Einführung in die Neuroinformatik

Tim Luchterhand, Paul Nykiel (Gruppe P)

8. Mai 2018

1 Lernregeln

1.1

$$
\begin{aligned}
\frac{\partial E}{\partial w} &= \frac{\partial}{\partial w}\left(\frac{1}{2}\sum_{\mu=1}^{M}\left(T^\mu - f(w \cdot x^\mu + b)\right)^2\right) \\
&= \sum_{\mu=1}^{M}\frac{1}{2} \cdot 2(T^\mu - f(w \cdot x^\mu + b)) \cdot -f'(w \cdot x^\mu + b) \cdot x^\mu \\
&= -\sum_{\mu=1}^{M}(T^\mu - f(w \cdot x^\mu + b)) \cdot f'(w \cdot x^\mu + b) \cdot x^\mu
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial b} &= \frac{\partial}{\partial b}\left(\frac{1}{2}\sum_{\mu=1}^{M}\left(T^\mu - f(w \cdot x^\mu + b)\right)^2\right) \\
&= \sum_{\mu=1}^{M}\frac{1}{2} \cdot 2(T^\mu - f(w \cdot x^\mu + b)) \cdot -f'(w \cdot x^\mu + b) \\
&= -\sum_{\mu=1}^{M}(T^\mu - f(w \cdot x^\mu + b)) \cdot f'(w \cdot x^\mu + b)
\end{aligned}
$$

1.2

(a) Inkrementelle Version:

$$
\begin{aligned}
w(t+1) &= w(t) + \eta \cdot (T^\mu - f(w \cdot x^\mu + b)) \cdot f'(w \cdot x^\mu + b) \cdot x^\mu \\
b(t+1) &= b(t) + \eta \cdot (T^\mu - f(w \cdot x^\mu + b)) \cdot f'(w \cdot x^\mu + b)
\end{aligned}
$$

(b) Batch Version:

$$\begin{aligned}
w(t+1) &= w(t) + \eta \cdot \sum_{\mu=1}^{M} (T^\mu - f(w \cdot x^\mu + b)) \cdot f'(w \cdot x^\mu + b) \cdot x^\mu \\
b(t+1) &= b(t) + \eta \cdot \sum_{\mu=1}^{M} (T^\mu - f(w \cdot x^\mu + b)) \cdot f'(w \cdot x^\mu + b)
\end{aligned}$$

1.3

(a) Ableitung der Transferfunktion:

$$f'(x) = -\frac{e^{-x}}{1 + e^{-x^2}}$$

Werte des Gradienten berechenen;

$$\begin{aligned}
\frac{\partial E}{\partial w}(-1,3) &= -\sum_{\mu=1}^{4} (T^\mu - f(-1 \cdot x^\mu + 3)) \cdot f'(-1 \cdot x^\mu + 3) \cdot x^\mu \\
&= -\big(((0 - f(4)) \cdot f'(4) \cdot -1) + 0 + ((0 - f(2)) \cdot f'(2) \cdot 1) + ((0 - f(1)) \cdot f'(1) \cdot 2)\big) \\
&= -\big(f(4)f'(4) - f(2)f'(2) - 2f(1)f'(1)\big) \\
&= f(2)f'(2) + 2f(1)f'(1) - f(4)f'(4) \\
&= -0.36 \\
\frac{\partial E}{\partial b}(-1,3) &= -\sum_{\mu=1}^{4} (T^\mu - f(-1 \cdot x^\mu + 3)) \cdot f'(-1 \cdot x^\mu + 3) \\
&= -\big(((0 - f(4)) \cdot f'(4)) + 0 + ((0 - f(2)) \cdot f'(2)) + ((0 - f(1)) \cdot f'(1))\big) \\
&= f(4)f'(4) + f(2)f'(2) + f(1)f'(1) \\
&= -0.0041
\end{aligned}$$

$$\Rightarrow \nabla E(w(0), b(0)) = \begin{pmatrix} -0.36 \\ -0.0041 \end{pmatrix}$$

(b) dsd

(c)

$$\begin{aligned}
w(1) &= w(0) + \eta \cdot \frac{\partial E}{\partial w}(w(0), b(0)) \\
&= -1 + 0.8 \cdot -0.36 \\
&= -1.288 \\
b(1) &= b(0) + \eta \cdot \frac{\partial E}{\partial b}(w(0), b(0)) \\
&= 3 + 0.8 \cdot -0.0041 \\
&= 2.997
\end{aligned}$$

(d) Der Gradientenabstieg findet nicht das globale Minimum sondern nur ein lokales Minimum, dort ist der Gradient aber ebenfalls $\vec{0}$.

1.4

(a) Pfad 1 ist vermutlich durch die Batch-Lernregel entstanden, Pfad 2 durch die inkrementelle Lernregel. Da bei der Batch-Lernregel der Gradient aus vielen Trainingssamples berechnet wird und dadurch eine Art Durchschnitt berechnet wird ist der Gradient genauer und springt weniger. Bei der inkrementellen Lernregel hingegen ist der Gradient eher ungenau, da er immer nur von einem Trainingssample berechnet wird, daher springt Pfad 2 mehr.

(b) Es ist möglich das beim inkrementellen Lernen kein globales Minimum gefunden wird, sondern nur ein Minimum welches für eine bestimmte Teilmenge der Trainingssamples gut passt.

1.5

Bei zu großer Lernrate kann die Minimierung der Fehlerfunktion zu einer Art überschwingen oder sogar zu einer Oszillation führen. Da der Pfad immer zu weit „springt“ erreicht er somit nie das Minimum.

1.6

Es wird versucht die Übertragungsfunktion des Neurons so anzupassen, das sie Möglichst nah an allen Punkten ist. Die Funktion hat jedoch immer eine Sigmoid-Funktion und lässt sich durch das Training nur Verschieben sowie Strecken/Stauchen. Da die Funktion aber streng monoton ist kann sie so nie alle Punkte optimal annähern.