SCC5871 – Introdução ao Aprendizado de Máquina

Aula 10 – Redes Neurais: Redes RBF e Regressão Não Linear

Prof. Ricardo J. G. B. Campello

PPG-CCMC / ICMC / USP

Aula de Hoje

- Modelos RBF
- Estimação dos Pesos de Redes RBF
  - Mínimos Quadrados
  - Back-propagation
- Determinação das Funções Radiais
- Aplicações em Regressão Não Linear
  - Mapeamentos Estáticos, Séries Temporais e Sistemas Dinâmicos

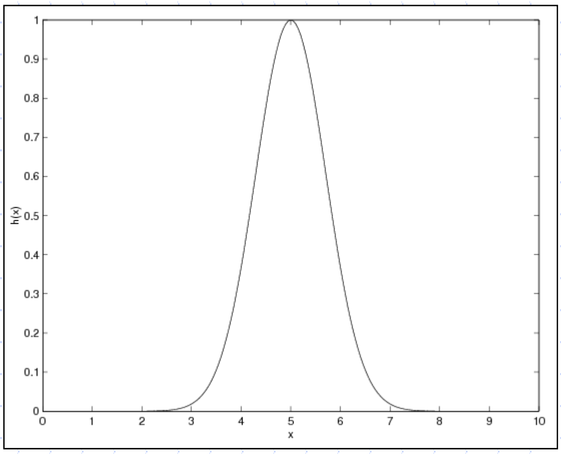
Redes RBF

- Redes RBF (*Radial Basis Functions*) são uma classe de redes com arquitetura *feedforward*
  - Assim como as MLPs, os valores das entradas se propagam na rede em um único sentido
- Essas redes diferem das MLPs especialmente no modelo de neurônio que utilizam
  - Neurônios com *resposta radial* a excitações
  - Modelam o conceito biológico de *campo receptivo*

3

Redes RBF

- Resposta Radial Típica:



4

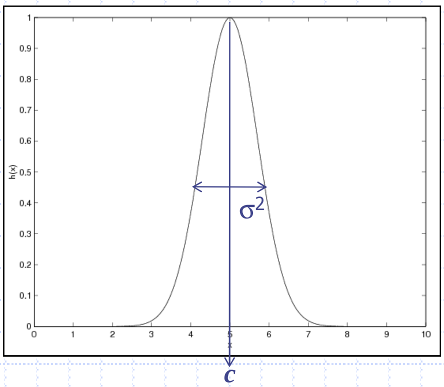
Redes RBF

- Resposta Radial
  - Presentes em alguns tipos de células nervosas:
    - Células auditivas: possuem maior sensibilidade a freqüências *próximas* a um determinado tom
    - Células da retina: maior sensibilidade a excitações luminosas *próximas* ao centro do seu campo receptivo
  - Modelo matemático:
    - **Função de Base Radial**

5

Função de Base Radial

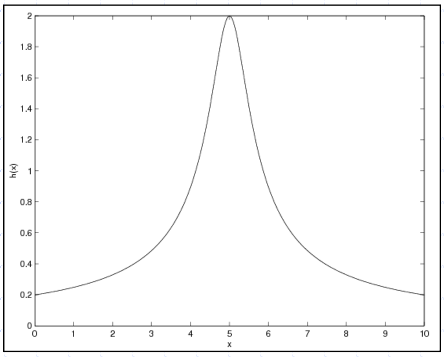
- Diferentes Modelos Matemáticos Possíveis
  - Gaussiana: $\boxed{h(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right)}$



6

Função de Base Radial

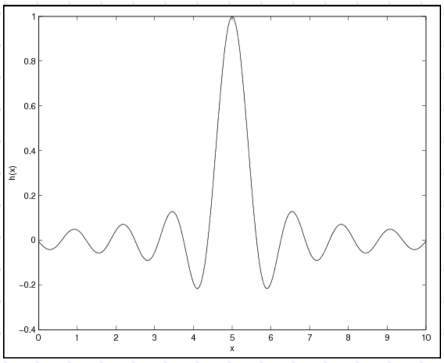
- Diferentes Modelos Matemáticos Possíveis
  - Multi-Quadrática Inversa: $h(x) = \dfrac{1}{\sqrt{(x-c)^2 + \sigma^2}}$



7

Função de Base Radial

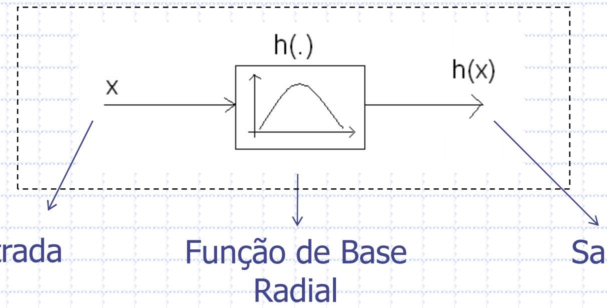
- Diferentes Modelos Matemáticos Possíveis
  - Chapéu Mexicano: $h(x) = \begin{cases} \dfrac{\sin((x-c)/\sigma)}{((x-c)/\sigma)} & \text{se} \quad x \neq c \\ 1 & \text{caso contrário} \end{cases}$



8

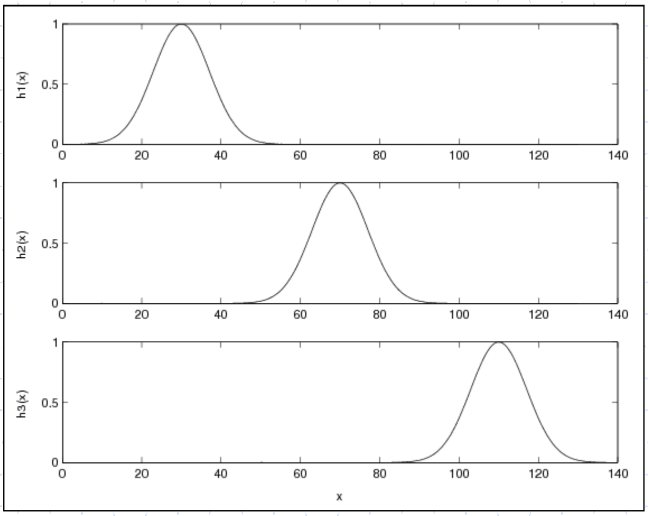
Modelo do Neurônio RBF

- Modelo Básico do Neurônio:



9

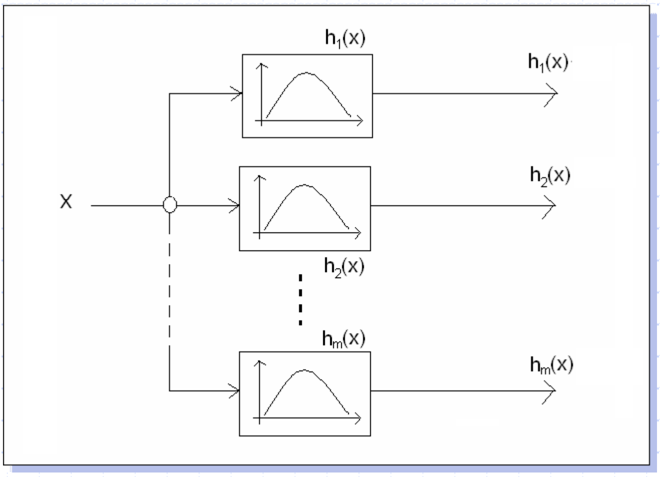
Campos Receptivos Distintos



10

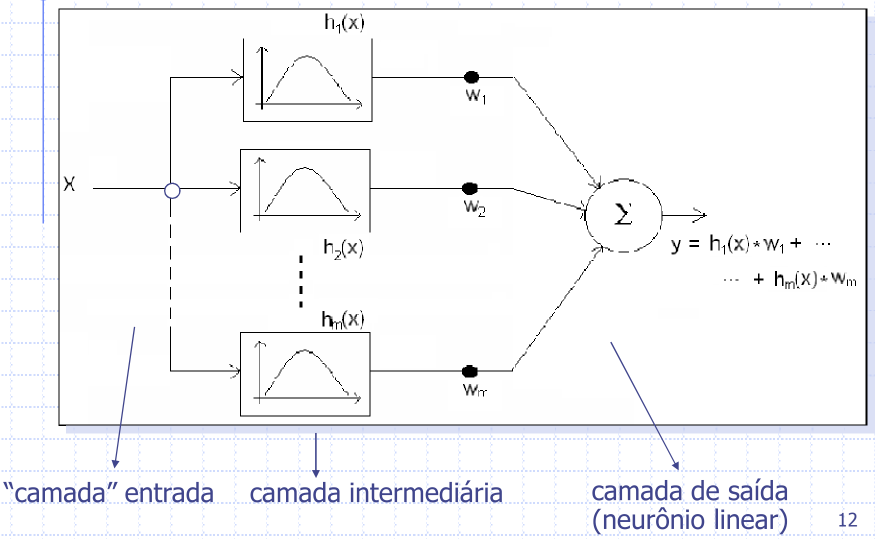
Múltiplos Neurônios

- Neurônios com campos receptivos distintos:



11

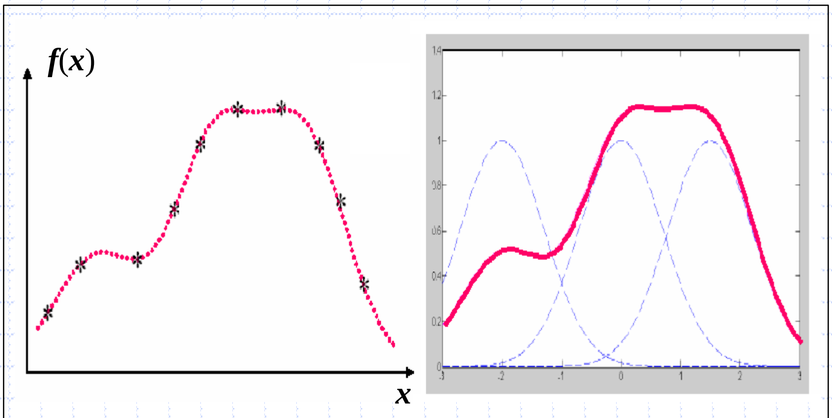
Modelo RBF (1 entrada)



12

Regressão Não Linear

- RBFs são **Aproximadores Universais** !

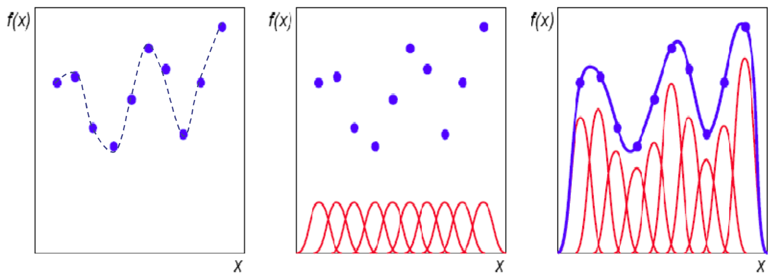


http://people.sabanciuniv.edu/berrin/cs512/lectures/8-nn6-rbf.ppt (09/2008)

13

Regressão Não Linear

- Tais como MLPs, RBFs são **Aproximadores Universais**:
  - Basta aumentar o número de neurônios (funções de base radial) para obter a capacidade de aproximação desejada.
  - Tomando cuidado com super-treinamento (*overfitting*) !

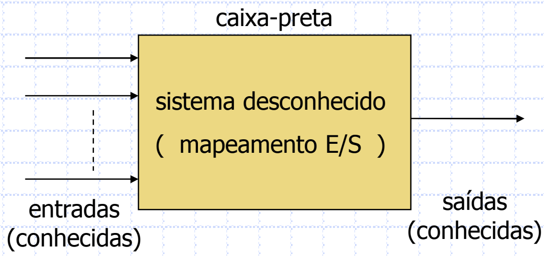


http://people.sabanciuniv.edu/berrin/cs512/lectures/8-nn6-rbf.ppt (09/2008)

14

Regressão Não Linear

- RBFs são **Aproximadores Universais**:
  - Diversos problemas podem ser representados como problemas de aproximação de funções
    - Previsão, Tomada de decisão, Classificação*, etc
    - Por exemplo, operador humano de um processo complexo...

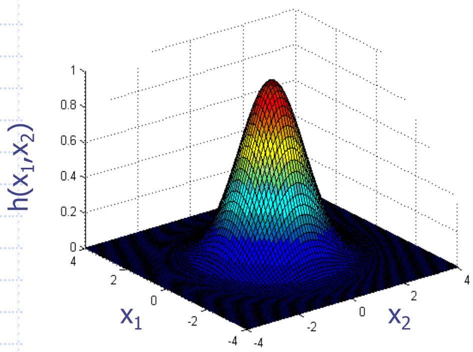


15

Modelo RBF (2 entradas)

- Função Radial para 2 Entradas (Gaussiana):

$$\boxed{h(x_1,x_2)=\exp\left(-\frac{(x_1-c_1)^2}{\sigma_1{}^2}\right)\times\exp\left(-\frac{(x_2-c_2)^2}{\sigma_2{}^2}\right)}$$

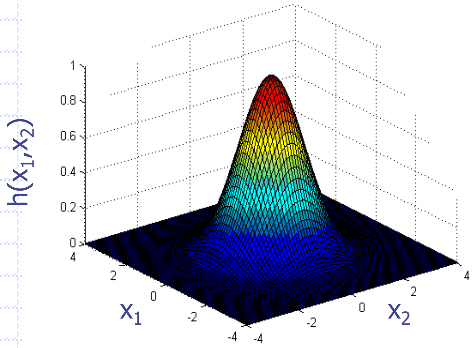


16

Modelo RBF (2 entradas)

- Função Radial para 2 Entradas (Gaussiana):

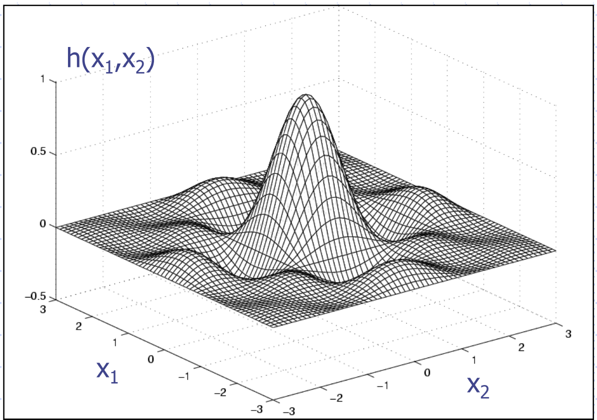
$$h(x_1, x_2) = \exp\left(-(\mathbf{x}-\mathbf{c})^{\mathrm{T}} \cdot \begin{bmatrix} 1/\sigma_1^{\ 2} & 0 \\ 0 & 1/\sigma_2^{\ 2} \end{bmatrix} \cdot (\mathbf{x}-\mathbf{c}) \right) \qquad \begin{cases} \mathbf{x} = [x_1 \quad x_2]^{\mathrm{T}} \\ \mathbf{c} = [c_1 \quad c_2]^{\mathrm{T}} \end{cases}$$



17

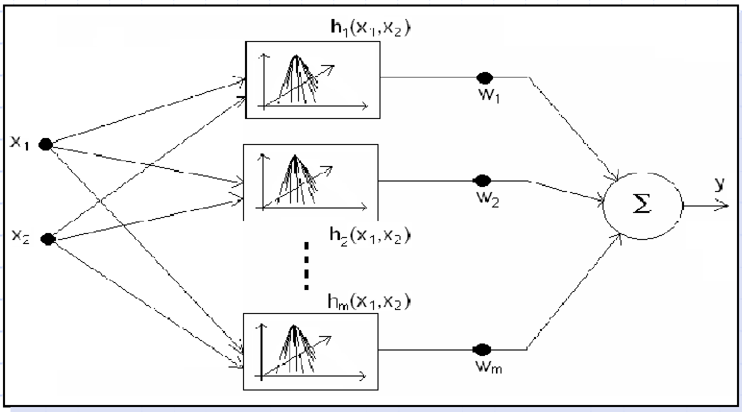
Modelo RBF (2 entradas)

- Função Radial para 2 Entradas (Chapéu Mexicano):



18

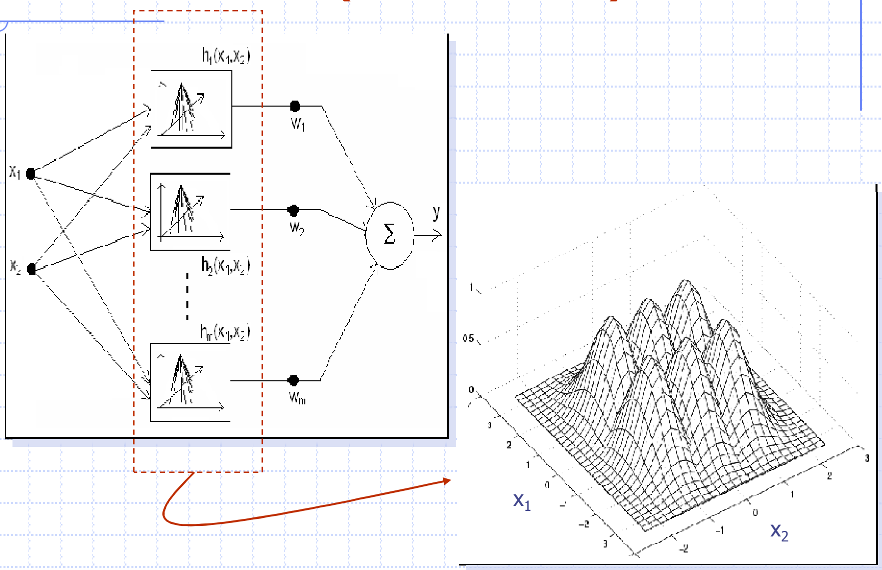
Modelo RBF (2 entradas)



$$y = \sum_{i=1}^{m} w_i \cdot h_i(x_1, x_2)$$

19

Modelo RBF (2 entradas)



Modelo RBF (n entradas)

- Função Radial para n Entradas (Gaussiana):

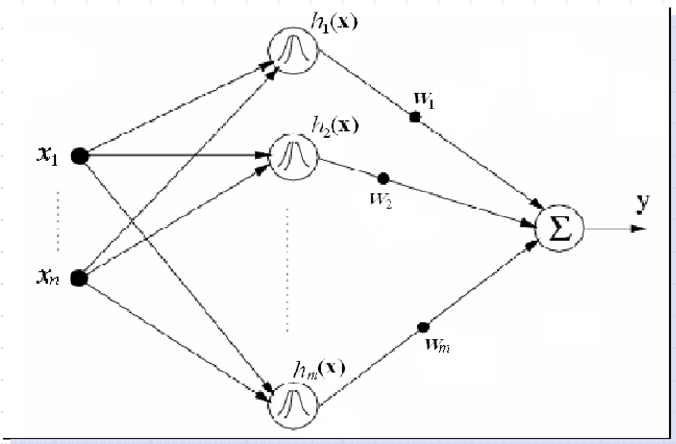
$$\boxed{h(x_1,\cdots,x_n)=\exp\left(-(\mathbf{x}-\mathbf{c})^{\mathrm{T}}\cdot\Phi^{-1}\cdot(\mathbf{x}-\mathbf{c})\right)}$$

$$\begin{cases}\mathbf{x}=[x_1 \quad \cdots \quad x_n]^{\mathrm{T}}\\ \mathbf{c}=[c_1 \quad \cdots \quad c_n]^{\mathrm{T}}\\ \Phi=\mathrm{diag}(\sigma_1^2,\cdots,\sigma_n^2)\end{cases}$$

- Nota: matriz Φ não precisa ser diagonal
  - flexibilidade x complexidade (no. de parâmetros)

21

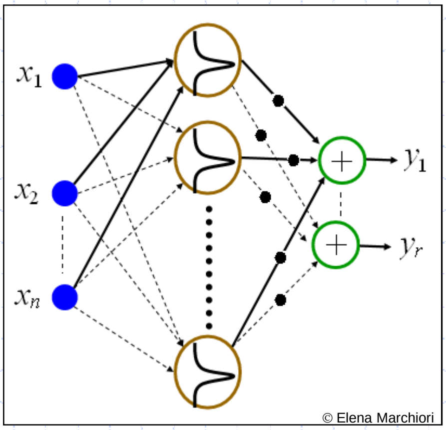
Modelo RBF (n entradas)



$$\boxed{y=\sum_{i=1}^{m} w_i\cdot h_i(x_1,\cdots,x_n) \quad = \quad \mathbf{w}^{\mathrm{T}}\cdot\mathbf{h}}$$

22

Modelo RBF (n entradas / r saídas)



$$\boxed{y_j = \sum_{i=1}^{m} w_{ji} \cdot h_i(x_1, \cdots, x_n)}$$ ou $$\boxed{\mathbf{y} = \mathbf{W} \cdot \mathbf{h}}$$

23

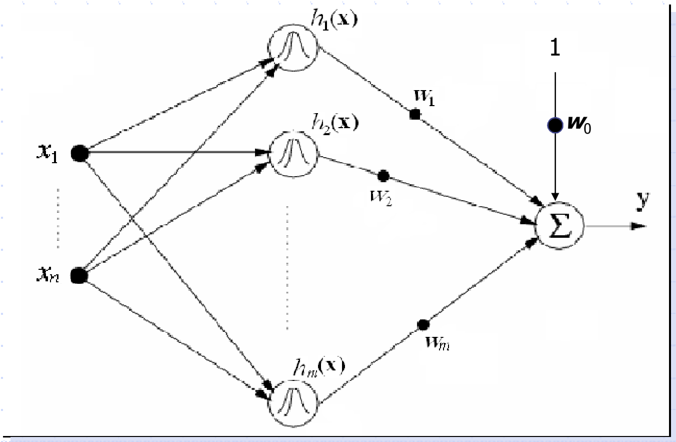
RBF vs. MLP

- Algumas diferenças e semelhanças entre RBFs e MLPs:

MLP	RBF
Aproximadores Universais	Aproximadores Universais
1 ou mais camadas intermediárias	Em geral 1 camada intermediária
Em geral demandam menos neurônios / parâmetros para uma mesma precisão	Em geral demandam mais neurônios / parâmetros para uma mesma precisão
Treinamento mais complexo	Treinamento mais simples
Difícil interpretação, senão impossível	Possível interpretação (se vista como sistema fuzzy)

24

Nota (Adição de Termo Constante)



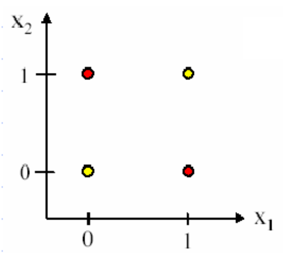
$$y = w_0 + \sum_{i=1}^{m} w_i \cdot h_i(x_1, \cdots, x_n)$$

25

Exercício (classificação)

- Sabemos que o problema XOR consiste em classificar 4 padrões (descritos por 2 variáveis) em 2 classes diferentes, ou seja:

x_1	x_2	Classe
0	0	1
0	1	2
1	0	2
1	1	1



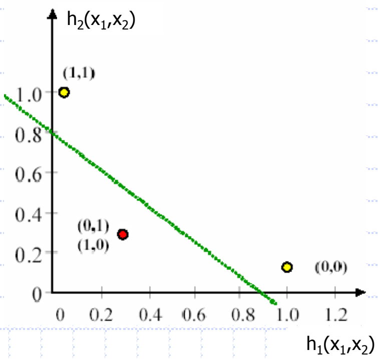
Note que esse problema não é linearmente separável, ou seja, não existe uma função linear que separe as classes no espaço $x_1 \times x_2$

26

Exercício (cont.)

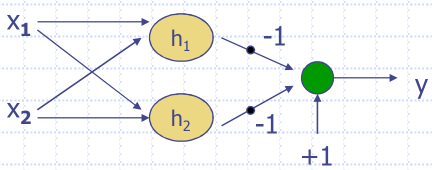
- No entanto, os padrões podem passar a ser linearmente separáveis após serem processados por funções de base radial, desde que essas sejam escolhidas de forma apropriada. P. ex., escolhendo 2 funções Gaussianas h_1 e h_2 com centros dados por [0,0] e [1,1], respectivamente, e desvios padrão unitários para ambas, tem-se:

x_1	x_2	$h_1(x_1,x_2)$	$h_2(x_1,x_2)$
0	0	1	0,1353
0	1	0,3679	0,3679
1	0	0,3679	0,3679
1	1	0,1353	1



Exercício (cont.)

- Considere então a rede RBF com essas funções de base radial e pesos $w_1 = w_2 = -1$ e $w_0 = 1$ (aditivo na saída):



- Escreva analiticamente a expressão da saída da rede em função das entradas x_1 e x_2 e mostre que essa rede é capaz de classificar corretamente os padrões XOR assumindo que esses padrões pertencem à classe 1 se y<0 e à classe 2 se y>0

28

Treinamento de Redes RBF

- Problemas:
  - Dadas as funções de base radial, como determinar o melhor vetor de pesos **w** ?
  - Como determinar bons parâmetros para as funções de base radial (centros e aberturas) ?
    - reduzir no. de funções para dada precisão, ou
    - aumentar precisão para dado no. de funções

29

Treinamento de Redes RBF

- Problemas:
  - **Dadas as funções de base radial, como determinar o melhor vetor de pesos w ?**
  - Como determinar bons parâmetros para as funções de base radial (centros e aberturas) ?
    - reduzir no. de funções para dada precisão, ou
    - aumentar precisão para dado no. de funções

30

Determinação dos Pesos

- Queremos determinar os pesos de uma rede RBF com funções radiais conhecidas para aproximar a função (mapeamento) entre um conjunto de N padrões de entrada $\mathbf{x}_k = [x_{1k} \dots x_{nk}]^T$ e saída $y(\mathbf{x}_k)$ para k=1,...,N
  - Por simplicidade, assume-se aqui que a saída é única
- Exemplo:
  - para N = 100 clientes de um banco, descritos por n = 5 variáveis (salário, idade, tempo de relacionamento com o banco, sexo, estado civil), queremos obter uma rede que, dado um cliente $\mathbf{x}_k$, responda se este cliente possui maior risco de ser caloteiro ($y(\mathbf{x}_k) = 1$) ou não ($y(\mathbf{x}_k) = 0$)

31

Determinação dos Pesos

- Como as m funções radiais são conhecidas, podemos calcular o valor dessas funções para cada padrão $\mathbf{x}_k$
- Representando esses valores em uma matriz tem-se:

$$\mathbf{H} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_m(\mathbf{x}_N) \end{bmatrix}$$

- As saídas desejadas (conhecidas) também podem ser representadas em um vetor:

$$\mathbf{y} = [y(\mathbf{x}_1) \quad \cdots \quad y(\mathbf{x}_N)]^T$$

32

Determinação dos Pesos

- Para uma representação ideal da saída, deseja-se que a rede RBF possua pesos **w** tais que:

$$\begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_m(\mathbf{x}_N) \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} y(\mathbf{x}_1) \\ \vdots \\ y(\mathbf{x}_N) \end{bmatrix}$$

ou seja: $\mathbf{H} \cdot \mathbf{w} = \mathbf{y}$

- Se **H** fosse quadrada, poderíamos multiplicar por $\mathbf{H}^{-1}$ em ambos os lados e resolver o problema como:

$$\mathbf{w} = \mathbf{H}^{-1} \cdot \mathbf{y}$$

33

Determinação dos Pesos

- Podemos obter uma matriz quadrada multiplicando ambos os lados da equação por $\mathbf{H}^{\mathrm{T}}$:

$$\mathbf{H}^{\mathrm{T}} \cdot \mathbf{H} \cdot \mathbf{w} = \mathbf{H}^{\mathrm{T}} \cdot \mathbf{y}$$

- Agora sim podemos multiplicar ambos os lados da equação pela inversa $(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}$ e obter a solução como:

$$\boxed{\mathbf{w} = \left(\mathbf{H}^{\mathrm{T}} \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{T}} \cdot \mathbf{y}}$$

34

Determinação dos Pesos

- Se houver o termo aditivo w_0 na saída, é preciso somente redefinir **H** e **w** de tal forma que:

$$\underbrace{\begin{bmatrix} 1 & h_1(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & h_1(\mathbf{x}_N) & \cdots & h_m(\mathbf{x}_N) \end{bmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} y(\mathbf{x}_1) \\ \vdots \\ y(\mathbf{x}_N) \end{bmatrix}}_{\mathbf{y}}$$

o que não afeta a solução:

$$\boxed{\mathbf{w} = \left(\mathbf{H}^{\mathrm{T}} \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{T}} \cdot \mathbf{y}}$$

35

Determinação dos Pesos

- Algoritmo:
  - Para um conjunto de k = 1, ..., N padrões de entrada e saída, $\{\mathbf{x}_k , y(\mathbf{x}_k)\}$, calcule a matriz **H**, construa o vetor **y** e calcule os pesos da rede segundo a equação:

$$\boxed{\mathbf{w} = \left(\mathbf{H}^{\mathrm{T}} \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{T}} \cdot \mathbf{y}}$$

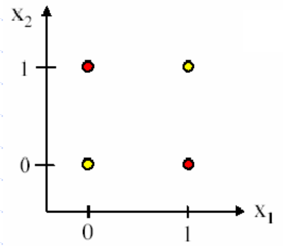
- Nota: É possível demonstrar que, se a representação não for exata, a solução acima é aquela que minimiza o erro quadrático entre a saída da rede e a saída desejada (**Mínimos Quadrados**) !

36

Exercício (classificação)

- Sabemos que o problema XOR consiste em classificar N = 4 padrões (descritos por n = 2 variáveis) em 2 classes diferentes:

k	x_1	x_2	y (classe)
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0



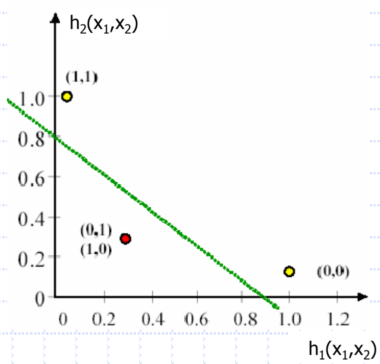
Note que esse problema não é linearmente separável, ou seja, não existe uma função linear que separe as classes no espaço $x_1 \times x_2$

37

Exercício (cont.)

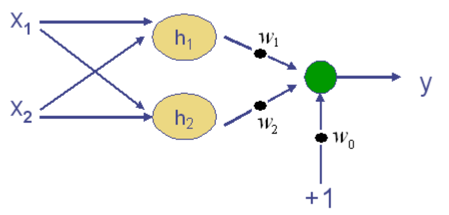
- No entanto, os padrões podem passar a ser linearmente separáveis após serem processados por funções de base radial, desde que essas sejam escolhidas de forma apropriada. P. ex., escolhendo 2 funções Gaussianas h_1 e h_2 com centros dados por [0,0] e [1,1], respectivamente, e desvios padrão unitários para ambas, tem-se:

x_1	x_2	$h_1(x_1,x_2)$	$h_2(x_1,x_2)$
0	0	1	0,1353
0	1	0,3679	0,3679
1	0	0,3679	0,3679
1	1	0,1353	1



Exercício (cont.)

- Considere então a rede RBF com essas funções de base radial e pesos w_1 e w_2, além do peso adicional w_0 (aditivo na saída), conforme abaixo:



- Obtenha os valores dos pesos w_0, w_1 e w_2 para que a rede consiga indicar precisamente em sua saída a classe correta (0 ou 1) de cada um dos 4 padrões de entrada do problema XOR.
  - Apresente a solução passo a passo, em detalhes e de forma justificada !

39

Determinação das Funções Radiais

- **Método 1** (Heurística Simples):
  - Distribuir as funções de maneira homogênea sobre o domínio das variáveis de entrada
  - Por exemplo, se a rede possui duas entradas, x_1 e x_2, que assumem valores no intervalo [-10,+10], então o domínio das funções são todos os pontos (x_1,x_2) tais que $x_1 \in [-10,+10]$ e $x_2 \in [-10,+10]$

40

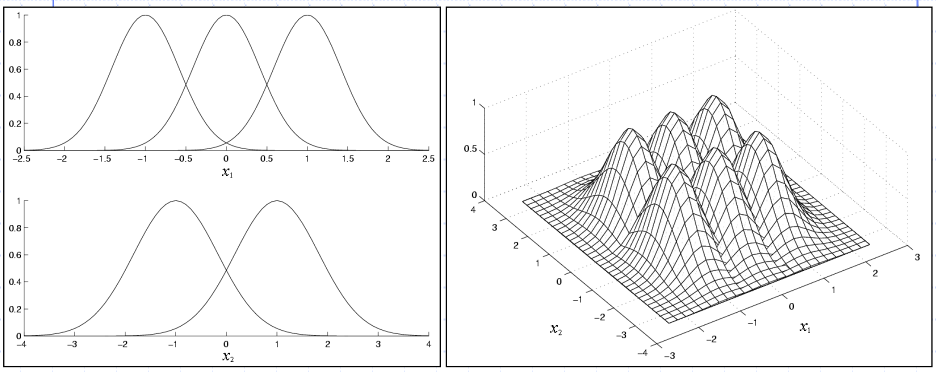
Determinação das Funções Radiais

- **Método 1** (cont.):
  - Os centros das funções com relação a cada variável (dimensão) podem ser distribuídos uniformemente (eqüidistantes) ao longo do domínio daquela var.
  - As aberturas das funções (desvios padrão) com relação a cada variável podem ser feitas iguais à distância entre dois centros consecutivos.

41

Determinação das Funções Radiais

- **Método 1** (exemplo):



42

Determinação das Funções Radiais

- **Método 2** (**back-propagation**):
  - Análogo ao treinamento de MLPs
  - A cada iteração ajusta-se o conjunto de parâmetros (centros e aberturas das funções) no sentido de minimizar o erro entre a saída da rede e a saída desejada para um conjunto de padrões
  - Usualmente tenta-se minimizar: $J = \frac{1}{N}\sum_{k=1}^{N}\left(y(\mathbf{x}_k) - y_{\mathrm{RBF}}(\mathbf{x}_k)\right)^2$
  - Demanda o cálculo do gradiente (derivadas) de J com relação aos parâmetros...
  - **Usualmente aplica-se para refinar o resultado inicial obtido com o método 1 ou método 3 (visto depois)**

43

Exemplo (regressão)

- **Problema**:
  - Aproximar a função f(x) = sen(2x) / exp(x/5)
- N = 21 padrões de entrada e saída (k = 1, ..., 21):

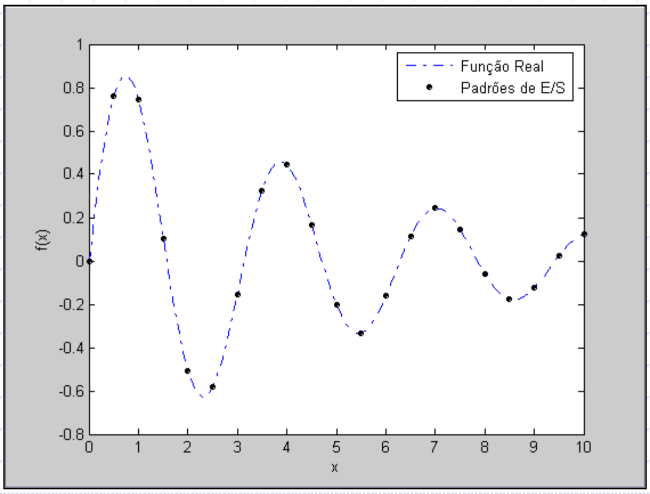
Padrão E/S	x	f(x)
1	0	0
2	0.5	0.7614
3	1.0	0.7445
4	1.5	0.1045
5	2.0	-0.5073
6	2.5	-0.5816
7	3.0	-0.1533
8	3.5	0.3262
9	4.0	0.4445
10	4.5	0.1676

11	5.0	-0.2001
12	5.5	-0.3329
13	6.0	-0.1616
14	6.5	0.1145
15	7.0	0.2443
16	7.5	0.1451
17	8.0	-0.0581
18	8.5	-0.1756
19	9.0	-0.1241
20	9.5	0.0224
21	10.0	0.1236

44

Exemplo (cont.)

- **Problema**: Aproximar a função f(x) = sen(2x) / exp(x/5)

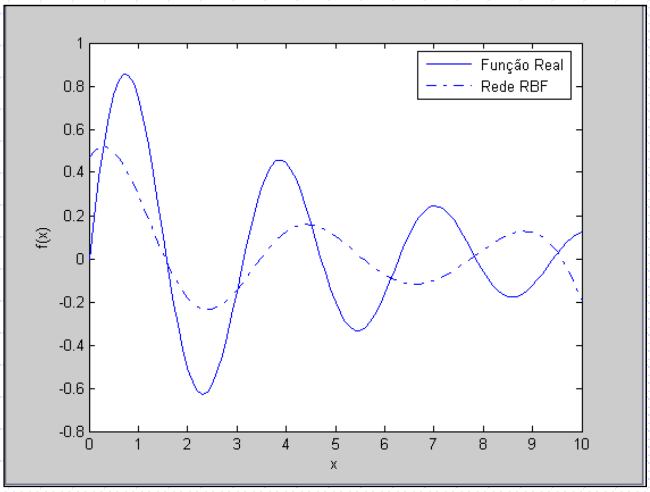


Exemplo (cont.)

- **Solução 1 (Método 1)**:
  - Rede RBF com 6 neurônios (funções Gaussianas)
  - Centros distribuídos de maneira uniforme
    - $c_1 = 0$, $c_2 = 2$, $c_3 = 4$, $c_4 = 6$, $c_5 = 8$, $c_6 = 10$
  - Desvios padrão iguais à distância entre dois centros consecutivos
    - $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = \sigma_5 = \sigma_6 = 2$
  - Apenas os pesos otimizados via Mínimos Quadrados (MQ):
    - 7 pesos (w_0 ... w_6) obtidos via MQ: $\mathbf{w} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}$

Exemplo (cont.)

- **Resultado Solução 1**:

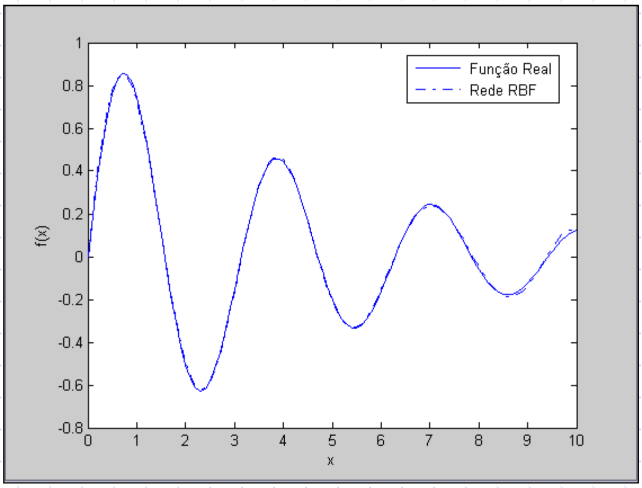


Exemplo (cont.)

- **Solução 2 (Método 1)**:
  - Rede RBF com 11 neurônios
  - Centros distribuídos de maneira uniforme
    - $c_1 = 0,\ c_2 = 1,\ c_3 = 2,\ \ldots,\ c_{11} = 10$
  - Desvios padrão iguais à distância entre dois centros consecutivos
    - $\sigma_1 = \sigma_2 = \ldots = \sigma_{11} = 1$
  - Apenas os pesos otimizados via Mínimos Quadrados (MQ):
    - 12 pesos ($w_0 \ldots w_{11}$) obtidos via MQ: $\mathbf{w} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}$

Exemplo (cont.)

- **Resultado Solução 2**:



49

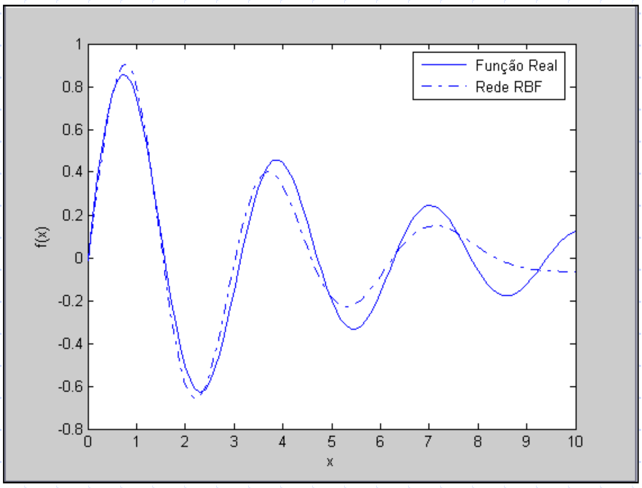
Exemplo (cont.)

- **Solução 3 (Método 1 + 2)**:
  - Rede RBF com 6 neurônios
  - Rede inicialmente configurada conforme Solução 1
  - Sintonia fina realizada com **back-propagation**

50

Exemplo (cont.)

- **Resultado Solução 3**:



51

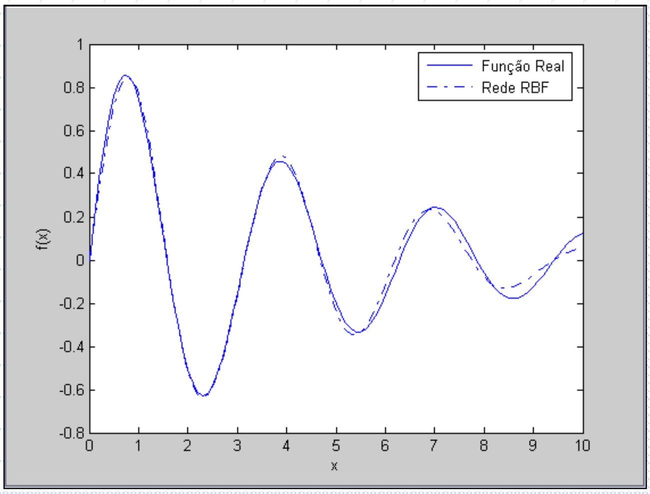
Exemplo (cont.)

- **Solução 4 (Método 1 + 2)**:
  - Rede RBF com 8 neurônios
  - Rede inicialmente configurada de forma análoga à Solução 1
    - Porém com 8 neurônios (Gaussianas)
  - Sintonia fina realizada com **back-propagation**

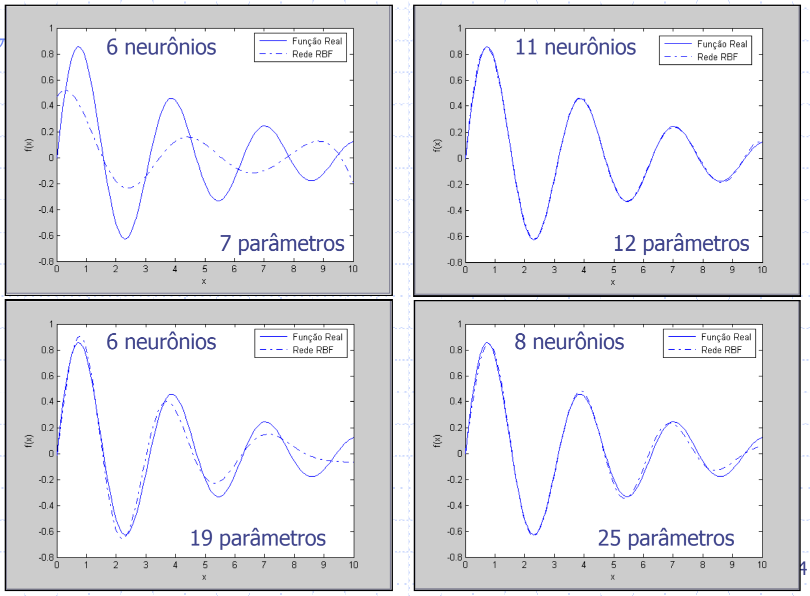
52

Exemplo (cont.)

- **Resultado Solução 4**:



Exemplo (cont.)



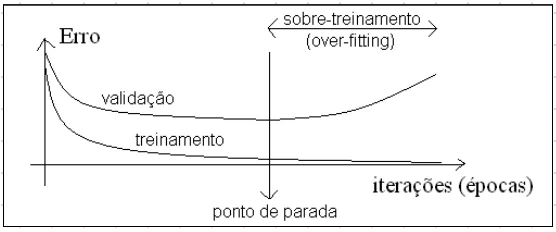
Exemplo (cont.)

- **Comparação**:
  - A heurística de distribuição homogênea das funções radiais (método 1) foi muito eficaz nesse exemplo agindo sozinha, mas não necessariamente é sempre assim
  - Embora a rede com 8 neurônios refinados com back-propagation tenha uma quantidade de parâmetros maior durante a fase de **treinamento**, é um modelo mais simples e compacto do que aquele com 11 neurônios (para **utilização**).
  - A rede com 11 neurônios poderia ficar ainda mais precisa se seus parâmetros também fossem refinados com back-propagation

55

Dicas Básicas

- Usualmente utiliza-se apenas uma parcela dos padrões disponíveis para treinar a rede e reserva-se uma outra parcela para teste ou **validação**
- Uma boa heurística para saber quando interromper o treinamento (sintonia fina) dos parâmetros via back-propagation é observar os erros de ambas as parcelas:

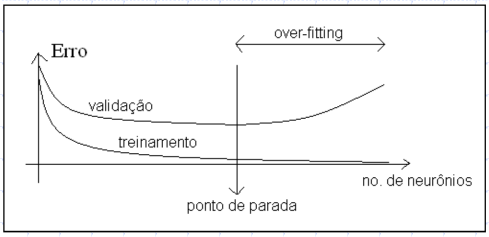


Early Stop

56

Dicas Básicas

- A mesma idéia pode ser usada para determinar quando parar de acrescentar neurônios à rede...



- Uma outra boa dica é normalizar os padrões de E/S de forma que cada variável tenha valores entre –1 e +1
  - Isso minimiza problemas numéricos durante o treinamento

57

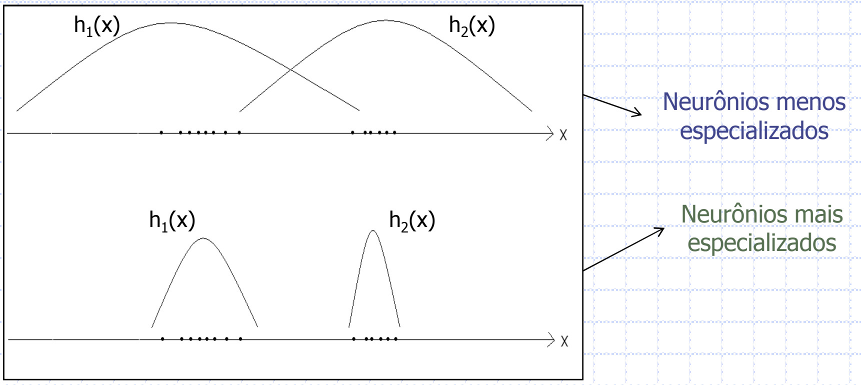
Determinação das Funções Radiais

- **Método 3** (Clustering – Vide Próxima Aula):
  - Agrupar os padrões em grupos (clusters) de padrões mais similares entre si do que aos demais padrões.
  - Associar um neurônio (função radial) para cada grupo de padrões, otimizando a representatividade de cada neurônio / função.
  - Idéia é que cada neurônio responda de forma apropriada e similar a um determinado conjunto de padrões similares
    - Tipicamente (sub)classes em **problemas de classificação**

58

Determinação das Funções Radiais

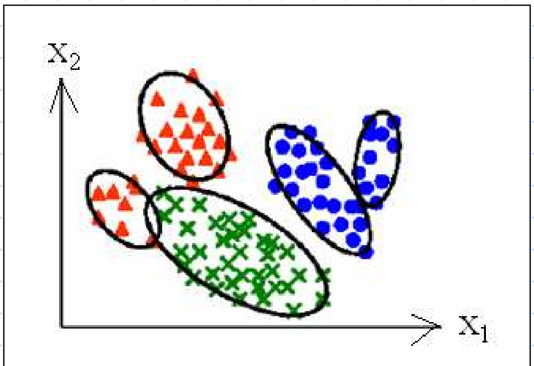
- **Método 3** (cont.):
  - Idéia intuitiva (1 var. de entrada):



59

Determinação das Funções Radiais

- **Método 3** (cont.):
  - Idéia intuitiva (2 vars. de entrada):



http://people.sabanciuniv.edu/berrin/cs512/lectures/8-nn6-rbf.ppt (09/2008)

60

Determinação das Funções Radiais

- **Método 3** (cont.):
  - Um dos algoritmos de agrupamento mais populares é o **K-means**, que agrupa N padrões em K grupos
    - fornece K protótipos (centróides) dos grupos
  - K-means agrupa os N padrões em K grupos com o objetivo de minimizar as distâncias entre os padrões pertencentes a um mesmo grupo e seu protótipo
    - veremos na próxima aula...
  - Mas como aplicar K-means para determinação das funções radiais...?

61

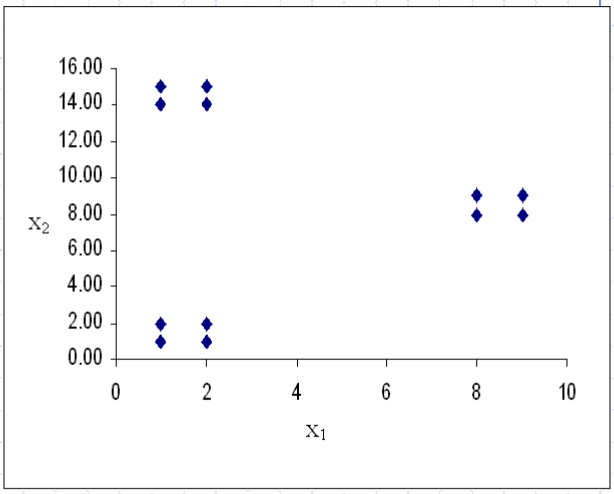
Determinação das Funções Radiais

- **Método 3** (cont.):
  - Executa-se K-means com **K** igual ao número **m** de neurônios (funções radiais) desejado na rede RBF
  - Toma-se o protótipo (centróide) de cada grupo resultante como centro de uma função radial
    - cada componente do centróide é o centro da função radial correspondente na respectiva variável
  - Toma-se os desvios padrão de cada grupo, em cada variável, como o desvio padrão da função radial correspondente na respectiva variável
    - forma simplificada...

62

Método 3 (Exemplo)

padrão (entrada)	x_1	x_2
1	1	2
2	2	1
3	1	1
4	2	2
5	**8**	**9**
6	**9**	**8**
7	**9**	**9**
8	**8**	**8**
9	1	15
10	2	15
11	1	14
12	2	14



Slide cedido pelo Prof. Eduardo Raul Hruschka

Método 3 (Exemplo)

- K-means com K = 3 produziria tipicamente os protótipos (1.5 , 1.5), (8.5 , 8.5), (1.5 , 14.5)
- O terceiro grupo, por exemplo, possui desvios padrão em x_1 e em x_2 respectivamente iguais a:

$$\sigma_1 = \sqrt{\frac{(1-1.5)^2+(2-1.5)^2+(1-1.5)^2+(2-1.5)^2}{4}} = 0.5$$

$$\sigma_2 = \sqrt{\frac{(15-14.5)^2+(15-1.5)^2+(14-14.5)^2+(14-14.5)^2}{4}} = 0.5$$

- Logo, a função radial (Gaussiana) correspondente pode ser definida como:

$$\boxed{h_3(x_1,x_2) = \exp\left(-\frac{(x_1-1.5)^2}{0.5^2}\right)\times\exp\left(-\frac{(x_2-14.5)^2}{0.5^2}\right)}$$

RBFs em Problemas de Regressão

- Em problemas de **regressão não linear**, a mesma arquitetura RBF é usada para modelar uma função F:
  - Na aproximação de **mapeamentos estáticos**
    - $y = F(x_1, x_2, \ldots, x_n)$
    - Exemplo: $y = (1 / x_1) + (x_2)^{1/2}$
  - Em previsão de **séries temporais**
    - $y_t = F(y_{t-i}, y_{t-j}, \ldots, y_{t-k})$
    - Exemplo: $y_t = 10y_{t-3} - (y_{t-5})^3 + \exp(-y_{t-8})$
  - Na modelagem de **sistemas dinâmicos**
    - $y_t = F(u_{t-1}, u_{t-2}, \ldots, u_{t-p}, y_{t-1}, y_{t-2}, \ldots, y_{t-q})$
    - Exemplo: $y_t = 3y_{t-1} + (u_{t-1})^{-1/2}$

Leitura e Exercícios

- Seção 9.1 de (Braga et al., 2007)

ou

- Capítulo 5 de (Haykin, 1999)

66

Referências

- Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 2nd Edition, 1999
- Braga, A. P., Carvalho, A. C. P. L. F., Ludemir, T. B., *Redes Neurais Artificiais: Teoria e Aplicações*, LTC, 2a Edição, 2007
- Kovács, *Redes Neurais Artificiais: Fundamentos e Aplicações*, Collegium Cognitio, 2ª Edição, 1996