These are the slides of the lecture

Pattern Recognition
Winter term 2011/12
Friedrich-Alexander University of Erlangen-Nuremberg.

These slides are for your personal usage only in order to prepare for the examination.

Erlangen, April 25, 2018
Dr.-Ing. Stefan Steidl

Pattern Recognition (PR)

Winter Term 2011/12

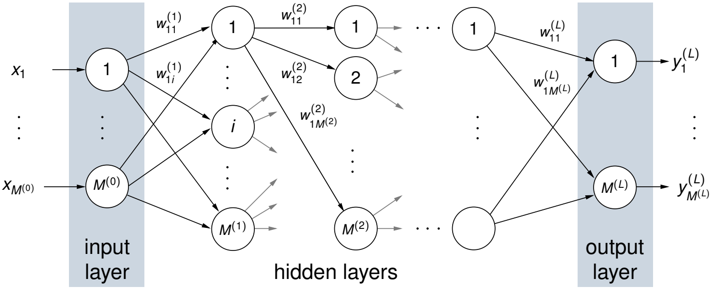
Stefan Steidl
Computer Science Dept. 5
(Pattern Recognition)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Multi-Layer Perceptrons

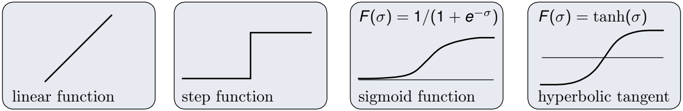
Multi-Layer Perceptrons

Topology



Multi-Layer Perceptrons (cont.)

Activation Functions



$$\mathrm{net}_j^{(l)} = \sum_{i=1}^{M^{(l-1)}} y_i^{(l-1)} w_{ij}^{(l)} - w_{0j}^{(l)}$$

$$y_j^{(l)} = f(\mathrm{net}_j^{(l)})$$

Backpropagation Algorithm

Supervised Learning Algorithm

- Gradient descent to adjust the weights reducing the training error ε:

$$\Delta w_{ij}^{(l)} = -\eta \frac{\partial \varepsilon}{\partial w_{ij}^{(l)}}$$

- Typical error function: mean squared error

$$\varepsilon_{\text{MSE}}(\boldsymbol{w}) = \frac{1}{2} \sum_{k=1}^{M^{(L)}} (t_k - y_k^{(L)})^2$$

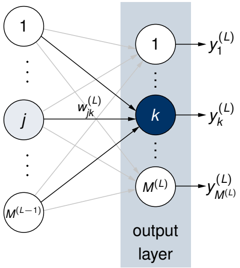
Backpropagation Algorithm (cont.)

Adjusting the weights $w_{jk}^{(L)}$ of the output layer

$$\frac{\partial \varepsilon_{\text{MSE}}}{\partial w_{jk}^{(L)}} = \frac{\partial \varepsilon_{\text{MSE}}}{\partial \text{net}_k^{(L)}} \cdot \frac{\partial \text{net}_k^{(L)}}{\partial w_{jk}^{(L)}} = -\delta_k^{(L)} \cdot y_j^{(L-1)}$$

The *sensitivity* $\delta_k^{(L)}$:

$$\begin{aligned}\delta_k^{(L)} &= -\frac{\partial \varepsilon_{\text{MSE}}}{\partial \text{net}_k^{(L)}} = -\frac{\partial \varepsilon_{\text{MSE}}}{\partial y_k^{(L)}} \cdot \frac{\partial y_k^{(L)}}{\partial \text{net}_k^{(L)}} \\ &= (t_k - y_k^{(L)})\, f'(\text{net}_k^{(L)})\end{aligned}$$



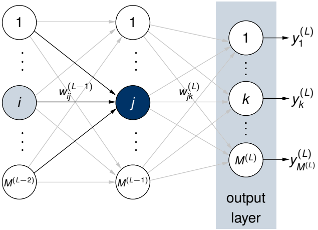
Backpropagation Algorithm (cont.)

Adjusting the weights $w_{jk}^{(L)}$ of the hidden layers

- Desired output values for the hidden layers are not known.
- For the weights $w_{ij}^{(L-1)}$ of the last hidden layer:

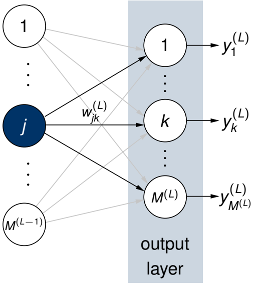
$$\frac{\partial \varepsilon_{\mathrm{MSE}}}{\partial w_{ij}^{(L-1)}} = \frac{\partial \varepsilon_{\mathrm{MSE}}}{\partial y_j^{(L-1)}} \cdot \frac{\partial y_j^{(L-1)}}{\partial \mathrm{net}_j^{(L-1)}} \cdot \frac{\partial \mathrm{net}_j^{(L-1)}}{\partial w_{ij}^{(L-1)}}$$

$$= \frac{\partial \varepsilon_{\mathrm{MSE}}}{\partial y_j^{(L-1)}} \cdot f'(\mathrm{net}_j^{(L-1)}) \cdot y_i^{(L-2)}$$



Backpropagation Algorithm (cont.)

$$
\begin{aligned}
\frac{\partial \varepsilon_{\text{MSE}}}{\partial y_j^{(L-1)}} &= \frac{\partial}{\partial y_j^{(L-1)}} \left[\frac{1}{2} \sum_{k=1}^{M^{(L)}} (t_k - y_k^{(L)})^2 \right] \\
&= -\sum_{k=1}^{M^{(L)}} (t_k - y_k^{(L)}) \frac{\partial y_k^{(L)}}{\partial y_j^{(L-1)}} \\
&= -\sum_{k=1}^{M^{(L)}} (t_k - y_k^{(L)}) \frac{\partial y_k^{(L)}}{\partial \text{net}_k^{(L)}} \cdot \frac{\partial \text{net}_k^{(L)}}{\partial y_j^{(L-1)}} \\
&= -\sum_{k=1}^{M^{(L)}} (t_k - y_k^{(L)})\, f'(\text{net}_k^{(L)})\, w_{jk}^{(L)} \\
&= -\sum_{k=1}^{M^{(L)}} \delta_k^{(L)} w_{jk}^{(L)}
\end{aligned}
$$



Backpropagation Algorithm (cont.)

Sensivity $\delta_j^{(l)}$ for any hidden layer l, $0 < l < L$

$$\delta_j^{(l)} = f'(\text{net}_j^{(l)}) \sum_{k=1}^{M^{(l+1)}} w_{jk}^{(l+1)} \, \delta_k^{(l+1)}$$

Update rule

$$\Delta w_{ij}^{(l)} = \eta \, \delta_j^{(l)} \, y_i^{(l-1)}$$