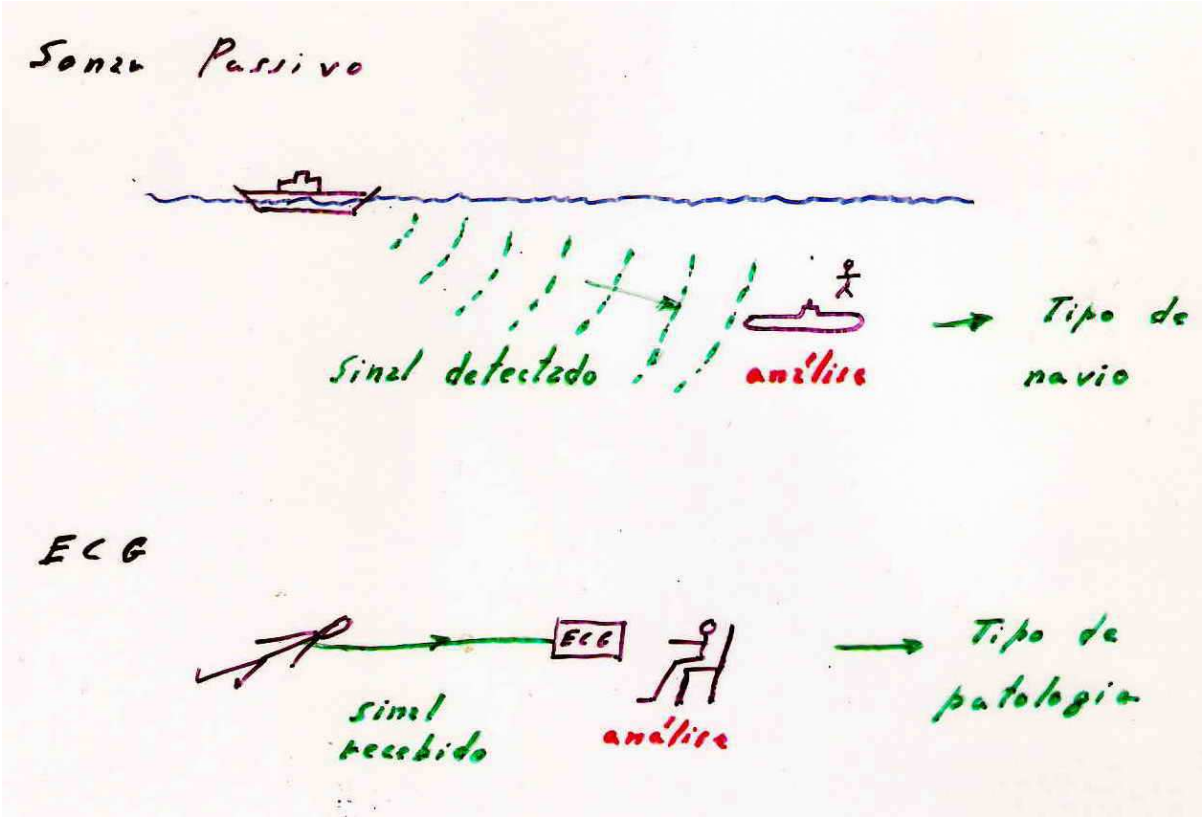
A Rede Neural como um Classificador

Associadores (variáveis de saída contínuas) e

$$\underline{\mathbf{X}} \Longrightarrow \underline{\mathbf{Y}} \qquad y_i \in (-1,+1)$$

Classificadores (variáveis de saída discretas)

$$\underline{\mathbf{X}} \Longrightarrow \underline{\mathbf{Y}} \qquad y_i \in \{-1,+1\}$$

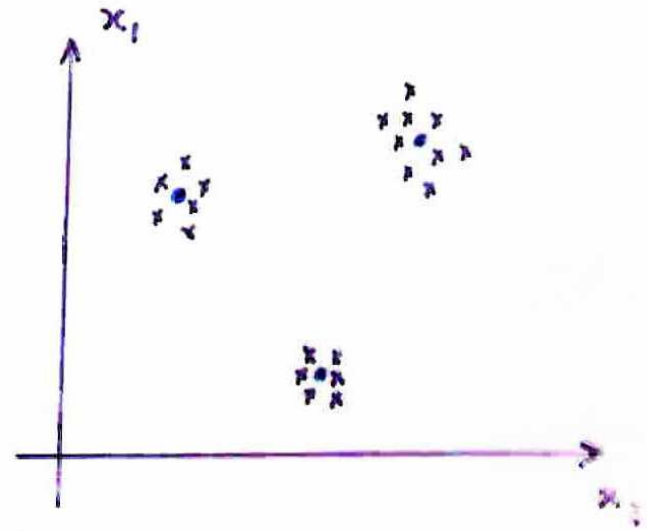


Tipos de classificadores:

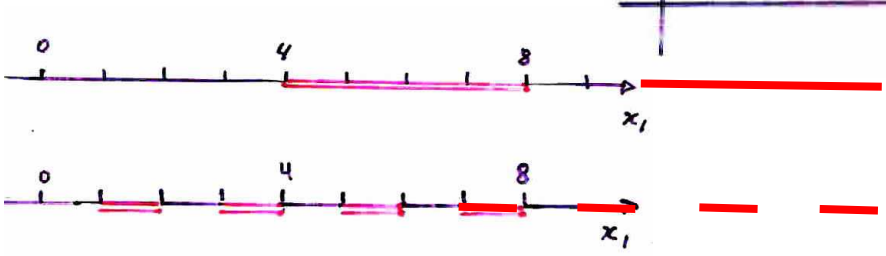
Por similaridade

$$\underline{x}^k, \underline{x}^l \in C_i$$

$$\left|\underline{x}^k - \underline{x}^l\right| < d$$



Arbitrários

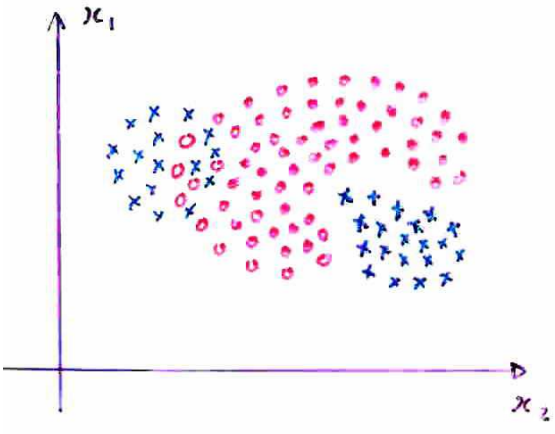


Regiões

- de difícil separação
- de fusão

confusão

entre classes



A rede neural como classificador

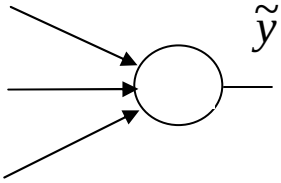
Saídas lógicas:

neurônio tipo tgh (.) na camada de saída

Convenção para as saídas

$$y \in \{-1,+1\}$$

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



Obs: não usar {0, 1} para y

Interpretação da saída da rede como variável lógica

$$y \in \{-1,+1\}$$

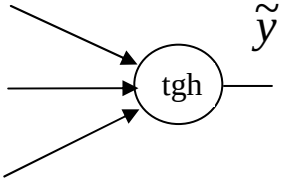
$$\tilde{y} = tgh(u) \in (-1,+1)$$

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

Erro de Classificação versus emq

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

$$\tilde{y}_{\log ico} = sign(\tilde{y})$$



Erro médio quadrático

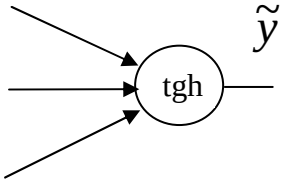
$$\frac{1}{N}\sum_i (y_i - \tilde{y}_i)^2$$

Erro de classificação

$$\frac{1}{4N}\sum_i (y_i - \tilde{y}_{i\ lógico})^2$$

% de classificações erradas em y_i

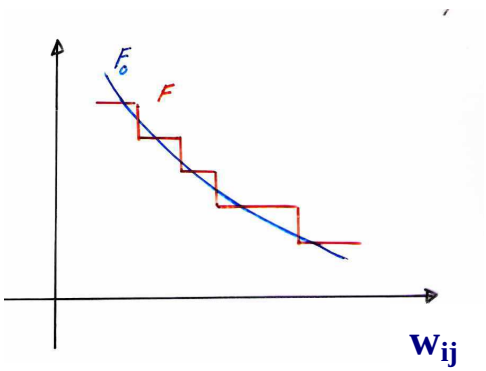
Variação do erro rms F_0 e do erro lógico F_1 com a variação das sinapses

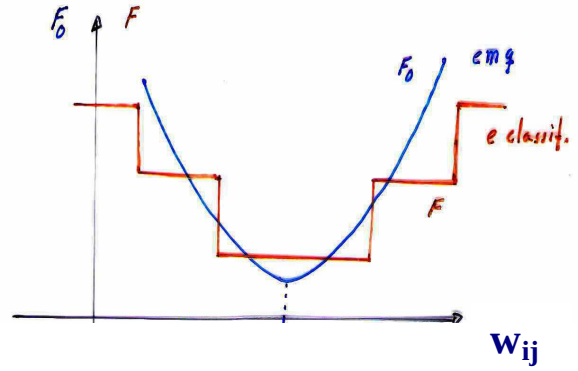


$$u = \sum w_i x_i$$

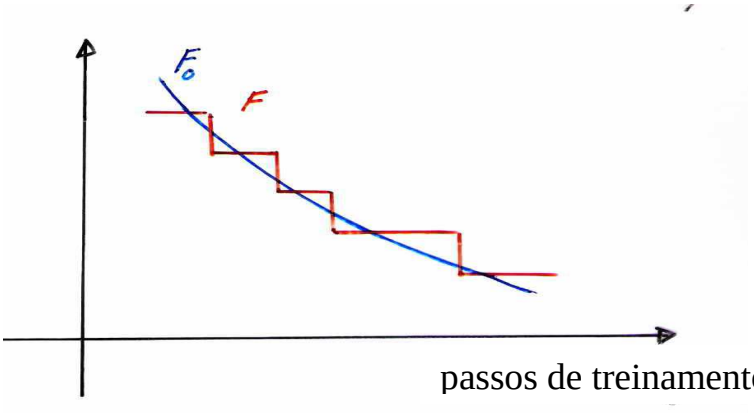
$$\tilde{y} = tgh(u) \qquad F_0 = F_0(\tilde{y})$$

$$\tilde{y}_{\log ico} = sign(\tilde{y}) \qquad F_1 = F_1(\tilde{y}_{\log ico})$$





Variação do erro rms F_0 e do erro lógico F_1 com a variação das sinapses



Como treinar a rede ?

$$\frac{\partial F_{class}}{\partial w_{ij}} = \frac{\partial F_{class}}{\partial \tilde{y}_{logico}} \frac{\partial \tilde{y}_{logico}}{\partial w_{ij}} \qquad \frac{\partial \tilde{y}_{logico}}{\partial w_{ij}} = \frac{\partial sign(\tilde{y})}{\partial w_{ij}}$$

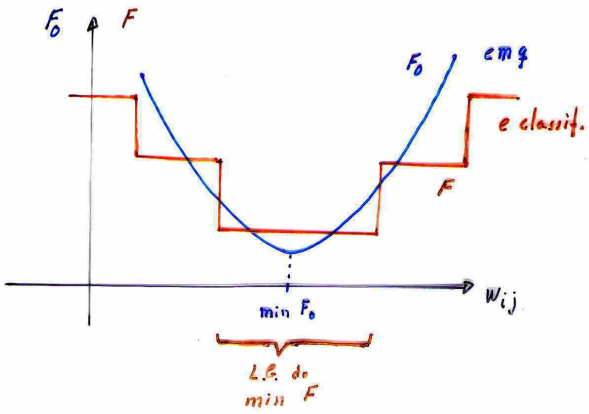
????

Teorema (sem prova):

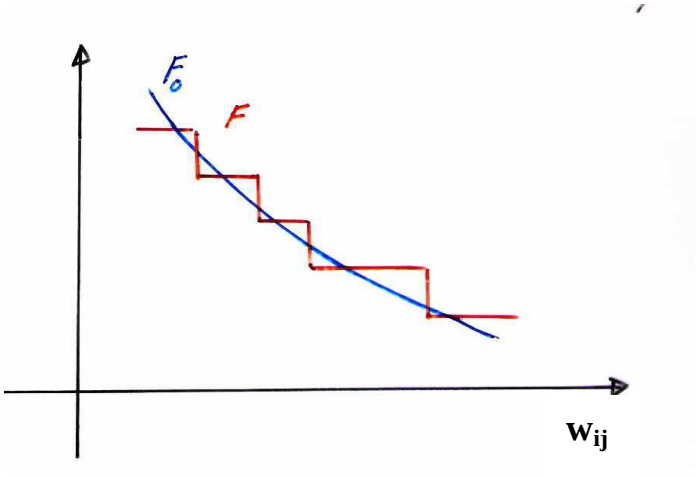
Se no treinamento de uma saída lógica y_i os valores esperados (-1, 1) coincidem com os valores de saturação do neurônio (tgh) então

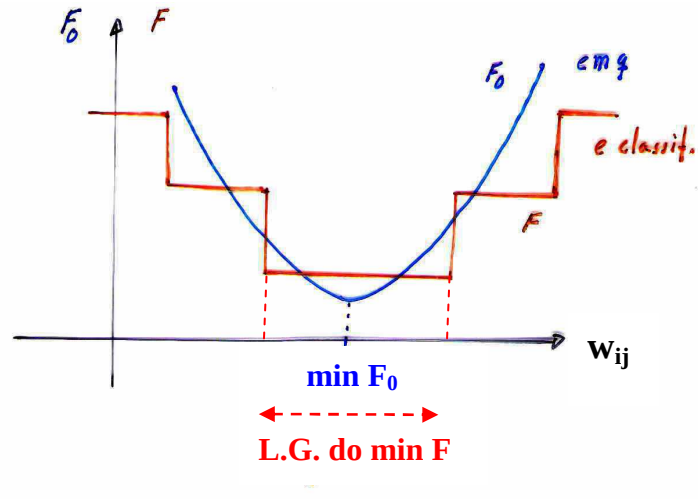
O minimante do erro médio quadrático também é minimante do erro médio de classificação.

Conclusão: o treinamento backpropagation minimiza o erro de classificação



Obs: a recíproca não é verdadeira, um minimante do erro de classificação não é obrigatóriamente minimante do emq





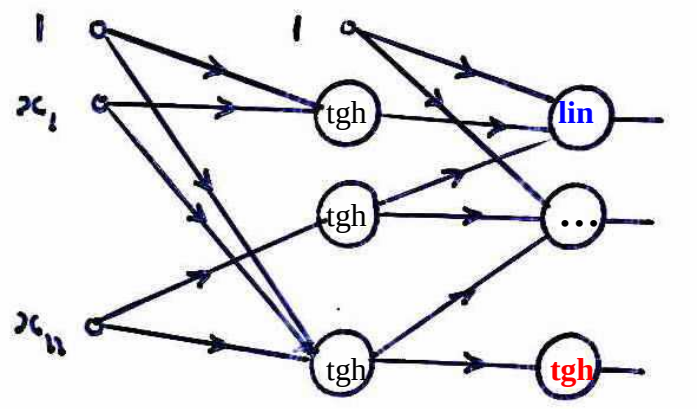
Obs 1:

Ponto fraco: O número de neurônios na camada intermediária não pode ser facilmente pré-determinado !

Obs 2: Capacidade de Mapeamento das Redes -
Saídas contínuas e lógicas na mesma rede

Teorema (sem prova):

Uma rede neural com duas camadas com (a) uma camada intermediária com neurônios tipo tgh(.) e (b) uma camada de saída com neuronios lineares para as saídas contínuas e neuronios tipo tgh para saídas lógicas treinada por backpropagation pode realizar qualquer mapeamento L^2 .

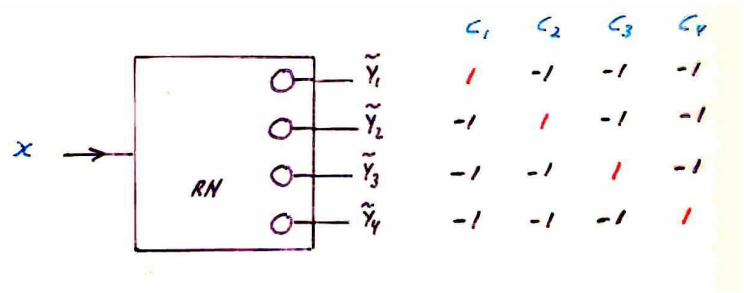


Em casos complexos a convergência do treinamento pode ficar dificultada pela natureza muito diferente dos erros das saídas lógicas e contínuas. Neste caso pode ser conveniente empregar duas redes distintas, uma para as saídas contínuas e outra para as saídas lógicas.

Caso Múltiplas Classes

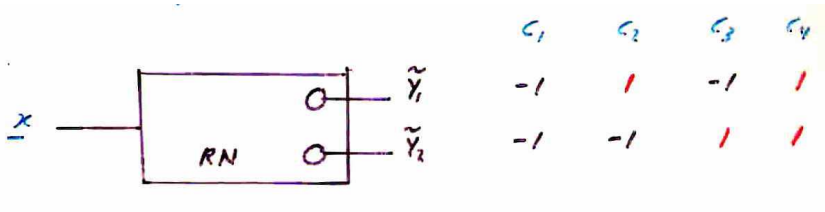
obrigatóriamente utilizar

Codificação Máximamente Esparsa



não utilizar

Codificação Compacta

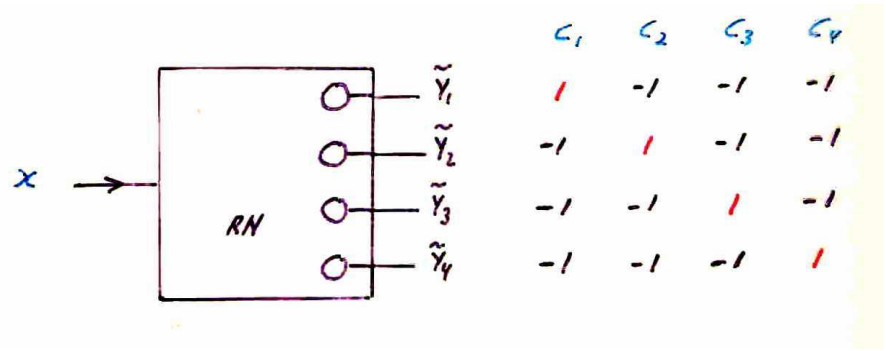


ex: BCD

classificação compacta: classificação + codificação simultaneamente.

não utilizar.

Codificação máximamente esparsa



Cada saída separa a sua classe das demais, i.e.

classe versus não classe

Separação de N classes ⟹ Separação de duas classes:

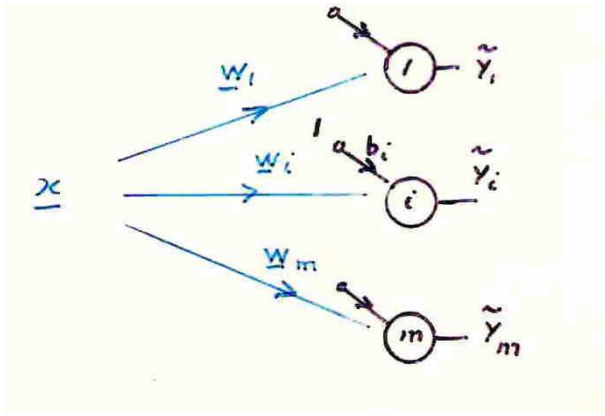
classe versus não classe

Capacidade de mapeamento

Redes com uma camada

m classes

$$y_i = \begin{cases} +1 & \Leftrightarrow \quad \underline{x} \in C_i \\ -1 & \Leftrightarrow \quad \underline{x} \notin C_i \end{cases}$$



O treinamento e a operação de cada neurônio é independente dos demais

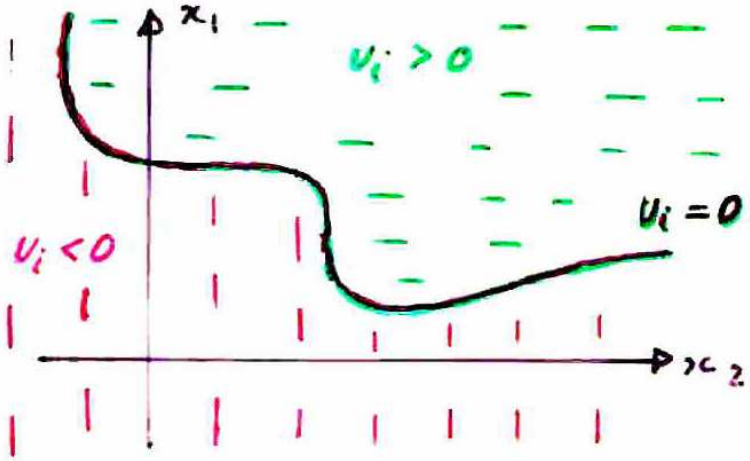
Cada neurônio i separa sua classe C_i de todas as demais

Operação

$$\underline{x} \Rightarrow \begin{cases} u_i > 0 \Leftrightarrow y_i > 0 \Leftrightarrow \underline{x} \in C_i \\ u_i < 0 \Leftrightarrow y_i < 0 \Leftrightarrow \underline{x} \notin C_i \end{cases}$$

Separador ?

u_i=0



Separador

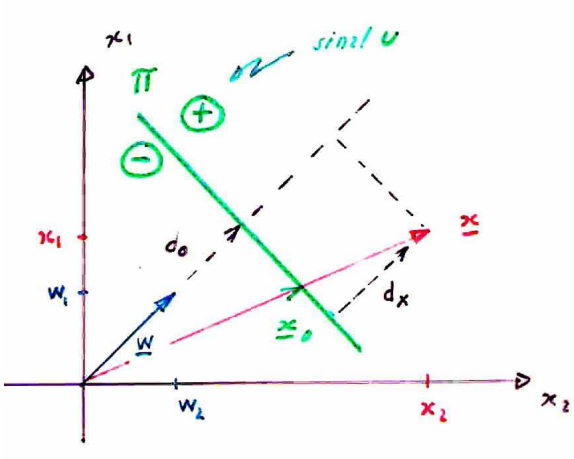
$u = \underline{w}^t \underline{x} + b = 0$ *plano* π $\underline{x} = \underline{x}_0$

$\underline{w}^t \underline{x}_0 = |\underline{w}|\ |\underline{x}_0| \cos \angle \underline{w}, \underline{x}_0 = -b$ $|\underline{x}_0| \cos \angle \underline{w}, \underline{x}_0 = ctte(\underline{x}_0)$

$\pi \perp \underline{w}$

$$d_0 = |\underline{x}_0| \cos \angle \underline{w}, \underline{x}_0 = -\frac{b}{|\underline{w}|}$$

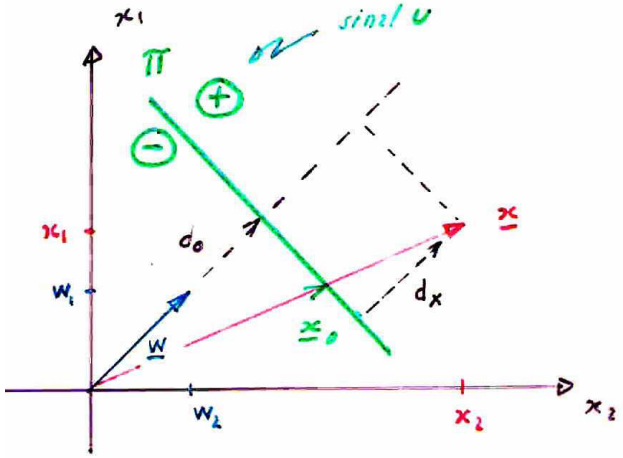
(medido no sentido de $\underline{w}$)



Lado positivo do plano ?

$u = \underline{x}^t \underline{w} + b > 0$

no sentido de $\underline{w}$



Distância de $\underline{x}$ ao plano

$$d_x = |\underline{x}_0| \cos \angle \underline{w}, \underline{x} - d_0 = \frac{\underline{w}^t \underline{x}}{|\underline{w}|} + \frac{b}{|\underline{w}|} = \frac{u}{|\underline{w}|}$$ (medido no sentido de $\underline{w}$)

Se normalizarmos $\underline{w} \Rightarrow \frac{\underline{w}}{|\underline{w}|}$ *e* $b \Rightarrow \frac{b}{|\underline{w}|}$

o classificador não se altera mas $d_x = u$

Interpretação geométrica / estatística

$$\underline{x} \in C_i \qquad \underline{w}^t \underline{x} > -b_i$$

$u_i = \underline{x}^t \underline{w}_i + b_i$

$$\underline{x} \notin C_i \qquad \underline{w}^t \underline{x} < -b_i$$

Pela definição da função objetivo os parâmetros do separador, $\underline{\mathbf{w}}_\mathbf{i}$ e $\mathbf{b_i}$ classificam a entrada de forma ótima, i.e.

minimizam o erro de classificação

levando em conta **inclusive** a população

>>> **a RN é um classificador Bayesiano**

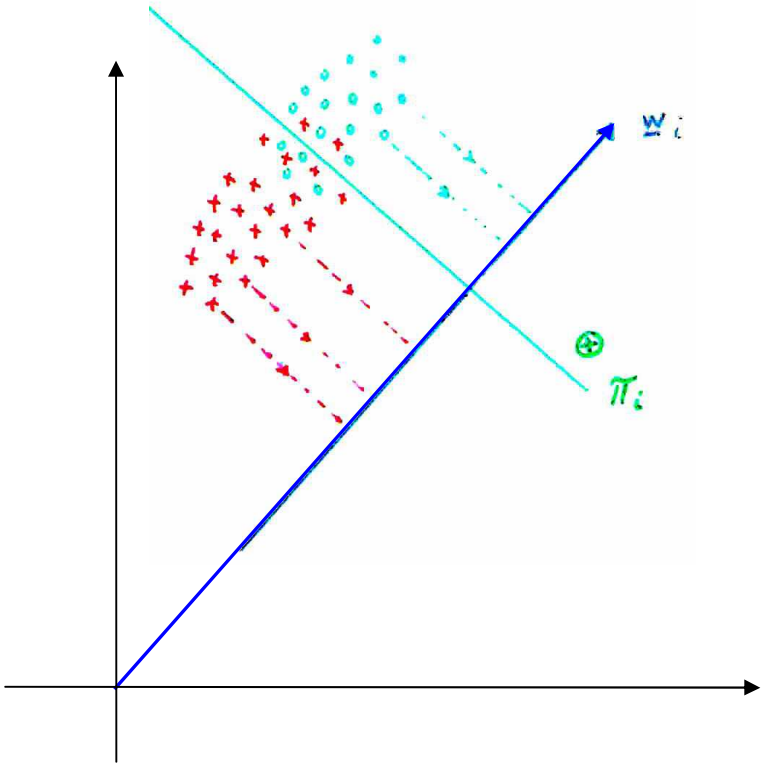
$\underline{\mathbf{w}}_\mathbf{i}$ – direção ótima na qual a entrada $\underline{\mathbf{x}}$ deve ser projetada
componente principal de discriminação

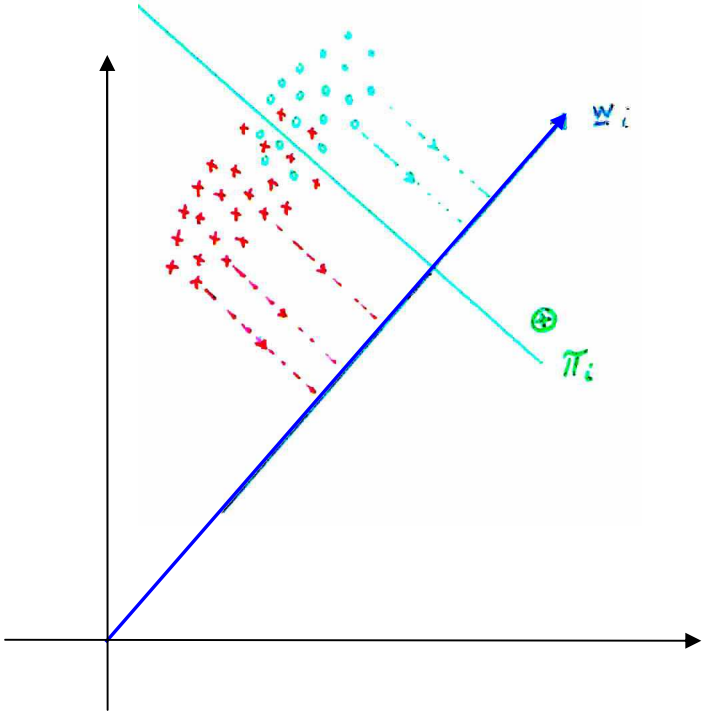
$\mathbf{b_i}$ **–** nível de discriminação ótimo

discriminante linear ótimo

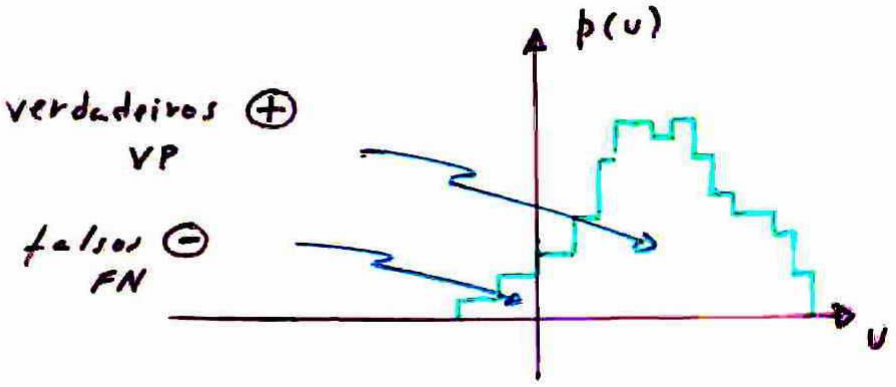
discriminante de Fischer

maximiza a probabilidade de acerto na classificação

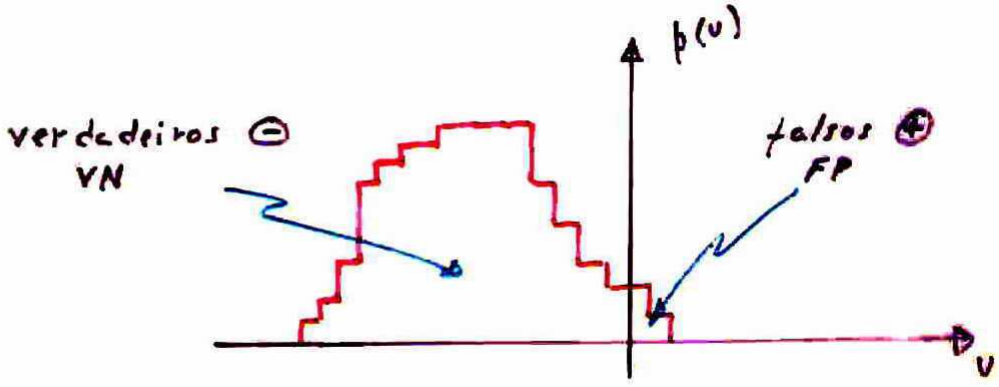


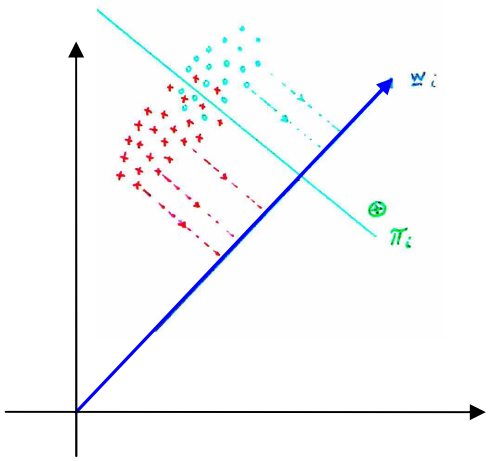


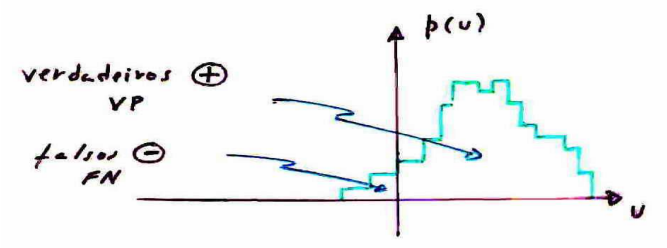
$\underline{x} \in C_i$

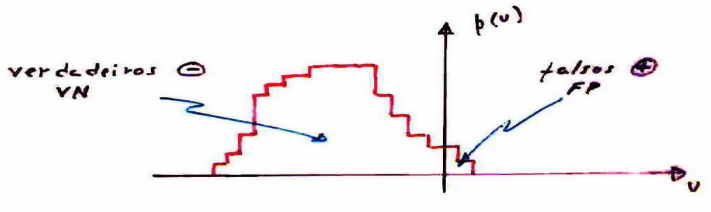


$\underline{x} \notin C_i$









Erro de classificação (minimizado):

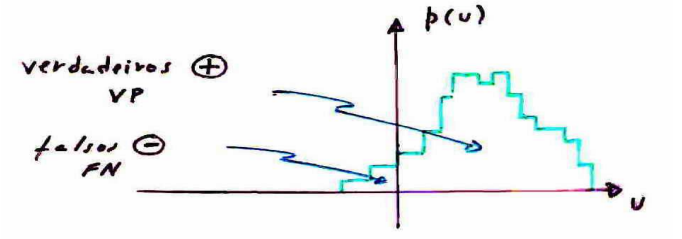
$$e_{class} = \frac{FP + FN}{VP + VN + FP + FN}$$

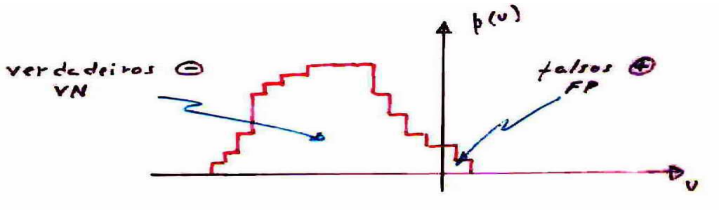
Outros parâmetros usuais

Taxa de acerto $TA = \frac{VP + VN}{VP + VN + FP + FN} = 1 - e_{class}$

Sensibilidade $S = \frac{VP}{VP + FN}$

Especificidade $E = \frac{VN}{VN + FP}$





Valor preditivo positivo $VPP = \frac{VP}{VP + FP}$

Valor preditivo negativo $VPN = \frac{VN}{VN + FN}$

Falsos Alarmes $FA = \frac{FP}{VN + FP} = 1 - E$

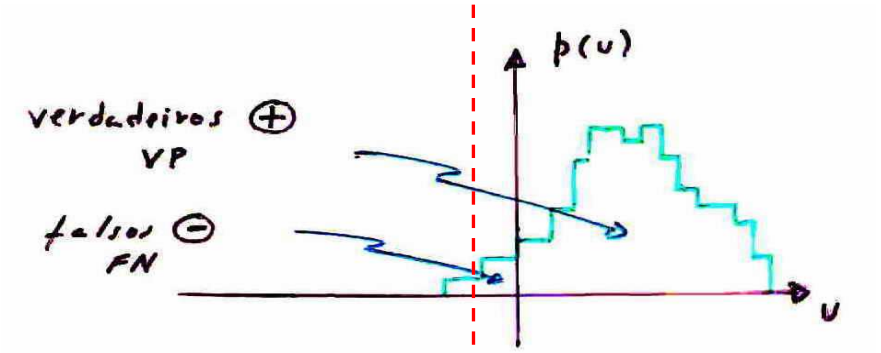
Falsas Perdas $FP = \frac{FN}{VP + FN} = 1 - S$

Ajustando a rede para

alterar os parâmetros

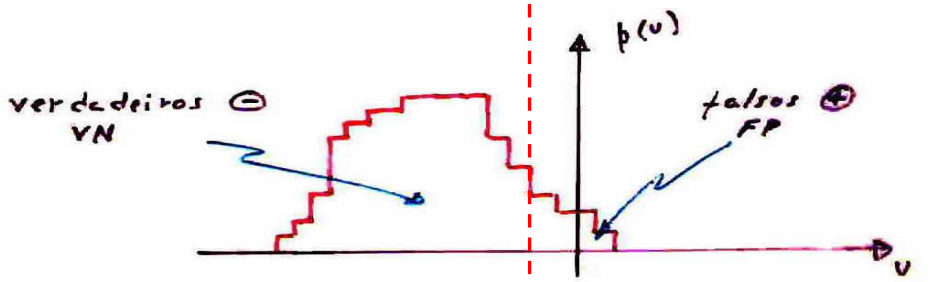
reduzir FN

(mas aumenta FP !)



reduzir (–b)

aumentar b



problema: baixa população nas caudas
analiticamente: aproximar a cauda da distribuição

Aproximação da cauda da distribuição

– caso cauda gaussiana

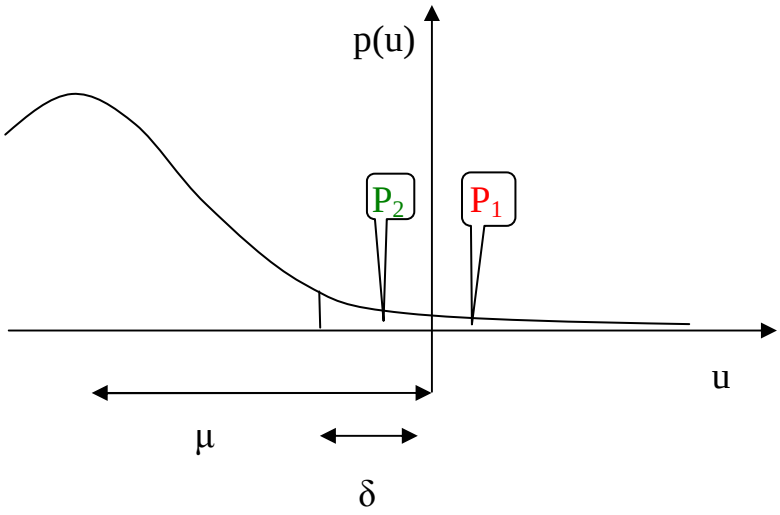
δ arbitrário, pequeno

$P_1 = 1 - erf(z_1)$

$P_1 + P_2 = 1 - erf(z_2)$

$z_1 = \mu/\sigma$

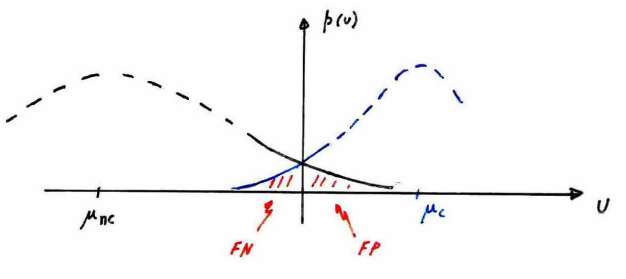
$z_2 = (\mu - \delta)/\sigma = z_1 - \delta/\sigma$

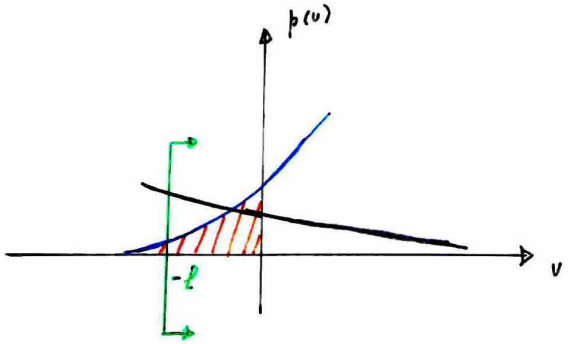


$\sigma = \delta/(z_1 - z_2)$

$\mu = \delta\, z_1/(z_1 - z_2)$

Ex: Ajuste de Falsas Perdas





$$FN = 1 - erf\left(\frac{\mu_C + l}{\sigma_C}\right) \qquad FN_{desejado} = 1 - erf(z)$$

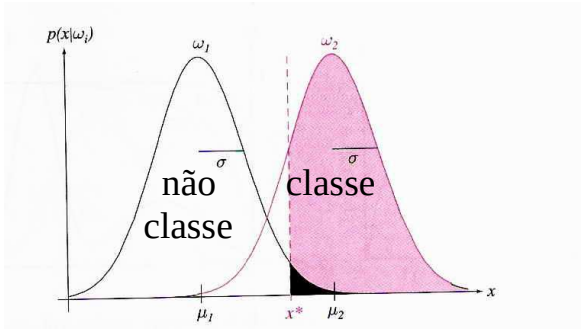
$$z = \frac{\mu_C + l}{\sigma_C} \qquad l = z\sigma_C - \mu_C \qquad b_{novo} = b_{antigo} + l$$

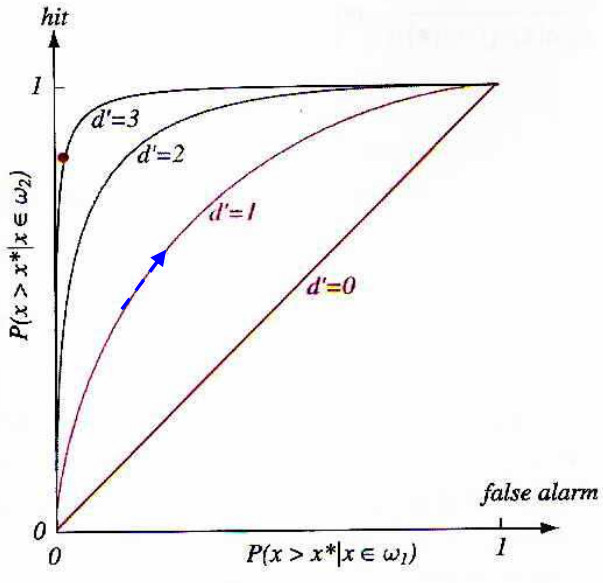
basta alterar o bias

Visualização: Curvas ROC
– receiver operating characteristic

Sensibilidade (hits) vs.

Falsos Alarmes

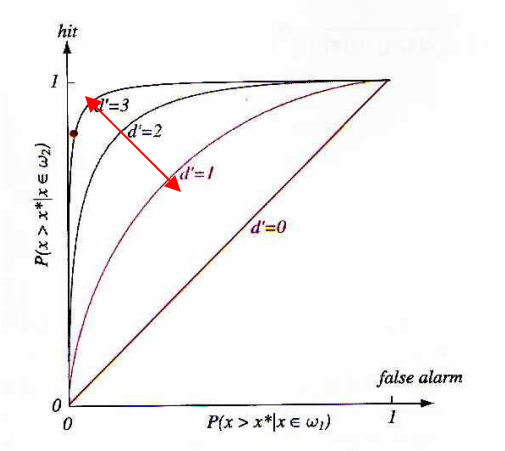




Fator de Mérito SP (soma-produto)

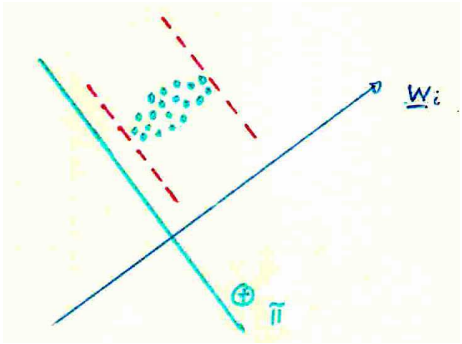
$$SP = \sqrt{\left(\frac{S+E}{2}\right)\sqrt{S.E}} = \sqrt{\left(\frac{S+(1-FA)}{2}\right)\sqrt{S(1-FA)}}$$

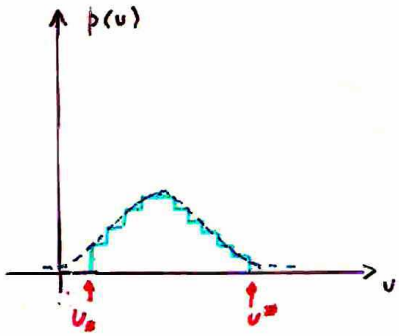
ponto ótimo = máx SP



Fechando (restringindo) a classificação

encapsulando a classe





$u_{\min}$, $u_{\max}$

$$\underline{x} \in C_i \quad \Leftrightarrow \quad u_{\min} < u_i < u_{\max}$$

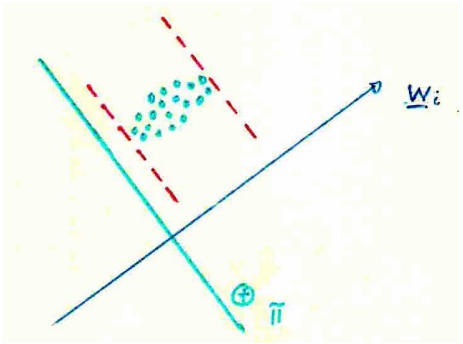
Múltiplas classes - Tabela de Confusão

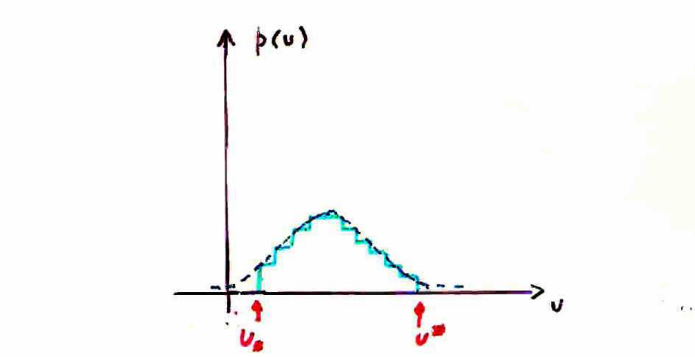
Classes corretas	Resultado da classificação					
	C_1	C_2	…	C_i	…	C_N
C_1						
C_2						
…						
C_j				a_{ij}		
…						
C_N						

a_{ij} - % de elementos de C_j classificados como C_i

Fechando (restringindo) a classificação

encapsulando a classe





$u_{\min}$, $u_{\max}$

$\underline{x} \in C_i \quad \Leftrightarrow \quad u_{\min} < u_i < u_{\max}$

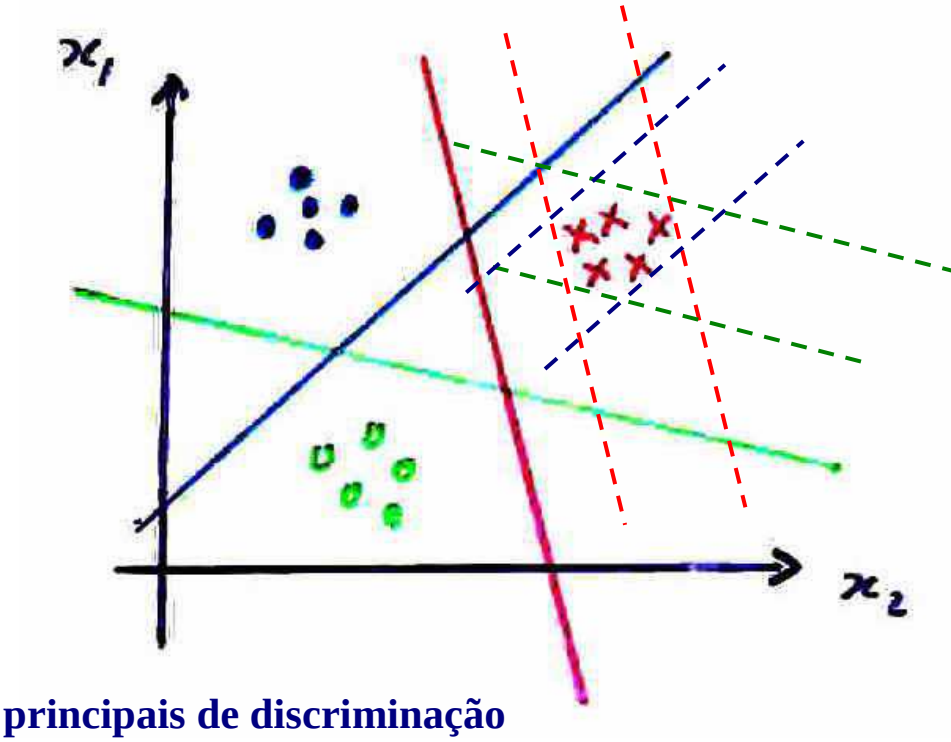
Fechamento em mais direções ?

Para pertencer a
classe vermelha

$u_1 < u_{verm} < u_2$

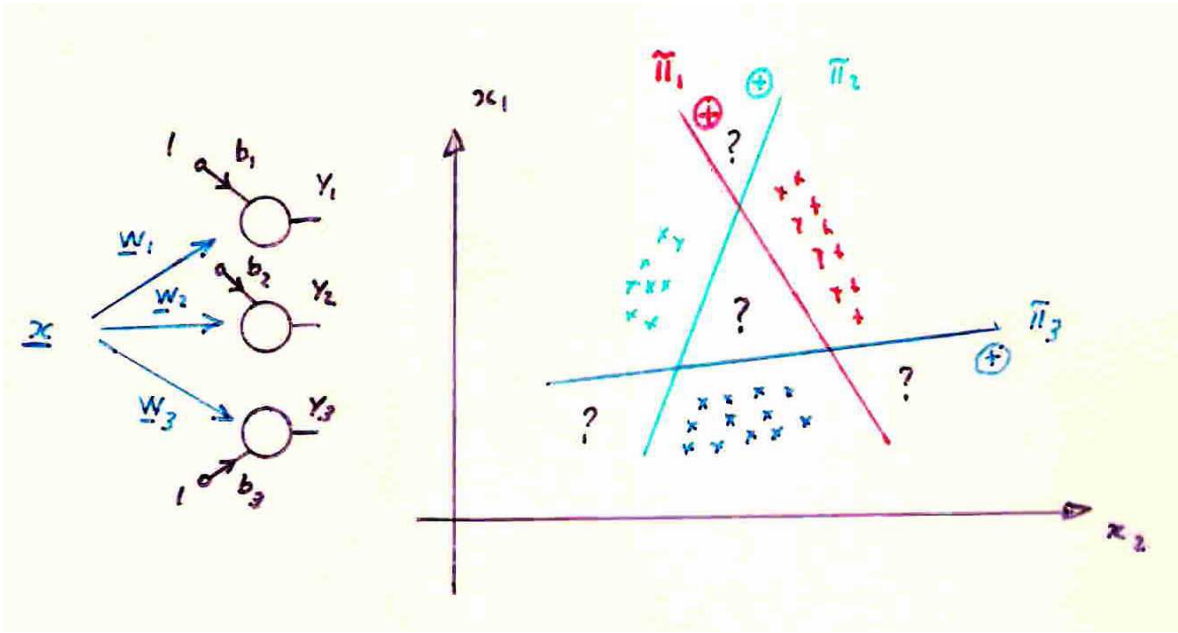
$u_3 < u_{verde} < u_4$

$u_5 < u_{azul} < u_6$



Ver PCD - componentes principais de discriminação

Abrindo a classificação



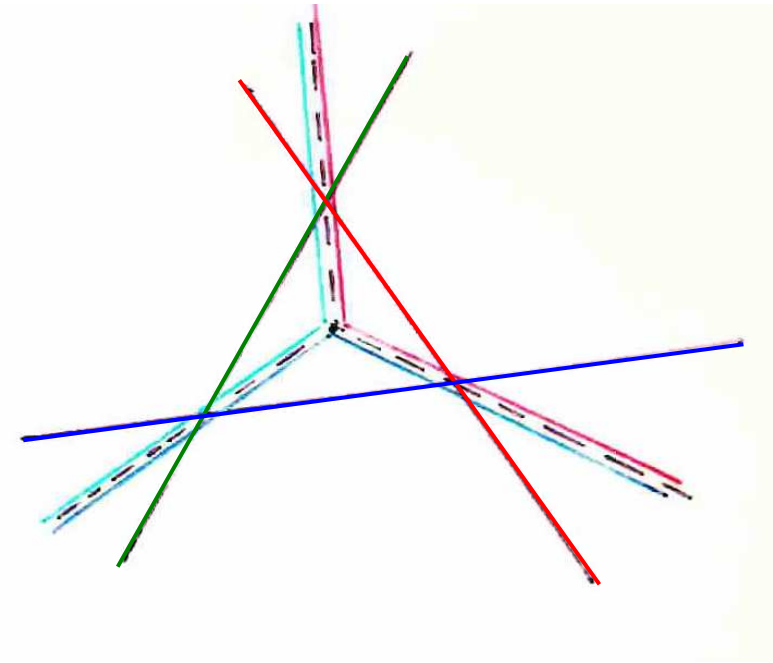
$$\underline{x} \in C_i \quad \Leftrightarrow \quad u_i > u_j \quad \forall \quad j \neq i$$

$$\underline{x} \in C_i \quad \Leftrightarrow \quad u_i > u_j \quad \forall \quad j \neq i$$

Separador

$$u_i = u_j$$

$$Se \quad |\underline{w}_i| = |\underline{w}_j| \quad d_i = d_j$$



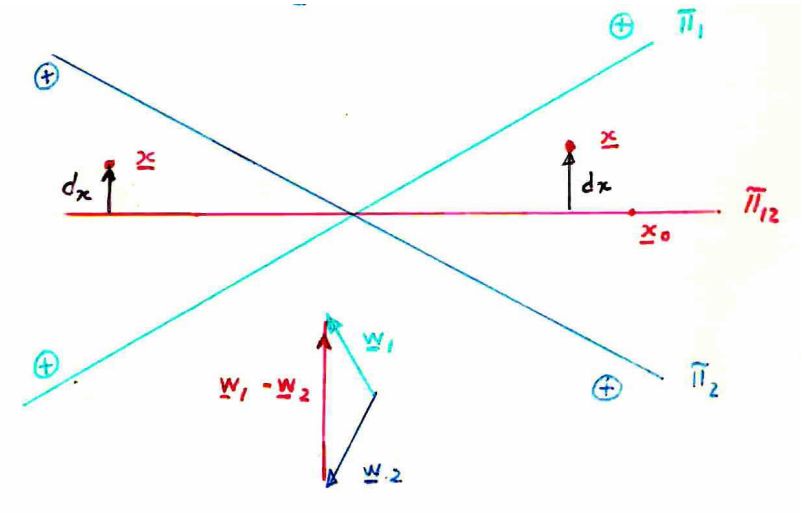
bissetrizes

Obs:

Obs 1: Distância da entrada à bissetriz

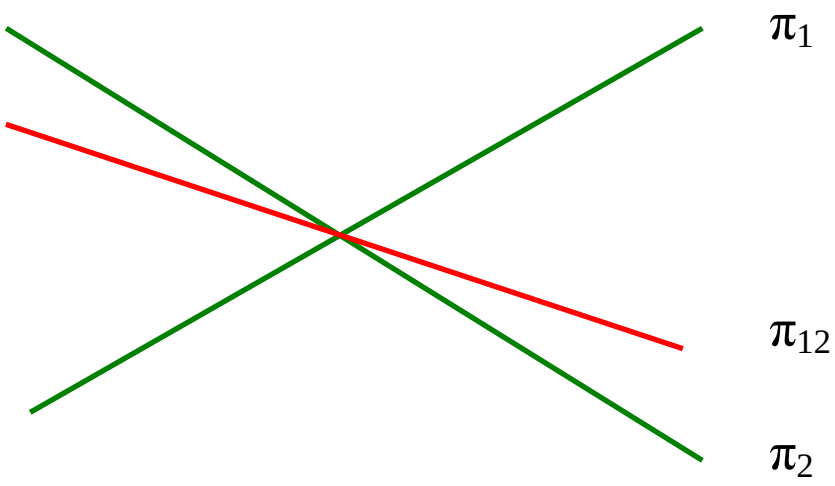
$$d_x = \frac{(\underline{w}_1 - \underline{w}_2)^t \underline{x} + (b_1 - b_2)}{|\underline{w}_1 - \underline{w}_2|} = \frac{u_1 - u_2}{|\underline{w}_1 - \underline{w}_2|}$$

no sentido de w_1-w_2



Obs 2: no caso em que a normalização dos módulos das sinapses $\underline{w}_i$ não foi feita, o novo separador é simplesmente um plano passando pela intersecção dos anteriores, mas não equidistante dos mesmos.

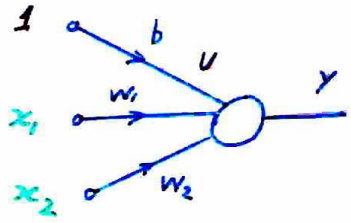
$$u_i = u_j \quad \Leftrightarrow \quad \frac{d_i}{|\underline{w}_i|} = \frac{d_j}{|\underline{w}_j|}$$

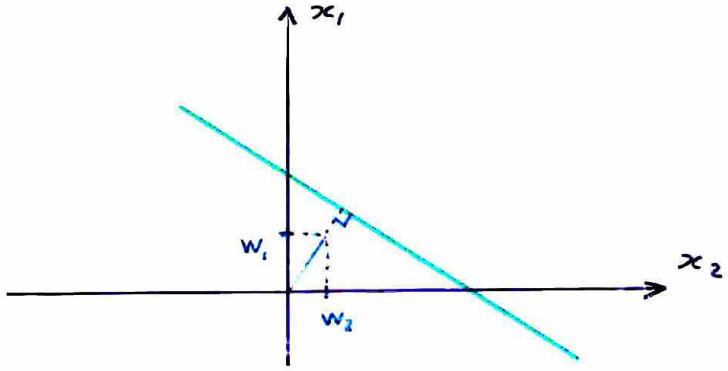


Capacidade de classificação

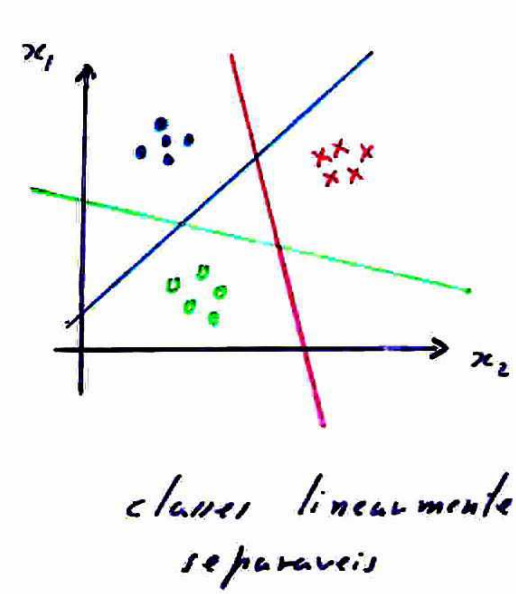
Redes de uma camada – separadores lineares

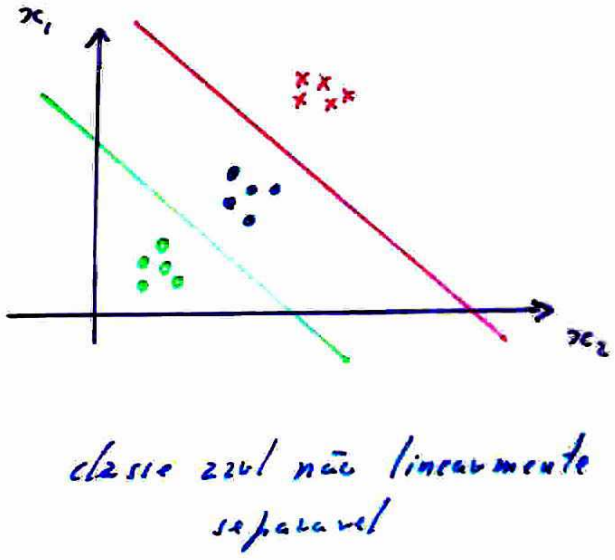
$$u = w_1 x_1 + w_2 x_2 + b = 0$$



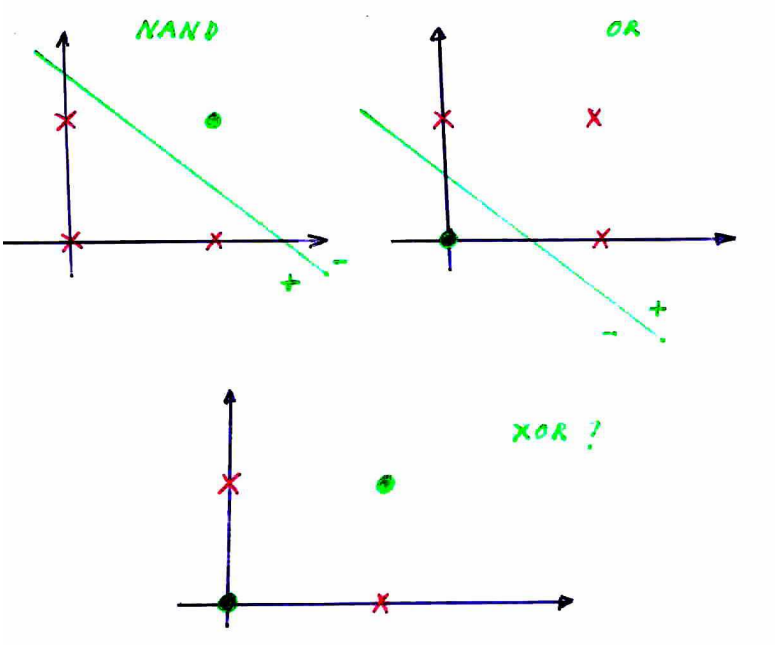


Classes linearmente e não linearmente separáveis





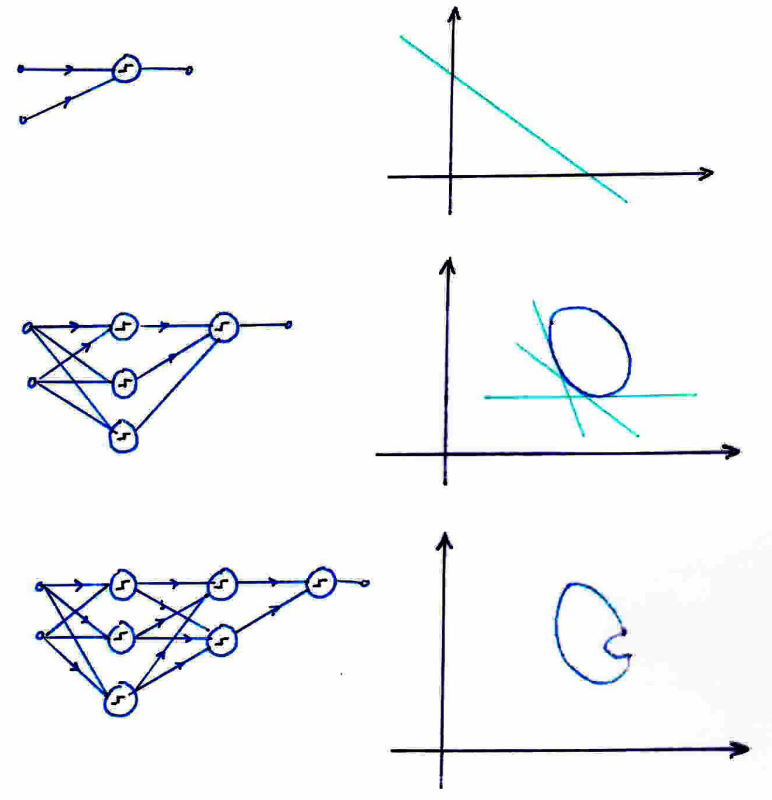
Os exemplos clássicos:



Redes com mais de uma camada:

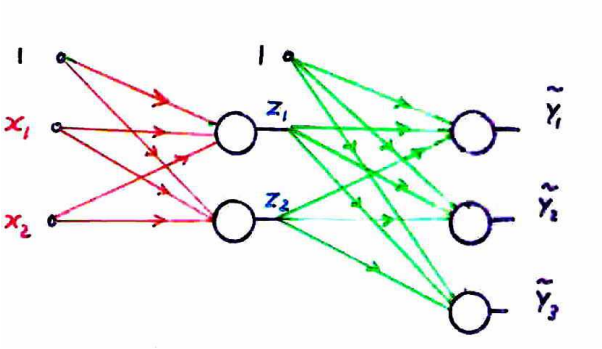
bastam duas camadas

(realização por mintermos ou maxtermos)



Redes com duas camada, neurônios tipo tgh(.)

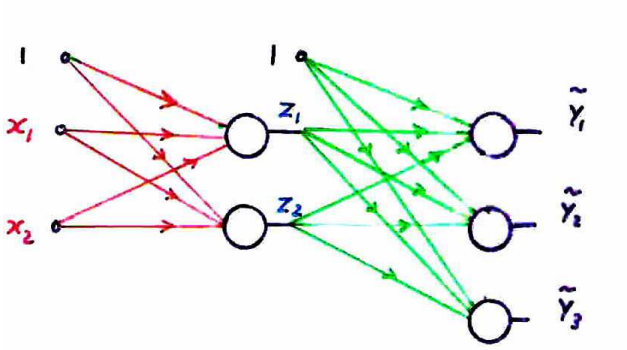
x >>> z >>> y

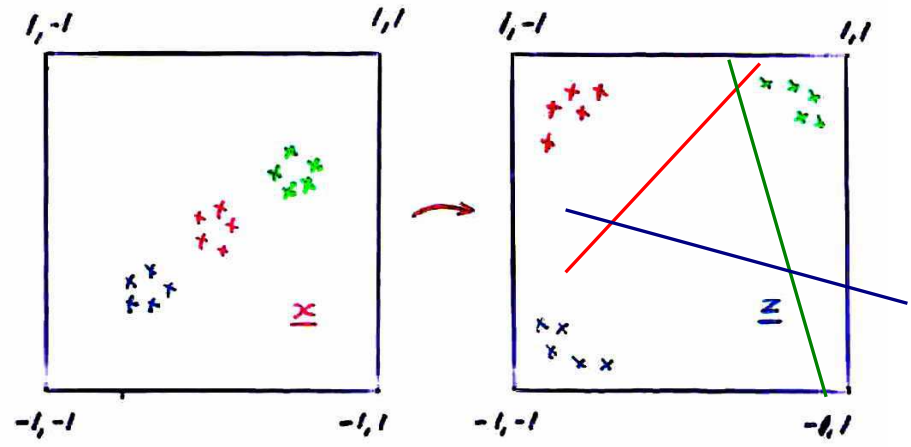


Camada de saída: separa classes linearmente separáveis no domínio z

Camada intermediária: mapeia classes não linearmente separáveis em x

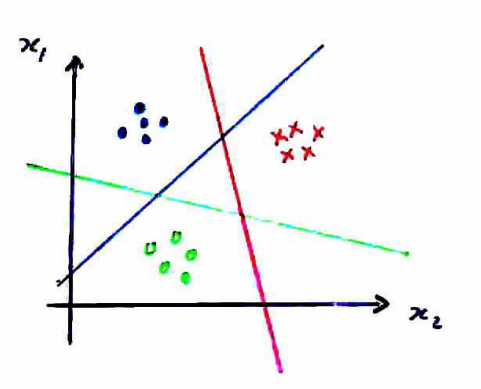
em classes linearmente separáveis em z



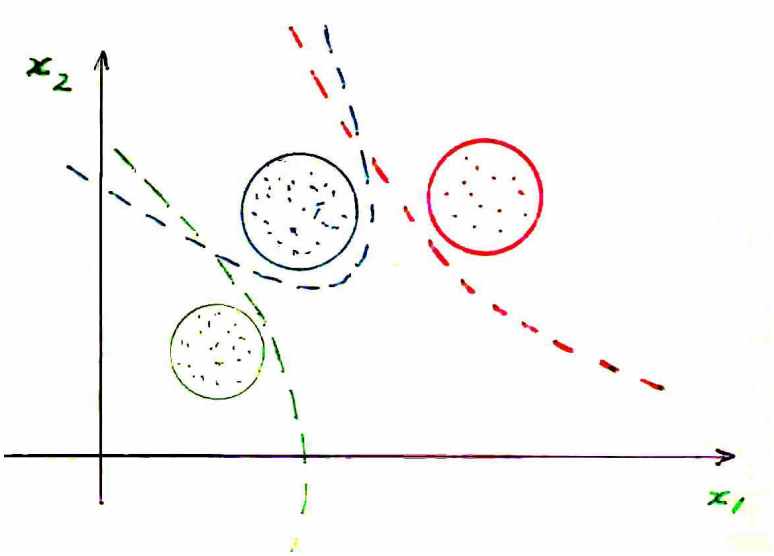


Partição do espaço de entrada

1 camada – planos separadores



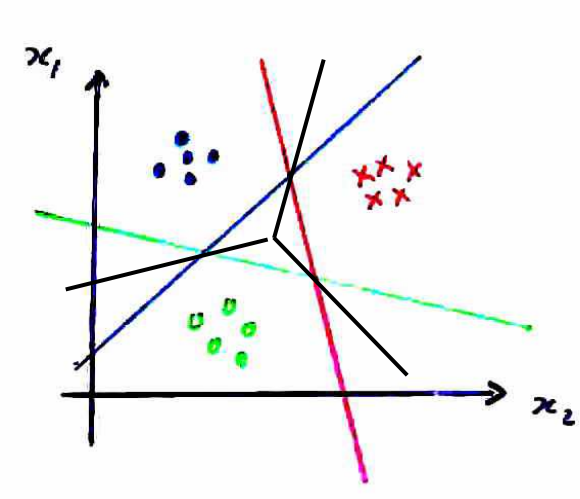
2 camadas – superfícies separadoras



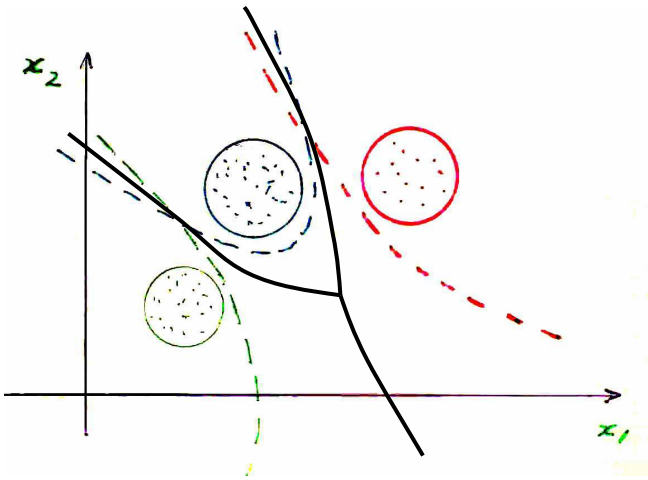
Abrindo a classificação

Vencedor – maior u (ou y) na camada de saída

1 camada – planos bissetores



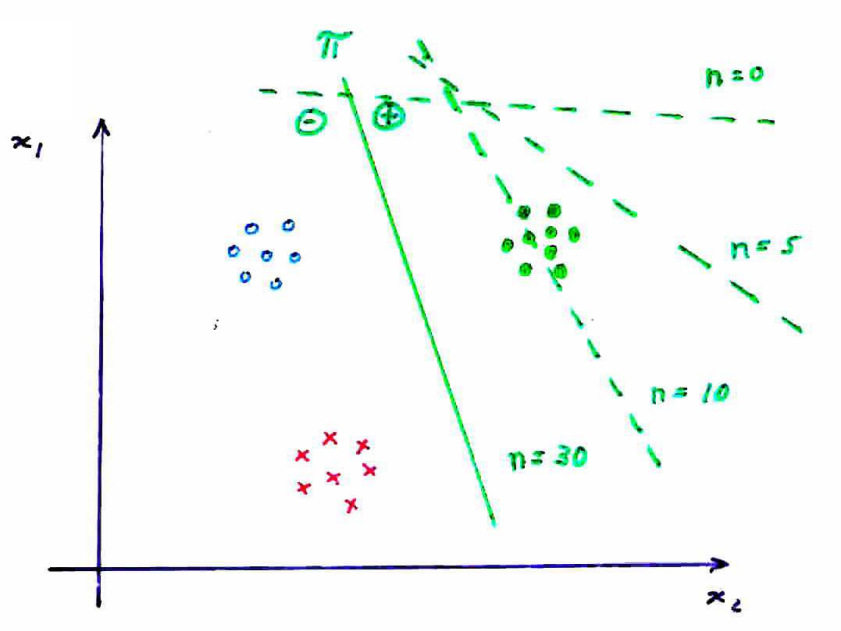
2 camadas - superfícies bissetoras



Treinamento da rede de uma camada

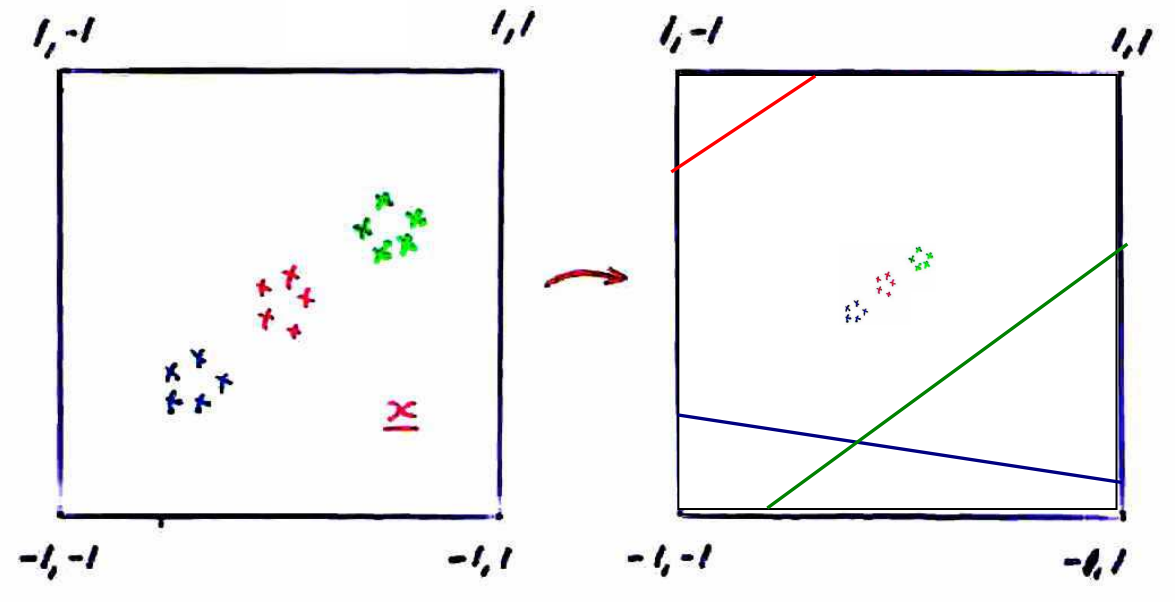
posição inicial

evolução

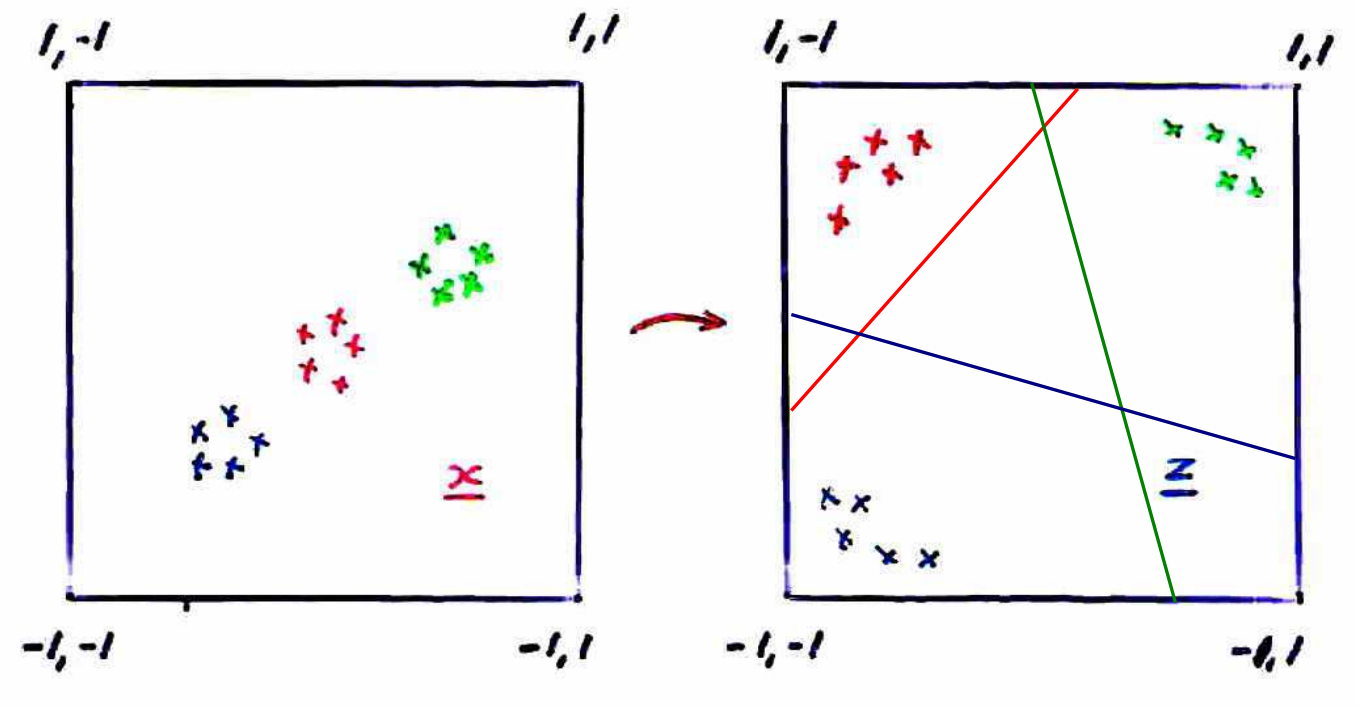


Treinamento da rede de duas camadas

posição inicial

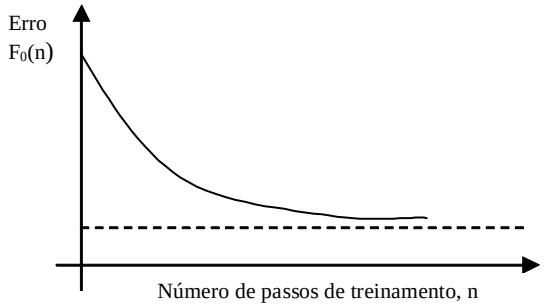


evolução

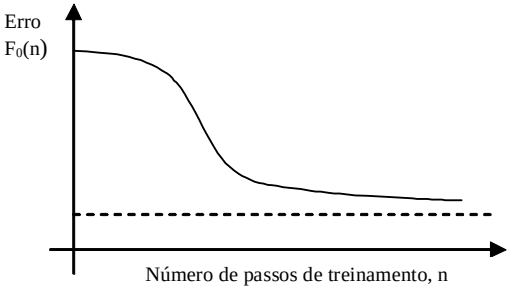


Evoluçao do erro de treinamento

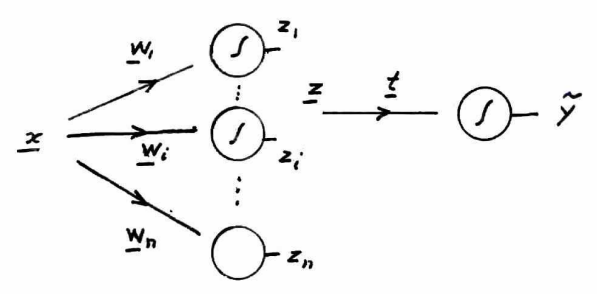
1 camada



2 camadas (classes não LS)



Fim do treinamento
Valores ótimos das sinapses



$$\varepsilon \to 0 \qquad |\tilde{y}_i| \to 1 \qquad |\underline{z}^t \underline{t}| \to \infty$$

$$|\,\underline{\mathbf{t}}\,| \longrightarrow \infty$$

$$\varepsilon \to 0 \qquad |z_i| \to 1\,(\max) \qquad |u_i| = |\underline{x}^t \underline{w}| \to \infty$$

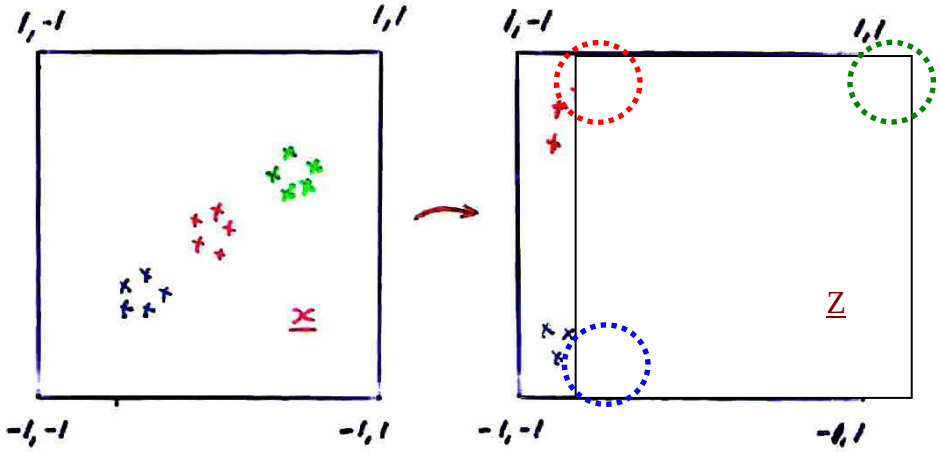
$$|\,\underline{\mathbf{w}}\,| \longrightarrow \infty$$

A rede paraliza no fim do treinamento !

desde que haja um número suficiente de neurônios na camada intermediária

as entradas são mapeadas nos vértices do hipercubo lógico do espaço z

(ou melhor, nas suas proximidades)



Re-interpretação do funcionamento das duas camadas da rede

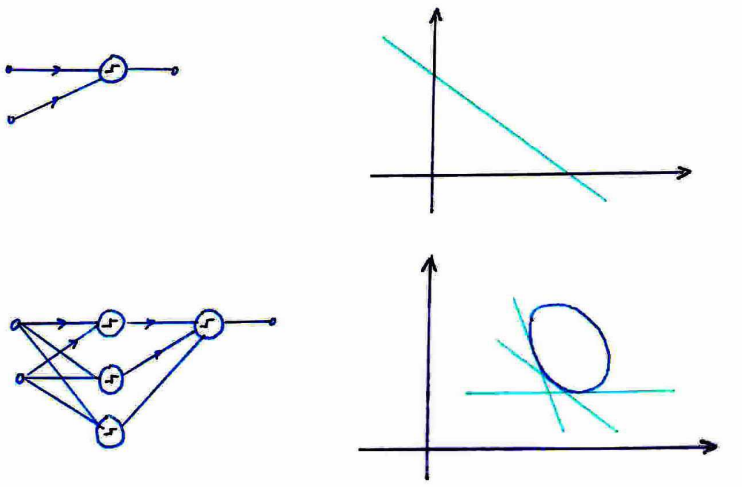
(no caso em que os neurônios saturam)

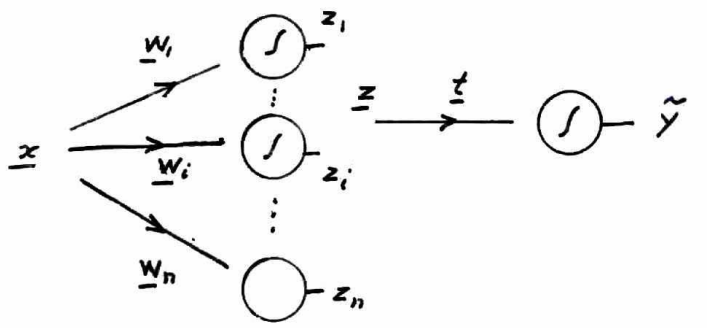
Para qualquer entrada os neurônios operam praticamente saturados

$$|z_i| \cong 1: \quad z_i \cong +1 \;\; ou \;\; z_i \cong -1$$

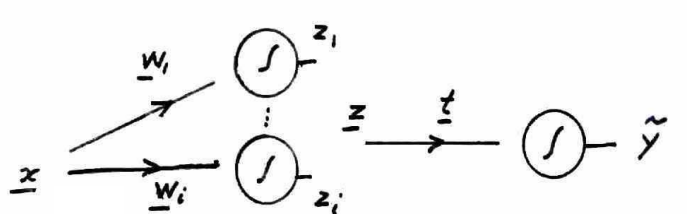
1ª camada: cada neurônio da camada intermediária gera um separador linear no domínio da entrada, definindo semiespaços com saída positiva (z_i = +1) ou negativa (z_i = -1). O vetor $\underline{z}$ mapeia cada região resultante da intersecção destes semiespaços em um vértice do hipercubo lógico no espaço $\underline{z}$.

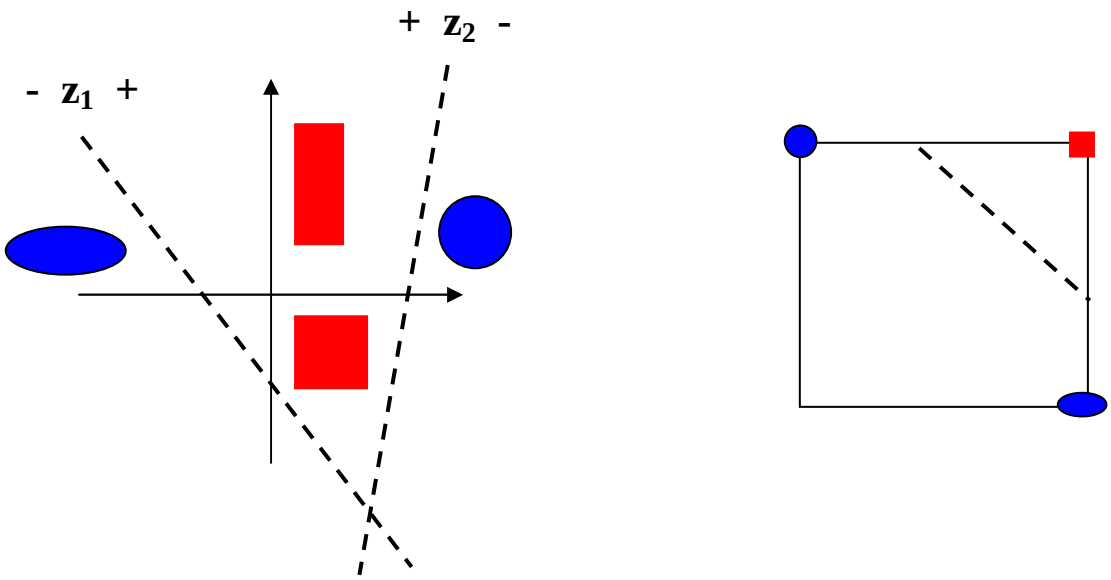
2ª camada: cada neurônio da camada de saída gera um separador linear no espaço $\underline{z}$ que separa alguns vértices deste espaço dos demais. Isto é, separa (agrupa) algumas regiões das demais.



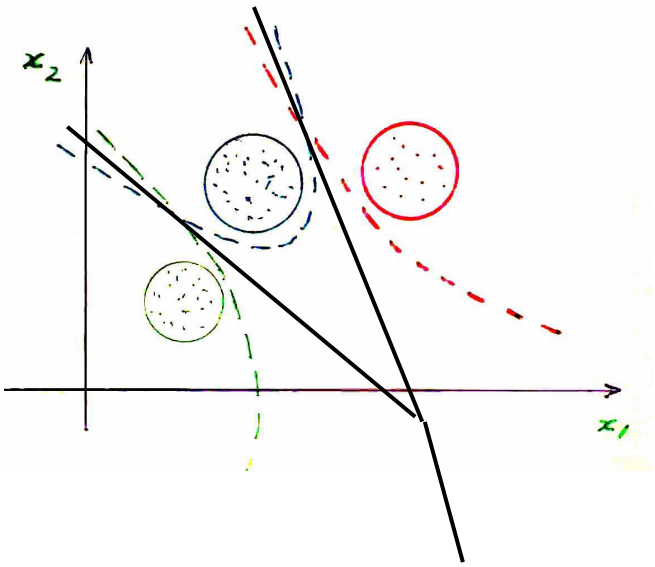


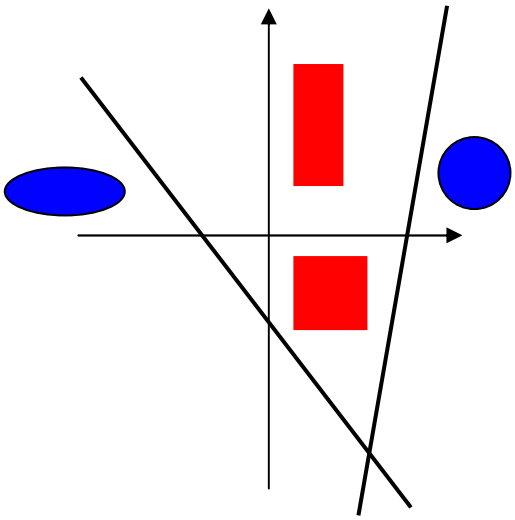
Exemplo: XOR





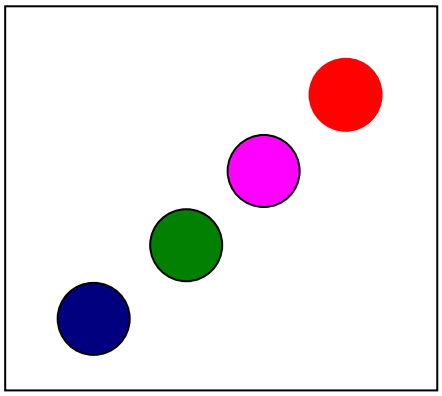
Obs 1: as superfícies separadoras degeneram em superfícies poligonais (o contorno das regiões resultantes da intersecção dos separadores) quando os neurônios operam saturados. Os espaços são politopos.

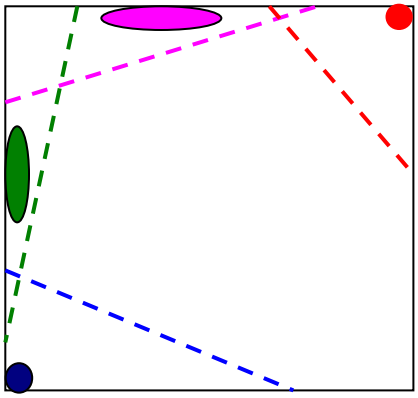




Obs 2: se o número de neurônios na camada intermediária não for suficientemente grande, as regiões de x definidas pela intersecção dos separadores são mapeadas em regiões nas faces do hipercubo lógico em z.

Ex: 2 neurônios na camada intermediária





E, claro, se o número de neurônios na camada intermediária for muito pequeno nem a classificação correta consegue ser realizada.

E as considerações geométricas e estatísticas ?

Valem para a camada de saída !