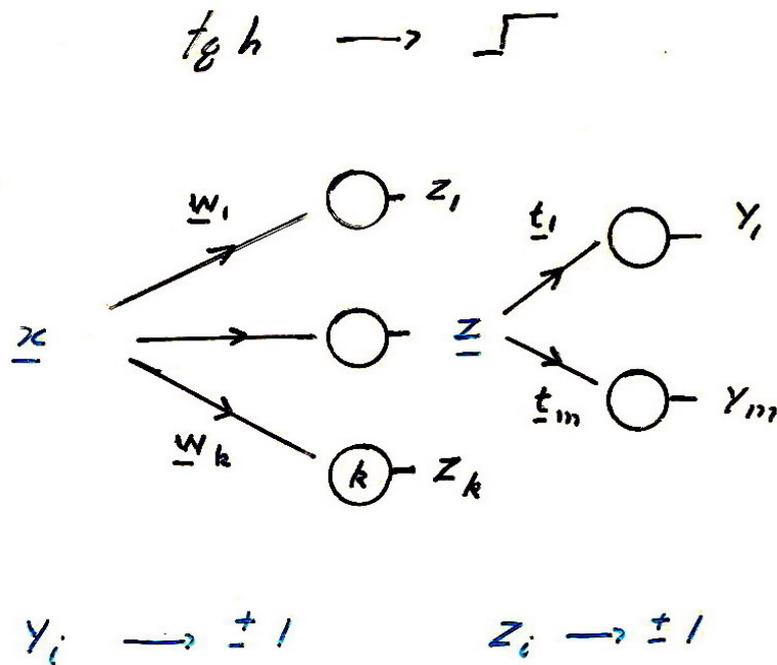


## Outras redes feedforward

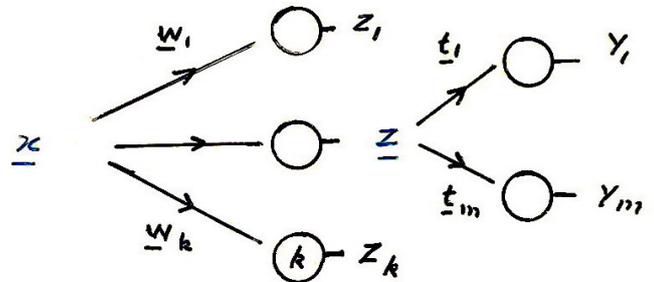
### 1 - Rede de Perceptrons - classificadores



Como saturar os neurônios ?

A rede é treinada

com neurônios tipo tgh (.)



$$y \in \{-1, 1\} \quad \tilde{y} \rightarrow \pm 1$$

$$\tilde{y} = f_{gh} u \quad u \rightarrow \pm \infty$$

$$u = \underline{w}^t \underline{x} + b \quad \underline{w}, b \rightarrow \infty$$

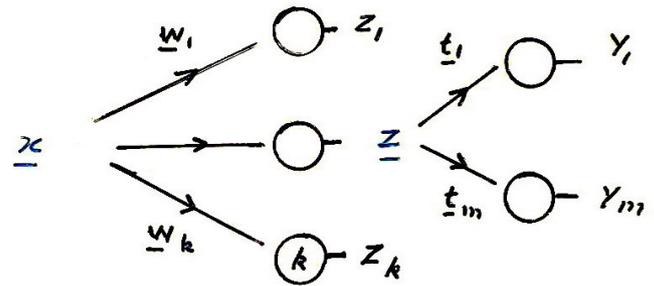
*a saída logica sempre satura!*

E a camada intermediária ?

Para  $\tilde{y} \rightarrow \pm 1$   $u_{saída} \rightarrow \pm \infty$

logo  $w_{saída}, b_{saída} \rightarrow \pm \infty$  e

$z_i \rightarrow \max$   $z_i \rightarrow \pm 1,$



logo os neurônios da camada intermediária também saturam, mas muito mais lentamente porque quando a camada de saída está próximo da saturação  $|\tilde{y}| \approx 1$   $\left| \frac{\partial \tilde{y}}{\partial u} \right| \approx 0$  quase nenhum erro é retropropagado e o treinamento da camada intermediária não evolui. Como resolver ?

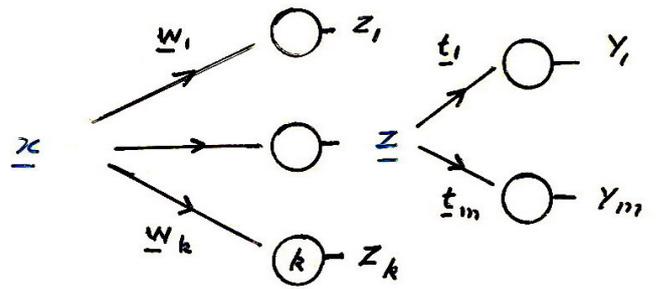
a - Usar entropia relativa (em vez de emq) como função objetivo F

Minimizar a função entropia relativa e não o emq. Se a distribuição dos erros for gaussiana a função entropia relativa apresenta os mesmos minimantes que a e.m.q

Como visto no capítulo “Outros critérios de erro” basta substituir **na rede associada** o ganho de pequenos sinais dos elementos da camada de saída de  $g = (1 - \tilde{y}^2)$  por  $g = 1$

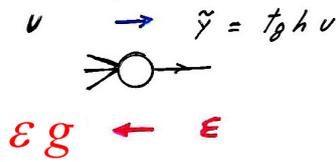
Com  $g = 1$  o erro retropropaga com facilidade provocando a saturação da camada intermediária.

Retropropagação do erro

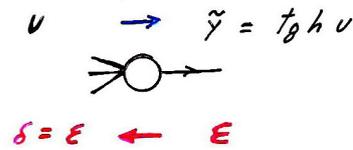


EMQ

Camada de saída



Camada de saída



b – Alterar a função objetivo

$$F = F_0 + \delta F_1$$

$$F_1 = \sum_{i=1}^k (1 - z_i^2)$$

$$F_1 = \sum_{i=1}^k (1 - |z_i|)$$

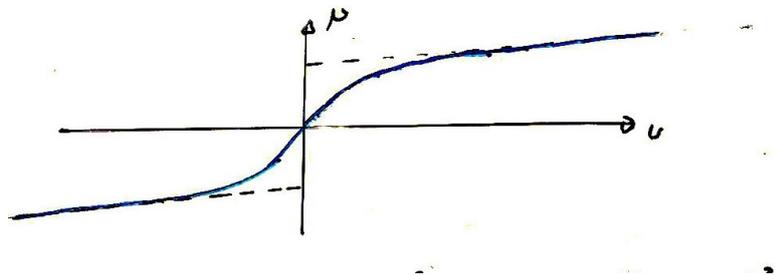
## Treinamento

- 1- Treinar com todos os neurônios  $tgh(u)$
- 2- Transformar os neurônios da camada intermediária em  $sign(u)$
- 3 – Retreinar apenas a camada de saída
- 4- Transformar os neurônios da camada de saída em  $sign(u)$

## 2 - Rede sem paralisia de treinamento

### Camada intermediária

usar a função de ativação não saturada



$$v = \beta u + tgh(u) \quad 0 < \beta \ll 1$$

$$\frac{\partial v}{\partial u} = 1 + \beta - (v - \beta u)^2$$

### Quickprop

Camada de saída em aproximadores: neurônio linear, não paralisa

Camada de saída em classificadores

Usar entropia relativa como função objetivo.

O neurônio satura mas o treinamento não paraliza

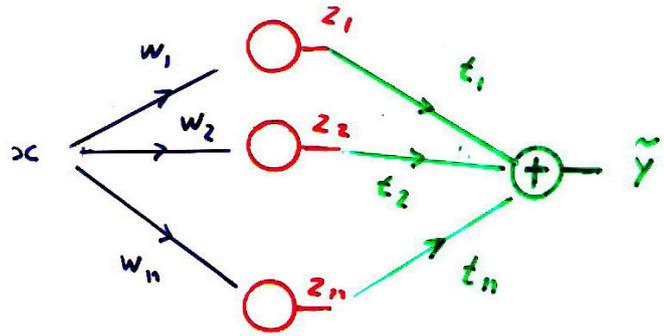
Camada de saída

$$\begin{array}{ccc}
 u & \xrightarrow{\quad} & \tilde{y} = tgh u \\
 & \text{---} \bigcirc \text{---} & \\
 \delta = \epsilon & \xleftarrow{\quad} & \epsilon
 \end{array}
 \qquad
 \frac{d\tilde{y}}{du} = 1$$

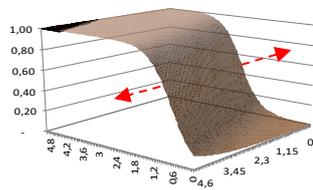
### 3 – RBF – Redes de Base Radial

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

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

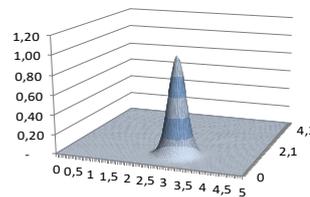


Neurônios tipo tgh(u)



atuação ampla, global

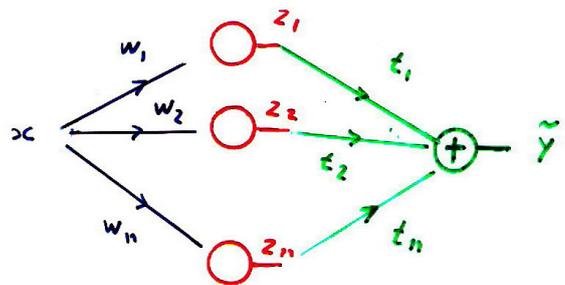
Neurônios de Base Radial



atuação local

### 4 - SVM – Máquinas de Vetor Suporte

$$\tilde{y} = \sum_j \varphi_j(\vec{x})$$



**Polinomial**

$$\varphi_j = P_j(\vec{x})$$

**Perceptrons**

$$\varphi_j = t_j \text{tgh}(\vec{w}_j^t \vec{x} + w_{j0})$$

**Base radial  
gaussiana**

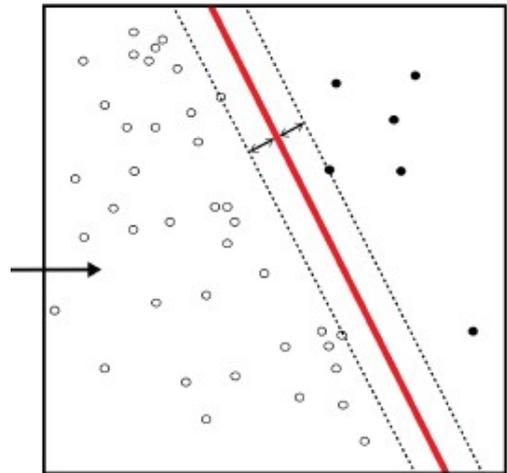
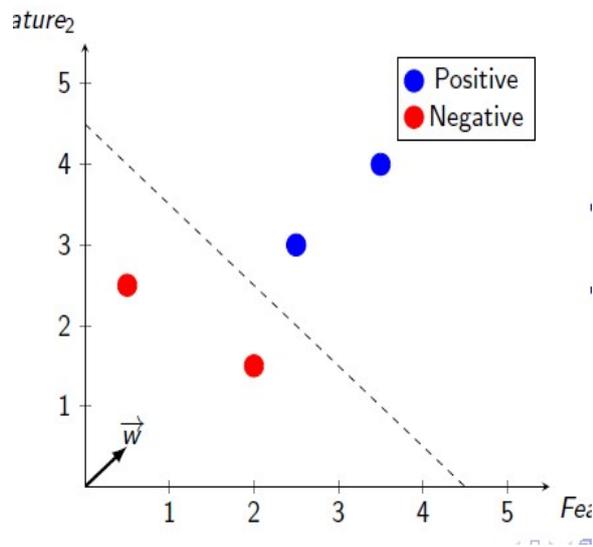
$$\varphi_j = t_j \exp(-\|\vec{x} - \vec{w}_j\|^2 / 2\sigma_j^2)$$

**etc.**

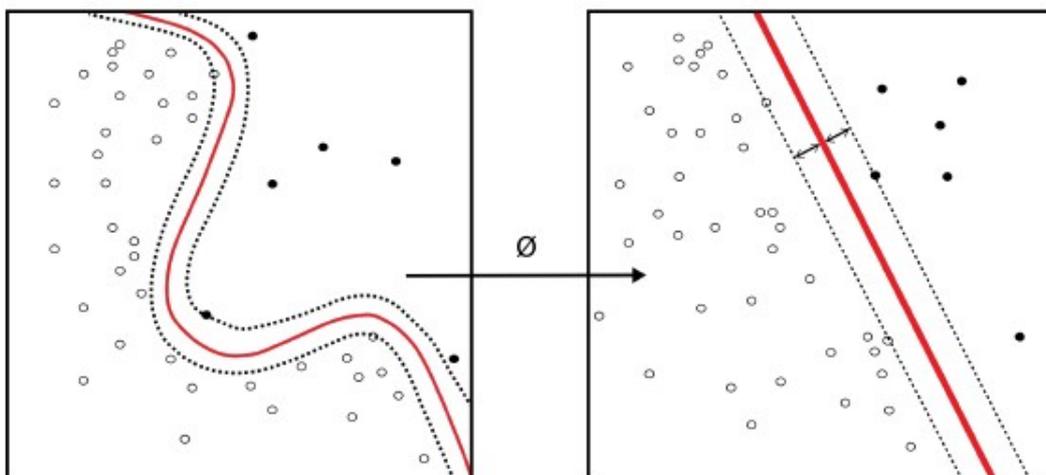
$$\varphi_j = \dots$$

mas o objetivo e o treinamento são diferentes.

### Critério para o melhor separador: Rua Mais Larga - Widest Street Approach

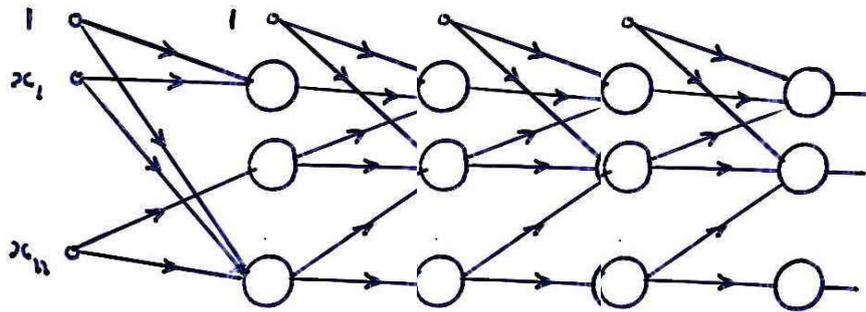


### Rede de duas camadas



# 5 – Deep Learning

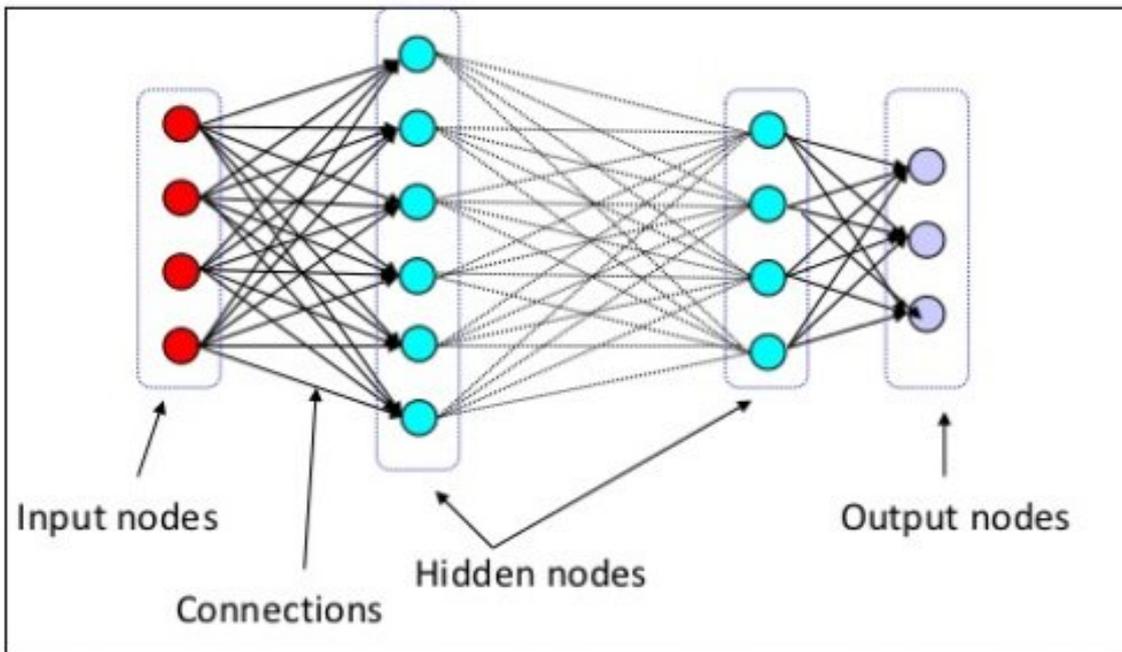
Cortex → Rede multicamadas



Treinamento por retropropagação do erro ?

Camadas intermediárias  
Trein. não supervisionado

Camada de saída  
Trein. Supervisionado



Compactação da informação  
Criação de padrões

Geração da saída  
(a partir dos padrões)

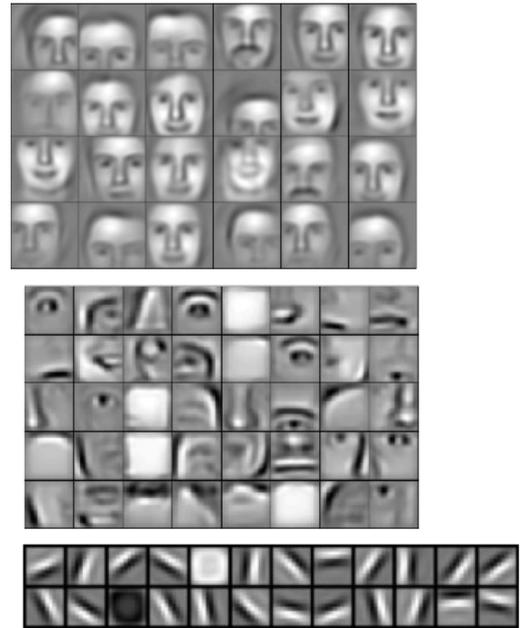
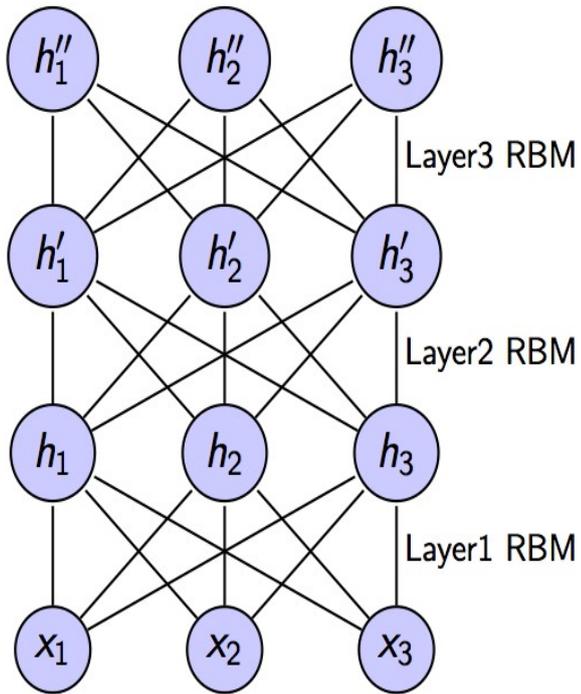
## **Treinamento das camadas intermediárias (não supervisionado)**

- **DBN - Deep Belief Networks – RBM - Restricted Boltzmann Machines** - geração de novos dados mantendo estrutura estatística do processo
- **SAE – Auto encoders** - compactação de dados de altas dimensões e estatística de alta ordem
- **CNN - Redes Neurais Convolucionais** - informações espaciais em grandes dimensões
- **RNN - Redes Neurais Recursivas** - informações temporais.

## **Treinamento da camada de saída (supervisionado)**

- **BP – Backpropagation**

## CNN - Redes Neurais Convolucionais



## CNN - Redes Neurais Convolucionais

