Einführung in Neuronale Netze
Backpropagation Learning

Herleitung des Gradientenabstiegs

Nachdem das Prinzip der Lernregel erläutert wurde, soll nun der Frage nach der Berechnung des negativen Gradienten nachgegangen werden.

Um die Berechnung von $-\nabla_{\vec{w}} F(\vec{w})$ zu vereinfachen, werden die partiellen Ableitungen der Fehlerfunktion nach den **q** Komponenten des Gewichtsvektors betrachtet.

Sei $\frac{\partial F}{\partial w_p}$ mit p $\in$ {1,..,q} die p-te partielle Ableitung der Fehlerfunktion. Für sie gilt :

$$\frac{\partial F(\vec{w})}{\partial w_p} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{\partial F_k(\vec{w})}{\partial w_p}.$$

In einem Backpropagation-Netz wird eine Kette von Funktionskompositionen berechnet. Daher ist die Anwendung der Kettenregel elementar für den Backpropagation-Algorithmus.

Schon bei der Betrachtung der Aktivierungsfunktion wird die Bedeutung der Kettenregel klar.

Die Aktivierung eines Neurons berechnet sich durch die gewichtete Summe der Ausgaben der Neuronen aus der Vorgängerschicht.

Es gilt für die Aktivierungsfunktion f_a bezüglich des Neurons $N_{h,i}$:

$$f_a \equiv A_{h,i} = \sum_{j=0}^{\#(U_{h-1})} o_{h-1,j}\ w_p\ .$$

Zu jedem p $\in$ {1,..,q} existiert ein Index (h,i,j) mit $w_{h,i,j} = w_p$, so daß gilt:

$$f_a \equiv A_{h,i} = \sum_{j=0}^{\#(U_{h-1})} o_{h-1,j}\ w_{h,i,j}\,.$$

$o_{h-1,j}$ bezeichnet die Ausgabe des j-ten Neurons der Vorgängerschicht.

Die Ausgabe des Neurons $N_{h,i}$ berechnet sich durch $o_{h,i} = s(A_{h,i})$, wobei s eine sigmoide Ausgabefunktion bezeichnet. Da die Ausgabe den Fehler F_k beinhaltet, hängt F_k funktional von $w_{h,i,j}$ ab.

Damit kommt die Kettenregel zur Anwendung. Es gilt:

$$\frac{\partial F_k(\vec{w})}{\partial w_p} \equiv \frac{\partial F_k(\vec{w})}{\partial w_{h,i,j}} = \frac{\partial F_k(\vec{w})}{\partial A_{h,i}} \cdot \frac{\partial A_{h,i}}{\partial w_{h,i,j}}\,.$$

Kürzt man $\frac{\partial F_k(\vec{w})}{\partial A_{h,i}}$ mit $\delta^k_{h,i}$ ab, ergibt sich daraus

$$\frac{\partial F_k(\vec{w})}{\partial w_p} = \delta^k_{h,i} \cdot \frac{\partial}{\partial w_{h,i,j}} \left(\sum_{r=0}^{\#(U_{h-1})} o^k_{h-1,r}\ w_{h,i,r} \right) = \delta^k_{h,i}\ o^k_{h-1,j}\,,$$

da für die Ableitungen der Aktivierungsfunktion gilt :

$$\frac{\partial (o^k_{h-1,r}\ w_{h,i,r})}{\partial w_{h,i,j}} = \begin{cases} 0 & \text{für } r \neq j \\ o^k_{h-1,j} & \text{für } r = j \end{cases}.$$

Damit lautet die partielle Ableitung $\frac{\partial F(\vec{w})}{\partial w_p} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \delta^k_{h,i}\ o^k_{h-1,j}\,.$

Es stellt sich natürlich noch die Frage nach der Berechnung von $\delta^k_{h,i}$ mit $\delta^k_{h,i} = \frac{\partial F_k(\vec{w})}{\partial A_{h,i}}$.

Da sich die Ausgabefunktionen der Ausgabeschicht und der verborgenen Schichten unterscheiden, muss auch $\delta^k_{h,i}$ unterschiedlich

berechnet werden.

1. Fall: h = H - 1

In der Ausgabeschicht entspricht die Ausgabefunktion der Identität. Somit entspricht die Ausgabe des Neurons dessen Aktivierung.

Es gilt also $o_{h,i} = A_{h,i}$ und damit $\delta^k_{h,i} = \frac{\partial F_k(\vec{w})}{\partial o_{h,i}}$. Daraus folgt :

$$\frac{\partial F_k(\vec{w})}{\partial o_{h,i}} = \frac{\partial}{\partial o_{h,i}} \left(\sum_{r=1}^{\#(U_{h-1})} (y^k_r - o^k_{h,i})^2 \right) = -2\,(y^k_i - o^k_{h,i}),$$

wobei y die korrekte Ausgabe repräsentiert.

Damit ergibt sich für die Ausgabeschicht : $\delta^k_{h,i} = -2\,(y^k_i - o^k_{h,i})$.

2. Fall : h < H - 1

Wenn die h-te Schicht die verborgene Schicht ist, ist die Ausgabefunktion sigmoid. Daher hängt F_k funktional über $o_{h,i}$ von $A_{h+1,r}$ ab.

$$\delta^k_{h,i} = \frac{\partial F_k(\vec{w})}{\partial A_{h,i}} = \frac{\partial F_k(\vec{w})}{\partial o_{h,i}} \cdot \frac{\partial o_{h,i}}{\partial A_{h,i}} = \frac{\partial F_k(\vec{w})}{\partial o_{h,i}} \cdot s'(A_{h,i}).$$

Da F_k über $A_{h+1,r}$ für $r = 1, \ldots, \#(U_{h+1})$ funktional von $o_{h,i}$ abhängt, gilt:

$$\frac{\partial F_k(\vec{w})}{\partial o_{h,i}} = \sum_{r=1}^{\#(U_{h+1})} \frac{\partial F_k}{\partial A_{h+1,r}} \cdot \frac{\partial A_{h+1,r}}{\partial o_{h,i}}.$$

Damit ergibt sich für die verborgenen Schichten:

$$\delta_{h,i}^k = s'(A_{h,i}) \sum_{r=1}^{\#(U_{h+1})} \delta_{h+1,r}^k \cdot w_{h+1,r,i} \cdot$$

Zusammenfassung

Die Ableitung der Fehlerfunktion nach dem Gewicht $w_{h,i,j}$ lautet:

$$\frac{\partial F(\vec{w})}{\partial w_{h,i,j}} = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} \delta_{h,i}^k \, o_{h-1,j}^k \cdot$$

Da sich sich die Ausgabefunktionen der Ausgabeneuronen und verborgenen Neuronen unterscheiden, differiert auch die Berechnung der δ.

Für die Ausgabeschicht gilt:

$$\delta_{h,i}^k = -2 \, (y_i^k - o_{h,i}^k) \cdot$$

Für die verborgenen Neuronen gilt:

$$\delta_{h,i}^k = s'(A_{h,i}) \sum_{r=1}^{\#(U_{h+1})} \delta_{h+1,r}^k \cdot w_{h+1,r,i} \cdot$$

Zeitpunkt und Art der Gewichtsveränderung

Hinsichtlich des Zeitpunktes der Anpassung der Gewichte im Backward Pass sind zwei Variationen möglich:

ONLINE	• Beim online-Training wird nach der **jeder** Präsentation eines Trainingsmusters eine Gewichtsmodifikation durchgeführt.

	• Die Lernregel lautet: $\vec{w}^{neu} = \vec{w}^{alt} - \eta \nabla_{\vec{w}} F(\vec{w})$.
OFFLINE	• Beim offline- oder auch batch-Training erfolgt die Modifikation der Gewichte erst nach der Präsentation einer **zuvor festgelegten Anzahl** der Trainingsmuster, welche Batchsize genannt wird. • Die Lernregel lautet: $w_{h,i,j}^{neu} = w_{h,i,j}^{alt} - \eta \cdot \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \delta_{h,i}^{k} o_{h-1,j}^{k}$.

Diese Lernregeln werden in Anlehnung an die δ-Regel als verallgemeinerte δ-Regel oder auch als Verallgemeinerung der Widrow - Hoff - Regel bezeichnet. Der in den Lernregeln auftretende Faktor $\eta > 0$ wird Lernrate genannt und steuert, um welchen Anteil von $-\nabla_{\vec{w}} F(\vec{w})$ der Vektor $\vec{w}$ verschoben wird.

Das online-Training wird bei praktischen Simulationen benutzt. Obwohl das online-Training weniger speicherintensiv ist, und neue Trainingsmuster problemlos nachtrainiert werden können, empfielt es sich dennoch, das offline-Training zum Training von Backpropagation Netzen zu nutzen.

Das offline-Training verhindert, daß die Änderung, die durch das eine Muster hervorgerufen wurde, durch das nächste Muster wieder rückgängig gemacht wird. Außerdem wird Annäherung an der Gradienten umso besser, je größer die Batchsize ist.

Unter einem Trainingszyklus oder sweep versteht man die vollständige Präsentation aller Trainingsmuster. In jedem sweep muß die Reihenfolge der Trainingsmuster verändert werden, da sonst die Gefahr besteht, daß das Backpropagation-Netz die Reihenfolge der Muster lernt. Damit würde es keine generelle Abbildungsvorschrift finden.

Übungsaufgabe

Das folgende Applet erläutert die Gewichtsmodifikation durch den Backpropagation-Algorithmus.

Bevor sie den Gradientenabstieg starten können, muß der Startpunkt des Verfahrens bestimmt werden. Dies geschieht, indem durch Betätigung des Buttons Zufallsinitialisierung ein Punkt im Gewichtsraum zufällig bestimmt wird. Sollten die Ergebnisse des Gradientenabstiegs sie nicht zufriedenstellen, sollten sie einen anderen Punkt wählen, an dem mit dem Gradientenabstieg begonnen wird. Sollten sich die Ergebnisse immer noch nicht verbessern, können sie die Anzahl der Neuronen in der verborgenen Schicht vergrößern, um die Leistungsfähigkeit des Netzes zu erhöhen.

Durch die Betätigung von Backpropagation wird der Gradientenabstieg gestartet. Mit Hilfe von

Folgendes Muster und Vorheriges Muster die verschiedenen Eingabemuster und die durch sie erzeugten Ausgaben betrachtet werden.

Wird der Forward- bzw. Backward-Pass zu schnell oder zu langsam dargestellt, kann durch die Änderung des DELAY die gewählte Geschwindigkeit in der Darstellung beeinflußt werden. Probleme in der Darstellung des Applets können durch einen Bildschirmrefresh behoben werden.

Die unterschiedliche Aktivierung der Neuronen und Belegung der Gewichte wird durch den Einsatz von Farben angedeutet. Je heller das Grün der Eingabeneuronen und der verborgenen Neuronen ist, umso größer ist die Ausgabe des entsprechenden Neurons. Rote Einfärbung der Ausgabeneuronen deuten einen großen Anteil am Netzwerkfehler an. Ein Übergang von rot nach gelb signalisiert eine Verringerung des Fehlersignals. Positive Gewichte werden blau, negative Gewichte rot eingefärbt.

Um an Informationen über ein spezielles Neuron und ihr Fehlersignal zu gelangen, klicken sie auf das gewünschte Neuron. In dem Textfield werden daraufhin alle Wertebelegungen und ihre Entwicklung während der Gewichtsmodifikation angezeigt. Der bei der Berechnung der Fehlersignale der Ausgabeschicht fehlende Faktor -2 wird durch die Lernrate kompensiert.

Zurück zum letzten Kapitel   Zum nächsten Kapitel