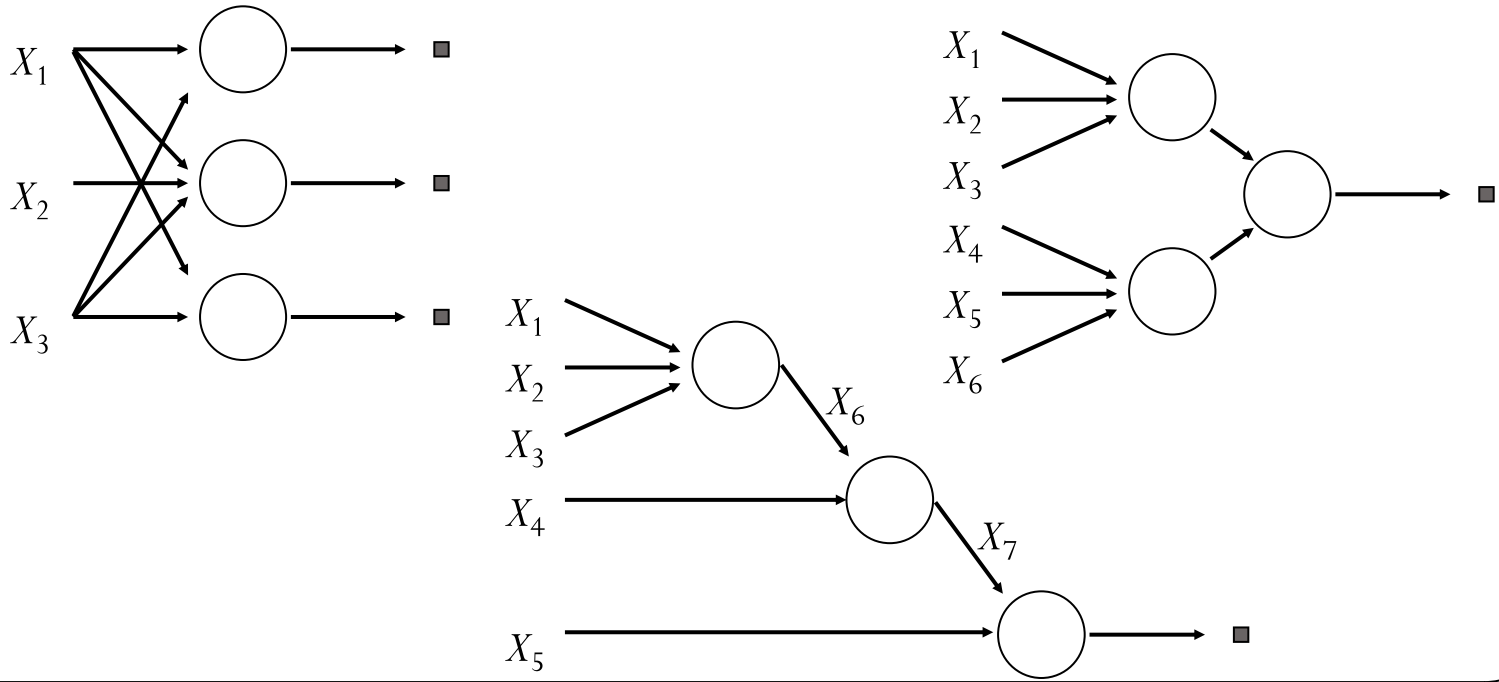
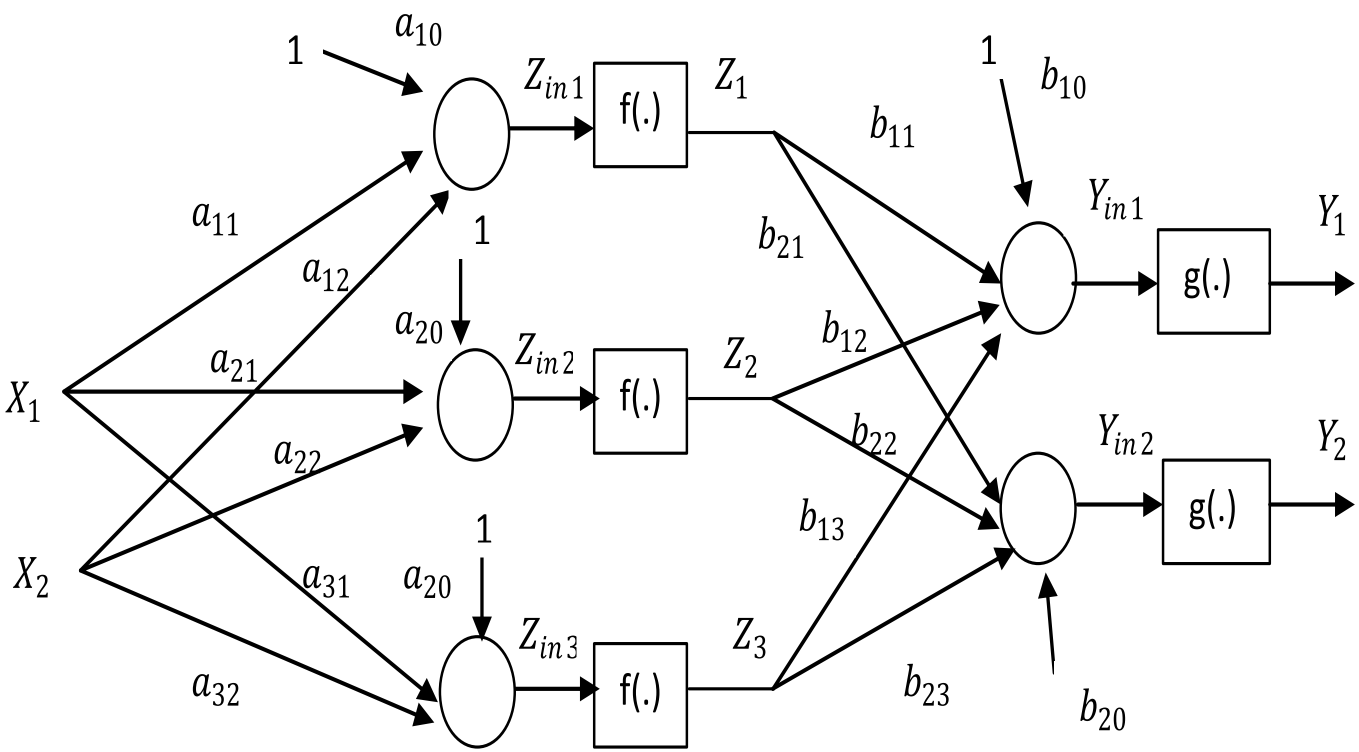
Redes Neurais

Introdução

- **Arquiteturas de RNA**
  - **Tipos de conexões dos nós**
    - ***Feedforward*, ou acíclica**
      - A saída do neurônio na i-ésima camada da rede não pode ser usada como entrada de nós em camadas de índice menor ou igual a i

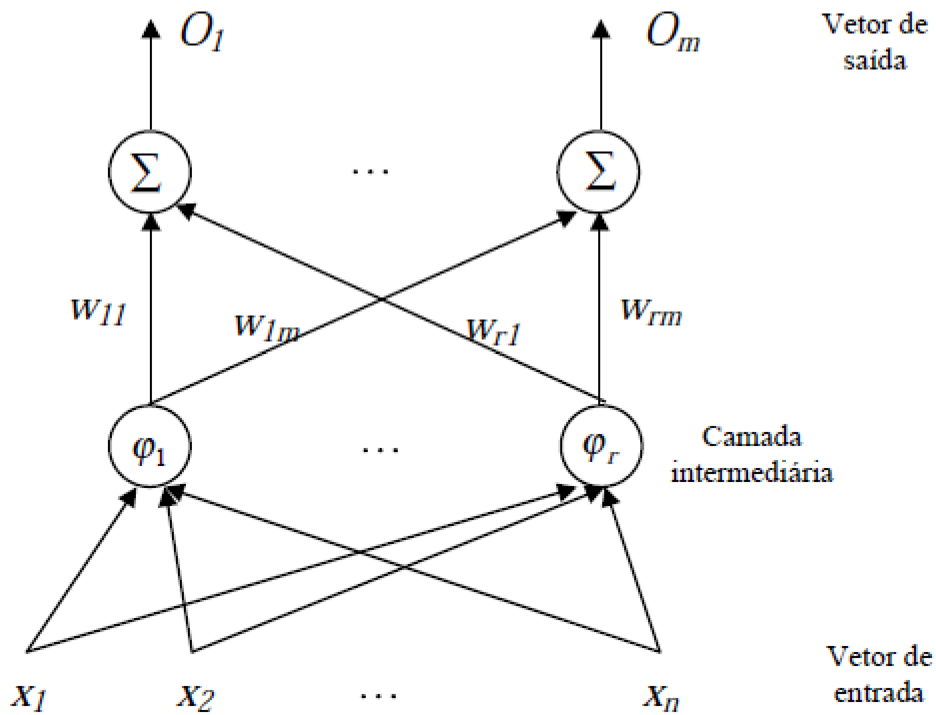


Rede perceptron multicamada (MLP)



- Exemplo de Rede Neural MLP com 2 entradas, 3 neurônios na camada escondida, 2 saídas

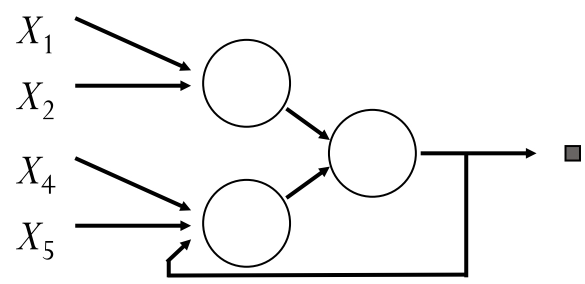
Rede Neural com Função de Base Radial - RBF



Introdução

- **Arquiteturas de RNA**
  - **Tipos de conexões dos nós**
    - ***Feedback*, ou cíclica**
      - A saída do neurônio na i-ésima camada da rede é usada como entrada de nós em camadas de índice menor ou igual a i

o Redes cuja saída final (única) é ligada às entradas comportam-se como autômatos reconhecedores de cadeias, onde a saída que é realimentada fornece o estado do autômato

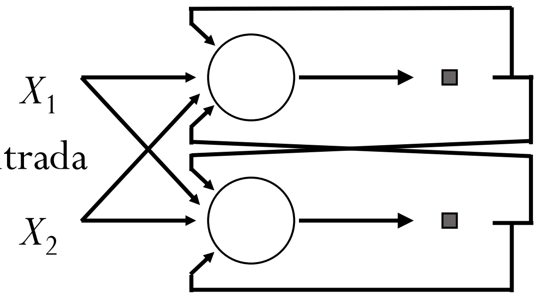


o **Auto-associativa**

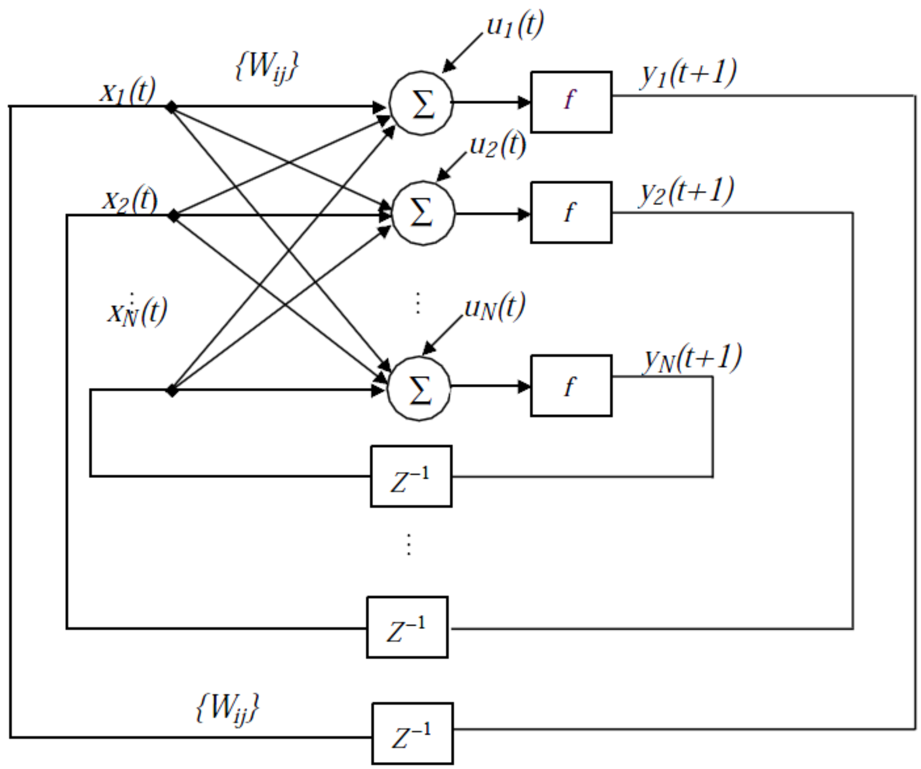
Todas as ligações são cíclicas

Associam um padrão de entrada com ele mesmo

São particularmente úteis para recuperação ou regeneração de um padrão de entrada

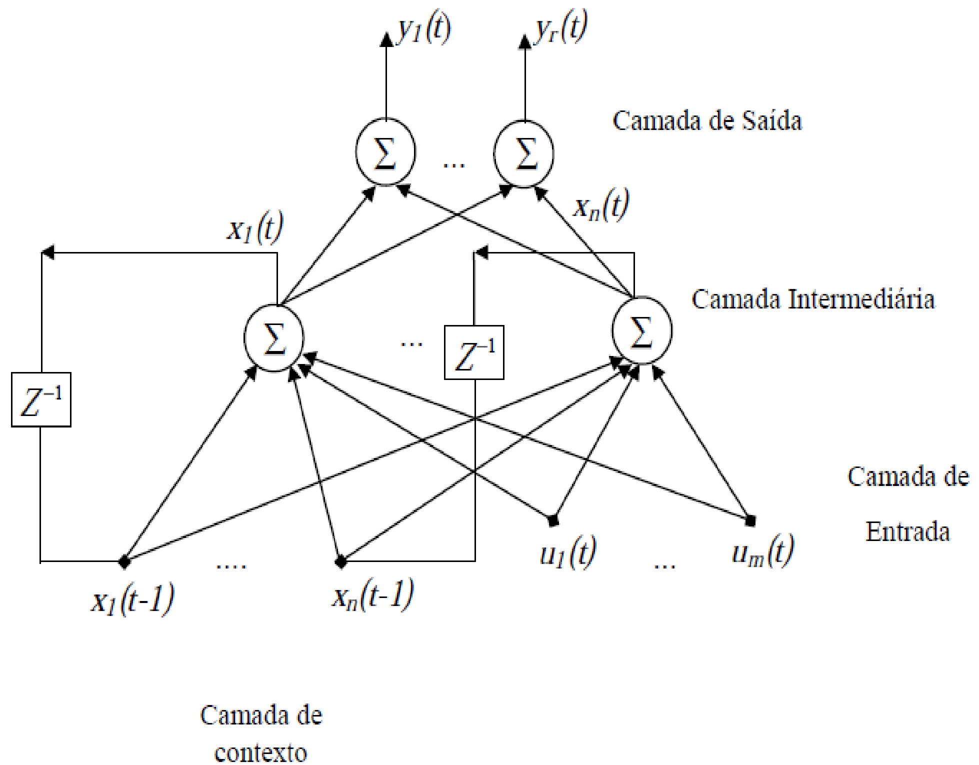


Redes Recorrentes



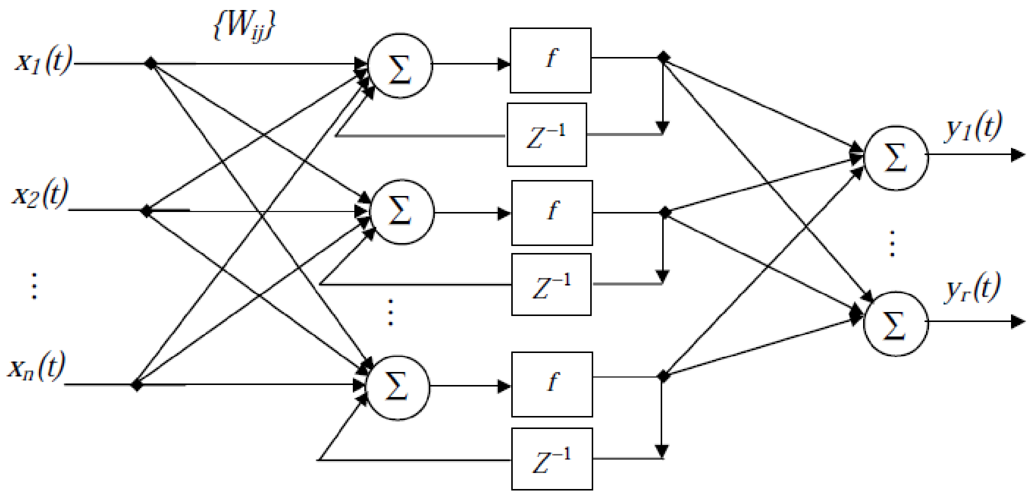
- Estrutura de uma rede neural totalmente recorrente (Rede de Hopfield com entradas externas)

Redes Recorrentes



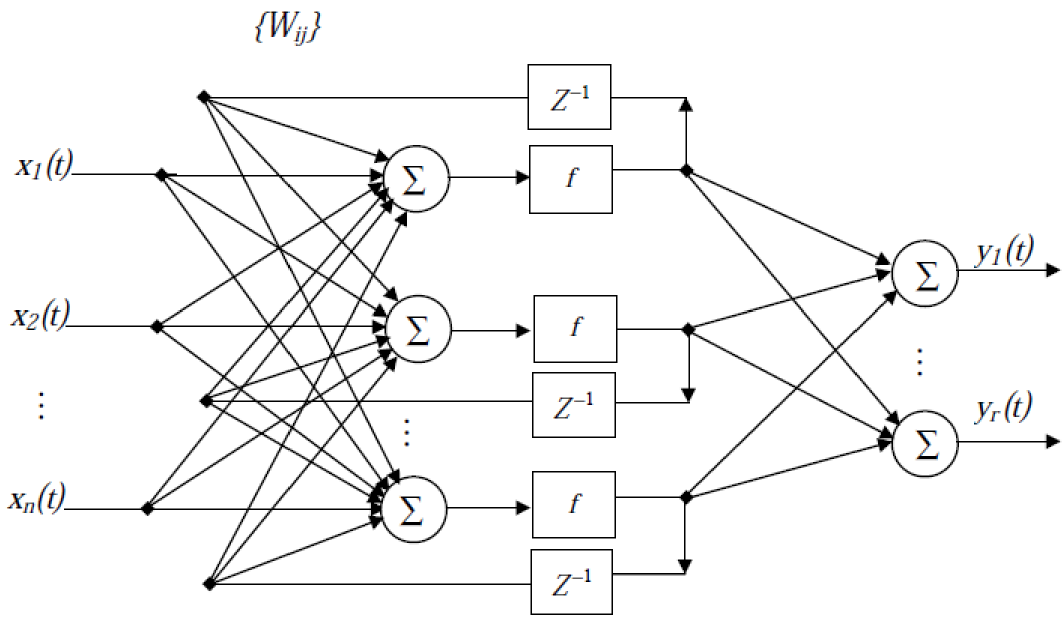
- Estrutura de uma Rede de Elman

Redes Recorrentes



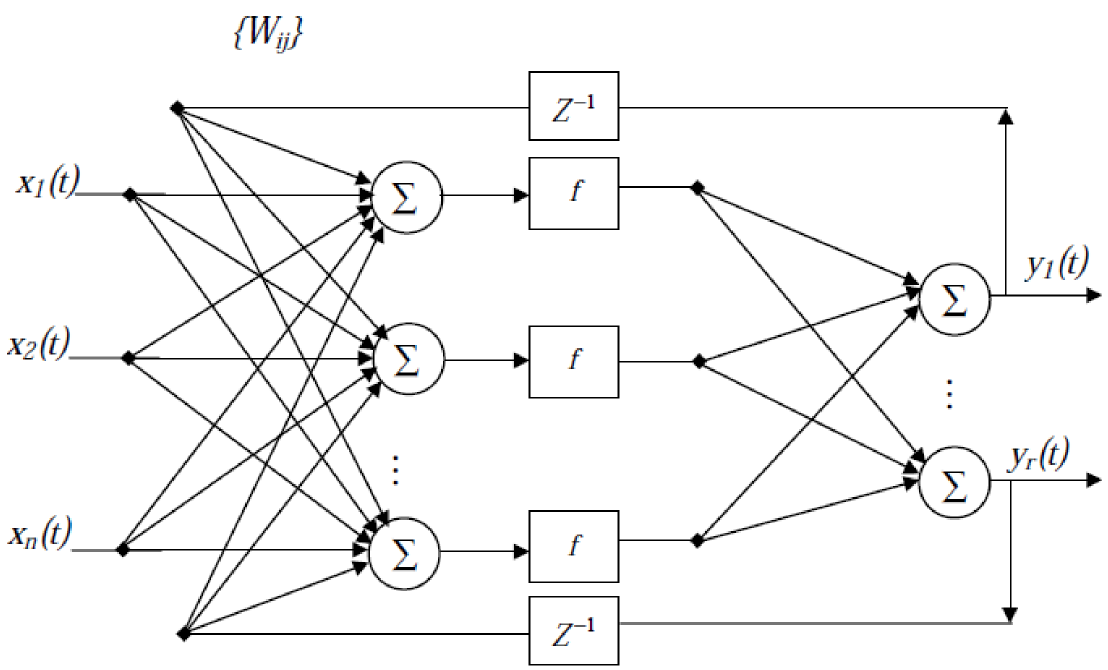
- Estrutura de uma rede neural com recorrência interna local

Redes Recorrentes



- Estrutura de uma rede neural com recorrência interna global

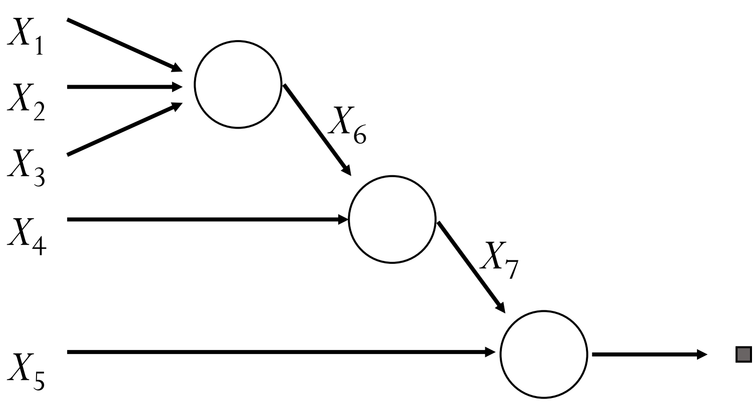
Redes Recorrentes

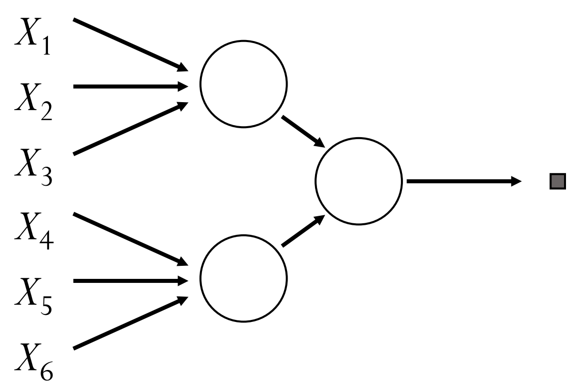


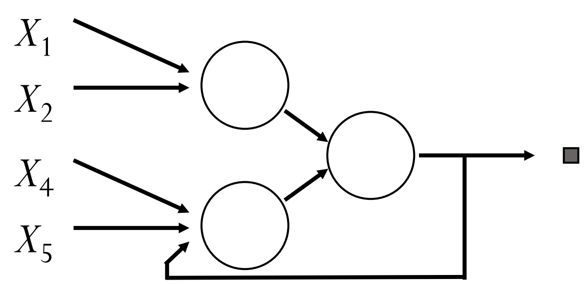
- Estrutura de uma rede neural com recorrência externa

Introdução

- **Arquiteturas de RNA**
  - **Conectividade**
    - **Fracamente (ou parcialmente) conectada**

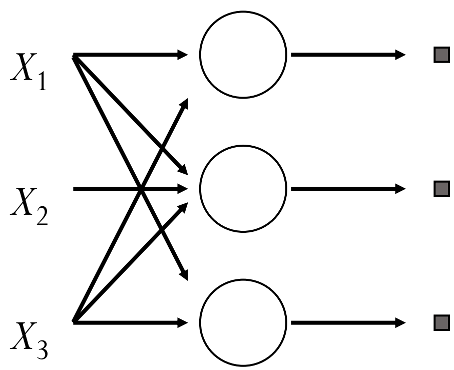


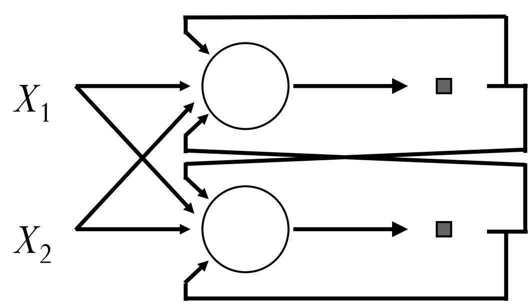




Introdução

- **Arquiteturas de RNA**
  - **Conectividade**
    - **Completamente conectada**

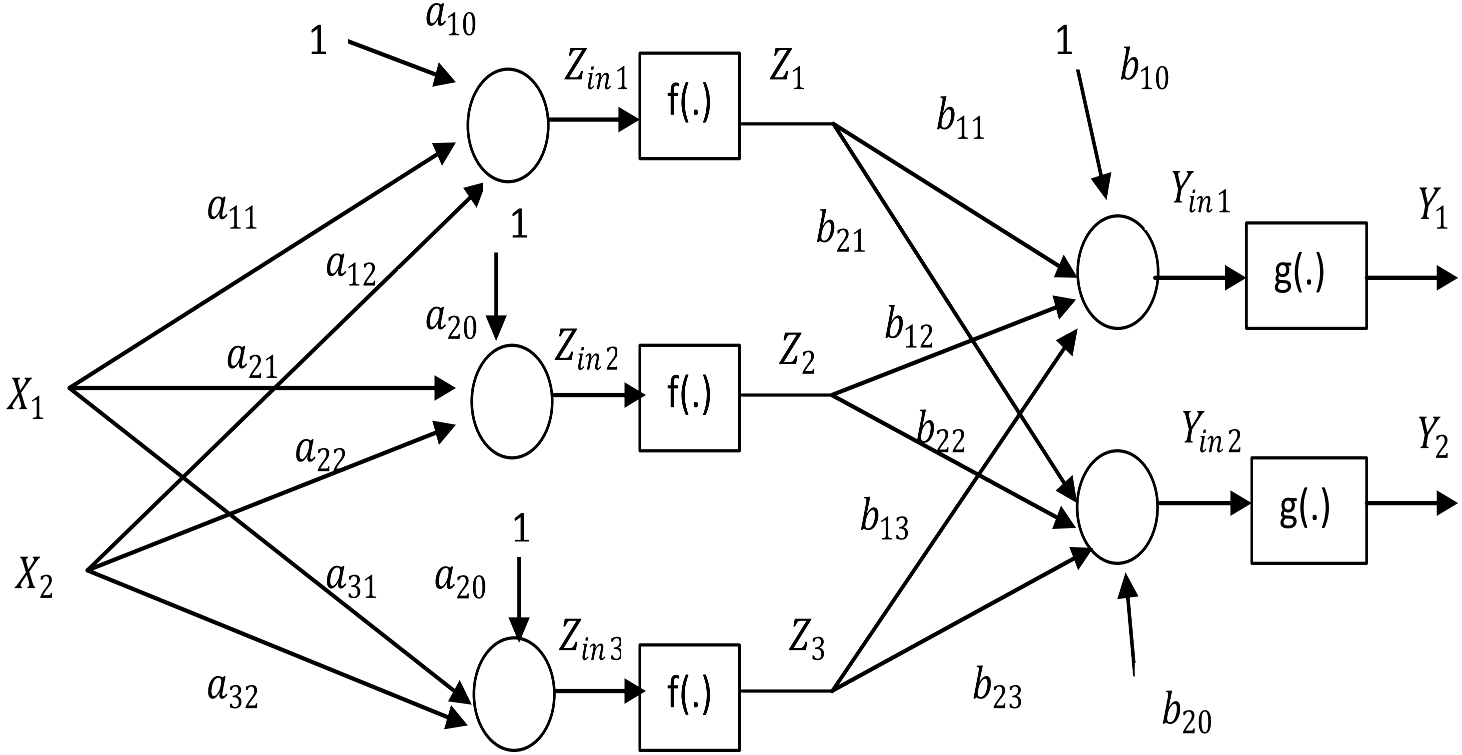




Introdução

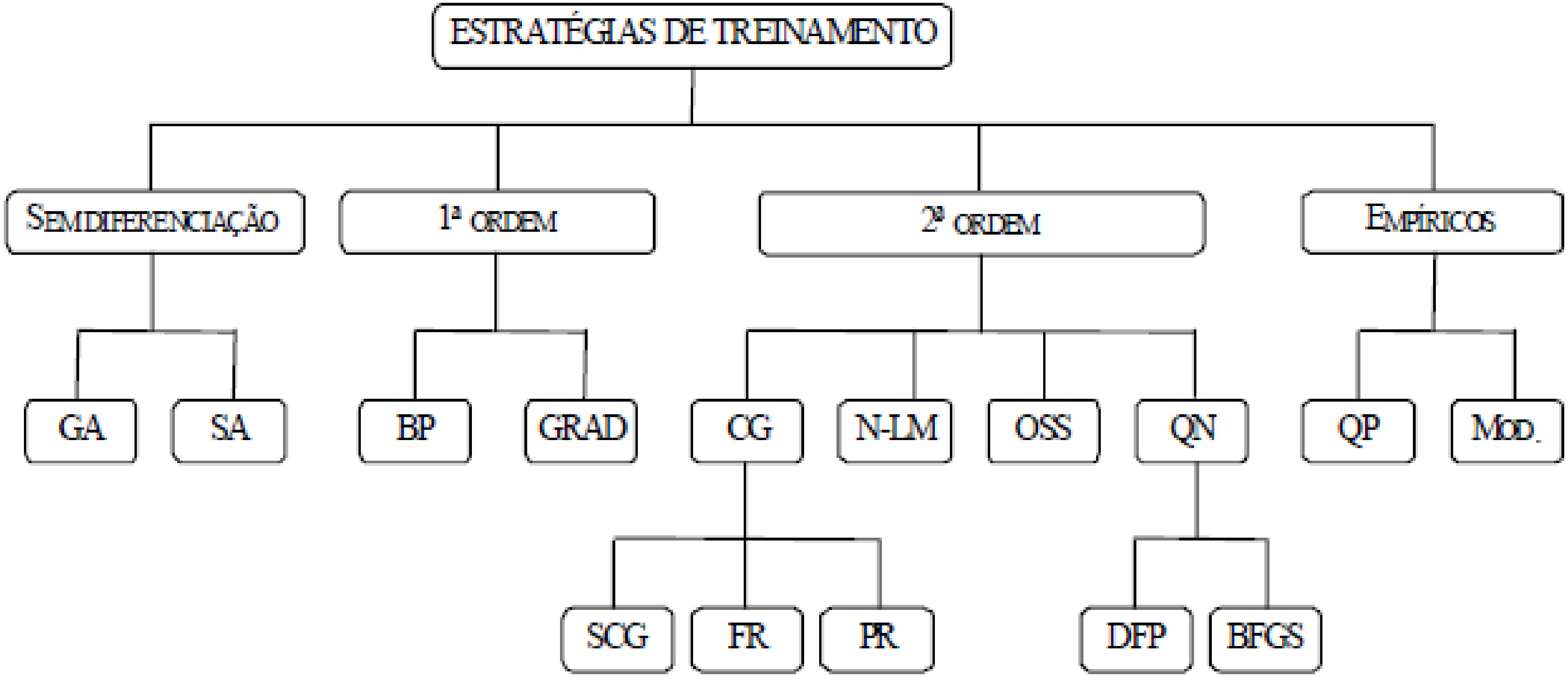
- **Arquiteturas de RNA**
  - Uma rede neural é caracterizada, principalmente …
    - … pela sua ***topologia*** (feedforward, feedback)
    - … pelas ***características dos nós*** (booleano, fuzzy, híbrido)
    - … pelas ***regras de treinamento*** (Hebb, backpropagation, …)

Rede perceptron multicamada (MLP)

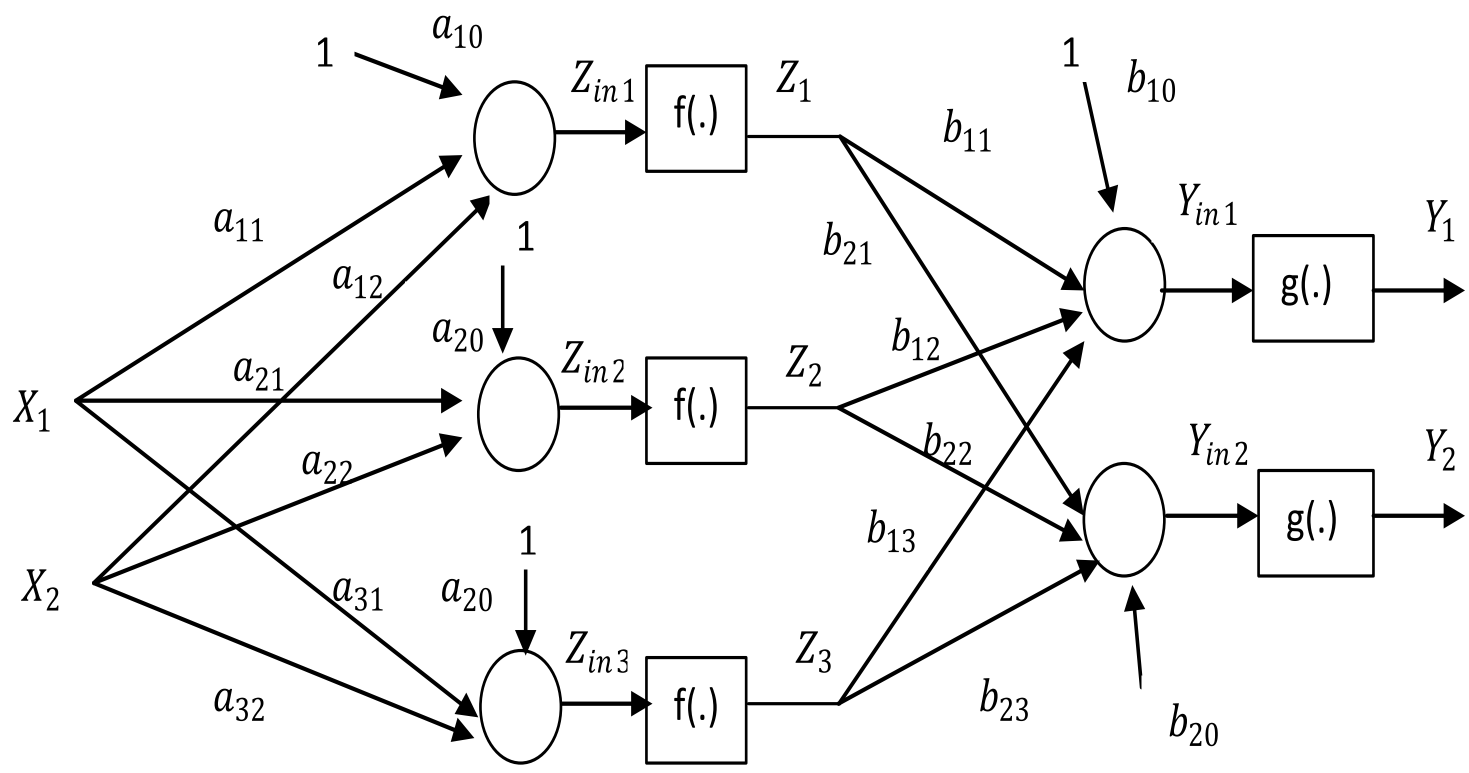


Exemplo de Rede Neural MLP com 2 entradas, 3 neurônios na camada escondida, 2 saídas

Estratégias de treinamento



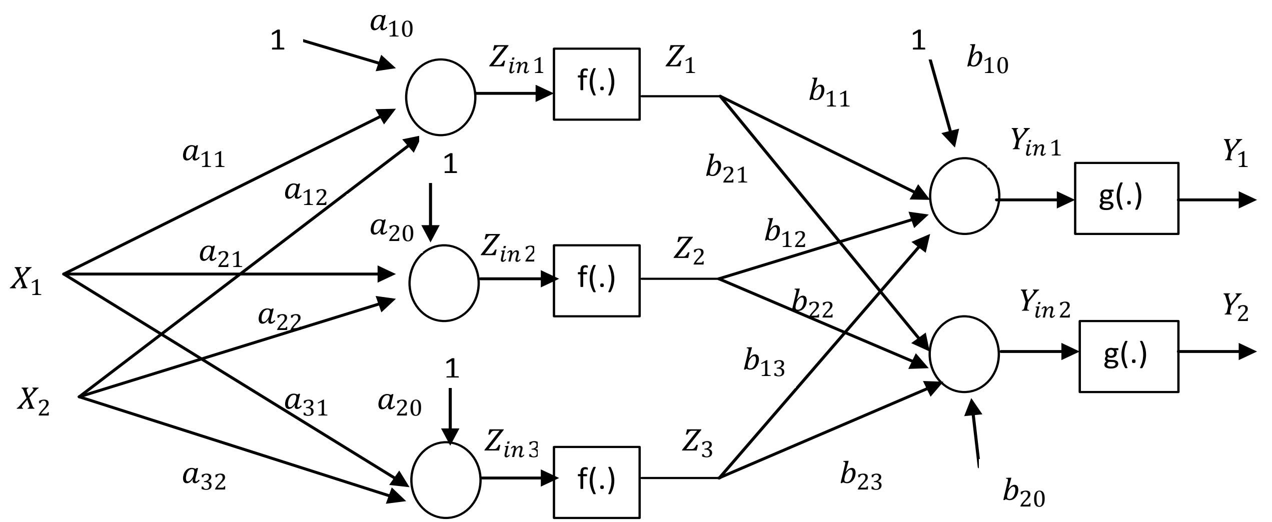
REDE PERCEPTRON MULTICAMADA (MLP)



$$A = \begin{bmatrix} a_{11} & a_{12} & a_{10} \\ a_{21} & a_{22} & a_{20} \\ a_{31} & a_{32} & a_{30} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{10} \\ b_{21} & b_{22} & b_{23} & b_{20} \end{bmatrix}$$

CÁLCULO DA SAÍDA DA REDE



- xj(n)- *j-ésima* entrada para padrão *n*
- *yk(n)- k-ésima saída da rede neural*
- *dk(n)- k-ésima saída desejada para o padrão*
- *h - número de neurônios na camada escondia*
- *ne - número de atributos de entrada*
- *ns - número de saída da rede neural*

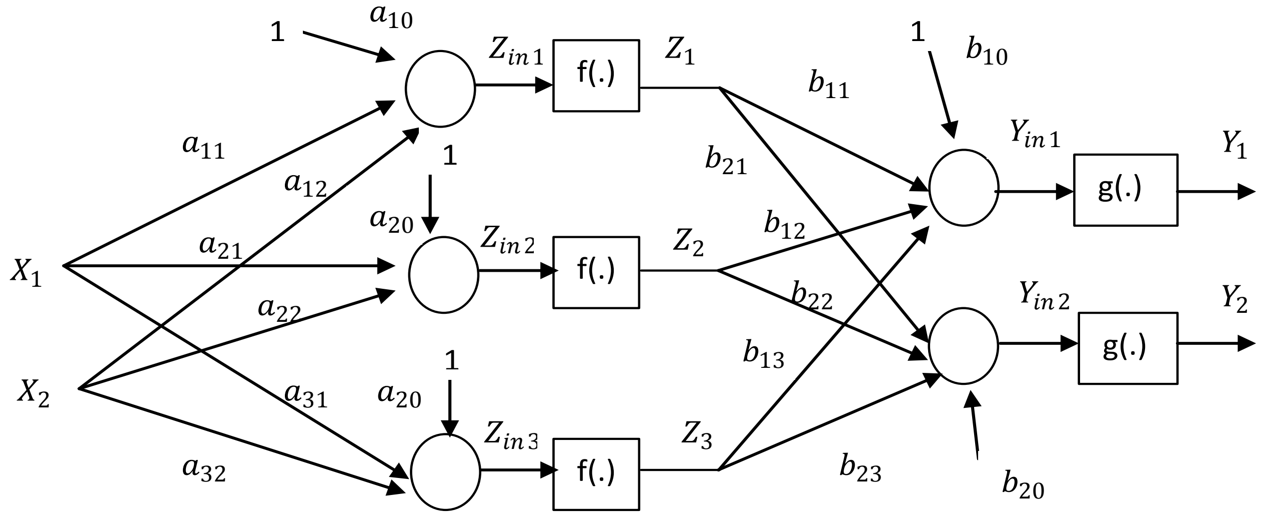
$$Zin_i(n) = \sum_{j=0}^{ne} a_{ij} X_j(n)$$

$$Z_i(n) = f(Zin_i(n))$$

$$Yin_k(n) = \sum_{i=0}^{h} b_{ki} Z_i(n)$$

$$Y_k(n) = g(Yin_k(n))$$

CÁLCULO DA SAÍDA DA REDE



$$Zin_i(n) = \sum_{j=0}^{ns} a_{ij} X_j(n)$$

$$Z_i(n) = f(Zin_i(n))$$

$$Yin_k(n) = \sum_{i=0}^{h} b_{ki} Z_i(n)$$

$$Y_k(n) = g(Yin_k(n))$$

CÁLCULO DO ERRO

$$e_k(n) = Y_k(n) - d_k(n)$$

$$E(n) = \frac{1}{2}\sum_{k=1}^{ns} e_k(n)^2$$

$$E_{T=}\frac{1}{N}\sum_{n=1}^{N} E(n)$$

VETOR GRADIENTE E HESSIANA

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{10} \\ a_{21} & a_{22} & a_{20} \\ a_{31} & a_{32} & a_{30} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{10} \\ b_{21} & b_{22} & b_{23} & b_{20} \end{bmatrix}$$

$$\nabla E_T = \begin{bmatrix} \dfrac{\partial E_T}{\partial a_{11}} \\ \vdots \\ \dfrac{\partial E_T}{\partial a_{30}} \\ \dfrac{\partial E_T}{\partial b_{11}} \\ \vdots \\ \dfrac{\partial E_T}{\partial b_{20}} \end{bmatrix} \qquad \nabla^2 E_T = \begin{bmatrix} \dfrac{\partial^2 E_T}{\partial {a_{11}}^2} & \cdots & \dfrac{\partial^2 E_T}{\partial a_{11} \partial b_{20}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 E_T}{\partial b_{20} \partial a_{11}} & \cdots & \dfrac{\partial^2 E_T}{\partial {b_{20}}^2} \end{bmatrix}$$

Cálculo do gradiente

Cálculo da Saída da Rede

$$Zin_i(n) = \sum_{j=0}^{ne} a_{ij} X_j(n)$$

$$Z_i(n) = f(Zin_i(n))$$

$$Yin_k(n) = \sum_{i=0}^{h} b_{ki} Z_i(n)$$

$$Y_k(n) = g(Yin_k(n))$$

Cálculo do erro

$$e_k(n) = Y_k(n) - d_k(n)$$

$$E(n) = \frac{1}{2} \sum_{k=1}^{ns} e_k(n)^2$$

$$E_{T} = \frac{1}{N} \sum_{n=1}^{N} E(n)$$

Cálculo $\dfrac{\partial E_T}{\partial b_{ki}}$

$$\frac{\partial E_T}{\partial b_{ki}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial E(n)}{\partial b_{ki}}$$

Sabemos que

$$\frac{\partial E(n)}{\partial b_{ki}} = \frac{\partial E(n)}{\partial e_k(n)} \cdot \frac{\partial e_k(n)}{\partial Y_k(n)} \cdot \frac{\partial Y_k(n)}{\partial Yin_k(n)} \cdot \frac{\partial Yin_k(n)}{\partial b_{ki}}$$

CÁLCULO DO GRADIENTE

$$E_T = \frac{1}{N}\sum_{n=1}^{N} E(n) \quad \Longrightarrow \quad \frac{\partial E_T}{\partial b_{ki}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial E(n)}{\partial b_{ki}}$$

$$E(n) = \frac{1}{2}\sum_{k=1}^{ns} e_k(n)^2 \quad \Longrightarrow \quad \frac{\partial E(n)}{\partial e_k(n)} = e_k(n)$$

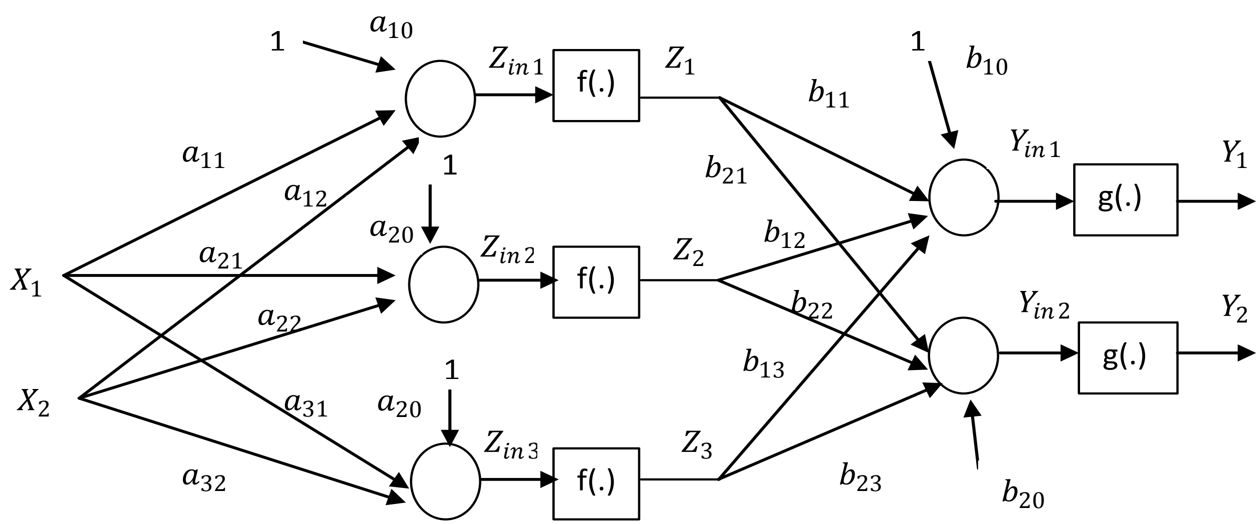
$$e_k(n) = Y_k(n) - d_k(n) \quad \Longrightarrow \quad \frac{\partial e_k(n)}{\partial Y_k(n)} = 1$$

$$Y_k(n) = g(Yin_k(n)) \quad \Longrightarrow \quad \frac{\partial Y_k(n)}{\partial Yin_k(n)} = \dot{g}(Yin_k(n))$$

$$Yin_k(n) = \sum_{i=0}^{h} b_{ki} Z_i(n) \quad \Longrightarrow \quad \frac{\partial Yin_k(n)}{\partial b_{ki}} = Z_i(n)$$

$$\frac{\partial E(n)}{\partial b_{ki}} = \frac{\partial E(n)}{\partial e_k(n)} \cdot \frac{\partial e_k(n)}{\partial Y_k(n)} \cdot \frac{\partial Y_k(n)}{\partial Yin_k(n)} \cdot \frac{\partial Yin_k(n)}{\partial b_{ki}} \quad \Longrightarrow \quad \frac{\partial E(n)}{\partial b_{ki}} = e_k(n).\dot{g}(Yin_k(n)).Z_i(n)$$

CÁLCULO DO GRADIENTE



Cálculo $\dfrac{\partial E_T}{\partial a_{ij}}$

$$E_T = \frac{1}{N}\sum_{n=1}^{N} E(n) \quad \Longrightarrow \quad \frac{\partial E_T}{\partial a_{ij}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial E(n)}{\partial a_{ij}}$$

$$\frac{\partial E(n)}{\partial a_{ij}} = \frac{\partial E(n)}{\partial Z_i(n)} \cdot \frac{\partial Z_i(n)}{\partial a_{ij}}$$

CÁLCULO DO GRADIENTE

$$\frac{\partial E(n)}{\partial a_{ij}} = \frac{\partial E(n)}{\partial Z_i(n)} . \frac{\partial Z_i(n)}{\partial a_{ij}}$$

Cálculo $\frac{\partial \mathrm{E(n)}}{\partial \mathrm{Z_i(n)}}$

$$E(n) = \frac{1}{2}\sum_{k=1}^{ns} e_k(n)^2 \longrightarrow \frac{\partial E(n)}{\partial Z_i} = \sum_{k=1}^{ns} \frac{\partial E(n)}{\partial e_k(n)} . \frac{\partial e_k(n)}{\partial Z_i}$$

$$E(n) = \frac{1}{2}\sum_{k=1}^{ns} e_k(n)^2 \longrightarrow \frac{\partial E(n)}{\partial e_k(n)} = e_k(n)$$

$$\frac{\partial E(n)}{\partial Z_i(n)} = \sum_{k=1}^{ns} e_k(n) . \frac{\partial e_k(n)}{\partial Z_i(n)}$$

$$\frac{\partial e_k(n)}{\partial Z_i(n)} = \frac{\partial e_k(n)}{\partial Y_k(n)} . \frac{\partial Y_k(n)}{\partial Yin_k(n)} . \frac{\partial Yin_k(n)}{\partial Z_i(n)}$$

CÁLCULO DO GRADIENTE

$$\frac{\partial e_k(n)}{\partial Z_i(n)} = \frac{\partial e_k(n)}{\partial Y_k(n)} \cdot \frac{\partial Y_k(n)}{\partial Yin_k(n)} \cdot \frac{\partial Yin_k(n)}{\partial Z_i(n)}$$

$$e_k(n) = Y_k(n) - d_k(n) \quad \Rightarrow \quad \frac{\partial e_k(n)}{\partial Y_k(n)} = 1$$

$$Y_k(n) = g(Yin_k(n)) \quad \Rightarrow \quad \frac{\partial Y_k(n)}{\partial Yin_k(n)} = \dot{g}(Yin_k(n))$$

$$Yin_k(n) = \sum_{i=0}^{h} b_{ki} Z_i(n) \quad \Rightarrow \quad \frac{\partial Yin_k(n)}{\partial Z_i(n)} = b_{ki}$$

$$\frac{\partial e_k(n)}{\partial Z_i(n)} = \frac{\partial e_k(n)}{\partial Y_k(n)} \cdot \frac{\partial Y_k(n)}{\partial Yin_k(n)} \cdot \frac{\partial Yin_k(n)}{\partial Z_i(n)} \quad \Rightarrow \quad \frac{\partial e_k(n)}{\partial Z_i(n)} = \dot{g}(Yin_k(n)).b_{ki}(n)$$

Cálculo do gradiente

$$\frac{\partial e_k(n)}{\partial Z_i(n)} = \dot{g}(Yin_k(n)).b_{ki}(n)$$

$$\frac{\partial E(n)}{\partial Z_i(n)} = \sum_{k=1}^{ns} e_k(n).\frac{\partial e_k(n)}{\partial Z_i(n)} \Rightarrow \frac{\partial E(n)}{\partial Z_i(n)} = \sum_{k=1}^{ns} e_k(n).\dot{g}(Yin_k(n)).b_{ki}$$

$$\frac{\partial E(n)}{\partial a_{ij}} = \frac{\partial E(n)}{\partial Z_i(n)}.\frac{\partial Z_i(n)}{\partial a_{ij}}$$

Cálculo $\dfrac{\partial Z_i(n)}{\partial a_{ij}}$

Cálculo do gradiente

Cálculo $\dfrac{\partial Z_i(n)}{\partial a_{ij}}$

$$\frac{\partial Z_i(n)}{\partial a_{ij}} = \frac{\partial Z_i(n)}{\partial Zin_i(n)} \cdot \frac{\partial Zin_i(n)}{\partial a_{ij}}$$

$$Z_i(n) = f(Zin_i(n)) \quad \Rightarrow \quad \frac{\partial Z_i(n)}{\partial Zin_i(n)} = \dot{f}(Zin_k(n))$$

$$Zin_i(n) = \sum_{j=0}^{ne} a_{ij} X_j(n) \quad \Rightarrow \quad \frac{\partial Z_i(n)}{\partial Zin_i(n)} = X_j(n)$$

$$\frac{\partial Z_i(n)}{\partial a_{ij}} = \frac{\partial Z_i(n)}{\partial Zin_i(n)} \cdot \frac{\partial Zin_i(n)}{\partial a_{ij}} \quad \Rightarrow \quad \frac{\partial Z_i(n)}{\partial a_{ij}} = \dot{f}(Zin_i(n)).X_j(n)$$

CÁLCULO DO GRADIENTE

Lembrando

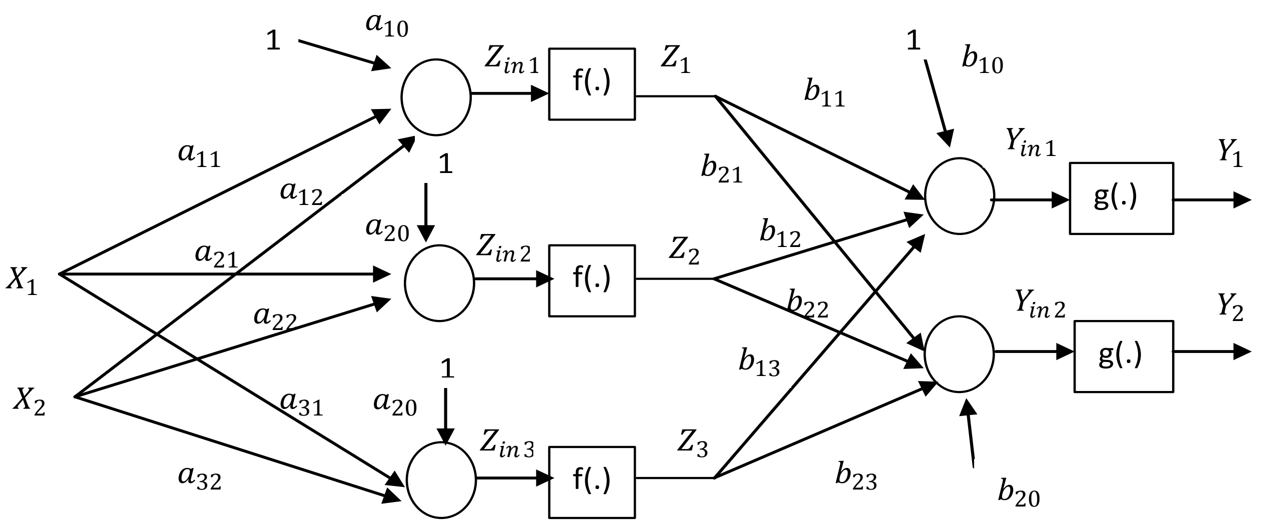
$$\frac{\partial E(n)}{\partial a_{ij}} = \frac{\partial E(n)}{\partial Z_i(n)} \cdot \frac{\partial Z_i(n)}{\partial a_{ij}}$$

$$\frac{\partial Z_i(n)}{\partial a_{ij}} = \dot{f}(Zin_i(n)).X_j(n)$$

$$\frac{\partial E(n)}{\partial Z_i(n)} = \sum_{k=1}^{ns} e_k(n).\dot{g}(Yin_k(n)).b_{ki}$$

$$\frac{\partial E(n)}{\partial a_{ij}} = \left(\sum_{k=1}^{ns} e_k(n).\dot{g}(Yin_k(n)).b_{ki}\right)\dot{f}(Zin_i(n)).X_j(n)$$

CÁLCULO DO GRADIENTE



$$\frac{\partial E(n)}{\partial b_{ki}} = e_k(n).\dot{g}(Yin_k(n)).Z_i(n)$$

$$\frac{\partial E(n)}{\partial a_{ij}} = \left(\sum_{k=1}^{ns} e_k(n).\dot{g}(Yin_k(n)).b_{ki}\right)\dot{f}(Zin_i(n)).X_j(n)$$

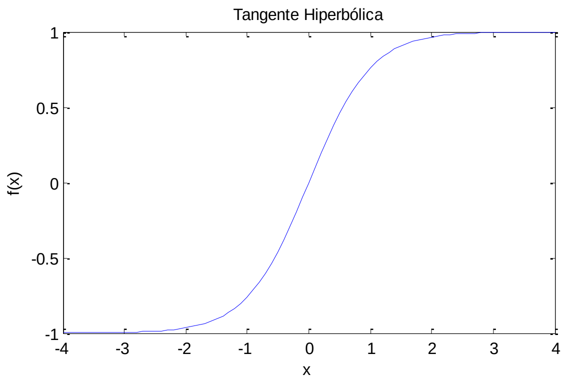
CÁLCULO DO GRADIENTE

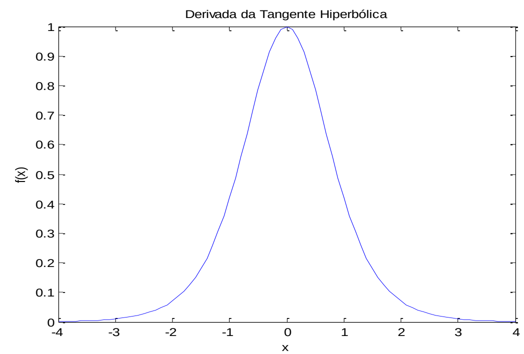
Tangente Hiperbólica

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

DERIVADA DA TANGENTE HIPERBÓLICA

$$\dot{f}(x) = 1 - f(x)^2$$



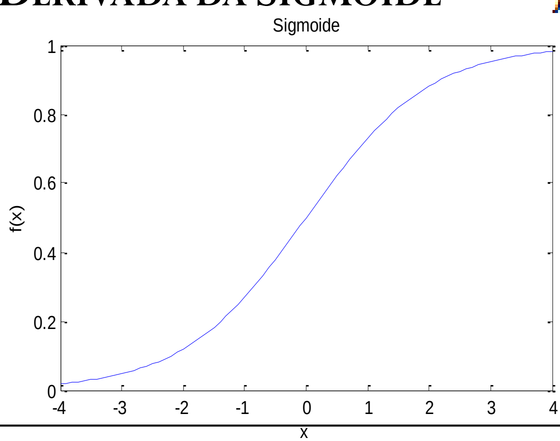


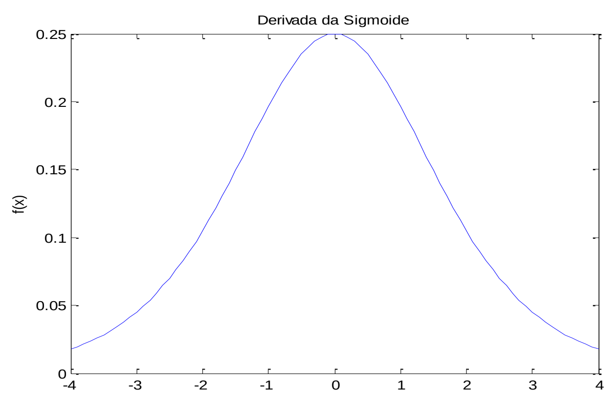
SIGMÓIDE

$$f(x) = \frac{1}{1 + e^{x}}$$

DERIVADA DA SIGMÓIDE

$$\dot{f}(x) = \left(1 - f(x)\right) f(x)$$





CÁLCULO DO GRADIENTE

PROCESSO ITERATIVO I – MÉTODO PADRÃO A PADRÃO

defina o número de neurônios na camada escondida
defina uma condição inicial para o vetor de pesos
defina um escalar ε >0 arbitrariamente pequeno
defina o número épocas máximo
calcule $E_T = \frac{1}{N}\sum_{n=1}^{N} E(n)$
faça k=0
enquanto $E_T \geq \varepsilon \;\&\; nep \leq nepmax$
    $nep = nep + 1$
    ordene aleatoriamente os padrões de entrada-saída
    para n de 1 até N faça
        calcule a saída da rede (equações 1, 3 4)
        calcula $E(n)$
        calcule $\frac{\partial E(n)}{\partial a_{ij}}, \frac{\partial E(n)}{\partial b_{ki}}$
        $a_{ij} = a_{ij} - \alpha \frac{\partial E(n)}{\partial a_{ij}},$
        $b_{ki} = b_{ki} - \alpha \frac{\partial E(n)}{\partial b_{ki}},$
    calcule $E_T = \frac{1}{N}\sum_{n=1}^{N} E(n)$
fimenquanto

CÁLCULO DO GRADIENTE

PROCESSO ITERATIVO I- MÉTODO EM LOTE OU BATELADA

defina o número de neurônios na camada escondida

defina uma condição inicial para o vetor de pesos

defina um escalar ε > 0 arbitrariamente pequeno

defina o número épocas máximo

calcule $E_T = \frac{1}{N}\sum_{n=1}^{N} E(n)$

faça k=0

enquanto $E_T \geq \varepsilon \; \& \; nep \leq nepmax$

→ $nep = nep + 1$

→ para n de 1 até N faça

→ → calcule a saída da rede (equações 1, 3 4)

→ → calcula $E(n)$

→ → calcule $\frac{\partial E(n)}{\partial a_{ij}}, \frac{\partial E(n)}{\partial b_{ki}}$

→ calcule $\frac{\partial E_T}{\partial a_{ij}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial E(n)}{\partial a_{ij}}$

→ calcule $\frac{\partial E_T}{\partial b_{ki}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial E(n)}{\partial b_{ki}}$

→ atualize $a_{ij} = a_{ij} - \alpha \frac{\partial E_T}{\partial a_{ij}}$,

→ atualize $b_{ki} = b_{ki} - \alpha \frac{\partial E_T}{\partial b_{ki}}$,

→ calcule $E_T = \frac{1}{N}\sum_{n=1}^{N} E(n)$

fimenquanto

CÁLCULO DO GRADIENTE

PROCESSO ITERATIVO I I- MÉTODO PADRÃO A PADRÃO

defina o número de neurônios na camada escondida

defina uma condição inicial para o vetor de pesos

defina um escalar ε > 0 arbitrariamente pequeno, $0 < r < 1$, $q > 1$

defina o número épocas máximo, $\alpha = 1$, $nep = 0$

calcule $E_T = \frac{1}{N}\sum_{n=1}^{N} E(n)$

enquanto $E_T \geq \varepsilon$ & $nep \leq nepmax$

- $nep = nep + 1$
- ordene aleatoriamente os padrões de entrada-saída
- para n de 1 até N faça
  - calcule a saída da rede (equações 1, 3 4)
  - calcula $E(n)$
  - calcule $\frac{\partial E(n)}{\partial a_{ij}}, \frac{\partial E(n)}{\partial b_{ki}}$
  - $\nabla E(n) = \begin{bmatrix} \frac{\partial E(n)}{\partial a_{ij}} \\ \vdots \\ \frac{\partial E(n)}{\partial b_{ki}} \end{bmatrix}$
  - $\nabla E(n) = \frac{\nabla E(n)}{\|\nabla E(n)\|}$
  - calcule $\frac{\partial E(n)}{\partial a_{ij}}, \frac{\partial E(n)}{\partial b_{ki}}$ a partir de $\nabla E(n)$

Cálculo do gradiente

Processo Iterativo I I- Método Padrão a Padrão

$a_{ij}^{prov} = a_{ij} - \alpha \frac{\partial E(n)}{\partial a_{ij}},$

$b_{ki}^{prov} = b_{ki} - \alpha \frac{\partial E(n)}{\partial b_{ki}},$

calcula $E(n)^{prov}$

enquanto $E(n)^{prov} > E(n)$

- $\alpha = r\alpha$
- $a_{ij}^{prov} = a_{ij} - \alpha \frac{\partial E(n)}{\partial a_{ij}},$
- $b_{ki}^{prov} = b_{ki} - \alpha \frac{\partial E(n)}{\partial b_{ki}},$
- calcula $E(n)^{prov}$

fimenquanto

$E(n) = E(n)^{prov}$

$\alpha = q\alpha$

fimpara

calcule $E_T = \frac{1}{N} \sum_{n=1}^{N} E(n)$

fimenquanto

Cálculo do gradiente

Processo Iterativo I I- Método em Lote ou batelada

defina o número de neurônios na camada escondida

defina uma condição inicial para o vetor de pesos

defina um escalar ε > 0 arbitrariamente pequeno, $0 < r < 1, q > 1$

defina o número épocas máximo, $\alpha = 1, nep = 0$

calcule $E_T = \frac{1}{N}\sum_{n=1}^{N} E(n)$

enquanto $E_T \geq \varepsilon \;\&\; nep \leq nepmax$

- $nep = nep + 1$
- ordene aleatoriamente os padrões de entrada-saída
- para n de 1 até N faça
  - calcule a saída da rede (equações 1, 3 4)
  - calcula $E(n)$
  - calcule $\frac{\partial E(n)}{\partial a_{ij}}, \frac{\partial E(n)}{\partial b_{ki}}$
- fimpara
- calcule $\frac{\partial E_T}{\partial b_{ki}} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial E(n)}{\partial b_{ki}}, \frac{\partial E_T}{\partial a_{ij}} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial E(n)}{\partial a_{ij}}$
- $\nabla E_T = \begin{bmatrix} \frac{\partial E_T}{\partial a_{ij}} \\ \vdots \\ \frac{\partial E_T}{\partial b_{ki}} \end{bmatrix}$

CÁLCULO DO GRADIENTE

PROCESSO ITERATIVO I I– MÉTODO EM LOTE OU BATELADA

- $\nabla E_T = \dfrac{\nabla E_T}{\|\nabla E_T\|}$
- calcule $\dfrac{\partial E_T}{\partial a_{ij}}, \dfrac{\partial E_T}{\partial b_{ki}}$ a partir de ∇E_T
- $a_{ij}^{prov} = a_{ij} - \alpha \dfrac{\partial E_T}{\partial a_{ij}},$
- $b_{ki}^{prov} = b_{ki} - \alpha \dfrac{\partial E_T}{\partial b_{ki}},$
- calcula $E_T{}^{prov}$
- enquanto $E_T{}^{prov} > E_T$
  - $\alpha = r\alpha$
  - $a_{ij}^{prov} = a_{ij} - \alpha \dfrac{\partial E_T}{\partial a_{ij}},$
  - $b_{ki}^{prov} = b_{ki} - \alpha \dfrac{\partial E_T}{\partial b_{ki}},$
  - calcula $E_T{}^{prov}$
- fimenquanto
- $E_T = E_T{}^{prov}$
- $\alpha = q\alpha$

fimenquanto

EXEMPLO

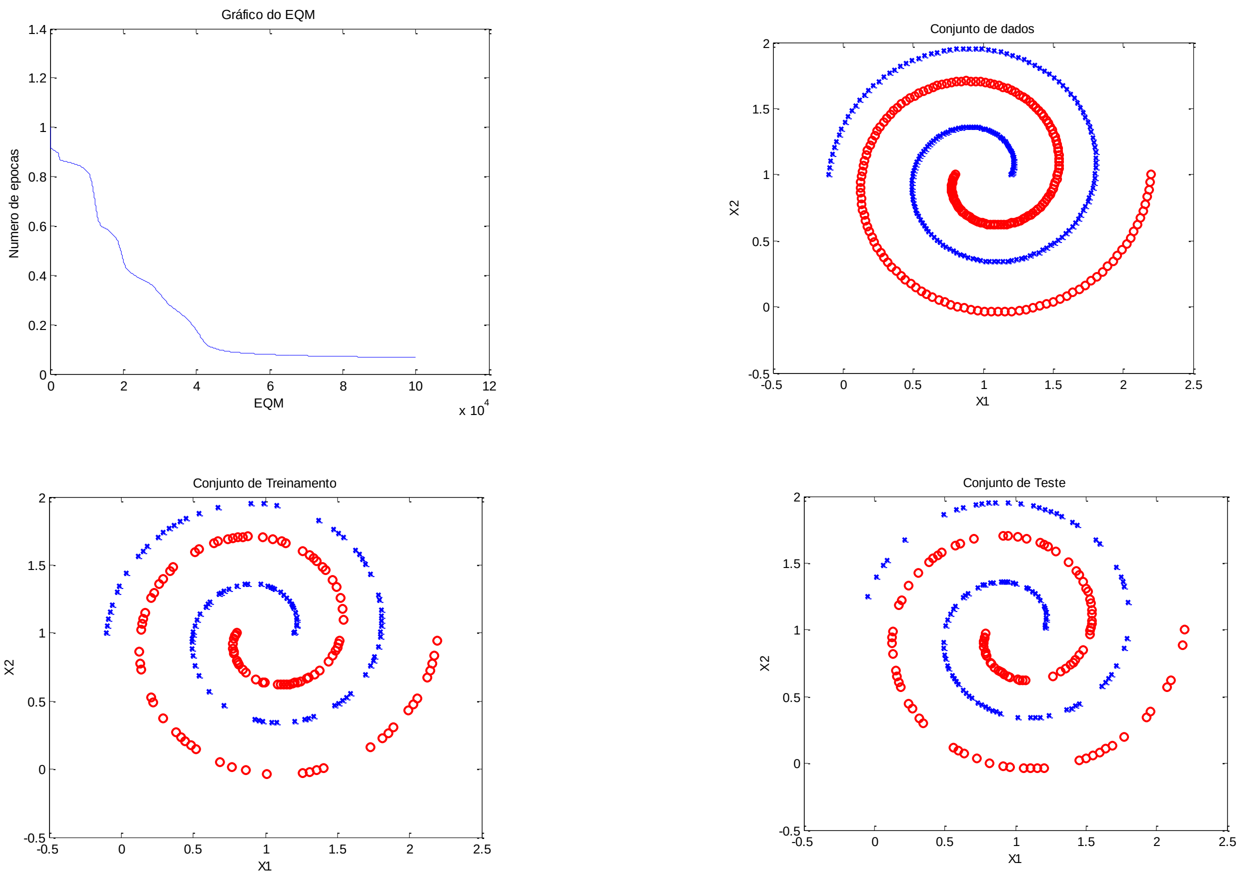


Gráfico do EQM
Numero de epocas
EQM
x 10^4
Conjunto de dados
X1
X2
Conjunto de Treinamento
X1
X2
Conjunto de Teste
X1
X2

Rede Neural com Função de Base Radial - RBF

Rede Neural com Função de Base Radial - RBF

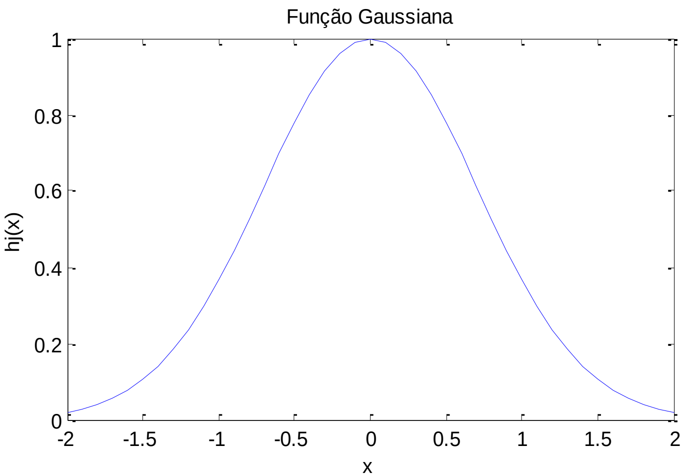
- Uma função de ativação de base radial é caracterizada por apresentar uma resposta que decresce (ou cresce) monotonicamente com a distância a um ponto central
- O centro e a taxa de decrescimento (ou crescimento) em cada direção são alguns dos parâmetros a serem definidos.
- Uma função de base radial monotonicamente decrescente típica é a função gaussiana, dada na forma

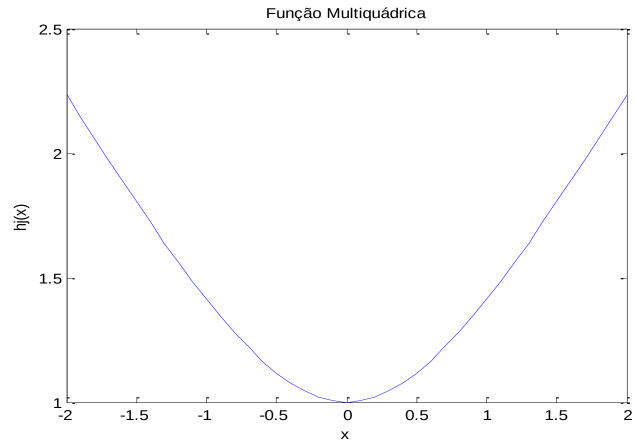
$$h_j(x) = exp\left(-\frac{(x - c_j)^2}{r_j^2}\right)$$

- Uma função de base radial monotonicamente crescente típica é a função multiquádrica, dada na forma

$$h_j(x) = \frac{\sqrt{r_j^2 + (x - c_j)^2}}{r_j}$$

Rede Neural com Função de Base Radial - RBF





- cj=0
- rj=1;
- x=-2:0.1:2;
- hj=exp(-(x-cj).^2/rj^2)
- figure(1)
- plot(x,hj)
- xlabel('x')
- ylabel('hj(x)')
- title('Função Gaussiana')

- cj=0
- rj=1;
- x=-2:0.1:2;
- hj=sqrt(rj^2+(x-cj).^2)/rj
- figure(2)
- plot(x,hj)
- xlabel('x')
- ylabel('hj(x)')
- title('Função Multiquádrica')

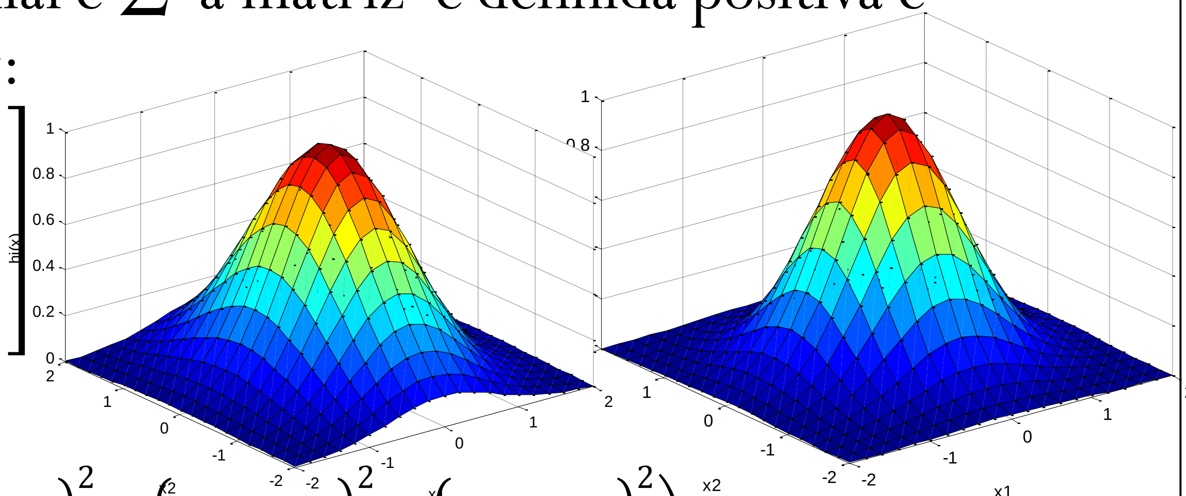
Rede Neural com Função de Base Radial - RBF

- No caso multidimensional, a função gaussiana $h_j(x)$ assume a forma

$$h_j(x) = \exp\left(-(x - c_j)^T \Sigma^{-1} (x - c_j)\right)$$

- onde $\boldsymbol{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$ é o veto de entradas,
- $c_j = [c_{j1} \quad c_{j2} \quad \cdots \quad c_{jn}]^T$ é o vetor que define o centro da função de base radial e Σ a matriz é definida positiva e diagonal, dada por:

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



- de modo que

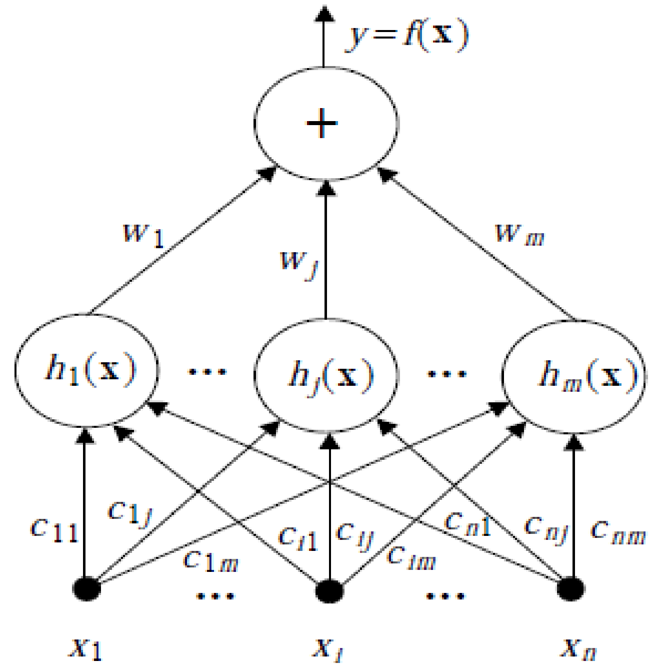
$$h_j(x) = \exp\left(-\frac{(x_{j1} - c_{j1})^2}{\sigma_{j1}} - \frac{(x_{j1} - c_{j2})^2}{\sigma_{j2}} \cdots \frac{(x_{j1} - c_{j2})^2}{\sigma_{jn}}\right)$$

Rede Neural com Função de Base Radial - RBF

- Neste caso, os elementos do vetor

  $$\sigma_j = [\sigma_{j1} \quad \sigma_{j2} \quad \cdots \quad \sigma_{jn}]^T$$

  são responsáveis pela taxa de decrescimento da gaussiana



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

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

Treinamento de uma Rede Neural com função de base radial - RBF

- Assim , com um modelo de classificação linear na forma

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

- 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-saida

$$\min_w J(w) = \min_w \sum_{i=1}^{N} (s_i - f(x_i))^2 = \min_w \sum_{i=1}^{N} \left(s_i - \sum_{j=1}^{m} w_j h(x_i) \right)^2$$

- O sistema de equações resultante é dado na forma

$$\frac{\partial J}{\partial w_j} = -2 \sum_{i=1}^{N} (s_i - f(x_i)) \frac{\partial f}{\partial w_j} = -2 \sum_{i=1}^{N} (s_i - f(x_i)) h_j(x_j), \quad j = 1, \cdots, m$$

Treinamento de uma Rede Neural com função de base radial - RBF

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

$$\frac{\partial J}{\partial w_j} = -2\sum_{i=1}^{N}(s_i - f(x_i))\frac{\partial f}{\partial w_j} = -2\sum_{i=1}^{N}(s_i - f(x_i))h_j(x_j), \quad j = 1, \cdots, m$$

- separando-se os termos que envolvem a incógnita $f(x_i)$, resulta:

$$\sum_{i=1}^{N} f(x_i)h_j(x_j) = \sum_{i=1}^{N}\left[\sum_{r=1}^{m} w_r h_r(x_i)\right]h_j(x_i) = \sum_{i=1}^{m} s_i h_j(x_i), \quad j = 1, \cdots, m$$

- portanto, existem m equações para obter as m incógnitas $\{w_r, r = 1, \cdots, m\}$

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

$$h_j^T f = h_j^T s, \quad j = 1, \cdots, m$$

$$\sum_{i=1}^{N}\left[\sum_{r=1}^{m} w_r h_r(x_i)\right] h_j(x_i) = \sum_{i=1}^{m} s_i h_j(x_i), \quad j = 1, \cdots, m$$

$$h_j^T f = h_j^T s, \quad j = 1, \cdots, m$$

- onde

$$h_j = \begin{bmatrix} h_j(x_1) \\ \vdots \\ h_j(x_N) \end{bmatrix}, f = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} = \begin{bmatrix} \sum_{r=1}^{m} w_r h_r(x_1) \\ \vdots \\ \sum_{j=1}^{m} w_r h_r(x_N) \end{bmatrix}, e\ s = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}$$

- como existem *m* equações, resulta:

$$\begin{bmatrix} h_1^T f \\ \vdots \\ h_m^T f \end{bmatrix} = \begin{bmatrix} h_1^T s \\ \vdots \\ h_m^T s \end{bmatrix}$$

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

$$H = [h_1 \quad h_2 \quad \cdots \quad h_m] = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_m(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_N) & h_2(x_N) & \cdots & h_m(x_N) \end{bmatrix}$$

$$\begin{bmatrix} h_1^T f \\ \vdots \\ h_m^T f \end{bmatrix} = \begin{bmatrix} h_1^T s \\ \vdots \\ h_m^T s \end{bmatrix}$$

$$H = [h_1 \quad h_2 \quad \cdots \quad h_m] = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_m(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_N) & h_1(x_N) & \cdots & h_m(x_N) \end{bmatrix}$$

- sendo possível reescrever o sistema de equações lineares como segue:

$$H^T f = H^T s$$

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

$$f_i = f(x_i) = \sum_{r=1}^{m} w_r h_r(x_i) = [h_1(x_i) \quad h_2(x_i) \quad \cdots \quad h_m(x_i)]w$$

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

$$f = Hw$$

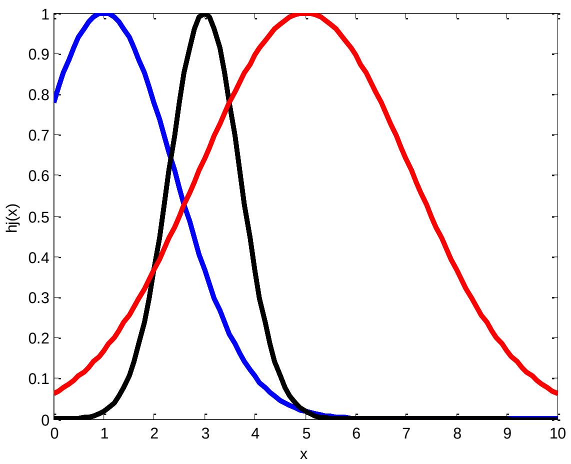
- Logo

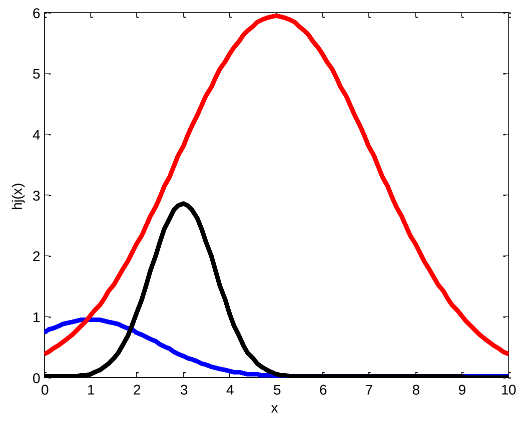
$$H^T H w = H^T s$$

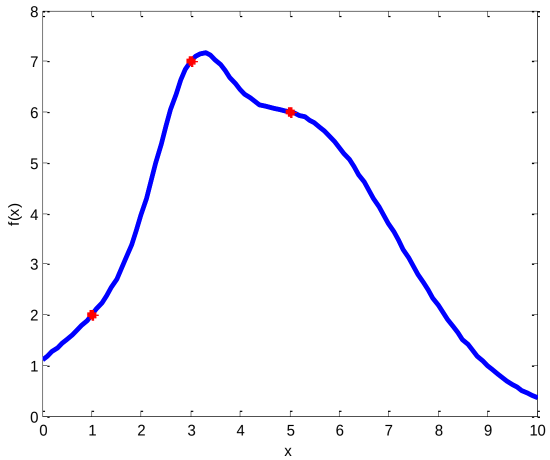
$$H^T H w = H^T s$$

$$H^T H w = H^T s \Rightarrow \boxed{w = (H^T H)^{-1} H^T s}$$

- **Aproximação usando rede neural RBF**
- Assuma que foram amostrados, na presença de ruído, gerando o conjunto de treinamento: $\{(1,2),(3,7),(5,6)\}$







Codigo

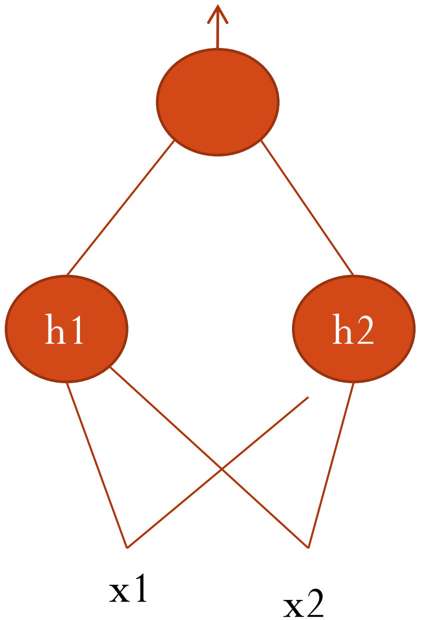
```
x=[1; 3; 5]; % entrada
y=[2; 7;6]; % saida
c=[1 3 5]; % centros
r=[2 1 3]; % variancia
N=length(x); % Numero de pontos de treinamento
m=length(c); % Numero de funções rbf
for i=1:N,
   for j=1:m,
      H(i,j)=exp(-(x(i)-c(j))*(x(i)-c(j))/r(j)^2);
   end
end
w=inv(H'*H)*H'*y;
y=H*w;
```

```
%Gerando as funções
x1=0:0.1:10,
for i=1:length(x),
   for j=1:m,
      h(i,j)=exp(-(x1(i)-c(j))*(x1(i)-
c(j))/r(j)^2);
   end
end
figure(1)
plot(x1,h(:,1),'b','linewidth',3)
hold on
plot(x1,h(:,2),'k','linewidth',3)
plot(x1,h(:,3),'r','linewidth',3)
xlabel('x'), ylabel('hj(x)')
figure(2)
plot(x1,w(1)*h(:,1),'b','linewidth',3)
hold on
plot(x1,w(2)*h(:,2),'k','linewidth',3)
plot(x1,w(3)*h(:,3),'r','linewidth',3)
xlabel('x‘),ylabel('hj(x)')
figure(3)
plot(x1,h*w,'b','linewidth',3)
hold on
plot(x,y,'r*','linewidth',3)
xlabel('x'), ylabel('f(x)')
```

Exemplo

Problema Ou-Exclusivo

X1	x2	y
0	0	1
1	0	0
0	1	0
1	1	1



Neurônio	c1	c2
h1(x)	0	0
h2(x)	1	1

Neurônio	σ1	σ2
h1(x)	1	1
h2(x)	1	1

Exemplo

$$\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \boldsymbol{x}_3\\ \boldsymbol{x}_4\end{bmatrix} = \begin{bmatrix}x_{11} & x_{12}\\ x_{21} & x_{22}\\ x_{31} & x_{32}\\ x_{41} & x_{42}\end{bmatrix} \qquad \begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \boldsymbol{x}_3\\ \boldsymbol{x}_4\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 1\\ 1 & 0\\ 1 & 1\end{bmatrix} \qquad \begin{bmatrix}\boldsymbol{s}_1\\ \boldsymbol{s}_2\\ \boldsymbol{s}_3\\ \boldsymbol{s}_4\end{bmatrix} = \begin{bmatrix}1\\ 0\\ 0\\ 1\end{bmatrix}$$

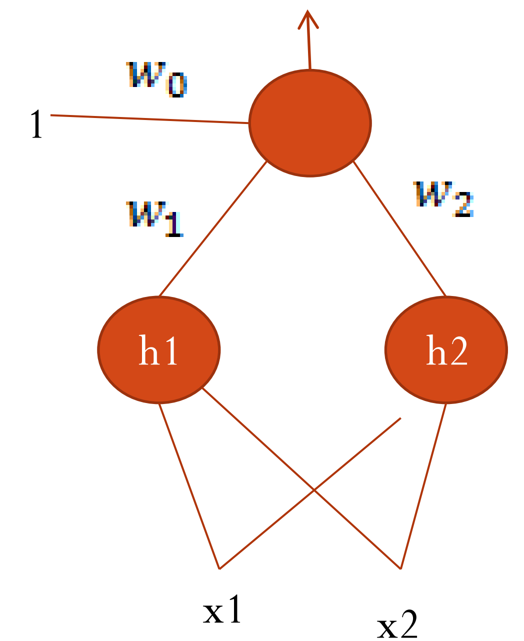
$$f(\boldsymbol{x}_1) = w_0 + w_1 h_1(\boldsymbol{x}_1) + w_2 h_2(\boldsymbol{x}_1)$$

$$f(\boldsymbol{x}_2) = w_0 + w_1 h_1(\boldsymbol{x}_2) + w_2 h_2(\boldsymbol{x}_2)$$

$$f(\boldsymbol{x}_3) = w_0 + w_1 h_1(\boldsymbol{x}_3) + w_2 h_2(\boldsymbol{x}_3)$$

$$f(\boldsymbol{x}_4) = w_0 + w_1 h_1(\boldsymbol{x}_4) + w_2 h_2(\boldsymbol{x}_4)$$

$$\frac{\partial f(\boldsymbol{x}_1)}{\partial w_0} = 1 \qquad \frac{\partial f(\boldsymbol{x}_1)}{\partial w_1} = h_1(\boldsymbol{x}_1)$$



$$\begin{bmatrix}f(\boldsymbol{x}_1)\\ f(\boldsymbol{x}_2)\\ f(\boldsymbol{x}_3)\\ f(\boldsymbol{x}_4)\end{bmatrix} = \begin{bmatrix}1 & h_1(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_1)\\ 1 & h_1(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_2)\\ 1 & h_1(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_3)\\ 1 & h_1(\boldsymbol{x}_4) & h_2(\boldsymbol{x}_4)\end{bmatrix}\begin{bmatrix}w_0\\ w_1\\ w_2\\ w_3\end{bmatrix}$$

$$f = Hw$$

Exemplo

$$\min_w J(w) = \min_w \sum_{i=1}^{4} (s_i - f(x_i))^2$$

Calculando a derivada em relação a wj

$$\frac{\partial J}{\partial w_j} = -2\sum_{i=1}^{4}(s_i - f(x_i))\frac{\partial f}{\partial w_j} = -2\sum_{i=1}^{4}(s_i - f(x_i))h_j(x_i), \quad j = 0, \cdots, 2 \qquad h_0(x_j) = 1$$

Igualando a zero, temos

$$\sum_{i=1}^{4}(s_i - f(x_i))h_j(x_j) = 0 \quad j = 0, \cdots, m$$

$$\sum_{i=1}^{4} s_i h_j(x_j) - \sum_{i=1}^{4} f(x_i)h_j(x_j) = 0 \quad j = 1, \cdots, m$$

$$\boxed{\sum_{i=1}^{N} s_i h_j(x_j) = \sum_{i=1}^{N} f(x_i)h_j(x_j) \quad j = 0, \cdots, m}$$

Exemplo

$$\sum_{i=1}^{N} s_i h_j(x_i) = \sum_{i=1}^{N} f(x_i) h_j(x_i) \qquad j = 0, \cdots, m$$

- Para j = 0

$$f(x_1) + f(x_2) + f(x_3) + f(x_4) = s_1 + s_2 + s_3 + s_4$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

$$h_0^T f = h_0^T s$$

- onde

$$h_0 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, f = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}, e\ s = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}$$

Exemplo

$$\sum_{i=1}^{N} s_i h_j(x_j) = \sum_{i=1}^{N} f(x_i) h_j(x_j) \qquad j = 0, \cdots, m$$

- Para j = 1

$$f(\boldsymbol{x}_1)h_1(\boldsymbol{x}_1) + f(\boldsymbol{x}_2)h_1(\boldsymbol{x}_2) + f(\boldsymbol{x}_3)h_1(\boldsymbol{x}_3) + f(\boldsymbol{x}_4)h_1(\boldsymbol{x}_4) = \boldsymbol{s}_1 h_1(\boldsymbol{x}_1) + \boldsymbol{s}_2 h_1(\boldsymbol{x}_2) + \boldsymbol{s}_3 h_1(\boldsymbol{x}_3) + \boldsymbol{s}_4 h_1(\boldsymbol{x}_4)$$

$$\begin{bmatrix} h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ f(\boldsymbol{x}_3) \\ f(\boldsymbol{x}_4) \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \\ \boldsymbol{s}_4 \end{bmatrix}$$

$$h_1^T f = h_1^T s$$

$$h_1 = \begin{bmatrix} h_1(x_1) \\ \vdots \\ h_1(x_N) \end{bmatrix}, f = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}, e\ s = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}$$

Exemplo

$$\sum_{i=1}^{N} s_i h_j(x_j) = \sum_{i=1}^{N} f(x_i) h_j(x_j) \qquad j = 0, \cdots, m$$

- Para um j qualquer

$$f(x_1)h_j(x_1) + f(x_2)h_j(x_2) + f(x_3)h_j(x_3) + f(x_4)h_j(x_4) = s_1 h_j(x_1) + s_2 h_j(x_2) + s_3 h_j(x_3) + s_4 h_j(x_4)$$

$$h_j = \begin{bmatrix} h_j(x_1) \\ \vdots \\ h_j(x_N) \end{bmatrix}, f = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}, e\ s = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}$$

$$h_j^T f = h_j^T s$$

Exemplo

$$\sum_{i=1}^{N} s_i h_j(x_i) = \sum_{i=1}^{N} f(x_i) h_j(x_i) \qquad j = 0, \cdots, m$$

- De forma geral

$$\begin{bmatrix} h_0^T f \\ \vdots \\ h_m^T f \end{bmatrix} = \begin{bmatrix} h_0^T s \\ \vdots \\ h_m^T s \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \\ h_2(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ f(\boldsymbol{x}_3) \\ f(\boldsymbol{x}_4) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \\ h_2(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

- Sabendo que

$$\begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ f(\boldsymbol{x}_3) \\ f(\boldsymbol{x}_4) \end{bmatrix} = \begin{bmatrix} 1 & h_1(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_1) \\ 1 & h_1(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_2) \\ 1 & h_1(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_3) \\ 1 & h_1(\boldsymbol{x}_4) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

- Logo

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \\ h_2(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} 1 & h_1(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_1) \\ 1 & h_1(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_2) \\ 1 & h_1(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_3) \\ 1 & h_1(\boldsymbol{x}_4) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \\ h_2(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

Exemplo

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \\ h_2(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} 1 & h_1(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_1) \\ 1 & h_1(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_2) \\ 1 & h_1(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_3) \\ 1 & h_1(\boldsymbol{x}_4) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ h_1(\boldsymbol{x}_1) & h_1(\boldsymbol{x}_2) & h_1(\boldsymbol{x}_3) & h_1(\boldsymbol{x}_4) \\ h_2(\boldsymbol{x}_1) & h_2(\boldsymbol{x}_2) & h_2(\boldsymbol{x}_3) & h_2(\boldsymbol{x}_4) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

- Logo

$$H^T * Hw = H^T s$$

$$w = (H^T * H)^{-1} H^T s$$

Calculando a matriz H

- Rede RBF
  - Dois neurônios
  - Centros
    - [0 0], [1 1]
  - σ, todos iguais a 1

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$h_1(\boldsymbol{x}_1) = exp\left(-\left(\frac{\boldsymbol{x}_{11}-\boldsymbol{c}_{11}}{\sigma^2}\right) - \left(\frac{\boldsymbol{x}_{12}-\boldsymbol{c}_{12}}{\sigma^2}\right)\right) = exp\left(-\left(\frac{\boldsymbol{x}_{11}-\boldsymbol{0}}{1^2}\right) - \left(\frac{\boldsymbol{x}_{12}-\boldsymbol{0}}{1^2}\right)\right)$$

$$h_2(\boldsymbol{x}_1) = exp\left(-\left(\frac{\boldsymbol{x}_{11}-\boldsymbol{c}_{21}}{\sigma^2}\right) - \left(\frac{\boldsymbol{x}_{12}-\boldsymbol{c}_{22}}{\sigma^2}\right)\right) = exp\left(-\left(\frac{\boldsymbol{x}_{11}-\boldsymbol{1}}{1^2}\right) - \left(\frac{\boldsymbol{x}_{12}-\boldsymbol{1}}{1^2}\right)\right)$$

$$H = \begin{bmatrix} 1 & 1 & 0.1353 \\ 1 & 0.3679 & 0.3679 \\ 1 & 0.3679 & 0.3679 \\ 1 & 0.1353 & 1 \end{bmatrix} \qquad w = \begin{bmatrix} 1.84 \\ 2.50 \\ 2.50 \end{bmatrix}$$

Cálculo da saída

- Relembrando

$$f(\boldsymbol{x_1}) = w_0 + w_1 h_1(\boldsymbol{x_1}) + w_2 h_2(\boldsymbol{x_1})$$

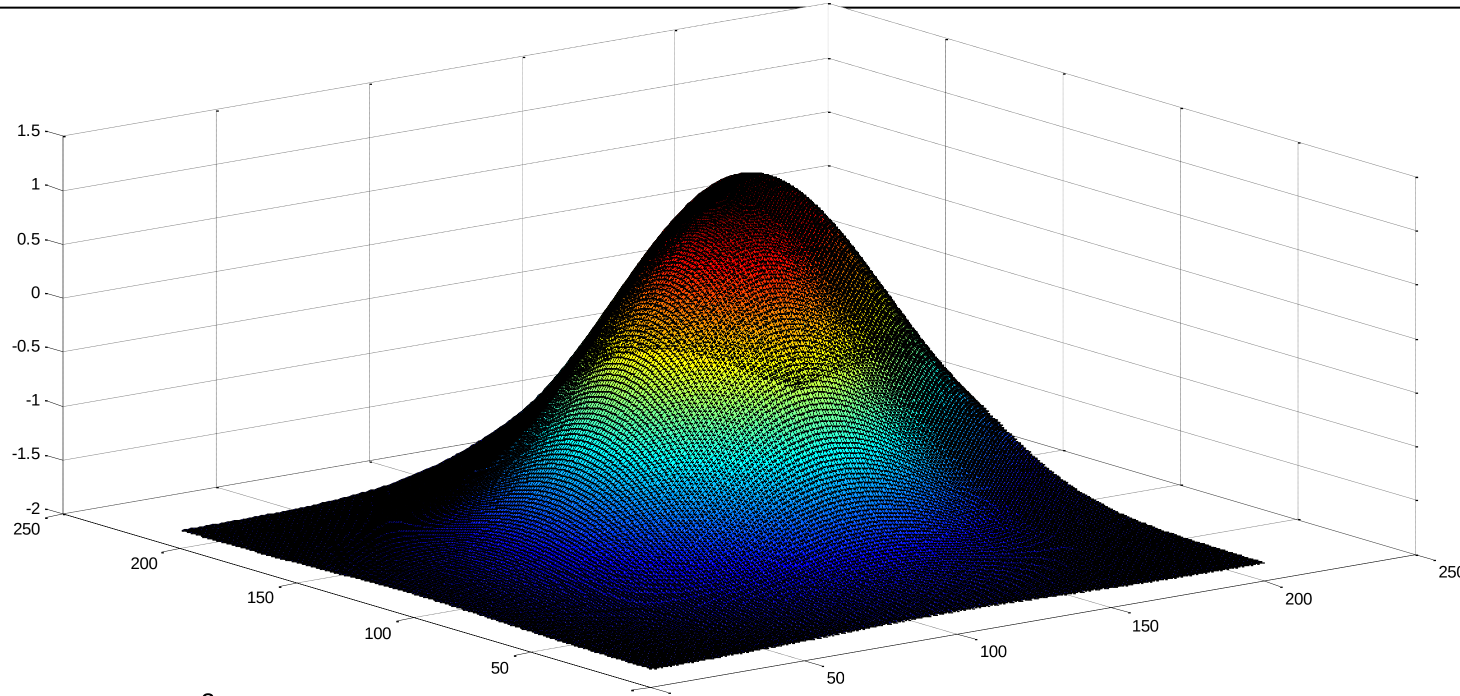
$$f(\boldsymbol{x_2}) = w_0 + w_1 h_1(\boldsymbol{x_2}) + w_2 h_2(\boldsymbol{x_2})$$

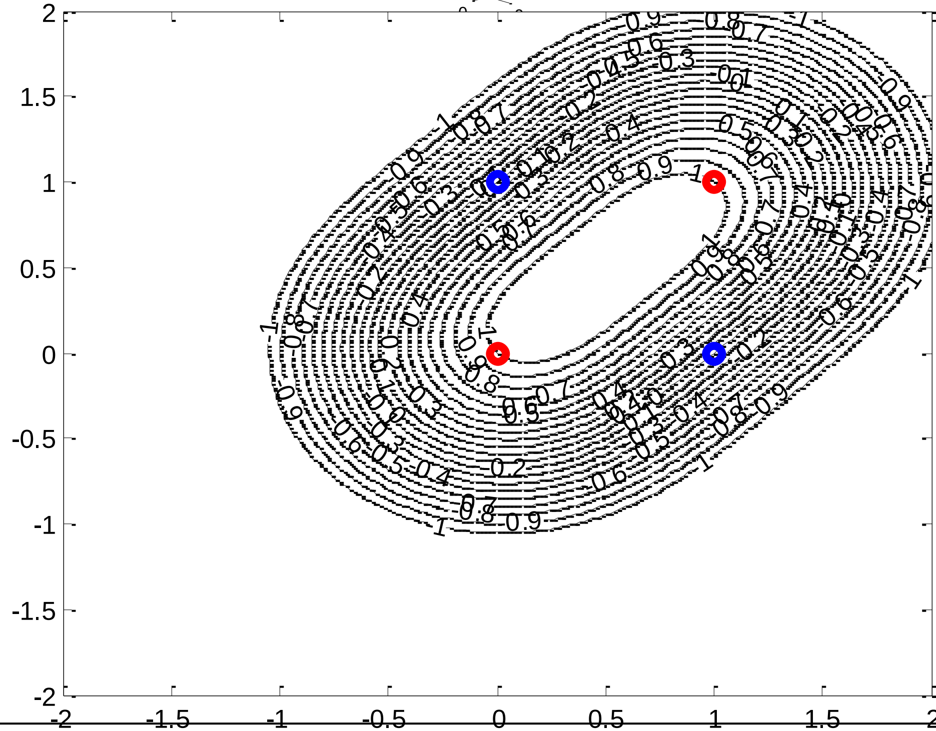
$$f(\boldsymbol{x_3}) = w_0 + w_1 h_1(\boldsymbol{x_3}) + w_2 h_2(\boldsymbol{x_3})$$

$$f(\boldsymbol{x_4}) = w_0 + w_1 h_1(\boldsymbol{x_4}) + w_2 h_2(\boldsymbol{x_4})$$

- Ou, seja $\quad f = Hw$

$$f = \begin{bmatrix} 1 & 1 & 0.1353 \\ 1 & 0.3679 & 0.3679 \\ 1 & 0.3679 & 0.3679 \\ 1 & 0.1353 & 1 \end{bmatrix} \begin{bmatrix} 1.84 \\ 2.50 \\ 2.50 \end{bmatrix} \qquad f = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$


1.5
1
0.5
0
-0.5
-1
-1.5
-2
250
200
150
100
50
50
100
150
200
250


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Implementação deadline 26/09

- Implementar um Rede Neural Artificial