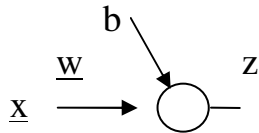


Redes de Base Radial - RBF - Radial Basis Function

1 - Introdução

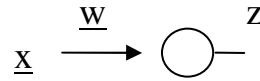
Neurônios tipo tgh(u)



$$u = \underline{w}^t \underline{x} + b$$

$$z = \tanh(u)$$

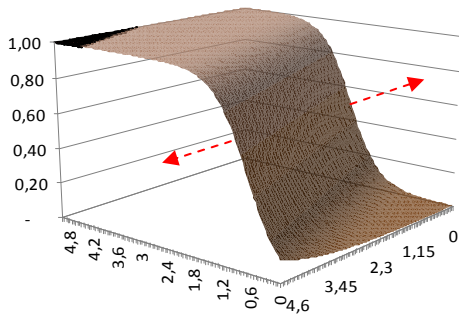
Neurônios de Base Radial



$$u = |\underline{x} - \underline{w}|^2$$

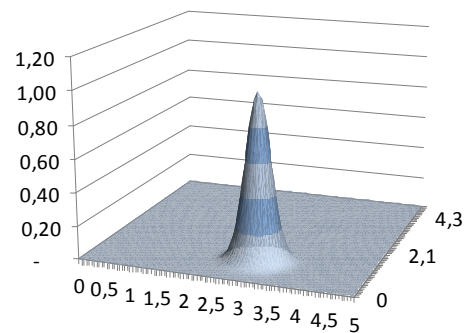
$$z = e^{-\frac{u}{2\sigma^2}}$$

Neurônios tipo tgh(u)



atuação ampla, global

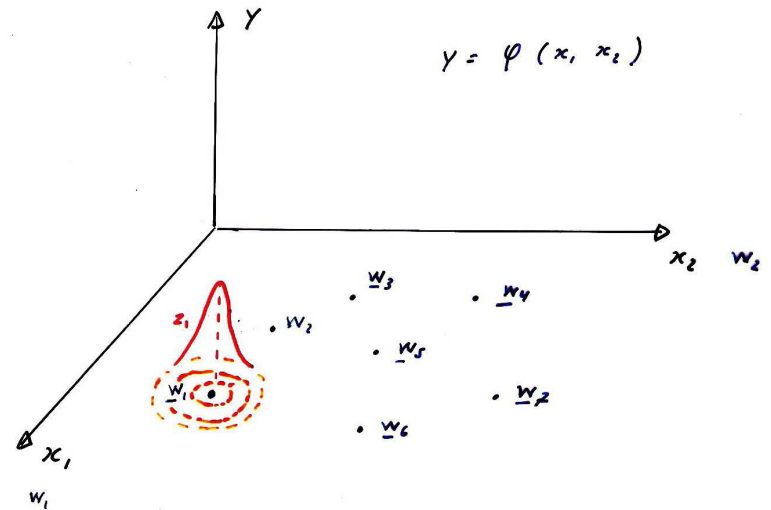
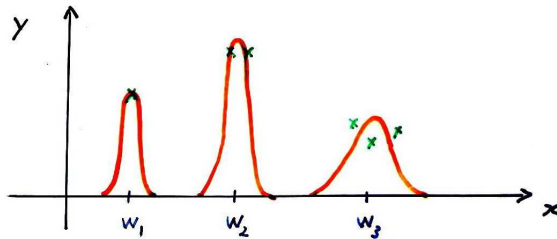
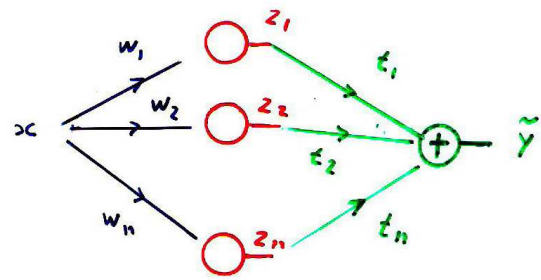
Neurônios de Base Radial



atuação local

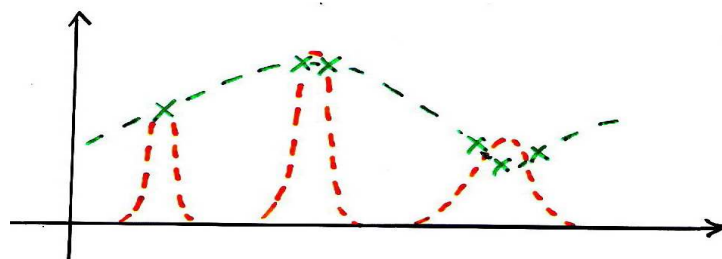
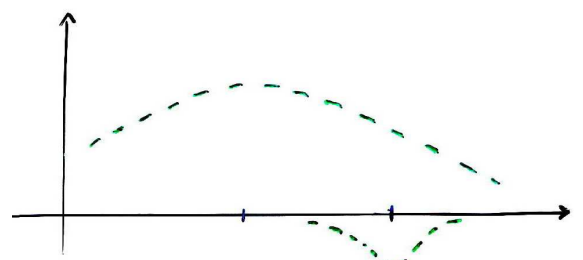
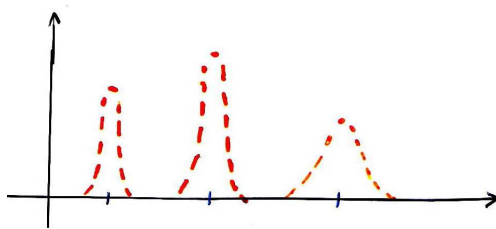
2 - Rede de Base Radial como Aproximador

atuação local



Composição da saída

local / global

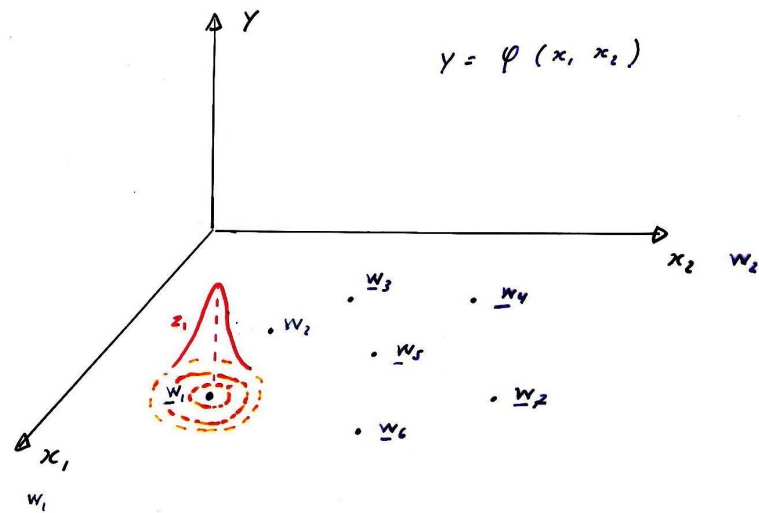
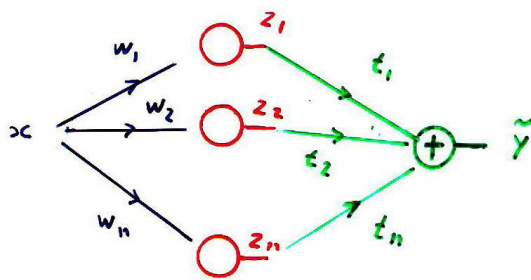


Função de base radial

$$f_i(\underline{w}_i, \underline{x}) \in [0,1] \approx \begin{cases} 1 & \text{para } \underline{x} \approx \underline{w}_i \\ 0 & \text{para } \underline{x} \neq \underline{w}_i \end{cases}$$

Rede de base radial

$$\tilde{y}(\underline{x}) \approx \begin{cases} t_i & \text{para } \underline{x} \approx \underline{w}_i \\ 0 & \text{para } \underline{x} \neq \underline{w}_i \end{cases}$$



Tipos de funções de base radial

$$d = |\underline{x} - \underline{w}|$$

gaussiana

$$z = e^{-\frac{d^2}{2\sigma^2}}$$

modal

$$z = e^{-\frac{d}{\sigma}}$$

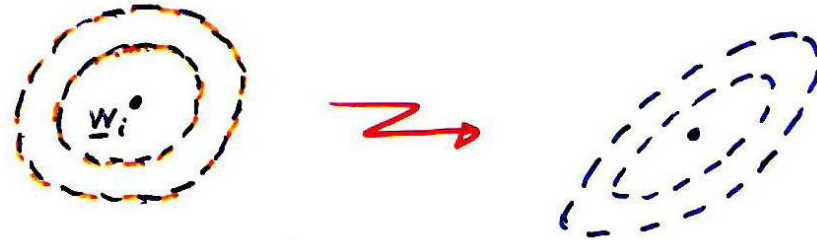
hiperbólica

$$z = \frac{1}{1 + \frac{d}{\sigma}}$$

etc...

Generalização da medida da distância

Curvas de nível – Lugar geométrico tal que $z = e^{-u}$ é constante



$$u = (\underline{x} - \underline{w})^t \underline{K} (\underline{x} - \underline{w})$$

$$\underline{K} = \frac{1}{2\sigma^2} \underline{I}$$

$\underline{K}$ é a matriz identidade
 LG é uma esfera centrada em $\underline{w}$
 Rede de base radial propriamente dita
 1 parâmetro à ajustar (σ)

$$\underline{K} = \text{diag}[k_{ii}]$$

$\underline{K}$ é uma matriz diagonal
 LG é um elipsóide com eixos alinhados com os do sistema (uma esfera na distância de Mahalanobis)
 * m parâmetros à ajustar

$$\underline{K} = [k_{ij}]$$

$\underline{K}$ é uma matriz triangular superior
 LG é um elipsóide com eixos em quaisquer direções
 * $m(m+1)/2$ parâmetros à ajustar !!!

* Obs: $m = \dim \underline{x}$

Usar

$$\underline{K} = \frac{1}{2\sigma^2} \underline{I}$$

$\underline{K}$ é a matriz identidade

LG é uma esfera centrada em $\underline{w}$
1 parâmetro à ajustar (σ)

ou

$$\underline{K} = \text{diag}[k_{ii}]$$

$\underline{K}$ é uma matriz diagonal

LG é um elipsóide com eixos alinhados com os do sistema (uma esfera na distância de Mahalanobis)

* m parâmetros à ajustar

Considerar usar PCA (componentes principais) como pré-processamento,

tornando as componentes ortogonais (e as elipses paralelas aos eixos) e normalizando (ou não) os desvios padrão de cada componente para 1

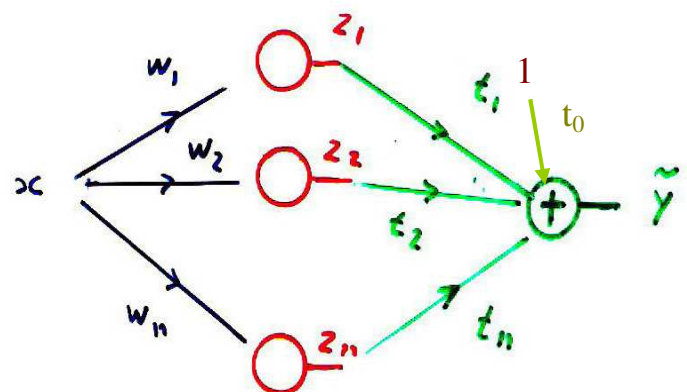
3 - Treinamento da RBF via backpropagation

Equações básicas da rede (signal feedforward)

$$\tilde{y} = \sum_{i=0}^n t_i z_i \quad z_0 = 1$$

$$z_i = e^{-\frac{d_i^2}{2\sigma_i^2}}$$

$$d_i^2 = \sum_{j=1}^m (w_{ij} - x_j)^2$$



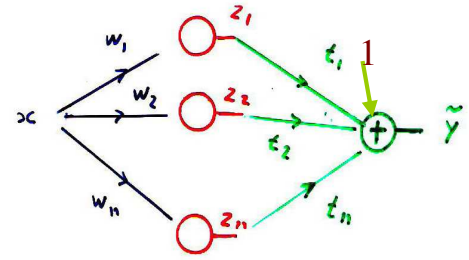
n - # neurônios na camada intermediária

m - dim $\underline{x}$

Função objetivo a minimizar

$$F = E \quad \varepsilon^2 \quad \varepsilon = y - \tilde{y}$$

$\forall_{\text{pares}}$

**Acréscimo a aplicar no parâmetro p (genérico)**

$$\Delta p = -\alpha \quad E \quad \frac{d\varepsilon^2}{dp}$$

$\forall_{\text{pares}}$

usando a regra da cadeia nas equações básicas da rede**Equações básicas da rede**

$$\tilde{y} = \sum_{i=1}^n t_i z_i \quad z_i = e^{-\frac{d_i^2}{2\sigma_i^2}}$$

$$d_i^2 = \sum_{j=1}^m (w_{ij} - x_j)^2$$

$$\frac{d\varepsilon^2}{dt_i} = \frac{d\varepsilon^2}{d\tilde{y}} \frac{d\tilde{y}}{dt_i} = (-2\varepsilon)(z_i)$$

$$\frac{d\varepsilon^2}{d\sigma_i} = \frac{d\varepsilon^2}{d\tilde{y}} \frac{d\tilde{y}}{dz_i} \frac{dz_i}{d\sigma_i} = (-2\varepsilon)(t_i) \left(\frac{d_i^2 z_i}{\sigma_i^3} \right)$$

$$\frac{d\varepsilon^2}{dw_{ij}} = \frac{d\varepsilon^2}{d\tilde{y}} \frac{d\tilde{y}}{dz_i} \frac{dz_i}{dd_i^2} \frac{dd_i^2}{dw_{ij}} = (-2\varepsilon)(t_i) \left(-\frac{z_i}{2\sigma_i^2} \right) [2(w_{ij} - x_j)]$$

Valores iniciais dos parâmetros:**vetores sinapse $\underline{w}$:**

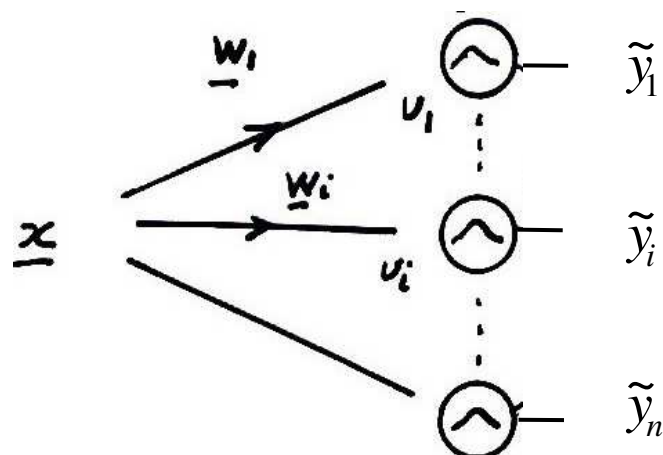
Melhor opção: clusterizar as entradas e escolher os centros dos clusters como valores iniciais dos vetores sinapse.

Mais simples: escolher as primeiras entradas que não estejam muito próximas entre si como valores iniciais para os vetores sinapse.

desvios σ_i :

Se foi feita a clusterização escolher σ como $\sim 25\%$ do diâmetro do cluster ou

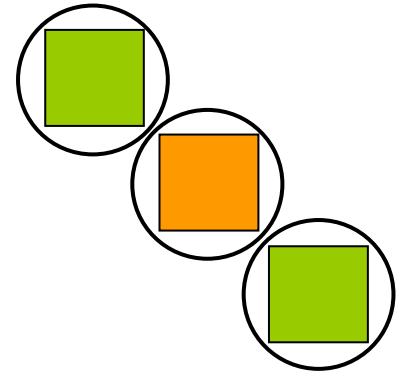
Se não, escolher como $\sim 50\%$ da distância média entre os centros iniciais escolhidos.

4 - Rede de Base Radial como Classificador**Redes com 1 camada****neurônios tipo RBF**

Interpretação da saída da rede como variável lógica

$$y \in [0,1] \quad \tilde{y} = e^{-u} \in (0,1]$$

$$\tilde{y}_{\text{lógico}} = \begin{cases} 1 & \text{se } \tilde{y} \geq 0.5 \\ 0 & \text{se } \tilde{y} < 0.5 \end{cases}$$



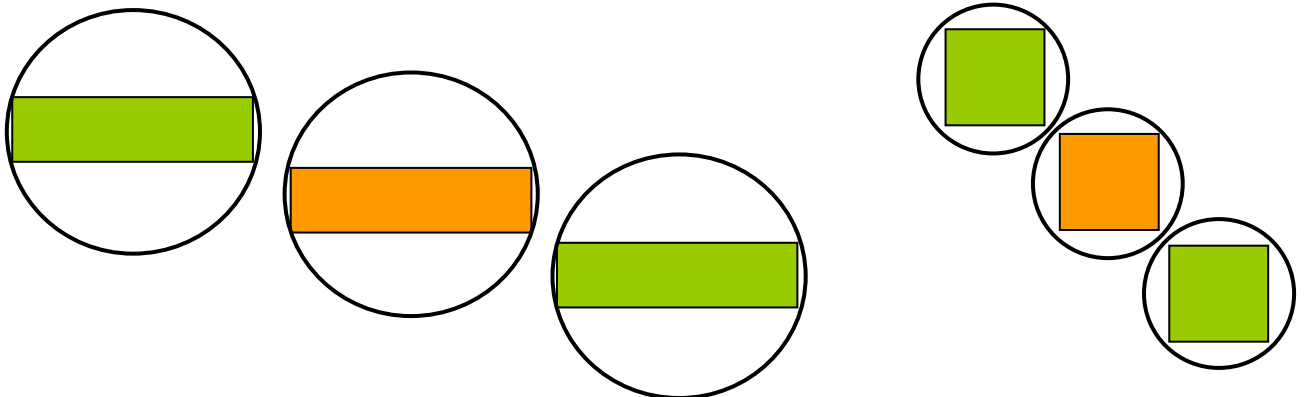
Separadores: esferas

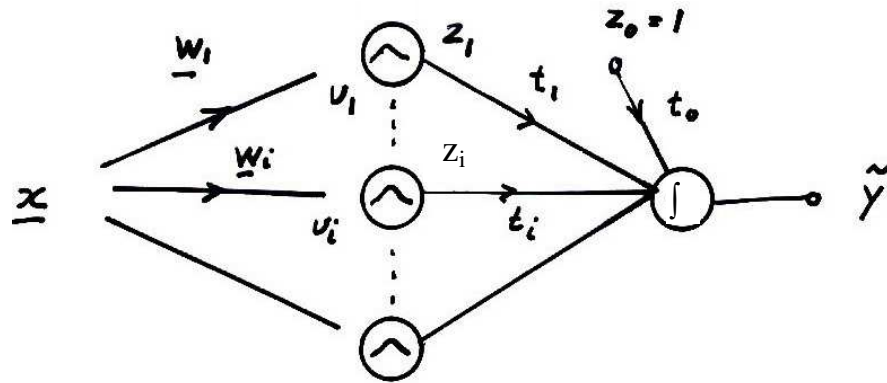
separa classes “esfericamente separáveis”

Treinamento: Backpropagation

Considerar usar PCA (componentes principais) como pré processamento,

normalizando os desvios padrão de cada componente para igualar o diâmetro de cada classe em todas as dimensões, facilitando torná-las “esfericamente separáveis”.



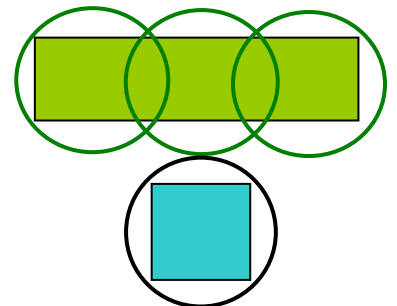
Redes com duas camadas:**Camada intermediária: neurônios RBF****Camada de saída: neurônios tgh(.) ou logísticos****Interpretação da saída da rede (tgh) como variável lógica**

$$y \in \{-1, +1\} \quad \tilde{y} = \tanh(u) \in (-1, +1)$$

$$\tilde{y}_{\text{lógico}} = \text{sign}(\tilde{y}) = \begin{cases} +1 & \text{se } \tilde{y} \geq 0 \\ -1 & \text{se } \tilde{y} < 0 \end{cases}$$

Separadores: composição de esferas

é um classificador universal

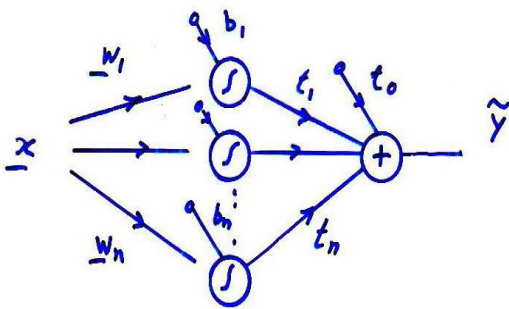
**Treinamento: Backpropagation**

4.1 - Capacidade de mapeamento das redes RBF:

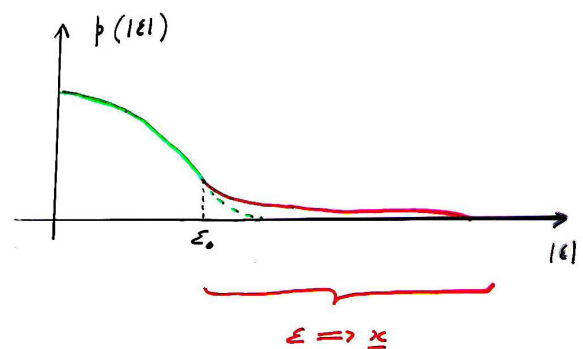
Existem teoremas análogos aos das redes com neurônios tipo tgh(.) para redes com neurônios tipo RBF na primeira camada e com neurônios lineares ou tipo tgh(.) na segunda camada, atuando respectivamente como aproximadores ou classificadores. Estes teoremas estabelecem que estas redes também são aproximadores universais.

5 - RBF em pós processamento

Correção de erros em redes BP ou RBF



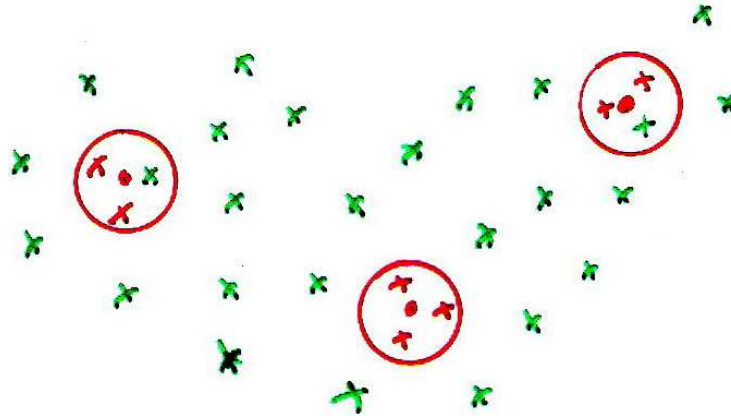
$$(x, y) \Rightarrow (x, y, \tilde{y}, \epsilon) \Rightarrow (\underline{x}, \epsilon)$$



$$|\epsilon| > \epsilon_0 \quad \underline{x} \in X_0$$

$$\text{Clusterizer} \quad \underline{x} \in X_0$$

Clusterizar todo $\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix}$ tal que $\varepsilon > \varepsilon_0$



Identificar toda a população das regiões de grandes erros, inclusive os pares com pequenos erros

(identificar e eliminar intrusos, se houver)

Rede aumentada

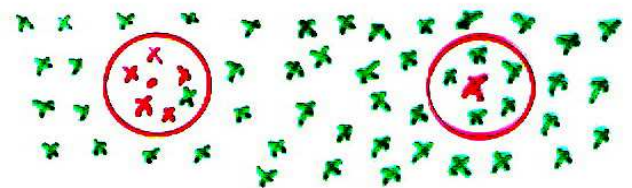
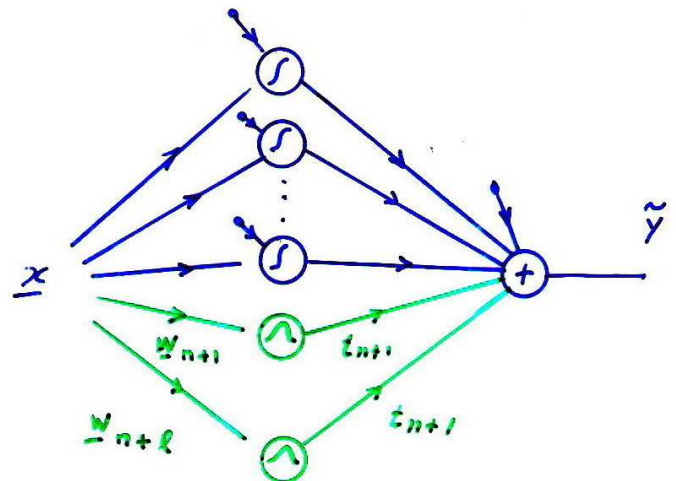
Valores iniciais dos neurônios RBF

$\underline{w}$ – centros das classes de erro

σ – 0,5 diâmetro das classes de erro

Treinamento por BP

Iniciar com os pares $(\underline{x}, \underline{y})$ que compõem as classes de erro elevado e seus vizinhos (toda a população das regiões de grandes erros)



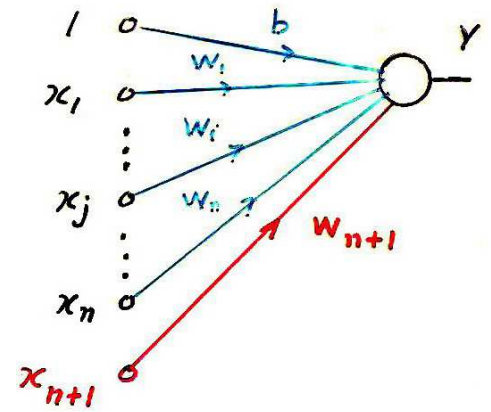
O processo pode ser aplicado em qualquer tipo de rede, “BP”, RBF ou outros.

6 - Rede alternativa à RBF

Perceptrons com entrada aumentada

$$x_{n+1} = \sum_{j=1}^n x_j^2$$

$$u = w_{n+1}x_{n+1} + \sum_{j=1}^n w_j x_j + b$$



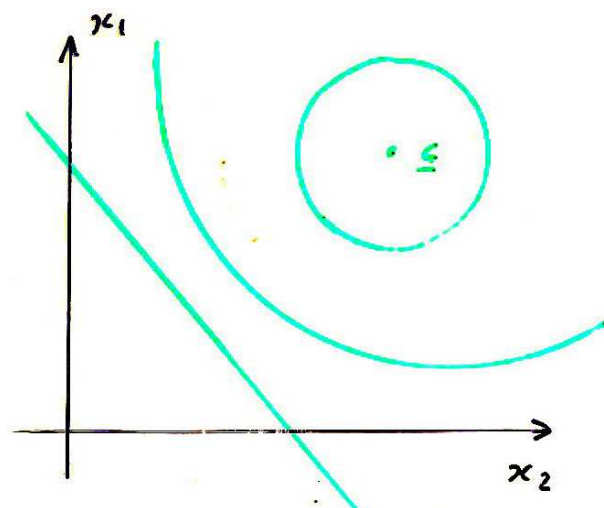
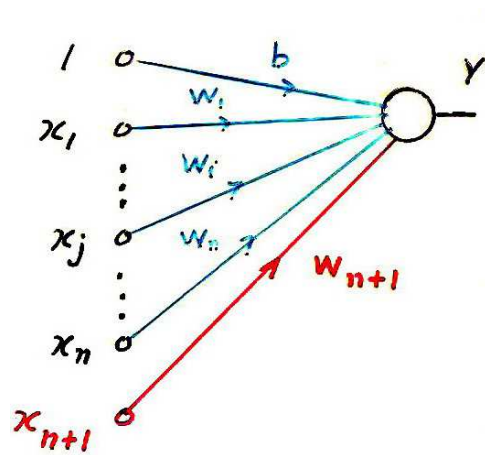
LG para $u = k = \text{constante}$

Esfera com centro com coordenadas

$$c_i = -\frac{w_i}{2w_{n+1}}$$

e raio

$$r = \frac{1}{2w_{n+1}} \left\{ \sum_{i=1}^n w_i^2 + w_{n+1}(k - b) \right\}^{1/2}$$



Obs:

1 - caso $w_{n+1} = 0$ a esfera degenera em um plano.

2 - caso $|\underline{x}| = 1$ para todo $\underline{x}$ (entradas com módulo normalizado) a entrada $\underline{x}_{n+1}$ é desnecessária. Como neste caso as entradas estão sobre uma hiper esfera de raio unitário, a intersecção do hiperplano separador com esta hiperesfera é outra hiperesfera. A normalização do módulo de $\underline{x}$ preservando a informação nele contida pode ser feita pela substituição de $\underline{x}$ por $\underline{x}'$, que tem uma componente a mais que $\underline{x}$.

$$x_i \longrightarrow x'_i = \frac{x_i}{M} \quad \forall i = 1, 2, \dots, n$$

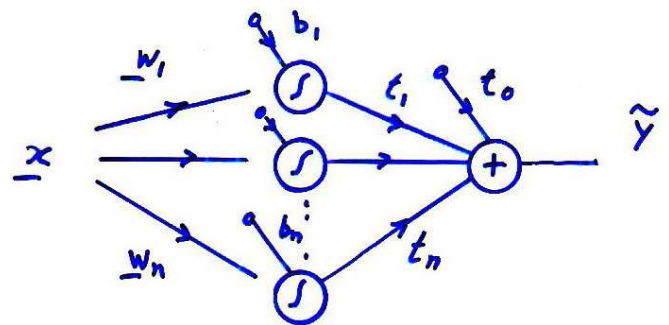
$$x'_{n+1} = \sqrt{1 - \sum_{i=1}^n x_i'^2} = \sqrt{1 - \frac{1}{M^2} \sum_{i=1}^n x_i^2}$$

onde $M = \text{constante} \geq \max |\underline{x}|^2$

Rede com perceptrons aumentados com 2 camadas**Rede como aproximador**

Camada intermediária – neurônios
tgh com entradas aumentadas

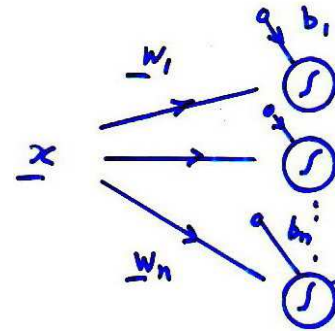
Camada de saída – neurônio linear



Rede como classificador

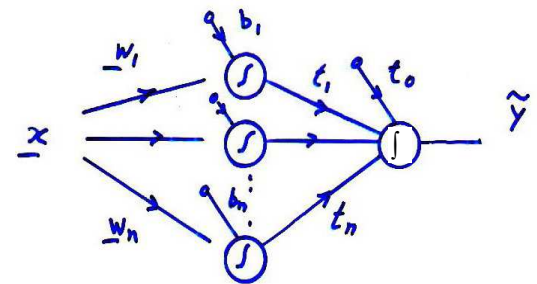
1 camada

- neurônios tgh com entradas aumentadas
- separadores esféricos



2 camadas – composição de esferas

- neurônios da camada intermediária tgh com entradas aumentadas
- neurônio da camada de saída tgh convencionais
- separador: composição de esferas



Treinamento:

Backpropagation convencional

Valores iniciais

Camada intermediária – similar aos da RBF

Camada de saída – aleatórios,
similar a rede com neurônios tgh convencionais.