

CONTEÚDO

- **Introdução**
 - Motivação, Objetivo, Definição, Características Básicas e Histórico
- **Conceitos Básicos**
 - Neurônio Artificial, Modos de Interconexão
- **Processamento Neural**
 - *Learning e Recall*
- **Regras de Aprendizado**
 - Perceptron, Back Propagation e Competitive Learning.



Regra de HEBB

- **Declaração de Hebb:**

*“Quando um axônio de uma **célula A** está próximo o suficiente de **excitar a célula B** e repetidamente ou persistentemente participa da ativação desta, um processo de crescimento ou mudança metabólica ocorre em uma ou ambas as células, de tal forma que a **eficiência de A em ativar B** é **umentada**”* ⇨ *Portanto, a cada apresentação do padrão a saída fica mais reforçada.*



Regra de HEBB

- **Em termos práticos:**
 - Se dois neurônios em cada lado de uma sinápsse (conexão) ***são ativados simultaneamente*** (sincronamente), então a “**força**” daquela ***sinápsse deve ser aumentada***.
 - Se dois neurônios em cada lado de uma sinápsse ***são ativados assincronamente***, então aquela ***sinápsse deve ser enfraquecida***.



Regra de HEBB

- **Conclusão:**

A **modificação** nas sinápses tem relação com a **correlação entre as atividades** dos dois neurônios envolvidos na conexão.



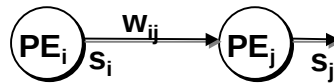
correlação + 1 o valor do peso **aumenta**
correlação - 1 o valor do peso **diminui**



A saída é reforçada a cada apresentação do padrão 1 **padrões frequentes terão maior influência no vetor de pesos do neurônio.**



Regra de HEBB



$$\Delta w_{ij} = h \cdot s_i \cdot s_j$$
$$0 < h < 1$$

“Hebbian Learning Rule”
taxa de aprendizado



- **Aprendizado Não-Supervisionado**
- **A informação está disponível localmente**



Algoritmos de Aprendizado

Os algoritmos podem ser definidos através das seguintes características:

- ❖ **Regra de Propagação**
- ❖ **Função de Ativação**
- ❖ **Topologia**
- ❖ **Regra de Aprendizado - Δw**

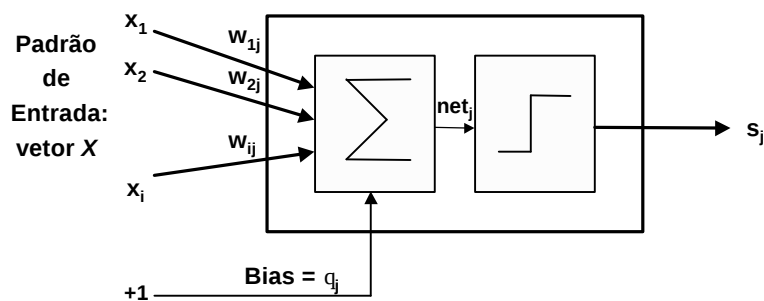


Perceptron



Perceptron

Na sua forma mais simples o modelo do processador consiste de:



Perceptron

- **Características Básicas:**

- Regra de Propagação → $net_j = \sum x_i \cdot w_{ij} + q_j$
- Função de Ativação → **Degrau**
- Topologia → **Uma única camada de processadores.**
- Algoritmo de Aprendizado → **Supervisionado:**
 $Dw_{ij} = h \cdot s_i \cdot (t_j - s_j)$
- Valores de Entrada/Saída → **Binários**



Algoritmo de Aprendizado

Iniciar os pesos sinápticos com valores randômicos e pequenos;

- 2 Aplicar um padrão de entrada, com seu respectivo valor de saída desejado (t_j), e verificar a saída da rede (s_j);

Calcular o erro na saída: $e_j = t_j - s_j$

- 4 Se $e_j = 0$, voltar ao passo 2;

Se $e_j \neq 0$, atualizar os pesos: $Dw_{ij} = h \cdot s_i \cdot e_j$

- 5 Voltar ao passo 2.



O Problema do OU-Exclusivo

Rosenblatt (1962) provou que:

Uma rede Perceptron é capaz de Aprender tudo que puder Representar.

Representação → refere-se à habilidade do sistema neural de representar (simular) uma função específica.

Aprendizado → refere-se à existência de um procedimento sistemático de aquisição de conhecimento (ajuste dos pesos), de forma a produzir a função desejada



O Problema do OU-Exclusivo

- Minsky & Papert provaram (*Perceptrons 1969*) que existem sérias restrições sobre o que as redes Perceptron são capazes de **Representar**.

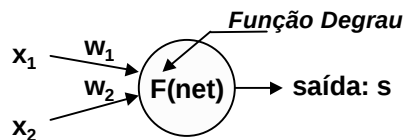


- Por exemplo, as redes Perceptron **NÃO** são capazes de **Representar** a função **OU-Exclusivo**



O Problema do OU-Exclusivo

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: $s = F(x_1w_1 + x_2w_2 + q)$

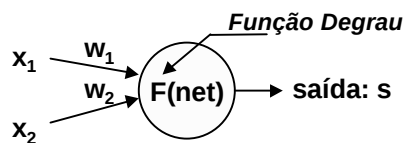
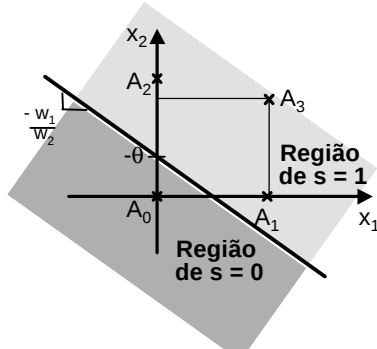
$$\text{net} = x_1w_1 + x_2w_2 + \theta \rightarrow \begin{cases} \text{Se net} \geq 0 \rightarrow s = 1 \\ \text{Se net} \leq 0 \rightarrow s = 0 \end{cases}$$

A rede Perceptron divide o plano $x_1 \times x_2$ em duas regiões (através da reta net)



O Problema do OU-Exclusivo

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: $s = F(x_1w_1 + x_2w_2 + q)$

$$\text{net} = x_1w_1 + x_2w_2 + \theta \rightarrow \begin{cases} \text{Se net} \geq 0 \rightarrow s = 1 \\ \text{Se net} \leq 0 \rightarrow s = 0 \end{cases}$$

A rede Perceptron divide o plano $x_1 \times x_2$ em duas regiões (através da reta net)



O Problema do OU-Exclusivo

Conclusão:

- mudando-se os valores de w_1 , w_2 e q , muda-se a **inclinação** e a **posição** da reta.



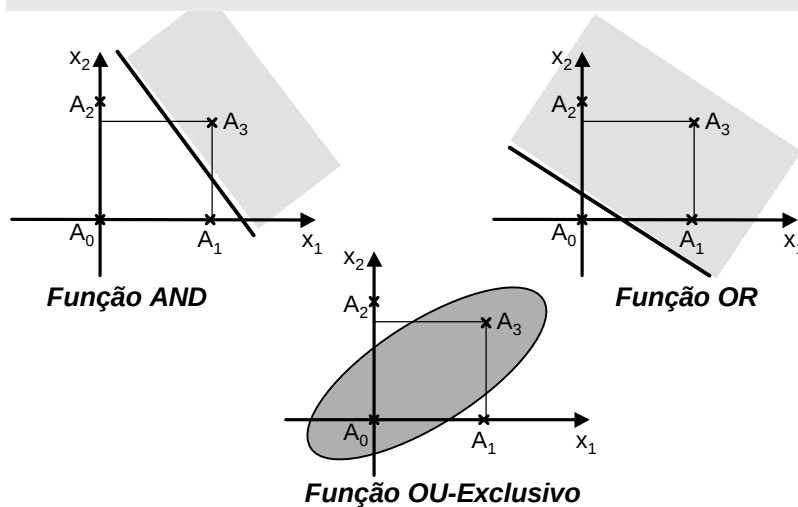
- Entretanto, é **impossível** achar uma reta que divida o plano de forma a separar os pontos A_1 e A_2 de um lado e A_0 e A_3 de outro.



- Redes de **1 única camada** só representam **funções linearmente separáveis!**



O Problema do OU-Exclusivo



O Problema do OU-Exclusivo

Minsky & Papert provaram que este problema pode ser solucionado adicionando-se uma outra *camada intermediária* de processadores.

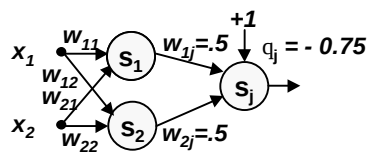


Multi-Layer Perceptrons



O Problema do OU-Exclusivo

Exemplo:

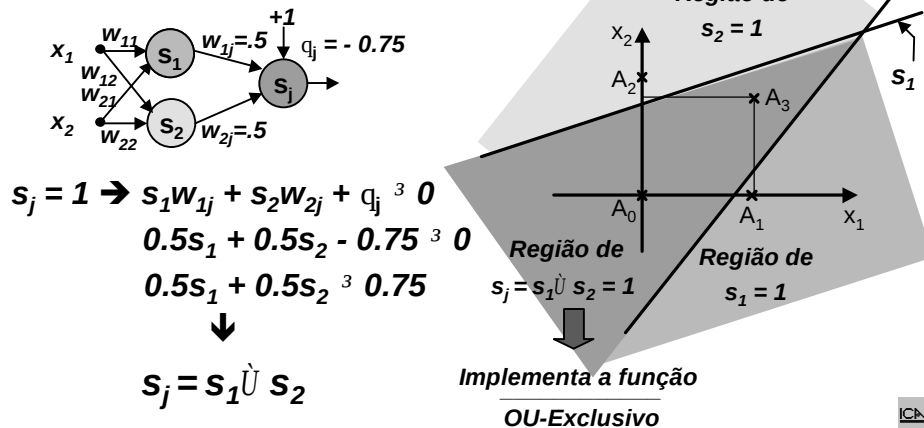


$$\begin{aligned}
 s_j = 1 &\Rightarrow s_1 w_{1j} + s_2 w_{2j} + q_j \stackrel{?}{=} 0 \\
 &0.5s_1 + 0.5s_2 - 0.75 \stackrel{?}{=} 0 \\
 &0.5s_1 + 0.5s_2 \stackrel{?}{=} 0.75 \\
 &\downarrow \\
 s_j &= s_1 \dot{\cup} s_2
 \end{aligned}$$



O Problema do OU-Exclusivo

Exemplo:



O Problema do OU-Exclusivo

- 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!

O Problema do OU-Exclusivo

- Em termos vetoriais:

Camada Escondida $\Downarrow$ $NET_1 = S_1 \cdot W_1$
 $S_1 = k_1 NET_1$

Camada de Saída $\Downarrow$ $S_2 = k_2 NET_2 = k_2(S_1 \cdot W_2)$ $\Downarrow$
 $S_2 = k_2 [(k_1 NET_1) W_2]$ $\Downarrow$
 $S_2 = k_2 [(k_1 S_1 W_1) W_2]$ $\Downarrow$
 $S_2 = k_2 k_1 (S_1 W_1) W_2$ $\Downarrow$
 $S_2 = K \cdot S_1 \cdot (W_1 \cdot W_2)$
 $S_2 = K \cdot S_1 \cdot W$



Equivalente a uma única camada



Verificando o Problema do OU-EXCLUSIVO

NeuralWorks Professional II

NWE

