

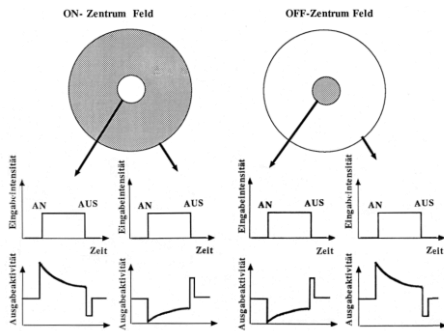
Radiale Basis- funktionen AS1-5

Approximation & Klassifikation mit RBF

Lernen in RBF-Netzen

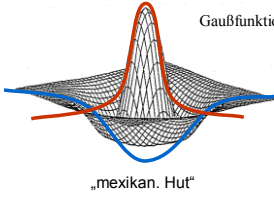
Anwendung RBF-Netze

Biologische receptive Felder



Biologische rezeptive Felder

Modellierung: Überlagerung radialer Basisfunktionen

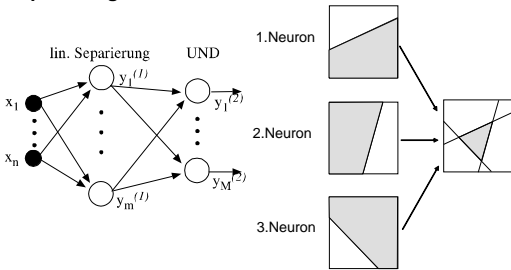


Gaußfunktion $S_G(a_i, x) = \exp(-(a_i - x)^2)$

„radiale Basisfunktion“:
 $S(r)$ monoton in r fallend
mit $r = |x - a_i|$

Multilayer-Klassifikation

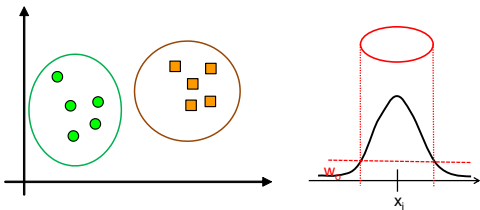
Separierung von Klassen



Klassifikation und RBF

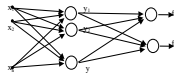
Motivation: lokale Cluster-Klassenbildung

$$\Omega_i = \{ \mathbf{x} \mid S(|\mathbf{x} - \mathbf{x}_i|) > w_0 \}$$



RBF-Netze

Typisch: 2-Schichten Netzwerk



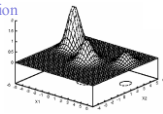
Aktivität

nicht normiert

$$f_i(\mathbf{x}) = \sum_{k=1}^m w_k y_k = \sum_{k=1}^m w_k S_k(\mathbf{x})$$

mit $S_k(\mathbf{x}) = S(\mathbf{c}_k, \mathbf{x}) = e^{-\frac{(\mathbf{c}_k - \mathbf{x})^2}{2\sigma^2}}$

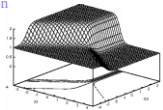
Approximation



normiert

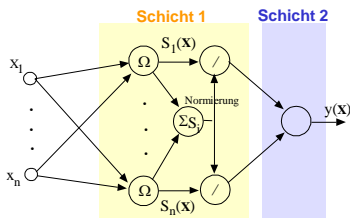
$$f_i(\mathbf{x}) = \sum_{k=1}^m w_k y_k = \frac{\sum_{k=1}^m w_k S_k(\mathbf{x})}{\sum_{j=1}^m S_j(\mathbf{x})}$$

Klassifikation



RBF-Netze

Aktivität Normiertes RBF-Netzwerk



$$y(x) = f(x) = \sum_i w_i \tilde{S}_i(x, c_i) \quad \text{mit} \quad \tilde{S}_i(x, c_i) = \frac{S_i(x, c_i)}{\sum_k S_k(x, c_k)}$$

Klassifikation mit RBF-Netzen

Beste Klassifizierung

Suche Klasse ω_i so, daß $p(\omega_i | \mathbf{x}) = \max_i p(\omega_i | \mathbf{x})$ Bayes-Klassifizierung

Wir wissen: $p(\omega_i | \mathbf{x}) = \frac{p(\omega_i, \mathbf{x})}{p(\mathbf{x})} = \frac{p(\omega_i, \mathbf{x})}{\sum_j p(\omega_j, \mathbf{x})}$

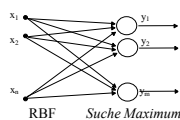
Annahme: Gaußverteilte Abweichungen der \mathbf{x} von den Klassenprototypen \mathbf{c}_i ,

also $p(\mathbf{c}_i, \mathbf{x}) = A e^{-\frac{(\mathbf{c}_i - \mathbf{x})^2}{2\sigma^2}} =: S(\mathbf{c}_i, \mathbf{x})$

➔ Bayes-Klassifizierung mit NN:

Suche Klasse ω_k so, daß mit $y_i = \frac{S_i(\mathbf{c}_i, \mathbf{x})}{\sum_j S_j(\mathbf{c}_j, \mathbf{x})}$

$y_k = \max_i y_i$ winner take all



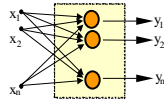
Klassifikation mit winner-take-all

Suche Maximum der Aktivität

Ein-Schicht-Netzwerk

Suche Klasse k so, dass mit $y_i = S(c_i, x) / \sum_j S(c_j, x)$

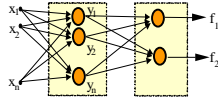
$$y_k = \max_i y_i$$



Zwei-Schichten-Netzwerk

Suche Klasse k so, dass mit $f_i = \sum_j w_j y_j$

$$f_k = \max_i f_k$$



⇒ Lernen **nur** der Gewichte für y_i bzw. f_i

Frage

• Was ist das Ziel der Bayes-Klassifikation?

• Antwort

1. Die minimale Fehlerwahrscheinlichkeit
2. Die maximale bedingte Wahrscheinlichkeit für eine Entscheidung
3. Die minimale Abweichung vom korrekten Wert
4. Die maximale Korrelation mit dem korrekten Wert

Approximation & Klassifikation mit RBF

Lernen in RBF-Netzen

Anwendung RBF-Netze

Lernverfahren

Ansätze

- Schichtweise Einzelanpassung
 - Anpassen der ersten Schicht (Zentrum +Breite)
 - Anpassen der zweiten Schicht (Gewichte)
- Gesamtanpassung, z.B. durch Backpropagation

Anpassung der ersten Schicht

Mechanismen

- **initiale Verteilung** (Anzahl, Lage und Form) der Glockenfunktionen
- **iterative Adaption** der RBF-Parameter an die Trainingsdaten

(1) Bekannte Trainingsdaten

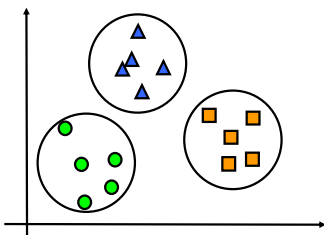
Clustersuche; RBF-Zentren = Clusterzentren
RBF-Breite = Clusterstreuung

(2) Unbekannte Trainingsdaten

- Überdeckung durch Vorwissen
- Überdeckung durch regelmäßiges Raster
- Sukzessive, fehlerorientierte Überdeckung
- Clusteranalyse durch Kohonen-Netze

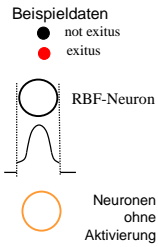
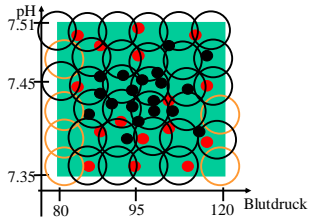
Erste Schicht: Initiale Verteilung

a) Überdeckung durch Vorwissen



Erste Schicht: Initiale Verteilung

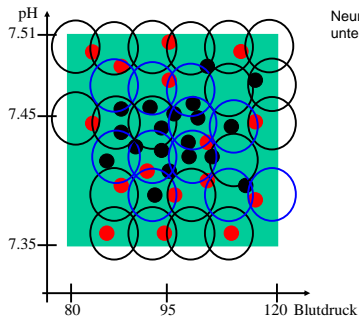
Überdeckung durch Fehlerminimierung
und regelmäßiges Raster



Nach jeder Epoche werden sie gestrichen.

Beispiel: Klassifikation medizinischer Daten fürs Überleben

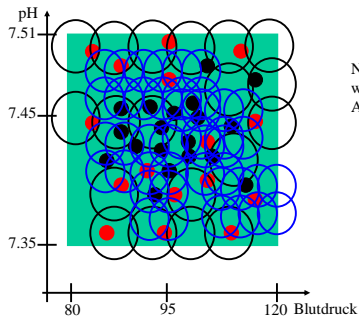
Erste Schicht: Initiale Verteilung



Neuronen mit unterschiedlicher Aktivierung



Erste Schicht: Initiale Verteilung

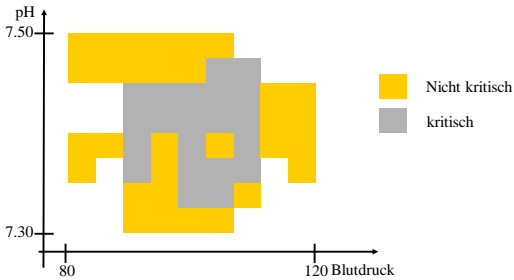


Nach jeder Epoche werden die geringen Aktivitäten geteilt.



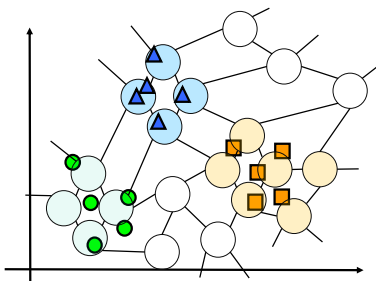
Erste Schicht: Initiale Verteilung

Ergebnis: angepasste Klassengrenzen



Erste Schicht: Initiale Verteilung

d) Clustering durch Kohonen-Maps



Erste Schicht: Initiale Verteilung

Clustering mit Kohonen-Map

$$|x - c_k| = \min_j |x - c_j| \quad \text{Auswahl des Gewinners}$$

• Kohonen map mit RBF

$$c_i(t+1) = c_i(t) + h(c_i, c_k, t) (x - c_i) \quad \text{für alle } i \text{ aus Nachbarschaft } (k)$$

$$c_i(t+1) = c_i(t) \quad \text{sonst}$$

• Rival-Penalty mit RBF

$$c_k(t+1) = c_k(t) + \gamma (x - c_k) \quad \text{für } k$$

$$c_i(t+1) = c_i(t) - \gamma (x - c_i) \quad \text{für zweitnächsten Nachbarn } (k)$$

$$c_i(t+1) = c_i(t) \quad \text{sonst}$$

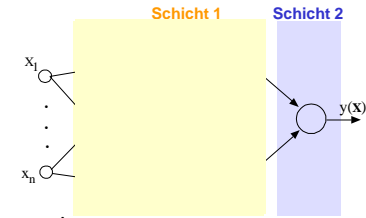
• Soft Winner-take-all mit RBF

$$|x| = |c| = \text{const}$$

$$c_i(t+1) = c_i(t) + h(c_i) (x - c_i) \quad \text{für } h(c_i) = \exp(z_i) / \text{SUM}(\exp(z_j))$$

Anpassung der zweiten Schicht

Normiertes RBF-Netz



$$y(\mathbf{x}) = \hat{f}(\mathbf{x}) = \sum_i w_i v_i \quad \text{mit } v_i = \tilde{S}(\mathbf{x}, \mathbf{c}_i)$$

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \gamma(t) (\mathbf{w}^T \mathbf{v} - \hat{f}(\mathbf{x})) \frac{\mathbf{v}}{|\mathbf{v}|^2} \quad \text{Widrow-Hoff Lernregel}$$

Frage

- Was sind die Vorteile und Nachteile des gemeinsamen Trainings mehrerer Schichten ?

Antwort

- **Vorteil:** Bessere Gesamtanpassung durch Koordination mehrerer Schichten
- **Nachteil:** lang dauernde bis nicht mögliche Konvergenz aller Schichtparameter

RBF Code

```

gamma := 0.1; (* Lernrate festlegen *)
REPEAT
  Read( PatternFile,x,L) (* Eingabe *)
  (* Aktivität bilden im Netz *)
  Sum := 0.0;
  FOR i:=1 TO m DO (* Für alle Neuronen der 1. Schicht *)
    v[i] := Sigm(x-x0[i]) (* Nicht-lin. Ausgabe *)
    Sum := Sum+v[i]; (* Gesamtaktivität bilden *)
  END;
  f := Z(w,v); (* Aktivität 2. Schicht: f(x)=w^T v *)
  f := f/Sum; (* und normieren *)

  (* Lernen der Gewichte der 2. Schicht *)
  v2 := Z(v,v) (* |v|^2 einmal ausrechnen *)
  FOR i:=1 TO m DO (* Für alle Dimensionen *)
    w[i] := w[i] - gamma*(f-L)*v[i]/v2 (* Gewichte verändern: Widrow-Hoff *)
  END;
UNTIL EndOf(PatternFile)

```

RBF Code

Glockenfunktion als Lookup-Tabelle

```
CONST sigma2 = 1.0      (*  $\sigma^2$  *);  
      s = 100           (* Anzahl der Tabellenwerte von  $e^{-x^2}$  *);  
      r = 0.001        (* RBF-Reichweiteschwelle *);  
VAR e : ARRAY[1..s] OF REAL;  
  
(* RBF Funktionstabelle e[,] einmal erstellen *)  
a := 0.0; da := -ln(r)/s;  
FOR i:=1 TO s DO e[i]:=exp(-a); a:=a+da; END;  
  
PROCEDURE S_rbf(x:VEKTOR):REAL;  
BEGIN  
  index := TRUNC((Z(x,x)/sigma2)/da);  
  IF index < s THEN RETURN e[index] ELSE RETURN 0.0 END  
END S_rbf;
```

Approximation & Klassifikation mit RBF

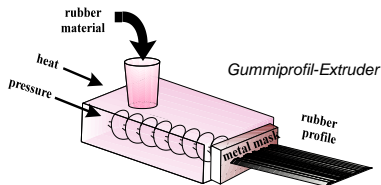
Lernen in RBF-Netzen

Anwendung RBF-Netze

Profillextrusion

Aufgabe

Erstellung von Metallschablonen (Profilwerkzeuge) für die Extrusion von Laufflächen-Profilbändern bei der Reifenherstellung



• Übliches Verfahren: "Trial and error", da "schwarze Kunst".

Profilextrusion

- **Probleme: Betriebswirtschaftliche Nachteile** durch "trial-and-error"
 - Verzögerter Fertigungsbeginn
 - Geringere Produktqualität: nur grobe Schätzung möglich
 - Höhere Einrichtungskosten der Fertigung:
 - umständliche, ressourcenintensive Suche nach dem gewünschten Prozesskennwerten (Masken-profil).
 - zusätzliche Belastung der Fertigungsplaner
 - Unterbrechung der Produktion (Umrüstung und Probebetrieb)
 - finanzieller und personeller Aufwand, jeweils die Metallmaske neu zu erstellen oder nachzuarbeiten.

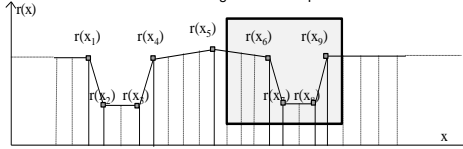
Profilextrusion

- **Probleme bei Mitarbeitern**
 - Unzufriedene Mitarbeiter: langweilige, nicht-kreative Arbeit; Wechsel problematisch
 - Nicht verfügbare Wissensdokumentation
 - Wissen bei Mitarbeiterkrankheit oder Wechsel nicht übertragbar und nicht akkumulierbar

Profilextrusion

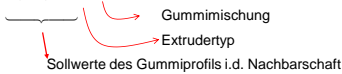
• Lösung: Adaptive Parameterschätzung

Parameter = Quellfaktoren entlang des Maskenprofils



Maske = unbekannte Funktion $r(x)$

$$r(x) = r(y_1, s_1, \dots, y_n, s_n, i, E, G)$$



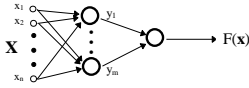
Quellverhalten \equiv lokale Abhängigkeit von der Nachbarschaft

Profilextrusion

- **Lösung:** Adaptive Formale Neuronale Netze

hier: RBF-Netz

$$\mathbf{x} = (x_1, \dots, x_n)$$



Funktionsapproximation (Aktivität)

$$y_i = S_i(\mathbf{x}, \mathbf{c}) = \exp(-d^2) \quad i=1..m \text{ Radiale Basisfunktionen}$$

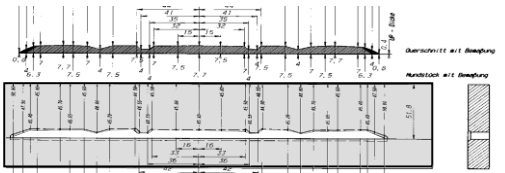
$$\text{mit } d^2 = \|\mathbf{M}(\mathbf{x}-\mathbf{c})\|^2 = (\mathbf{x}-\mathbf{c})^T \mathbf{M} (\mathbf{x}-\mathbf{c}) \text{ und } \mathbf{M}, \mathbf{c} \text{ Parameter}$$

$$F(\mathbf{x}) = \sum_{j=1}^m w_j y_j = \mathbf{w}^T \mathbf{y}$$

$$y_0 \equiv 1, w_0 \equiv \text{Mittelwert}$$

Profilextrusion

- **Training:** Nur 5 Profile gleicher Gummimischung und Extrudertyps



Profilextrusion

- **Lernalgorithmus**
 - **Wachsendes Netz**
Einfügen am Ort des größten Fehlers
 - **Einmalige Anpassung** eines Neurons zur maximalen Fehlerkompensation
 - **Abbruch** bei Fehlerunterschreitung
Minimales Netz für gewünschte Leistung
