These are the slides of the lecture

Pattern Recognition
Winter term 2020/21
Friedrich-Alexander University of Erlangen-Nuremberg.

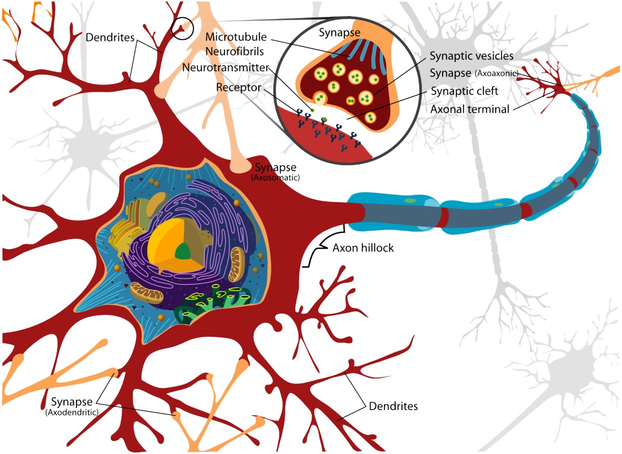
Erlangen, January 8, 2021
Prof. Dr.-Ing. Andreas Maier

Pattern Recognition (PR)

Prof. Dr.-Ing. Andreas Maier
Pattern Recognition Lab (CS 5), Friedrich-Alexander-Universität Erlangen-Nürnberg
Winter Term 2020/21

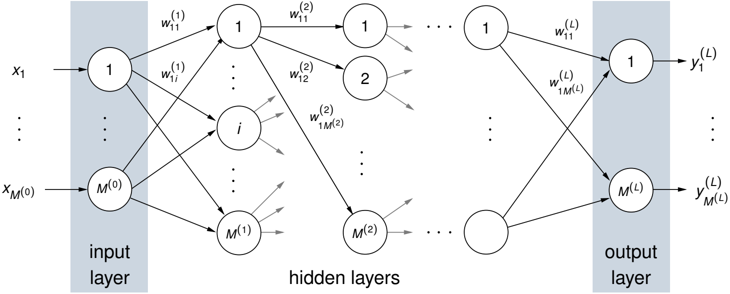
Multi-Layer Perceptrons

Physiological Motivation



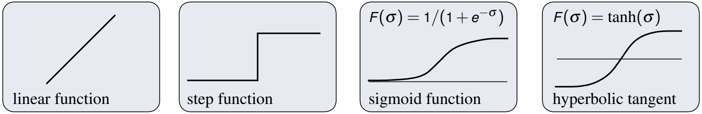
Multi-Layer Perceptrons

Topology



Multi-Layer Perceptrons (cont.)

Activation Functions



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

$$y_j^{(l)} = f\left(\text{net}_j^{(l)}\right)$$

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

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

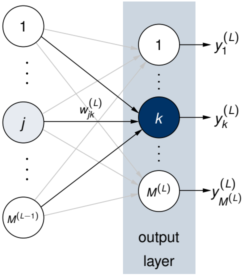
Backpropagation Algorithm (cont.)

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

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

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

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

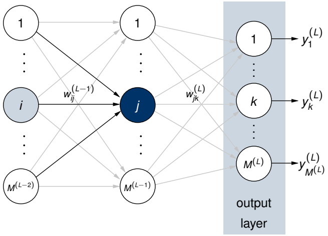


Backpropagation Algorithm (cont.)

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:

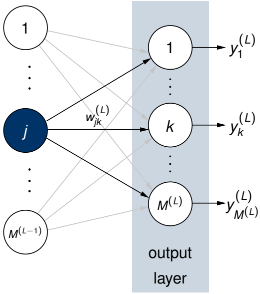
$$
\begin{aligned}
\frac{\partial \varepsilon_{\text{MSE}}}{\partial w_{ij}^{(L-1)}} &= \frac{\partial \varepsilon_{\text{MSE}}}{\partial y_j^{(L-1)}} \cdot \frac{\partial y_j^{(L-1)}}{\partial \text{net}_j^{(L-1)}} \cdot \frac{\partial \text{net}_j^{(L-1)}}{\partial w_{ij}^{(L-1)}} \\
&= \frac{\partial \varepsilon_{\text{MSE}}}{\partial y_j^{(L-1)}} \cdot f'\left(\text{net}_j^{(L-1)}\right) \cdot y_i^{(L-2)}
\end{aligned}
$$



Backpropagation Algorithm (cont.)

- The differentiation of $\partial \varepsilon_{\text{MSE}}$ w. r. t. $y_j^{(L-1)}$ can be computed as the sum of the sensitivity values $\delta_k^{(L)}$ of the layer above weighted by the weights $w_{jk}^{(L)}$:

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



Backpropagation Algorithm (cont.)

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

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

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

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

Linear Network in Matrix Notation

- Fully connected layers can be expressed as matrix multiplications.

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}_3(\hat{\boldsymbol{f}}_2(\hat{\boldsymbol{f}}_1(\boldsymbol{x}))) = \boldsymbol{W}_3 \boldsymbol{W}_2 \boldsymbol{W}_1 \boldsymbol{x}$$

Linear Network in Matrix Notation

- Fully connected layers can be expressed as matrix multiplications.

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}_3(\hat{\boldsymbol{f}}_2(\hat{\boldsymbol{f}}_1(\boldsymbol{x}))) = \boldsymbol{W}_3 \boldsymbol{W}_2 \boldsymbol{W}_1 \boldsymbol{x}$$

- Associated loss function:

$$L(\boldsymbol{\theta}) = \frac{1}{2} || \boldsymbol{W}_3 \boldsymbol{W}_2 \boldsymbol{W}_1 \boldsymbol{x} - \boldsymbol{y} ||_2^2$$

Linear Network in Matrix Notation

- Fully connected layers can be expressed as matrix multiplications.

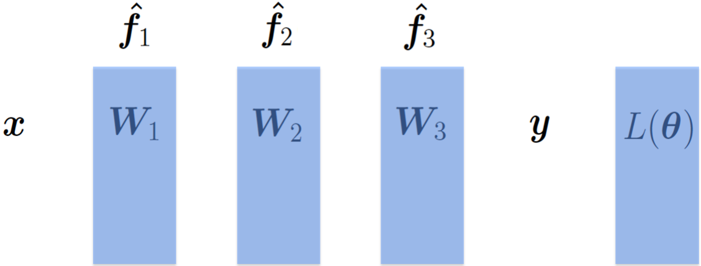
$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}_3(\hat{\boldsymbol{f}}_2(\hat{\boldsymbol{f}}_1(\boldsymbol{x}))) = \boldsymbol{W}_3 \boldsymbol{W}_2 \boldsymbol{W}_1 \boldsymbol{x}$$

- Associated loss function:

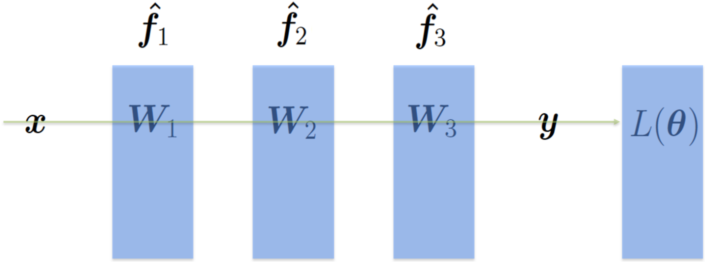
$$L(\boldsymbol{\theta}) = \frac{1}{2} || \boldsymbol{W}_3 \boldsymbol{W}_2 \boldsymbol{W}_1 \boldsymbol{x} - \boldsymbol{y} ||_2^2$$

- Gradients?

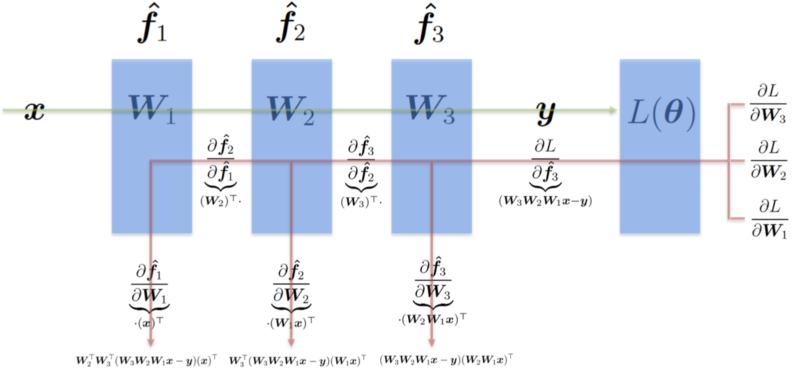
Linear Network in Matrix notation



Linear Network in Matrix notation



Linear Network in Matrix notation



Lessons Learned

- Physiological background: neurons, synapses, action potentials,
- Topology of multi-layer perceptrons
- Activation functions
- Backpropagation algorithm: gradient descent method

Pattern Recognition Lab

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition



Further Readings

. . . from physiology:

- Robert F. Schmidt (Hrsg.):
  Neuro- und Sinnesphysiologie,
  3., korrigierte Auflage, Springer, Berlin, 1998
- Robert F. Schmidt, Florian Lang, Martin Heckmann (Hrsg.):
  Physiologie des Menschen mit Pathophysiologie,
  31., neu bearb. u. aktual. Auflage, Springer, Berlin, 2010