1 Feed-Forward Berechnung

$$z_0^l = b^l + \sum_{j=0}^{n^{l-1}} a_j^{l-1} \cdot w_{j,0}^l$$

$$\vec{a}^l = W^l \cdot \vec{z}^{l-1} + \vec{b}^l$$

2 Backpropagation

2.1 Verlustfunktion

$$E_j = \frac{1}{2}(y_j - a_j^L)^2 \qquad E = \frac{1}{2}\sum_j (y_j - a_j^L)^2$$

$$\frac{dE_j}{da_j^L} = (a_j^L - y_j)$$

2.2 Lernvorgang

Anpassen der Gewichts- und BiasNeuronen:

$$w^l \to w^l - \frac{\eta}{N} \cdot \sum_{k=0}^{N} \frac{\partial E^k}{\partial w^l}$$

$\eta =$ Lernrate $\qquad N =$ Anzahl der Trainingsbeispiele $\qquad k =$ Trainingsbeispiel

2.3 sigmoid Funktion

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

$$\frac{d}{dx}\sigma(x) = \sigma(x) \cdot (1 - \sigma(x))$$

2.4 Fehler

$$\delta_j^l = \frac{\partial E}{\partial z_j^l} \qquad \frac{\partial E}{\partial w_{j,i}^l} = \delta_i^l \cdot a_j^{l-1} \qquad \frac{\partial E}{\partial b_j^l} = \delta_j^l$$

Berechnung der letzten Schicht:

$$\delta_j^L = (a_j^L - y_j) \cdot \sigma(z_j^L) \cdot (1 - \sigma(z_j^L))$$

$$\vec{\delta^L} = (\vec{z^L} - \vec{y^L}) \odot \sigma'(\vec{a^L})$$

Berechnung der restlichen Schichten:

$$\delta_j^l = [\sum_{i=0}^{n^{l+1}} \delta_i^{l+1} \cdot w_{j,i}^{l+1}] \cdot \sigma'(z_j^l)$$

$$\vec{\delta^l} = (W^{l+1})^T \cdot \vec{\delta^{l+1}} \odot \sigma'(\vec{z^l})$$

3 Batch-weise Berechnung mit Matrizen

3.1 Feed-Forward

$$A^l = \begin{pmatrix} a_0^{l,0} & a_0^{l,1} & ... & a_0^{l,k} \\ a_1^{l,0} & a_1^{l,1} & ... & a_1^{l,k} \\ ... & ... & ... & ... \\ a_j^{l,0} & a_j^{l,1} & ... & a_j^{l,k} \end{pmatrix} \qquad\qquad k : \text{index des Datensatz}$$

Berechnung:

$$A^l = \sigma(W^l \cdot A^{[l-1]} + \vec{b^l})$$

Addition des vektors wie folgt:

$$\begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_2 \end{pmatrix} + \begin{pmatrix} d \\ e \\ f \end{pmatrix} = \begin{pmatrix} a_0 + d & a_1 + d & a_2 + d \\ b_0 + e & b_1 + e & b_2 + e \\ c_0 + f & c_1 + f & c_2 + f \end{pmatrix}$$

3.2 Backpropagation

$$[\delta^l] = \begin{pmatrix} \delta_0^{l,0} & \delta_0^{l,1} & ... & \delta_0^{l,k} \\ \delta_1^{l,0} & \delta_1^{l,1} & ... & \delta_1^{l,k} \\ ... & ... & ... & ... \\ \delta_j^{l,0} & \delta_j^{l,1} & ... & \delta_j^{l,k} \end{pmatrix}$$

$$[\delta^L] = (Z^L - Y^L) \odot \sigma'(A^L)$$

$$[\delta^l] = (W^{l+1})^T \cdot [\delta^{l+1}] \odot \sigma'(A^l)$$

4 Berechnung der Änderungsraten

$$\frac{\partial E}{B^l} = [\delta^l]$$

$$\frac{\partial E}{W^l} = [\delta^l] \cdot (A^{l-1})^T$$

5 Quellen

Michael Nielsen - Neural Networks and Deep Learning(Algorithmus)
Michael Kipp - Neurale Netze und Deep Learning (Algorithmus)
Sudeep Raja - A Derivation of Backpropagation in Matrix Form (Prüfen der Batch-weisen Berechnung)