Redes Neurais com Função de Ativação de Base Radial

Índice Geral

1 Introdução

- O problema de estimar um mapeamento de entrada-saída a partir de um conjunto de exemplos pode ser visto como o processo de síntese de uma aproximação de uma função multidimensional não-linear, ou como o processo de reconstrução de uma hiper-superfície.
- Neste tópico, veremos que este problema já vem sendo abordado por pesquisadores em diversas áreas e várias nomenclaturas têm sido empregadas na designação da mesma tarefa.
- Entretanto, esta perspectiva de estimar um mapeamento de entrada-saída baseado em exemplos está intimamente relacionada com as técnicas clássicas de aproximação conhecidas na matemática e na estatística.

2 Regressão Paramétrica e Não-Paramétrica

- Há dois grandes problemas na ciência moderna:

1. Mais pessoas do que se aceita como razoável usam terminologias diferentes para resolver os mesmos problemas;
2. Muito mais pessoas usam a mesma terminologia para abordar questões completamente distintas.

Autoria desconhecida

- Diferentes denominações para o problema de estimar uma função a partir de exemplos de estímulo-resposta (entrada-saída):

  1. Regressão (paramétrica/não-paramétrica)
  2. Aproximação de funções
  3. Identificação de sistemas
  4. Aprendizado supervisionado

- Conjunto de treinamento: Estímulo / Entrada / Variáveis independentes
  Resposta / Saída / Variáveis dependentes

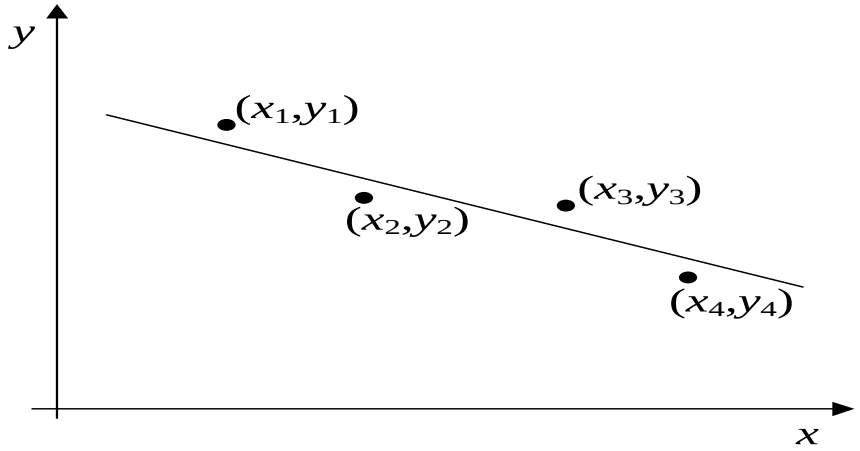
- <u>Regressão paramétrica</u>: A forma do relacionamento funcional entre as variáveis dependentes e independentes é conhecida, mas podem existir parâmetros cujos valores são desconhecidos, embora passíveis de serem estimados a partir do conjunto de treinamento.

- Em problemas paramétricos, os parâmetros livres, bem como as variáveis dependentes e independentes, geralmente têm uma interpretação física.
- Exemplo: Ajuste de uma reta a uma distribuição de pontos

$f(x) = y = ax + b$

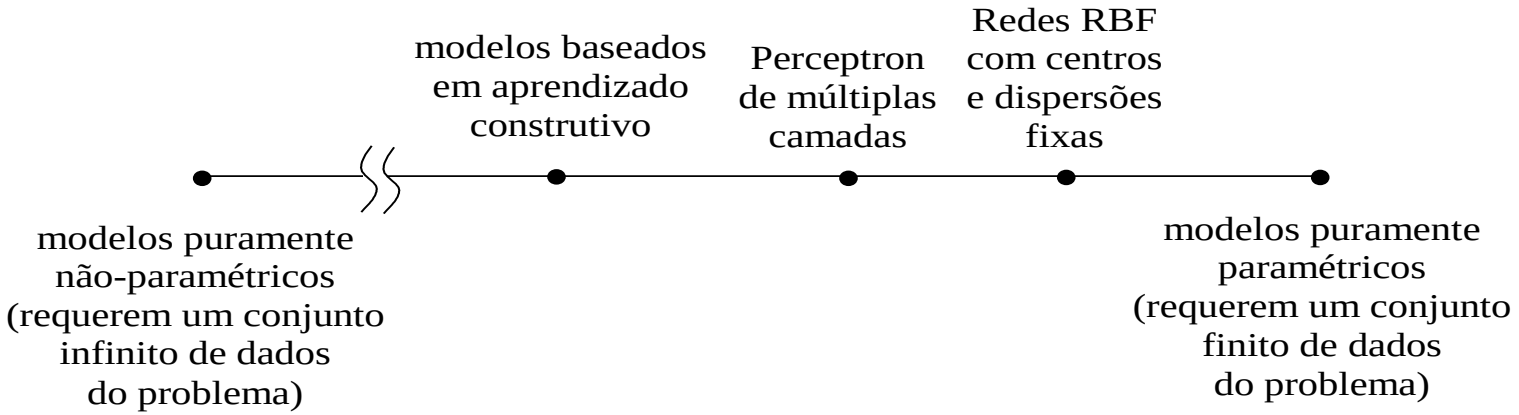
a,b desconhecidos

y: sujeito a ruído



- Regressão não-paramétrica: Sua característica distintiva é a ausência (completa ou quase completa) de conhecimento a priori a respeito da forma da função que está sendo estimada. Sendo assim, mesmo que a função continue a ser estimada a partir do ajuste de parâmetros livres, o conjunto de "formas" que a função pode assumir (classe de funções que o modelo do estimador pode prever) é muito amplo.

- Como consequência, vai existir um número elevado de parâmetros (por exemplo, quando comparado ao número de dados de entrada-saída para treinamento), os quais não mais admitem uma interpretação física isolada.



- Todos os modelos de regressão que não são puramente paramétricos são denominados não-paramétricos ou semi-paramétricos. Esta denominação não deve causar confusão, principalmente levando-se em conta que modelos de regressão puramente não-paramétricos são intratáveis.

- Com base no exposto acima, fica evidente que redes neurais artificiais para treinamento supervisionado pertencem à classe de modelos de regressão não-paramétricos. Sendo assim, os pesos não apresentam um significado físico particular em relação ao problema de aplicação. Mas certamente existem modelos de redes neurais "mais não-paramétricos" que outros.
- Além disso, estimar os parâmetros de um modelo não-paramétrico (por exemplo, pesos de uma rede neural artificial) não é o objetivo primário do aprendizado supervisionado. O objetivo primário é estimar a "forma" da função em uma região compacta do espaço de aproximação (ou ao menos a saída para certos valores desejados de entrada).
- Por outro lado, em regressão paramétrica, o objetivo primário é estimar o valor dos parâmetros, por dois motivos:
  1. A "forma" da função já é conhecida;
  2. Os parâmetros admitem uma interpretação física.

3 Aprendizagem como Aproximação de Funções

- O problema de aprender um mapeamento de um espaço de entrada para um espaço de saída é essencialmente equivalente ao problema de sintetizar uma memória associativa que retorna uma saída aproximada quando apresentada aos dados de entrada, e capaz de *generalizar* quando apresentada a novas entradas.
- As redes neurais do tipo RBF foram desenvolvidas baseadas na *teoria de aproximação de funções*, onde residem também seus aspectos formais.
- A teoria de aproximação trata o problema genérico de aproximar uma função contínua e multivariada $y(\mathbf{x})$ por uma função de aproximação $f(\mathbf{w},\mathbf{x})$ dado um número fixo de parâmetros $\mathbf{w}$ ($\mathbf{x}$ e $\mathbf{w}$ são vetores reais de dimensões q e h, respectivamente).
- Sendo assim, existem dois aspectos importantes do modelo a serem definidos: a função $f(\cdot)$ e os parâmetros $\mathbf{w}$.

- Escolhida uma função $f(\cdot)$ específica, o problema se reduz à determinação do conjunto de parâmetros $\mathbf{w}$ que fornece a melhor aproximação possível de $y(\cdot)$ para o conjunto de exemplos (dados).
- Se $y(\mathbf{X})$ é uma função contínua definida sobre o conjunto $\mathbf{X}$, e $f(\mathbf{w},\mathbf{X})$ é uma função de aproximação que depende continuamente de $\mathbf{w} \in \mathrm{P}$ e $\mathbf{X}$, então o problema de aproximação resume-se a determinar os parâmetros $\mathbf{w}^*$ tais que:

  $d[f(\mathbf{w}^*,\mathbf{X}), y(\mathbf{X})] < d[f(\mathbf{w},\mathbf{X}), y(\mathbf{X})], \qquad \forall\, \mathbf{w} \in \mathrm{P}.$

  onde $d(\cdot,\cdot)$ é uma métrica de distância que permite avaliar a qualidade da aproximação.
- Sendo assim, o problema de aprendizagem corresponde a coletar os dados de entrada (exemplos ou amostras de treinamento) e seus correspondentes valores de saída desejada $\{(\mathbf{x}_1,\mathbf{d}_1),...,(\mathbf{x}_N,\mathbf{d}_N)\}$ e aplicá-los no processo de definição dos parâmetros $\mathbf{w}$ do modelo de aproximação.

- *Generalização* significa estimar a saída d_j para valores de $\mathbf{x}_j$ que não foram utilizados na determinação do modelo.
- A *interpolação* é o limite da aproximação quando não existe ruído nos dados (POGGIO & GIROSI, 1989).
- Sob essa perspectiva de aproximação, o processo de aprendizagem é *mal-condicionado* no sentido de que a informação contida nos dados não é suficiente para reconstruir unicamente o mapeamento em regiões onde os dados não estão disponíveis. Além disso, os dados geralmente são ruidosos.
- Sendo assim, é preciso considerar a priori algumas características do mapeamento como, por exemplo, que ele é *suave*: pequenas variações em alguns parâmetros de entrada causarão pequenas variações na saída.
- Outra restrição que pode ser imposta diz respeito ao tipo de mapeamento como, por exemplo, linear ou polinomial.

- Técnicas que exploram as restrições de suavidade em um problema de aproximação são conhecidas como *técnicas de regularização* (POGGIO & GIROSI, 1989).

4 Modelos de Regressão Lineares

- Um modelo $f(\mathbf{x})$ para uma função $y(\mathbf{x})$ assume a forma:

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j h_j(\mathbf{x}),$$

representando uma combinação linear de um conjunto de m funções fixas, geralmente denominadas funções-base, por analogia com o conceito de vetor gerado a partir de uma combinação linear de vetores-base.

- As funções-base e quaisquer parâmetros que elas possam conter são considerados fixos. Neste caso, se as funções-base $h_j(\cdot)$, j=1,...,m, forem não-lineares, o modelo será não-linear, mas linear nos parâmetros, porque a flexibilidade de $f(\cdot)$, ou seja,

sua habilidade de assumir "formas" diferentes, deriva apenas da liberdade de escolher diferentes valores para os coeficientes da combinação linear, w_j (j=1,...,m).

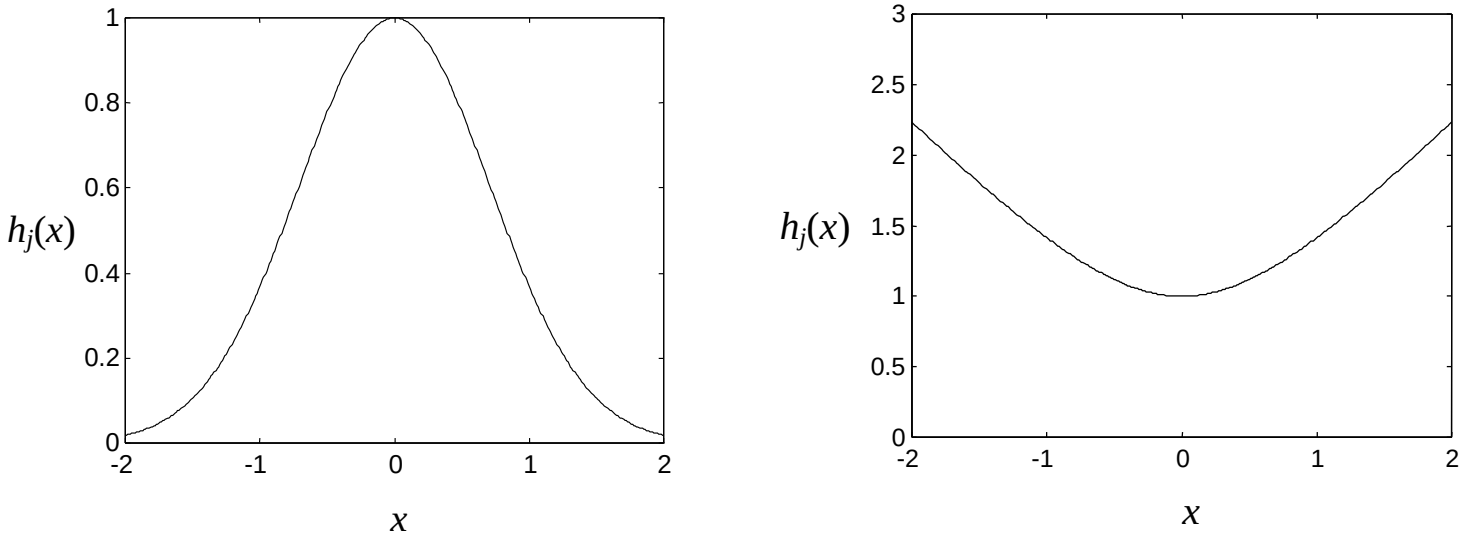
- Caso os parâmetros destas funções-base sejam ajustáveis, e sendo funções não-lineares, o modelo será não-linear e também não-linear nos parâmetros.
- Em princípio, "qualquer" conjunto de funções pode ser escolhido para formar a "base" $\{h_j(\cdot),\ j=1,...,m\}$, embora existam (o que não implica que possam ser facilmente obtidos) conjuntos mais adequados para cada problema e também conjuntos suficientemente flexíveis para apresentarem desempenho adequado frente a uma ampla classe de funções $y(\cdot)$.
- Como um caso particular e de grande interesse prático (em virtude da maior facilidade de manipulação matemática e maior simplicidade na dedução de propriedades básicas), existem os modelos lineares nos parâmetros cujas funções-base são compostas por elementos de uma única classe de funções.

- Exemplos de classes de funções-base:
  - Funções-base polinomiais: $h_j(x) = x^{j-1}$
  - Funções-base senoidais (série de Fourier): $h_j(x) = sen\left(\frac{2\pi j\left(x-\theta_j\right)}{m}\right)$
  - Funções-base logísticas (perceptron): $h_j(\mathbf{x}) = \frac{1}{1+\exp\left(\mathbf{b}_j^T\mathbf{x} - b_{0j}\right)}$
- Nota 1: No exemplo de regressão paramétrica apresentado no início deste tópico, foi utilizada a função $f(x) = ax + b$, que representa um modelo com funções-base polinomiais $h_1(x) = 1$ e $h_2(x) = x$, e coeficientes $w_1 = b$ e $w_2 = a$.
- Nota 2: O grande atrativo apresentado pelos modelos lineares nos parâmetros está na possibilidade de obter o conjunto de coeficientes da combinação linear de forma fechada, através da aplicação de técnicas de solução baseadas em quadrados mínimos. O mesmo não se aplica (na maioria absoluta dos casos) a modelos não-lineares, os quais requerem processos numéricos iterativos (técnicas de otimização não-linear) para se obter a solução.

5 Função de Base Radial

- Uma função de ativação de base radial é caracterizada por apresentar uma resposta que decresce (ou cresce) monotonicamente com a distância a um ponto central.
- O centro e a taxa de decrescimento (ou crescimento) em cada direção são alguns dos parâmetros a serem definidos. Estes parâmetros devem ser constantes caso o modelo de regressão seja tomado como linear nos parâmetros ajustáveis.
- Uma função de base radial monotonicamente decrescente típica é a função gaussiana, dada na forma:
  - $h_j(x) = \exp\left(-\frac{(x - c_j)^2}{r_j^2}\right)$, para o caso escalar (veja Figura 1(a));
- Uma função de base radial monotonicamente crescente típica é a função multiquádrica dada na forma:

- $h_j(x) = \dfrac{\sqrt{r_j^2 + (x - c_j)^2}}{r_j}$, para o caso escalar (veja Figura 1(b));



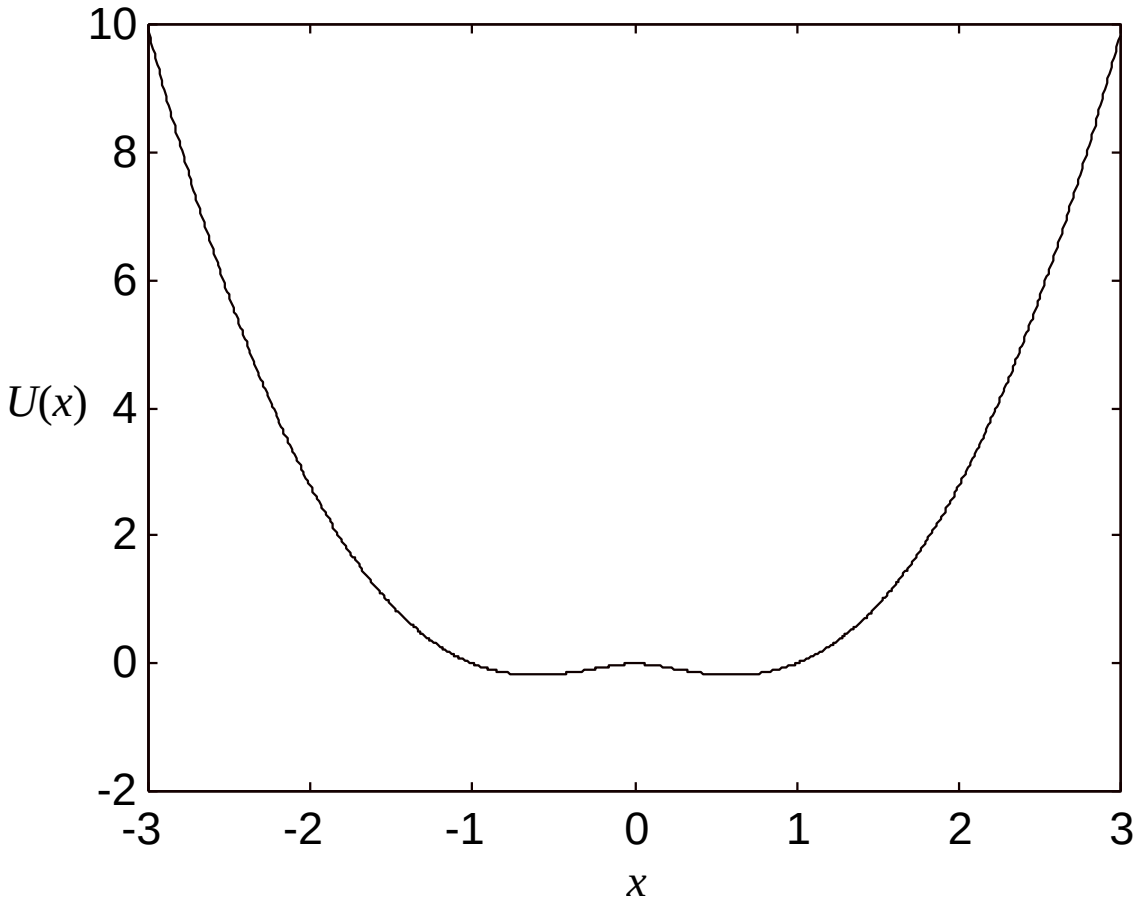
(a) (b)

Figura 1 – Exemplos de funções de base radial monovariáveis, com $c_j = 0$ e $r_j = 1$

- Também apresenta propriedades interessantes a função *thin plate spline*. Considerando uma e duas variáveis, ela assume a forma:

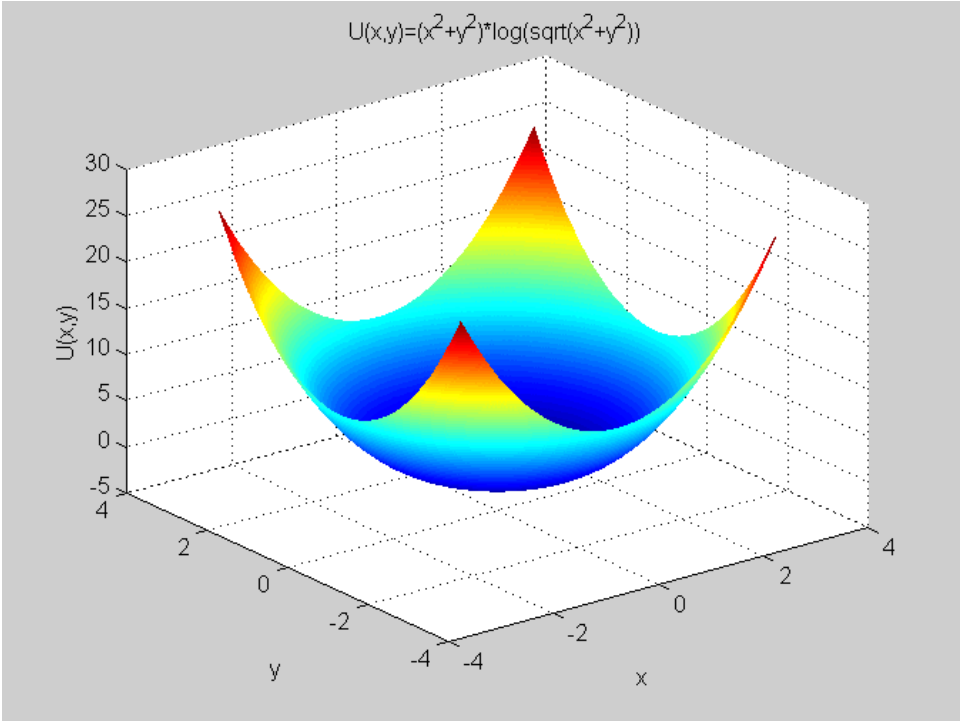
$$h_j(x) = U(x) = (x - c_j)^2 \log(|x - c_j|)$$

$$U(x) = r^2 \log(r) = x^2 \log(|x|)$$



Thin plate spline considerando $c_j = 0$.

$$h_j(x,y) = \left((x-x_j)^2 + (y-y_j)^2\right)\log\left(\sqrt{(x-x_j)^2 + (y-y_j)^2}\right)$$



Thin plate spline considerando $(x_j, y_j) = (0,0)$.

- No caso multidimensional e tomando a função gaussiana, $h_j(\mathbf{x})$ assume a forma:

$$h_j(\mathbf{x}) = \exp\left(-\left(\mathbf{x}-\mathbf{c}_j\right)^T \Sigma_j^{-1}\left(\mathbf{x}-\mathbf{c}_j\right)\right) \qquad (1)$$

onde $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ é o vetor de entradas, $\mathbf{c}_j = \begin{bmatrix} c_{j1} & c_{j2} & \cdots & c_{jn} \end{bmatrix}^T$ é o vetor que define o centro da função de base radial e a matriz Σ_j é definida positiva e diagonal, dada por:

$$\Sigma_j = \begin{bmatrix} \sigma_{j1} & 0 & \cdots & 0 \\ 0 & \sigma_{j2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{jn} \end{bmatrix},$$

de modo que $h_j(\mathbf{x})$ pode ser expandida na forma:

$$h_j(\mathbf{x}) = \exp\left(-\frac{\left(x_1 - c_{j1}\right)^2}{\sigma_{j1}} - \frac{\left(x_2 - c_{j2}\right)^2}{\sigma_{j2}} - \ldots - \frac{\left(x_n - c_{jn}\right)^2}{\sigma_{jn}} \right). \qquad (2)$$

- Neste caso, os elementos do vetor $\sigma_j = \left[\sigma_{j1} \quad \sigma_{j2} \quad \cdots \quad \sigma_{jn}\right]^T$ são responsáveis pela taxa de decrescimento da gaussiana junto a cada coordenada do espaço de entrada, e o argumento da função exponencial é uma norma ponderada da diferença entre o vetor de entrada e o centro da função de base radial.
- A matriz Σ_j pode ser não-diagonal, mas ainda simétrica e definida positiva. No entanto, por envolver muitos parâmetros, geralmente não é utilizada.

6 Rede Neural RBF (*Radial Basis Function Neural Network*)

- As funções de base radial (são funções não-lineares) podem ser utilizadas como funções-base em qualquer tipo de modelo de regressão não-linear (linear ou não-linear nos parâmetros) e, particularmente, como função de ativação de qualquer tipo de rede multicamada.
- O fato do modelo de regressão resultante ser linear ou não-linear nos parâmetros se deve à possibilidade ou não de se ajustar os centros e as dispersões das funções.

- As redes neurais com função de ativação de base radial (RBF) apresentam três diferenças principais em relação às redes tipo perceptron multicamadas:
  - ✓ Elas sempre apresentam uma única camada intermediária;
  - ✓ Neurônios de saída são sempre lineares;
  - ✓ Os neurônios da camada intermediária têm uma função de base radial como função de ativação, ao invés de uma função sigmoidal ou outras.
- Como exposto acima, se apenas os pesos da camada de saída formarem o conjunto de parâmetros ajustáveis, então a rede neural é linear nos parâmetros. Caso contrário, ou seja, quando os centros $\mathbf{c}_j$ e as matrizes Σ_j, $j = 1,...,n$, também são ajustáveis, a rede neural é não-linear nos parâmetros, admitindo o próprio algoritmo de retro-propagação do erro para o processo de ajuste via treinamento supervisionado, como feito no caso do perceptron multicamadas, embora aqui os mínimos locais tenham uma influência bem maior e sugere-se evitar este mecanismo de ajuste.

- A arquitetura da rede é apresentada na Figura 2, para o caso de uma única saída, resultando no seguinte mapeamento de entrada-saída:

$$y = \sum_{j=1}^{m} w_j h_j(\mathbf{x})$$

- Caso $\mathbf{c}_j$ e Σ_j, $j = 1,...,n$, sejam ajustáveis, a saída assume a forma:

$$y = \sum_{j=1}^{m} w_j h_j(\mathbf{c}_j, \Sigma_j, \mathbf{x}).$$

- Substituindo as formas compactas e expandidas de $h_j(\mathbf{x})$, dadas respectivamente pelas equações (1) e (2), resultam:

$$y = \sum_{j=1}^{m} w_j \exp\left(-\left(\mathbf{x}-\mathbf{c}_j\right)^T \Sigma_j^{-1}\left(\mathbf{x}-\mathbf{c}_j\right)\right)$$

e

$$y = \sum_{j=1}^{m} w_j \exp\left(-\frac{\left(x_1-c_{j1}\right)^2}{\sigma_{j1}} - \frac{\left(x_2-c_{j2}\right)^2}{\sigma_{j2}} - ... - \frac{\left(x_n-c_{jn}\right)^2}{\sigma_{jn}}\right)$$

- Uma versão para múltiplas saídas é apresentada na Figura 3.

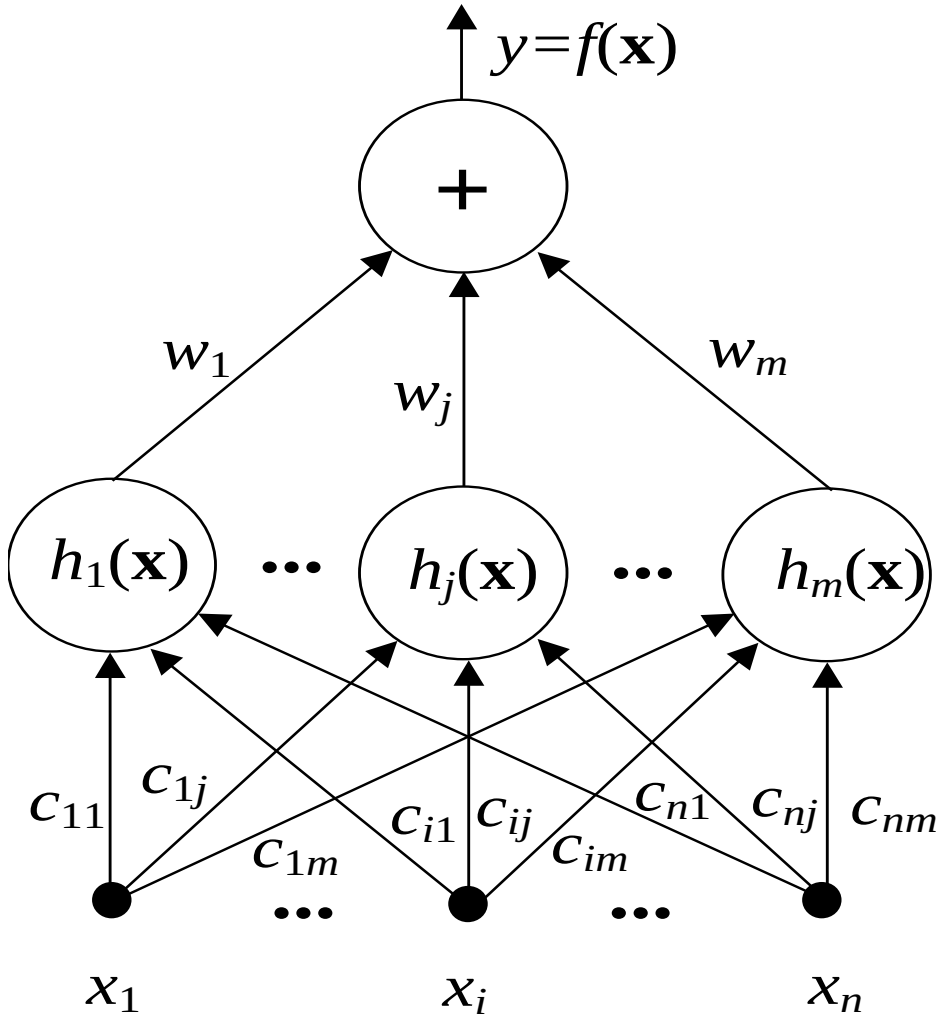


Figura 2 – Rede neural de base radial (BROOMHEAD & LOWE, 1988)

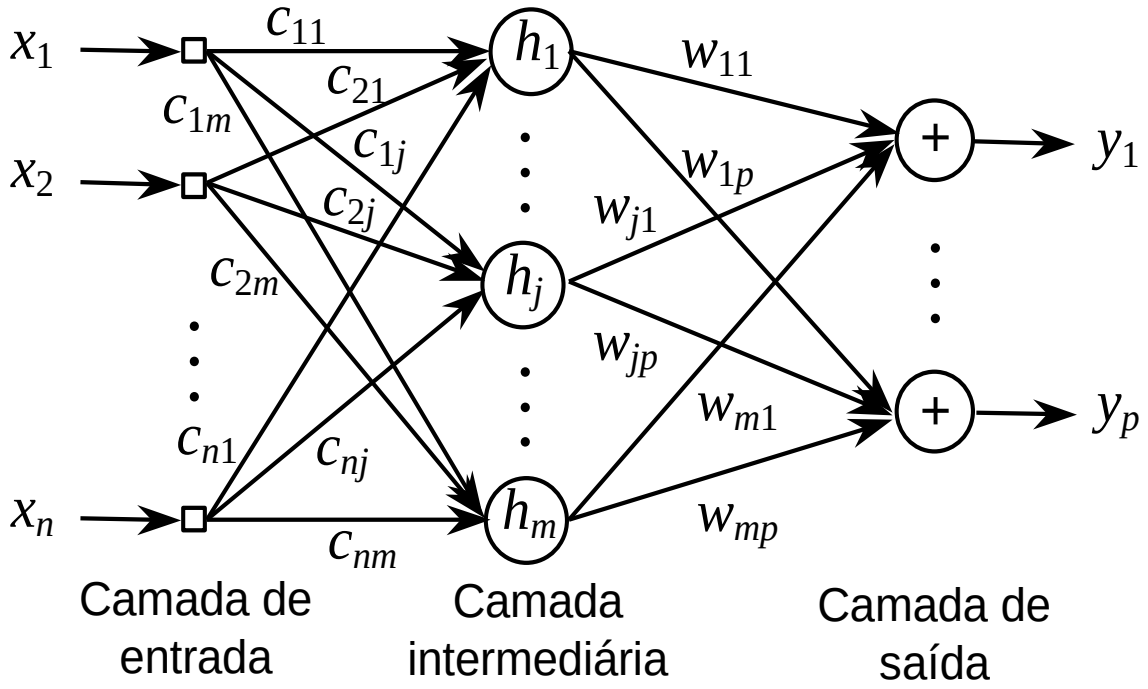


Figura 3 – Rede neural de base radial com múltiplas saídas

- Ao invés da ativação interna de cada neurônio da camada intermediária se dar pelo emprego do produto escalar (produto interno) entre o vetor de entradas e o vetor de pesos, como no caso do perceptron, ela é obtida a partir de uma norma ponderada da diferença entre ambos os vetores.

7 Métodos de Treinamento já Propostos na Literatura

- Várias abordagens para o treinamento de redes neurais com funções de base radial já foram propostas. Geralmente, elas podem ser divididas em duas partes:
  - Definição dos centros, forma e dispersão das funções de base radial, normalmente baseada em treinamento não-supervisionado (quantização vetorial ou algoritmo de treinamento competitivo) ou computação evolutiva;
  - Aprendizado dos pesos da camada de saída, responsáveis pela combinação linear das ativações da camada intermediária, empregando técnicas como pseudo-inversão e OLS (*Orthogonal Least Squares*) (CHEN *et al.*, 1991).
- Em BILLINGS & ZHENG (1995), é proposto um algoritmo genético para evoluir somente os centros, sendo que estes assumem valores inteiros distintos. Após determinados os centros das funções, os pesos são encontrados utilizando o algoritmo OLS.

- Em MOCLOONE *et al*. (1998) é apresentada uma estratégia híbrida, a qual divide o treinamento da rede em duas etapas: treinamento da parte não-linear (centros e dispersões), utilizando o método do gradiente, e treinamento da parte linear (pesos de saída), utilizando OLS. Como já mencionado, o método do gradiente não é indicado para a parte não-linear, a menos que seja aplicado apenas para refinar centros e dispersões já adequadamente (e não aleatoriamente) definidos.
- O algoritmo de aprendizado proposto por HOLCOMB & MORARI (1992) introduz incrementalmente funções de base radial para a camada intermediária, de modo que cada função seja aproximadamente ortogonal às demais. O algoritmo inicia com um único neurônio e outros neurônios são adicionados à rede quando necessário. A localização dos centros da função de base radial associada a cada neurônio é otimizada usando técnicas convencionais de otimização. Entretanto, o critério de término deste algoritmo construtivo não é robusto e pode ser dependente do problema em questão (BILLINGS & ZHENG, 1995).

- Já em relação aos algoritmos propostos por LEE & RHEE (1991) e MUSAVI *et al.* (1992), são utilizados métodos de "clusterização" hierárquica supervisionada. No trabalho de LEE & RHEE (1991), o aprendizado começa com um único neurônio com uma base radial ampla, e introduz-se neurônios adicionais quando necessário. A extensão das bases radiais pode ser alterada ao longo do treinamento. No trabalho de MUSAVI *et al.* (1992), o aprendizado começa com um grande número de neurônios, sendo que alguns são combinados (resultando em um único) quando possível, atualizando os centros e bases a partir daqueles previamente existentes antes da combinação. Estes dois algoritmos foram primeiramente desenvolvidos para reconhecimento de padrões e podem não fornecer soluções adequadas para problemas de identificação de sistemas (BILLINGS & ZHENG, 1995).
- Revisões amplas de métodos de treinamento para redes neurais RBF podem ser encontradas em BUHMANN (2003), em SCHWENKER *et al*. (2001) e em YEE & HAYKIN (2001).

8 Capacidade de Aproximação Universal

- Dado um número suficiente de neurônios com função de base radial, qualquer função contínua definida numa região compacta pode ser devidamente aproximada usando uma rede *RBF* (PARK & SANDBERG, 1991).
- As redes *RBF* são redes de aprendizado local, de modo que é possível chegar a uma boa aproximação desde que um número suficiente de dados para treinamento seja fornecido na região de interesse.
- Em contraste, *perceptrons* multicamadas são redes de aprendizado "global" (em virtude da natureza das funções de ativação: são *ridge functions*) que fazem aproximações de efeito global, em regiões compactas do espaço de aproximação.
- Redes neurais com capacidade de aproximação local são muito eficientes quando a dimensão do vetor de entradas é reduzida. Entretanto, quando o número de entradas não é pequeno, as redes *MLP* apresentam uma maior capacidade de generalização (HAYKIN, 2008).

- Isto ocorre porque o número de funções de base radial deve aumentar exponencialmente com o aumento da dimensão da entrada.

9 O Método dos Quadrados Mínimos para Modelos Lineares nos Parâmetros

- Quando o treinamento supervisionado é aplicado a modelos lineares nos parâmetros, o método dos quadrados mínimos conduz a um problema de otimização que apresenta solução na forma fechada.
- Assim, com um modelo de regressão linear na forma (considerando uma saída):

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j h_j(\mathbf{x})$$

e o conjunto de treinamento dado por $\{(\mathbf{x}_i, s_i)\}_{i=1}^{N}$, o método dos quadrados mínimos se ocupa em minimizar (em relação aos coeficientes da combinação

linear) a soma dos quadrados dos erros produzidos a partir de cada um dos N padrões de entrada-saída.

$$\min_{\mathbf{w}} J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} \left(s_i - f(\mathbf{x}_i)\right)^2 = \min_{\mathbf{w}} \sum_{i=1}^{N} \left(s_i - \sum_{j=1}^{m} w_j h_j(\mathbf{x}_i) \right)^2$$

9.1 Obtenção da Solução Ótima

- Do Cálculo Elementar, sabe-se que a aplicação da condição de otimalidade (restrições atendidas pelos pontos de máximo e mínimo de uma função diferenciável) permite obter a solução ótima do problema de otimização $\min_{\mathbf{w}} J(\mathbf{w})$, na forma:

  1. Diferencie a função em relação aos parâmetros ajustáveis;
  2. Iguale estas derivadas parciais a zero;
  3. Resolva o sistema linear de equações resultante.

- No caso em questão, os parâmetros livres são os coeficientes da combinação linear, dados na forma do vetor de pesos $\mathbf{w} = \begin{bmatrix} w_1 & \cdots & w_j & \cdots & w_m \end{bmatrix}^T$.

- O sistema de equações resultante é dado na forma:

$$\frac{\partial J}{\partial w_j} = -2\sum_{i=1}^{N}\left(s_i - f(\mathbf{x}_i)\right)\frac{\partial f}{\partial w_j} = -2\sum_{i=1}^{N}\left(s_i - f(\mathbf{x}_i)\right)h_j(\mathbf{x}_i) = 0, \quad j=1,...,m.$$

- Separando-se os termos que envolvem a incógnita $f(\cdot)$, resulta:

$$\sum_{i=1}^{N} f(\mathbf{x}_i)h_j(\mathbf{x}_i) = \sum_{i=1}^{N}\left[\sum_{r=1}^{m} w_r h_r(\mathbf{x}_i)\right]h_j(\mathbf{x}_i) = \sum_{i=1}^{N} s_i h_j(\mathbf{x}_i), \quad j=1,...,m.$$

- Portanto, existem m equações para obter as m incógnitas $\{w_r, r=1,...,m\}$. Exceto sob condições "patológicas", este sistema de equações vai apresentar uma solução única. Por exemplo, a solução única é garantida caso os centros das funções de base radial não sejam coincidentes.

- Para encontrar esta solução única do sistema de equações lineares, é interessante recorrer à notação vetorial, fornecida pela álgebra linear, para obter:

$$\mathbf{h}_j^T\mathbf{f} = \mathbf{h}_j^T\mathbf{s}, \quad j=1,...,m,$$

onde

$$\mathbf{h}_j = \begin{bmatrix} h_j(\mathbf{x}_1) \\ \vdots \\ h_j(\mathbf{x}_N) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \sum_{r=1}^{m} w_r h_r(\mathbf{x}_1) \\ \vdots \\ \sum_{r=1}^{m} w_r h_r(\mathbf{x}_N) \end{bmatrix} \quad \text{e} \quad \mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}.$$

- Como existem *m* equações, resulta:

$$\begin{bmatrix} \mathbf{h}_1^T\mathbf{f} \\ \vdots \\ \mathbf{h}_m^T\mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^T\mathbf{s} \\ \vdots \\ \mathbf{h}_m^T\mathbf{s} \end{bmatrix}$$

- Definindo a matriz **H**, com sua *j*-ésima coluna dada por $\mathbf{h}_j$, temos:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_m \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & h_2(\mathbf{x}_N) & \cdots & h_m(\mathbf{x}_N) \end{bmatrix}$$

sendo possível reescrever o sistema de equações lineares como segue:

$$\mathbf{H}^T\mathbf{f} = \mathbf{H}^T\mathbf{s}$$

- O *i*-ésimo componente do vetor **f** pode ser apresentado na forma:

$$f_i = f(\mathbf{x}_i) = \sum_{r=1}^{m} w_r h_r(\mathbf{x}_i) = \begin{bmatrix} h_1(\mathbf{x}_i) & h_2(\mathbf{x}_i) & \cdots & h_m(\mathbf{x}_i) \end{bmatrix}\mathbf{w}$$

permitindo expressar **f** em função da matriz **H**, de modo que:

$$\mathbf{f} = \mathbf{H}\mathbf{w}$$

- Substituindo no sistema de equações lineares, resulta a solução ótima para o vetor de coeficientes da combinação linear (que correspondem aos pesos da camada de saída da rede neural de base radial):

$$\mathbf{H}^T\mathbf{H}\mathbf{w} = \mathbf{H}^T\mathbf{s} \Rightarrow \mathbf{w} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{s}$$

- Esta equação de solução do problema dos quadrados mínimos é conhecida como equação normal. Para que exista a inversa de $\mathbf{H}^T\mathbf{H}$, basta que a matriz $\mathbf{H}$ tenha posto completo, já que geralmente vale $m \leq N$.

9.2 Exemplo

- O modelo linear de regressão mais simples é a reta, aplicada nos casos em que a entrada é escalar: $f(x) = w_1 h_1(x) + w_2 h_2(x)$, onde $h_1(x) = 1$ e $h_2(x) = x$.
- Considere que foram amostrados, na presença de ruído, três pontos da curva $y = x$, gerando o conjunto de treinamento: $\{(x_i, s_i)\}_{i=1}^{3} = \{(1,1.1),(2,1.8),(3,3.1)\}$.
- Obviamente, não se conhece a equação da curva, mas apenas estes três pontos amostrados.
- Para estimar w_1 e w_2, vamos proceder de acordo com os passos do método dos quadrados mínimos.

$$H = \begin{bmatrix} h_1(x_1) & h_2(x_1) \\ h_1(x_2) & h_2(x_2) \\ h_1(x_3) & h_2(x_3) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \qquad \mathbf{s} = \begin{bmatrix} 1.1 \\ 1.8 \\ 3.1 \end{bmatrix} \qquad \mathbf{w} = \left(H^T H\right)^{-1} H^T \mathbf{s} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Para o mesmo conjunto de pontos, considere agora que

$$f(x) = w_1 h_1(x) + w_2 h_2(x) + w_3 h_3(x),$$

onde $h_1(x) = 1$, $h_2(x) = x$ e $h_3(x) = x^2$. Enquanto no caso anterior tínhamos $m < N$, agora temos $m = N$.

- O efeito da adição da função-base extra $h_3(x)$ representa a adição de uma coluna $\mathbf{h}_3 = \begin{bmatrix} h_3(x_1) \\ h_3(x_2) \\ h_3(x_3) \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 9 \end{bmatrix}$ junto à matriz H, e a solução assume a forma $\mathbf{w} = \begin{bmatrix} 1 \\ -0.2 \\ 0.3 \end{bmatrix}$.

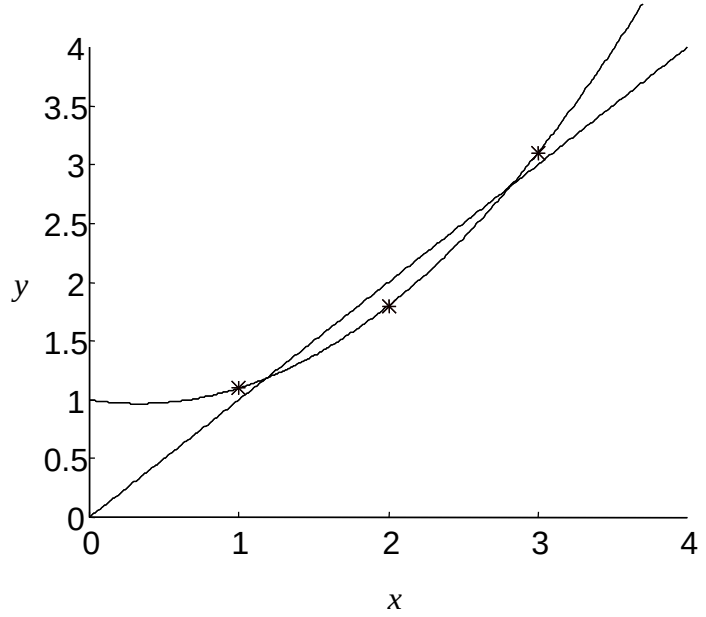
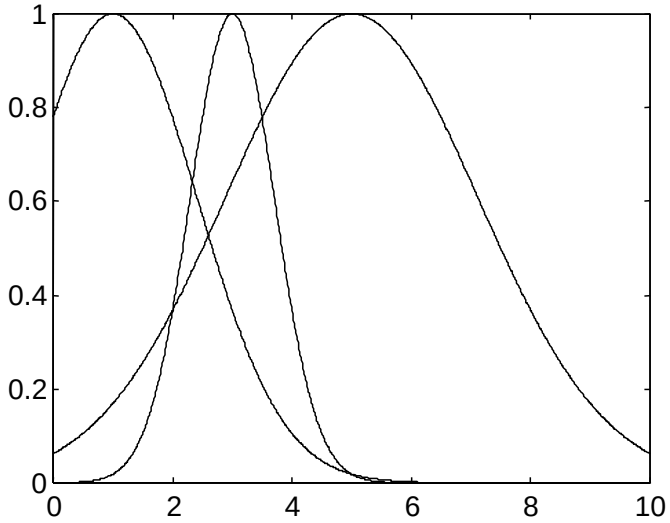
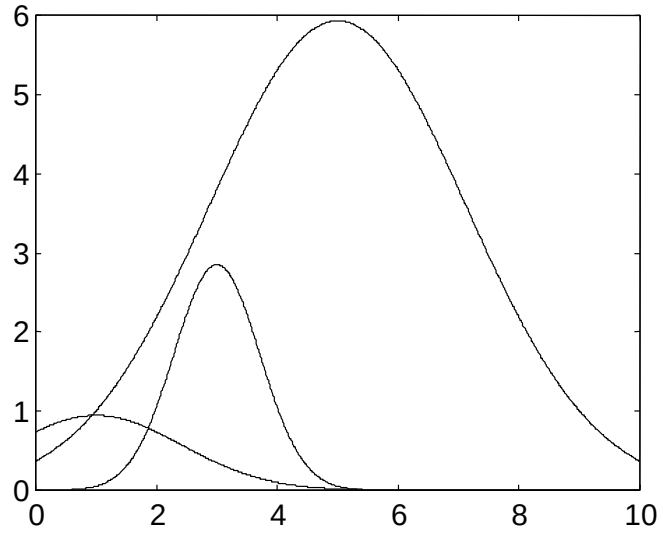


Figura 4 – Modelos de regressão linear (ordem 1 e ordem 2).

- Observe que ambos os modelos são lineares nos parâmetros (daí a denominação de regressão linear), embora para $m = 3$ tenhamos um modelo não-linear.

9.3 Aproximação usando rede neural RBF



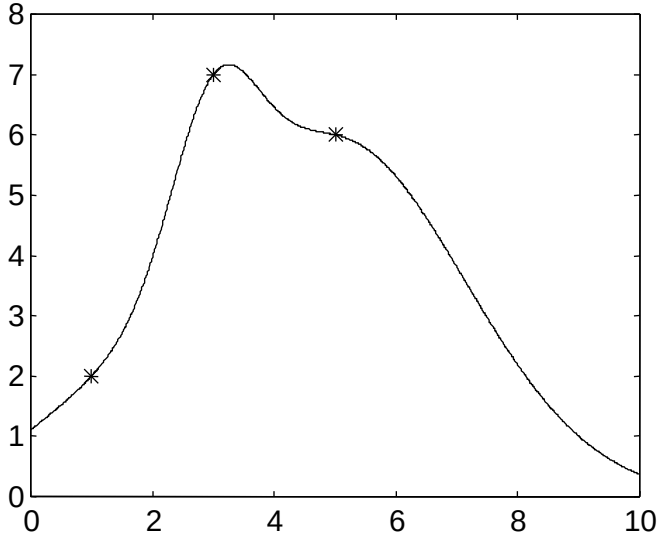


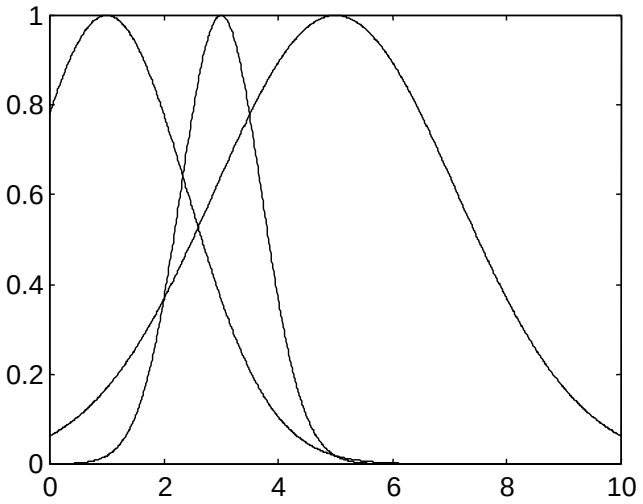
Caso 1: $m = N$

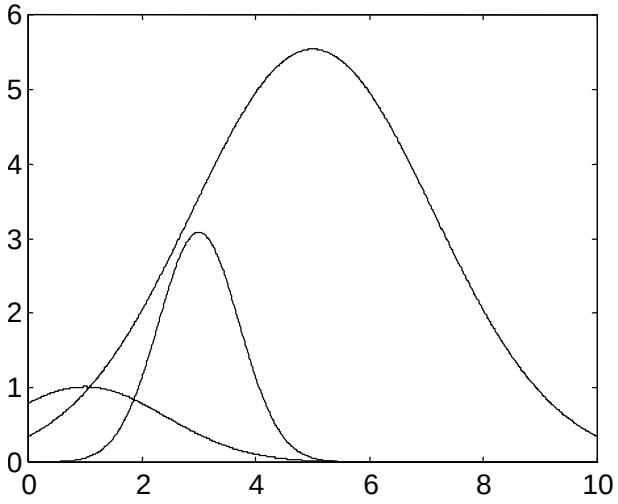
Pontos amostrados: (1,2); (3,7); (5,6)

$$\mathbf{c} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}; \quad \mathbf{r} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} 0.945 \\ 2.850 \\ 5.930 \end{bmatrix}$$

Obs: As funções de base radial têm centros nos valores de x e dispersões arbitrárias.





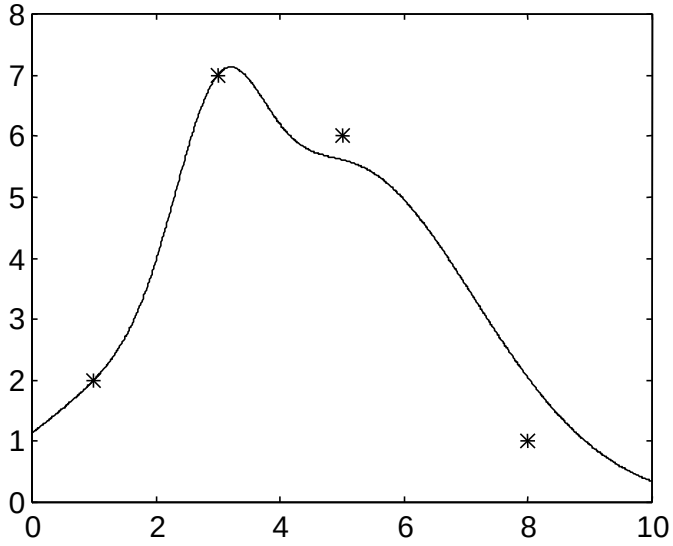


Caso 2: $m < N$

Pontos amostrados: (1,2); (3,7); (5,6); (8,1)

$$\mathbf{c} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}; \quad \mathbf{r} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} 1.012 \\ 3.084 \\ 5.538 \end{bmatrix}$$

Obs: As funções de base radial são as mesmas do Caso 1.



9.4 Técnicas para Determinação dos Centros e Dispersões

- No caso de algoritmos que se ocupam apenas com o ajuste dos pesos da camada de saída de uma rede RBF (modelos lineares nos parâmetros), é necessário estabelecer algum critério para fixação dos centros.
- Existem critérios para o caso de número variável de centros (redes construtivas, por exemplo), mas serão mencionados aqui apenas aqueles geralmente empregados para o caso de um número fixo e previamente especificado de centros.
- Existem basicamente 3 alternativas:
    1. Espalhar os centros uniformemente ao longo da região em que se encontram os dados;
    2. Escolher aleatoriamente, ou segundo algum critério específico, um subconjunto de padrões de entrada como centros;
    3. Auto-organizar os centros, de acordo com a distribuição dos dados de entrada.

- Quanto às dispersões das funções de base radial, embora elas possam ser distintas (e até ajustáveis) para cada centro, usualmente se adota uma única dispersão para todos os centros, na forma (HAYKIN, 2008):

$$\sigma = \frac{d_{\max}}{\sqrt{2m}}$$

onde m é o número de centros, e $d_{\max}$ é a distância máxima entre os centros.

- Com este critério de dispersão, evita-se que as funções de base radial sejam excessivamente pontiagudas, ou então com uma base demasiadamente extensa.

9.4.1 Seleção de Centros por Auto-Organização

- Para auto-organizar os centros, é suficiente aplicar algum algoritmo capaz de refletir a distribuição dos dados de entrada.
- O algoritmo a ser apresentado a seguir, é um algoritmo de clusterização denominado *k*-means (COVER & HART, 1967; MACQUEEN, 1967). Este algoritmo

se assemelha ao adotado para as redes de Kohonen, mas não leva em conta noções de vizinhança entre os centros. **Ele também está sujeito a mínimos locais**.

- Sejam $\{\mathbf{c}_j(t)\}_{j=1}^{m}$ os centros das funções de base radial na iteração t. O algoritmo k-means padrão-a-padrão pode ser descrito da seguinte forma:

1. *Inicialização*: Escolha valores aleatórios distintos para os centros $\mathbf{c}_j(t)$.
2. *Amostragem*: Tome aleatoriamente um vetor $\mathbf{x}_i$ do conjunto de padrões de entrada;
3. *Matching*: Determine o índice k do centro que mais se aproxima deste padrão, na forma: $k(\mathbf{x}_i) = \arg\min_j \|\mathbf{x}_i(t) - \mathbf{c}_j(t)\|, \quad j = 1,...,m$.
4. *Ajuste*: Ajuste os centros usando a seguinte regra:

$$\mathbf{c}_j(t+1) = \begin{cases} \mathbf{c}_j(t) + \gamma[\mathbf{x}_i(t) - \mathbf{c}_j(t)], & k = k(\mathbf{x}_i) \\ \mathbf{c}_j(t), & \text{alhures} \end{cases}$$

   onde $\gamma \in (0,1)$ é a taxa de ajuste.

5. *Ciclo*: Repita os Passos 2 a 5 para todos os N padrões de entrada e até que os centros não apresentem deslocamento significativo após cada apresentação completa dos N padrões.

- Sejam $\{\mathbf{c}_j(t)\}_{j=1}^{m}$ os centros das funções de base radial na iteração t. O algoritmo k-means em batelada pode ser descrito da seguinte forma:

1. *Inicialização*: Escolha valores aleatórios distintos para os centros $\mathbf{c}_j(t)$.

2. *Matching*: Determine o índice k do centro que mais se aproxima de cada padrão, na forma: $k(\mathbf{x}_i) = \arg\min_j \|\mathbf{x}_i(t) - \mathbf{c}_j(t)\|, \quad j = 1,...,m$.

3. *Ajuste*: Defina a nova posição de cada um dos m centros como a média dos padrões cujo índice corresponde àquele centro.

4. *Ciclo*: Repita os Passos 2 e 3 enquanto houver mudança de índice de centro para pelo menos um dos padrões.

9.5 Aplicação das propostas de determinação de centros e dispersão

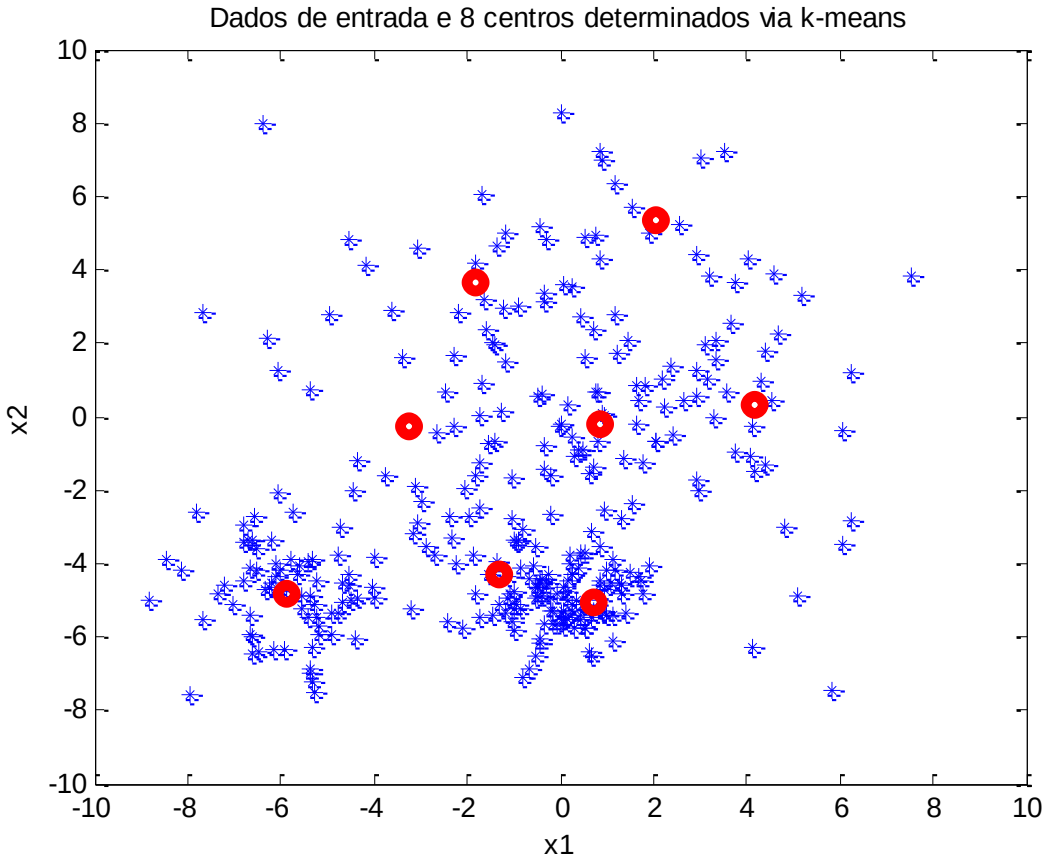


Figura 5 – Proposta de posicionamento dos centros das funções de base radial para uma rede neural RBF com 8 neurônios na camada intermediária

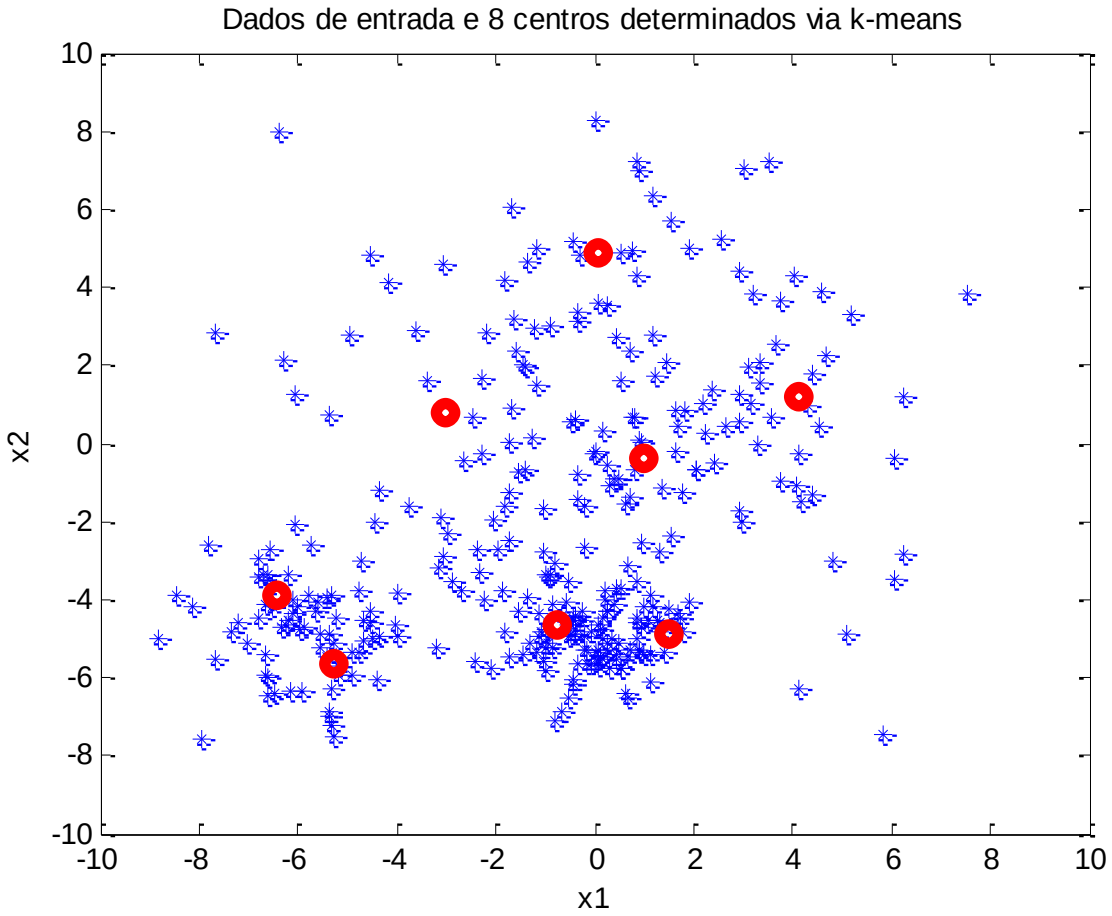


Figura 6 – Outra proposta de posicionamento dos centros para os mesmos dados, produzida por uma segunda execução do algoritmo k-means.

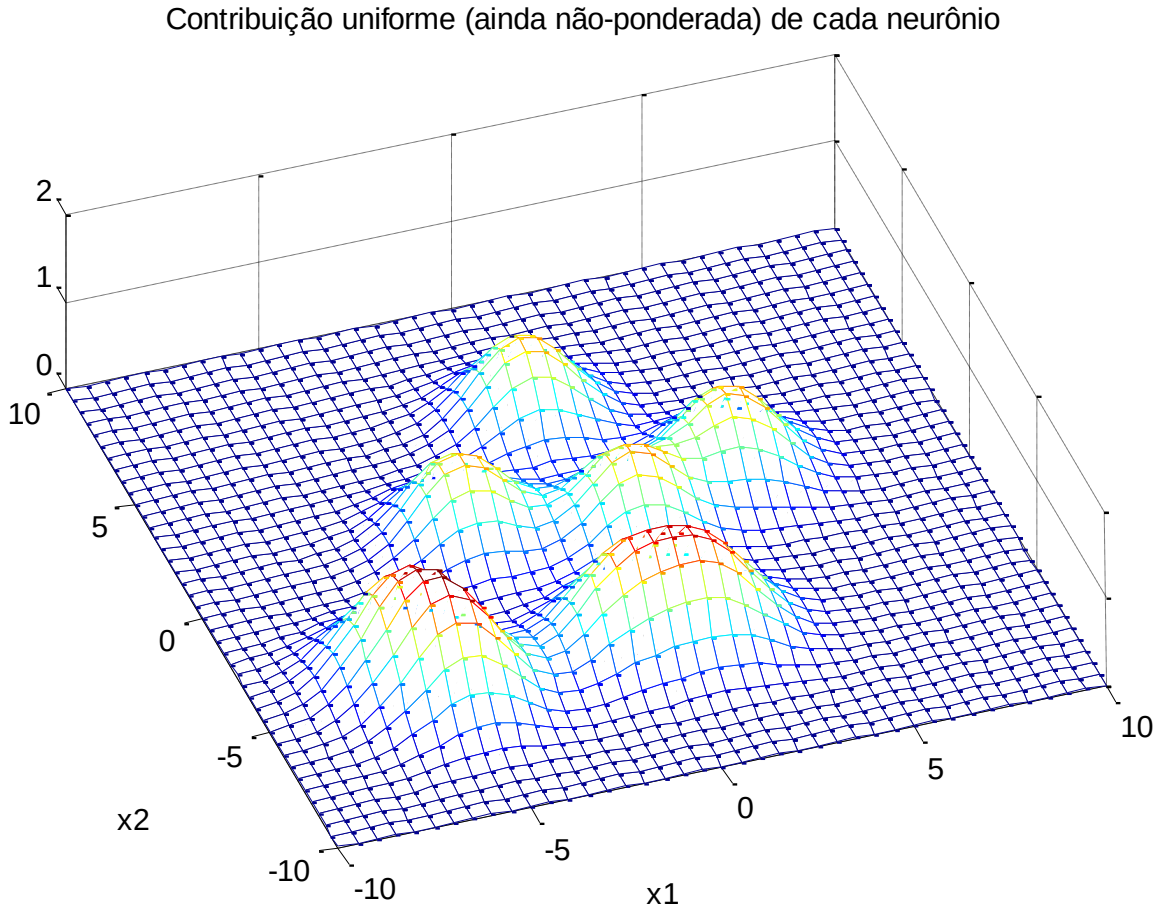


Figura 7 – Ativação dos neurônios da rede neural RBF com os centros da Figura 6, considerando todos os pesos de saída iguais a 1 e ausência de peso de bias. A dispersão é a mesma para todas as funções de ativação, dada pela fórmula da pg. 38.

10 Regularização no ajuste de quadrados mínimos

- Não foi abordado neste tópico do curso o caso regularizado para ajuste de quadrados mínimos (YEE & HAYKIN, 2001), mas esta técnica já foi tratada em profundidade no Tópico 4, Parte 1, no contexto de máquinas de aprendizado extremo.
- Cabe salientar que técnicas de regularização são diretamente aplicáveis a redes neurais com função de ativação de base radial, quando se ocupa do problema de regressão linear associado ao ajuste dos pesos da camada de saída.

11 Outras perspectivas para o problema de quadrados mínimos

- O problema de otimização dos pesos da camada de saída de uma rede neural RBF é linear e a solução de quadrados mínimos foi apresentada como opção, inclusive com a possibilidade de regularização, evidenciada na seção 10.

- Entretanto, o seu emprego supõe algumas hipóteses, como a definição a priori dos centros e dispersões e a ausência de ruído nas variáveis de entrada. Apenas as variáveis de saída admitem ruído.
- Existem formulações alternativas para quadrados mínimos, sendo que serão citadas a seguir três dentre as mais relevantes:
    1. Quadrados mínimos recursivos (*recursive least squares*) (HAYKIN, 2013): é um filtro adaptativo que recursivamente otimiza parâmetros de uma função custo. Não requer a inversão de matrizes.
    2. Quadrados mínimos ortogonais (*orthogonal least squares*) (CHEN et al., 1989; CHEN et al., 1991): As colunas da matriz **H** do slide 30, por serem linearmente independentes, podem ser feitas ortogonais, tornando mais simples o cálculo da contribuição individual de cada coluna de **H** na redução do erro na saída. Tendo acesso a essa contribuição individual, é possível selecionar as colunas (neurônios) mais representativas(os).

3. Quadrados mínimos totais (*total least squares*) (VAN HUFFEL & VANDERWALLE, 1991): em lugar de abordar o problema $A\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}$, considera o problema $(A + \Delta A)\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}$. Nem sempre admite solução e, quando ela existe, pode haver diferenças significativas para a solução de quadrados mínimos.

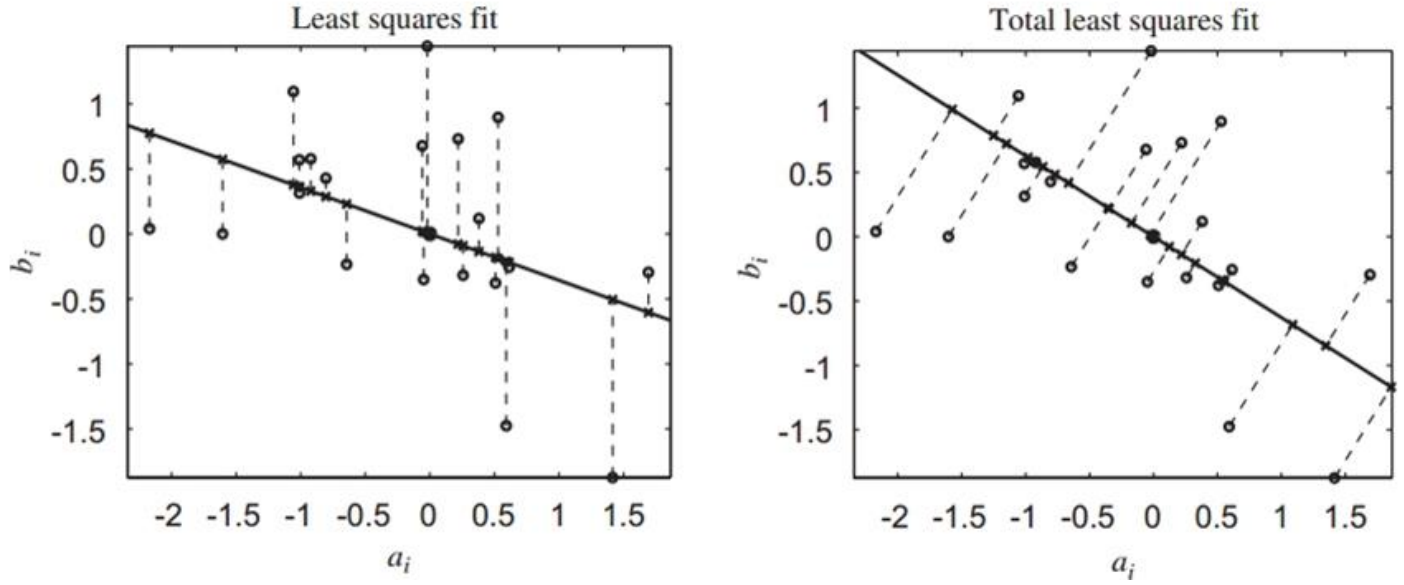


Figura extraída de: http://www.ayushaggarwal.in/sub/experience/imagesn/tls2.png

12 Referências

BILLINGS, S.A. & ZHENG, G.L., "Radial Basis Function Networks Configuration Using Genetic Algorithms", *Neural Networks*, vol. 8, no. 6, pp. 877-890, 1995.

BROOMHEAD, D.S. & LOWE, D. "Multivariate functional interpolation and adaptive networks", *Complex Systems*, 2: 321-355, 1988.

BUHMANN, M.D. "Radial Basis Functions: Theory and Implementations", Cambridge University, 2003.

CHEN, S., BILLINGS, S.A. & LUO, W. "Orthogonal least squares methods and their application to non-linear system identification", *International Journal of Control*, vol. 50, no. 5, pp. 1873-1896, 1989.

CHEN, S., COWAN, C.F.N. & GRANT, P.M. "Orthogonal Least Squares Algorithm for Radial Basis Function Networks", *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.

COVER, T.M. & HART, P.E. "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, 13:21-27, 1967.

HAYKIN, S. "Adaptive Filter Theory", 5th edition, Pearson, 2013.

HAYKIN, S. "Neural Networks and Learning Machines", 3rd edition, Prentice Hall, 2008.

HOLCOMB, T.R. & MORARI, M. "PLS/Neural Networks", *Computers & Chemical Engineerin*, vol. 16, no. 4, pp. 393-411, 1992.

LEE, S. & RHEE, M.K. "A Gaussian potencial function network with hierarchically self-organizing learning", *Neural Networks*, vol. 4, pp. 207-224, 1991.

MACQUEEN, J. "Some methods for classification and analysis of multivariate observation", *in* L.M. LeCun and J Neyman (eds.) *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1:281-297, 1967.

MOCLOONE, S., BROWN, M.D., IRWIN, L. "A hybrid linear/nonlinear training algorithm for feedforward neural networks", *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 669-684, 1998.

MUSAVI, M.T., AHMED, W., CHAN, K.H., FARIS, K.B. & HUMMELS, D.M. "On the training of radial basis function classifiers", *Neural Networks*, vol. 5, pp. 595-603, 1992.

ORR, M.J.L. "Introduction to Radial Basis Function Networks", *Technical Report*, Centre for Cognitive Science, University of Edinburgh, Scotland, 1996. (*http://www.anc.ed.ac.uk/~mjo/papers/intro.ps*)

PARK, J. & SANDBERG, I.W. "Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2): 246-257, 1991.

SCHWENKER, F., KESTLER, H.A. & GÜNTER, P. "Three learning phases for radial-basis-function networks", *Neural Networks*, vol. 14, nos. 4-5, pp. 439-458, 2001.

VAN HUFFEL, S. & VANDERWALLE, J. "The Total Least Squares Problem: Computational Aspects and Analysis", SIAM, 1991.

YEE, P.V. & HAYKIN, S. "Regularized Radial Basis Function Networks: Theory and Applications", John Wiley & Sons, 2001.

13 Bibliografia complementar

BISHOP, C.M. "Improving the generalisation properties of radial basis function neural networks", *Neural Networks*, 3(4): 579-588, 1991.

BISHOP, C.M. "Neural Networks for Pattern Recognition", Clarendon Press, 1995.

CHEN, C.-L., CHEN, W.-C. & CHANG, F.-Y. "Hybrid learning algorithm for Gaussian potential function networks", *IEE Proceedings D*, 140(6): 442-448, 1993.

CHEN, S., CHNG, E.S. & ALKADHIMI, K. "Regularized Orthogonal Least Squares Algorithm for Constructing Radial Basis Function Networks", *International Journal of Control*, 64(5): 829-837, 1996.

DE CASTRO, L.N. & VON ZUBEN, F.J. An Immunological Approach to Initialize Centers of Radial Basis Function Neural Networks. *Anais do V Congresso Brasileiro de Redes Neurais*, pp. 79-84, Rio de Janeiro, 2 a 5 de Abril de 2001.

DE CASTRO, L.N. & VON ZUBEN, F.J. A Pruning Self-Organizing Algorithm to Select Centers of Radial Basis Function Neural Networks. *in* Kurková, V., Steele, N.C., Neruda, R., Kárný, M. (eds.) *Proceedings of the International Conference of Artificial Neural Networks and Genetic Algorithms* (ICANNGA'01), pp. 114-117, Prague, Czech Republic, April 22-25, 2001.

DE CASTRO, L.N. & VON ZUBEN, F.J. Automatic Determination of Radial Basis Functions: An Immunity-Based Approach. *International Journal of Neural Systems*, 2001. (to appear)

FREEMAN, J.A.S. & SAAD, D. "Learning and Generalization in Radial Basis Function Networks", *Neural Computation*, **7**: 1000-1020, 1995.

FRITZKE, B. "Fast learning with incremental RBF Networks", *Neural Processing Letters*, 1(1): 2-5, 1994.

FRITZKE, B. "Supervised Learning with Growing Cell Structures", *in* J. Cowan, G. Tesauro & J. Alspector (eds.) *Advances in Neural Information Processing Systems*, Morgan Kaufmann Publishers, 6: 255-262, 1994.

GIROSI, F. "Some extensions of radial basis functions and their applications in artificial intelligence", Computers & Mathematics with Applications, vol. 24, no. 12, pp. 61-80, 1992.

GOMM, J.B. & YU, D.L. "Selecting Radial Basis Function Network Centers with Recursive Orthogonal Least Squares Training", *IEEE Transactions on Neural Networks*, 11(2):306-314, 2000.

HWANG, Y.-S. & BANG, S.-Y. "An Efficient Method to Construct a Radial Basis Function Neural Network Classifier", *Neural Networks*, 10(8): 1495-1503, 1997.

KARAYIANNIS, N.B. "Gradient Descent Learning of Radial Basis Neural Networks", *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1815-1820, 1997.

KARAYIANNIS, N.B. & MI, G.W. "Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with Network Growth Techniques", *IEEE Transactions on Neural Networks*, 8(6): 1492-1506, 1997.

KUBAT, M. "Decision trees can initialize radial-basis function networks", *IEEE Transactions on Neural Networks*, 9(5): 813-821, 1998.

LIPPMANN, R.P. "Pattern Classification Using Neural Networks", *IEEE Communications Magazine*, November, pp. 47-63, 1989.

MICCHELLI, C.A. "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions", *Constructive Approximation*, 2: 11-22, 1986.

MOODY, J. & DARKEN, C. "Fast Learning in Networks of Locally-Tuned Processing Units", *Neural Computation*, 1: 281-294, 1989.

MULGREW, B. "Applying Radial Basis Functions", *IEEE Signal Proc. Magazine*, pp. 50-66, 1996.

ORR, M.J.L. "Regularisation in the Selection of Radial Basis Function Centres", *Neural Computation*, **7**(3): 606-623, 1995.

ORR, M.J.L. "Optimising the widths of radial basis functions", *Proceedings of the Fifth Brazilian Symposium on Neural Networks*, Belo Horizonte, Brazil, 1998.

ORR, M.J.L. "Recent Advances in Radial Basis Function Networks", *Technical Report*, Institute for Adaptive and Neural Computation, University of Edinburgh, Scotland, 1999. (*http://www.anc.ed.ac.uk/~mjo/papers/recad.ps*)

POGGIO, T. & GIROSI, F. "Networks for Approximation and Learning", *Proceedings of the IEEE*, 78(9): 1481-1497, 1990.

SUTANTO, E.L., MASON, J.D. & WARWICK, K. "Mean-tracking clustering algorithm for radial basis function centre selection. *International Journal of Control*, 67(6): 961-977, 1997.

WANG, Z. & ZHU, T. "An Efficient Learning Algorithm for Improving Generalization Performance of Radial Basis Function Neural Networks", *Neural Networks*, 13(4-5): 545-553, 2000.

WETTSCHERECK, D. & DIETTERICH, T. "Improving the Performance of Radial Basis Function Networks by Learning Center Locations", *Advances in Neural Information Processing Systems*, 4:1133-1140, 1992.

WHITEHEAD, B.A. & CHOATE, T.D. "Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction", *IEEE Transactions on Neural Networks*, **7**(4): 869-880, 1996.

WHITEHEAD, B.A. & CHOATE, T.D. "Evolving Space-Filling Curves to Distribute Radial Basis Functions Over an Input Space", *IEEE Transactions on Neural Networks*, 5(1): 15-23, 1994.

YINGWEI, L., SUNDARARAJAN, N. & SARATCHANDRAN, P. "A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function Neural Networks", *Neural Computation*, pp. 461-478, 1996.