Basisfunktionen

Volker Tresp

I am an AI optimist. We've got a lot of work in machine learning, which is sort of the polite term for AI nowadays because it got so broad that it's not that well defined.

`Bill Gates (Scientific American Interview, 2004)`

"If you invent a breakthrough in artificial intelligence, so machines can learn," Mr. Gates responded, "that is worth 10 Microsofts." (Quoted in NY Times, Monday March 3, 2004)

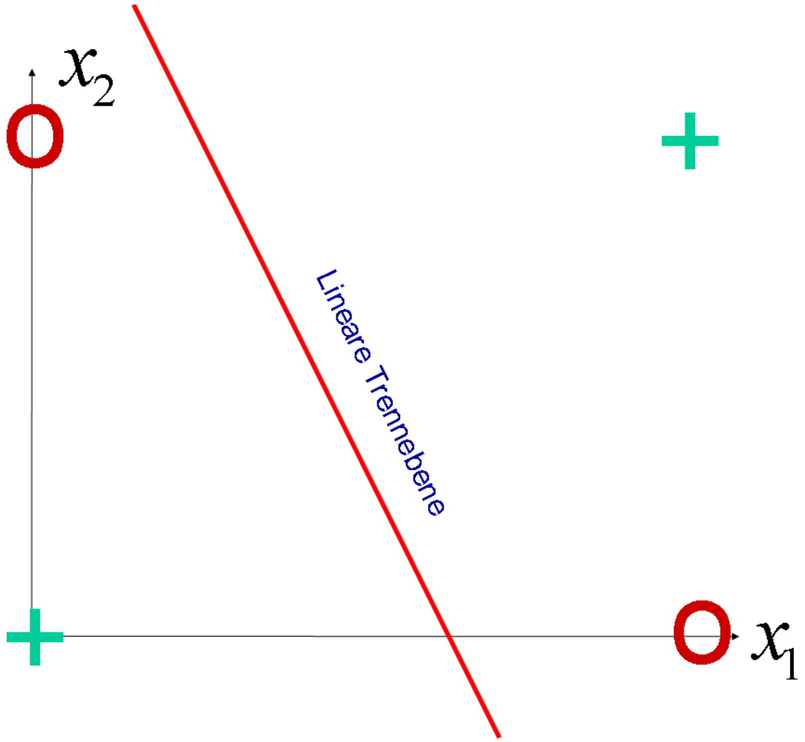
Nichtlineare Abbildungen und Klassifikatoren

- Regression:
  - Es ist eher unwahrscheinlich, dass die wahre Funktion $f(\mathbf{x})$ linear ist, obwohl dies bei Problemstellungen in hohen Eingangsdimensionen durchaus eine praktische Annahme ist. (Oder in der Physik: $F = ma$)
  - Wir wollen die Mächtigkeit unseres Modelles erhöhen, so dass auch beliebige nichtlineare funktionelle Abhängigkeiten gelernt werden können
- Klassifikation:
  - Ebenso kann man annehmen, dass lineare Trennflächen für die Mehrzahl der Anwendungen nicht optimal sind
  - Wir wollen die Mächtigkeit unseres Modelles erhöhen, so dass auch beliebige nichtlineare Trennflächen modelliert werden können

Trick

- Der Trick besteht darin, die Eingangsdaten in einen hoch dimensionalen Raum zu transformieren, in dem das Problem dann wieder linear ist!
- Andere Sichtweise: definition geeigneter Merkmale
- Andere Sichtweise: definition geeigneter Basisfunktionen

XOR ist nicht linear separierbar

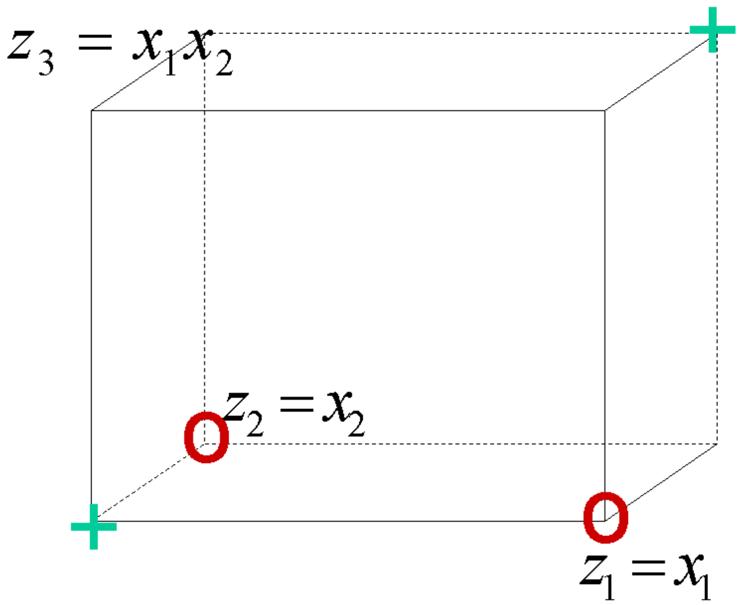


Wie kann man es dennoch schaffen, unter Zuhilfenahme eines linearen Klassifikators, eine nichtlineare Trennfläche zu realisieren?

Hinzunahme weiterer Basisfunktionen

- Lineares Modell: Eingangsvektor: $1, x_1, x_2$
- Lineares Modell mit zusätzlicher Basisfunktion: $1, x_1, x_2, x_1 x_2$
- Der Wechselwirkungsterm (Interaktionsterm) $x_1 x_2$ koppelt die Eingänge in einer nichtlinearen Weise

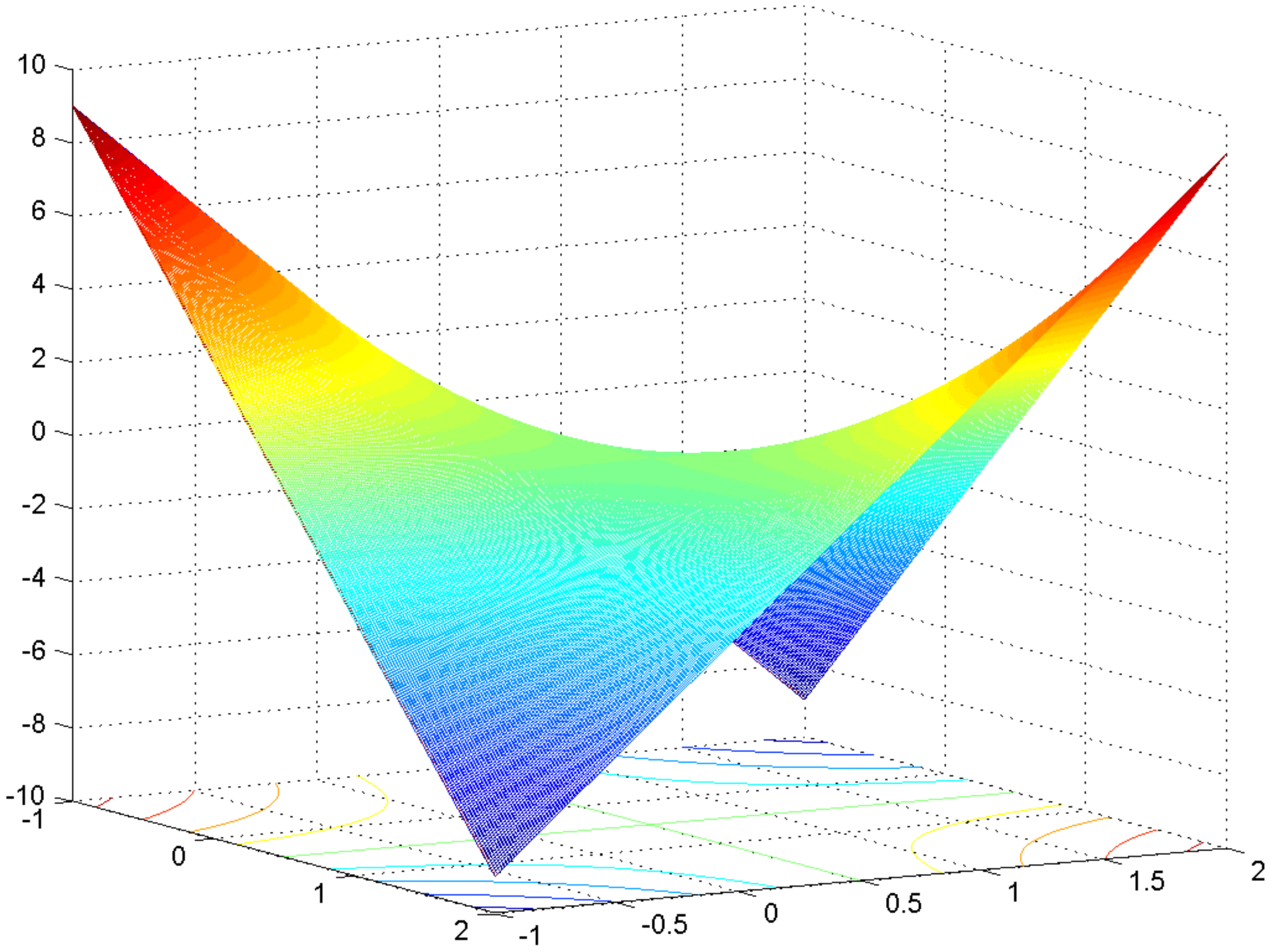
Mit $z_3 = x_1 x_2$ wird das XOR linear separierbar



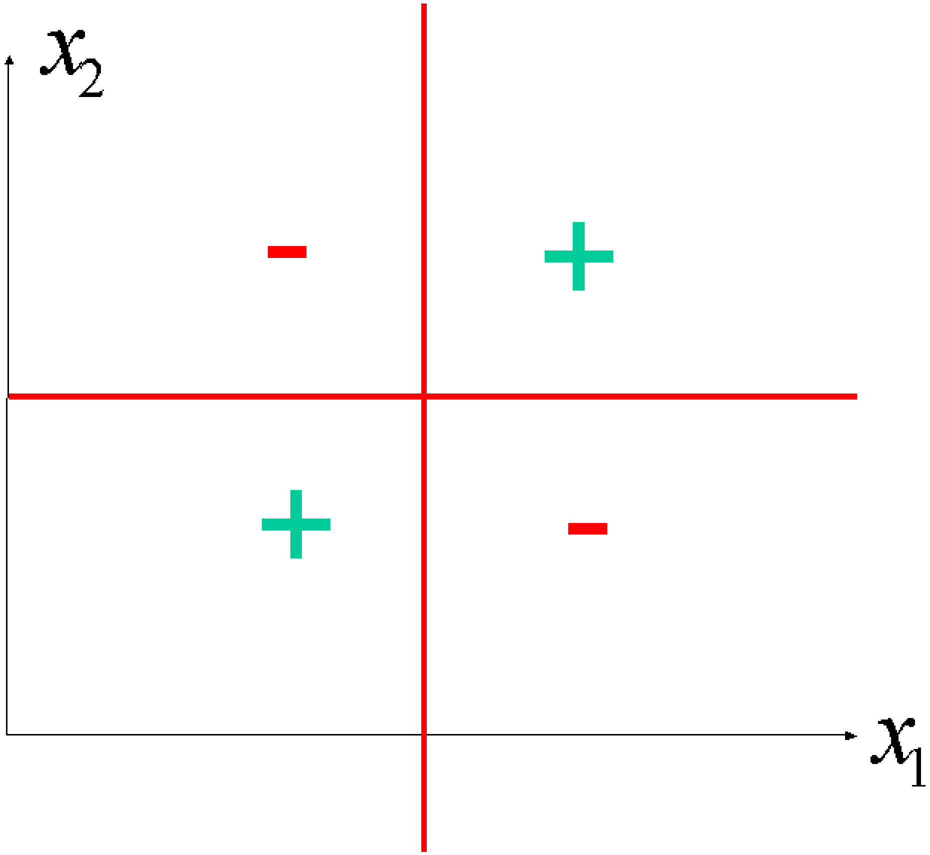
$$f(\mathrm{x}) = 1 - 2x_1 - 2x_2 + 4x_1x_2 = \phi_1(x) - 2\phi_2(x) - 2\phi_3(x) + 4\phi_4(x)$$

mit $\phi_1(x) = 1, \phi_2(x) = x_1, \phi_3(x) = x_2, \phi_4(x) = x_1x_2$

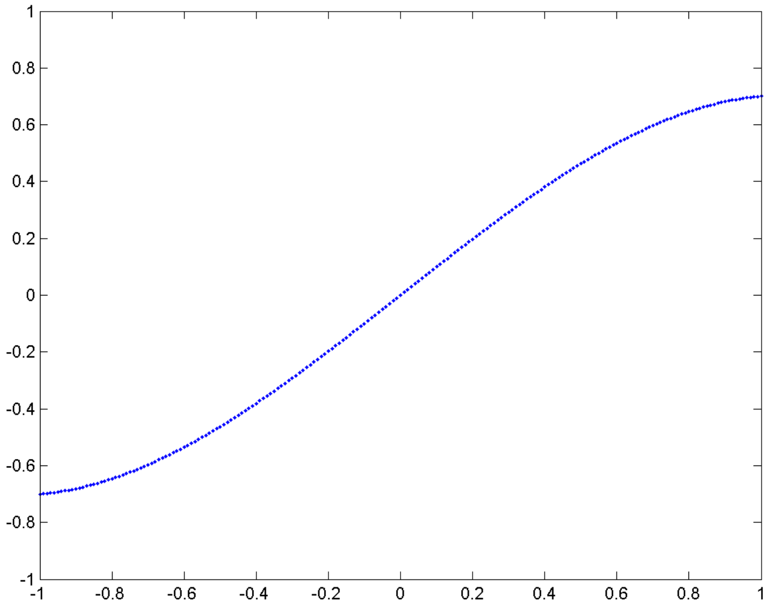
$$f(\mathbf{x}) = 1 - 2x_1 - 2x_2 + 4x_1x_2$$



Trennebenen

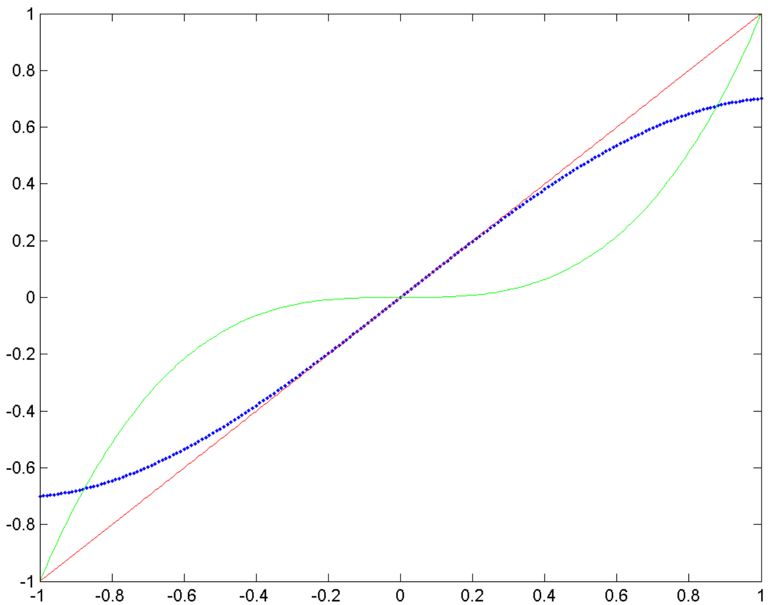


Eine Nichtlineare Abbildung



Wie kann man es schaffen, unter Zuhilfenahme einer linearen Regression, eine nichtlineare Abbildung zu realisieren?

$$f(x) = x - 0.3x^3$$



Basisfunktionen $\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2, \phi_4(x) = x^3$ und $\mathbf{w} = (0, 1, 0, -0.3)$

Grundidee

- Die Grundidee ist denkbar einfach: Neben den Eingangsvariablen $\mathbf{x}_i$ formen wir zusätzliche Variablen, die sich als deterministische Funktionen der Eingangsvariablen berechnen lassen

- Beispiel: polynomiale Basisfunktionen

$$\{1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1^2, x_2^2, x_3^2\}$$

- Basisfunktionen $\{\phi_h(\mathbf{x})\}_{h=1}^{M_\phi}$

- Im Beispiel:

$$\phi_1(\mathbf{x}) = 1 \quad \phi_2(\mathbf{x}) = x_1 \quad \phi_6(\mathbf{x}) = x_1x_3 \quad \ldots$$

- Unabhängig von der Wahl der Basisfunktionen, wird die Regression mit den obigen Gleichungen (der linearen Regression) berechnet

Review: Lineare Regression (Penalized LS)

- Mehrdimensionales lineares Modell:

$$f(\mathbf{x}_i, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_{i,j} = \mathbf{x}_i^T \mathbf{w}$$

- Regularisierte Kostenfunktion

$$\text{cost}^{pen}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \lambda \sum_{i=0}^{M-1} w_i^2$$

- Die PLS-Lösung

$$\hat{\mathbf{w}}_{pen} = \left(\mathbf{X}^T\mathbf{X} + \lambda I\right)^{-1} \mathbf{X}^T \mathbf{y} \quad \text{mit} \quad \mathbf{X} = \begin{pmatrix} x_{1,0} & \dots & x_{1,M-1} \\ \dots & \dots & \dots \\ x_{N,0} & \dots & x_{N,M-1} \end{pmatrix}$$

Regression mit Basisfunktionen

- Modell mit Basisfunktionen:

$$f(\mathbf{x}_i, \mathbf{w}) = \sum_{j=1}^{M_\phi} w_j \phi_j(\mathbf{x}_i)$$

- Regularisierte Kostenfunktion

$$J_N^{pen}(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \lambda \sum_{i=1}^{M_\Phi} w_i^2$$

- Die PLS-Lösung

$$\hat{\mathbf{w}}_{pen} = \left(\mathbf{\Phi}^T \mathbf{\Phi} + \lambda I\right)^{-1} \mathbf{\Phi}^T \mathbf{y} \quad \text{mit} \quad \mathbf{\Phi} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \dots & \phi_{M_\phi}(\mathbf{x}_1) \\ \dots & \dots & \dots \\ \phi_1(\mathbf{x}_N) & \dots & \phi_{M_\phi}(\mathbf{x}_N) \end{pmatrix}$$

Konstruktion nichtlinearer Systeme für Regression und Klassifikation

- Regression:

$$f(\mathbf{x}) = \sum_{j=1}^{M_\phi} w_j \phi_i(\mathbf{x})$$

Wie diskutiert lassen sich die least squares oder die penalized least squares Lösungen berechnen

- Klassifikation mit Diskriminantenfunktion:

$$f(\mathbf{x}) = h(\mathbf{x}) = \sum_{j=1}^{M_\phi} w_j \phi_j(\mathbf{x})$$

Die Perceptron Lernregel lässt sich anwenden, wenn man $1, x_{i,1}, x_{i,2}, \ldots$ ersetzt durch $\phi_1(x_i), \phi_2(x_i), \ldots$

Herausforderung: Basisfunktionen

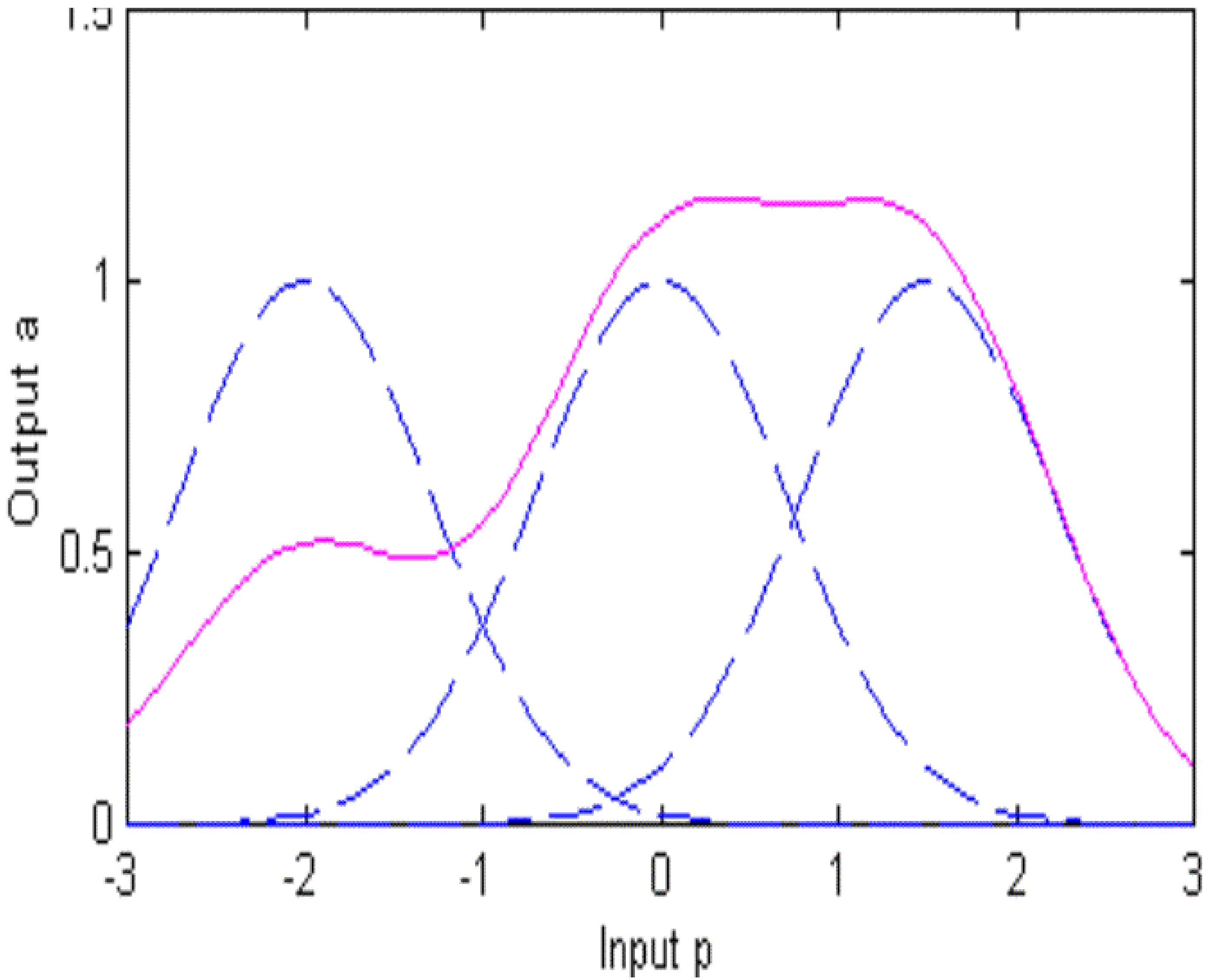
- Zur Berechnung der Parameter sind die Lösungen bekannt
- Das eigentliche Problem ist nun, aufgabenangepasste Basisfunktionen auszuwählen, so dass die wahre Abhängigkeit möglichst gut modelliert werden kann

Radiale Basisfunktionen (RBF)

- Wir hatten schon polynomiale Basisfunktionen kennengelernt
- Eine weitere beliebte Klasse von Basisfunktionen sind Radiale Basisfunktionen (RBF). Typische Vertreter sind Gauss-Glocken

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2s_j^2}|\mathbf{x} - \mathbf{c}_j|^2\right)$$

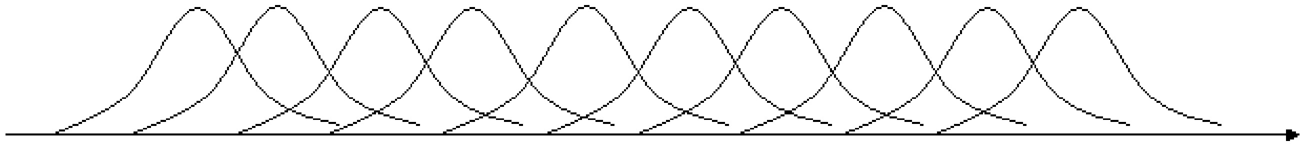
Drei RBFs (blau) formen $f(x)$ (pink)

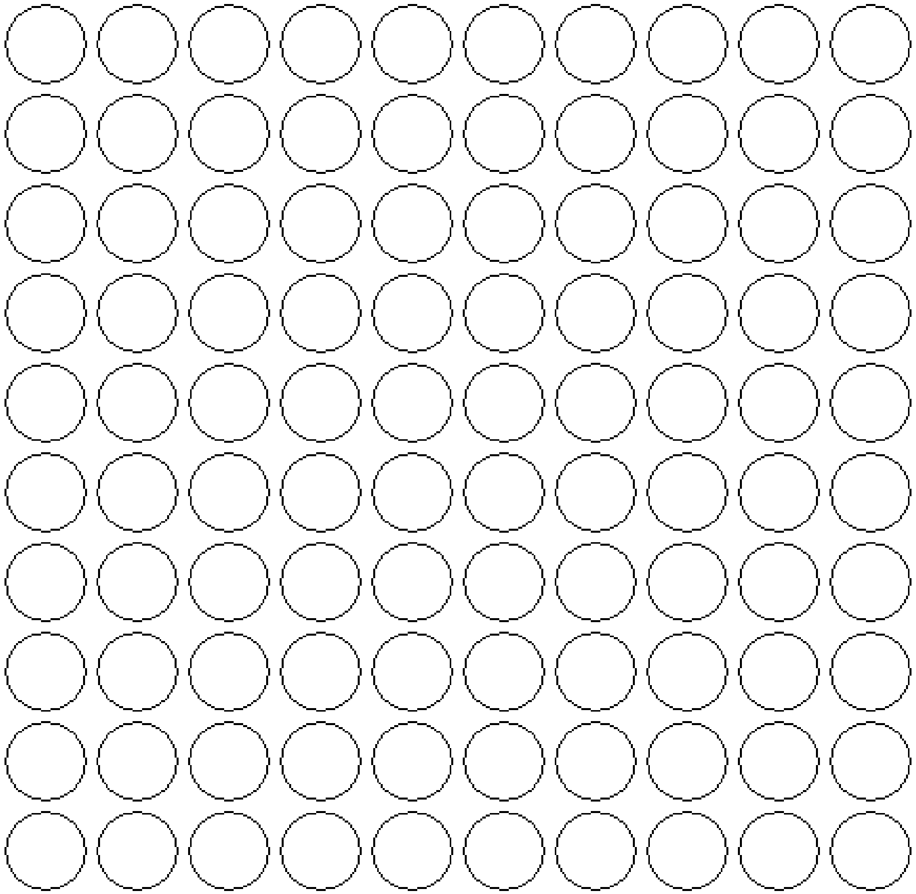


Optimale Basisfunktionen

- Soweit schien alles etwas zu gut um wahr zu sein
- Hier ist der Pferdefuß: die Anzahl "sinnvoller" Basisfunktionen steigt exponentiell an mit der Anzahl der Eingangsvariablen
- Will ich z.B. K RBF's pro Dimension spendieren, benötige ich für M Dimensionen K^M RBFs
- ähnlich schnell steigt die Anzahl sinnvoller Polynome mit der Dimensionalität
- *Die zentrale Frage zur Lösung nichtlinearer Probleme: wie erhalte ich eine kleine Anzahl problemangepasster Basisfunktionen*

10 RBFs in einer Dimension





100 RBFs in zwei Dimensionen

Modellselektion: Polynomiale Basisfunktionen

- Zunächst wird ein lineares Modell mit den Eingangsvariablen geformt
- Es werden polynomiale Basisfunktionen hinzugenommen und es wird geprüft, welche signifikant das Modell verbessern
- Besonders geeignet für Klassifikationsaufgaben (Polynomklassifikatoren: Siemens-Dematic OCR, J. Schürmann):
  - Dimensionsreduktion durch PCA
  - Begrenzte Dimensionserhöhung durch Einführung signifikanter Polynome
  - Lineare Klassifikation

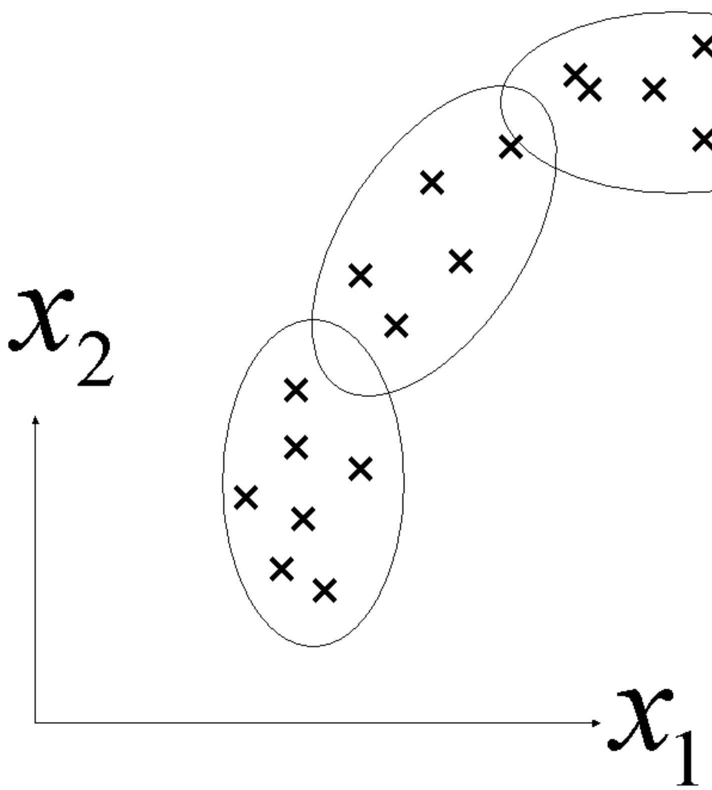
Modellselektion: RBFs

- Es kann sinnvoll sein, die Daten im Eingangsraum zu Gruppieren (Clustern) und die Clusterzentren als Zentren der Gauß-Glocken zu übernehmen

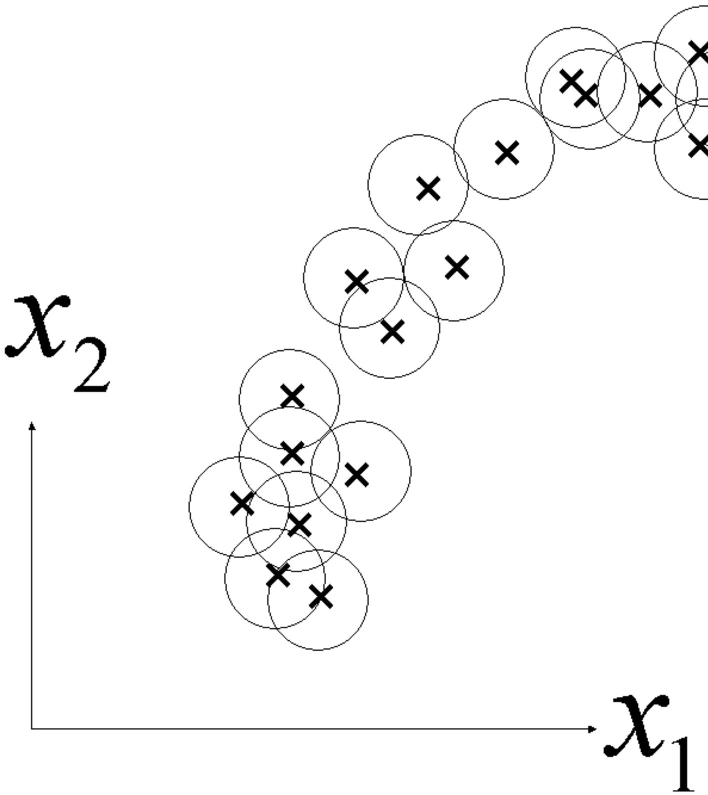
- Die Weiten der Gauss-Glocken ergeben sich z. B. dann aus der Varianz der Verteilung der Daten in den jeweiligen Clustern

- Ein weiteres sinnvolles Vorgehen besteht darin, soviele Gauß-Glocken zu definieren, wie es Datenpunkte gibt $M_\phi = N$. Die Zentren der Gauß-Glocken werden durch die Datenpunkte bestimmt, die Weiten der Gauß-Glocken werden über Kreuzvalidierung bestimmt (siehe Gauss Prozess Regression)

RBFs durch Clustern



RBFs für jeden Datenpunkt



Applikationsspezifische Merkmale

- Die Transformation der "Rohdaten" in eine applikationsspezifische Repräsentation stellt anwendungsspezifische Basisfunktionen dar; besonders wenn die Repräsentation hoch dimensional ist
  - Gegeben ein Bild mit $256 \times 256 = 65536$ Pixeln als Eingangsvektor und einem linearen Klassifikator ist eine Gesichtserkennung aussichtslos
  - Basierend auf weniger als 100 geeigneten Merkmalsvektoren und einem linearen Klassifikator kann man hingegen gut Ergebnisse erzielen
- Die Ableitung von oft einer großen Anzahl von Merkmalen für Dokumente, Bilder, Gen Sequenzen, ... ist ein sehr aktives Forschungsgebiet
- Wenn die Vorverarbeitung schon eine hohe Zahl von Merkmalen liefert, dann ist die Wahrscheinlichkeit hoch, dass eine lineares System das Problem bereits lösen kann.
- Dies ist absolut bemerkenswert: Probleme können in hohen Dimensionen einfacherer werden; vergleiche: Curse of Dimensionality (Bellman)