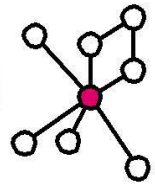
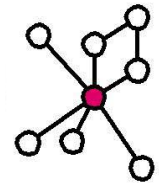
ENDOWED CHAIR OF THE HERTIE FOUNDATION
Knowledge and Data Engineering
DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE



Vorlesung Künstliche Intelligenz Wintersemester 2006/07

Teil III: Wissensrepräsentation und Inferenz

Nachtrag zu Kap.5: Neuronale Netze



Perzeptron

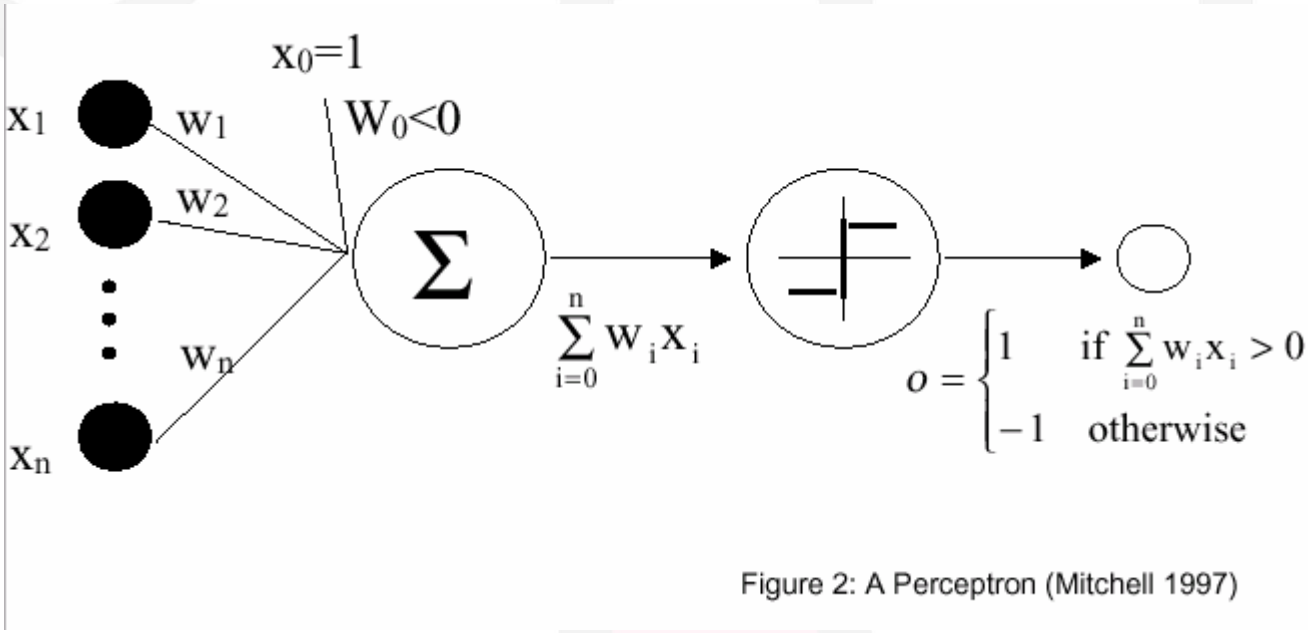
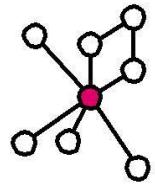


Figure 2: A Perceptron (Mitchell 1997)

- Input $\boldsymbol{x} = (x_1, \ldots, x_n)$
- Gewichte $\boldsymbol{w} = (w_1, \ldots, w_n)$
- Output (o)



Delta-Regel

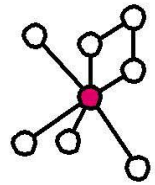
- Beim Training werden die Beispiele dem Netz als Input präsentiert.
- Output ist für die Beispiele bekannt
  --> überwachte Lernaufgabe (supervised)

  (hier: liegt Beispiel in X oder Y?)
- Soll und Ist-Output werden verglichen.
  Bei Diskrepanz werden Schwellenwert und Gewichte nach folgender Delta-Regel angepasst:

$$w_{i,neu} = w_{i,alt} + \eta x_i * (Output_{soll} - Output_{ist})$$

(mit $w_0 = -s$, $x_0 = 1$)

Lernrate



Delta-Regel

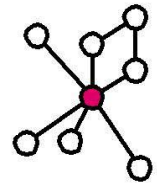
Annahme hier: - Algorithmus mit Lernrate $\eta = 1$
- als Output nur 0 und 1 möglich (d.h. Trennung von zwei Klassen wird gelernt)

Start: Der Gewichtsvektor $\boldsymbol{w}_0$ wird zufällig generiert.
Setze t:=0.

Testen: Ein Punkt $\boldsymbol{x}$ in $X \cup Y$ wird zufällig gewählt
Falls $\boldsymbol{x} \in X$ und $\boldsymbol{w}_t \cdot \boldsymbol{x} > 0$ gehe zu Testen
Falls $\boldsymbol{x} \in X$ und $\boldsymbol{w}_t \cdot \boldsymbol{x} \leq 0$ gehe zu Addieren
Falls $\boldsymbol{x} \in Y$ und $\boldsymbol{w}_t \cdot \boldsymbol{x} < 0$ gehe zu Testen
Falls $\boldsymbol{x} \in Y$ und $\boldsymbol{w}_t \cdot \boldsymbol{x} \geq 0$ gehe zu Subtrahieren

Addieren: Setze $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{x}$.
Setze t := t + 1. Gehe zu Testen

Subtrahieren: Setze $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \boldsymbol{x}$.
Setze t := t + 1. Gehe zu Testen



Beispiel für Anwendung der Delta-Regel

Wir wollen das logische Und lernen.

i	t
0 0	0
0 1	0
1 0	0
1 1	1

Start:

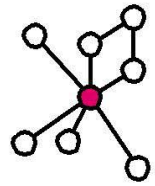
Der Gewichtsvektor w_0 wird „zufällig" generiert:
w_1:=0, w_2:=0, Schwellwert θ = w_2:= 0

0 0 0

Zeit |x_1 x_2| t | o |Error| *zu_addieren/subtr.* | *neue_Gewichte* |

	i	t	a_v	e	$\Delta W(u_1,v)$	$\Delta W(u_2,v)$	$\Delta\theta$	$W(u_1,v)$	$W(u_2,v)$	θ
1. Epoche	0 0	0	0	0	0	0	0	0	0	0
	0 1	0	0	0	0	0	0	0	0	0
	1 0	0	0	0	0	0	0	0	0	0
	1 1	1	0	1	1	1	−1	1	1	−1
2. Epoche	0 0	0	1	−1	0	0	1	1	1	0
	0 1	0	1	−1	0	−1	1	1	0	1

In unserer Notation:
w_1 w_2 w_0



Beispiel für Anwendung der Delta-Regel

Wir wollen das logische Und lernen.

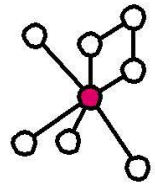
i	t
0 0	0
0 1	0
1 0	0
1 1	1

Falls $\boldsymbol{x} \in X$ und $\boldsymbol{w}_t \cdot \boldsymbol{x} \leq 0$ gehe zu Addieren

Zeit |x_1 x_2| t | o |Error| *zu_addieren/subtr.* | *neue_Gewichte* |

	i	t	a_v	e	$\Delta W(u_1,v)$	$\Delta W(u_2,v)$	$\Delta\theta$	$W(u_1,v)$	$W(u_2,v)$	θ
1. Epoche	0 0	0	0	0	0	0	0	0	0	0
	0 1	0	0	0	0	0	0	0	0	0
	1 0	0	0	0	0	0	0	0	0	0
	1 1	1	0	1	1	1	−1	1	1	−1
2. Epoche	0 0	0	1	−1	0	0	1	1	1	0
	0 1	0	1	−1	0	−1	1	1	0	1

$1x0 + 1x0 - 1_{const} \times 0 = 0$

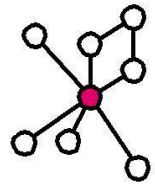


Beispiel für Anwendung der Delta-Regel

Addieren: Setze $w_{t+1} = w_t + x$.

Zeit $|x_1\ x_2|$ t | o $|Error|$ *zu_addieren/subtr.* | *neue_Gewichte* |

	i	t	a_v	e	$\Delta W(u_1,v)$	$\Delta W(u_2,v)$	$\Delta\theta$	$W(u_1,v)$	$W(u_2,v)$	θ
1. Epoche	0 0	0	0	0	0	0	0	0	0	0
	0 1	0	0	0	0	0	0	0	0	0
	1 0	0	0	0	0	0	0	0	0	0
	1 1	1	0	1	1	1	−1	1:=0+1	1:=0+1	−1:= 0-1
2. Epoche	0 0	0	1	−1	0	0	1	1	1	0
	0 1	0	1	−1	0	−1	1	1	0	1



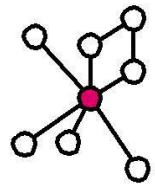
Beispiel für Anwendung der Delta-Regel

Addieren: Setze $w_{t+1} = w_t + x$.

Subtrahieren: Setze $w_{t+1} = w_t - x$.

Zeit |x_1 x_2| t | o |Error| *zu_addieren/subtr.* | *neue_Gewichte* |

	i	t	a_v	e	$\Delta W(u_1,v)$	$\Delta W(u_2,v)$	$\Delta\theta$	$W(u_1,v)$	$W(u_2,v)$	θ
1. Epoche	0 0	0	0	0	0	0	0	0	0	0
	0 1	0	0	0	0	0	0	0	0	0
	1 0	0	0	0	0	0	0	0	0	0
	1 1	1	0	1	1	1	−1	1	1	−1
2. Epoche	0 0	0	1	−1	0	0	1	1:=1+0	1:=1+0	0:= -1+1
	0 1	0	1	−1	0	−1	1	1	0	1



Delta-Regel - Beispiel

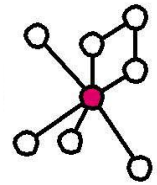
Zeit $|x_1\ x_2|\ t$ | o $|Error|$ *zu_addieren/subtr.* | *neue_Gewichte* |

	i	t	a_v	e	$\Delta W(u_1,v)$	$\Delta W(u_2,v)$	$\Delta\theta$	$W(u_1,v)$	$W(u_2,v)$	θ
1. Epoche	0 0	0	0	0	0	0	0	0	0	0
	0 1	0	0	0	0	0	0	0	0	0
	1 0	0	0	0	0	0	0	0	0	0
	1 1	1	0	1	1	1	−1	1	1	−1
2. Epoche	0 0	0	1	−1	0	0	1	1	1	0
	0 1	0	1	−1	0	−1	1	1	0	1
	1 0	0	0	0	0	0	0	1	0	1
	1 1	1	0	1	1	1	−1	2	1	0
3. Epoche	0 0	0	0	0	0	0	0	2	1	0
	0 1	0	1	−1	0	−1	1	2	0	1
	1 0	0	1	−1	−1	0	1	1	0	2
	1 1	1	0	1	1	1	−1	2	1	1
4. Epoche	0 0	0	0	0	0	0	0	2	1	1
	0 1	0	0	0	0	0	0	2	1	1
	1 0	0	1	−1	−1	0	1	1	1	2
	1 1	1	0	1	1	1	−1	2	2	1
5. Epoche	0 0	0	0	0	0	0	0	2	2	1
	0 1	0	1	−1	0	−1	1	2	1	2
	1 0	0	0	0	0	0	0	2	1	2
	1 1	1	1	0	0	0	0	2	1	2
6. Epoche	0 0	0	0	0	0	0	0	2	1	2
	0 1	0	0	0	0	0	0	2	1	2
	1 0	0	0	0	0	0	0	2	1	2
	1 1	1	1	0	0	0	0	2	1	2

entnommen Nauk, Kruse, S. 50

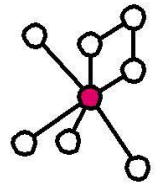
Epoche ohne Veränderung → Ende

Backpropagation-Algorithmus



- Die Gewichtsänderungen können auf zwei Arten erfolgen:
  - Online Training:jedes Gewicht wird sofort angepasst (folgt nur im Mittel dem Gradienten)
  - Batch-Verfahren: es werden alle Datensätze präsentiert, die Gewichtsänderung des Gewichtes berechnet, summiert und dann erst angepasst (entspricht dem Gradienten über dem Datensatz)

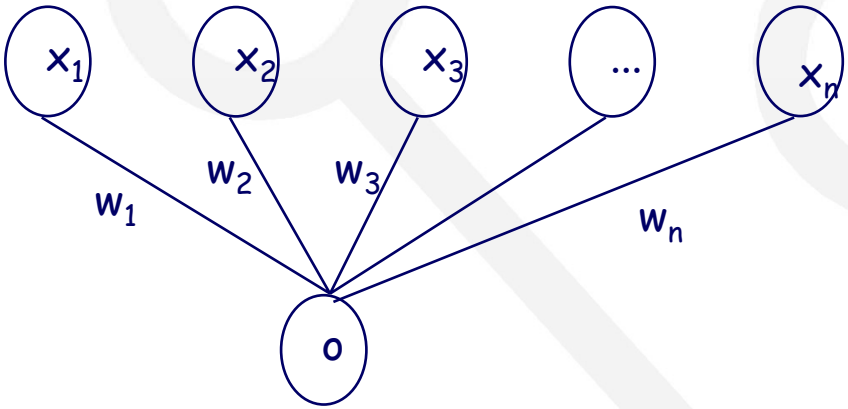
wurde hier angewandt



Agenda

1. Einführung
2. Einfaches Perzeptron
3. Multi-Layer-Perzeptron
   - Vektorschreibweise der Deltaregel
   - Schichten des MLP
   - Backpropagation
   - Probleme der Backpropagation
   - Varianten der Backpropagation

Delta-Regel als Ableitungsregel für Perzeptron



$o = \theta(\underline{x} \cdot \underline{w})$

Fehlergradient:

$F = (o - t)^2 = (\theta(\underline{x} \cdot \underline{w}) - t)^2$

$\partial F / \partial w_i = \partial(o - t) / \partial w_i$

$= \partial(\theta(\underline{x} \cdot \underline{w}) - t)^2 / \partial w_i$

$= \underbrace{2\, \theta'(\underline{x} \cdot \underline{w})}_{\eta} (o-t)\, x_i$

Die Delta-Regel kann als Gradientenabstieg mit (variablem) Lernfaktor interpretiert werden:

$\Delta w_i = \eta\, (o-t)\, x_i$ mit $\eta = 2\, \theta'(\underline{x} \cdot \underline{w})$

(unter der Annahme: θ ist diff.-bar)

2-Layer-Perzeptron

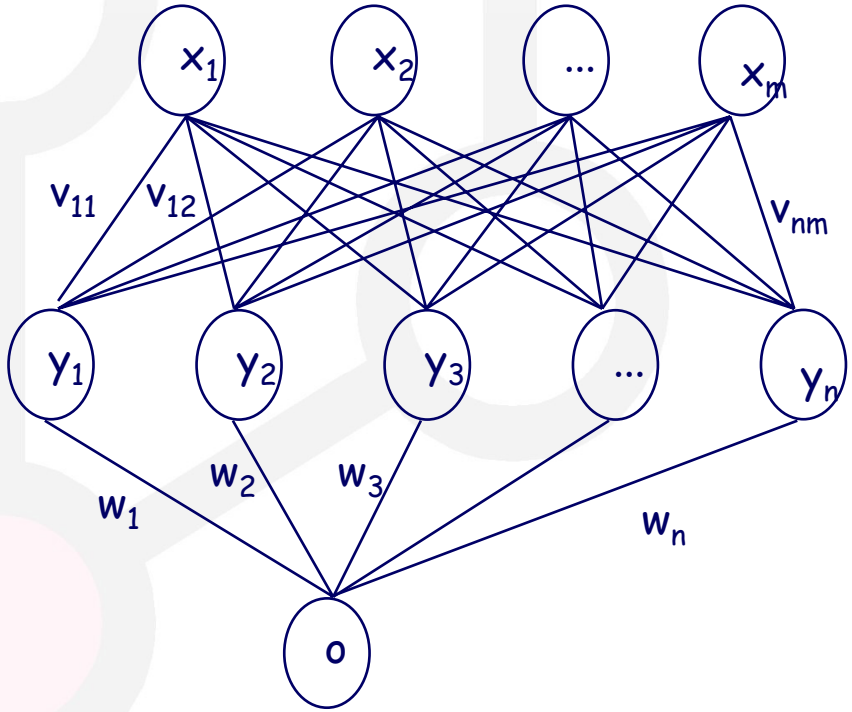
Input-Vektor $\underline{x}$

Gewichtsmatrix $\underline{v}$

Aktivitätsvektor $\underline{y}$

Gewichtsvektor $\underline{w}$

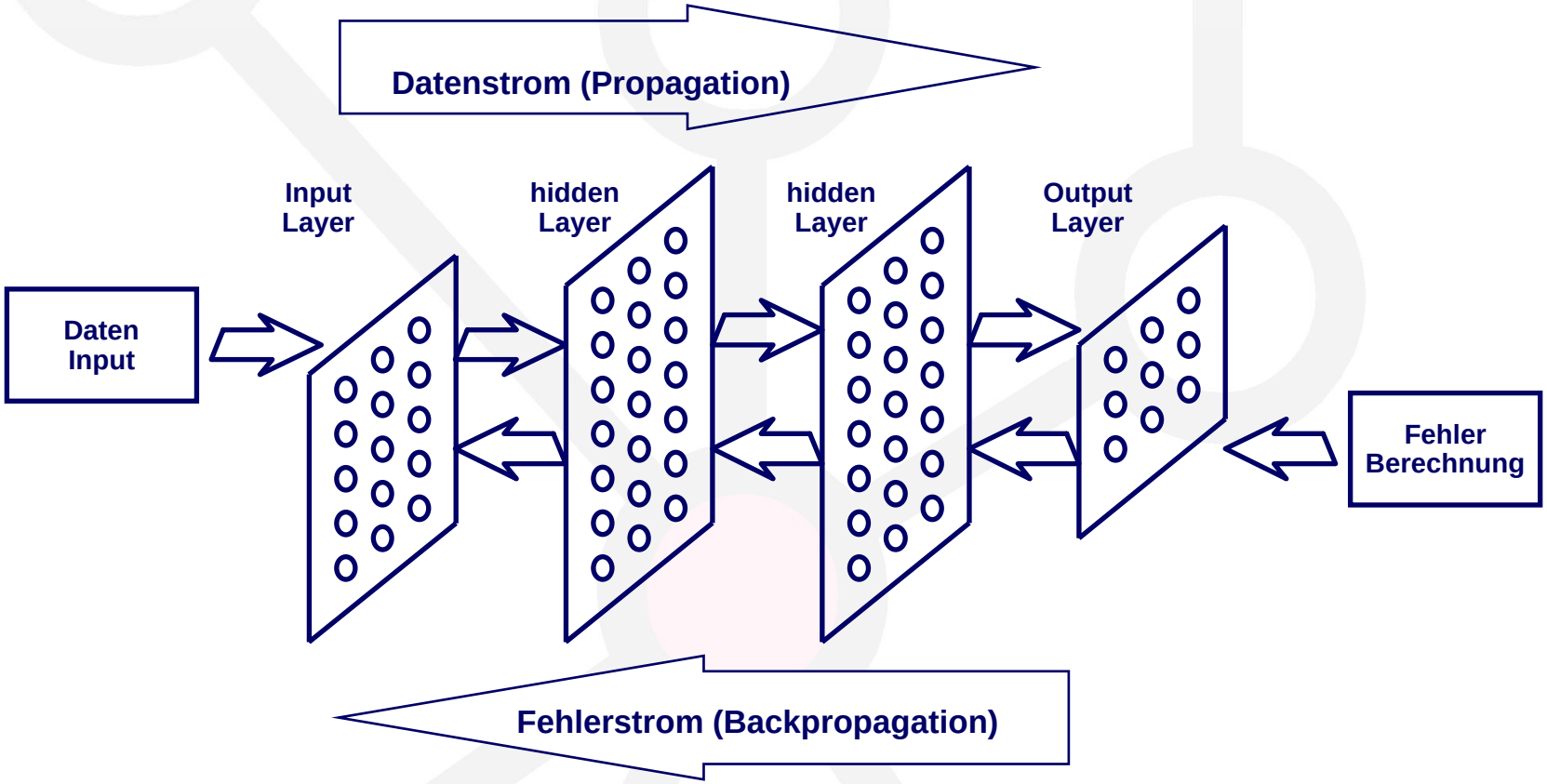
Output o



$$\underline{y} = \theta(\underline{v} \cdot \underline{x})$$

$$o = \theta(\underline{w} \cdot \underline{y})$$

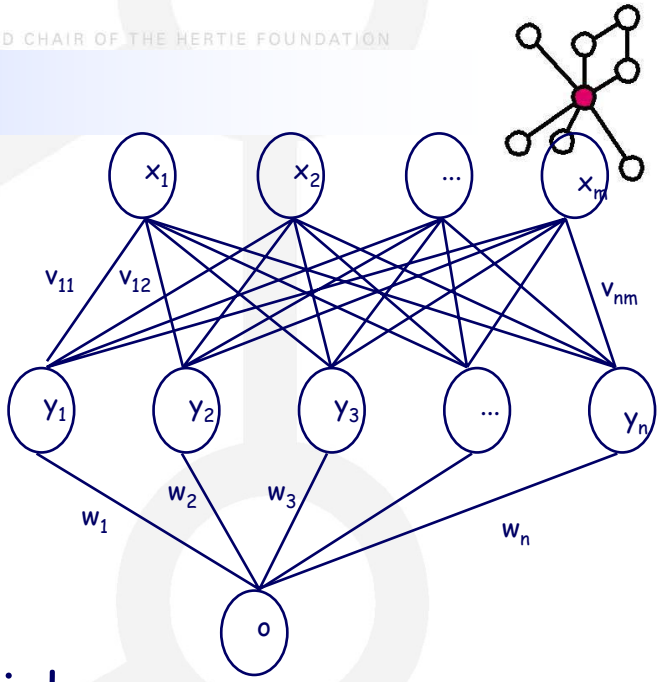
Backpropagation



Multi-Layer-Perzeptron

Fehlerfunktion F (mittlerer quadratischer Fehler) für das Lernen:

$$F_D = \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2$$

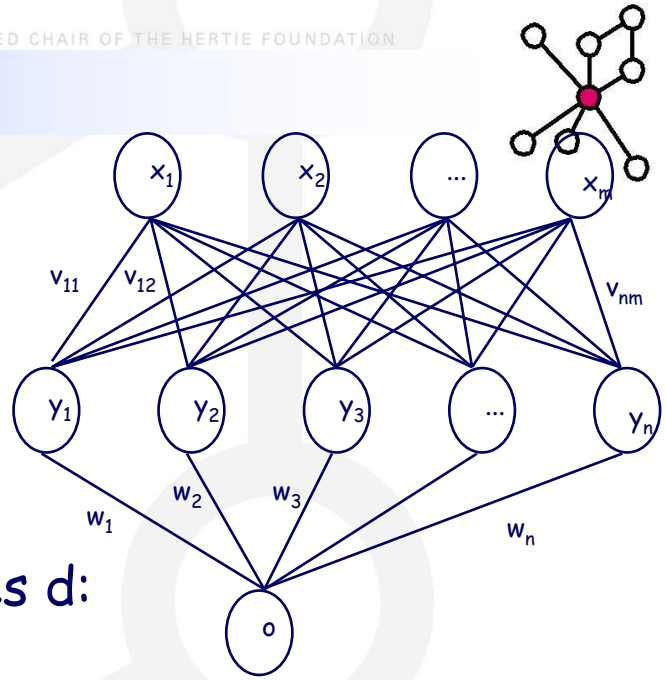


wobei gilt:

D Menge der Trainingsbeispiele
t_d korrekter Output für $d \in D$
o_d berechneter Output für $d \in D$

Die Gewichte müssen so angepasst werden, daß der Fehler minimiert wird. Dazu bietet sich das Gradientenabstiegsverfahren an.
(D.h.: Bergsteigerverfahren mit Vektorraum der Gewichtsvektoren als Suchraum!)

Multi-Layer-Perzeptron



Sei nun ein $d \in D$ gegeben.
Anders geschrieben ist

$F_d = (o - t)^2 = (\theta(\underline{\mathbf{w}} \cdot \underline{\mathbf{y}}) - t)^2$

Der Fehlergradient für w_i lautet für dieses d:

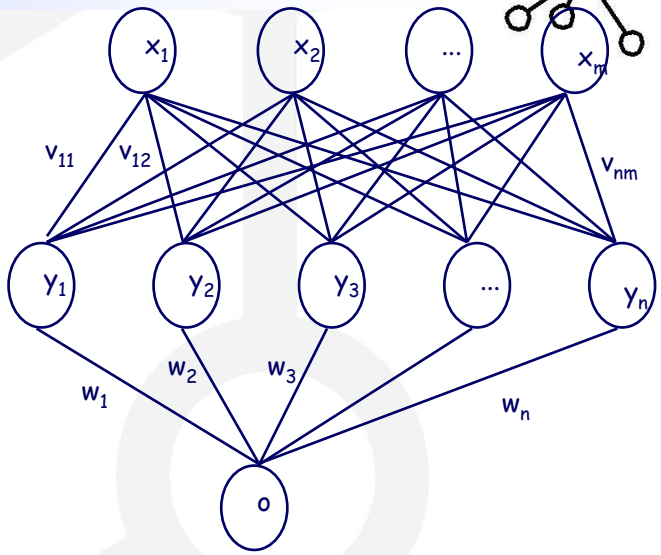
$$\partial F / \partial w_i = \partial (o-t)^2 / \partial w_i = \ldots = 2 \cdot (o-t) \cdot \theta'(\underline{\mathbf{w}} \cdot \underline{\mathbf{y}}) y_i$$

Wir setzen also wie bei der Delta-Regel:

$\Delta w_i = \eta\,(o-t)\,y_i$ mit $\eta = 2\,\theta'(\underline{\mathbf{y}} \cdot \underline{\mathbf{w}})$

und weiter $w_i^{neu} := w_i^{alt} - \Delta w_i$

Multi-Layer-Perzeptron



Fehlergradient für v_{ij} lautet:

$$\partial F/\partial v_{ij} = \partial F/\partial y_i \cdot \partial y_i/\partial v_{ij}$$

...

$$= 2\cdot(o-t) \cdot \theta'(\underline{w}\cdot\underline{y})\cdot w_i \cdot \theta'(\underline{v}_i\cdot\underline{x})\cdot x_j$$

Fehler bei der Ausgabe

Info von Zwischenschicht

Gewicht

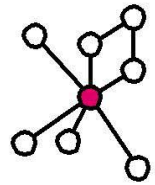
Input

Info von Inputschi

Wir setzen also wie bei der Delta-Regel:

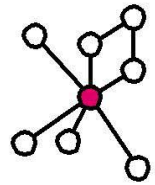
$\Delta v_{ij} = \eta\,(o-t)\,x_j$ mit $\eta = 2\,\theta'(\underline{w}\cdot\underline{y})\cdot w_i \cdot \theta'(\underline{v}_i\cdot\underline{x})$

und weiter $v_{ij}^{neu} := v_{ij}^{alt} - \Delta v_{ij}$



Beispiel analog zur Übung (aber mit anderen Gewichten)

- Wir wollen XOR lernen.
- Als Schwellwertfunktion verwenden wir die Sigmoid-Funktion mit Steigung 1: $\theta(x):=1/(1+e^{-x})$
- Ihre Ableitung lautet $\theta'(x) = e^{-x}/(1 + e^{-x})^2 = \theta(x)(1-\theta(x))$
- Angenommen, die Gewichte seien momentan $v_{11}=0.5$, $v_{12}=0.75$, $v_{21}=0.5$, $v_{22}=0.5$, $w_1=0.5$, $w_2=0.5$.
- Wir berechnen einmal Propagation+Backpropagation für $x_1=1$, $x_2=1$.



Beispiel analog zur Übung

- Angenommen, die Gewichte seien momentan
  v_{11}=0.5, v_{12}=0.75, v_{21}=0.5, v_{22}=0.5,
  w_1=0.5, w_2=0.5 .

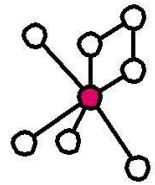
- Wir berechnen einmal Propagation+Backpropagation für x_1=1, x_2=1.

- Propagation:
  - $y_1 := \theta(0.5 \cdot 1 + 0.75 \cdot 1) = \theta(1.25) = 0.78$
  - $y_2 := \theta(0.5 \cdot 1 + 0.5 \cdot 1) = \theta(1.0) = 0.73$
  - $o := \theta(w_1 \cdot y_1 + w_2 \cdot y_2) = \theta(0.5 \cdot 0.78 + 0.5 \cdot 0.73) = \theta(0.755) = 0.68$

- Backpropagation Teil 1:
  - $\Delta w_i = \eta\,(o-t)\,y_i$ mit $\eta = 2\,\theta'(\underline{\mathbf{y}} \cdot \underline{\mathbf{w}})$ ergibt

    $\Delta w_i = 2\,\theta'(\underline{\mathbf{y}} \cdot \underline{\mathbf{w}})\,(o-t)\,y_i$
    $= 2\,\theta'(0{,}755)\,(0.68-0)\,y_i$
    $= 2\,\theta(0{,}755)(1-\theta(0{,}755))0.68\,y_i$
    $= 2 \cdot 0.68(1-0.68)0.68\,y_i$
    $= 0.30\,y_i$

    $\Delta w_1 = 0.3\,y_1 = 0.23$, also $w_1^{neu} := w_1^{alt} - \Delta w_1 := 0.5 - 0.23 = 0.27$
    $\Delta w_2 = 0.3\,y_2 = 0.22$, also $w_2^{neu} := w_2^{alt} - \Delta w_2 := 0.5 - 0.22 = 0.28$



Beispiel analog zur Übung

- Backpropagation Teil 2:
  - $\Delta v_{ij} = \eta\,(o-t)\,x_j$ mit $\eta = 2\,\theta'(\underline{w}\cdot\underline{y})\cdot w_i\cdot\theta'(\underline{v}_i\cdot\underline{x})$ ergibt

$$\begin{aligned}\Delta v_{ij} &= 2\,\theta'(\underline{w}\cdot\underline{y})\cdot w_i\cdot\theta'(\underline{v}_i\cdot\underline{x})\,(o-t)\,x_j\\ &= 2\,\theta'(0.755)\cdot w_i\cdot\theta'(\underline{v}_i\cdot\underline{x})\,0.68\,x_j\\ &= 2\,\theta(0{,}755)(1-\theta(0{,}755))\,w_i\cdot\theta'(\underline{v}_i\cdot\underline{x})\,0.68\,x_j\\ &= 2\cdot 0.68(1-0.68)w_i\cdot\theta'(\underline{v}_i\cdot\underline{x})\,0.68\,x_j\\ &= 0.30\,w_i\cdot\theta'(\underline{v}_i\cdot\underline{x})\,0.68\,x_j\end{aligned}$$

$\Delta v_{11} = 0.3\cdot 0.5\cdot\theta'(0.5\cdot 1+0.75\cdot 1)\cdot 0.68\cdot 1 = 0.3\cdot 0.5\cdot 0.78(1-0.78)\cdot 0.68\cdot 1 = 0.017$
$\Delta v_{12} = 0.3\cdot 0.5\cdot\theta'(0.5\cdot 1+0.75\cdot 1)\cdot 0.68\cdot 1 = 0.017$
$\Delta v_{21} = 0.3\cdot 0.5\cdot\theta'(0.5\cdot 1+0.5\cdot 1)\cdot 0.68\cdot 1 = 0.3\cdot 0.5\cdot 0.73(1-0.73)\cdot 0.68\cdot 1 = 0.02$
$\Delta v_{22} = 0.3\cdot 0.5\cdot\theta'(0.5\cdot 1+0.5\cdot 1)\cdot 0.68\cdot 1 = 0.02$
Also
$v_{11}^{neu} := v_{11}^{alt} - \Delta v_{11} := 0.5-0.017 = 0.483$
$v_{12}^{neu} := v_{12}^{alt} - \Delta v_{12} := 0.75-0.017 = 0.733$
$v_{21}^{neu} := v_{21}^{alt} - \Delta v_{21} := 0.5-0.02 = 0.48$
$v_{22}^{neu} := v_{22}^{alt} - \Delta v_{22} := 0.5-0.02 = 0.48$

- Eine erneute Propagation würde nun ergeben:
  - $y_1 := \theta(0.483\cdot 1 + 0.733\cdot 1) = \theta(1.216) = 0.77$
  - $y_2 := \theta(0.48\cdot 1 + 0.48\cdot 1) = \theta(0.96) = 0.72$
  - $o := \theta(w_1\cdot y_1 + w_2\cdot y_2) = \theta(0.27\cdot 0.77 + 0.28\cdot 0.72) = \theta(0.41) = 0.60$ statt vorher 0.68