PERCEPTRON

- **Características Básicas**
- **Modelo de Neurônio**
- **Estrutura da Rede**
- **Algoritmo de Aprendizado**

CARACTERISTICAS BASICAS

- Regra de propagação $net_j = \sum_i x_i w_{ij} + \theta_i$

- Função de ativação: Degrau

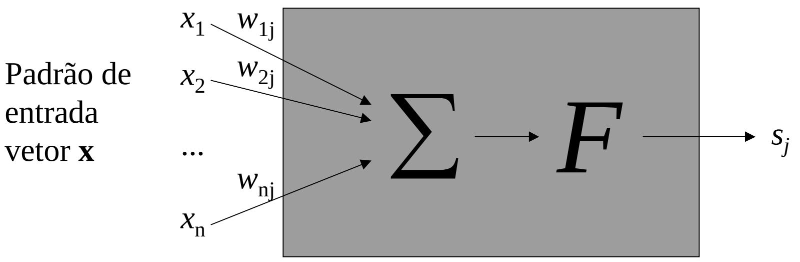
- Topologia: uma única camada de processadores

- Algoritmo de Aprendizado: $\Delta w_{ij} = \eta x_i (t_j - s_j)$
  (supervisionado)

- Valor de Entrada/Saída: Binários

MODELO DO NEURÔNIO

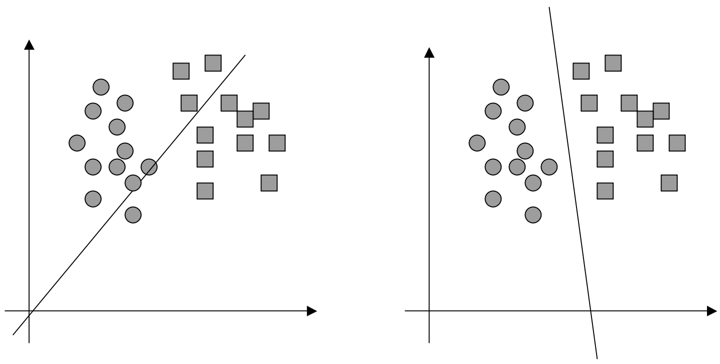
Na sua forma mais simples, o modelo do processador consiste em:



$$s_j = F(net_j) = F\left(\sum_i x_i w_{ij} + \theta_i\right) = \begin{cases} 1 & net_j > 0 \\ 0 & net_j \le 0 \end{cases}$$

PERCEPTRON

Finalidade do Termo *Bias*:



$\sum_i x_i w_{ij} = 0$ Define um hiperplano passando pela origem

$\sum_i x_i w_{ij} + \theta_i = 0$ Desloca-se o hiperplano da origem

ALGORITMO DE APRENDIZADO

1) iniciar os pesos sinápticos com valores randomicos e pequenos ou iguais a zero;
2) aplicar um padrão com seu respectivo valor desejado de saída (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 $E_j \neq 0$, atualiza os pesos: $\Delta w_{ij} = \eta x_i E_j$;
5) volta ao passo 2.

ALGORITMO DE APRENDIZADO

IMPORTANTE

- não ocorre variação no peso se a saída estiver correta;
- caso contrario, cada peso é incrementado de η quando a saída é menor que o target e decrementado de η quando a saída é maior que o target.

$$\boxed{\Delta w_{ij} = \eta\, x_i\, e_j}$$

PROCESSO DE APRENDIZADO

• Processo de minimização do erro quadrático pelo método do *Gradiente Descendente*

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

• Cada peso sináptico *i* do elemento processador *j* é atualizado proporcionalmente ao *negativo da derivada parcial do erro* deste processador com relação ao peso.

PROCESSO DE APRENDIZADO

Calcula Δw_{ij}

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

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

$$x_j = \sum x_i w_{ij} + \theta_j$$

$$2 \times \frac{1}{2} \times \left(t_j - x_j \right)(-1)$$

$$x_i$$

$$\boxed{\Delta w_{ij} = -\eta \left(t_j - x_j \right) x_i}$$

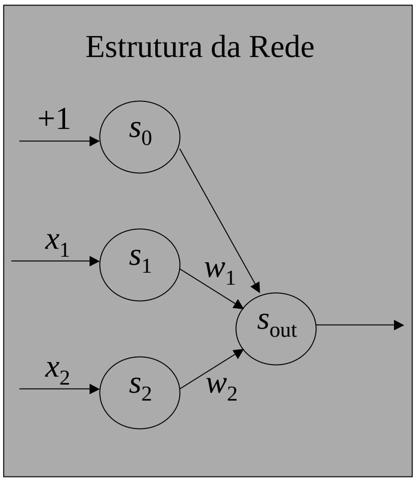
EXEMPLO

Simulação do Operador Lógico AND

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

1ª Cicle

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

$= f(0\times1+0\times0+0\times0) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(0\times1+0\times1+0\times0) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(0\times1+0\times0+0\times1) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(0\times1+0\times1+0\times1) = f(0) = 0 \Longrightarrow s_{out} \neq t$

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

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

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

EXEMPLO

2ª Ciclo

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

$= f(0.5\times 1+0.5\times 0+0.5\times 0) = f(0.5) = 1 \Longrightarrow s_{out} \neq t$

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

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

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

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

$= f(0\times 1+0.5\times 0+0.5\times 1) = f(0.5) = 1 \Longrightarrow s_{out} \neq t$

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

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

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

EXEMPLO

2ª Ciclo

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

$= f(-0.5\times 1+0.5\times 1+0\times 0) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(-0.5\times 1+0.5\times 1+0\times 1) = f(0) = 0 \Longrightarrow s_{out} \neq t$

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

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

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

EXEMPLO

3ª Ciclo

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

$= f(0\times1+1\times0+0.5\times0) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(0\times1+1\times0+0.5\times1) = f(0.5) = 1 \Longrightarrow s_{out} \neq t$

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

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

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

EXEMPLO

3ª Ciclo

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

$= f(-1\times1+1\times1+0\times0) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(-1\times1+1\times1+0\times1) = f(0) = 0 \Longrightarrow s_{out} \neq t$

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

$w_1= w_1+(t-s_{out})x_1= 1 + 0.5 \times(1-0) \times1 = 1.5$

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

EXEMPLO

4ª Ciclo

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

$= f(-0.5\times1+1.5\times0+0.5\times0) = f(-0.5) = 0 \Longrightarrow s_{out} = t$

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

$= f(-0.5\times1+1.5\times0+0.5\times1) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(-0.5\times1+1.5\times1+0.5\times0) = f(1) = 1 \Longrightarrow s_{out} \neq t$

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

$w_1= w_1+(t-s_{out})x_1= 1.5 + 0.5 \times(0-1) \times1 = 1$

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

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

$= f(-1\times1+1\times1+0.5\times1) = f(0.5) = 1 \Longrightarrow s_{out} = t$

EXEMPLO

5ª Ciclo

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

$= f(-1\times1+1\times0+0.5\times0) = f(-1) = 0 \Longrightarrow s_{out} = t$

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

$= f(-1\times1+1\times0+0.5\times1) = f(-0.5) = 0 \Longrightarrow s_{out} = t$

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

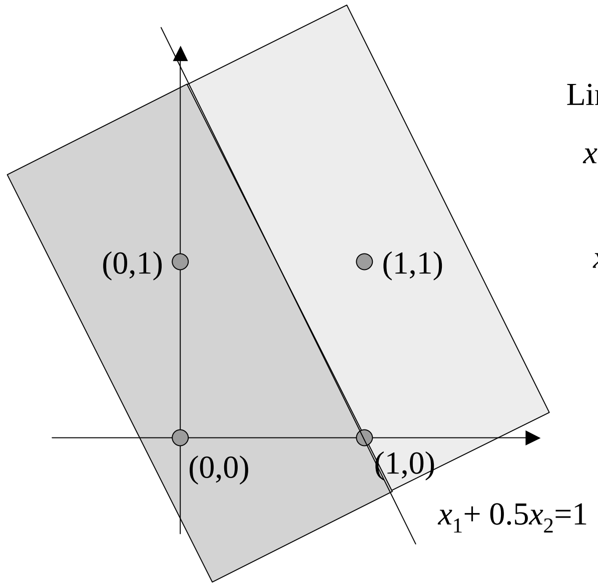
$= f(-1\times1+1\times1+0.5\times0) = f(0) = 0 \Longrightarrow s_{out} = t$

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

$= f(-1\times1+1\times1+0.5\times1) = f(0.5) = 1 \Longrightarrow s_{out} = t$

$\boxed{w_0 = -1,\ w_1= 1,\ w_2 = 0.5}$

INTERPRETAÇÃO GEOMÉTRICA



Linha de Decisão:

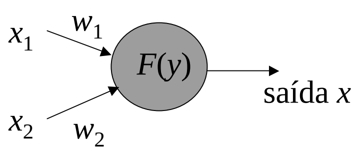
$$x_1 w_1 + x_2 w_2 = -\theta$$

$$\Downarrow$$

$$x_1 + 0.5\,x_2 = 1$$

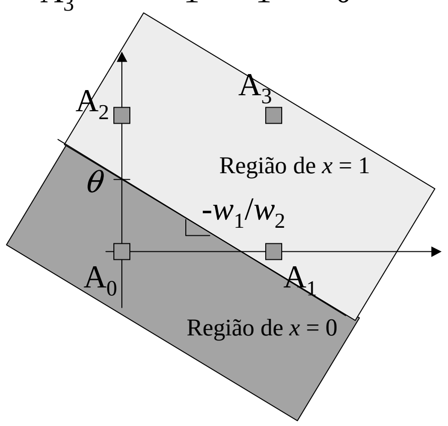
O PROBLEMA DO OU-EXCLUSIVO (XOR)

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 definição do neurônio: $x = F(x_1 w_1 + x_2 w_2 + \theta)$

$$y = x_1 w_1 + x_2 w_2 + \theta \rightarrow \begin{cases} se \;\; y \geq 0 \rightarrow x = 1 \\ se \;\; y < 0 \rightarrow x = 0 \end{cases}$$



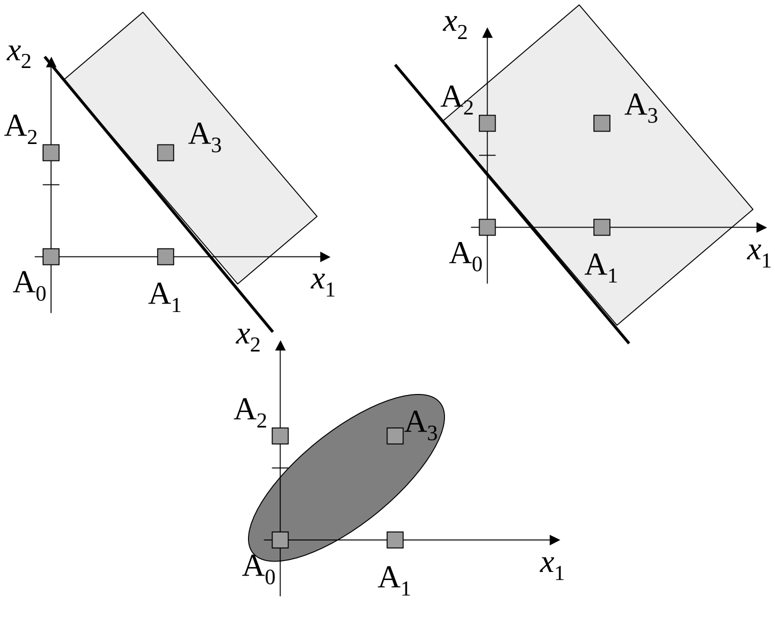
A rede perceptron divide o plano $X_1 \times X_2$ em duas regiões (através da reta y).

O PROBLEMA DO OU-EXCLUSIVO (XOR)

Conclusão

- mudando-se os valores de w_1, w_2 e θ, muda-se a inclinação e a posição da reta;
- entretanto é impossível achar uma reta que divide o plano de forma separar os pontos A_1 e A_2 de um lado e A_0 e A_3 de outro
- redes de 1 única camada só representam **funções linearmente separáveis**

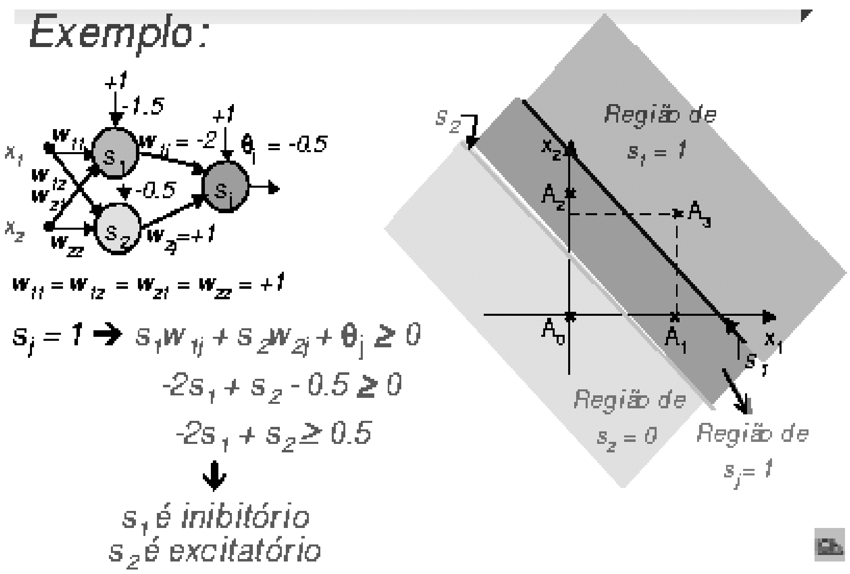
O PROBLEMA DO OU-EXCLUSIVO (XOR)



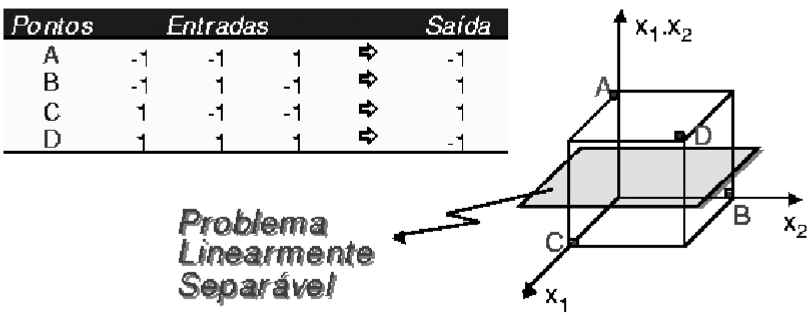
O PROBLEMA DO OU-EXCLUSIVO (XOR)

Minsky & Papert provaram que este problema pode ser solucionado adicionando-se uma outra camada intermediaria de processadores- Multi-Layer Perceptron (MLP)

O PROBLEMA DO OU-EXCLUSIVO (XOR)



O PROBLEMA DO OU-EXCLUSIVO (XOR)



Pontos	Entradas				Saída
A	-1	-1	1	⇨	-1
B	-1	1	-1	⇨	1
C	1	-1	-1	⇨	1
D	1	1	1	⇨	-1

UMA OBSERVAÇÃO

- Redes Neurais de múltiplas camadas só oferecem vantagens sobre as de uma única camada se existir uma função de ativação não-linear entre as camadas.

Camada Escondida: $y_1 = x_0W_1$

$x_1 = k_1y_1$

Camada de Saída: $x_2 = k_2y_2 = k_2(x_1W_2)$

$= k_2((k_1y_1)W_2)$

$= k_2((k_1x_0W_1)W_2)$

$= k_2k_1(x_0W_1)W_2$

$= Kx_0(W_1W_2)$

$= Kx_0W$

Equivalente a uma única camada

MULTI-LAYER PERCEPTRON

- Redes de apenas uma camada só representam funções linearmente separáveis
- Redes de múltiplas camadas solucionam essa restrição
- O desenvolvimento do algoritmo Back-Propagation foi um dos motivos para o ressurgimento da área de redes neurais