



# Redes Neurais Artificiais

UFPI

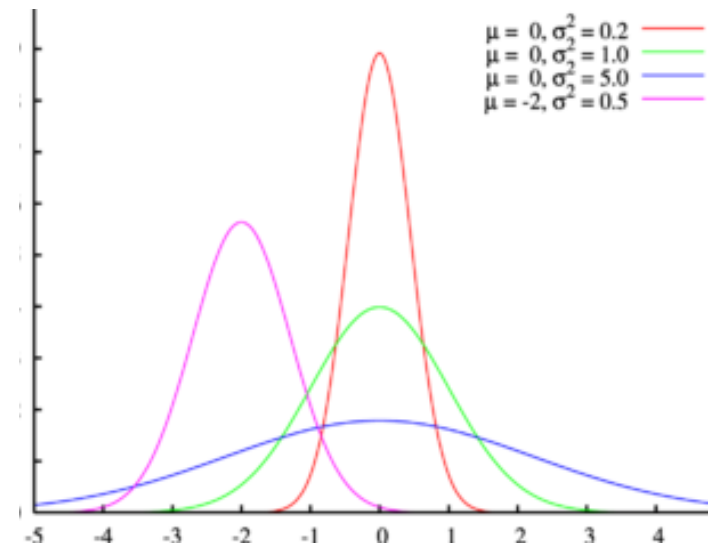
PROF. ME. FILIPE FONTINELE DE ALMEIDA

FILIFEFONTINELI@GMAIL.COM

# Treinamento de RBF

- ▶ Definição:
- ▶ rede neural onde a função de ativação dos neurônios da camada escondida é do tipo de base gaussiana

$$out_i = e^{-\frac{u_i^2}{2\sigma_i^2}}$$
$$u_i = \|x - c_i\|$$

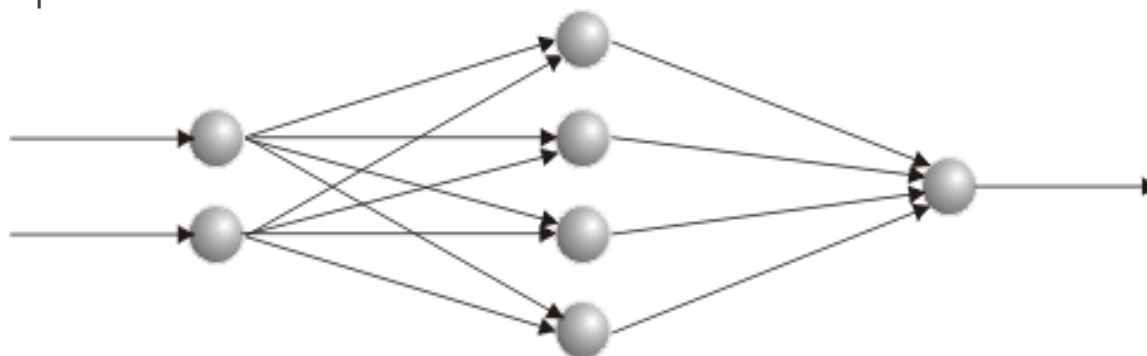


# Treinamento de RBF

- ▶ Definições:
  - ▶  $c_i \rightarrow$  vetor constante que representa o centro do neurônio  $i$  da camada escondida
  - ▶  $\sigma_i \rightarrow$  raio do neurônio  $i$  da camada escondida que define a abertura da função
  - ▶  $|| \cdot || \rightarrow$  norma euclidiana

# Treinamento de RBF

- ▶ Estrutura da rede:
  - ▶ estritamente *feedforward*
  - ▶ uma única camada escondida
  - ▶ quantidade arbitrária de neurônios
  - ▶ bias permitido

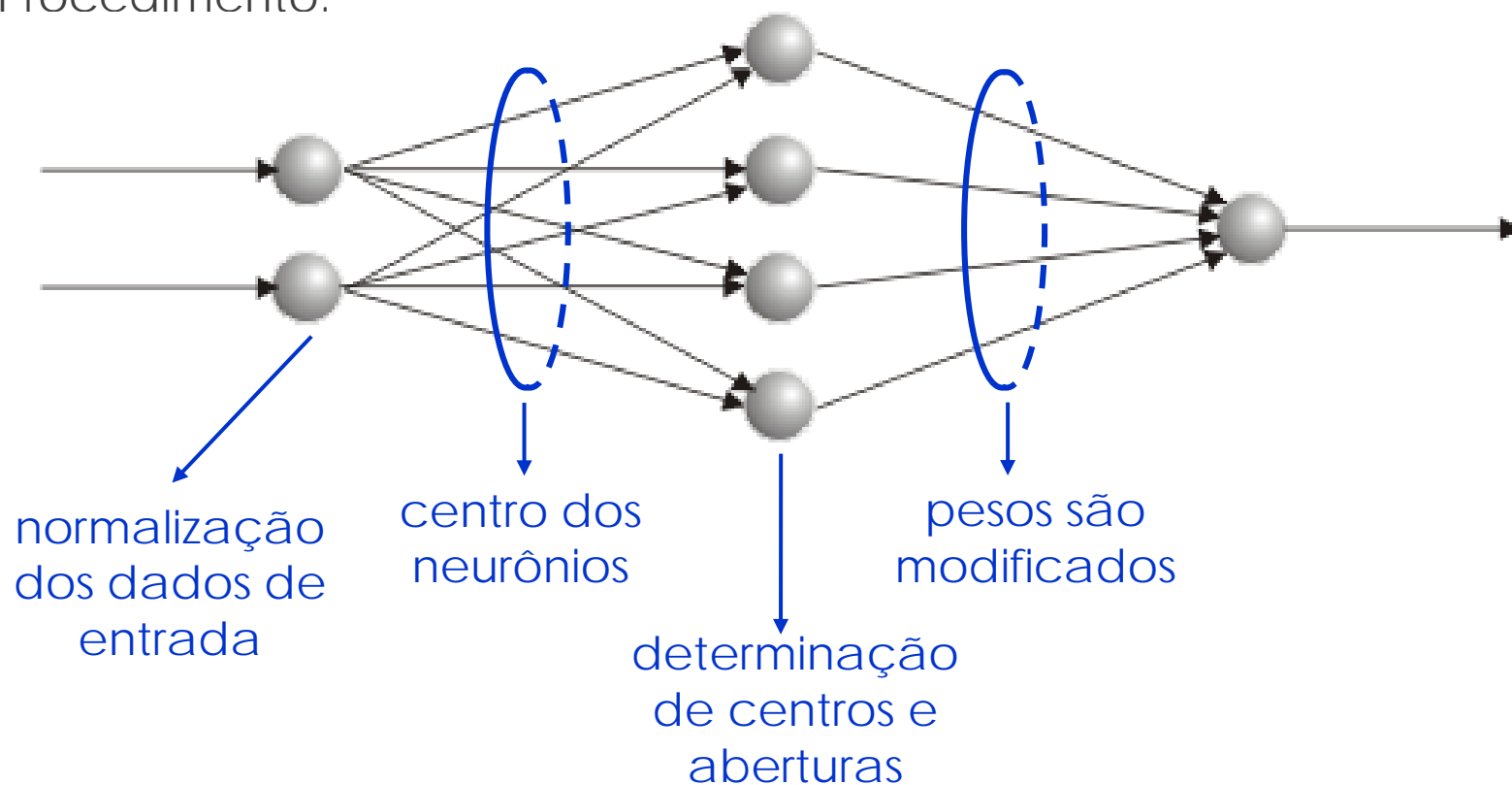


# Treinamento de RBF

- ▶ Requisito:
  - ▶ dados de entrada-saída conhecidos
- ▶ Procedimento:
  - ▶ projeto da camada de entrada
  - ▶ projeto da camada escondida
  - ▶ projeto da camada de saída

# Treinamento de RBF

► Procedimento:



# Treinamento de RBF

- ▶ Projeto da camada de entrada:
  - ▶ número de neurônios dado pelo problema
  - ▶ vetor de entrada  $x = [x_1 \ x_2 \ x_3 \ \dots \ x_m]'$
  - ▶ normalização para  $[0,1]$

$$\bar{x}_k = \frac{x_k - x_k^{\min}}{x_k^{\max} - x_k^{\min}}$$

- ▶ normalização para  $[-1,1]$

$$\bar{x}_k = \frac{2x_k - x_k^{\max} - x_k^{\min}}{x_k^{\max} - x_k^{\min}}$$



# Treinamento de RBF

- ▶ Projeto da camada escondida:
  - ▶ número de neurônios dado pelo projetista
  - ▶ determinação do centro dos neurônios
  - ▶ Algoritmo WTA
    1. definir taxa de aprendizagem  $\eta$
    2. inicializar aleatoriamente os centros  $c_j$
    3. para cada vetor de entrada  $x$  (total de  $N$ ) atualizar centro mais próximo de  $x$ 
$$c_i = c_i + \eta \cdot [x - c_i]$$
    4. voltar ao passo 3 se  $E_q$  grande



# Treinamento de RBF

- ▶ Projeto da camada escondida:
  - ▶ centros dados por  $c_j = [c_{j1} \ c_{j2} \ c_{j3} \ ... \ c_{jm}]'$
  - ▶  $c_{ji}$  representa peso entre neurônio  $i$  da entrada e neurônio  $j$  escondido
  - ▶  $E_q \rightarrow$  erro de quantização

$$E_q = \frac{\sum_{t=1}^N \|x_t - c_n\|^2}{N}$$

# Treinamento de RBF

- ▶ Projeto da camada escondida:
  - ▶ determinação da abertura dos neurônios
    1. raio único:  $\sigma_i = cte = k \cdot d_{\max}(c_i, c_j)$
    2. metade da distância até centro mais perto

$$\sigma_i = \frac{d_{\min}(c_i, c_j)}{2}$$

3. média da distância até centros mais pertos

$$\sigma_i = \sum_{t=1}^T d_{\min}(c_i, c_t) / T$$

# Treinamento de RBF

- ▶ Projeto da camada de saída:
  - ▶ determinação dos pesos
  - ▶ igual ao treinamento da rede MLP

$$\Delta W_{ij} = \eta \cdot e_i \cdot \dot{f}_i \cdot out_j$$

- ▶ considerar demais parâmetros congelados

# Treinamento de RBF

- ▶ Algoritmo:
  - ▶ Definir normalização da entrada
  - ▶ Determinar centro dos neurônios
  - ▶ Determinar a abertura dos neurônios
  - ▶ Congelar centros e aberturas
  - ▶ Determinar pesos da camada de saída

# Treinamento de RBF

- ▶ Observações:
  - ▶ Taxa de aprendizado definido pelo projetista
    - ▶ valores altos aceleram aprendizado
    - ▶ valores baixos evitam oscilações
  - ▶ Número de neurônios definido pelo projetista
  - ▶ Aplicação idêntica à da MLP