Introdução a Sistemas Inteligentes

Tópicos em Redes Neurais III: Redes Neurais RBF – 2ª Parte

Prof. Ricardo J. G. B. Campello

ICMC / USP

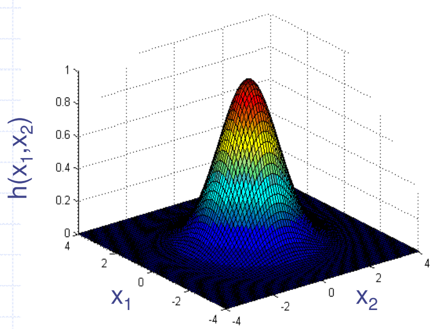
Aula de Hoje

- Revisão de Modelos RBF
- Treinamento de Modelos RBF
  - Estimação dos Pesos
    - Mínimos Quadrados
    - Back-Propagation
  - Estimação das Funções de Base Radial
  - Exemplo em Regressão Não Linear

Modelo RBF (revisão)

- Função Radial para n Entradas (Gaussiana):

$$h(x_1, \ldots, x_n) = \exp\left(-\frac{(x_1 - c_1)^2}{\sigma_1^{\;2}}\right) \times \cdots \times \exp\left(-\frac{(x_n - c_n)^2}{\sigma_n^{\;2}}\right)$$



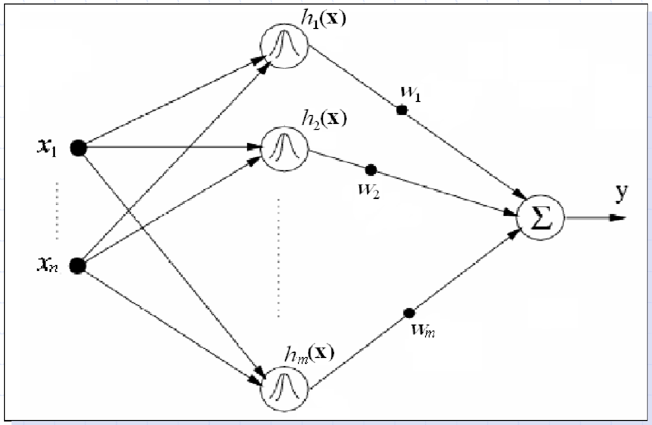
Modelo RBF (revisão)

- Função Radial para n Entradas (Gaussiana):

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

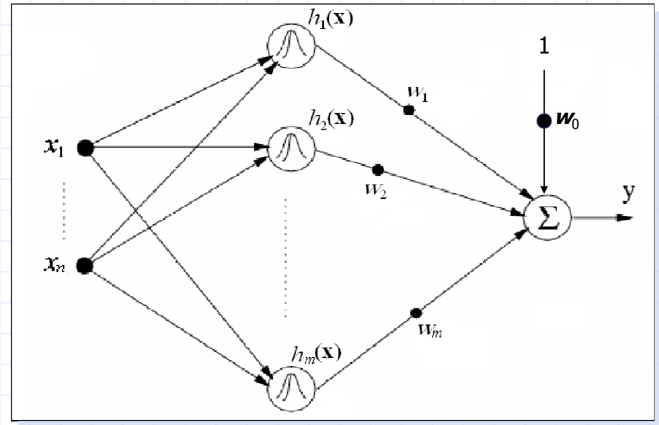
$$\begin{cases} \mathbf{x} = [x_1 \quad \cdots \quad x_n]^{\mathrm{T}} \\ \mathbf{c} = [c_1 \quad \cdots \quad c_n]^{\mathrm{T}} \\ \Phi = \mathrm{diag}\left(\sigma_1^2, \cdots, \sigma_n^2\right) \end{cases}$$

Modelo RBF (revisão)



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

Modelo RBF (revisão)



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

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

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

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 $\mathbf{x}_k = [x_{1k} \ldots x_{nk}]^T$ 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 ($y(\mathbf{x}_k) = 0$)

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:

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

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:

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

Determinação dos Pesos

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

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

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

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

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:

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

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:

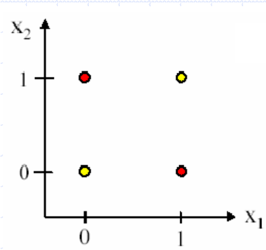
$$\boxed{\mathbf{w} = \left(\mathbf{H}^{\mathrm{T}} \cdot \mathbf{H}\right)^{-1} \cdot \mathbf{H}^{\mathrm{T}} \cdot \mathbf{y}}$$

- 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**) !

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

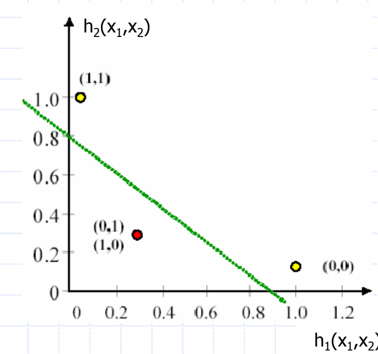


Note que esse problema não é linearmente separável, ou seja, não existe uma função linear que separe as classes no espaço $x_1 \times x_2$

Exercício (cont.)

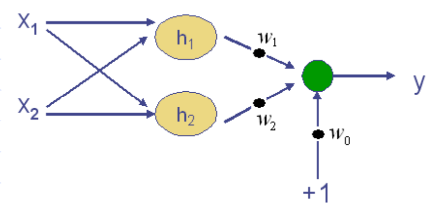
- No entanto, os padrões podem passar a ser linearmente separáveis após serem processados por funções de base radial, desde que essas sejam escolhidas de forma apropriada. P. ex., escolhendo 2 funções Gaussianas h_1 e h_2 com centros dados por [0,0] e [1,1], respectivamente, e desvios padrão unitários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

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

Determinação das Funções Radiais

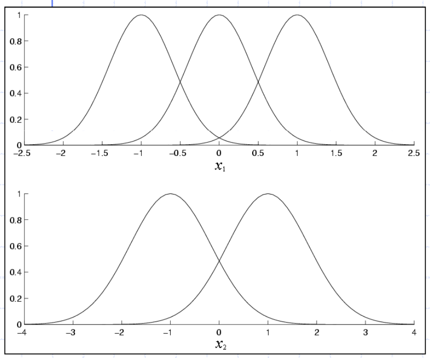
- **Método 1** (Heurística Simples):
  - Distribuir as funções de maneira homogênea sobre o domínio das variáveis de entrada
  - Por exemplo, se a rede possui duas entradas, x_1 e x_2, que assumem valores no intervalo [-10,+10], então o domínio das funções são todos os pontos (x_1,x_2) tais que $x_1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

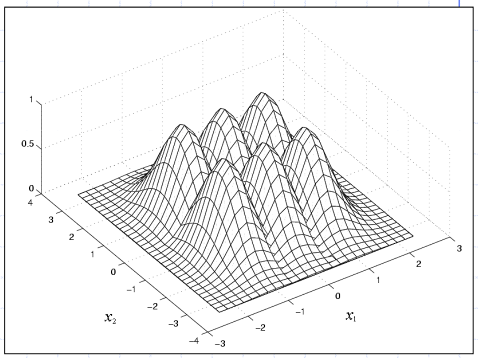
Determinação das Funções Radiais

- **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.

Determinação das Funções Radiais

- **Método 1** (exemplo):





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)**

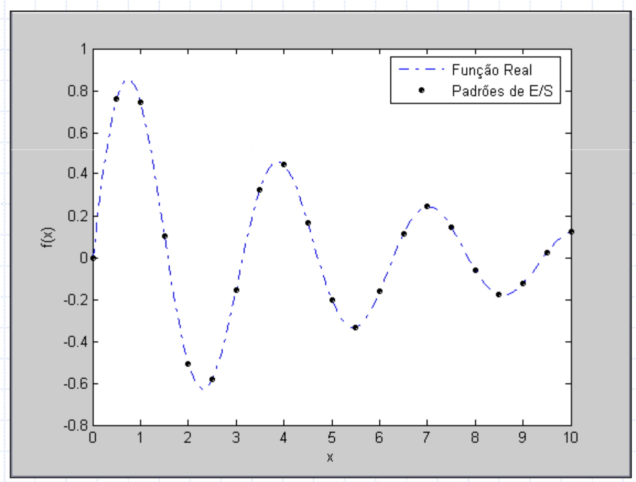
Exemplo (regressão)

- **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

Exemplo (cont.)

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

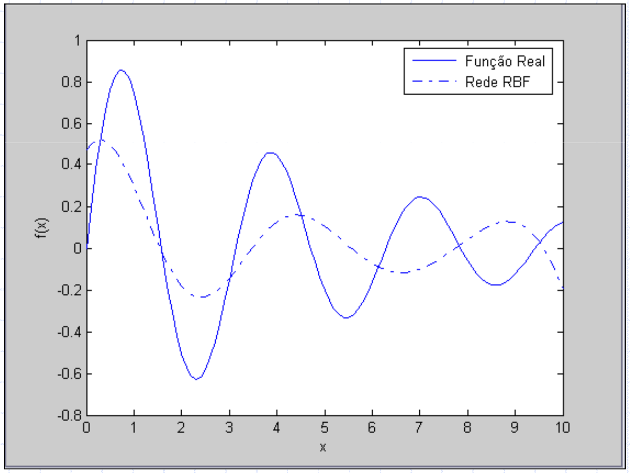


Exemplo (cont.)

- **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**:

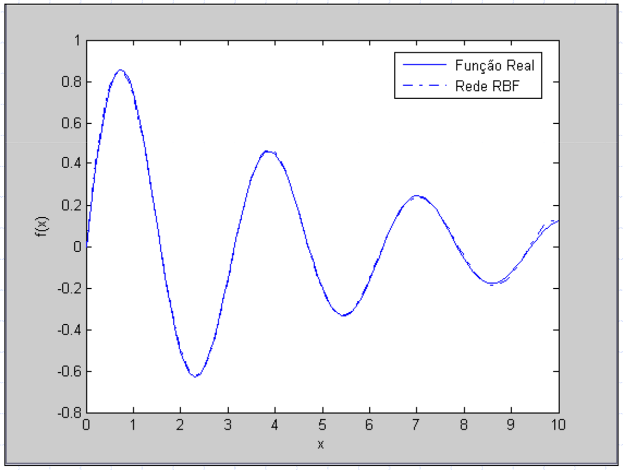


Exemplo (cont.)

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

Exemplo (cont.)

- **Resultado Solução 2**:

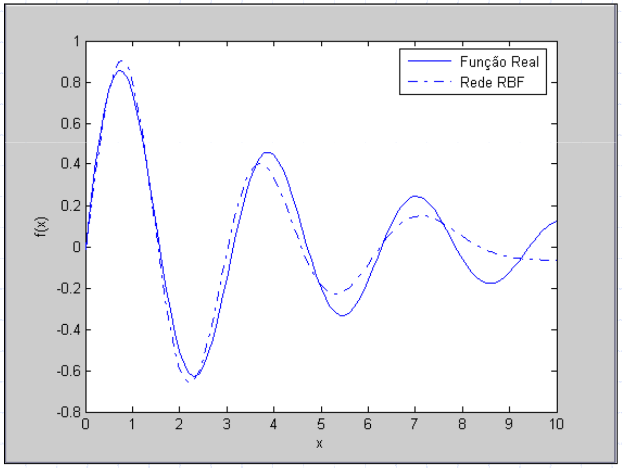


Exemplo (cont.)

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

Exemplo (cont.)

- **Resultado Solução 3**:

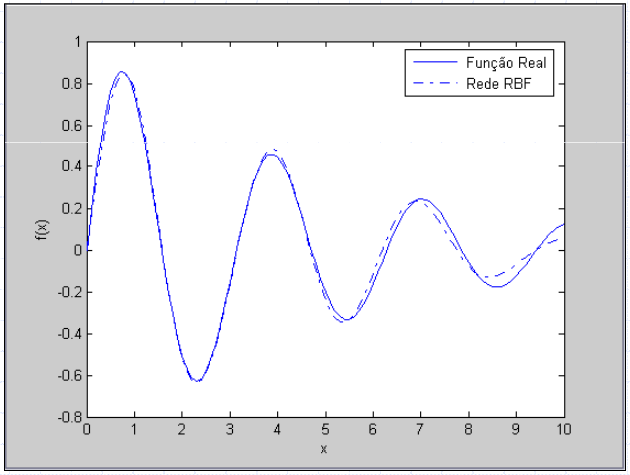


Exemplo (cont.)

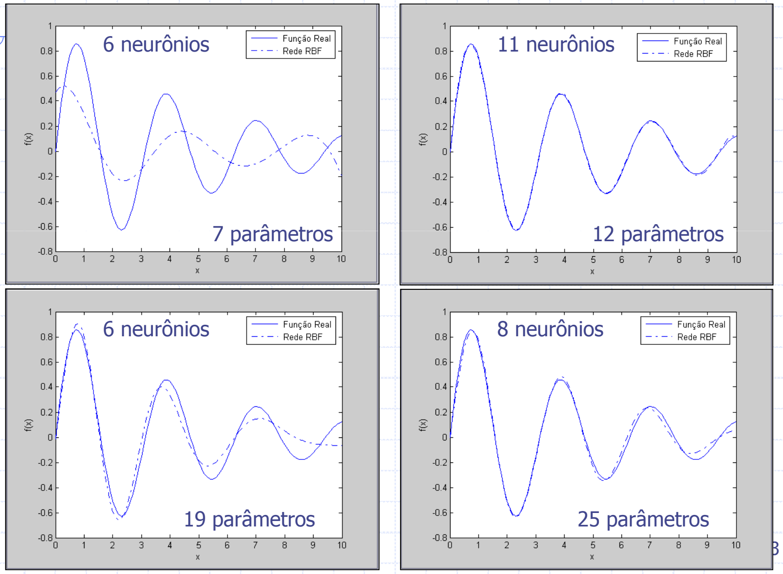
- **Solução 4 (Método 1 + 2)**:
  - Rede RBF com 8 neurônios
  - Rede inicialmente configurada de forma análoga à Solução 1
    - Porém com 8 neurônios (Gaussianas)
  - Sintonia fina realizada com **back-propagation**

Exemplo (cont.)

- **Resultado Solução 4**:



Exemplo (cont.)

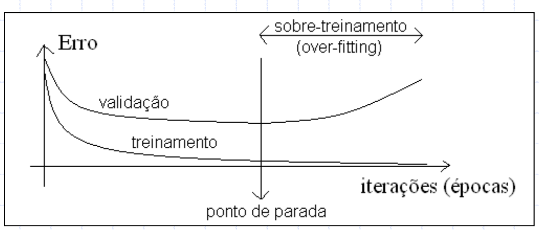


Exemplo (cont.)

- **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

Dicas Básicas

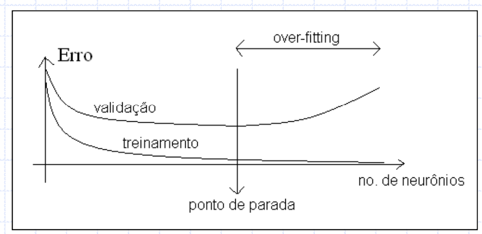
- Usualmente utiliza-se apenas uma parcela dos padrões disponíveis para treinar a rede e reserva-se uma outra parcela para teste ou **validação**
- Uma boa heurística para saber quando interromper o treinamento (sintonia fina) dos parâmetros via back-propagation é observar os erros de ambas as parcelas:



➡ ***Early Stop***

Dicas Básicas

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



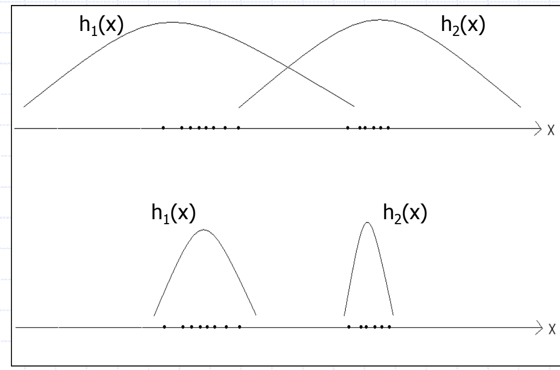
- Uma outra boa dica é normalizar os padrões de E/S de forma que cada variável tenha valores entre –1 e +1
  - Isso minimiza problemas numéricos durante o treinamento

Determinação das Funções Radiais

- **Método 3** (Clustering):
  - 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**

Determinação das Funções Radiais

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

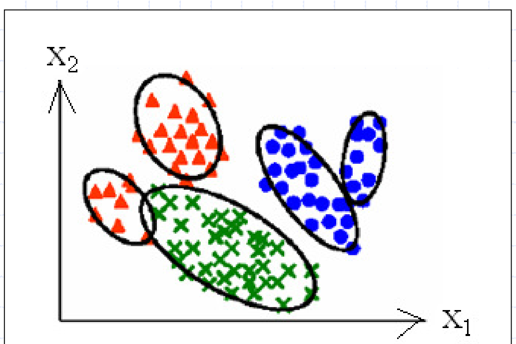


Neurônios menos especializados

Neurônios mais especializados

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)

Determinação das Funções Radiais

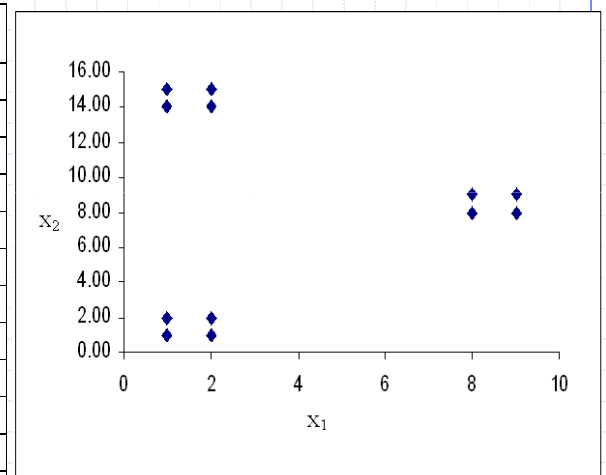
- **Método 3** (cont.):
  - Um dos algoritmos de agrupamento mais populares é o **K-means**, que agrupa N padrões em K grupos
    - fornece K protótipos (centróides) dos grupos
  - K-means agrupa os N padrões em K grupos com o objetivo de minimizar as distâncias entre os padrões pertencentes a um mesmo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

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

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:

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

Exercícios

- Complete o exemplo anterior, ou seja, apresente as funções de base radial h_1 e h_2 correspondentes aos dois primeiros grupos encontrados pelo algoritmo K-means
- Obtenha 9 funções (h_1, ..., h_9), mas agora utilizando o método 1 (heurística simples)
  - Avalie a máxima ativação de cada uma das 9 funções para os dados em mãos e discuta a representatividade delas frente a este cenário

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