Praktikum – Neuronale Netze

Michael Gendreyzig

6. Dezember 1995

Hyper-Basisfunktionen-Netze (HBF-Netze)

Zur Musterklassifizierung:
direkter Funktionswert f (gewichtete Summe) = lineare Aktivierungsfunktion im Ausgabeneuron $\longleftrightarrow$ sigmoide Aktivierungsfunktion

$\longrightarrow$ **Generalisierung** $\longrightarrow$

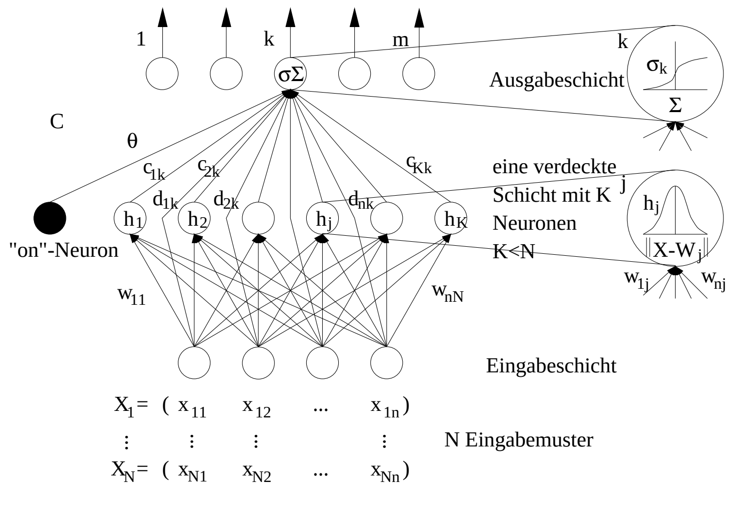
HBF mit individueller Gewichtung von $\|X - W_j\|$ (Argumente der Zentrumsfunktionen) durch:

1. Gewichtung des Abstands $\|X - W_j\|$ mit Faktor
2. Veränderung der Zentrumsfunktion mit zweitem Parameter
3. individuelle Gewichtung der $(x_i - w_{ij}) \quad \forall i$ des Zentrumsvektors
   $\Longrightarrow$ Ellipsoide statt Hyperkugel
4. Verwendung unterschiedlicher Radien und Achsen der Zentrumsfunktionen (optimal gedrehte Ellipsoide)

Untenstehende Formel ergibt sich aus dem RBF-Netz mit folgenden Erweiterungen:

- Schar von Funktionen f_k *(mehrwertige Funktionen)*,
- linearem Anteil von X in f *(Glättung)*,
- sigmoider Aktivierungsfunktion in Ausgabeneuronen und
- individueller Zentrumsfunktionen h_j für alle Neuronen

$$
\begin{aligned}
o_k(X) = \sigma(f_k(X)) &= \sigma\left(\sum_{j=1}^{K} c_{jk}\ h_j(\|X - W_j\|) + \sum_{i=1}^{n} d_{ik} x_i + \theta_k\right) \\
&= \sigma\left(\sum_{j=1}^{K} c_{jk}\ h(\|X - W_j\|, p_j) + \sum_{i=1}^{n} d_{ik} x_i + \theta_k\right)
\end{aligned}
$$



Iteratives Nachtrainieren der RBF-/HBF-Netze

Gründe für iterative Trainingsverfahren:

- W_j i.a. **nicht** optimal
  $\rightarrow$ Verschiebung der Zentren/Stützstellen, Neuberechnung der $c_{jk} \Longrightarrow$ Leistungssteigerung der HBF
- Gewichte d_{ik} nicht direkt berechnet.
  $\rightarrow$ Nachtrainieren ausgehend von sehr kleinen Anfangswerten
- Berechnung der Inversen/Pseudoinversen birgt numerische Probleme $\Longrightarrow$ c_{jk} nicht optimal berechnet.
  $\rightarrow$ Optimierung *(Residuum)*, Nachtraining
- Wahl der p_j (Breite der h_j) nicht immer optimal gewählt.
  $\rightarrow$ Verbesserung durch Lernen

Ziel des Trainings: Minimierung der Fehlerfunktion

$$E = \frac{1}{2} \sum_{k=1}^{m} \sum_{i=1}^{N} (y_{ik} - o_k(X_i))^2$$

$\Longrightarrow$ (nach batch-Backpropagation):

$$\triangle W_j = -\eta_1 \frac{\partial E}{\partial W_j}, \triangle p_j = -\eta_2 \frac{\partial E}{\partial p_j}, \triangle c_{jk} = -\eta_3 \frac{\partial E}{\partial c_{jk}}, \triangle d_{ik} = -\eta_3 \frac{\partial E}{\partial d_{ik}}, \triangle \theta_k = -\eta_3 \frac{\partial E}{\partial \theta_k}$$

$\Longrightarrow$ (Verbesserung):

$$\triangle a(t+1) = -\eta \frac{\partial E}{\partial a} + \mu \triangle a(t) \quad für \quad a = W_j, c_{jk}, d_{ik}, \theta_k, p_j$$

Wahl der Zentren und Radien in RBF-Netzen

- Zentren als Teilmenge der Trainingsmuster gewählt, eventuelle Modifikation der Eingabemuster und Verbindungsgewichte
- LVQ (learning vector quantization)
- selbstorganisierende Karten