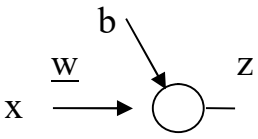
# Redes de Base Radial - RBF - Radial Basis Function

## 1 - Introdução

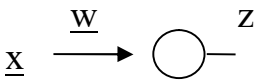
### Neurônios tipo tgh(u)




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

$$z = tgh(u)$$

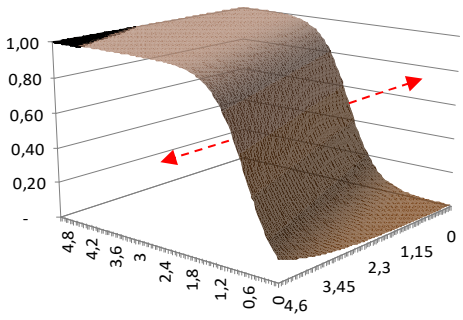
### Neurônios de Base Radial



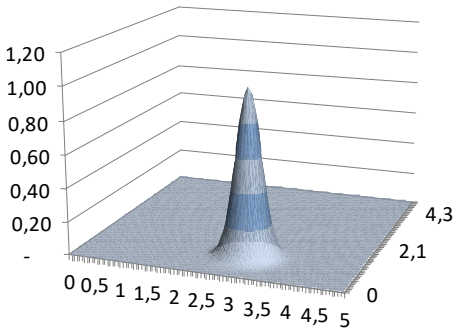

$$u = \left|\underline{x} - \underline{w}\right|^2$$

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

### Neurônios tipo tgh(u)




### Neurônios de Base Radial

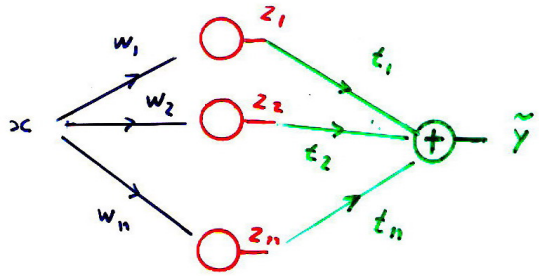


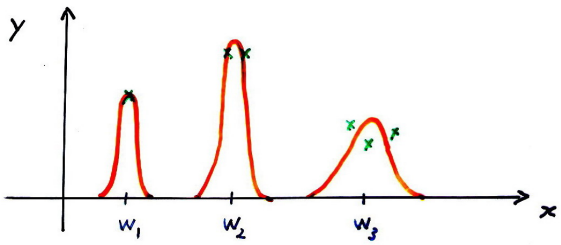

### atuação ampla, global

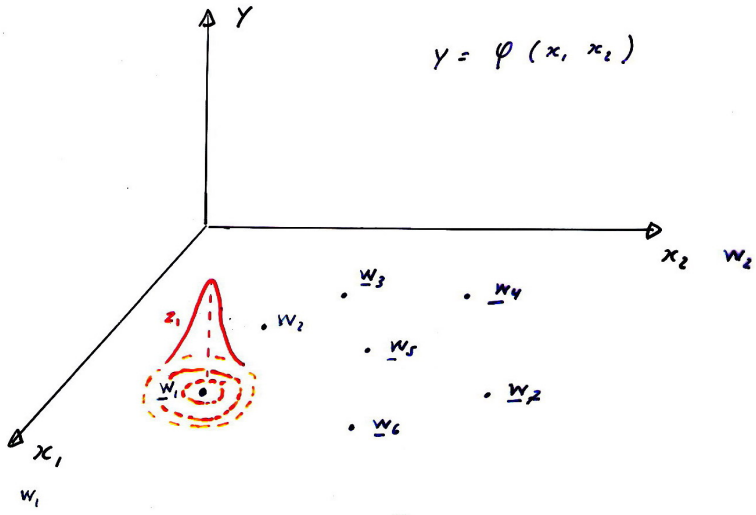
### atuação local

## 2 - Rede de Base Radial como Aproximador

### atuação local

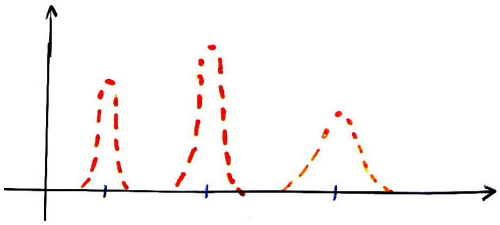


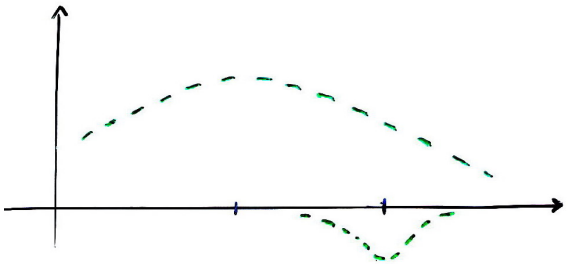



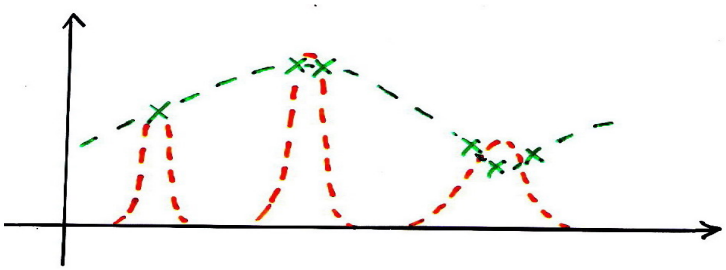




### Composição da saída

### local / global





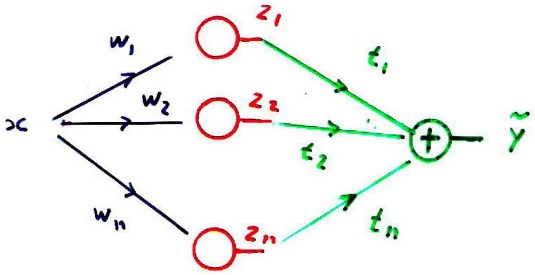


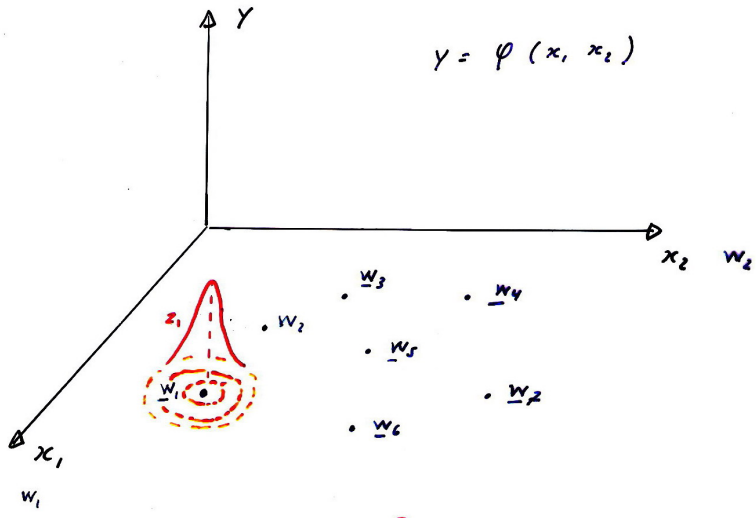
## Função de base radial

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

## Rede de base radial

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




## Tipos de funções de base radial

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

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

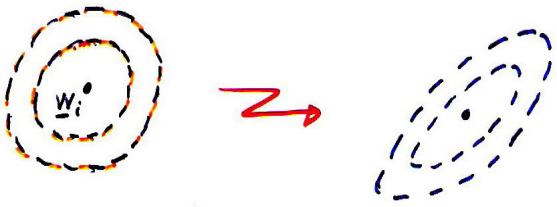
**modal** $$z = e^{-\frac{d}{\sigma}}$$

**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**




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

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

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

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

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

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

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 x

**Usar**

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

K é a matriz identidade
LG é uma esfera centrada em w
1 parâmetro à ajustar (σ)

ou

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

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 os elipsoides paralelos aos eixos) e normalizando (ou não) os desvios padrão de cada componente para 1 (e transformando os elipsoides em esferas)

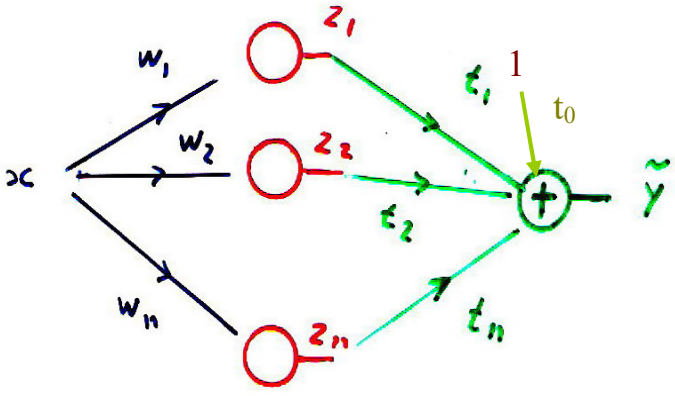
## 3 - Treinamento da RBF via backpropagation

### Equações básicas da rede (signal feedforward)

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

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

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

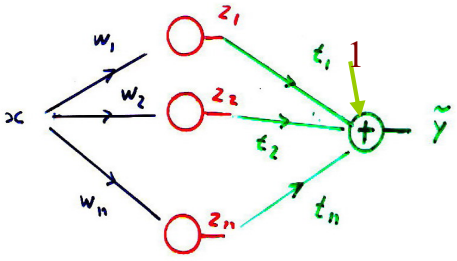



n - # neurônios na camada intermediária

m – dim x

**Função objetivo a minimizar**

$$F = \underset{\forall pares}{E}\ \varepsilon^2 \qquad \varepsilon = y - \tilde{y}$$




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

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

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

**Equações básicas da rede**

$$\tilde{y} = \sum_{i=0}^{n} t_i z_i \qquad z_i = e^{-\frac{d_i^2}{2\sigma_i^2}}$$

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

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

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

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

**Valores iniciais dos parâmetros:**

## vetores sinapse 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 $\sigma_i$:

Se foi feita a clusterização escolher σ como ~ 25 % do diâmetro do cluster ou

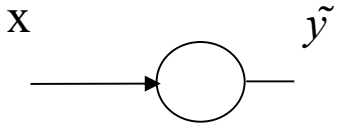
Se não, escolher como ~ 50 % da distância média entre os centros iniciais escolhidos.

## 4 – Saídas lógicas

### neurônio tipo RBF na camada de saída

### *Convenção para as saídas*

$$y \in \{0,1\}$$

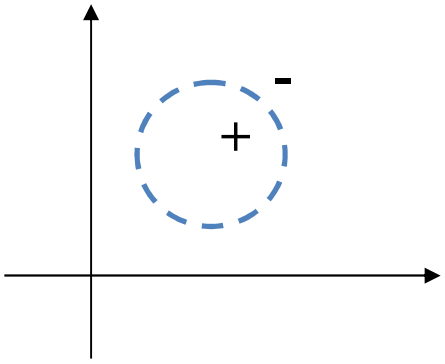



$$\tilde{y} = e^{-\frac{d^2}{2\sigma^2}} \in (0,1)$$

### Interpretação da saída da rede como variável lógica

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

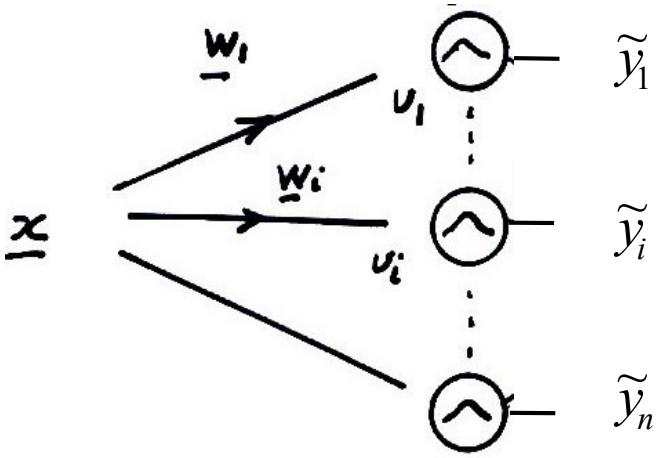
$$\tilde{y}_{logico} = sign(\tilde{y} - 0.5) = \begin{cases} -1 \\ +1 \end{cases}$$



## 4.1 - Rede de Base Radial como Classificador

### Redes com 1 camada

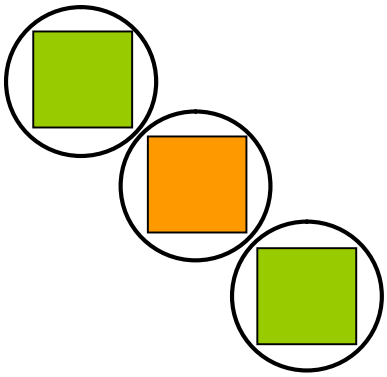
### neurônios tipo RBF




### Interpretação da saída da rede como variável lógica (neurônio RBF)

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

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

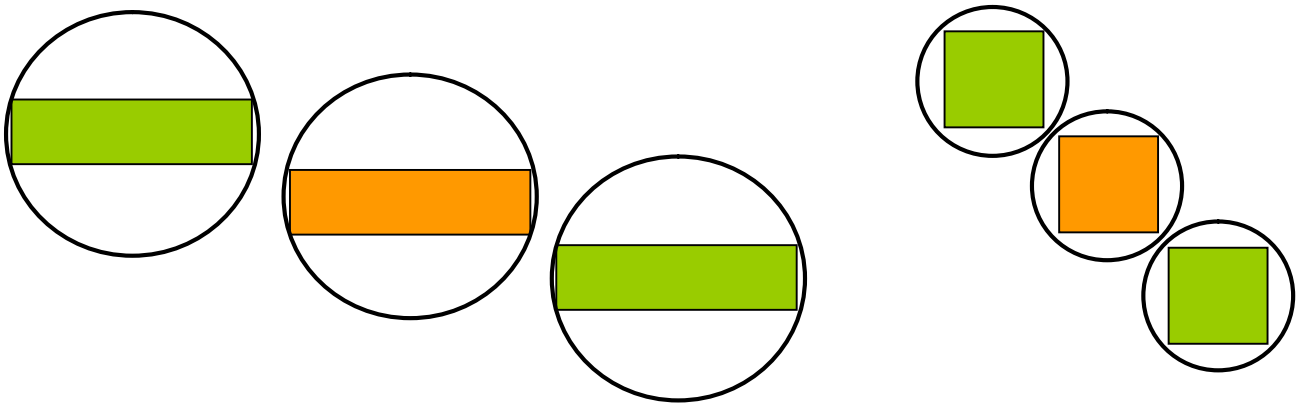


### Separadores: esferas

**separa classes "esfericamente separáveis"**

### Treinamento: Backpropagation

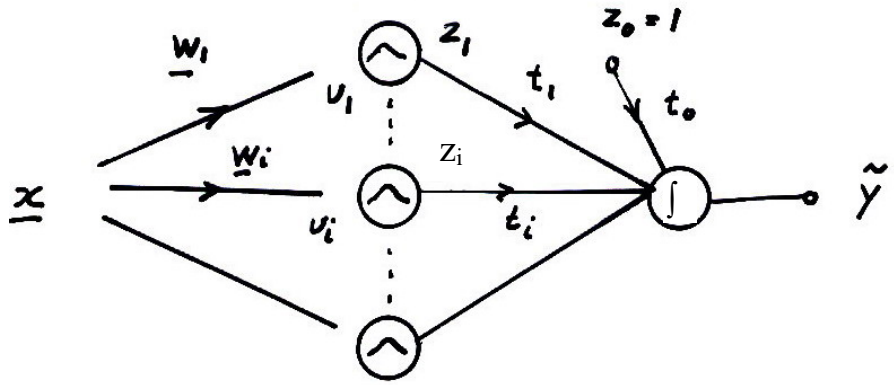
**Considerar usar PCA** (componentes principais) como pré processamento, normalizando os desvios padrão de cada componente (por classe) para igualar o diâmetro de cada classe em todas as dimensões, facilitando torná-las "esfericamente separáveis".



**Redes com duas camadas:**

**Camada intermediária: neurônios RBF**

**Camada de saída: neurônios RBF, tgh(.) ou logísticos**

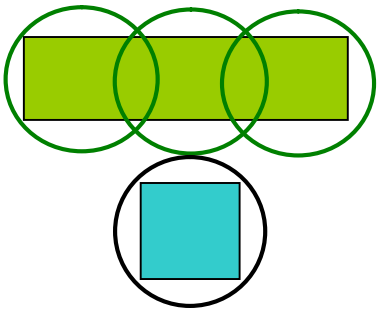


## Interpretação da saída da rede (tgh) como variável lógica

$$y \in \{-1,+1\} \qquad \tilde{y} = tgh(u) \in (-1,+1)$$

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

## Separadores: composição de esferas



**é um classificador universal**

## Treinamento: Backpropagation

## 5 - 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.**

# 6 - Ruído em Classificadores RBF

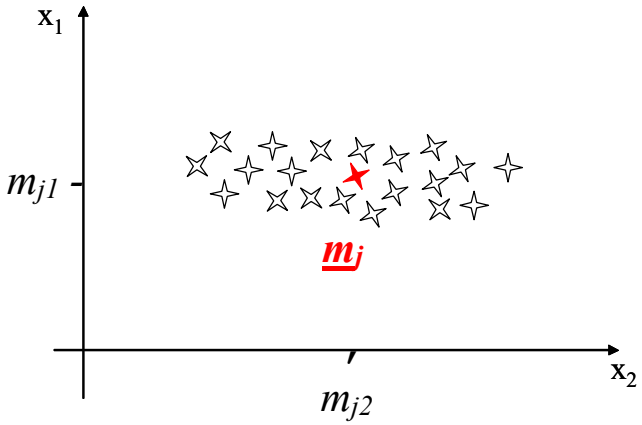
## 6.1 - Hipóteses:

1 - Os elementos **x** de uma classe $C_j$ são constituídos pelo padrão $\mathbf{m_j}$ da classe, ao qual foi adicionado um ruído **r**.

$$\vec{m}_j = \underset{\forall \vec{x} \in C_j}{E}\, \vec{x} = \frac{1}{n_j} \sum_{\forall \vec{x} \in C_j} \vec{x}$$

$$\vec{m}_j = \left[ m_{j1} \ldots m_{jk} \ldots \right]^t$$

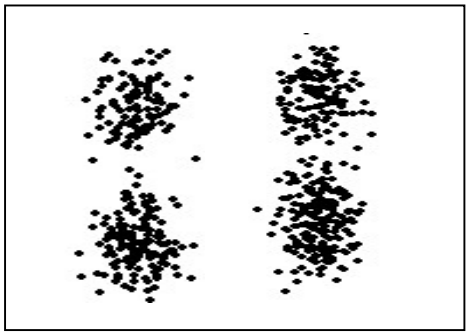
$$\vec{x} \in C_j \quad \therefore \quad \vec{x} = \vec{m}_j + \vec{r}$$



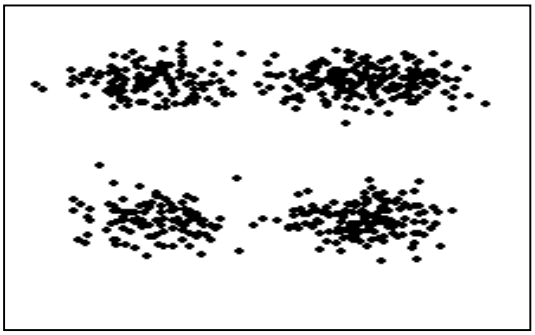

2 - O ruído **r** é estocástico com média zero e distribuição aproximadamente gaussiana.

3 - As características do ruído independem da classe (ou não).

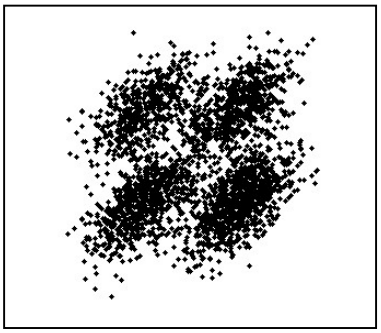
4 - Uma classe pode ser constituída por subclasses, às quais se aplicam as hipóteses acima.



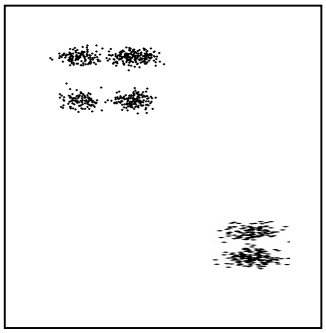
Ruído não correlato, $\sigma_1 \cong \sigma_2$



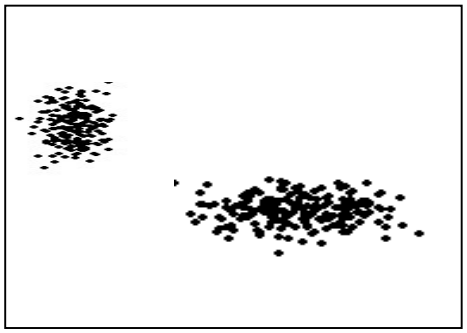
Ruído não correlato, $\sigma_1 \neq \sigma_2$



Ruído correlato, $x_1 = f(x_2)$



Classes compostas



Hipótese 3 não atendida

## 6.2 - Caracterização do ruído

**As classes de cada entrada (elemento) x são conhecidas. Neste caso seu padrão $\vec{m}_j$ também é conhecido**

$$\vec{m}_j = \underset{\forall \vec{x}_i \in C_j}{E}\ \vec{x}_i = \frac{1}{n_j} \sum_{\forall \vec{x}_i \in C_j} \vec{x}_i$$

**Pares entrada-saída** $\left[\vec{x}, C(\vec{x})\right]$

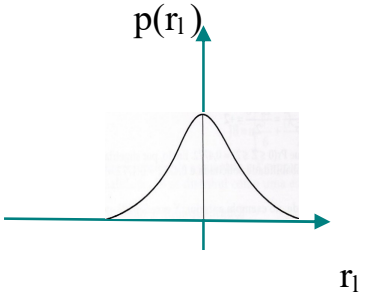
A caracterização do ruído, em cada dimensão *l*, $r_l$ , pode ser realizada a partir da extração de $\vec{r}$ das entradas pertencentes à classe

$$\forall\, \vec{x} \in C_j \qquad \vec{x} = \vec{m}_j + \vec{r} \qquad \therefore \qquad \vec{r} = \vec{x} - \vec{m}_j \quad e \quad r_l = x_l - m_{jl}$$

$$\forall \vec{x} \in C_j \qquad \vec{x} = \vec{m}_j + \vec{r} \qquad \therefore \qquad \vec{r} = \vec{x} - \vec{m}_j \qquad e \qquad r_l = x_l - m_{jl}$$

A análise do conjunto das variáveis $r_l$ permite conhecer sua distribuição, seu valor eficaz $\sigma$, sua correlação com o ruído nas demais dimensões $r_k \mid k \neq l$, etc.

Isto permitirá descorrelacionar os ruídos das diferentes dimensões, normalizar o valor eficaz por dimensão, etc., obtendo um conjunto de dados muito mais simples de classificar usando RBF, como será visto mais tarde.



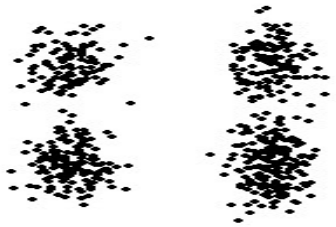

## 6.3 – Ruído - caso ideal

**Classes com ruídos por componente com mesma variância e não correlatos - Classes esféricas**

***Considerando o caso em que que:***

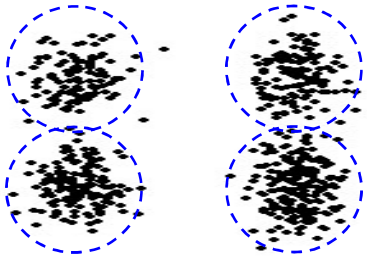
**– As classes foram geradas a partir de padrões adicionados de ruído aproximadamente gaussiano com média nula.**

**– que o valor rms (desvio padrão) do ruído de cada componente $x_j$ é σ , independente da componente e da classe.**



**então**

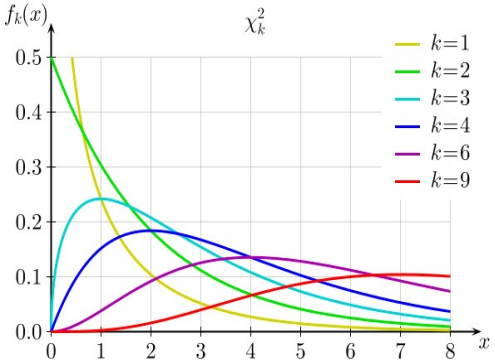
**o separador ideal para as classes são esferas, que são também o separador natural por RBF.**



**Se o ruído em cada dimensão é gaussiano com σ = 1 então as classes tem raio tal que $r_0^2$ tem distribuição chi-quadrado.**

**Em consequência o valor de $r_0^2$ deve ser escolhido baseado nesta distribuição**

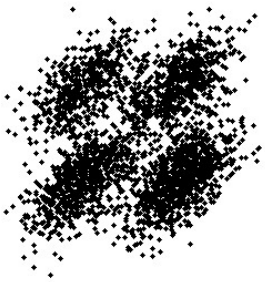
**Tentaremos chegar próximo a esta condição através do pré-processamento**




## 7 - Pré-processamento da entrada para classificadores RBF

### 7.1 - Branqueamento do Ruído – gerando classes esféricas *classes sphereing*

**Se os ruídos das componentes forem correlatos os domínios das classes serão elipsoides e estarão contidos por esferas maiores, que incluem espaços da não classe.**



**O branqueamento do ruído leva a classes esféricas, que permitirá o uso de um classificador muito mais simples. O branqueamento do ruído dos dados é obtido em duas etapas.**

**Inicialmente a descorrelação dos ruídos $r_i$ deve ser obtida representando os dados x (padrão da classe + ruído) em uma base B do espaço das componentes principais (PCA) do ruído r de cada dado de entrada (PCA dos ruídos, e não dos sinais!). Calcular as PCA de r e a matriz de mapeamento B do ruído r em sua PCA p**

$$\vec{x} \in C_j \qquad \vec{x} = \vec{m}_j + \vec{r} \qquad \therefore \qquad \vec{r} = \vec{x} - \vec{m}_j$$

**Utilizando B determinamos o mapeamento de cada ruído r em sua PCA p**

$$\vec{p} = \underline{B}^t \vec{r} \qquad \text{onde} \qquad [B] = [\vec{b}_1 \ \vec{b}_2 \ \vec{b}_3 \ ... \vec{b}_m] \qquad e \qquad p_i = \vec{b}_i^t \vec{r}$$

**e a variância de cada componente do ruído $p_i$**

$$\sigma_i^2 = E[p_i]^2 = E[\vec{b}_i^t \vec{r}]^2$$

**A segunda etapa deve ser a equalização da potência do ruído nas diversas dimensões. A matriz de pesos $\underline{\sigma}^{-1}$ para normalização das potências dos ruídos será**

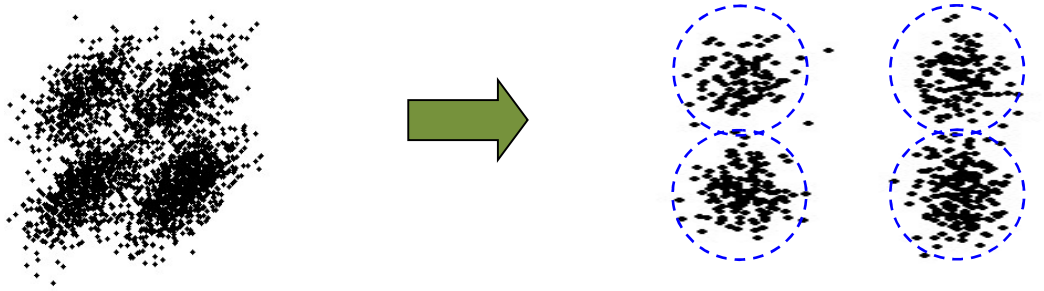
$$\underline{\sigma}^{-1} = diag\left(\frac{1}{\sigma_i}\right)$$

**Os novos dados à classificar serão**

$$\vec{z} = \underline{\sigma}^{-1} \underline{B}^t \vec{x} = \underline{\sigma}^{-1} \underline{B}^t \vec{m}_j + \underline{\sigma}^{-1} \underline{B}^t \vec{r} = \vec{m}_{j\,novo} + \vec{r}_{novo}$$

**As classes passam a ter um novo padrão $\vec{m}_{j\,novo}$ adicionado de um ruído branco $\vec{r}_{novo}$ com desvio padrão unitário.**

$$\vec{m}_{j\,novo} = \underline{\sigma}^{-1} \underline{B}^t \vec{m}_j \qquad \vec{r}_{novo} = \underline{\sigma}^{-1} \underline{B}^t \vec{r} \qquad \sigma(\vec{r}_{novo}) = \vec{1} = [1\ 1\ 1 ... \ 1]^t$$



## 7.2 Algumas observações:

- **A multiplicação por $B^t$ representa uma rotação e por isto não altera a posição relativa das classes. Mas a multiplicação por $\underline{\sigma}^{-1}$ altera.**

- **Como $\underline{B}$ é ortonormal e $\underline{\sigma}^{-1}$ é diagonal o conhecimento da matriz branqueadora $\underline{N}^t$ permite determinar B e $\underline{\sigma}^{-1}$**

$$[N]^t = [\vec{n}_1^t\ \vec{n}_2^t\ \vec{n}_3^t \ldots \vec{n}_m^t] = \underline{\sigma}^{-1}[\vec{b}_1^t\ \vec{b}_2^t\ \vec{b}_3^t \ldots \vec{b}_m^t]$$

$$\vec{n}_i^t \vec{n}_i = \left(\sigma_i^{-1}\vec{b}_1^t\right)\left(\sigma_i^{-1}\vec{b}_1\right) = \sigma_i^{-2} \qquad e \qquad \vec{b}_i = \sigma_i \vec{n}_i$$

- **O uso de PCA sugere a eliminação de componentes pouco relevantes, i.e., com pequenos**

$$\sigma_i^2 = E[p_i]^2 = E[\vec{b}_i^t \vec{r}]^2$$

**e a consequente redução da dimensionalidade e simplificação do classificador. Entretanto, a eliminação de uma componente principal para $\vec{r}$ somente pode ser feita se for pouco relevante para $\vec{x}$, isto é, se**

$\sigma^2(z_i) = E[\frac{1}{\sigma_i}\vec{b}_i^t\vec{x}]^2$ **for muito pequeno comparado com seus pares.**

**A relevância para $\vec{x}$ é que determina a eliminação ou não da componente !**

- **A multiplicação por P realiza um escalamento diferente por componente e pode alterar a classificação, principalmente se houver grande dispersão entre os $\sigma_i$**

- **Caso os ruídos sejam não correlatos, a primeira etapa é obviamente desnecessária**

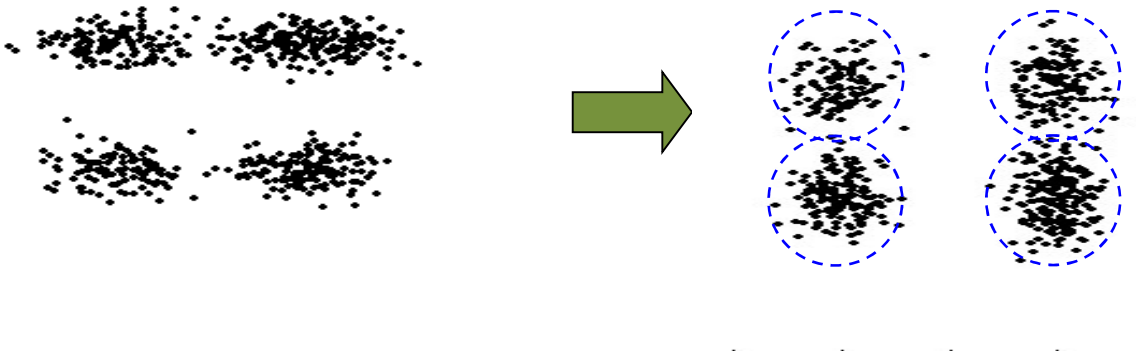
$$\underline{B} = \underline{I}$$

**E a dispersão em cada dimensão i é dada por**

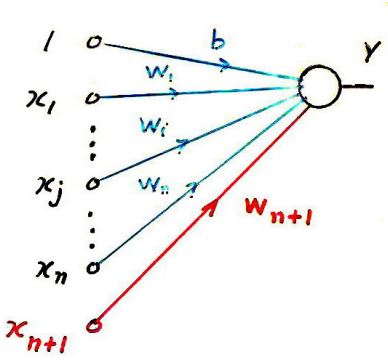
$$\sigma_i^2 = E[r_i]^2 \qquad \text{e} \qquad \underline{\sigma}^{-1} = diag\left(\frac{1}{\sigma_i}\right)$$

**Os novos dados à classificar serão**

$$\vec{z} = \underline{\sigma}^{-1}\vec{x} = \underline{\sigma}^{-1}\vec{m}_j + \underline{\sigma}^{-1}\vec{r} = \vec{m}_{j\,novo} + \vec{r}_{novo}$$



## 8 - Rede alternativa à RBF (parabólica)




### Perceptrons com entrada aumentada

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

### Separador u = 0

$$u = \sum_{j=0}^{n+1} w_j x_j = 0 \qquad w_{n+1} \text{ arbitrário}$$

$$u = w_{n+1} \sum_{j=1}^{n} x_j^2 + \sum_{j=1}^{n} w_j x_j + w_0 = 0 \qquad eq(1)$$

### LG da esfera com centro em $\underline{c} = c_i]$ e raio r

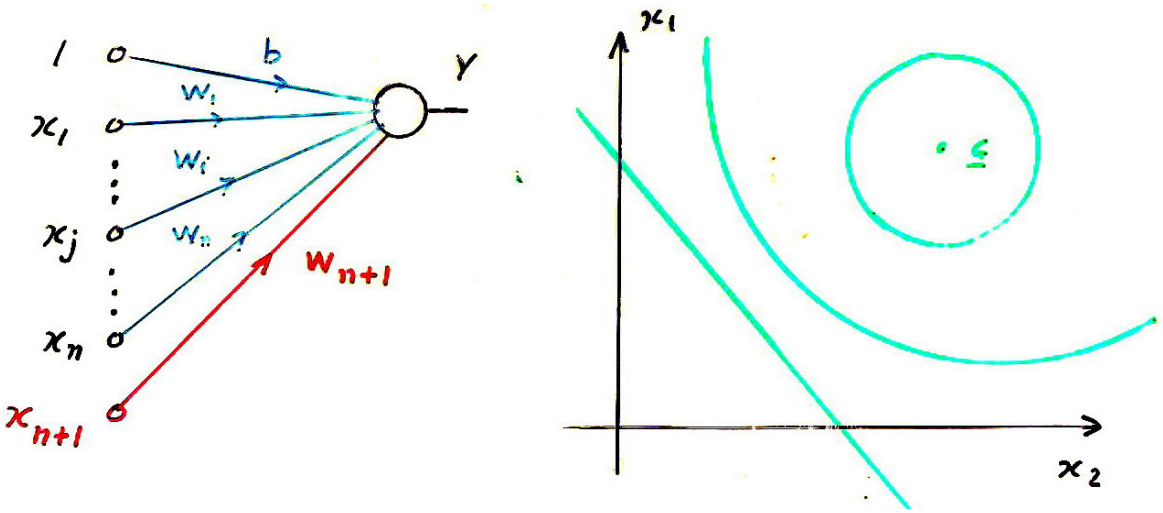
$$u = w_{n+1} \left\{ \sum_{j=1}^{n} \left( x_j - c_j \right)^2 - r^2 \right\} = 0 \qquad w_{n+1} \text{ arbitrário}$$

$$u = w_{n+1} \sum_{j=1}^{n} x_j^2 + w_{n+1} \sum_{j=1}^{n} \left( -2 c_j x_j \right) + w_{n+1} \left( \sum_{j=1}^{n} c_j^2 - r^2 \right) = 0 \qquad eq(2)$$

### Comparando termo à termo as eq (1) e (2) o separador é uma esfera com

**centro com coordenadas** $c_i = -\dfrac{w_i}{2 w_{n+1}}$

**e raio** $r = \left\{ \sum_{i=1}^{n} c_i^2 - \dfrac{w_0}{w_{n+1}} \right\}^{1/2} = \dfrac{1}{2 w_{n+1}} \left\{ \sum_{i=1}^{n} w_i^2 - 4 w_{n+1} w_0 \right\}^{1/2}$




$w_{n+1}$ **é arbitrário, mas sua polaridade determina a polaridade de** $u$ **no interior/exterior da esfera.**

**Para pontos no interior da esfera** $sign(u) = -sign(w_{n+1})$

**Para pontos no exterior da esfera** $sign(u) = sign(w_{n+1})$

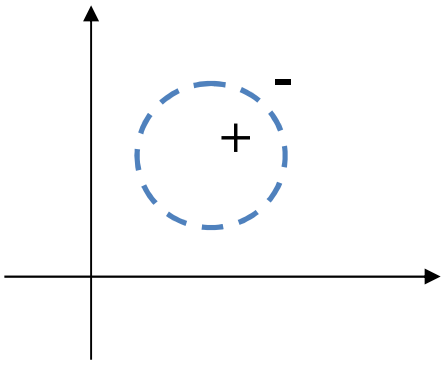
**Assim, por simplicidade para** $u > 0$ **no interior da esfera arbitramos inicialmente**

$$w_{n+1} = -1$$

$$u = -\sum_{j=1}^{n} x_j^2 + \sum_{j=1}^{n} w_j x_j + w_0 = 0$$

$$c_i = \frac{w_i}{2}$$

$$r = \left\{ \sum_{i=1}^{n} c_i^2 + w_0 \right\}^{1/2} = \left\{ \frac{1}{4} \sum_{i=1}^{n} w_i^2 + w_0 \right\}^{1/2}$$



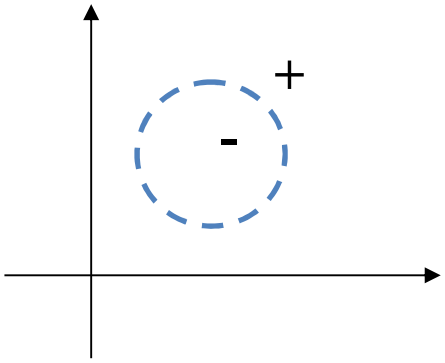
**E para $u<0$ no interior da esfera arbitramos inicialmente**

$$w_{n+1} = +1$$

$$u = \sum_{j=1}^{n} x_j^2 + \sum_{j=1}^{n} w_j x_j + w_0 = 0$$

$$c_i = -\frac{w_i}{2}$$

$$r = \left\{ \sum_{i=1}^{n} c_i^2 - w_0 \right\}^{1/2} = \left\{ \frac{1}{4} \sum_{i=1}^{n} w_i^2 - w_0 \right\}^{1/2}$$




**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 uma dimensão menor. 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 $\underline{x}$.

$$x_i \longrightarrow x_i' = \frac{x_i}{M} \qquad \forall\ i = 1,2,\ldots,n$$

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

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

3 – Como na RBF o processo pode ser extendido para elipsoides paralelos aos eixos (K diagonal) ou genéricos (K triangular) comparando o LG destes elipsoides

$$u = w_{n+1}\left\{(\underline{x}-\underline{c})^t\,\underline{K}\,(\underline{x}-\underline{c})-r\right\}=0 \qquad w_{n+1} \text{ arbitrário}$$

com o separador genérico de 2ª ordem

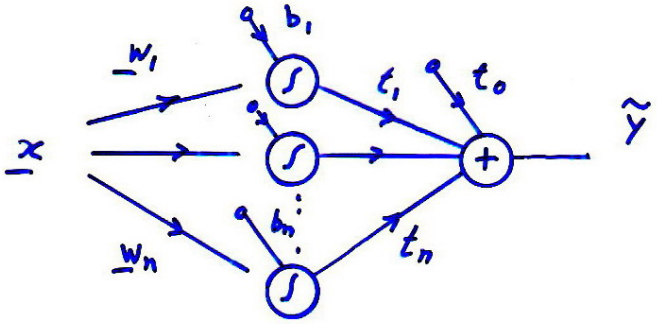
$$u = w_{n+1}\sum_{j=1}^{n} x_j^2 + \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} w_{ij}x_i\,x_j + \sum_{i=1}^{n} w_i x_i + w_0$$

Com o custo do aumento do número de sinapses à treinar

**8.1 Rede com perceptrons aumentados**

**Rede como aproximador com 2 camadas**

Camada intermediária – neurônios tgh ou logística com entradas aumentadas



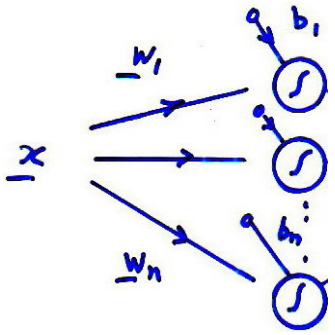

$$u = \sum_{j=0}^{n+1} w_j x_j \qquad x_{n+1} = \sum_{j=1}^{n} x_j^2$$

$$v = tgh(u)$$

Camada de saída – neurônio linear

## Rede como classificador

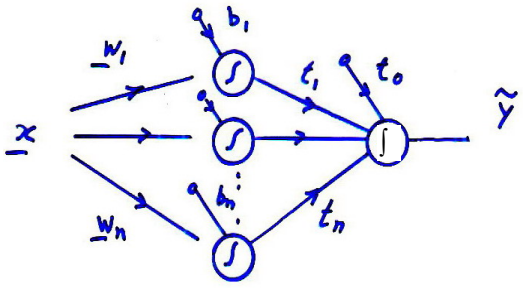
### 1 camada



- neurônios tgh ou logística com entradas aumentadas

– separadores esféricos

### 2 camadas – composição de esferas



- neurônios da camada intermediária tgh ou logística com entradas aumentadas

- neurônio da camada de saída tgh ou logística convencionais

- separador: composição de esferas

## Treinamento:

Backpropagation convencional

Valores iniciais

Camada intermediária – similar aos da RBF

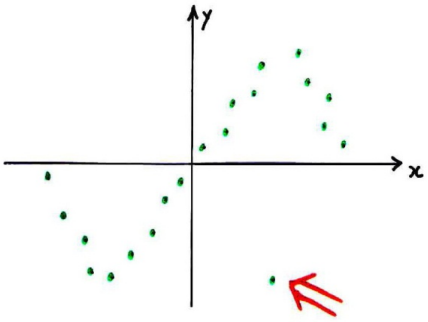
Camada de saída – aleatórios,
similar a rede com neurônios tgh convencionais.

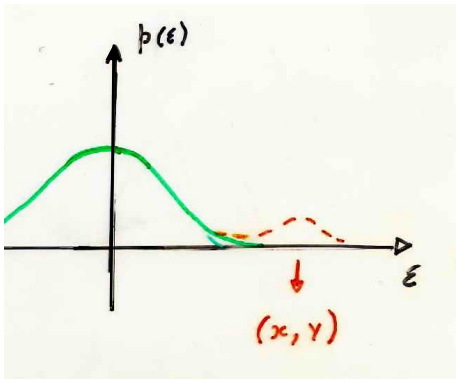
# 9 – RBF para correção de erros localizados

## Histograma dos erros (mandatório)

$$(\underline{x}, \underline{y}) \longrightarrow \underline{\tilde{y}} \longrightarrow \varepsilon = |\underline{y} - \underline{\tilde{y}}|$$

ε > valor esperado ⟶ **Provável anomalia no par (x, y)**



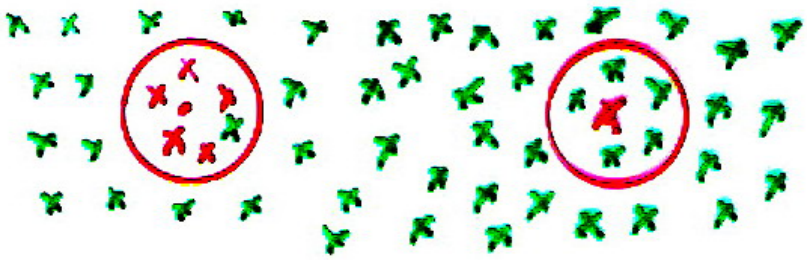




## Examinar conjuntos de treinamento, teste e validação

| Anomalias possíveis | Correções possíveis |
|---|---|
| Intrusos | Eliminar |
| Regiões de baixa população | Balancear populações |
| Mapeamento localmente complexos | Melhorar a capacidade de mapeamento local |

**Localização (clusterização) das anomalias**

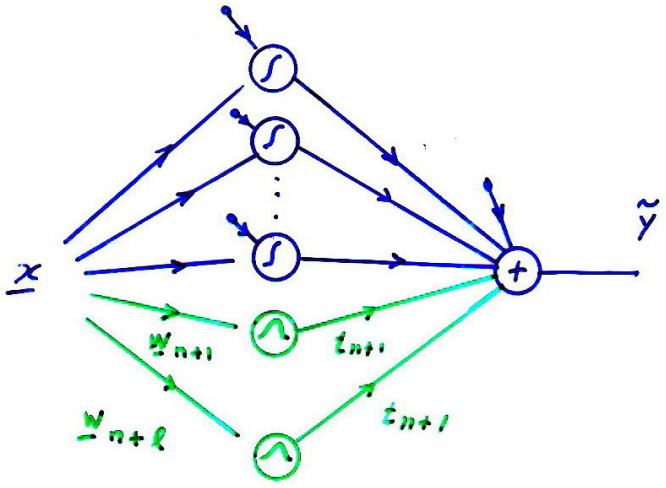
**Clusterizar todo** $\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix}$ **tal que** $\varepsilon > \varepsilon_0$



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

**(identificar e eliminar intrusos, se houver; balancear populações, se for o caso)**

## Aumentar a capacidade de processamento local - Rede aumentada




**Valores iniciais dos neurônois RBF**

**w ~ centros das classes de erro**

**σ ~ 0,5 diâmetro das classes de erro**

**Treinamento por BP**

**Iniciar com os pares ( x, 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.**