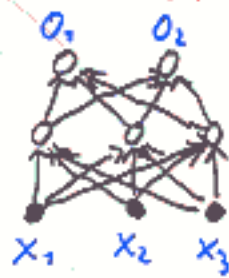


4. Mehrschichtiges Perzeptron (MLP)



Output Schicht

„hidden“ Schicht (mind. eine)

Input schicht

- vorwärts gerichtete Verbindungen
- neben Inputschicht mindestens zwei weitere Schichten
- Übertragungsfunktionen $g(h)$ sind nicht linear

Bemerkung: wenn $g(h)$ linear

→ mehrschichtiges Netz äquivalent zu einschichtigem

Warum MLPs?

- Grenzen einschichtiger Netze überwinden (XOR)
- können beliebige Fkt. approximieren

Lernaufgabe (wie bisher)

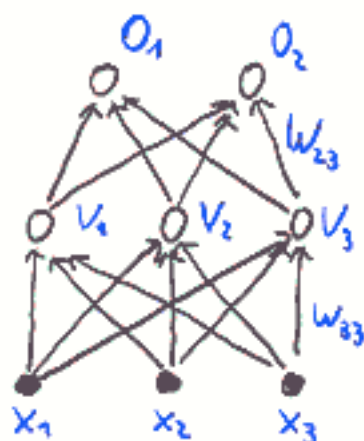
assoziiere Ausgabemuster Y_i^m mit Eingabemuster X_i^m

→ Gewichte w_{ij}^m entsprechend anpassen (Lernalgorithmen)

Back-Propagation-Algorithmus

- Lernalgorithmus für MLPs
- Voraussetzung: Übertragungsfkt. sind differenzierbar
- benutzt Gradientenabstiegsverfahren

Darstellung am zweischichtigen Netz



O_i Output (index i)

W_{ij}

V_j hidden (index j)

w_{jk}

x_k Input (index k)

$$V_j = g(h_j^{\wedge}) = g\left(\sum_k w_{jk} x_k^{\wedge}\right)$$

$$O_i = g(h_i^{\wedge}) = g\left(\sum_j w_{ij} V_j^{\wedge}\right)$$

Fehler fkt.:

$$\begin{aligned} E(\omega) &= \frac{1}{2} \sum_i [y_i^{\wedge} - O_i^{\wedge}]^2 \\ &= \frac{1}{2} \sum_i \left[y_i^{\wedge} - g\left(\sum_j w_{ij} g\left(\sum_k w_{jk} x_k^{\wedge}\right)\right) \right]^2 \end{aligned}$$

→ Gradientenabstiegsverfahren

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = \eta \sum_i \underbrace{(y_i^{\wedge} - O_i^{\wedge}) g'(h_i^{\wedge})}_{=: \delta_i^{\wedge} \text{ (Fehler output)}} V_j^{\wedge} = \eta \sum_i \delta_i^{\wedge} V_j^{\wedge}$$

$$\begin{aligned} \Delta w_{jk} &= -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \sum_i \frac{\partial E}{\partial V_j^{\wedge}} \frac{\partial V_j^{\wedge}}{\partial w_{jk}} \\ &= \eta \sum_i (y_i^{\wedge} - O_i^{\wedge}) g'(h_i^{\wedge}) w_{ij} g'(h_j^{\wedge}) x_k^{\wedge} \\ &= \eta \sum_i \underbrace{\delta_i^{\wedge} w_{ij} g'(h_j^{\wedge})}_{=: \delta_j^{\wedge} \text{ (Fehler hidden Schicht)}} x_k^{\wedge} = \eta \sum_i \delta_j^{\wedge} x_k^{\wedge} \end{aligned}$$

• warum „back-propagation“?

Fehler der vorletzten Schicht aus Fehler der letzten Schicht (output) berechnet

→ entgegengesetzt zur Richtung des Signalflusses (vorwärts)

• Muster werden einzeln betrachtet sein updaten

• update-Regel ist lokal

• Komplexität:

insgesamt n Verbindungen

⇒ Berechnung von $E(\tilde{w})$: $O(n)$ Operationen

Berechnung der n Ableitungen: $O(n^2)$ Operat.

→ aber mit back-propagation nur $O(n)$

Wahl von $g(h)$

$$g(h) = f_A(h) = \frac{1}{1 + \exp(-2\beta h)} \quad \Rightarrow g'(h) = \frac{dg}{dh} = 2\beta g(1-g)$$

$$g(h) = \tanh(\beta h) \quad \Rightarrow g'(h) = \beta(1-g^2)$$

→ besonders bequem, da $g'(h)$ aus $g(h)$ berechnet werden kann

• aber andere Übertragungsfunktionen auch möglich

müssen differenzierbar sein $\Rightarrow g(h) = \operatorname{sgn}(h)$ nicht möglich

Algorithmus für mehrere Schichten

- Netz mit Inputschicht und M weiteren Schichten

V_i^m output i -tes Neuron der m -ten Schicht; $V_i^0 \equiv x_i$

w_{ij}^m Verbindung von V_j^{m-1} zu V_i^m , $m=1, 2, \dots, M$

→ Prozedur für ein Muster:

1. initialisiere w_{ij}^m (kleine zufällige Werte)

2. $V_k^0 = x_k^k \quad \forall k$ (input Schicht = Eingabe Muster)

3. $V_i^m = g(h_i^m) = g\left(\sum_j w_{ij}^m V_j^{m-1}\right)$

$\forall i, m$ bis V_i^M (output) berechnet

4. Fehler für Output Schicht:

$$\delta_i^M = g'(h_i^M) [y_i^M - V_i^M]$$

5. Fehler der vorhergehenden Schichten:

$$\delta_i^{m-1} = g'(h_i^{m-1}) \sum_j w_{ji}^m \delta_j^m$$

für $m=M, M-1, \dots, 2$

6. updaten der Gewichte

$$\Delta w_{ij}^m = \eta \delta_i^m V_j^{m-1}$$

7. zurück zu 2. für nächstes Eingabe Muster

Variationen des Back-propagation Algorithmus

• alternative Fehlerfunktion

andere Fkt $E(\hat{w})$

statt $(y_i^A - o_i^A)^2$ allgemein $F(y_i^A, o_i^A)$ mit globalem Minimum bei $y_i^A = o_i^A$

\Rightarrow Fehler δ_i^A für output Schicht anders
alle anderen Gleichungen des Algorithmus unverändert

Bsp.: relative Entropie

$$E = \sum_{i \in \mu} \left[\frac{1}{2} (1 + y_i^A) \log \frac{1 + y_i^A}{1 + o_i^A} + \frac{1}{2} (1 - y_i^A) \log \frac{1 - y_i^A}{1 - o_i^A} \right]$$

mit $g(h) = \tanh(h)$ folgt

$$\delta_i^A = y_i^A - o_i^A$$

d.h. ohne Faktor $g'(h_i^A)$

andere Ansatz

$$\delta_i^A = (g'(h_i^A) + 0.1) (y_i^A - o_i^A)$$

als Kompromiß zwischen LMS und rel. Entropie

• Trägheits-Term

$$\Delta w_{pq}(t+1) = -\eta \frac{\partial E}{\partial w_{pq}} + \alpha \Delta w_{pq}(t)$$

normaler
Back-prop. Alg.

Trägheitsparameter
 $\alpha = [0, 1]$ z.B. $\alpha = 0.9$

\rightarrow Schwingungen der w_{pq} werden gedämpft

