Universidade Federal do Rio de Janeiro

PROGRAMA DE MESTRADO
IM/DCC - NCE

Os Modelos Perceptron, Adaline e Madaline

www.labic.nce.ufrj.br

Antonio G. Thomé
thome@nce.ufrj.br
Sala - 1013
(021)2598-3268

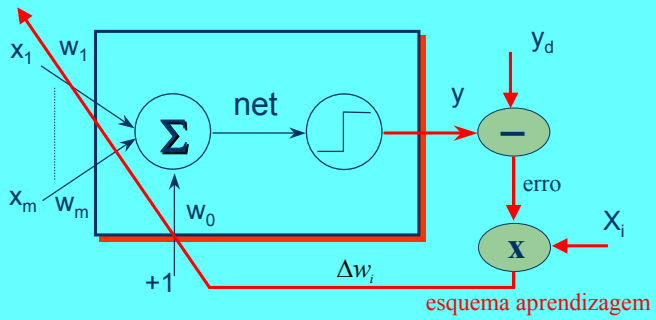
O Modelo PERCEPTRON

Rosemblat - 1962

PERCEPTRON

- objetivo:
  - Atuar como classificador e como gerador de funções lógicas binárias
- características
  - aprendizado supervisionado
  - representação binária
  - apenas uma camada de pesos ajustáveis

Perceptron - Arquitetura Básica



Ativação

$$net = \sum_{i=0}^{m} w_i x_i \,/\, w_i, x_i \in \mathcal{R}$$

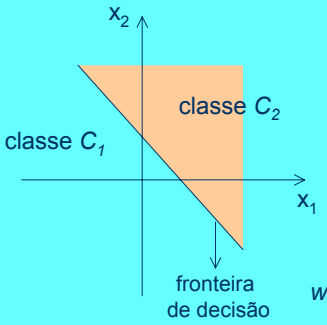
Propagação - degrau

$$y = \begin{cases} 1 & se \quad net > 0 \\ 0 & se \quad net \leq 0 \end{cases}$$

onde: w_0 é o limiar *(threshold)* - também aprendido pela rede

Fronteira de Decisão

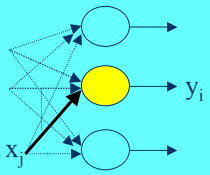
- atribui o ponto representado pelas entradas x_1, x_2, ..., x_n à classe C_1 se a saída y do Perceptron for igual a *1* ou à classe C_2 se ela for *0*



$w_1x_1 + w_2x_2 + b = 0$

Demo: nnd4db
Decision bounderies

Regra de Aprendizado do Perceptron



$$w_{ij}(k+1) = w_{ij}(k) + \lambda * e_i(k) * x_j$$

Onde:

- $e_i(k) = yd_i - y_i$
- λ – taxa de aprendizado

Demo: nnd4pr
Perceptron Learning Rule

Treinamento do Perceptron

- algoritmo

1 – inicializar λ e o vetor de pesos W

2 – repetir

3 – para cada par do conjunto de treinamento $\Gamma = \{(x^i, y_d^i)\}_{i=1}^{p}$

3.1 – atualizar o vetor de pesos para cada um dos nodos da rede segundo a regra

$$W(k+1) = W(k) + \lambda \cdot e \cdot X(k)$$

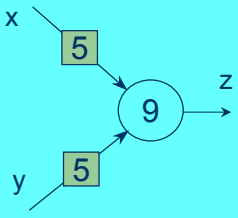
4 – até $e = 0$ para todos os p elementos do conjunto de treinamento em todos os neurônios da rede

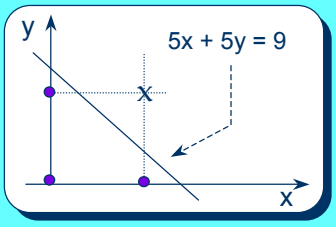
Demo:
Classif w/ 2 input neurons

Separabilidade Linear

• Porta "E"

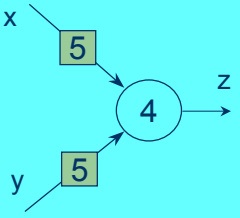
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

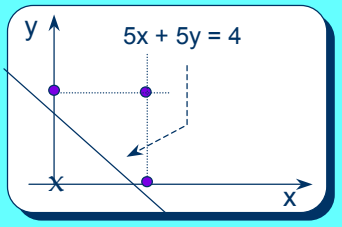




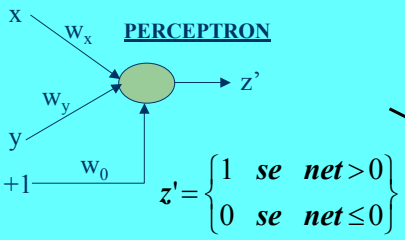
• Porta "OU"

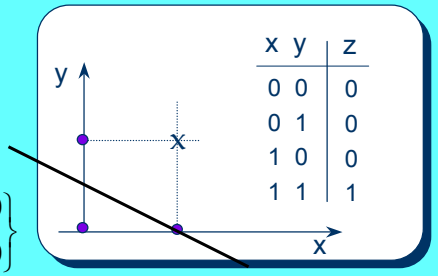
x	y	z
0	0	0
0	1	1
1	0	1
1	1	1





• Porta "E"



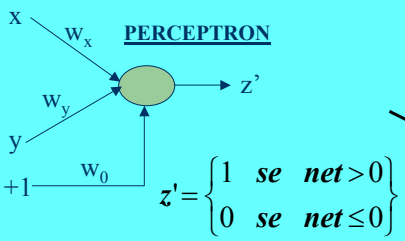


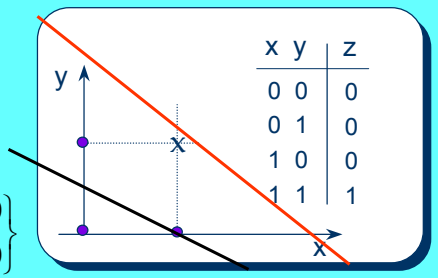
- Passo 1 – iniciar w_i e λ (taxa de aprendizado)
  - $w_x = +1$; $w_y = +2$; $w_0 = -1$; $\lambda = 1$

x+2y-1=0

- Passo 2 – apresenta primeiro padrão de amostra $\{x_1,y_1\} \equiv \{0,0\}$
  - net(1)= $w_x * x_1 + w_y * y_1 + w_0 * 1 = -1 \Rightarrow z'(1) = 0$ -- correto

• Porta "E"

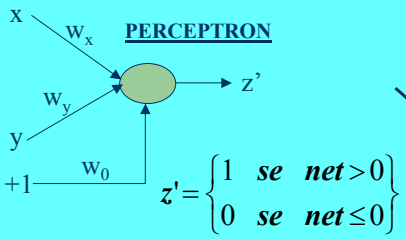




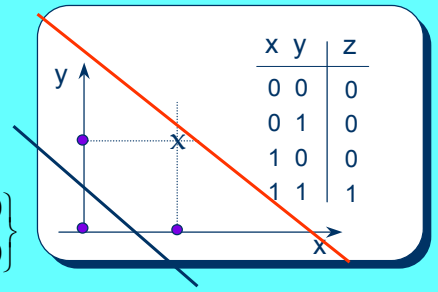
- Passo 3 – apresenta segundo padrão de amostra $\{x_2,y_2\} \equiv \{0,1\}$
  - net(2)= $1*0 + 2*1 + -1*1 = +1 \Rightarrow z'(2) = 1$ – incorreto
  - **adapta os pesos:**
    - $w_x(3) = w_x(2) + 1*(z(2)-z'(2))*x(2) = 1 + (-1)*0 = 1$
    - $w_y(3) = w_y(2) + 1*(z(2)-z'(2))*y(2) = 2 + (-1)*1 = 1$
    - $w_0(3) = w_0(2) + 1*(z(2)-z'(2))*1 = -1 + (-1)*1 = -2$

x+y-2=0

• Porta "E"



$$z' = \begin{cases} 1 & se \quad net > 0 \\ 0 & se \quad net \le 0 \end{cases}$$



• Passo 3 – apresenta terceiro padrão de amostra $\{x_3,y_3\} \equiv \{1,0\}$

• net(3)= 1*1 + 1*0 + -2* 1 = -1 $\Rightarrow$ z'(3) = 0 – correto

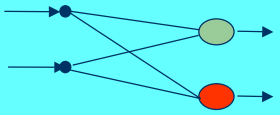
• Passo 4 – apresenta quarto padrão de amostra $\{x_4,y_4\} \equiv \{1,1\}$

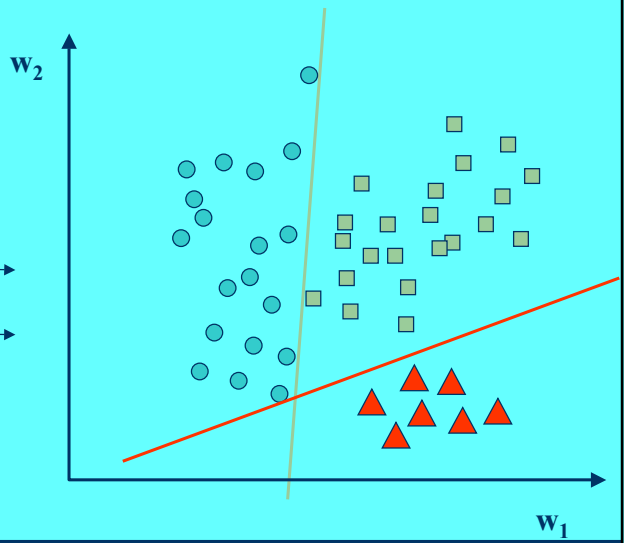
• net(4)= 1*1 + 1*1 + -2* 1 = 0 $\Rightarrow$ z'(4) = 0 – incorreto

• $w_x(4) = 1 + (1)*1 = 2$; $w_y(4) = 1 + (1)*1 = 2$; $w_0(4) = -2 + (1)*1 = -1$

2x+2y-1=0

PERCEPTRON – Capacidade de Classificação

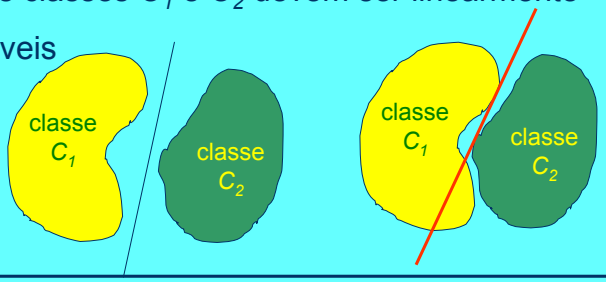




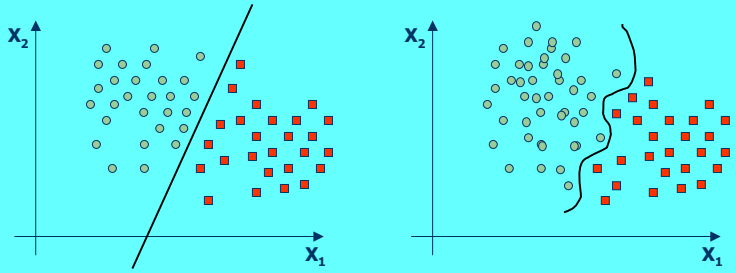
Demo:
Linearly non separable vectors

Limitações do Perceptron

- não admite mais de uma camada de pesos ajustáveis
- aprendizado nem sempre ocorre
- as duas classes C_1 e C_2 devem ser linearmente separáveis



Problemas linearmente separáveis são aqueles que podem ser resolvidos utilizando uma reta ou hiperplano como fronteira de decisão.

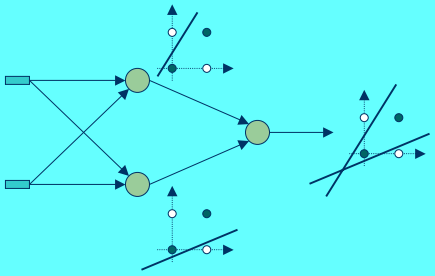


Realidade

- Classificação perfeita quase nunca é possível
- Nem todos os problemas são linearmente separáveis
- Difícil validar qual o melhor número de categorias para um determinado problema
- Grande sobreposição de classes (???)
- Causas freqüentes de erros:
  - Padrões não podem ser linearmente separados
  - Características podem ser inadequadas para distinguir as diferentes classes
  - Características podem ser muito correlacionadas
  - Podem haver subclasses distintas nos dados

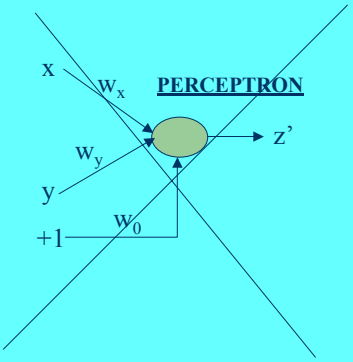
PERCEPTRON – 01 camada & linearmente separáveis

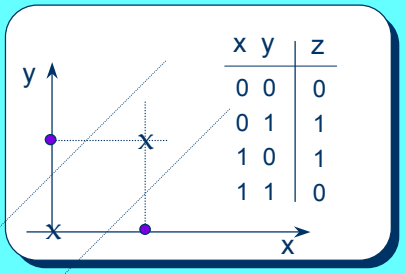
?

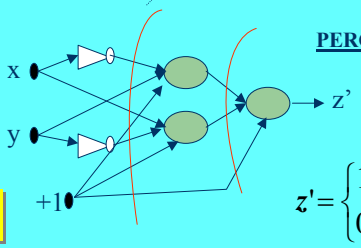


- Combinar Camadas
  - Poderia ser uma solução?
- Problema
  - Como treinar as camadas mais internas?

• Porta "XOR"







$$x \oplus y = \bar{x} \bullet y + x \bullet \bar{y}$$

$$z' = \begin{cases} 1 & se \quad net > 0 \\ 0 & se \quad net \le 0 \end{cases}$$

Exemplo

- considere um conjunto de pessoas (4 elementos) formado por homens e mulheres. Treinar uma rede Perceptron, que seja capaz de reconhecer o sexo das pessoas.

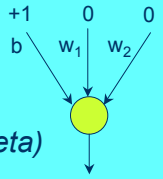
nome	feminino	masculino	codificação
José		X	0 0
Paulo		X	0 1
Maria	X		1 0
Lúcia	X		1 1

Exemplo (cont.)

- condições de disparo
  - se $net > 0$ $y = 1$ (feminino)
  - se $net \leq 0$ $y = 0$ (masculino)
- início do treinamento
  - $b = 0$
  - $w_1 = 0$
  - $w_2 = 0$

Exemplo (cont.)

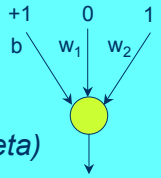
- apresentação do primeiro elemento à rede neural
  - $net = b * 1 + w_1 * 0 + w_2 * 0$
  - $net = 0 * 1 + 0 * 0 + 0 * 0 = 0$
  - $y = 0$ e $d = 0$ *(resposta correta)*



Exemplo (cont.)

- apresentação do segundo elemento à rede neural

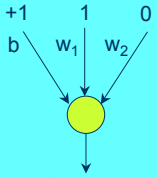
- $net = b * 1 + w_1 * 0 + w_2 * 1$
- $net = 0 * 1 + 0 * 0 + 0 * 1 = 0$
- $y = 0$ e $d = 0$ *(resposta correta)*



Exemplo (cont.)

- apresentação do terceiro elemento à rede neural

- $net = b * 1 + w_1 * 1 + w_2 * 0$
- $net = 0 * 1 + 0 * 1 + 0 * 0 = 0$
- $y = 0$ e $d = 1$ *(resposta incorreta)*



- $b = b + 1 = 0 + 1 = 1$
- $w_1 = w_1 + 1 = 0 + 1 = 1$
- $w_2 = w_2 + 0 = 0 + 0 = 0$

$net = b * 1 + w_1 * 1 + w_2 * 0$

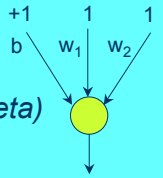
$net = 1 * 1 + 1 * 1 + 0 * 0 = 2$

$y = 1$ e $d = 1$ *(resposta correta)*

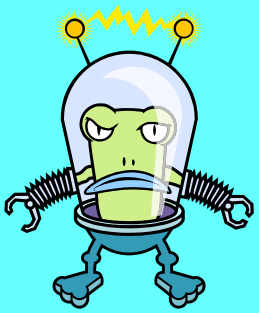
Exemplo (cont.)

- apresentação do quarto elemento à rede neural

  - $net = b * 1 + w_1 * 1 + w_2 * 1$
  - $net = 1 * 1 + 1 * 1 + 0 * 1 = 2$
  - $y = 1$ e $d = 1$ (resposta correta)



Modelo ADALINE



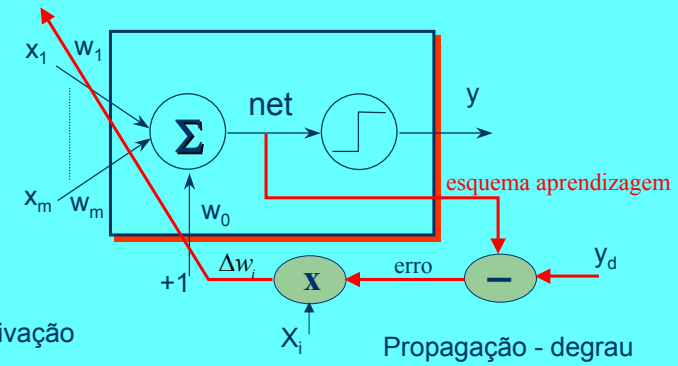
Widrow & Hoff - 1963

ADALINE

- características
  - ADALINE e Perceptron surgiram quase que simultaneamente
  - tem seus pesos ajustados em função do erro de sua saída linear, antes da aplicação da função de propagação
  - continua com o problema de uma única camada de pesos ajustáveis
  - a função de custo a ser minimizada é quadrática nos pesos de entrada, e a minimização é feita pelo método do gradiente

Demo:
Training w/ Linear neuron

ADALINE - Arquitetura Básica



Ativação

$$net = \sum_{i=0}^{m} w_i x_i \,/\, w_i, x_i \in \mathcal{R}$$

Propagação - degrau

$$y = \begin{cases} 1 & se \quad net > 0 \\ 0 & se \quad net \le 0 \end{cases}$$

onde: w_0 é o limiar *(threshold)* - também aprendido pela rede

Esquema de Treinamento - Regra Delta

- atualização dos pesos pelo método do gradiente
  - o ajuste ocorre quando $e \neq 0 \Rightarrow \Delta W \neq 0$
- função de erro a ser minimizada

$$E(k) = \frac{e(k)^2}{2} = \frac{\left(y_d(k) - net(k)\right)^2}{2}$$

- objetivo
  - obter a direção do ajuste a ser aplicado ao vetor de pesos de forma a caminhar de forma incremental em direção à solução ótima

Dedução da Regra Delta (dedução)

- o ajuste é feito na direção contrária do gradiente

$$\Delta W_i(k+1) \approx -\lambda \times \nabla E(k)$$

$$\nabla E(k) = \frac{\partial E(k)}{\partial w_i(k)} = \frac{\partial E(k)}{\partial net_i(k)} \frac{\partial net_i(k)}{\partial w_i(k)} / (todo\ i)$$

$$\frac{\partial E(k)}{\partial net_i(k)} = \frac{\partial (y_d(k) - net_i(k))^2}{2 \partial net_i(k)} = -\frac{2(y_d(k) - net_i(k))}{2} = -e(k)$$

$$\frac{\partial net(k)}{\partial w_i(k)} = \frac{\partial (\sum w_j x_j)}{\partial w_i(k)} = x_i(k)$$

$$\Delta w_i(k+1) \approx -\lambda(-e) x_i(k) = \lambda e(k) x_i(k)$$

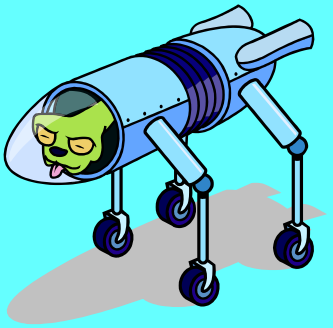
Demo: nnd4pr
Linear Class System

Vantagens e Limitações do ADALINE

- Vantagens:
  - Treinamento mais suave que o do Perceptron
- Limitações:
  - As mesmas do Perceptron
  - O treinamento inclusive pode ser muito mais demorado

(*) muito empregado na área de filtros digitais adaptativos

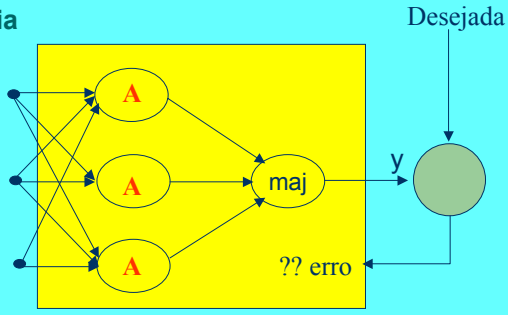
Modelo - MADALINE



Madaline

- uma das primeiras redes neurais em camadas treináveis com múltiplos elementos adaptativos
- uma camada composta por ADALINE´s é conectada a um ADALINE com parâmetros fixos
- critérios de decisão: “OU” “E” “MAIORIA”

Exemplo - Maioria



- Passo 1 – um padrão é apresentado a cada ADALINE
- Passo 2 – a saída y é calculada em concordância com a maioria
- Passo 3 – a saída gerada é comparada com a desejada
- Passo 4 – se errada, o ADALINE com menor erro é escolhido para treinar