einem variablen Schwellenwert gleich, da die Ausgangsgewichte der Bias-Neuronen durch die Lernregel modifiziert werden.

Die bisherige Schwellenwertfunktion lautet: $\vec{w} \cdot \vec{x} \geq \theta$. Damit gilt:

$$\vec{w} \cdot \vec{x} - \theta \geq 0 \iff \sum_{i=1}^{n} \vec{w}_i \cdot \vec{x}_i - \theta \geq 0.$$

Die Aktivierung eines Neurons im Backpropagation-Netz wird berechnet, indem der Schwellenwert auf 0 gesetzt und ein zusätzlicher Bias-Input eingeführt wird:

$$\sum_{i=1}^{n} \vec{w}_i \cdot \vec{x}_i + \vec{w}_0 \geq 0.$$

Somit entspricht $\vec{w}_0$ einem neuen Schwellenwert, weil gilt: $\vec{w}_0 = -\theta$. Da $\vec{w}_0$ durch die

Backpropagation-Lernregel modifiziert wird, entspricht der Biaseingang somit einem variablen Schwellenwert.

Die Ausgabefunktion

Mit der Anzahl der Schichten steigert sich die Mächtigkeit der Netzes. Allerdings können lineare Netze nur lineare Funktionen approximieren. An dieser Eigenschaft ändert auch die Einführung weiterer verborgener Verarbeitungseinheiten nichts.

Um auch nicht-lineare Funktionen darstellen zu können, muß also eine nicht-lineare Komponente in das Netz eingebaut werden. Hierfür bietet sich eine nicht-lineare Ausgabefunktion an, welche die Aktivierung des Neurons zu dessen Ausgabe transformiert.

Als Ausgabefunktion dient oft die sigmoide Funktion:

$$s(x) = \frac{1}{1 + e^{-x}}.$$

Es existieren aber noch weitere Beispiele für sigmoide Funktionen. Alle sigmoiden Funktionen gemein, daß sie die Treppenfunktion approximieren und differenzierbar sind. Die Differenzierbarkeit geht entscheidend in die Herleitung des Gradientenabstieg-Verfahrens ein.

Es besteht die Möglichkeit, das Argument der Ausgabefunktion, also die Aktivierung des Neurons, mit einer Konstante > 0 zu multiplizieren. Je größer die Konstante ist, um so besser ist die Approximation der Treppenfunktion. In der Praxis wird die Konstante auf 1 gesetzt.

Zusammenfassend lassen sich folgende Eigenschaften einer sigmoiden Ausgabefunktion festhalten:

- Eine sigmoide Ausgabefunktion ermöglicht die Approximation einer nicht-linearen Funktion durch ein Backpropagation-Netz.
- Eine sigmoide Ausgabefunktion approximiert die Treppenfunktion.
- Eine sigmoide Ausgabefunktion ist differenzierbar.

Aber auch die Fehlerfunktion wird durch die Wahl einer sigmoiden Ausgabefunktion beeinflußt.

Die Fehlerkurve wird geglättet.
Die Fehlerkurve besitzt nur in einem Minimum oder Maximum eine waagerechte Tangente. $\Rightarrow \quad -\nabla_{\vec{w}} F(\vec{w})$ zeigt immer in Richtung des Minimums.
Sigmoide Ausgabefunktionen forcieren die Entstehung lokaler Minima.

Die Gewichtsmodifikation

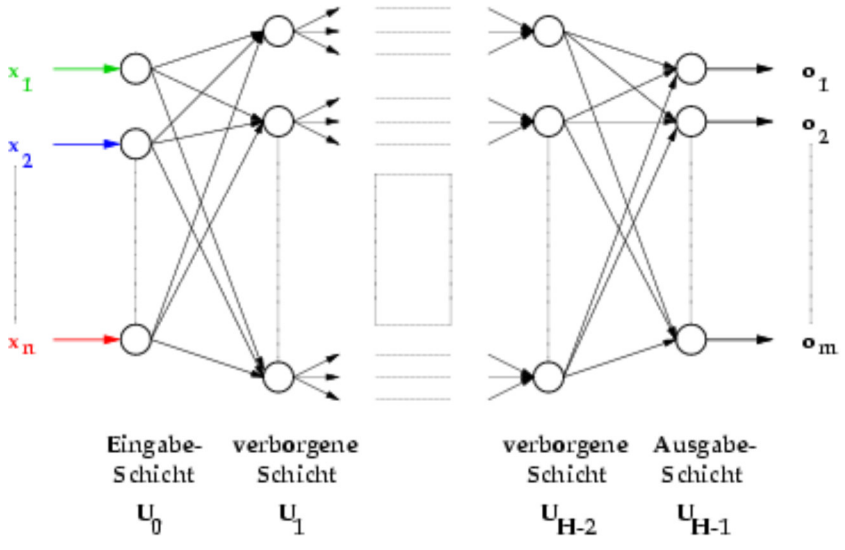
Das Lernen erfolgt bei Backpropagation-Netzen durch überwachtes Lernen. Das Training durchläuft alternierend drei Phasen.

Die 3 Phasen des Backpropagation-Algorithmus:

1. Forward Pass
2. Bestimmung des Fehlers
3. Backward Pass

1. Forward Pass

Dem Netz wird ein beliebiger Eingabevektor $\vec{x}$ aus der Trainingsmenge präsentiert. Ist $\#(U_0) = n$ so gilt $\vec{x} = (x_1, \ldots, x_n)$.



In der Schicht U_1 wird für jedes Neuron die Aktivierung berechnet und dann mittels der Ausgabefunktion die Ausgabe ermittelt. Die Ausgaben der Schicht U_1 bilden die Eingabe für die Schicht U_2.
Die Ausgabe der Vorgängerschicht bildet also die Eingabe der folgenden Schicht.

Die Daten durchlaufen somit das Netz schichtweise von links nach rechts. Erreicht der Forward Pass die Ausgabeschicht, wird die Ausgabe bestimmt und der Vektor $\vec{o} = (o_1, ..., o_m)^T$ mit $m = \#(U_{H-1})$ wird ausgegeben.

2. Fehlerbestimmung

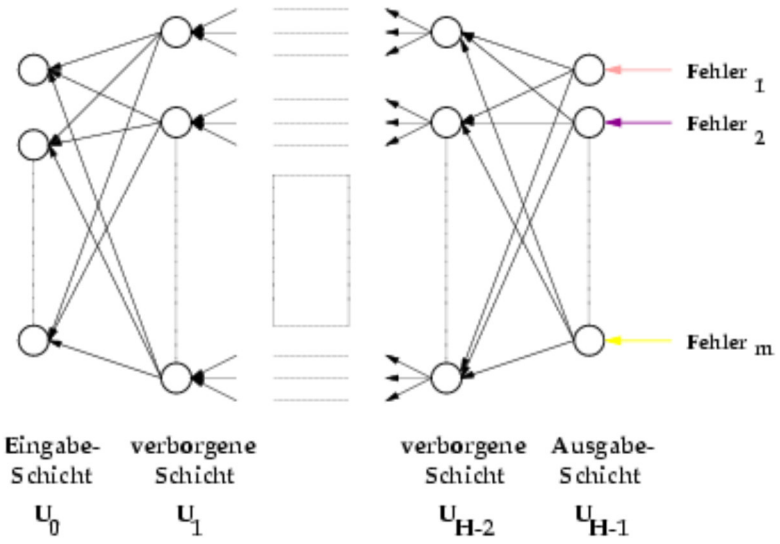
Die Eingabe $\vec{x}$ ist ein Element der Trainingsmenge. Zu jedem Element der Trainingsmenge ist auch die gewünschte Ausgabe des Netzes bekannt. Mit Hilfe der Fehlerfunktion wird der Fehler des Netzes bestimmt. Eine vorgegebene Güteschwelle entscheidet über den weiteren Verlauf des Trainings.

Liegt der Fehler oberhalb dieser Schwelle, erfolgt eine Modifikation des Netzes durch den Backward Pass. Liegt der Fehler unterhalb einer vorgegebenen Güteschwelle, wird das Training beendet und gegebenenfalls eine Testphase eingeleitet um die Generalisierungsfähigkeit zu überprüfen.

3. Backward Pass

In diesem Schritt werden sukzessive die Verbindungen zwischen den Neuronen in einem Backpropagation-Netz modifiziert. Die Modifikation erfolgt mit Hilfe einer Lernregel, für die der Fehler des Netzes die Grundlage bildet.

Der Backward Pass erfolgt in entgegengesetzter Richtung zum Forward Pass. Daher spricht man auch von einem rückwärtsverteilten Fehler.



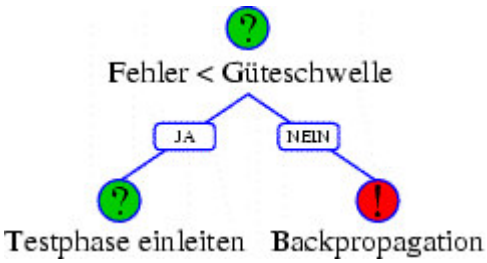
- Im ersten Schritt werden die Gewichte zwischen der Ausgabeschicht U_{H-1} und der mit ihr verbundenen verborgenen Schicht U_{H-2} modifiziert.

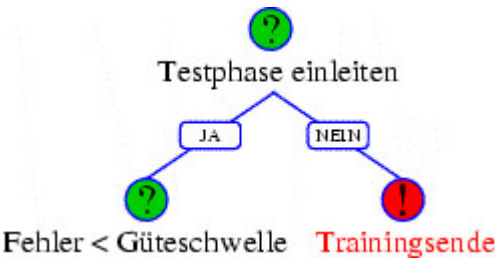
- In den folgenden Schritten werden die Verbindungen zwischen den verborgenen Schichten modifiziert.
- Im letzten Schritt werden die Verbindungen zwischen der verborgenen Schicht U_1 und der Eingabeschicht U_0 modifiziert.

Die Generalisierungsfähigkeit des Netzes

Die Trainingsdaten stellen Stützstellen dar, über die das Backpropagation-Netz eine Funktion approximiert. Aber auch außerhalb der Stützstellen soll das Netz eine genügende Genauigkeit aufweisen. Diese bedeutende Eigenschaft wird als Generalisierungsfähigkeit des Netzes bezeichnet. Die Generalisierungsfähigkeit wird mit Hilfe einer Testmenge verifiziert, die in keiner Komponente mit der Trainingsmenge übereinstimmt.

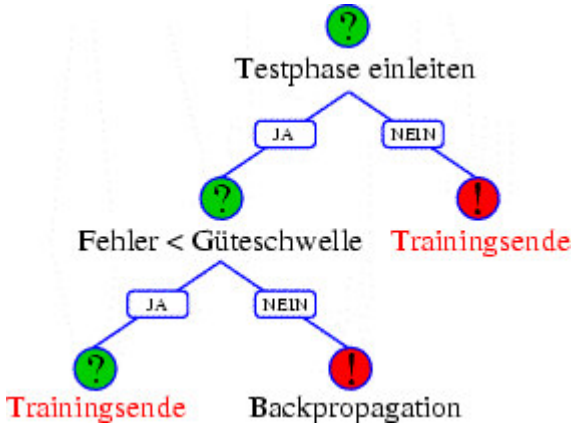
Die Trainingsphase dauert an, bis die gewünschte Genauigkeit erreicht wurde. Um die Generalisierungsfähigkeit des Netzes zu verifizieren, kann an diesem Punkt des Trainings eine Testphase eingeleitet werden.





Wird das Training des Netzes abgebrochen, werden die Trainingsmuster perfekt erkannt. An diesem Punkt des Trainigs kann jedoch nicht beurteilt werden, ob das Netz lediglich die einzelnen Muster gelernt hat, oder ob eine grundlegende Abbildungsvorschrift für die Trainigsmuster gefunden wurde.

In der Testphase wird dem Netz die Testmenge präsentiert. Der zur Testmenge gehörige Fehler entscheidet über den weiteren Verlauf des Trainings. Liegt der Fehler weiterhin unterhalb der zuvor definierten Güteschwelle, kann das Training endgültig abgeschlossen werden. Ist der Fehler zu groß, muß das Netz nachtraininiert werden.



Die Anzahl innerer Einheiten

Im Allgemeinen steigert die Anzahl verborgener Schichten die Mächtigkeit des Netzes. Allerdings gibt es keine Möglichkeit die optimale Anzahl verborgener Schichten für ein spezielles Problem zu bestimmen. Die Praxis hat gezeigt, daß häufig eine verborgene Schicht ausreicht.

Aber auch die otimale Anzahl der Neuronen in einer verborgenen Schicht wird durch das Problem bestimmt. Generell kann Folgendes über die Anzahl der inneren Einheiten in einem Backpropagation-Netz festgehalten werden:

Anzahl innerer Einheiten zu klein	1. Das Backpropagation-Netz kann das Problem nicht lösen.
Anzahl innerer Einheiten zu groß	• Es besteht die Gefahr des Overtrainings. • Das Problem wird nur Teilweise durch Generalisierung erkannt. • Die überflüssigen Einheiten sind ausschließlich zur Erkennung der restlichen Trainingsmuster bestimmt (quasi lokale Speicherung). • Es wird keine grundlegende Abbildungsvorschrift gefunden.

Dem Problem des Overtrainings kann durch rechtzeitige Validation begegnet werden. Allerdings reduziert die Validation die Geschwindigkeit des Verfahrens. Unterscheiden sich Trainingsfehler und Testfehler jedoch sehr, sollte dennoch die Anzahl der Testphasen während des Trainigs erhöht werden.

Zurück zum letzten Kapitel Zum nächsten Kapitel