Einführung in Neuronale Netze
Backpropagation Learning

Modifikationen des Gradientenabstiegs

Um die Probleme des Backpropagation-Verfahrens zu bewältigen, wurden von verschieden Autoren Modifikationen des Verfahrens entwickelt. Fast alle diese Modifikationen beruhen auf Heuristiken, die meist eine spürbare Beschleunigung des Konvergenzverfahrens bewirken.

Allerdings lassen sich stets Fälle konstruieren, in denen die Annahmen der Heuristiken nicht mehr gelten. Daher ist es sogar möglich, daß eine Verschlechterung gegenüber dem klassischen Gradientenabstieg eintritt.

Im Folgenden werden einige der verbreitesten Modifikationen kurz beschrieben.

Die Betrachtung der Modifikationen wird vereinfacht, indem die Gewichte systematisch mit i = 1,...,q durchnummeriert werden.

Momentum-Version

Diese Modifikation des Backpropagation-Algorithmus geht auf Hinton und Williams zurück und wurde erstmals 1986 in Parallel Distributed Processing: Explorations in the Microstructure of Cognition beschrieben.

Dieses Verfahren ist auch unter dem Namen Konjugierter Gradientenabstieg (conjugate gradient descent) bekannt.

Nachdem sich die Bestimmung der optimalen Lernrate als sehr schwierig erwiesen hat, lag die

Die jeweilige Situation wird erkannt, indem den vorhergegangenen Gewichtsänderungen eine

Idee sehr nahe, dieses Problem durch das Verfahren selbst lösen zu lassen. Die Momentum-Version erhöht die Schrittweite η auf flachen Niveaus und reduziert η in Tälern.

gewisse Bedeutung bei der Berechnung der aktuellen Gewichtsveränderung eingeräumt wird.

Heuristik

- In flachen Plateaus bleibt das Vorzeichen des Gradienten in zwei aufeinander folgenden Schritten unverändert.
- In einem Tal wechselt das Vorzeichen des Gradienten in zwei aufeinander folgenden Schritten.

Wird diese Heuristik berücksichtigt, ergibt sich folgende Rechenvorschrift zur Modifikation der Gewichte:

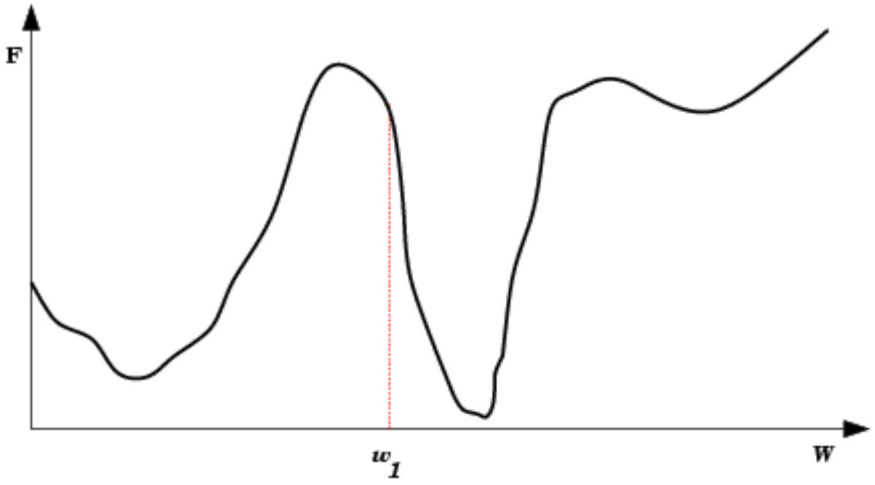
$$w_i(t+1) = w_i(t) + \triangle w_i(t)$$

$$\begin{aligned} \triangle w_i(t) &= -(1-\alpha)\ \eta\ \frac{\partial F(t)}{\partial w_i(t)} + \alpha \triangle\ w_i(t-1) \\ &= -(1-a)\ \eta\ \sum_{j=0}^{t} \alpha^j \frac{\partial F(t-j)}{\partial w_i(t-j)}. \end{aligned}$$

Dabei entspricht F dem MSE, $\alpha \in [0,1[$ dem Moment und η der Lernrate.
$\triangle w_i(t-1)$ beschreibt die Gewichtsmodifikation im letzten Schritt.

$\triangle w_i(t)$ ist der durch α^j exponentiell gewichtete Durchschnitt aller bis zu diesem Zeitpunkt berechneten Ableitungen. Das Moment α steuert das Verhältnis zwischen der aktuellen Veränderung und den alten Veränderungen von w_i. Für $\alpha = 0$ entspricht die obige Vorschrift der verallgemeinerten δ-Regel.

Die Summe der bis zum Zeitpunkt t-1 berechneten Gewichtsveränderungen wird Momentumterm genannt. Die exponetielle Gewichtung der Summe durch die α^j bewirkt, daß eine Ableitung mit zunehmendem Alter immer mehr an Bedeutung verliert, da α^j für große j gegen 0 konvergiert.



Durch Betätigen von vorheriger Schritt und nächster Schritt wird die Auswirkung des Momentumterms auf den Gradientenabstieg verdeutlicht. Dieselbe Fehleroberfläche bewirkte bei Standard-Backpropagation lediglich die Oszilation des Verfahrens.

Durch die Addition des Momentumterms wird dem Verfahren eine gewisse Trägheit verliehen. Dadurch steigt die Tendenz, die Richtung der Änderung beizubehalten. So wird es dem Verfahren erleichtert, Minima in steilen Tälern zu erreichen und flache Plateaus schneller zu überwinden.

Die Momentum-Version hat zwei Schwächen:

1. Der Beschleunigung des Gradientenabstiegs sind auch in der Momentum-Version Grenzen gesetzt. Wenn alle Ableitungen konstant gleich sind, besitzt die gewichtete Summe eine obere Schranke. Damit ist auch die größtmögliche Gewichtsveränderung beschränkt. Die Beschleunigung des Verfahrens ist also nicht grenzenlos.

2. Die Summe ab **l=1** kann betragsmäßig größer als die Summe ab **l=0** sein. Unterscheiden sich zudem noch die Vorzeichen, wird $\vec{w}$ in die falsche Richtung verschoben. Es findet also nicht mehr zwangsläufig ein Gradientenabstieg statt. Dadurch wird die Konvergenz des Verfahrens beeinträchtigt.

Wahl von Moment und Lernrate

Die Bestimmung von α und η hängt auch in dieser Situation von der Beschaffenheit der Fehleroberfläche ab. Generell kann man sagen, daß das Moment sehr viel größer als die Lernrate gewählt wird (z.B. α = **0.9** und η = **0.05**). Diese Wahl von α ist typisch.

In stark gekrümmten Gebieten versagt das Verfahren jedoch bei dieser großen Belegung von α schnell. Somit muß auch bei diesem Verfahren der optimale Wert experimentell bestimmt werden.

Weight Decay

Diese Art der Modifikation geht auf Paul Werbos zurück und basiert auf der Idee, große Gewichte zu eliminieren, da diese die Zerklüftung der Fehleroberfläche forcieren würden. Zudem wird die Generalisierungsfähigkeit des Netzes erhöht und die Initialisierung der Gewichte verliert an Bedeutung. Aus neurobiologischer Sicht sind große Gewichte ohnehin unplausibel.

Werbos versucht den Fehler zu minimieren und gleichzeitig die Gewichte klein zu halten. Erreicht wird dies durch die Einbindung der Forderung als Term in die Fehlerfunktion. Je größer die Gewichte sind, umso größer ist der Term, der zum MSE addiert wird.

$$F^{neu} = F + \frac{d}{2} \sum_i w_i^2$$

Die Bildung der partiellen Ableitung erfolgt nach der folgenden Formel:

$$\frac{\partial F^{neu}}{\partial w_i} = \frac{\partial F}{\partial w_i} + d \cdot w_i$$

Zu große Werte für ***d*** würden die Gewichte permanent auf zu kleinen Werten halten. Daher empfielt die Praxis, ***d*** nicht größer als **0.005**, aber auch nicht kleiner als **0.03** zu wählen.

Damit lautet der weight decay : $\triangle w_i(t) = \eta \frac{\partial F(t)}{\partial w_i(t)} - d \cdot w_i(t-1)$

Quickpropagation

Es erwies sich bald, daß das Newton-Verfahren zur Bestimmung der Nullstellen der Fehlerfunktionen zu rechenintensiv ist. Scott Fahlman modifizierte 1988 das Newton-Verfahren, welches als Quickprop-Verfahren bekannt wurde.

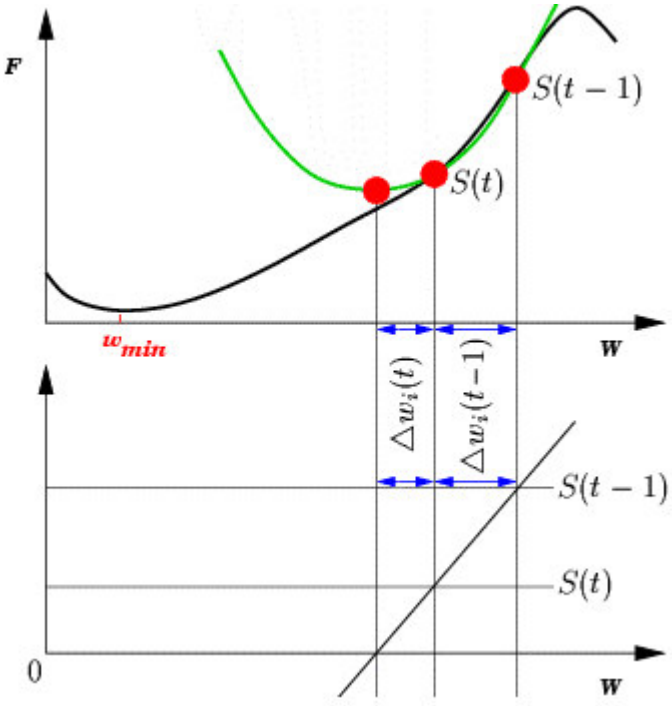
Zwar ist Quickprop viel rechenintensiver als die Standard-Backpropagation, aber in der Praxis hat es sich als relativ schnelles Verfahren bewährt.

Quickprop berechnet die erste Gewichtsveränderung üblicherweise mit Standard-Backpropagation und richtet sich dann nach der folgenden Heuristik:

Heuristik

- Ein Tal innerhalb einer Fehleroberfläche kann durch eine nach oben offene Parabel beschrieben werden.

Es besteht also die Möglichkeit, das Minimum der Fehlerfunktion zu bestimmen, indem der Scheitelpunkt der Parabel berechnet wird.



Zur Berechnung der Parabel werden die Ableitungen der Fehlerfunktion $S = \partial F / \partial w_i$ zum aktuellen Zeitpunkt ***t*** und zum vorangehenden Zeitpunkt ***t-1*** genutzt. Ist zusätzlich die Gewichtsveränderung $\triangle w_i(t-1)$ bekannt, kann $\triangle w_i(t)$ und damit der Scheitelpunkt der Parabel bestimmt werden. Die Betrachtung des Beispiels verdeutlicht, daß auch Quickprop nicht in einem Schritt in das Minimum der Fehlerfunktion führt. Dies liegt in der groben Approximation der Fehlerfunktion begründet. Allerdings nähert sich das Verfahren immer weiter dem Minimum an, da es sich um ein iteratives Verfahren handelt.

Die einzige Unbekannte in der obigen Grafik ist $\triangle w_i(t)$.
Die Berechnung erfolgt mit der folgenden Formel, die dem Momentum-Term ähnelt:

$$\frac{\triangle w_i(t)}{\triangle w_i(t-1)} = \frac{S(t)}{S(t-1) - S(t)}$$

$$\text{mit} \quad S(t) = \frac{\partial F}{\partial w_i(t)}.$$

Damit lautet die Formel für die aktuelle Gewichtsveränderung:

$$\Delta w_i(t) = \frac{S(t)}{S(t-1) - S(t)} \cdot \Delta w_i(t-1) \cdot$$

Bei Betrachtung des Quotienten lassen sich **vier Fälle** unterscheiden:

1.	**S(t) < S(t-1)** **sgn(S(t)) = sqn(S(t-1))**	Die aktuelle Steigung S(t) ist kleiner als die vorangegangene Steigung S(t-1) und hat das gleiche Vorzeichen. Dann erfolgt eine Gewichtsänderung in die selbe Richtung wie zuvor.
2.	**S(t) > S(t-1)** **sqn(S(t)) = sqn(S(t-1))**	Die aktuelle Steigung ist größer als die vorangegangene Steigung. In diesem Fall wäre die Parabel nicht nach oben, sondern nach unten geöffnet. Somit entfernt sich das Verfahren von dem Minimum und die Heuristik versagt.
3.	**sgn(S(t)) ≠ sgn(S(t-1))**	Die aktuelle Steigung verläuft in entgegengesetzter Richtung zu der vorangegangenen Steigung. Also wurde das Minimum übersprungen und das Verfahren befindet sich nun auf der gegenüberliegenden Seite des Tals. Im nächsten Schritt wird dann eine Position zwischen den zwei vorangehenden Positionen auf der Fehleroberfläche gewählt.
4.	**S(t) = S(t-1)**	Die aktuelle Steigung ist gleich der vorangegangenen Steigung. In diesem Fall würde der Quotient unendlich groß werden bzw. die Berechnung wegen Division durch 0 abbrechen und die Heuristik versagt.

Das Problem aus dem 4. Fall kann durch die Einführung eines maximalen Wachstumfaktors μ gelöst werden. Durch die zusätzliche Bedingung

$$| \Delta w_i(t) | \leq \mu \cdot | \Delta w_i(t-1) |$$

wird die Gewichtsänderung durch das μ-fache der letzten Gewichtsänderung beschränkt.

Allerdings darf μ nicht zu groß gewählt werden. Es empfielt sich ein Wert zwischen **1,75** und **2,25**. Um zusätzlich das Anwachsen der Gewichte zu beschränken, kann Quickprop mit weight decay kombiniert werden.

Die $\delta - \delta$ - und die $\bar{\delta} - \delta$ - Regel

Jacobs und Sutton machten 1988 naheliegende Vorschläge zur Verbesserung der Backpropagation-

Lernregel.

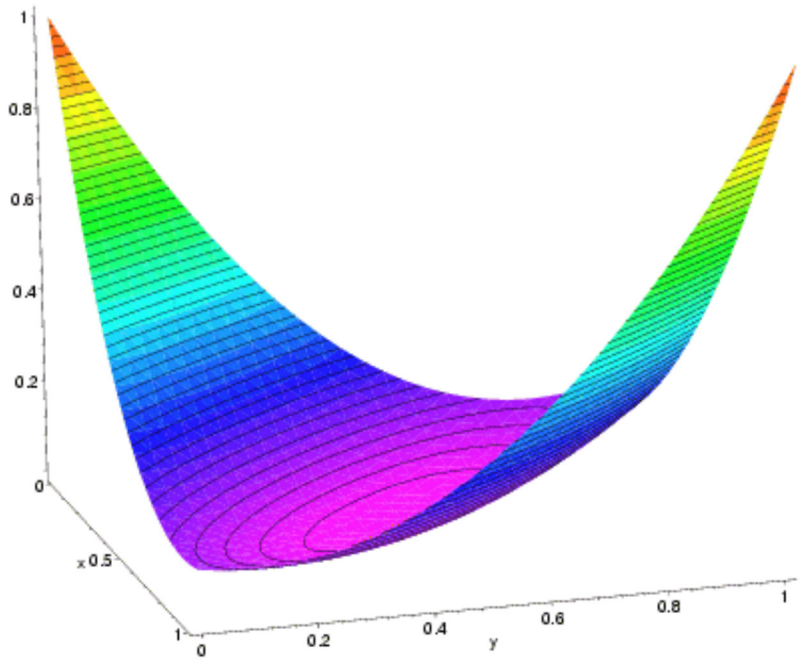
Das Krümmungsverhalten der Fehleroberfläche ist in jeder Dimension unterschiedlich. Dieser Eigenschaft wird eine einheitliche Lernrate nicht gerecht. Daher stellten Jacobs und Sutton die Forderung nach einer individuellen Lernrate für jedes Gewicht im Netz.

Da sich die Krümmung der Fehleroberflläche stetig ändert, sollten die Lernraten in der Lage sein, ihren Wert verändern zu können. Die Steuerung der Lernrate erfolgt nach den folgenden zwei Heuristiken:

Heuristik

- Jedes Gewicht benötigt eine individuelle Lernrate.
- Haben mehrere aufeinanderfolgende Ableitungen nach einem Gewicht dasselbe Vorzeichen, wird die Lernrate erhöht. Wechseln die Ableitungen für ein Gewicht in mehreren aufeinander folgenden Schritten das Vorzeichen, wird die Lernrate verringert.

Diese Heuristik ähnelt der Heuristik der Momentum-Version. Allerdings lassen sich auch hier Situationen konstruieren, in denen diese Heuristik versagt:



Für jeden Punkt oberhalb der Talsohle ist die Krümmung der Oberfläche in x- und in y-Richtung so stark, daß die obige Heuristik eine Verringerung der Lernrate bewirken würde.

Allerdings wäre eine Vergrößerung der Lernrate wünschenswert, um die Sohle des Tals schneller zu erreichen.

Einführung individueller Lernraten

Es gilt $\vec{w} \in \mathbb{R}^q$. Die Heuristik fordert somit die Einführung von q individuellen Lernraten $\eta_1, \ldots, \eta_q > 0$.

Damit lautet die neue Lernregel zur Modifikation eines Gewichts

$$w_i^{neu} = w_i^{alt} - \eta_i \frac{\partial F}{\partial w_i},$$

und die Lernregel für alle Gewichte

$$\vec{w}^{neu} = \vec{w}^{alt} - (\eta_1 E_{1,1} \bigtriangledown F(\vec{w}) + \ldots + \eta_q E_{q,q} \bigtriangledown F(\vec{w}))$$
$$= \vec{w}^{alt} - \sum_{i=1}^{q} \eta_i E_{i,i} \bigtriangledown F(\vec{w}).$$

Dabei werden die Lernraten systematisch durchnummeriert und den w_i die entsprechende Lernrate $\eta_i > 0$ zugeordnet. $E_{i,i}$ ist eine q x q-Matrix, die nur in der k-ten Zeile und Spalte eine 1 trägt und ansonsten in jeder Komponente $1 \leq k \leq q$ Null ist.

Bei diesem Verfahren wird nicht mehr $F(\vec{w})$ minimiert, sondern gleichzeitig für jede Koordinate von $\vec{w}$ nach dem $min_{w_i} F(\vec{w})$ gesucht. Deshalb spricht man bei diesem Verfahren von einem Koordinatenabstieg und nicht mehr von einem Gradientenabstieg. Um zu Beweisen, daß das Verfahren global konvergent ist, wird folgender Hilfssatz benötigt:

Hilfssatz

Sei $G : D \subset \mathbb{R}^q \to \mathbb{R}$ *differenzierbar und* $\vec{w} \in \mathring{D}$.
Für ein $\vec{v} \in \mathbb{R}^q$ *gelte* $\nabla G(\vec{w})\vec{v} > 0$.

Dann existiert ein $\beta > 0$, *so daß gilt :*

$$G(\vec{w} - \alpha\vec{v}) < G(\vec{w}) \quad \forall\, \alpha \in (0, \beta).$$

Beweis :

Wegen der Differenzierbarkeit von G in $\vec{w}$ ist $\lim_{\alpha \to \infty} \frac{G(\vec{w} - \alpha\vec{v}) - G(\vec{w})}{\alpha} + \nabla G(\vec{w})\vec{v} = 0.$

Da $\vec{w} \in \mathring{D}$, gibt es ein $\beta > 0$, so daß gilt : $\vec{w} - \alpha\vec{v} \in D \quad \forall \alpha \in (0, \beta)$.

Aufgrund des Hilfsatzes kann β so klein gewählt werden, daß aus $\nabla G(\vec{w})\vec{v} > 0$

$$\frac{G(\vec{w} - \alpha\vec{v}) - G(\vec{w})}{\alpha} + \nabla G(\vec{w})\vec{v} < \nabla G(\vec{w})\vec{v} \quad \forall \alpha \in (0, \beta)$$ folgt.

$$\Rightarrow \quad \frac{G(\vec{w} - \alpha\vec{v}) - G(\vec{w})}{\alpha} < 0 \quad \forall \alpha \in (0, \beta)$$

$\Rightarrow$ Behauptung.

Satz

Das parallele Koordinatenabstiegsverfahren besitzt die Eigenschaft globaler Konvergenz, wenn die Lernraten eine gewisse Schranke nicht überschreiten.

Beweis :

Der MSE $F : \mathbb{R}^q \to \mathbb{R}$ ist differenzierbar in jedem beliebigen Vektor $\vec{w} \in \mathbb{R}^q$.

Sei $\vec{w} \in \mathbb{R}^q$ beliebig mit $\nabla F(\vec{w}) \neq 0$

und zusätzlich $\vec{v} = \left(\eta_1' \frac{\partial F}{\partial w_1}, ..., \eta_q' \frac{\partial F}{\partial w_q} \right)^T \in \mathbb{R}^q$ mit $\eta_1', ..., \eta_q' > 0$.

Dann gilt : $$\nabla F(\vec{w})\vec{v} = \left(\frac{\partial F}{\partial w_1}, ..., \frac{\partial F}{\partial w_q} \right) \left(\eta_1' \frac{\partial F}{\partial w_1}, ..., \eta_q' \frac{\partial F}{\partial w_q} \right)^T$$

$$= \sum_{i=1}^{q} \eta_i' \left(\frac{\partial F}{\partial w_i} \right)^2 > 0.$$

Damit sind die Voraussetzungen des Hilfssatzes erfüllt und damit existiert ein $\beta > 0$ mit

$$F(\vec{w} - \alpha\vec{v}) < F(\vec{w}) \quad \forall \alpha \in (0, \beta).$$

Für $\vec{v} = 0$ gilt sogar die Gleichheit. Insgesamt folgt die Behauptung, denn die η_i' müssen lediglich der folgenden Bedingung genügen: $\eta_i = \eta_i' \alpha_0$ für **i = 1,...,q** mit $\alpha_0 \in (0, \beta)$.

Der Vektor, den das Koordinatenabstiegsverfahren von $\vec{w}$ subtrahiert, lautet somit $\alpha_0 \vec{v}$.

Die $\delta - \delta$ - Regel

Die Modifikation der Gewichte erfolgt nach der oben beschriebenen Formel

$$w_i^{neu} = w_i^{alt} - \eta_i \frac{\partial F}{\partial w_i},$$

In Bezug auf den Zeitpunkt der Gewichtsänderung lautet die Formel

$$w_i(t+1) = w_i(t) - \eta_i(t+1) \cdot \frac{\partial F}{\partial w_i(t)}.$$

Die Herleitung für die Lernregel zur Lernratenmodifikation geschieht analog zur Herleitung der verallgemeinerten δ-Regel. Die Regel führt allerdings keinen Gradientenabstieg auf der Fehleroberfläche über dem Gewichtsraum durch.

Vielmehr versucht das Verfahren den Fehler des Netzes zu minimieren, indem es einen Gradientenabstieg auf der Fehleroberfläche über den Lernratenraum durchführt. Der Lernratenraum wird durch die $\eta_i = \eta_i' \alpha_0$ aufgespannt.

Aber auch diese Art der Modifikation der Lernrate basiert auf der bekannten Fehlerfunktion:

$$\eta_i(t+1) = \eta_i(t) + \triangle \eta_i(t)$$

$$\triangle \eta_i(t) = \gamma \frac{\partial F(t)\ \partial F(t-1)}{\partial w_i(t)\ \partial w_i(t-1)},$$

wobei $\gamma > 0$ die Schrittweite des Gradientenabstiegs im Lernratenraum ist.

Durch die Multiplikation zweier zeitlich aufeinanderfolgender Fehlerterme wird die Lernrate vergrößert, wenn beide Ableitungen dasselbe Vorzeichen haben. Unterscheiden sich die Vorzeichen, verringert sich die Lernrate.

Obwohl sich diese Implementierung genau an die Heuristik hält, treten bei genauerer Betrachtung einige Probleme auf:

In flachen Gebieten der Fehleroberfläche sind aufeinander folgende Ableitungen betragsmäßig klein. Ist ihr Betrag sogar < 1, verringert sich ihr Produkt noch weiter. Damit wird die Geschwindigkeit des Verfahrens verringert und nicht gesteigert. Abhilfe schafft dann nur noch eine Vergrößerung von γ. Allerdings würde diese Wahl von γ das folgende Problem forcieren:

In stark gekrümmten Gebieten können die Ableitungen betragsmäßig sehr große Werte annehmen, die sich zudem im Vorzeichen unterscheiden. Wird $\triangle \eta_i(t)$ zu groß, bewirkt die Lernratenmodifikation nicht nur eine Verringerung der Lernrate. Sie kann sogar dazu führen, daß die Lernrate negativ wird. Damit würde der zugehörige Gewichtsvektor in die falsche Richtung verschoben werden.

Eine gute Lernratenmodifikation bedarf also nicht nur einer guten Heuristik, vielmehr wird eine Steuerung benötigt, die die Sonderfälle behandelt. Die folgende Modifikation löst das Problem mit Hilfe zusätzlicher Parameter.

Die $\bar{\delta} - \delta$ - Regel

Auch bei dieser Lernregel wird mit Hilfe von individuellen, variablen Lernraten eine Gewichtsmodifikation durchgeführt.

Die Gewichtsänderung erfolgt wie bei der $\delta - \delta$ - Regel durch

$$w_i^{neu} = w_i^{alt} - \eta_i \frac{\partial F}{\partial w_i},$$

Die Lernraten werden nach folgender Formel modifiziert :

$$\eta_i(t+1) = \eta_i(t) + \triangle \eta_i(t)$$

Für die Veränderung der Lernrate gilt :

$$\triangle \eta_i(t) = \begin{cases} \kappa & , \text{falls } \bar{\delta}_i(t-1)\delta_i(t) > 0 \\ -\varphi \cdot \eta_i(t) & , \text{für } \bar{\delta}_i(t-1)\delta_i(t) < 0 \\ 0 & \text{sonst} \end{cases}$$

$$\text{mit} \quad \delta_i(t) = \frac{\partial F(t)}{\partial w_i(t)}.$$

$$\text{Dann gilt} \quad : \quad \bar{\delta}(t) = (1-\theta)\,\delta_i(t) + \theta\,\bar{\delta}_i(t-1)$$

$$= (1-\theta) \sum_{j=0}^{t} \theta^j \ \delta_i(t-j)$$

$w_i(t)$ ist ein Gewicht des Netzes zum Zeitpunkt t und $\eta_i(t)$ die zugehörige Lernrate.
κ, φ, θ sind Konstanten mit $\varphi, \theta \in [0,1]$ und $\kappa > 0$.

Wie bei der Momentum-Version verlieren die Ableitungen mit dem "Alter" an Bedeutung, da $\bar{\delta}(t)$ der exponentiell gewichtete Durchschnitt aller bis zum Zeitpunkt ***t*** berechneten Ableitungen darstellt.

Die $\delta - \delta$ - Regel realisiert die Heuristik wie folgt:

- Bei einer flachen Fehleroberfläche wird die Lernrate um eine Konstante κ vergrößert.
- Bei einer stark gekrümmten Fehleroberfläche wird die Lernrate um den φ-ten Anteil verringert.

Die Modifikation der Lernrate erfolgt asymetrisch.

Die Vergrößerung der Lernraten erfolgt linear. Damit wird verhindert, daß die Lernrate sprunghaft anwachsen kann.
Die Lernrate wird exponentiell verringert. Damit ist gewährleistet, daß immer $\eta_i > 0$ gilt und die Lernraten schnell verringert werden können.

Die Praxis hat bewiesen, daß mit Hilfe dieser Modifikationen die Schwächen der $\delta - \delta$ - Regel behoben wurden. Allerdings wird die Leistungsfähigkeit der Modifikation maßgeblich durch die Wahl von κ beeinflußt.

Wird κ zu klein gewählt, können die Lernraten nur langsam wachsen. In diesem Fall würden flache Plateaus sehr langsam durchlaufen werden.

Wird κ zu groß gewählt, wachsen die Lernraten zu schnell an und das Verfahren wird ungenau.

Die $\bar{\delta} - \delta$ - Regel kann durch zwei Modifikationen verbessert werden:

1. Die $\bar{\delta} - \delta$-Regel kann mit der Momentum-Version kombiniert werden.
2. Eine geeignete Steuerung für die Konstante κ kann eingeführt werden.

Beide Modifikationen lassen sich mit Hilfe von Fuzzy-Controlern realisieren.

Growing und Pruning Algorithmen

Die Entwicklung von Growing- und Pruning-Algorithmen liegt u.a. im Moving Target Problem begründet. Ein Beispiel für das Moving Target Problem ist der sogenannte Herden-Effekt:

Das zu lösende Problem besteht aus zwei Teilproblemen A und B, wobei A ein viel größeres Fehlersignal liefert als B. Die verdeckten Neuronen im Backpropagation-Netz versuchen zunächst, daß Fehlersignal zu minimieren, welches vom Teilproblem A erzeugt wird. Das Teilproblem B wird also lange ignoriert, bis das Problem A gelöst ist. Die Lösung von B bewirkt wiederum das Auftreten von Problem A. Durch die ständige Änderung des zu lösenden Teilproblems benötigt das Verfahren sehr lange, bis ein ansprechendes Gesammtergebniss erreicht wird.

Die einzige Lösung des Problems ist eine Aufteilung der Neuronen in den verborgenen Schichten. Da das Neuronale Netz das Problem lösen kann, wenn es nur lange genug trainiert wird, ist anzunehmen, daß es ebenfalls die Notwendigkeit zur Aufspaltung der inneren Einheiten erkennt.

Der Herden-Effekt kann vermieden werden, indem in jedem Trainigsschritt nur eine geringe Anzahl von Gewichten modifiziert werden. Growing Algorithmen arbeiten nach diesem Prinzip.

Cascade Correlation ist ein Growing-Algorithmus, der die Gewichte von nur einem Neuron im Netz modifiziert. Das Verfahren startet mit der durch die Problemstellung vorgegebnen Anzahl von Ein- und Ausgabeneuronen. Daraufhin wird ein verdecktes Neuron eingeführt und trainiert, bis keine Verringerung des Fehlers mehr zu erkennen ist. Wird das Netz um weitere Neuron erweitert, können die Eingabegewichte der zuvor eingeführten verdeckten Neuronen nicht mehr verändert werden.

Obwohl in jedem Schritt lediglich die Eingabegewichte eines einzigen verdeckten Neurons verändert werden können, ist das Verfahren bei vielen Problemen schneller als alle Varianten von Backpropagation. Dies ist allerdings leicht nachzuvollziehen, da die Ausgaben der verborgenen Neuronen bei hinzufügen von weiteren Neuronen konstant bleiben und das Moving Target Problem nahezu ausgeschaltet wird.

Ein anderer Ansatz wird von Pruning-Algorithmen verfolgt. Sie starten mit einem großen Netz und eliminieren dann die überflüssigen Einheiten. Detailierte Informationen können Seminarvorträgen und

Diplomarbeiten entnommen werden, die auf der WWW-Seite der Arbeitsgemeinschaft von Prof. Lippe zu finden sind.


Zurück zum letzten Kapitel