Radiale Basis-funktionen

AS2-5

www.uni-frankfurt.de

Approximation & Klassifikation mit RBF

Lernen in RBF-Netzen

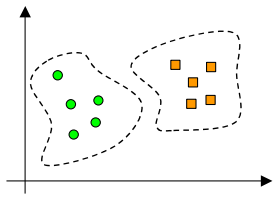
support vector-Maschinen

Anwendung RBF-Netze

Radiale Basisfunktionen

Motivation: lokale Cluster-Klassenbildung

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



Radiale Basisfunktionen

UNIVERSITÄT FRANKFURT AM MAIN

Definition **Glockenfunktionen**

Funktion S_G mit den Eigenschaften

- $S_G(z) \geq 0$, $S_G(-\infty) = S_G(\infty) = 0$,
- $0 < \int_{-\infty}^{\infty} S_G(x)dx < \infty$
- Es ex. ein $a>0$ mit $S_G(z)$ nicht anwachsend $\forall z \in [a,\infty)$, nicht abfallend $\forall z \in (-\infty,a)$

Also ist $S_G(a)$ globales Maximum.

Glockenfunktionen

UNIVERSITÄT FRANKFURT AM MAIN

Beispiele

- **Kombination** von Quetschfunktionen

$$S_G(x_1,..,x_n) = \max(0, 1+\sum_{i=1}^{n} b(x_i) - 1) \; mit \; b(x_i) = \frac{S_Q(1+x_i) + S_Q(1-x_i) - 1}{2S_Q(0) - 1}$$

- **Ableitungen** von Quetschfunktionen

$$S_G(x) = \frac{\partial S_Q}{\partial x}$$

- **Produkte** von Glockenfunktionen

$$S_G(x_1,..,x_n) = S_G(x_1) \times \ldots \times S_G(x_n)$$

- **allgemeine** Radiale Basisfunktionen

$$S_G(\mathbf{x}) = h(|\mathbf{x}|), \qquad \mathbf{x} \in \Re^n \text{ , } h(.) \text{ streng monoton fallend}$$

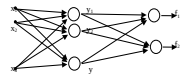
- aus Intervallen **zusammengesetzte** Funktionen

$S_G(z) = (1-z^2)^{2n}$ im Intervall $z \in [-1,+1]$, sonst null.

RBF-Netze

UNIVERSITÄT FRANKFURT AM MAIN

Typisch: 2-Schichten Netzwerk

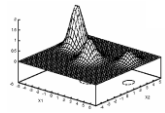


Aktivität

nicht normiert

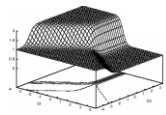
$$f_i(\mathbf{x}) = \sum_{k=1}^{m} w_k y_k = \sum_{k=1}^{m} w_k S_k(\mathbf{x})$$

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



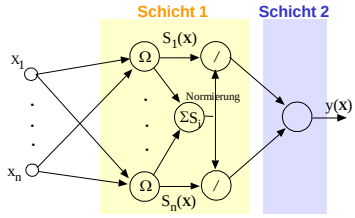
normiert

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



Radiale Basisfunktionen

Aktivität Normiertes RBF-Netzwerk



$y(x) = f(\mathbf{x}) = \sum_i w_i \tilde{S}_i(\mathbf{x},\mathbf{c}_i)$ mit $\tilde{S}_i(\mathbf{x},\mathbf{c}_i) = \dfrac{S_i(\mathbf{x},\mathbf{c}_i)}{\sum_k S_k(\mathbf{x},\mathbf{c}_k)}$

Radiale Basisfunktionen

Basisfunktionen maximaler Information (Entropie)

$H(p^*) = \max_p H(p(x))$ $\quad x \in \Re,$ $\quad p^*(x) = ?$

NB1: $\int p(x)\,dx = 1$ $\quad$ oder $g_1(x) := \int p(x)dx - 1 = 0$

NB2: $\sigma^2 = \langle x^2 \rangle = \int_{-\infty}^{+\infty} p(x)\,x^2\,dx$ $\quad$ oder $g_2(x) := \int_{-\infty}^{+\infty} p(x)x^2dx - \sigma^2 = 0$

Ansatz ***Lagrange-Funktion***

$$L(p, \mu_1,\mu_2) := H(p) + \mu_1 g_1(p) + \mu_2 g_2(p)$$

$$\frac{\partial L(p^*)}{\partial p} = 0, \qquad \frac{\partial L(\mu_i)}{\partial \mu_i} = 0$$

(Rechnung Kap.5.2)

Ergebnis $p^*(x) = A \exp(-x^2/2\sigma^2)$ **Gauß'sche Glockenkurve**

Radiale Basisfunktionen

Basisfunktionen maximaler Information (Entropie)

$H(p^*) = \max_p H(p)$ $\quad x \in [0,1],$ $\quad p^*(x) = ?$

NB: $\int_0^1 p(x)\,dx = 1$ $\quad$ oder $g(x) := \int_0^1 p(x)dx - 1 = 0$ ***ausreichende NB***

Ansatz ***Lagrange-Funktion***

$$L(p,\mu) := H(p) + \mu g(p)$$

$$\frac{\partial L(p)}{\partial p} = 0 = \frac{\partial L(p)}{\partial p}, \qquad \frac{\partial L(\mu)}{\partial \mu} = 0$$

(Rechnung analog Kap.5.2)

Ergebnis: $p^*(x) = \text{const}$ **Uniforme Verteilung**

Transformation mit maximaler Information

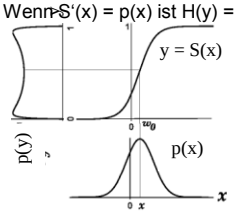
[-∞,+∞] ∋ x → [0,1] **Max. Information** bei uniformer pdf !

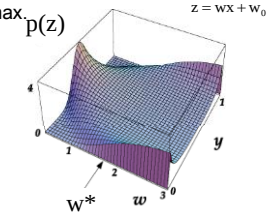
Wie ? (Rechnung Anhang A.4)

$$S_F(z) = \frac{1}{1+\exp(-z)}$$

$$z = wx + w_0$$

Wenn S'(x) = p(x) ist H(y) = max.

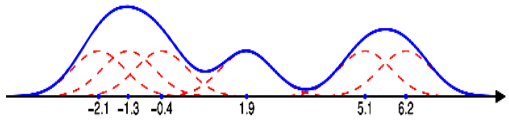




Einstellung von S(x) mittels **w**

Parzen Window - Methode

Approximation durch Überlagerung von Basisfunktionen



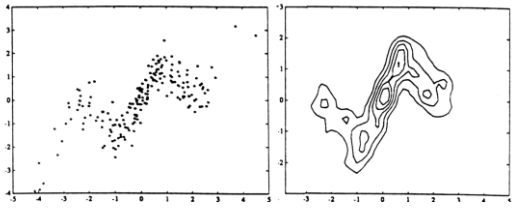
$$\rho(x) = \frac{1}{N}\sum_{i=1}^{N} W(x - x_i) \qquad W(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}.$$

Perfekte Approximation bei abnehmender Breite σ, wobei

$$\lim_{N\to\infty} \sigma(N) = 0, \qquad \lim_{N\to\infty} N\sigma^n(N) = \infty$$

Parzen Window

Approximation durch Überlagerung von Basisfunktionen



Normierung der Variablen

UNIVERSITÄT FRANKFURT AM MAIN

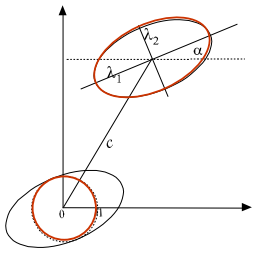
Problem

- PCA etc. problematisch bei heterogenen Variablen,
  z.B. (x_1 [cm], x_2[Pascal], x_3 [°C])
  Welche Einheiten pro Dimension?
- Welche Relation sollen die Einheiten zueinander haben ?

Normierung der Variablen

UNIVERSITÄT FRANKFURT AM MAIN

Lösung einheitliche Transformation aller Variablen
durch Skalierung **S,** Drehung **D,** Verschiebung **V**



$\mathbf{x} \rightarrow \mathbf{z} = \mathbf{SDVx} = \mathbf{Mx}$

$d^2 = \mathbf{z}^2 = \mathbf{z}^T\mathbf{z} = \mathbf{x}^T\mathbf{M}^T\mathbf{M}\mathbf{x}$

$d^2 = (\mathbf{x}-\mathbf{c})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{c})$

Mahalanobis-Abstand

RBF-Ausgabefunktion

$S_G(\mathbf{x}) = A \exp(-(\mathbf{x}-\mathbf{c})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{c}))$

Klassifikation mit RBF-Netzen

UNIVERSITÄT FRANKFURT AM MAIN

Beste Klassifizierung

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

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

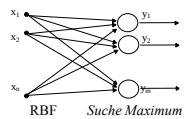
Annahme: Gaußverteilte Abweichungen der $\mathbf{x}$ von den Klassenprototypen $\mathbf{c}_i$,

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*



RBF *Suche Maximum*

Klassifikation mit winner-take-all

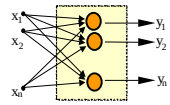
UNIVERSITÄT FRANKFURT AM MAIN

Suche Maximum der Aktivität

Ein-Schicht-Netzwerk

Suche Klasse *k* so, dass mit $y_i = S(\mathbf{c}_i,\mathbf{x}) / \sum_j S(\mathbf{c}_j,\mathbf{x})$

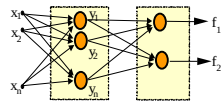
$y_k = \max_i y_i$



Zwei-Schichten-Netzwerk

Suche Klasse *k* so, dass mit $f_i = \sum_i w_i y_i$

$f_i = \max_k f_k$



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

UNIVERSITÄT FRANKFURT AM MAIN

Approximation & Klassifikation mit RBF

Lernen in RBF-Netzen

support vector-Maschinen

Anwendung RBF-Netze

Lernverfahren

UNIVERSITÄT FRANKFURT AM MAIN

Ansätze

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

Anpassung der ersten Schicht

Phasen

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

Initiale Verteilung

- *Bekannte Trainingsdaten*
  Clustersuche, RBF-Zentren = Clusterzentren; RBF-Breite = Clusterstreuung
- *Unbekannte Trainingsdaten*
  - Sukzessiver Netzaufbau
  - Überdeckung durch Fehlerminimierung
  - Überdeckung durch regelmäßiges Raster
  - Clusteranalyse durch Kohonen-Netze

Anpassung der ersten Schicht

Initiale Verteilung

Sukzessiver, fehlerorientierter Netzaufbau

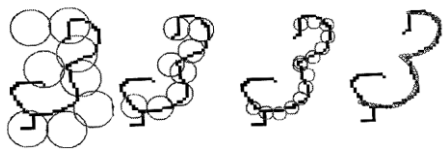
- Start mit einem Neuron
- Füge ein neues Neuron hinzu für jedes Beispiel mit hohem Fehler (Abweichung vom gewünschten Netz-Ausgabewert)
- Verändere die Parameter bei den Nachbarn so, daß der Fehler verringert wird (Einpassen des neuen Neurons)

Das Netzwerk wächst solange, bis der Approximationsfehler auf das gewünschte Maß zurückgegangen ist.

Anpassung der ersten Schicht

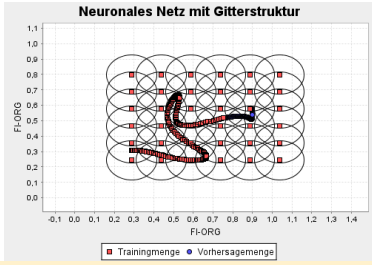
Initiale Verteilung

Adaptiver und sukzessiver Netzaufbau für Abdeckung einer Testverteilung



RBF-Probleme

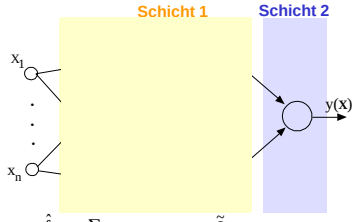
- Sigmoidale Ausgabefkt auch für Extrapolation,
- RBF-Ausgabefkt nur für **Intrapolation**.



ronen

Anpassung der zweiten Schicht

Normiertes RBF-Netz



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

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

Anpassung der zweiten Schicht

- **TLMSE: *Eigenvektor fitting***

$$\mathbf{w}(t) = \mathbf{w}(t\text{-}1) - \gamma(t)\, y\, [\mathbf{x}(t) - \mathbf{w}(t\text{-}1) y]$$ *negative Oja Lernregel*

mit Mittelwertskorrektur $y = (\mathbf{x}\text{-}\mathbf{x}_0)^T\mathbf{w}$

- **Minimisierung der Entropie**

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \gamma \text{ grad } H(y(\mathbf{w}))$$

Approximation von p(x) mit Parzen Windows:

Rechnung

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \gamma(y\text{-}y_k)(\mathbf{x} - \mathbf{x}_k)$$ *Hebb'sche Regel*

Ausgabe y, frühere Ein/Ausgabe k

UNIVERSITÄT FRANKFURT AM MAIN

Approximation & Klassifikation mit RBF

Lernen in RBF-Netzen

support vector-Maschinen

Anwendung RBF-Netze

Gesamtanpassung

UNIVERSITÄT FRANKFURT AM MAIN

- **Lernen mit Backpropagation**

  Zielfunktion $R(M) = \langle (f(\mathbf{x},M) - F(\mathbf{x}))^2 \rangle = \langle r(\mathbf{x},M) \rangle$

  1.Schicht: Lernen der RBF-Koeffizienten M_{ij} durch Gradientenalgorithmus

  $$M^k_{ij}(t+1) = M^k_{ij}(t) - \gamma \frac{\partial}{\partial M^k_{ij}} r(\mathbf{x},\mathbf{M}^k)$$

  2. Schicht: *Standard, z.B. BP*

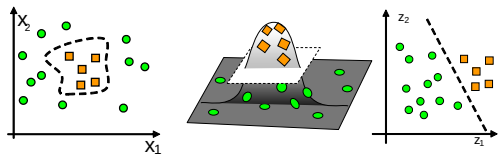
- **Klassifikation durch *support vector-Maschinen***

Gesamtanpassung: nicht-lin. Separierung

UNIVERSITÄT FRANKFURT AM MAIN

- **Klassifikation**

Idee: Verwenden von RBF für lineare Separierung



Hyperfläche *Einbettung* *Hyperebene*

Gesamtanpassung: *Einbettung*

Einbettung in einen Hyperraum

Ziel Klassifikation mit linearer Separierung

$$f_i(z) = \mathrm{sgn}(\mathbf{w}_i^T\mathbf{z} + b) = \begin{cases} +1 & \text{wenn } \mathbf{z} \text{ aus Klasse } i \\ -1 & \text{sonst} \end{cases}$$

Mittel Einbettungsfunktion φ finden: Muster $\mathbf{x} \rightarrow \mathbf{z}$, Prototyp $\mathbf{c}_i \rightarrow \mathbf{w}$

$$f_i(x) = \mathrm{sgn}(\varphi(\mathbf{c}_i)^T\varphi(\mathbf{x}) + b) = \begin{cases} +1 & \text{wenn } \mathbf{x} \text{ aus Klasse } i \\ -1 & \text{sonst} \end{cases}$$

$$= \mathrm{sgn}(K(\mathbf{c}_i,\mathbf{x}) + b) \quad \text{mit } K(\mathbf{c}_i,\mathbf{x}) = \varphi(\mathbf{c}_i)^T\varphi(\mathbf{x}) \quad \textit{kernel function}$$

φ nicht notwendig, es reicht die skalare Funktion $K(\mathbf{c}_i,\mathbf{x})$.

Warum ?

Beispiel !

Gesamtanpassung: *Einbettung*

Mercer's theorem

Sei eine Funktion $K(\mathbf{x},\mathbf{y})$ gegeben.

Erfüllt sie die Bedingung $\int_{-\infty}^{\infty} K(x,y)g(x)g(y)dxdy \geq 0$

für alle Funktionen g mit $\int_{-\infty}^{\infty} g^2(x)dx < \infty$,

so wird sie „positiv definierte Kernfunktion" (*positiv definite kernel*) genannt und **es gibt** eine Funktion $\varphi(\mathbf{x})$ mit $K(\mathbf{x},\mathbf{y}) = \varphi(\mathbf{x})^T\varphi(\mathbf{y})$, einem Skalarprodukt in einem hochdimensionalen Raum.

Typische Kernfunktionen

$K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/a}$	*Gaußfunktion*
$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T\mathbf{y}+1)^d$	*Polynom vom Grad d*
$K(\mathbf{x},\mathbf{y}) = \tanh(\mathbf{x}^T\mathbf{y}-\theta)$	*Multi-layer-Perzeptron*

Gesamtanpassung

Ziel für lin. Separierung: Klassifikationsfehler minimieren

$R(a) = \int \tfrac{1}{2}|f_a(z) - y| \, dp(z,y)$ kontinuierl. Fall

$R_e(a) = 1/N \sum \tfrac{1}{2}|f_a(z_i) - y_i|$ diskreter Fall

beim Training mit N Mustern.

Erreichbar im diskreten Fall mit Mindestwahrscheinlichkeit $1-\eta$

$R(a) \leq R_e(a) + \phi(\frac{h}{N}, \frac{\log\eta}{N})$ Vapnik 1979

mit $\phi(a,b) = \sqrt{a(\log(2/a)+1) - b + (\log 4)/N}$

bei beliebiger Klassifkation $f_a(z)$

und gegebener „**Diagnosevariabilität" h**

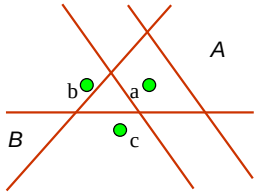
Gesamtanpassung

UNIVERSITÄT FRANKFURT AM MAIN

„Diagnosevariabilität“ h VC-Dimension

h = maximale Anzahl der Punkte, die durch die Diagnosemaschine auf 2^h Arten in zwei Klassen geteilt werden können.

Beispiel h = 3 Punkte, 2^3=8 Diagnosearten möglich. h < 4



A={}, B={a,b,c}
A={a}, B={b,c}
A={a,b}, B={c}
A={a,c}, B={b}

sowie 4 Möglichkeiten bei Umbenennung A→B, B→A

Gesamtanpassung: *support vector machine*

UNIVERSITÄT FRANKFURT AM MAIN

Forderung für lin. Separierung

„Lege die Hyperebene so, dass sie maximalen Abstand zu allen Grenzpunkten hat“

$$\min_i | \mathbf{w}^T\mathbf{z}_i + b | = 1 \qquad \textit{Mindestabstand = 1}$$

$$f_{w,b}(\mathbf{z}_i) = \mathrm{sgn}(\mathbf{w}^T\mathbf{z}_i + b) \overset{!}{=} y_i \in \{+1,-1\} \quad \text{Klassifizierung}$$

$$(\mathbf{w}^T\mathbf{z}_i + b)\, y_i \geq 1$$

$$(\mathbf{w}^T\mathbf{z}_i + b)\, y_i \geq 1 - \xi_i \qquad \textit{Minimierung des strukturellen Risikos}$$

ξ_i *Schlupfvariable*

Gesamtanpassung : *support vector machine*

UNIVERSITÄT FRANKFURT AM MAIN

Ansatz *support vector* – Maschine

- Alle Muster sind in in einem Cluster: $|\mathbf{z}_i - \mathbf{a}| < r$ Kugelradius
- Endliche Beschreibung der Trennung $|\mathbf{w}| \leq A$

$\Rightarrow$ $\mathbf{h < r^2A^2 + 1}$ Vapnik 1995

Reduzierung des Klassifizierungsfehlers durch Beschränkung von *h*

Neues Ziel: Minimierung von

$$T(\mathbf{w}, \xi_i) = \tfrac{1}{2}\,\mathbf{w}^2 + \gamma \sum_{i=1}^{N} \xi_i$$

mit NB $g(\mathbf{w},i) = 1 - (\mathbf{w}^T\mathbf{z}_i + b)y_i - \xi_i = 0$

Gesamtanpassung : *support vector machine*

Ansatz *Lagrangefunktion*

$$L(w,\mu_1,..,\mu_N) = T(w, \xi_i) + \sum_{i=1}^{N} \mu_i g(\mathbf{w}, i)$$

$$\frac{\partial}{\partial \mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{N} \mu_i y_i \mathbf{z}_i = 0 \quad \text{oder} \quad \mathbf{w} = \sum_{i=1}^{N} \mu_i y_i \mathbf{z}_i$$

Bestimmung der μ_i durch Maximierung von

$$W(\alpha) = \sum_{i=1}^{N} \mu_i - \tfrac{1}{2} \mathbf{w}^2 \qquad \textit{quadrat. Optimierung}$$

mit NB $\quad 0 < \mu_i < \gamma, \; i = 1,...,N \;$ und $\sum_{i=1}^{N} \mu_i y_i = 0$

$\mu_i \neq 0$: $\mathbf{z}_i$ = Support-Vektoren

Gesamtanpassung : *support vector machine*

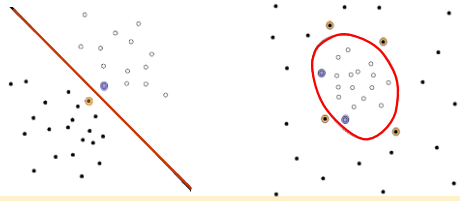
Ergebnis: Klassifizierung mit

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T \mathbf{z} + b) = \mathrm{sgn}\left(\sum_{i=1}^{N} \mu_i y_i K(\mathbf{c}_i, \mathbf{x}) + b\right)$$ 2-Schicht-RBF-Netz

Beispiel

Supportvektoren

lin. Separierung entspricht nicht-lin. Separierung



support vector - Maschine

UNIVERSITÄT FRANKFURT AM MAIN

Ergebnisse

Daten	Klassifikations-Fehlerrate		
US Postal Service	Klass. RBF	RBF mit SV Zentren	Reine SV-Maschine
Training (7291 Muster)	**1,7 %**	**0,0 %**	**0,0 %**
Test (2007 Muster)	**6,7 %**	**4,9 %**	**4,2 %**

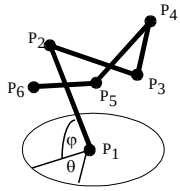
Frage: Warum ist diese Gegenüberstellung problematisch ?

Erkennen von 3D-Figuren

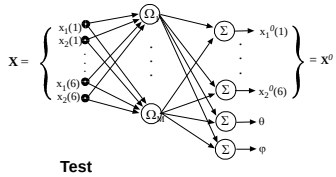
UNIVERSITÄT FRANKFURT AM MAIN

Training Feste Figur $\mathbf{x}_i$ aus 6 Punkten,
40-100 Random-Projekt. auf 2D-Fläche

Poggio, Edelman 1990



M Trainingsmuster = *M* hidden units



$\theta \in [0°,30°,60°,...,360°]$
$\varphi \in [0°,30°,60°,...,180°]$
6x12=78 Standardwinkel

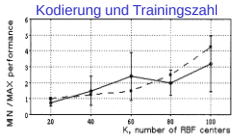
Test
Erkenne Figurtyp + Winkel θ und φ

Erkennen von 3D-Figuren

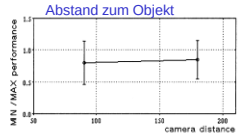
UNIVERSITÄT FRANKFURT AM MAIN

- **Ergebnisse** Erkennungsleistung

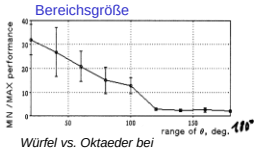
Kodierung und Trainingszahl



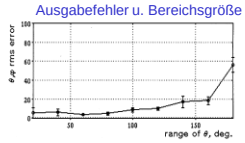
Abstand zum Objekt



Bereichsgröße



Ausgabefehler u. Bereichsgröße



Würfel vs. Oktaeder bei
40 Trainingsmustern / RBFs