Prof. Dr. Gerd Stumme, Robert Jäschke
Fachgebiet Wissensverarbeitung

12.12.2007

6. Übung „Künstliche Intelligenz“

Wintersemester 2007/2008

Neuronale Netze

1. Gegeben sei die folgende logische Funktion:

x_1	x_2	t
0	0	1
0	1	1
1	0	1
1	1	0

   a) Lernen Sie die Funktion mit einem einfachen Perzeptron. Verwenden Sie dabei die Sprungfunktion von Folie 13 (Kapitel 5). Passen Sie die Gewichte gemäß der Delta-Regel an. Die Anfangswerte seien $x_0 = 1, w = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$ (d. h. $s = -0.5$) sowie $\eta = 0.5$.

Es ergibt sich folgender Ablauf:

x			t	w			a	s	o
1	0	0	1	0.5	0.5	0.5	0.5	-0.5	1
1	0	1	1	0.5	0.5	0.5	1	-0.5	1
1	1	0	1	0.5	0.5	0.5	1	-0.5	1
1	1	1	0	0.5	0.5	0.5	1.5	-0.5	1
1	0	0	1	0	0	0	0	0	1
1	0	1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1
1	0	0	1	-0.5	-0.5	-0.5	-0.5	0.5	0
1	0	1	1	0	-0.5	-0.5	-0.5	0	0
1	1	0	1	0.5	-0.5	0	0	-0.5	1
1	1	1	0	0.5	-0.5	0	0	-0.5	1
1	0	0	1	0	-1	-0.5	0	0	1
1	0	1	1	0	-1	-0.5	-0.5	0	0
1	1	0	1	0.5	-1	0	-0.5	-0.5	1
1	1	1	0	0.5	-1	0	-0.5	-0.5	1
1	0	0	1	0	-1.5	-0.5	0	0	1
1	0	1	1	0	-1.5	-0.5	-0.5	0	0
1	1	0	1	0.5	-1.5	0	-1	-0.5	0
1	1	1	0	1	-1	0	0	-1	1
1	0	0	1	0.5	-1.5	-0.5	0.5	-0.5	1
1	0	1	1	0.5	-1.5	-0.5	0	-0.5	1
1	1	0	1	0.5	-1.5	-0.5	-1	-0.5	0
1	1	1	0	1	-1	-0.5	-0.5	-1	1
1	0	0	1	0.5	-1.5	-1	0.5	-0.5	1
1	0	1	1	0.5	-1.5	-1	-0.5	-0.5	1
1	1	0	1	0.5	-1.5	-1	-1	-0.5	0
1	1	1	0	1	-1	-1	-1	-1	1
1	0	0	1	0.5	-1.5	-1.5	0.5	-0.5	1
1	0	1	1	0.5	-1.5	-1.5	-1	-0.5	0
1	1	0	1	1	-1.5	-1	-0.5	-1	1
1	1	1	0	1	-1.5	-1	-1.5	-1	0
1	0	0	1	1	-1.5	-1	1	-1	1
1	0	1	1	1	-1.5	-1	0	-1	1
1	1	0	1	1	-1.5	-1	-0.5	-1	1
1	1	1	0	1	-1.5	-1	-1.5	-1	0

Die Gewichte für das trainierte Perzeptron sind also $w = \begin{pmatrix} 1 \\ -1.5 \\ -1 \end{pmatrix}$.

b) Um welchen logischen Operator handelt es sich bei der obigen Funktion?

Es handelt sich um den NAND-Operator.

2. Lernen der Gewichte mit der Delta Regel

a) Erklären Sie das Prinzip der Delta-Regel!

Die Gewichte des Neuronalen Netzes werden so adaptiert, dass bei mehrmaliger Iteration der Delta-Regel die richtige Antwort gegeben wird (Antwort sei z. B. die Verknüpfung von Binärwerten durch einen NAND-Operator). Das funktioniert so: Es wird der Output desjenigen Neurons verstärkt, welches die richtige Antwort gegeben hat. Der Output derjenigen Neuronen, welche den falschen Wert liefern (falscher Ist-Wert) wird geschwächt. Die Schwächung bzw. die Stärkung des Outputs erfolgt durch die Adaption der dazugehörigen Gewichte.

b) Wie wirkt sich die Veränderung der Lernrate auf die Adaptivität der Gewichte aus?

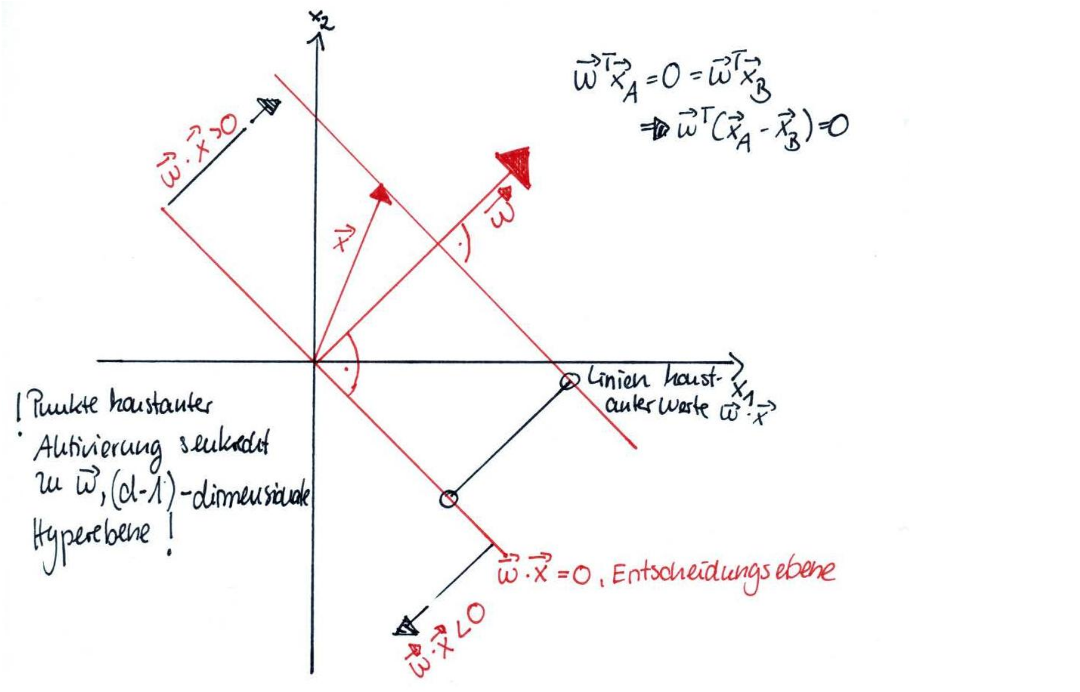
Allgemein gilt: Die Wahl einer kleinen Lernrate läßt das Neuronale Netz „langsam lernen", d. h. die Gewichte *w* werden in kleinen Schritten adaptiert. Umgekehrt werden die Gewichte bei einer großen Lernrate schneller adaptiert.

c) Wieso ist ein nichtlinearer Zusammenhang zwischen der Eingabe und der Ausgabe mit einem einstufigen Perzeptron nicht erlernbar?

vgl. Folie 31.

d) Diskutieren Sie den Begriff der *Hyperebene* im Zusammenhang mit dem Begriff des *Gewichtsvektors* (veranschaulichen Sie sich die beiden Begriffe graphisch)!

Der Gewichtsvektor steht senkrecht zur Entscheidungsebene, wobei die Hyperebene als Neuronenaktivität aufgefasst werden kann.



3. Das Multi-Lagen-Perzeptron

a) Erläutern Sie das Multi-Lagen-Perzeptron (MLP) und das Prinzip des Backpropagation-Algorithmus!

Ein MLP ist ein mehrlagiges Feed-Forward-Netz, d. h. es gibt keine Schleifen. Hierbei besitzt das Neuronale Netz sogenannte Hidden-Layers, eine Eingabeschicht und eine Ausgabeschicht. Das Prinzip des Backpropagation-Algorithmus besteht darin, den aktuellen Fehler (gewollter Output abzüglich des tatsächlichen Output) vom Output bis zum Input über jedes Gewicht zurückzuverfolgen. Für jedes Gewicht wird sein Anteil zum Fehler berechnet und danach mit der Gradientenmethode das Gewicht vergrößert oder verkleinert.

b) Gegeben sei folgendes Neuronales Netz (siehe Abb. 1) mit einer sigmoiden Outputfunktion (mit Steigung 1). Die Ableitung der Outputfunktion θ ist $\theta'(x) = \theta(x)(1 - \theta(x))$. Der Gebrauch einer sigmoiden Aktivierungsfunktion ist notwendig, da diese differenzierbar ist.

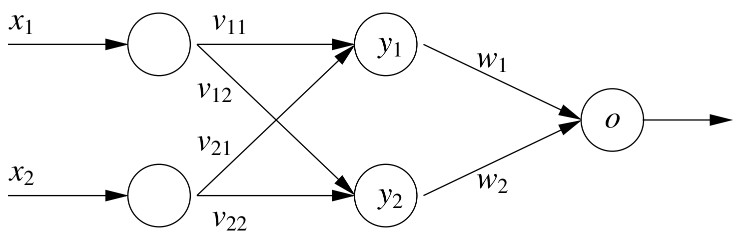


Abbildung 1: Ein Multi-Lagen-Perzeptron

Nutzen Sie nun die XOR-Funktion aus Abbildung 1 und berechnen Sie durch Backpropagation die Gewichtsänderungen, indem Sie die letzte Zeile als Input an das Netz anlegen. Die Gewichte seien wie folgt initialisiert worden: $v = \begin{pmatrix} 0.5 & 0.75 \\ 0.25 & 0.25 \end{pmatrix}$, $w = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$.

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	0

Tabelle 1: XOR-Verschaltung

Lösung:

Für den Gradientenabstieg wird eine Aktivierungsfunktion benötigt, welche differenzierbar ist. Dieses kann z. B. eine sigmoide Aktivierungsfunktion sein.
Für die Berechnung des Fehlergradienten $\delta F/\delta w_i$ (siehe Folie 40), muss zunächst y sowie der Output o berechnet werden:

$$
\begin{aligned}
y = \theta(vx) &= \theta\left(\begin{pmatrix} 0.5 & 0.75 \\ 0.25 & 0.25 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) = \theta\left(\begin{pmatrix} 1.25 \\ 0.5 \end{pmatrix}\right) = \begin{pmatrix} 0.78 \\ 0.62 \end{pmatrix} \\
o = \theta(w^T y) &= \theta\left(\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}^T \begin{pmatrix} 0.78 \\ 0.62 \end{pmatrix}\right) = \theta(0.7) = 0.67
\end{aligned}
$$

Weiter berechnen wir:

$$
\begin{aligned}
2(o-t)\theta'(w^T y) &= 2(o-t)\theta(w^t y)(1-\theta(w^T y)) \\
&= 2(o-t)o(1-o) \\
&= 2 \cdot (0.67 - 0) \cdot 0.67 \cdot (1 - 0.67) \\
&= 0.3
\end{aligned}
$$

Es ergibt sich für die Fehlergradienten $\dfrac{\delta F}{\delta w_i}$ (und somit für die Korrekturfaktoren Δw_i):

$$
\Delta w := \frac{\delta F}{\delta w} = \begin{pmatrix} \frac{\delta F}{\delta w_1} \\ \frac{\delta F}{\delta w_2} \end{pmatrix} = 2(o-t)\theta'(wy^T)y = 0.3\begin{pmatrix} 0.78 \\ 0.62 \end{pmatrix} = \begin{pmatrix} 0.23 \\ 0.18 \end{pmatrix}
$$

und damit für die neuen Gewichte:

$$
w := w - \Delta w = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.23 \\ 0.18 \end{pmatrix} = \begin{pmatrix} 0.27 \\ 0.32 \end{pmatrix}
$$

Nun wird der Gradient $\dfrac{\delta F}{\delta v_{ij}}$ berechnet. Wir erhalten diesen als

$$
\begin{aligned}
\frac{\delta F}{\delta v_{ij}} &= 2(o-t)\theta'(w^T y)w_i\theta'(v_i x)x_j \\
&= 0.3 w_i \theta'(v_i x)x_j \\
&= 0.3 w_i \theta(v_i x)(1-\theta(v_i x))x_j \\
&= 0.3 w_i y_i (1-y_i)x_j
\end{aligned}
$$

(siehe Kapitel 5, Folie 41).
Damit ergibt sich für $\dfrac{\delta F}{\delta v_{ij}}$:

$$
\begin{aligned}
\Delta v_{11} &= 0.3 \cdot w_1 \cdot 0.78 \cdot (1-0.78) \cdot x_1 = 0.3 \cdot 0.27 \cdot 0.78 \cdot 0.22 \cdot 1 = 0.0138 \\
\Delta v_{12} &= 0.3 \cdot w_1 \cdot 0.78 \cdot (1-0.78) \cdot x_2 = 0.3 \cdot 0.27 \cdot 0.78 \cdot 0.22 \cdot 1 = 0.0138 \\
\Delta v_{21} &= 0.3 \cdot w_2 \cdot 0.62 \cdot (1-0.62) \cdot x_1 = 0.3 \cdot 0.32 \cdot 0.62 \cdot 0.38 \cdot 1 = 0.0220 \\
\Delta v_{22} &= 0.3 \cdot w_2 \cdot 0.62 \cdot (1-0.62) \cdot x_2 = 0.3 \cdot 0.32 \cdot 0.62 \cdot 0.38 \cdot 1 = 0.0220
\end{aligned}
$$

und somit

$$
\begin{aligned}
v_{11} &:= v_{11} - \Delta v_{11} = 0.5 + 0.0138 = 0.4862 \\
v_{12} &:= v_{12} - \Delta v_{12} = 0.75 + 0.0138 = 0.7362 \\
v_{21} &:= v_{21} - \Delta v_{21} = 0.25 + 0.0220 = 0.2280 \\
v_{22} &:= v_{22} - \Delta v_{22} = 0.25 + 0.0220 = 0.2280
\end{aligned}
$$

sowie

$$
\begin{aligned}
y &= \theta(vx) = \theta\left(\begin{pmatrix} 0.4862 & 0.7362 \\ 0.2280 & 0.2280 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) = \theta\left(\begin{pmatrix} 1.2223 \\ 0.4561 \end{pmatrix}\right) = \begin{pmatrix} 0.7725 \\ 0.6121 \end{pmatrix} \\
o &= \theta(w^T y) = \theta\left(\begin{pmatrix} 0.27 \\ 0.32 \end{pmatrix}^T \begin{pmatrix} 0.7725 \\ 0.6121 \end{pmatrix}\right) = \theta(0.4015) = 0.5990
\end{aligned}
$$

Neuer Output bei $(1, 1)$ ist damit 0.5990 und nicht mehr 0.67. Die Anpassung der Gewichte hat also die Ausgabe weiter in Richtung t verschoben.