LECTURE 03 - PERCEPTRONS

NEURAL

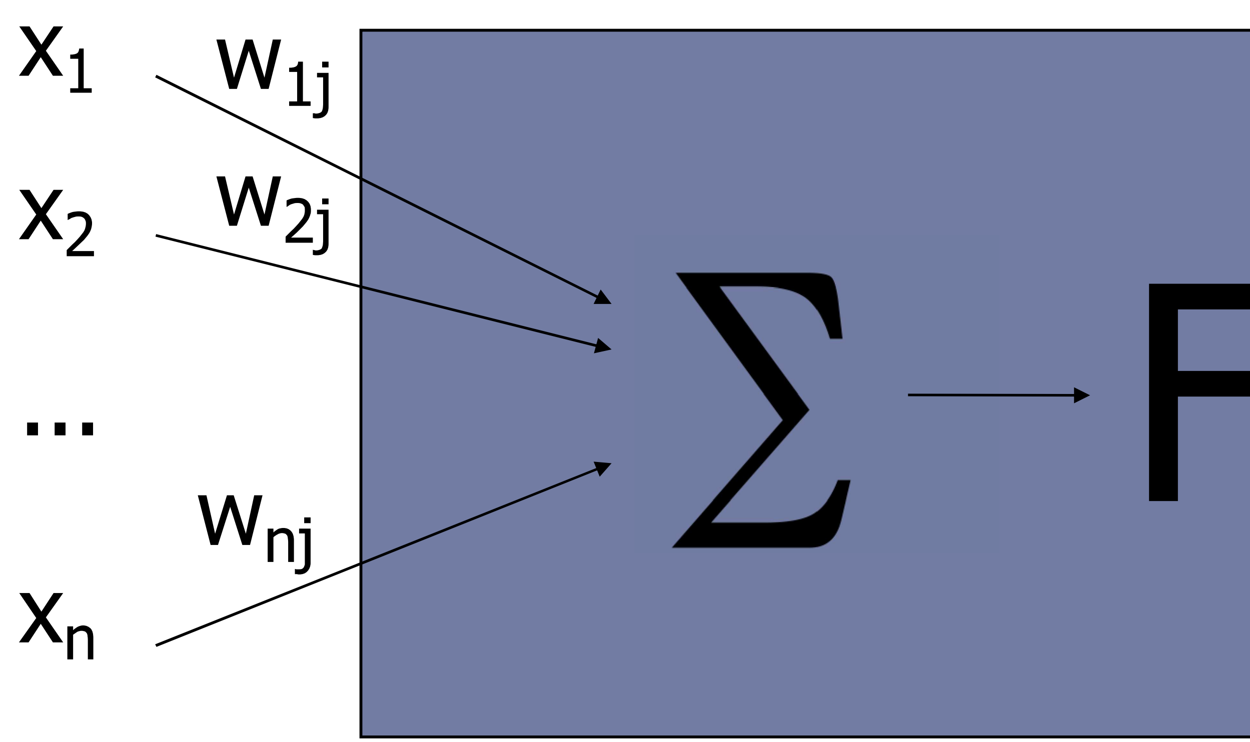
AGENDA

1. Introdução
2. Portas de limiar
3. Perceptron
4. Adaline

INTRODUÇÃO

1. O Neurônio MCP: na sua forma mais simples, em:

Padrão de entrada
vetor **x**

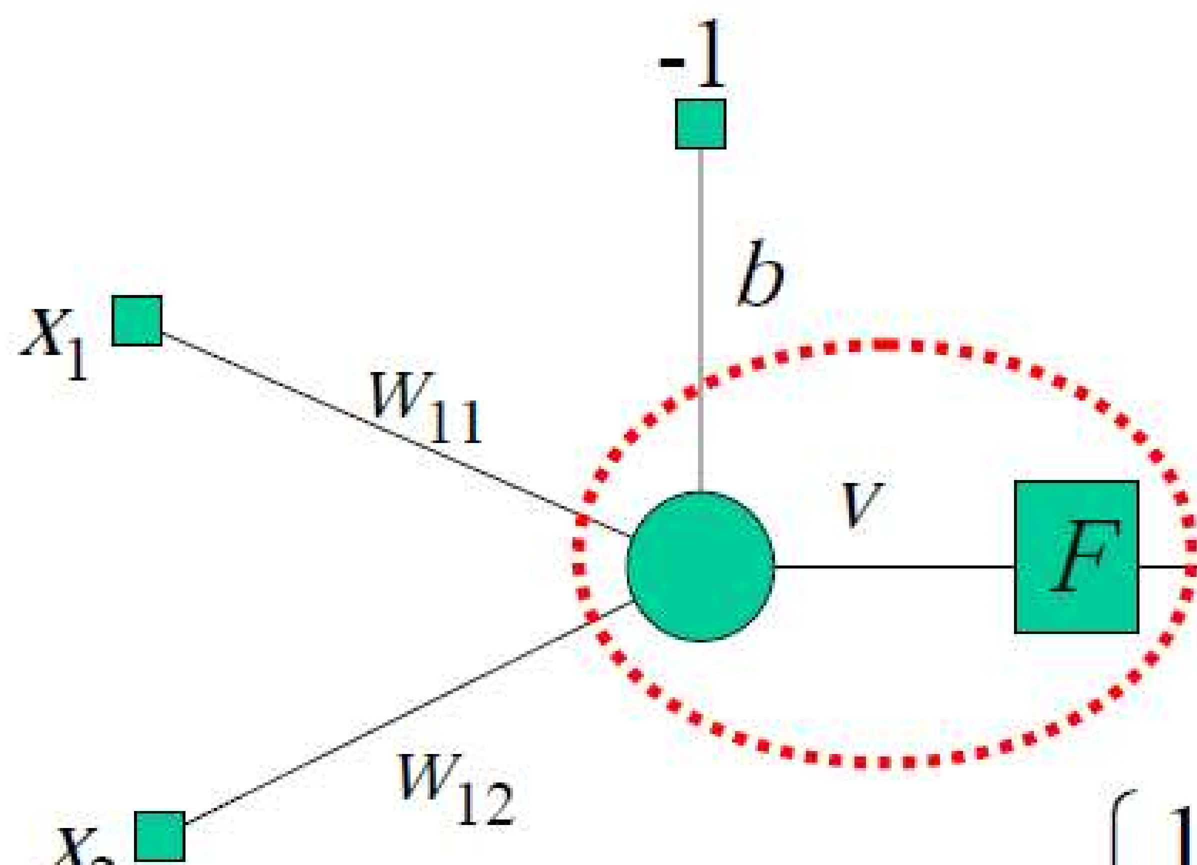


INTRODUÇÃO

1. O Neurônio MCP resolve o problema da porta A

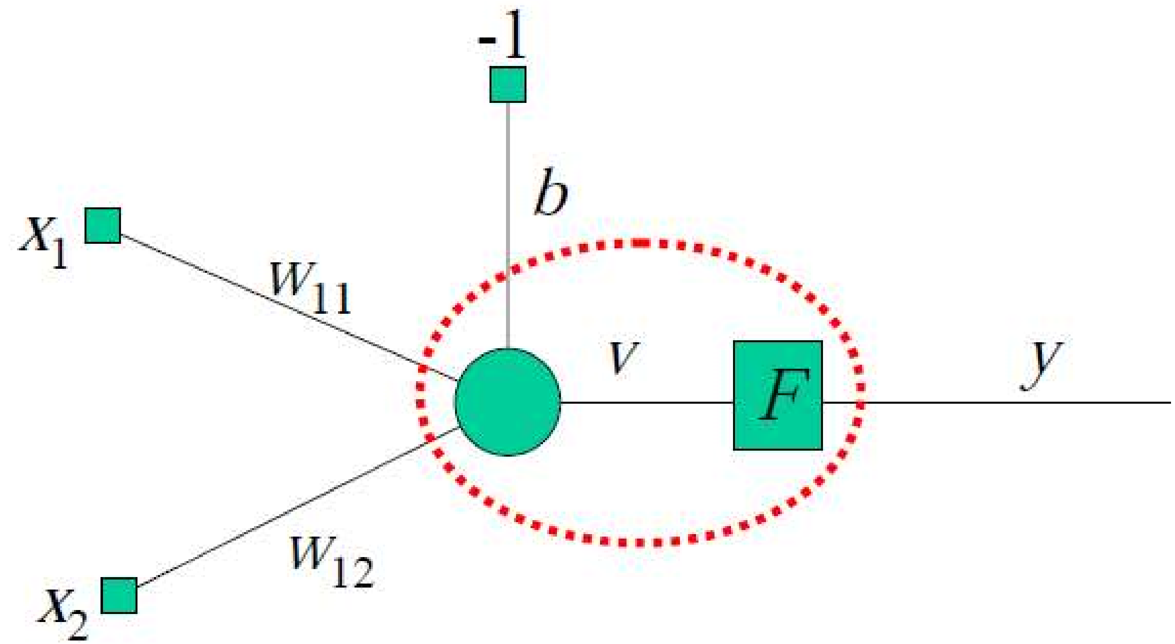
AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



AND

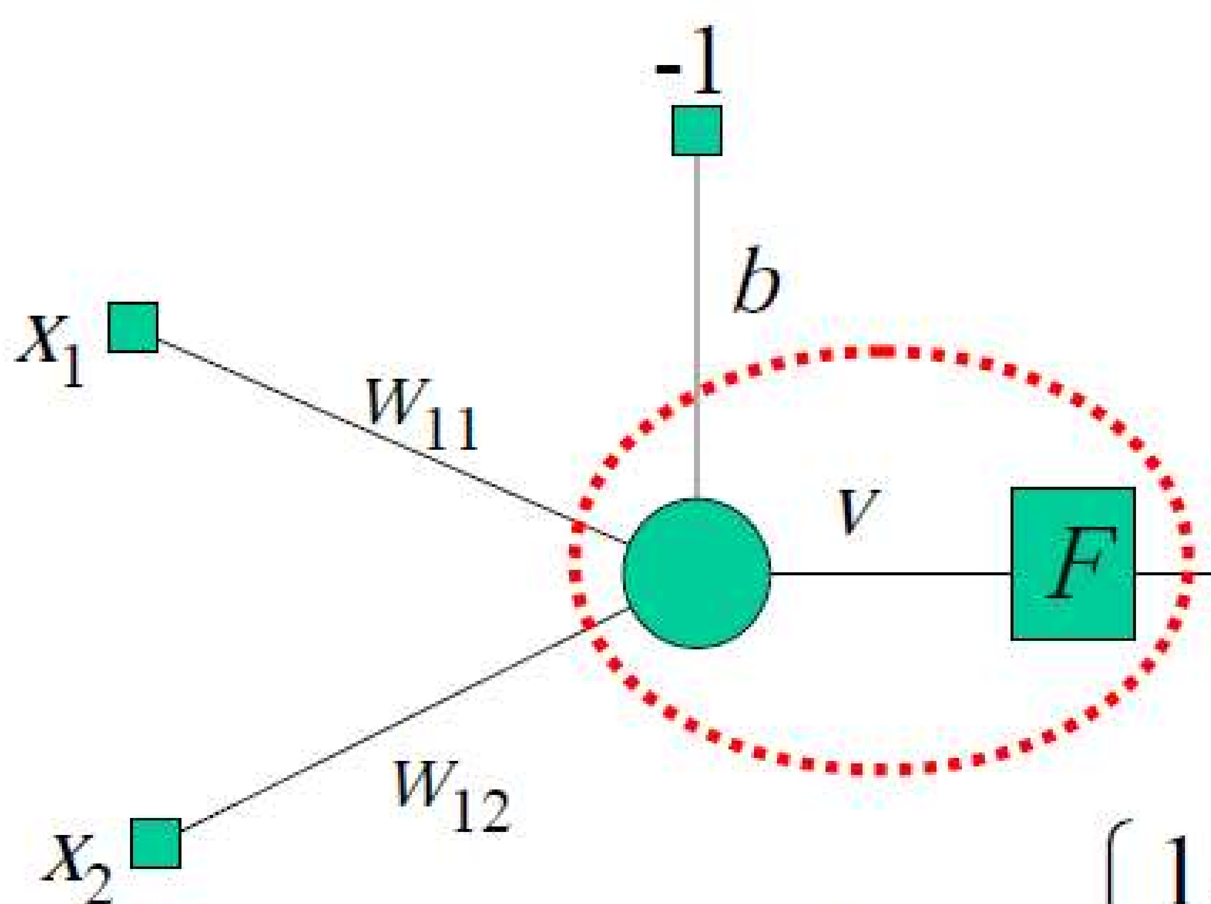
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



INTRODUÇÃO

1. E o problema do OR, como ficaria?

OR		
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



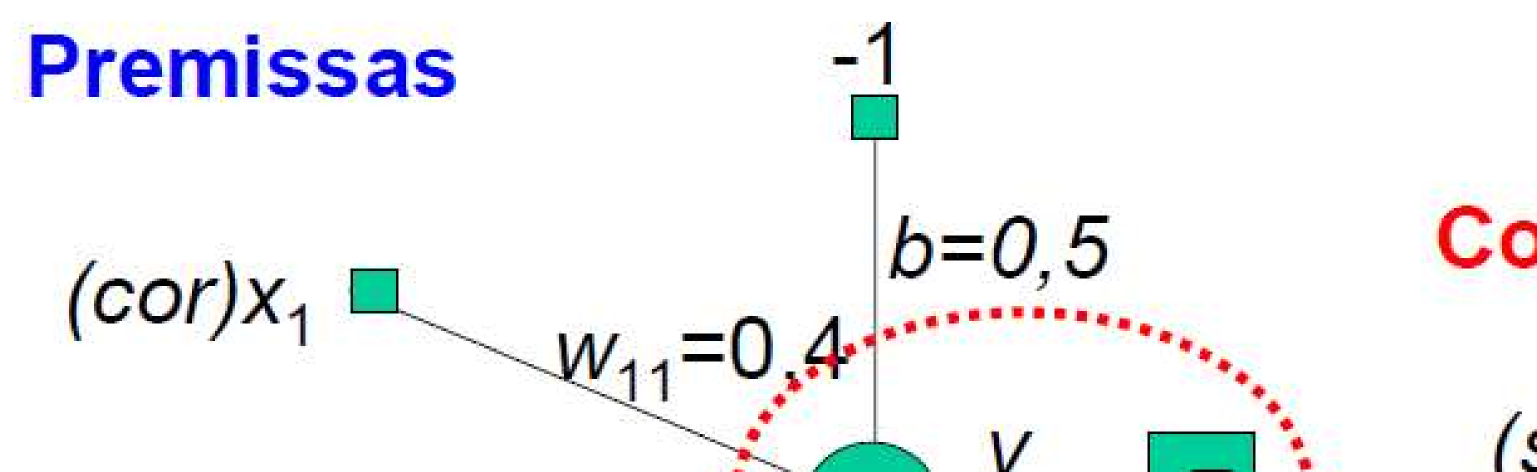
INTRODUÇÃO: EXEMPLOS

IF cor=vermelha AND forma=redonda THEN fruta=

IF cor=vermelha AND forma=ovalada THEN fruta=

IF cor=amarela AND forma=redonda THEN fruta=

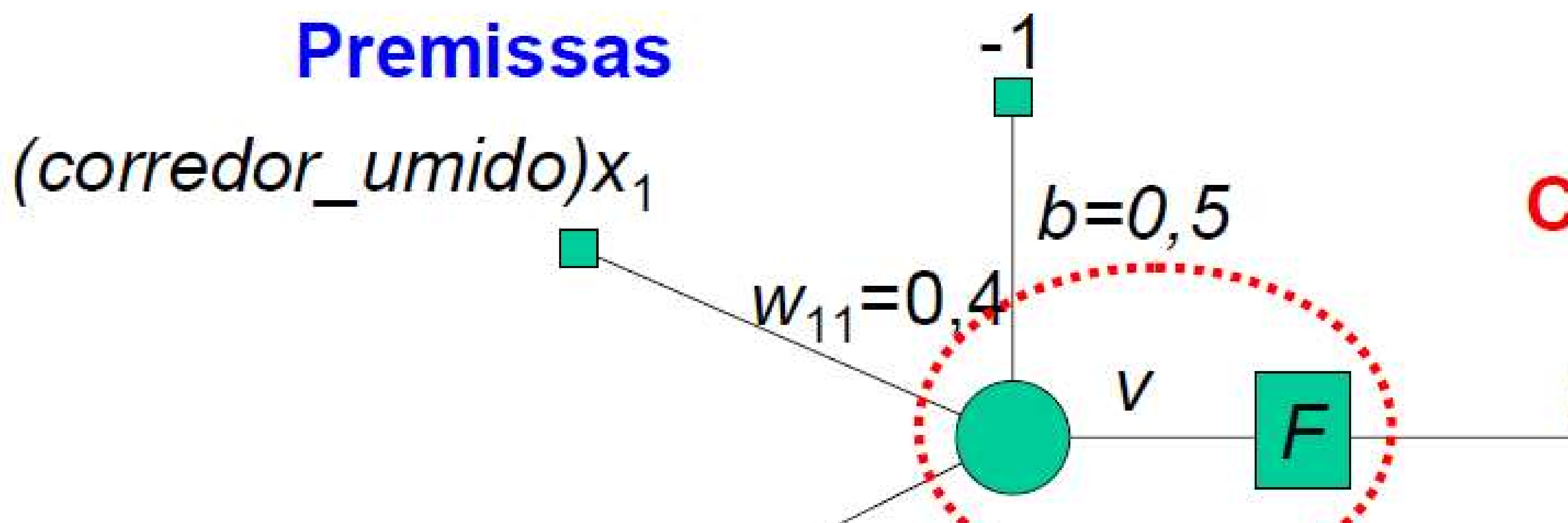
IF cor=amarela AND forma=ovalada THEN fruta=l



INTRODUÇÃO: EXEMPLOS

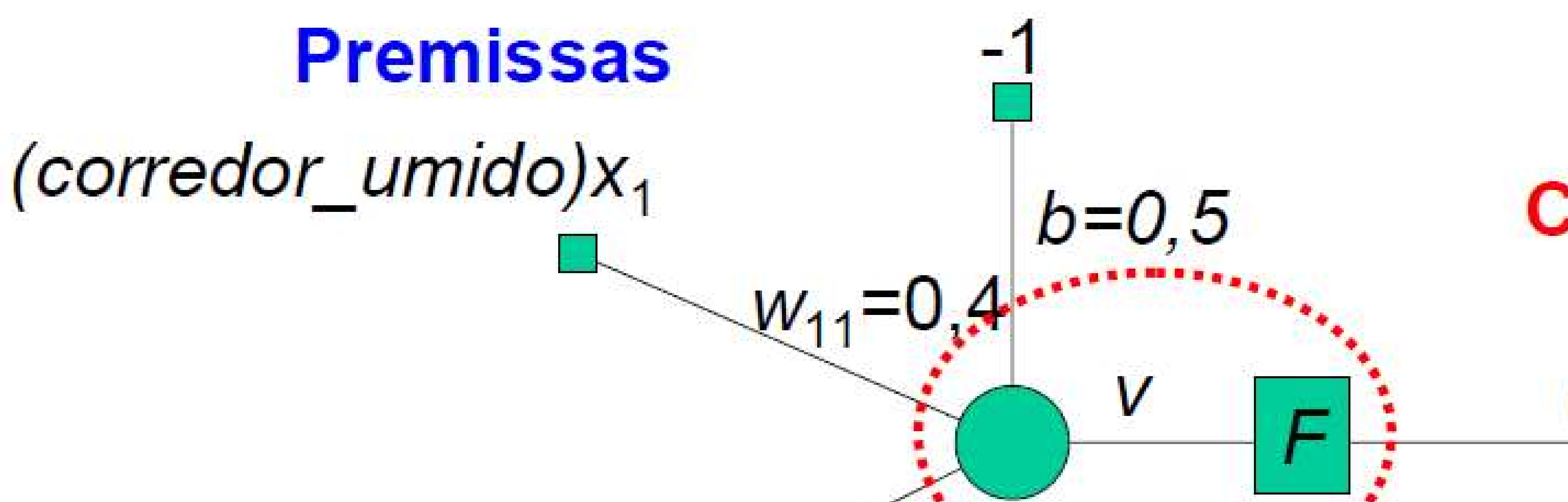
se corredor_umido e banheiro_seco
entao problema_cozinha.

Premissas



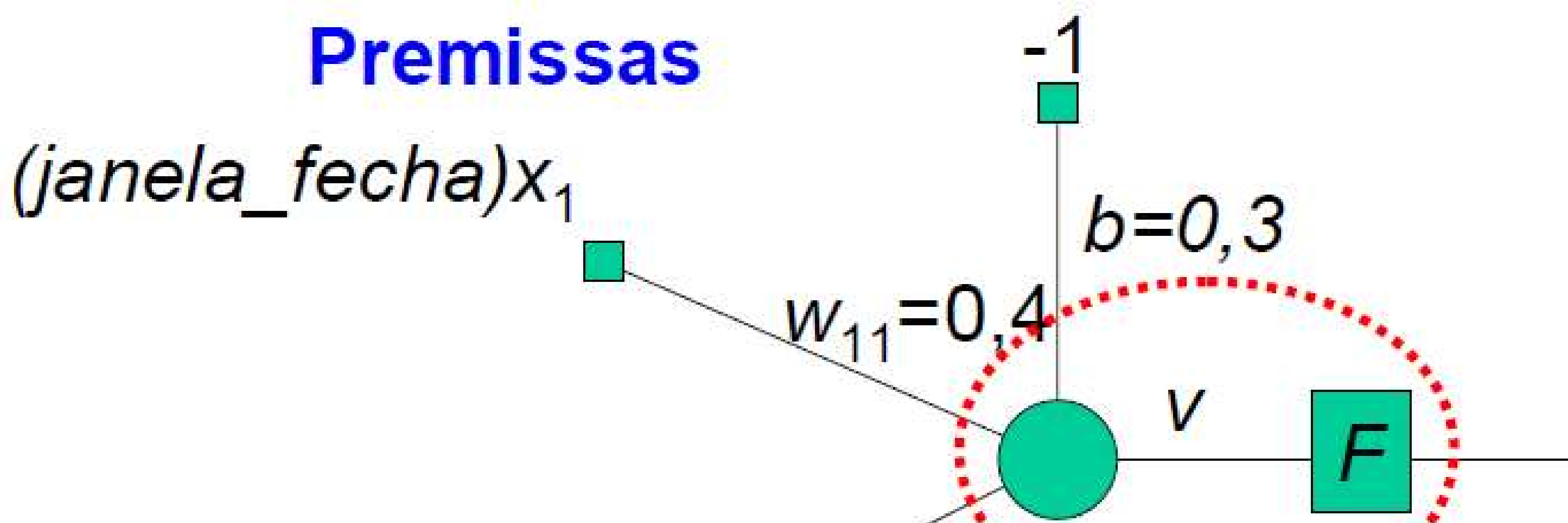
INTRODUÇÃO: EXEMPLOS

se corredor_umido e cozinha_seca
entao vazamento_banheiro.



INTRODUÇÃO: EXEMPLOS

se janela_fechada ou nao_chove
entao nao_entrou_agua_de_fora.

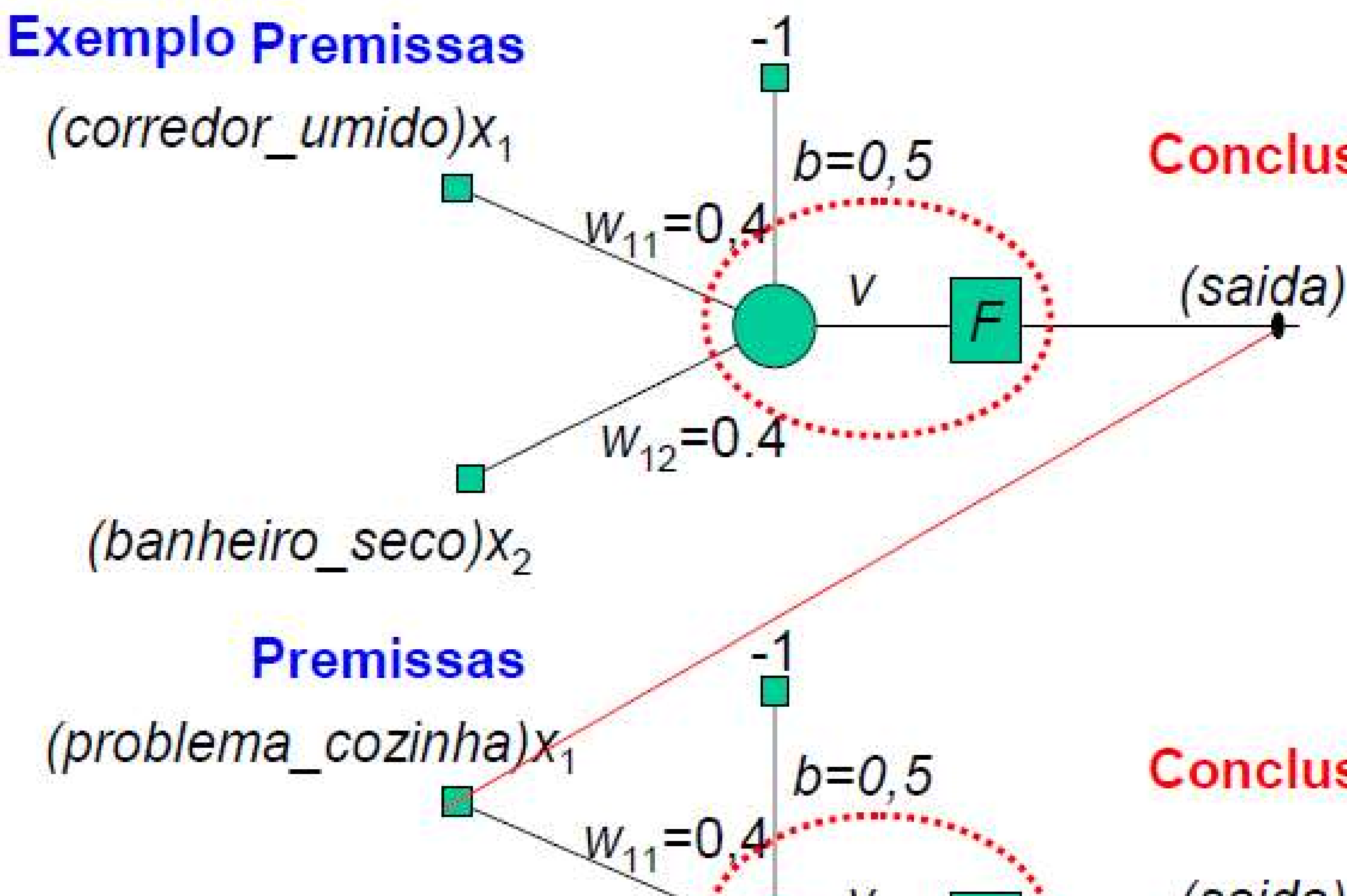


INTRODUÇÃO: EXEMPLOS

```
se      corredor_umido e banheiro_seco
entao   problema_cozinha : 1.

se      problema_cozinha e nao_entrou_agu
entao   vazamento_cozinha : 0.9.
```

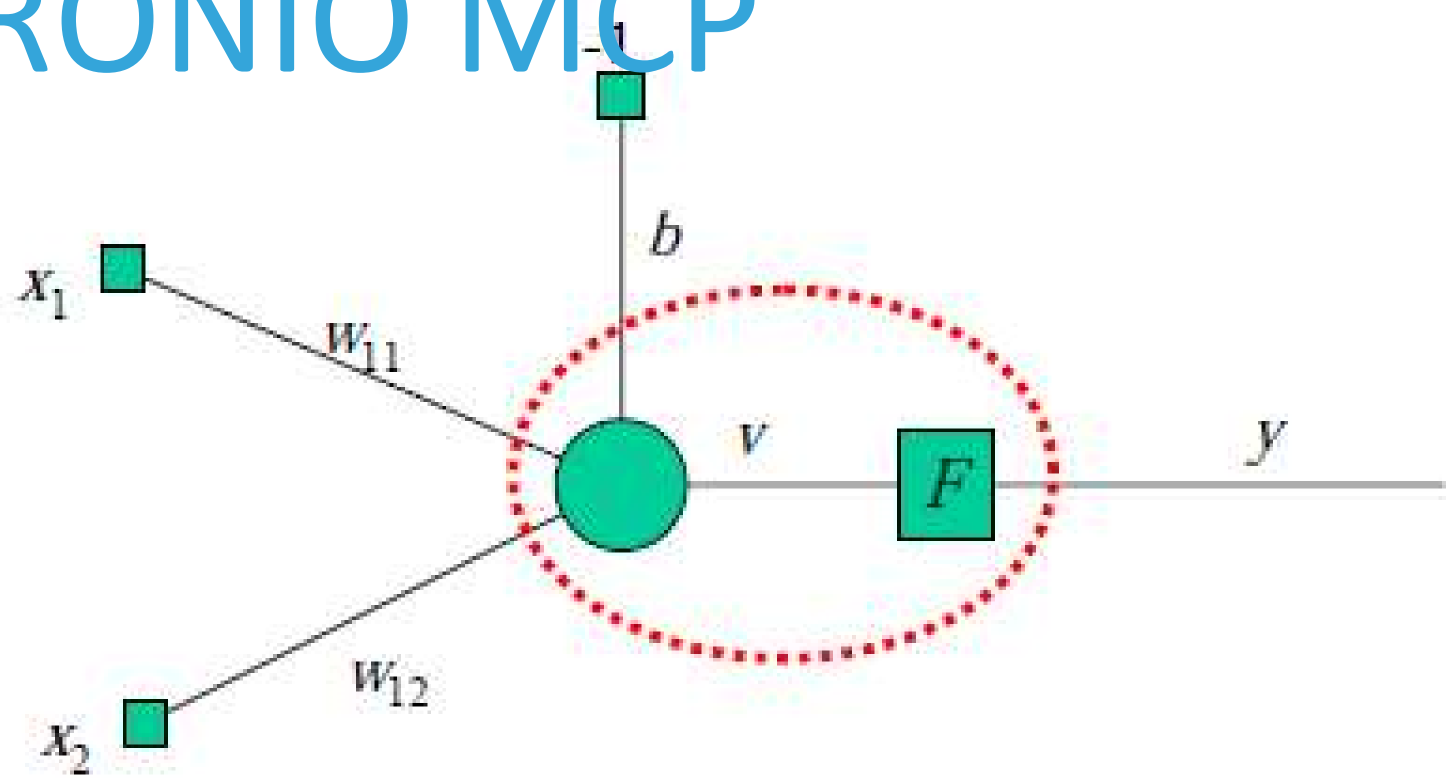
INTRODUÇÃO: EXEMPLOS

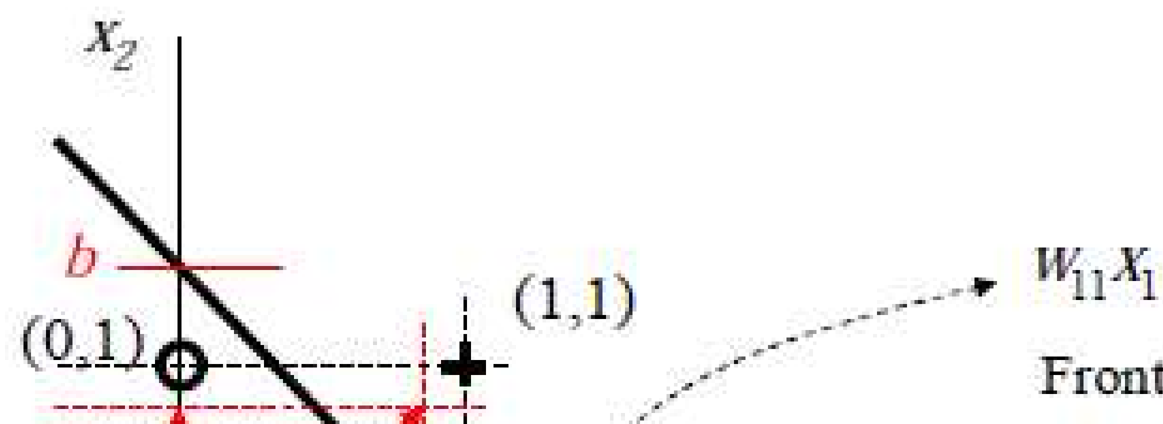


NEURÔNIO MCP

1. O que é o BIAS?
2. O que o neurônio MCP representa?

NEURÔNIO MCP





PORTAS DE LIMIAR

- As portas de limiar podem ser divididas em dois principais:

1. Portas de Limiar Lineares
    1. O próprio modelo MCP
    2. Capacidade limitada: funções linearmente separáv
2. Portas de Limiar Polinomial, etc.

PERCEPTRON

1. O perceptron é uma rede neural simples
2. Utilizada para classificação de padrões linearmen separáveis
3. É formado por neurônios do tipo MCP com pesos ajustáveis e um bias (portas de limiar lineares)
4. Qual a diferença entre o perceptron e o neurônio

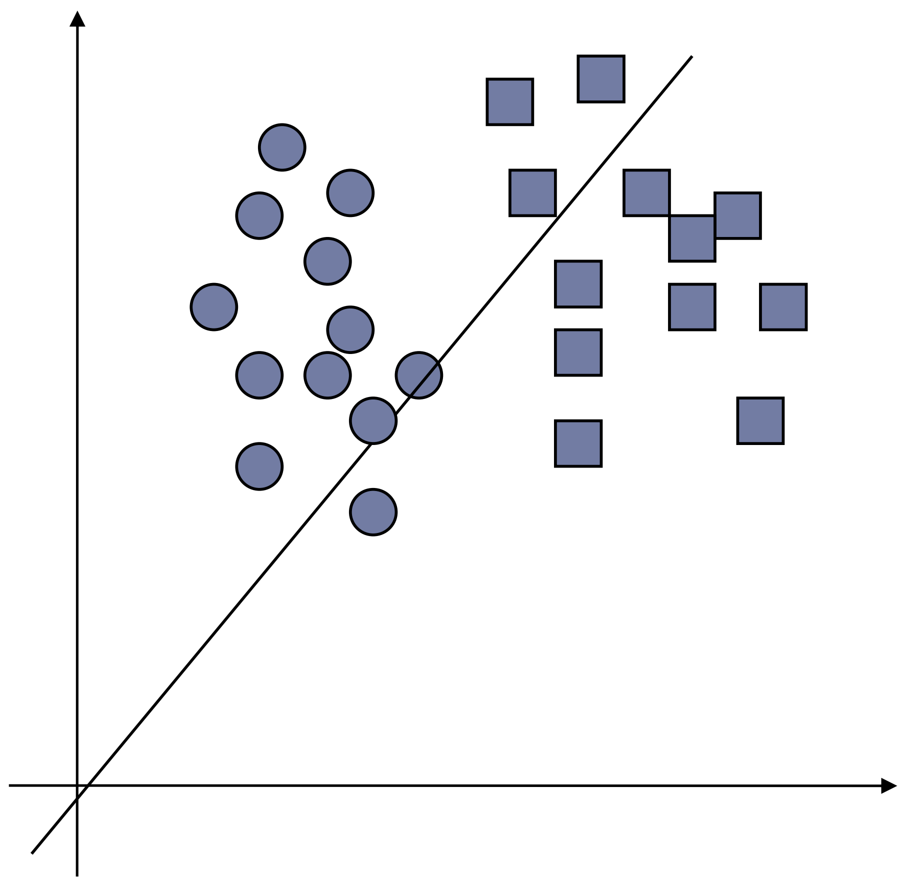
CARACTERÍSTICAS BÁSICAS

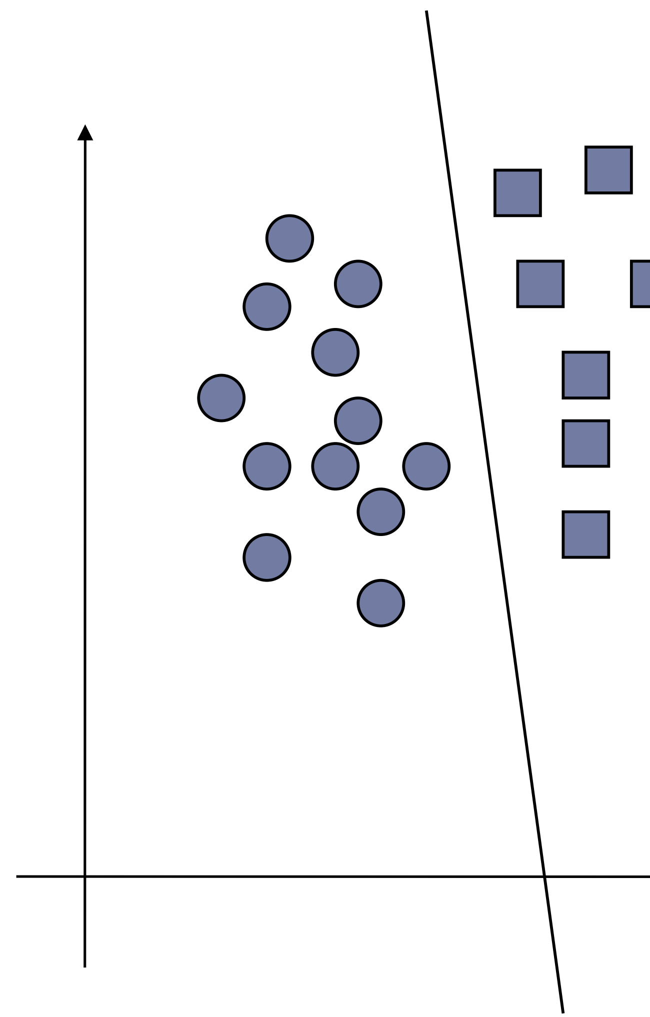
1. Regra de propagação $v_j = \sum_i x_i w_{ij} +$
2. Função de ativação: Degrau
3. Topologia: uma única camada de processadores original é composto por várias camadas, mas ap ajustável)
4. Algoritmo de Aprendizado (supervisionado):

ALGORITMO DE APRENDIZAGE

1. IMPORTANTE
    1. não ocorre variação no peso se a saída e
correta
    2. caso contrario, cada peso é incrementac
a saída é menor que o alvo decrementad
a saída é maior que o alvo

PERCEPTRON - REVISITANDO O





$\sum x_i w_i = 0$ Define um hiperplano passand

PERCEPTRON – ADAPTAÇÃO DO

1. De uma forma geral, o processo de adapta consiste na obtenção do (de)incremento Δ aplicado ao vetor de pesos w

2. Com o conjunto de pesos w(n+1), a rede c apresentar um sinal mais próximo a saída do que a saída produzida utilizando w(n)

ALGORITMO DE TREINAMENTO

1. Iniciar os pesos sinápticos com valores aleatórios
   ou iguais a zero;
2. Aplicar um padrão com seu respectivo valor desej
   (t_j) e verificar a saída da rede (s_j);
3. Calcula o erro na saída $e_j = t_j - s_j$;
4. Se $e_j = 0$, volta ao passo 2;

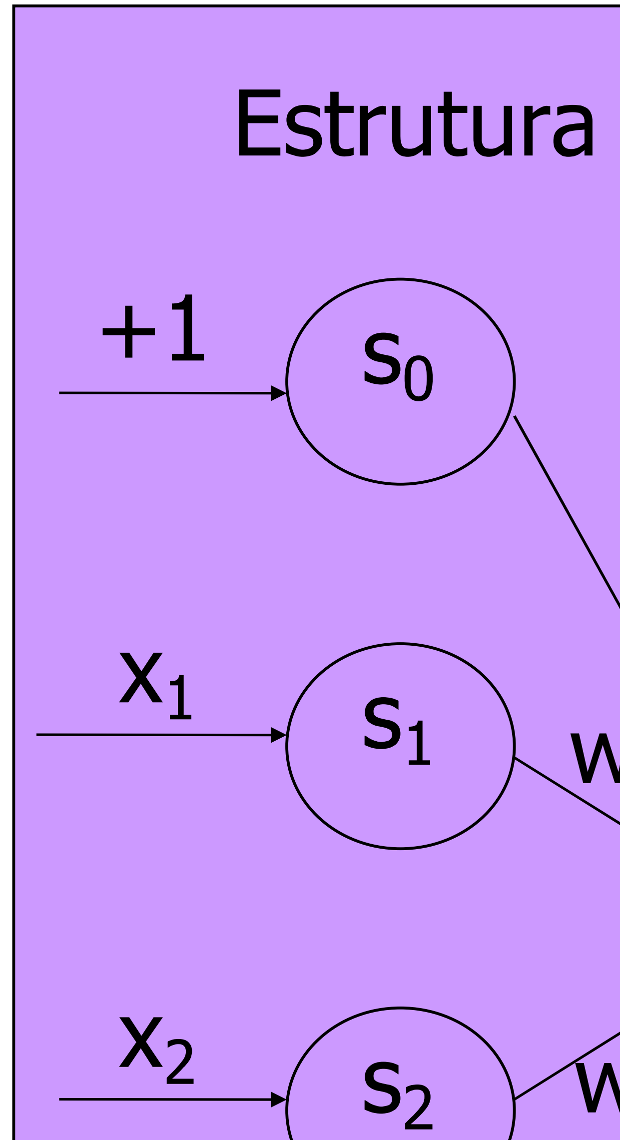
   Se $e_j != 0$, atualiza os pesos:

EXEMPLO DE PERCEPTRON

AND	x_0	x_1	x_2	t
Entrada 1:	1	0	0	0
Entrada 2:	1	0	1	0
Entrada 3:	1	1	0	0
Entrada 4:	1	1	1	1

Peso inicial: $w_0 = 0$, $w_1 = 0$, $w_2 = 0$

Taxa de aprendizado: $\eta = 0.5$



EXEMPLO DE PERCEPTRON - CICLO 1 (OU É

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+0\times0+0\times0) = f(0) = 0 \rightarrow$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+0\times1+0\times0) = f(0) = 0 \rightarrow$

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+0\times0+0\times1) = f(0) = 0 \rightarrow$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+0\times1+0\times1) = f(0) = 0 \rightarrow$

$w_0 = w_0+(t-s_{out})x_0 = 0 + 0.5 \times(1-0) \times1=0.5$

EXEMPLO DE PERCEPTRON - CICLO 2

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0.5\times1+0.5\times0+0.5\times0) = f(0.5) = 1$

$w_0 = w_0+(t-s_{out})x_0 = 0.5 + 0.5 \times(0-1) \times1 = 0$

$w_1 = w_1+(t-s_{out})x_1 = 0.5 + 0.5 \times(0-1) \times0 = 0.$

$w_2 = w_2+(t-s_{out})x_2 = 0.5 + 0.5 \times(0-1) \times0 = 0.$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+0.5\times0+0.5\times1) = f(0.5) = 1$

$w_0 = w_0+(t-s_{out})x_0 = 0 + 0.5 \times(0-1) \times1 = -0.5$

$w_1 = w_1+(t-s_{out})x_1 = 0.5 + 0.5 \times(0-1) \times0 = 0$

EXEMPLO DE PERCEPTRON - CICLO 2 (CONT

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+0.5\times1+0\times0) = f(0) = 0 \rightarrow$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+0.5\times1+0\times1) = f(0) = 0 \rightarrow$

$w_0 = w_0+(t-s_{out})x_0 = -0.5 + 0.5 \times(1-0) \times1 =$

$w_1 = w_1+(t-s_{out})x_1 = 0.5 + 0.5 \times(1-0) \times1 = 1$

$w_2 = w_2+(t-s_{out})x_2 = 0 + 0.5 \times(1-0) \times1 = 0.5$

EXEMPLO DE PERCEPTRON - CICLO 3

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+1\times0+0.5\times0) = f(0) = 0 \longrightarrow s$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(0\times1+1\times0+0.5\times1) = f(0.5) = 1 \longrightarrow s$

$w_0 = w_0+(t-s_{out})x_0 = 0 + 0.5 \times(0-1) \times1 = -0.5$

$w_1 = w_1+(t-s_{out})x_1 = 1 + 0.5 \times(0-1) \times0 = 1$

$w_2 = w_2+(t-s_{out})x_2 = 0.5 + 0.5 \times(0-1) \times1 = 0$

EXEMPLO DE PERCEPTRON - CICLO 3 (C

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+1\times1+0\times0) = f(0.5) = 1 \longrightarrow s_{ou}$

$w_0 = w_0+(t-s_{out})x_0 = -0.5 + 0.5 \times(0-1) \times1 = -$

$w_1 = w_1+(t-s_{out})x_1 = 1 + 0.5 \times(0-1) \times1 = 0.5$

$w_2 = w_2+(t-s_{out})x_2 = 0 + 0.5 \times(0-1) \times0 = 0.0$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-1\times1+0.5\times1+0\times1) = f(-0.5) = 0 \longrightarrow s_{o}$

$w_0 = w_0+(t-s_{out})x_0 = -1 + 0.5 \times(1-0) \times1 = -0.5$

$w_1 = w_1+(t-s_{out})x_1 = 0.5 + 0.5 \times(1-0) \times1 = 1.0$

EXEMPLO DE PERCEPTRON - CICLO 4

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+1.0\times0+0.5\times0) = f(-0.5) = 0$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+1.0\times0+0.5\times1) = f(0) = 0$

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+1.0\times1+0.5\times0) = f(0.5) = 1$

$w_0 = w_0+(t-s_{out})x_0 = -0.5 + 0.5 \times(0-1) \times1 = -$

$w_1 = w_1+(t-s_{out})x_1 = 1.0 + 0.5 \times(0-1) \times1 = 0$

EXEMPLO DE PERCEPTRON - CICLO 4 (C

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$f(-1\times1+0.5\times1+0.5\times1) = f(0.0) = 0$

$w_0 = w_0+(t-s_{out})x_0 = -1.0 + 0.5 \times(1\text{-}0) \times1$

$w_1 = w_1+(t-s_{out})x_1 = 0.5 + 0.5 \times(1\text{-}0) \times1 =$

$w_2 = w_2+(t-s_{out})x_2 = 0.5 + 0.5 \times(1\text{-}0) \times0 =$

5ª Ciclo

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

EXEMPLO DE PERCEPTRON - CICLO 5

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-0.5\times1+1\times0+1\times1) = f(0.5) = 1 \longrightarrow s_{out}$

$w_0 = w_0+(t-s_{out})x_0 = -0.5 + 0.5 \times(0-1) \times1 = -1.0$

$w_1 = w_1+(t-s_{out})x_1 = 1.0 + 0.5 \times(0-1) \times0 = 1.0$

$w_2 = w_2+(t-s_{out})x_2 = 1.0 + 0.5 \times(0-1) \times1 = 0.5$

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-1.0\times1+1\times1+0.5\times0) = f(0.0) = 0 \longrightarrow s_{out}$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

EXEMPLO DE PERCEPTRON - CICLO 6 (F

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-1.0\times1+1\times0+0.5\times0) = f(-1.0) = 0 \longrightarrow s_o$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-1.0\times1+1\times0+0.5\times1) = f(-0.5) = 0 \longrightarrow s_o$

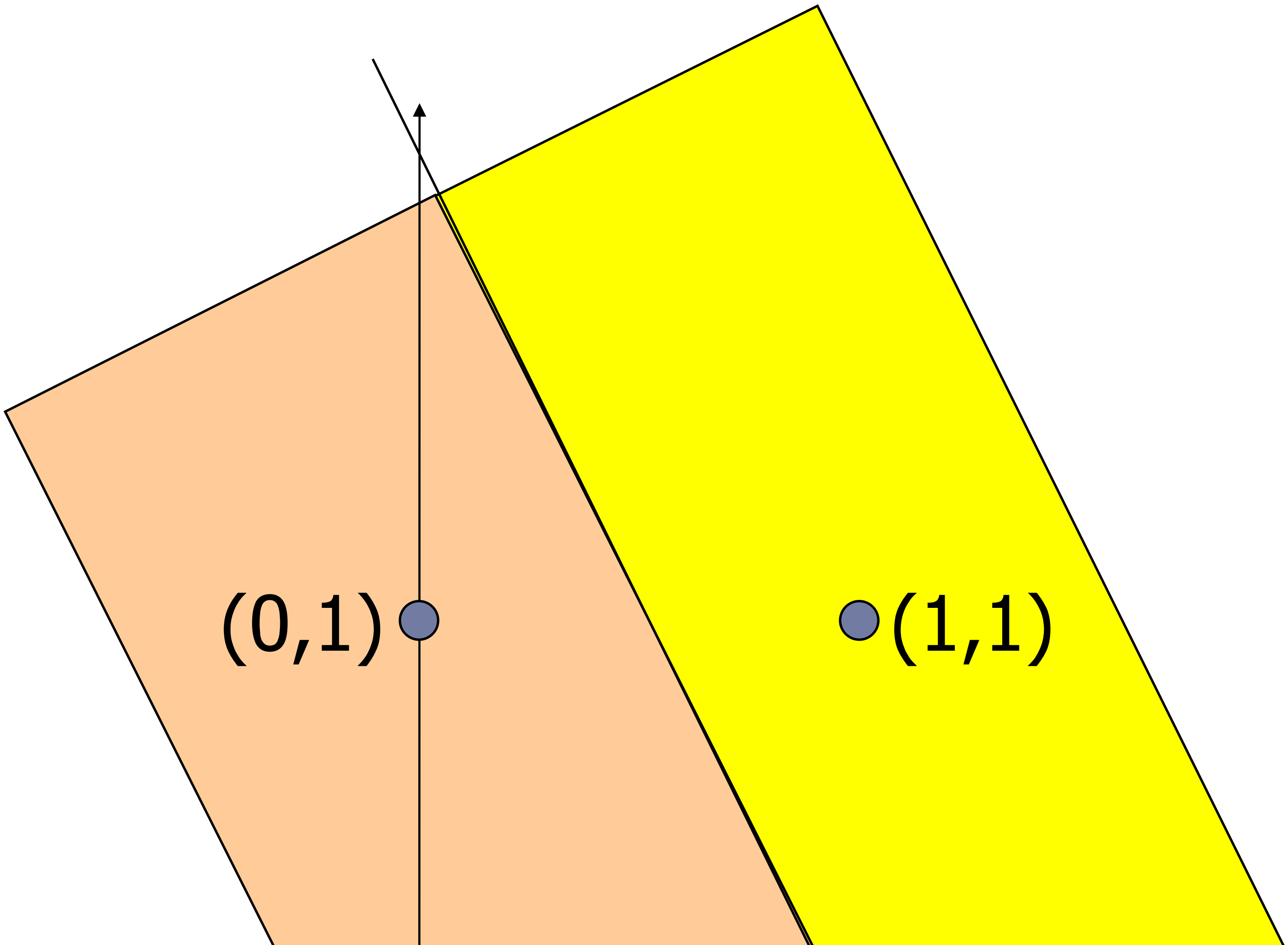
Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-1.0\times1+1\times1+0.5\times0) = f(0.0) = 0 \longrightarrow s_o$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$

$= f(-1.0\times1+1\times1+0.5\times1) = f(0.5) = 1 \longrightarrow s_o$

INTERPRETAÇÃO GEOMÉTRICA



Linha de

$x_1w_1+x_2$

VARIANDO A TAXA DE APREND

- Taxa constante:

$$\eta(n) = \eta_0 \qquad \text{for all } n$$

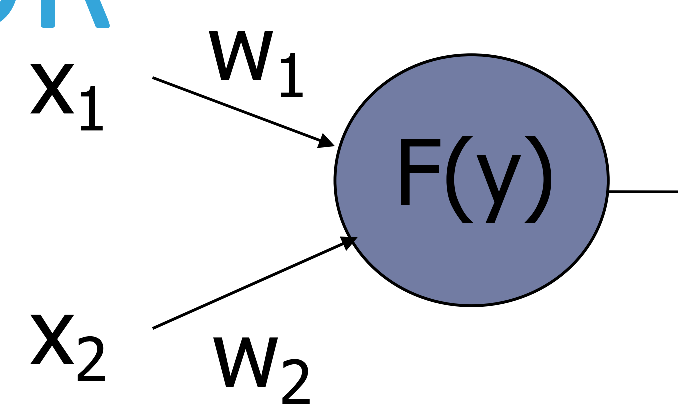
- Taxa variando no tempo:

$$\eta(n) = \frac{c}{n}$$

- Estratégia "Procura-então-converge":

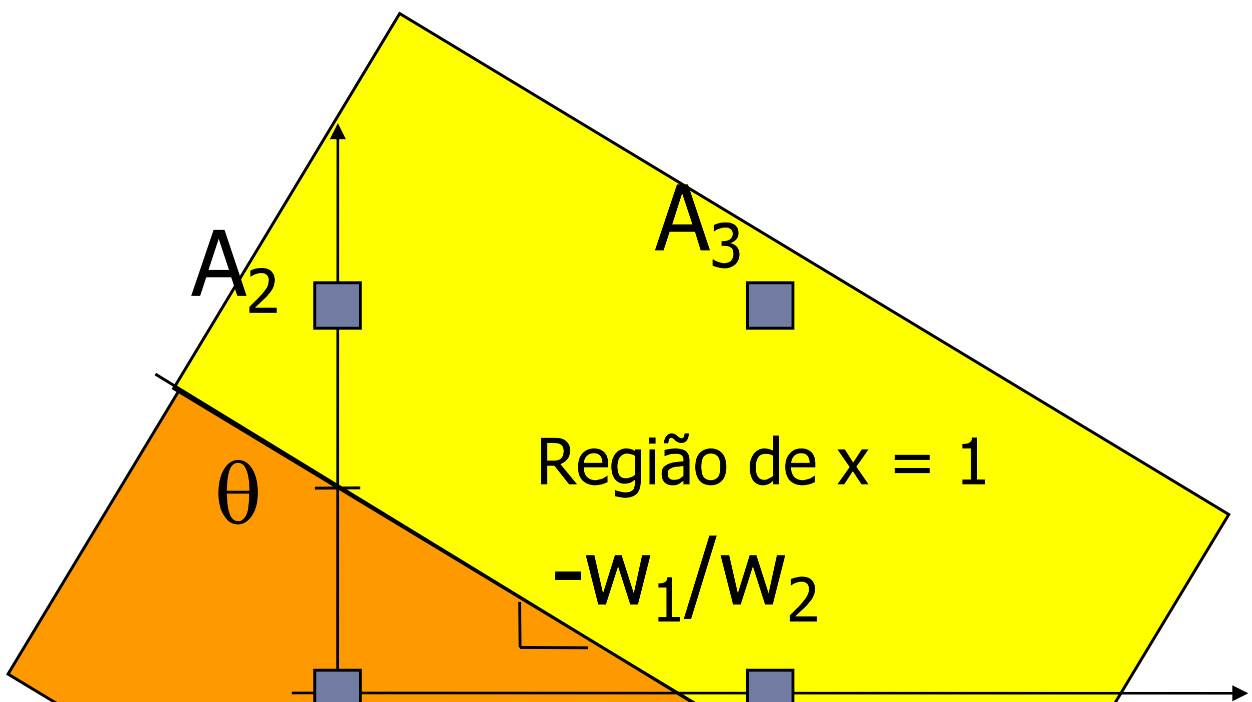
O PROBLEMA DO XOR

PONTO	X_1	X_2	Saída
A_0	0	0	0
A_1	0	1	1
A_2	1	0	1
A_3	1	1	0



De acordo com a
neurônio: x = F(x

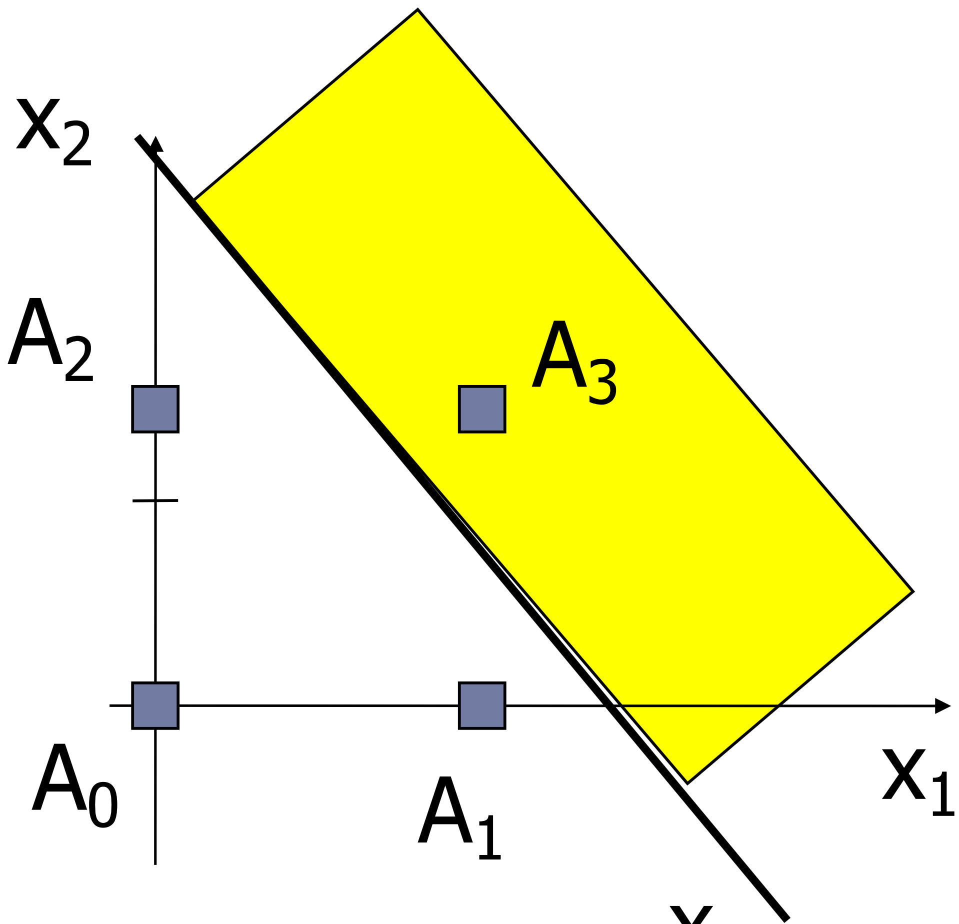
$$y = x_1 w_1 + x_2 w_2 + \theta \rightarrow$$

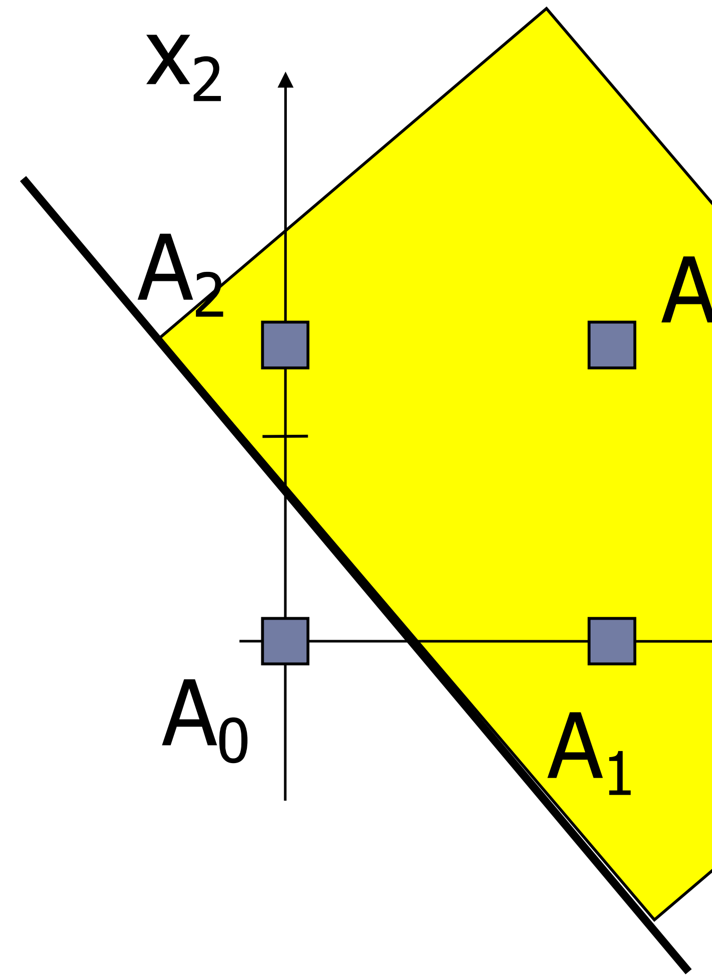


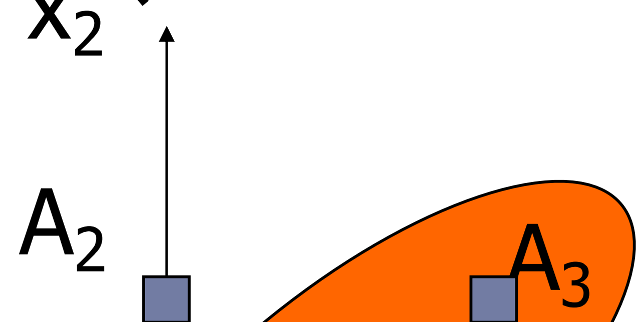
O PROBLEMA DO XOR

- Mudando-se os valores de $w1$, $w2$ e θ, muda-s
inclinação e a posição da reta;
- Entretanto é impossível achar uma reta que di
de forma separar os pontos A1 e A2 de um lad
de outro
- Resumo: redes com uma única camada só repr
funções linearmente separáveis

O PROBLEMA DO XOR

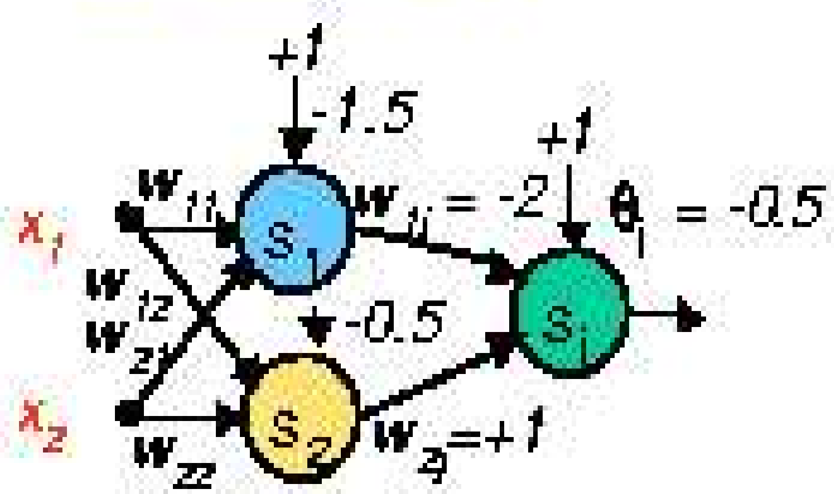






O PROBLEMA DO XOR

Exemplo:

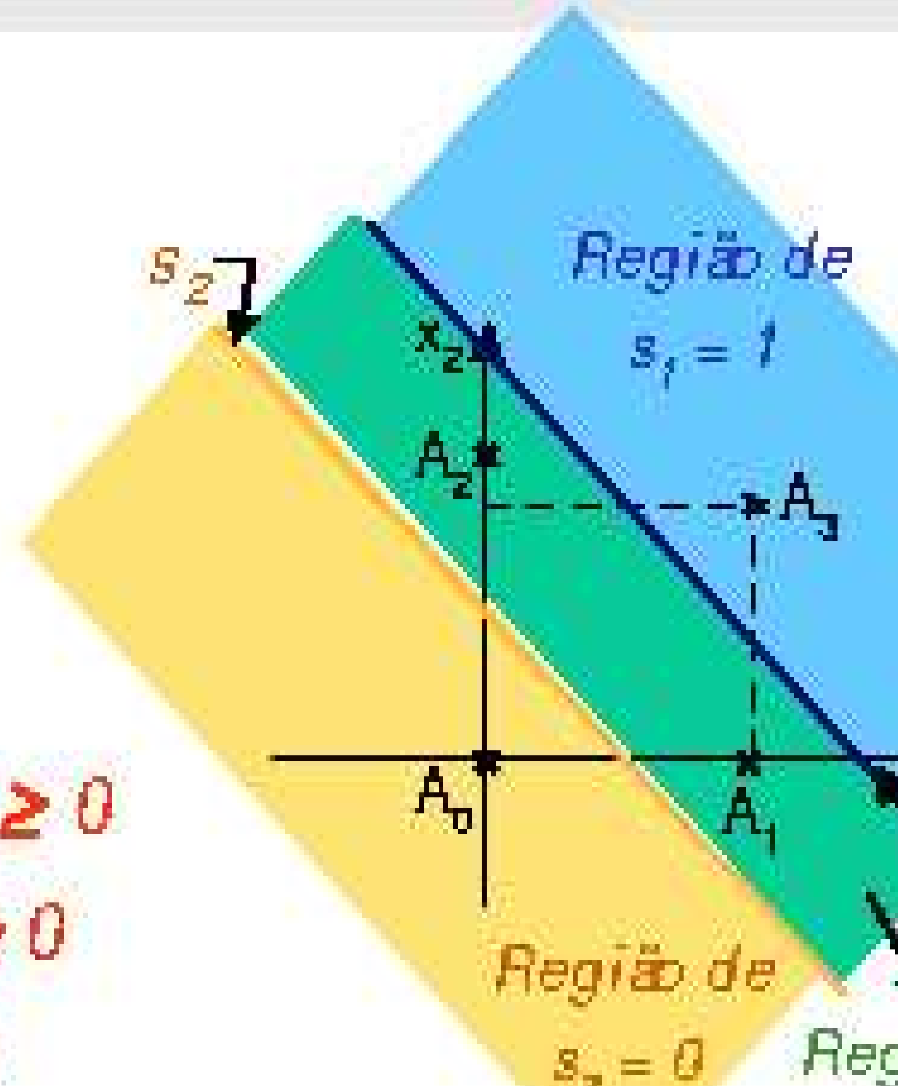


$$w_{11} = w_{12} = w_{21} = w_{22} = +1$$

$$s_j = 1 \rightarrow s_1 w_{1j} + s_2 w_{2j} + \theta_j \geq 0$$

$$-2s_1 + s_2 - 0.5 \geq 0$$

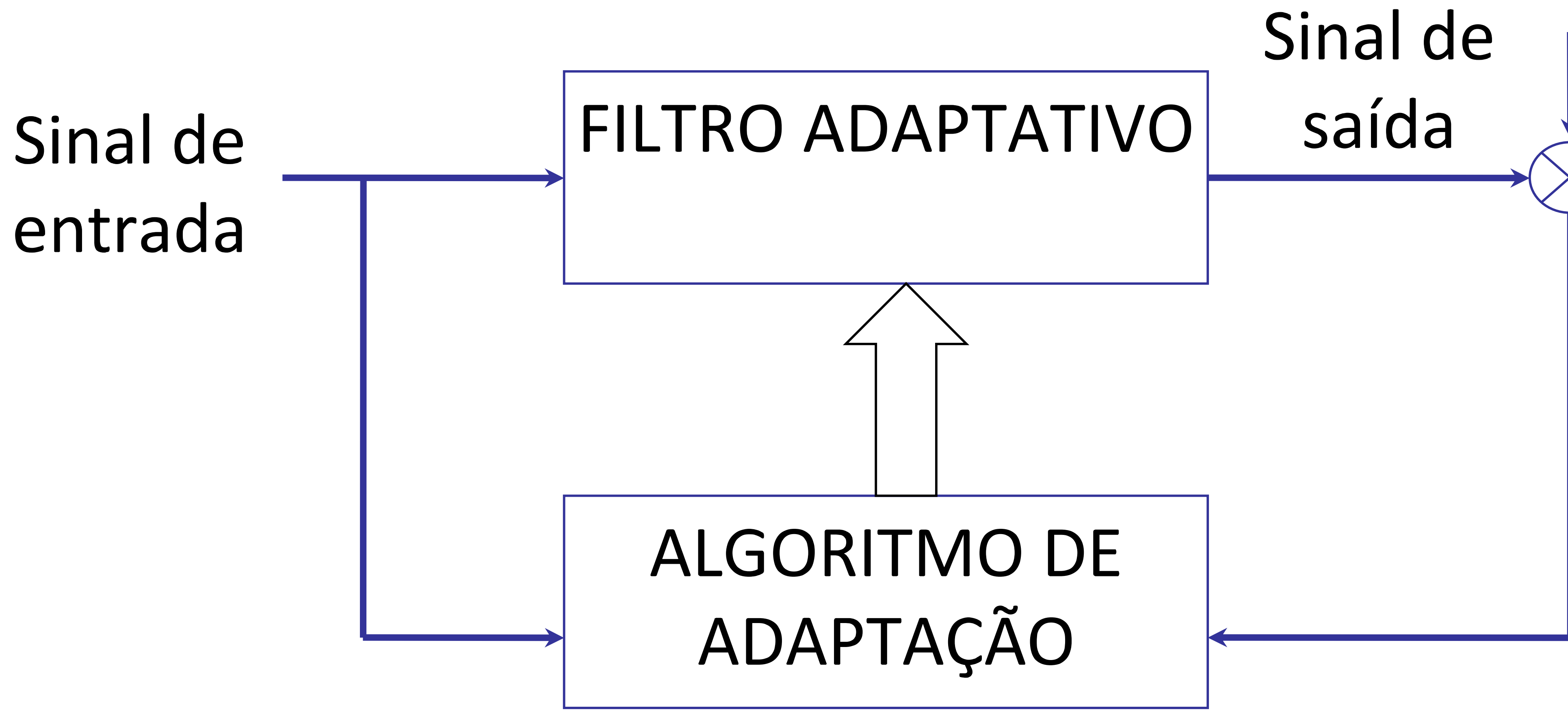
$$-2s_1 + s_2 \geq 0.5$$



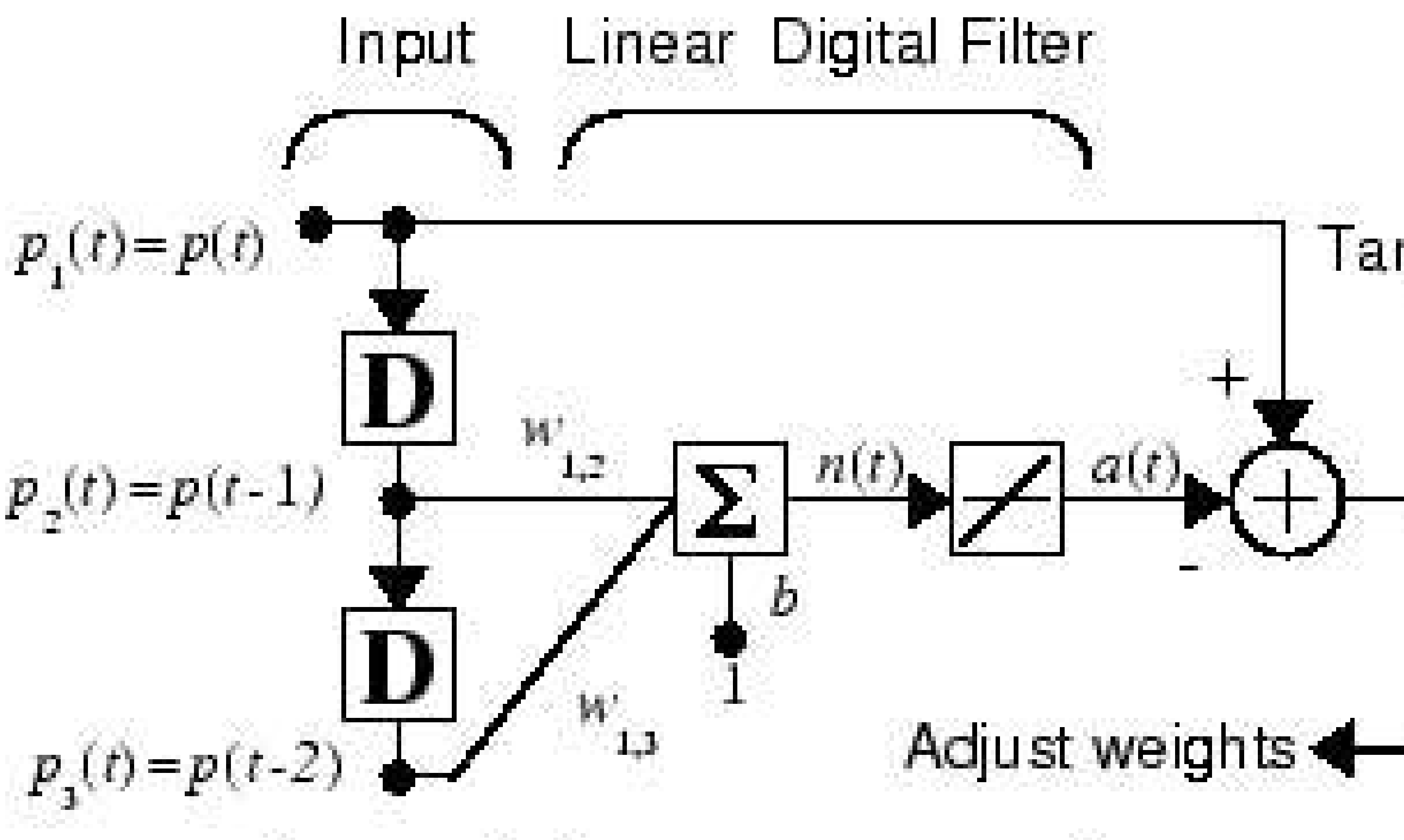
FILTRAGEM ADAPTATIVA

1. Filtros adaptativos são dispositivos auto-ajustáveis em algoritmos recursivos, que modificam seus parâ acordo com critérios pré-estabelecidos.
2. Em ambiente estacionário, acompanham a soluçã problema.
3. Em ambiente não estacionário, acompanham as m do sinal envolvido.
4. Base para o desenvolvimento do Adaline

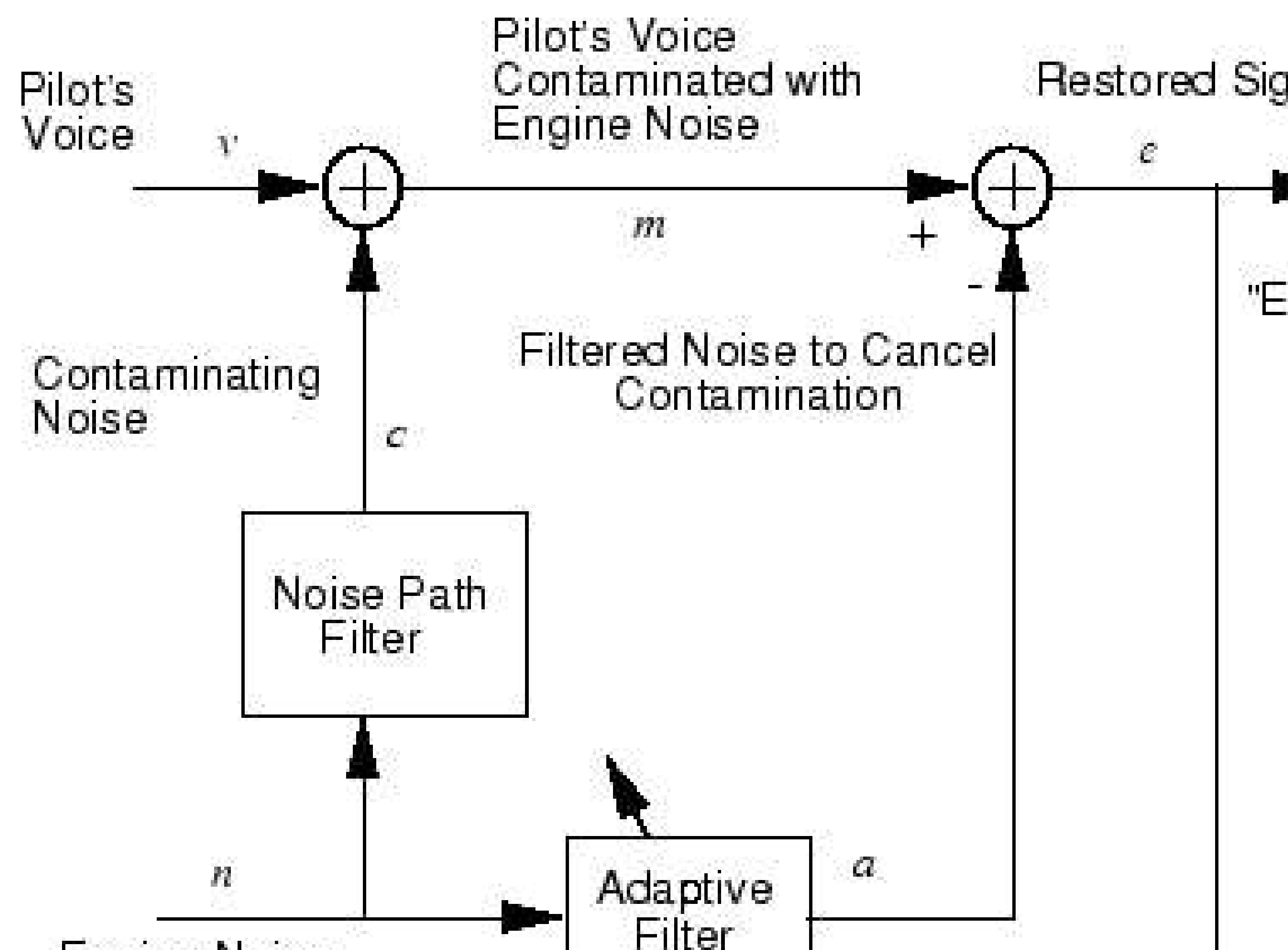
FILTRAGEM ADAPTATIVA



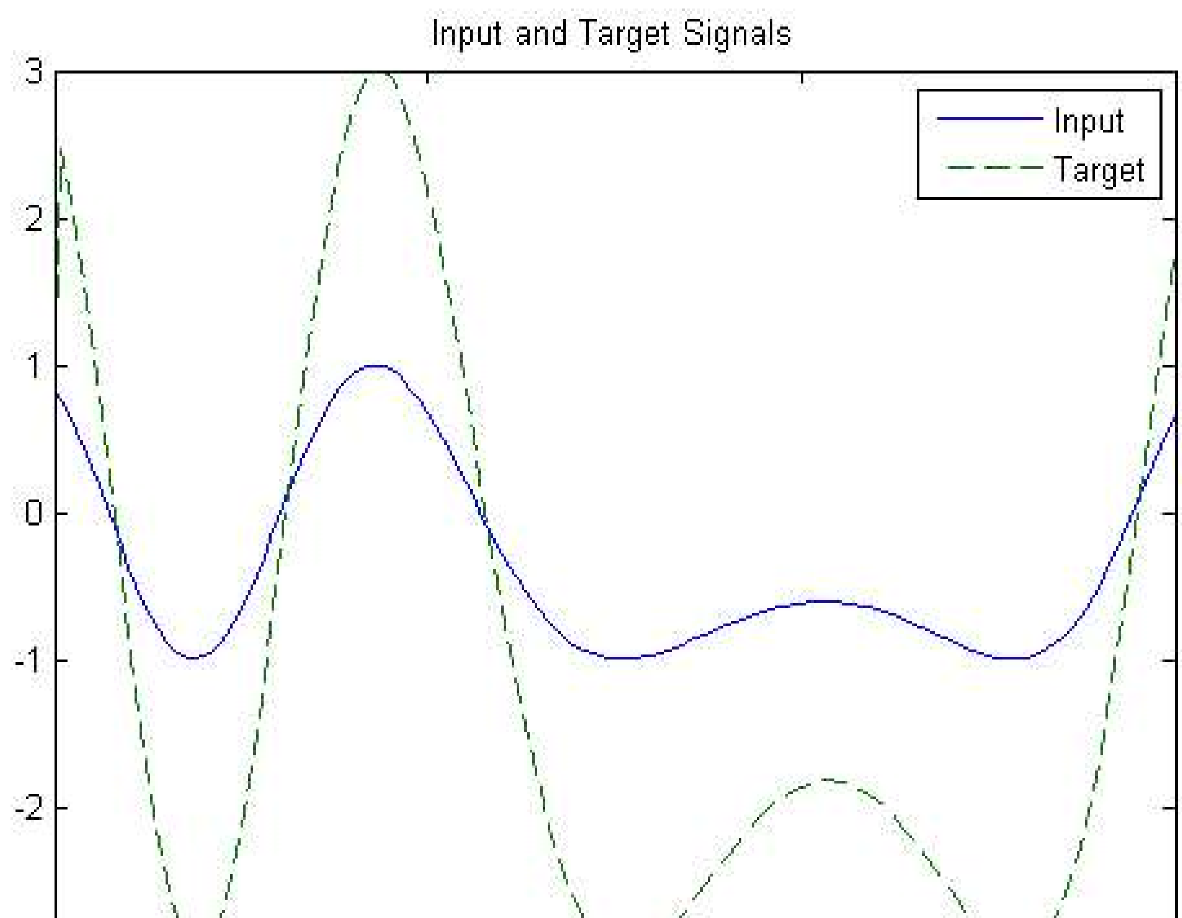
ADALINE: PREDIÇÃO

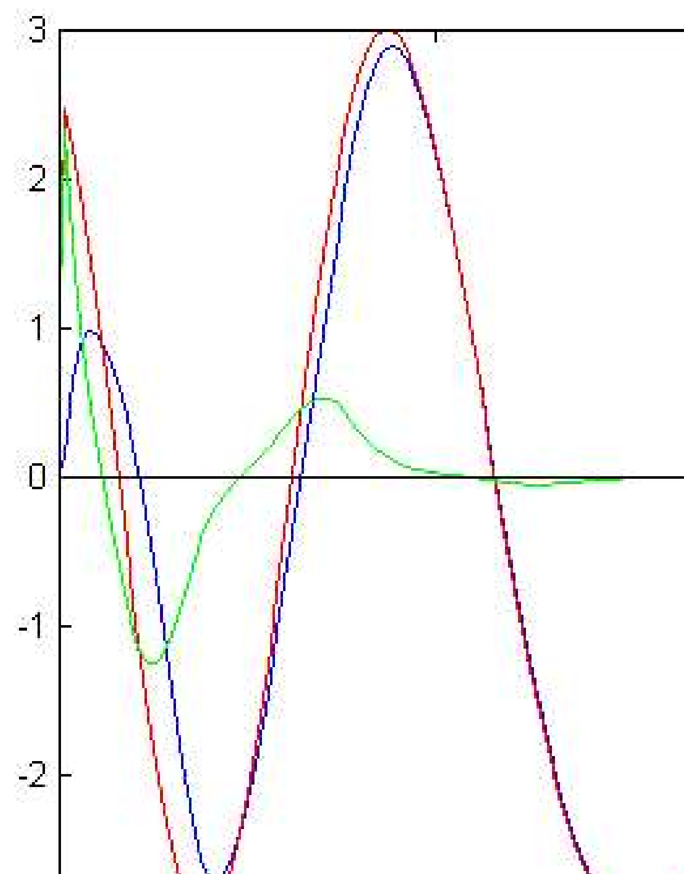


ADALINE: CANCELAMENTO DE



FILTRAGEM ADAPTATIVA



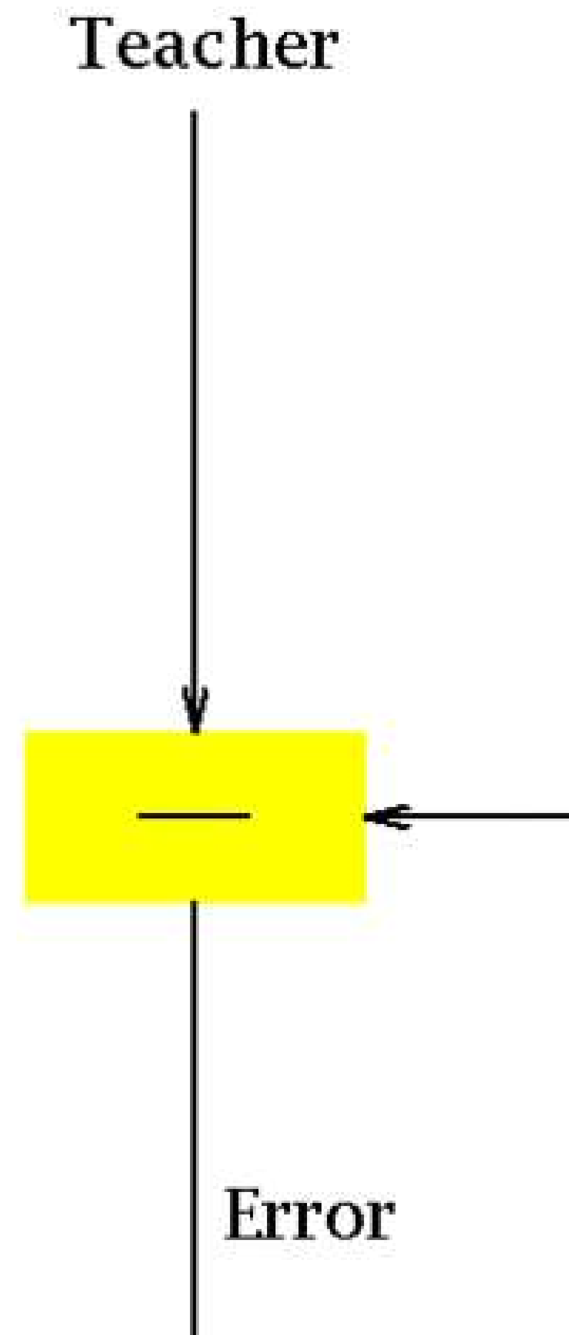


ADALINE

1. ADAptive LINear Element ou
2. ADAptive LInear NEuron

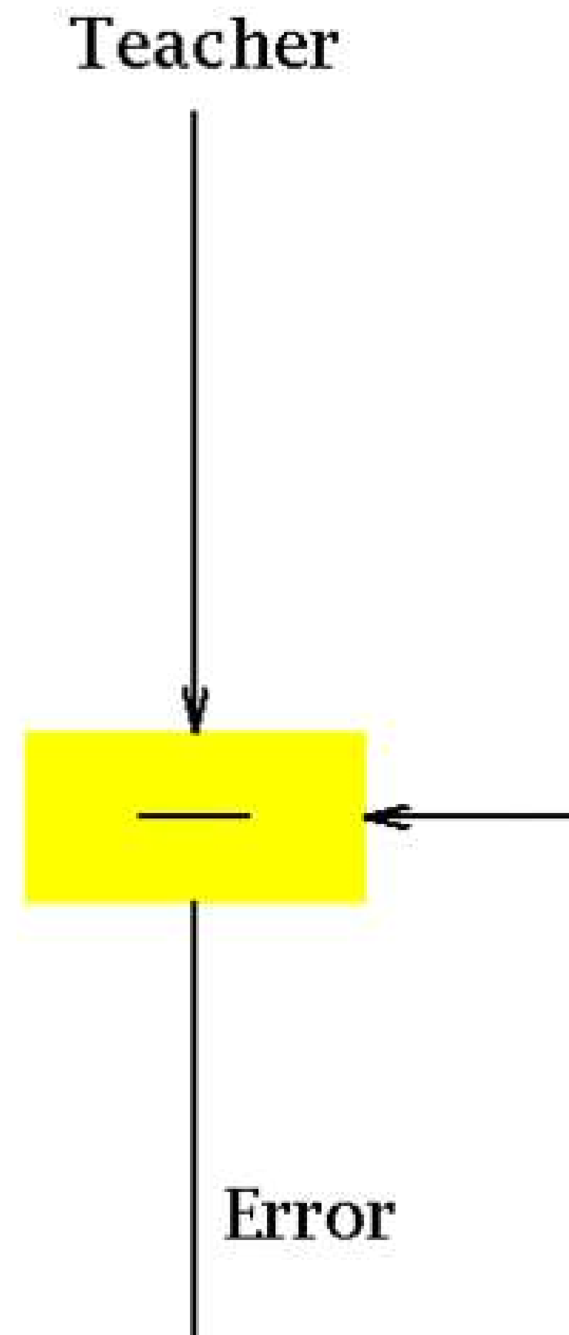
4. Concebido por Widrow e Hoff (1960)
5. Máquina adaptável para classificações de padrões
   problemas lineares

ADALINE



ADALINE

- O Modelo Adaline tem seus pesos adaptados em função do erro de sua saída linear (antes da aplicação da função de ativação (daí o nome



ADALINE

1. O algoritmo de aprendizagem visa minimizar o
saídas em relação aos valores desejados (conju
treinamento)

2. A função de custo a ser minimizada é a soma d
quadráticos:

$$E = \frac{1}{\;}\sum\left(d \quad s\right)^2$$

PROCESSO DE APRENDIZAGEM

- Processo de minimização do erro quadrático pel
Gradiente Descendente

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

- Cada peso sináptico i do elemento processador j
proporcionalmente ao negativo da derivada parc
deste processador em relação ao peso.

PROCESSO DE APRENDIZAGEM

Calcula Δw_{ij}

$$\Delta w_{ij} = -\eta \frac{\partial E_p}{\partial w_{ij}} = -\eta \boxed{\frac{\partial E_p}{\partial s_j}} \boxed{\frac{\partial s_j}{\partial w_{ij}}}$$

$$E_p = \frac{1}{2} \sum_j \left(d_j - s_j\right)^2$$

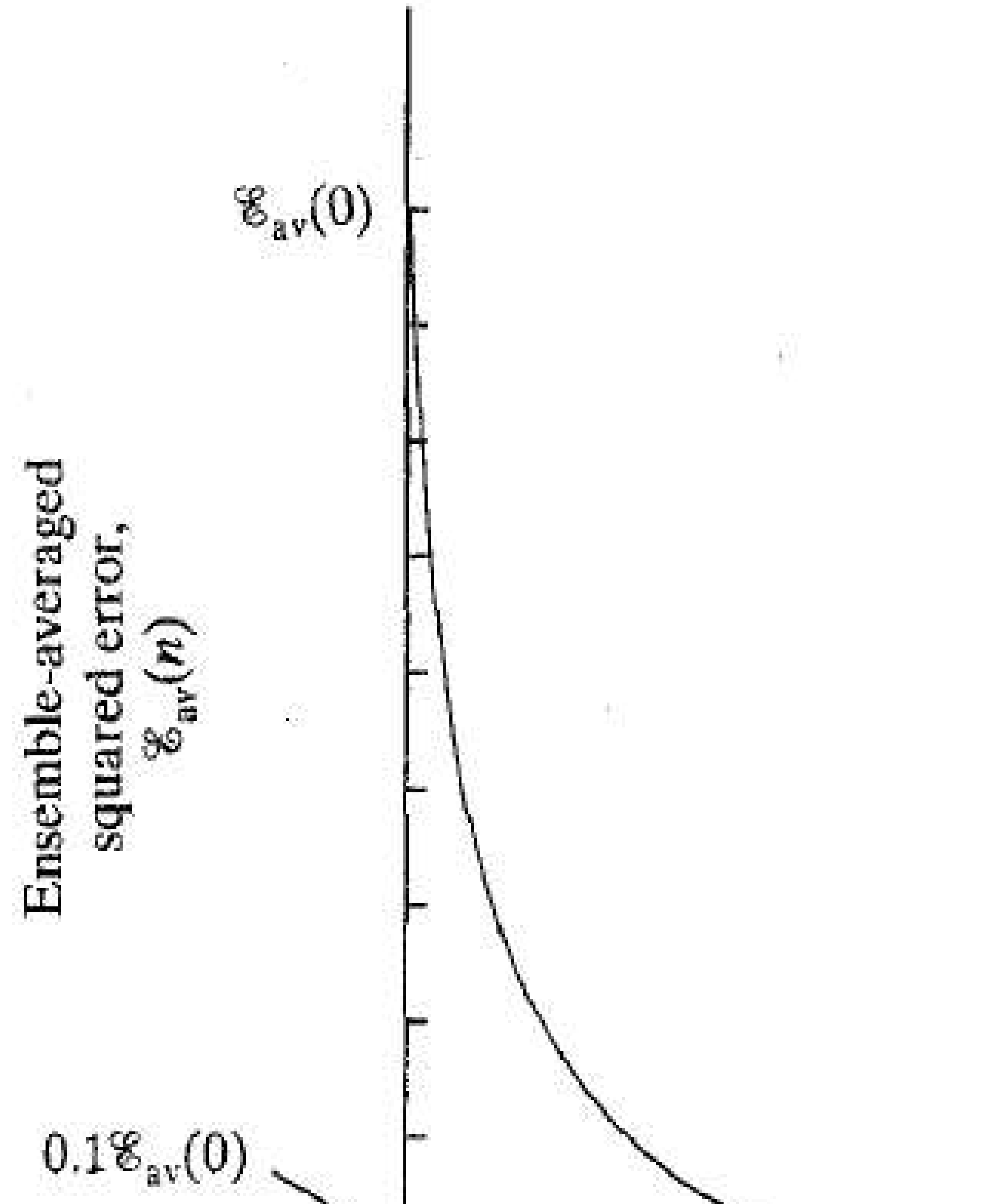
$$s_j = \sum$$

$$2 \times \frac{1}{2} \times \left(d_j - s_j\right)(-1)$$

$$x_i$$

ADALINE

- Curva de Aprendizagem



ADALINE

- É importante observar que a regra Delta (LMS
uma estimativa do vetor de pesos que resultar
utilização do método da descida mais íngreme
descendente). Por quê?

- “Método do Gradiente Estocástico”

ADALINE VS. PERCEPTRON

- Qual a diferença entre o processo de atualizaç
do Adaline e do Perceptron?

$$\Delta w_{ij} = \eta\left(d_j - s_j\right)x_i$$

UMA OBSERVAÇÃO!

- Redes Neurais de múltiplas camadas só oferecem va sobre as de uma única camada se existir uma funçã não-linear entre as camadas. Supondo:

$$\mathbf{y} = (x\mathbf{w}_A)\mathbf{w}_B$$

- Utilizando propriedades da álgebra linear

- Logo $\forall(\mathbf{w}_A \text{ e } \mathbf{w}_B)\ \exists\, \mathbf{w}_C \mid \mathbf{w}_C =$

MULTI LAYER PERCEPTRON (M

- Redes com apenas uma camada representar
  funções linearmente separáveis (correto?)
- Redes de múltiplas camadas solucionam ess
- O desenvolvimento do algoritmo Back-Propa
  um dos motivos para o reaquecimento da ár
  neurais

NEXT

1. Redes Multi Layer Perceptron (MLP) e o algorit
   Retropropagação (Backpropagation)