

# REDES NEURAIS ARTIFICIAIS

# REDES NEURAIS ARTIFICIAIS

- O QUE É UMA REDE NEURAL
- NEURÔNIOS BIOLÓGICOS
- CÉREBRO HUMANO E CAPACIDADE DE GENERALIZAÇÃO.
- Modelo McCulloch e Pitts
- FUNÇÕES DE ATIVAÇÃO
- APRENDIZADO
- APRENDIZADO SUPERVISIONADO
- CORREÇÃO DE ERROS
- PERCEPTRON DE UMA CAMADA E SEU ALGORITMO DE APRENDIZADO
- Rede Neural Multilayer Perceptron

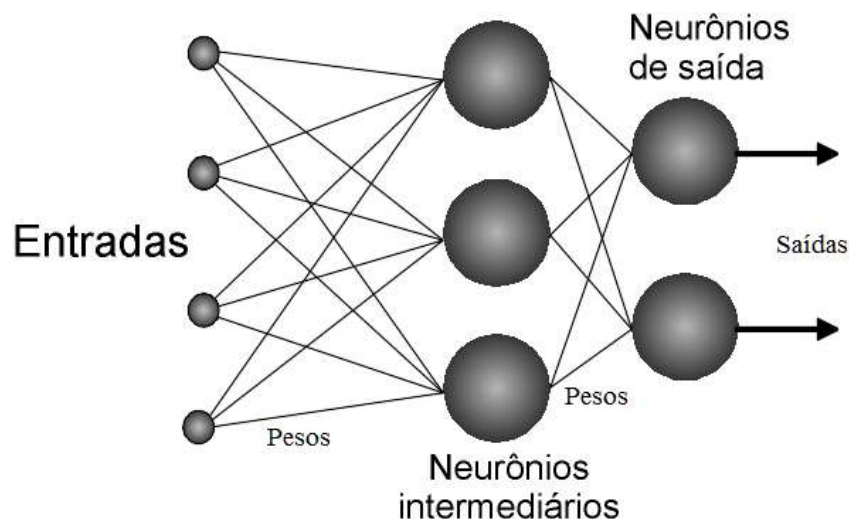
# REDES NEURAIS ARTIFICIAIS

- ÁREA DE REDES NEURAIS
- CONHECIDA TAMBÉM COMO CONEXIONISMO OU SISTEMAS DE PROCESSAMENTO PARALELO E DISTRIBUÍDO.
- É UMA FORMA DE COMPUTAÇÃO NÃO ALGORITMICA QUE RELEMBRAM A ESTRUTURA DE PROCESSAMENTO DO CÉREBRO HUMANO.



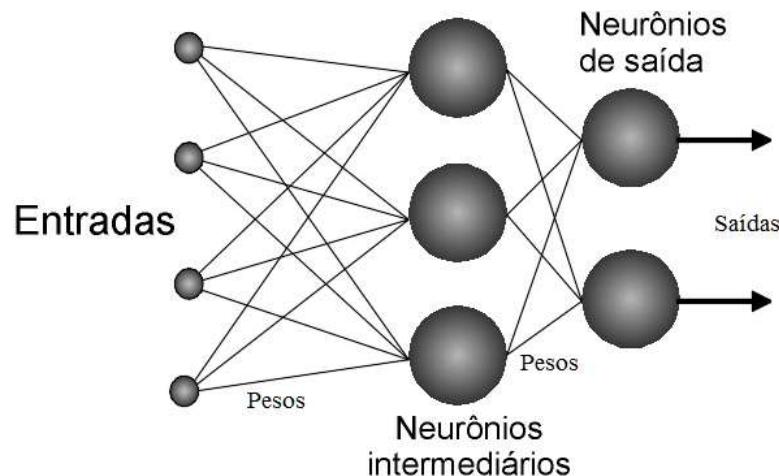
# REDES NEURAIS ARTIFICIAIS

- As RNAs são sistemas paralelos e distribuídos compostos por unidades de processamentos simples (nodos ou neurônio artificial) que computam determinadas funções matemáticas (normalmente não lineares).
- Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente, unidirecionais.
- Estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede.



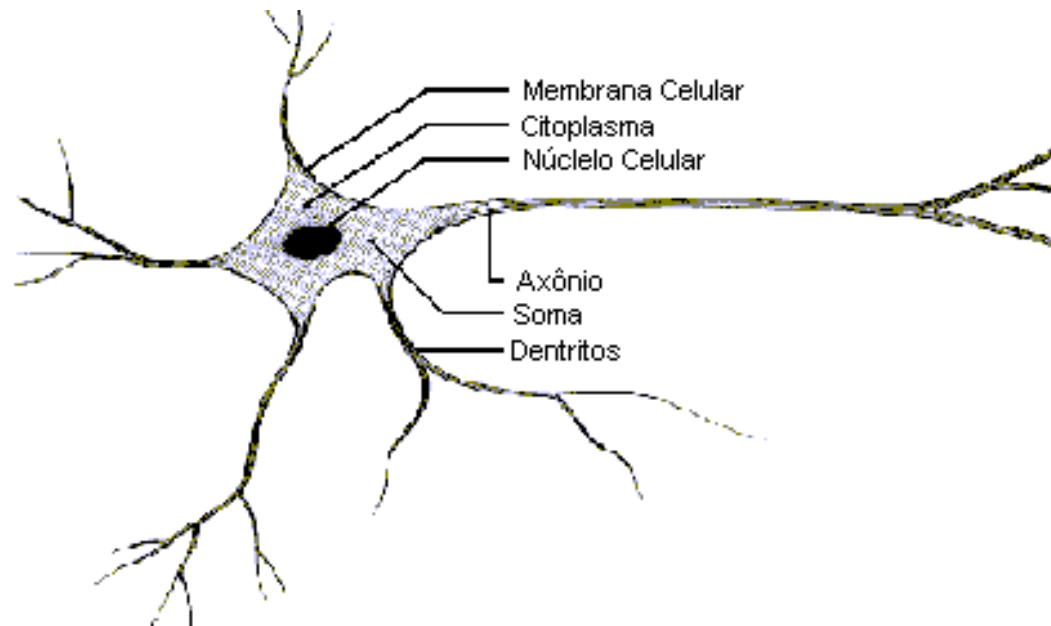
# REDES NEURAIS ARTIFICIAIS

- O PARALELISMO É A PRINCIPAL CARACTERÍSTICA DE UMA REDE NEURAL, UMA VEZ QUE CRIA A POSSIBILIDADE DE DESEMPENHO SUPERIOR EM RELAÇÃO A SOLUÇÃO DE PROBLEMAS BASEADOS NOS MODELOS ALGORÍTMICOS.
- UMA RNA É CAPAZ DE APRENDER ATRAVÉS DE EXEMPLOS E DE GENERALIZAR A INFORMAÇÃO APRENDIDA.
- A GENERALIZAÇÃO É A CAPACIDADE DE APRENDER ATRAVÉS DE UM CONJUNTO REDUZIDO DE EXEMPLOS E POSTERIORMENTE DAR RESPOSTAS A DADOS NÃO APRESENTADOS A REDE.



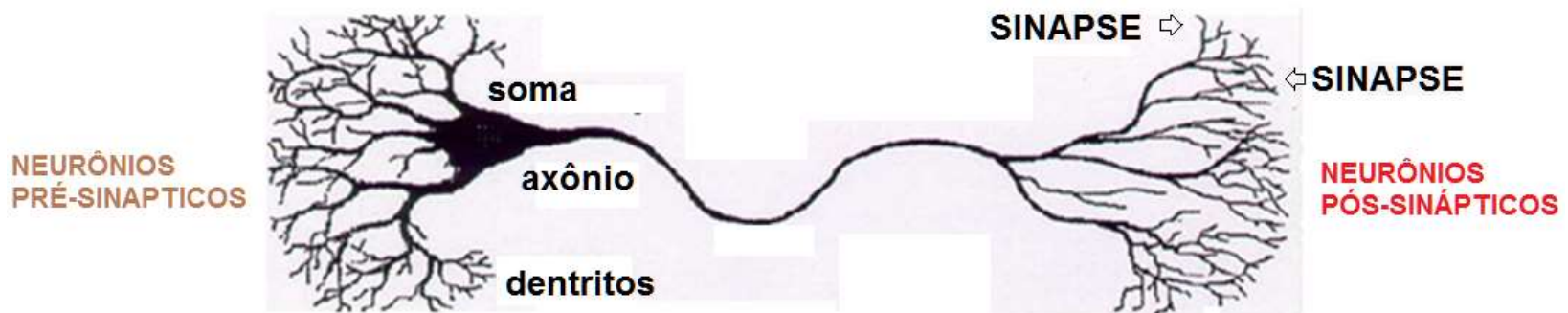
# NEURÔNIO BIOLÓGICO

- O corpo do neurônio (soma) mede apenas alguns milésimos de milímetros. Os dendritos apresentam poucos milímetros de comprimento. O axônio pode ser mais longo.
- Os dendritos têm como função, receber informações, ou *impulsos nervosos*, oriundas de outros neurônios, e conduzi-las até o corpo celular. Ali, a informação é processada e novos impulsos são gerados. Estes impulsos são transmitidos a outros neurônios (nodos), passando pelo do axônio e atingindo os dendritos dos neurônios seguintes



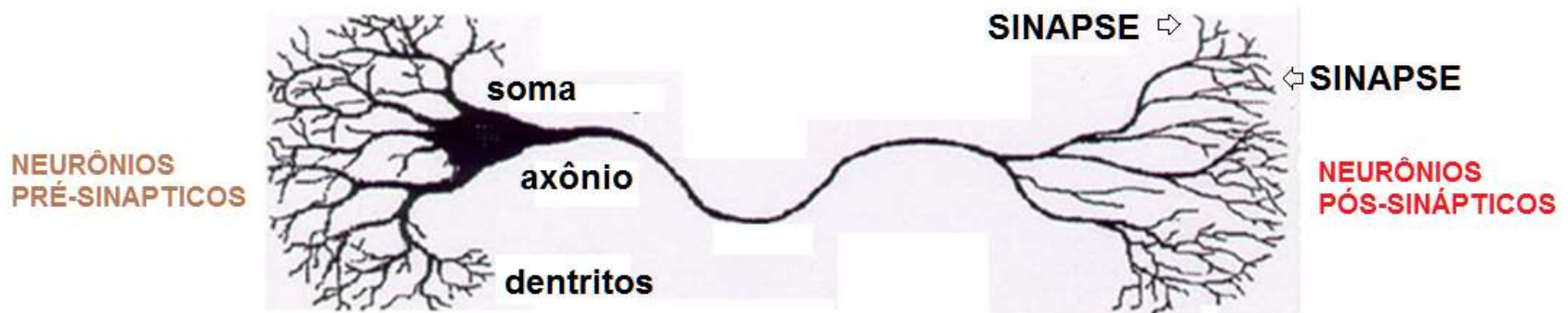
# NEURÔNIO BIOLÓGICO

- Sinapse é o ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro.
- É pelas sinapses que os neurônios se unem funcionalmente, formando as redes neurais.
- As sinapses funcionam como válvulas, sendo capazes de controlar a transmissão de impulsos, isto é, o fluxo da informação entre os nodos na rede neural.
- O efeito das sinapses é variável, é esta variação que dá ao neurônio capacidade de adaptação.



# NEURÔNIO BIOLÓGICO

- Os sinais oriundos dos neurônios *pré-sinápticos* são passados para o corpo do neurônio (soma), onde são comparados com outros sinais recebidos pelo neurônio. Se o percentual em um intervalo curto de tempo é suficientemente alto, a célula “dispara”, produzindo um impulso que é transmitido para as células seguintes (neurônios *pós-sinápticos*)





# CÉREBRO HUMANO

- Este sistema simples é responsável pela maioria das funções realizadas pelo nosso cérebro. A capacidade de solucionar funções complexas surge com a operação em paralelo de todos os  $10^{11}$  nodos do nosso cérebro.



# MOTIVAÇÃO

Constatação que o cérebro processa informações de forma diferente dos computadores convencionais

## CÉREBRO

velocidade 1 milhão de vezes mais lenta que qualquer "gate" digital → processamento extremamente rápido no reconhecimento de padrões



## COMPUTADOR

processamento extremamente rápido e preciso na execução de seqüência de instruções → muito mais lento no reconhecimento de padrões



**Processamento altamente paralelo  
( $10^{11}$  neurônios com  $10^4$  conexões cada)**

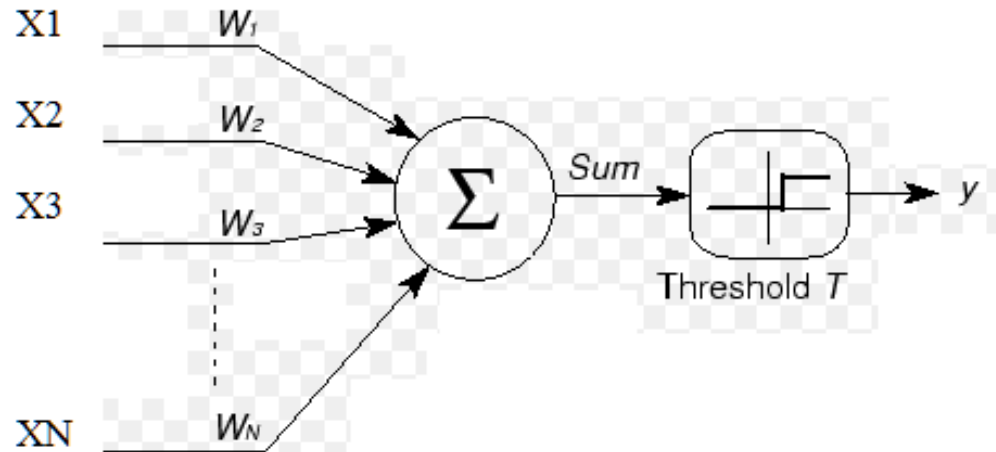
# NEURÔNIOS ARTIFICIAIS

## (Modelo McCulloch e Pitts)

- Modelo proposto por McCulloch e Pitts (MCP) em 1943.
- É uma simplificação do que se sabia à respeito do neurônio.
- McCulloch era psiquiatra e neuroanatomista e passou cerca de 20 anos refletindo e estudando sobre a representação do sistema nervoso
- Em 1942 ele convidou Pitts, que era matemático, para fazer parte das suas pesquisas.
- Os dois foram os primeiros a descreverem um modelo artificial para um neurônio biológico em 1943.

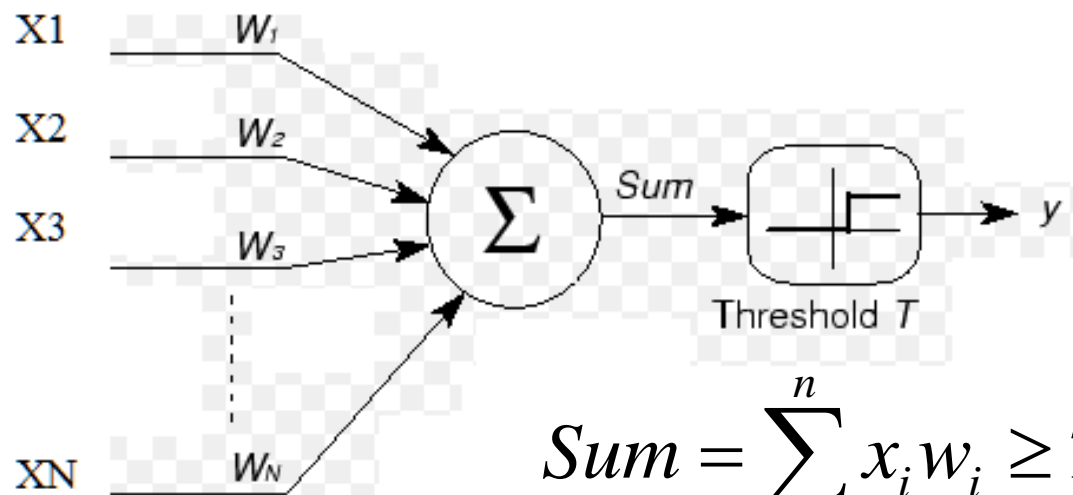
# NEURÔNIOS ARTIFICIAIS (Modelo McCulloch e Pitts)

- Terminais de entrada  $x_1, x_2, \dots, x_n$  (que representam os dendritos)
- Terminal de saída  $y$  (que representa o axônio).
- Emulação das Sinapses: os terminais de entrada do neurônio têm pesos acoplados  $w_1, w_2, \dots, w_n$ . O efeito de uma sinapse particular  $i$  no neurônio pós-sináptico é dado por:  $x_i w_i$ . Os pesos determinam “em que grau” o neurônio deve considerar sinais de disparo que ocorrem naquela conexão.



# NEURÔNIOS ARTIFICIAIS (Modelo McCulloch e Pitts)

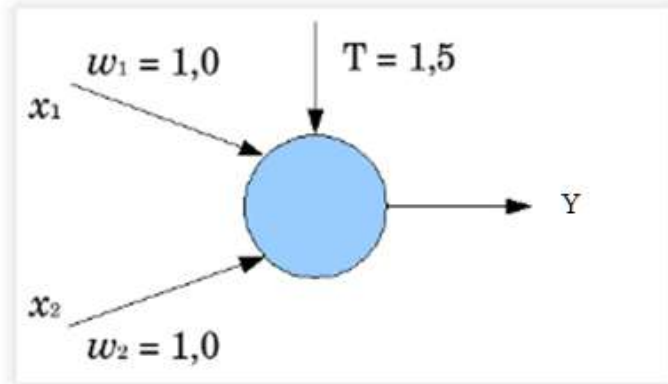
- No modelo MCP, a ativação do neurônio é obtida através da aplicação de uma “função de ativação”, que ativa a saída ( $Y=1$ ) ou não ( $Y=0$ ), dependendo do valor da soma ponderada das suas entradas. Na descrição original do modelo MCP, a função de ativação é dada pela função limiar descrita a seguir. O nodo MCP terá então sua saída ativa quando.



$$Sum = \sum_{i=1}^n x_i w_i \geq T$$

# NEURÔNIOS ARTIFICIAIS (Modelo McCulloch e Pitts)

- Com este modelo é possível representar operações Booleanas importantes.
- Na figura abaixo, os pesos ( $w_1$  e  $w_2$ ) e o limiar  $T$  são ajustados para responder à operação AND (mostrada na tabela verdade logo em seguida).
- Se tomarmos como exemplo  $x_1 = 0$  e  $x_2 = 1$ , a soma ponderada será  $Soma = (0 \times 1) + (1 \times 1) = 1$  e quando comparada com  $T$ , o resultado será  $Y=0$  ( $soma = 1 < 1.5$ )  $\rightarrow Y = 0$ .



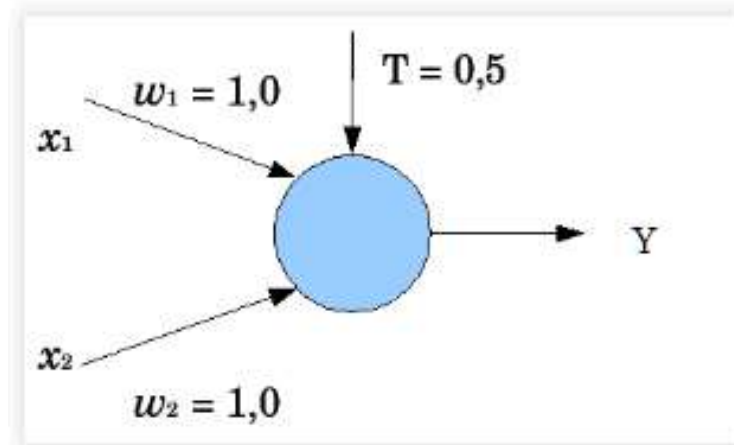
Neurônio que representa a operação AND

Operação AND		
A	B	Saída
0	0	0
0	1	0
1	0	0
1	1	1

Tabela verdade para AND

# NEURÔNIOS ARTIFICIAIS (Modelo McCulloch e Pitts)

- O mesmo pode ser feito com a operação lógica OR, basta simplesmente alterar o valor de T para 0.5.



Neurônio que representa a operação OR

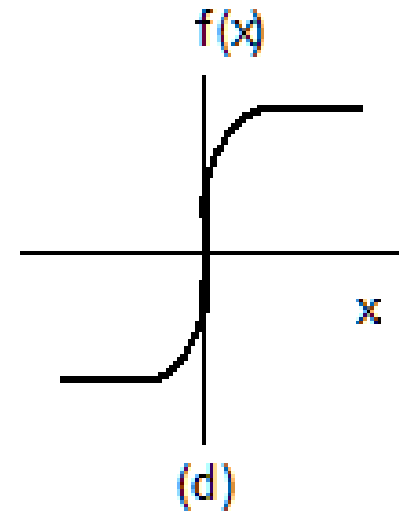
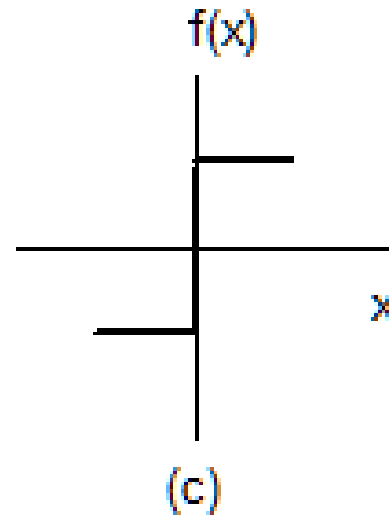
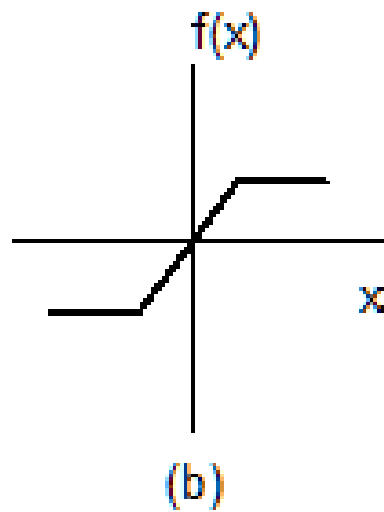
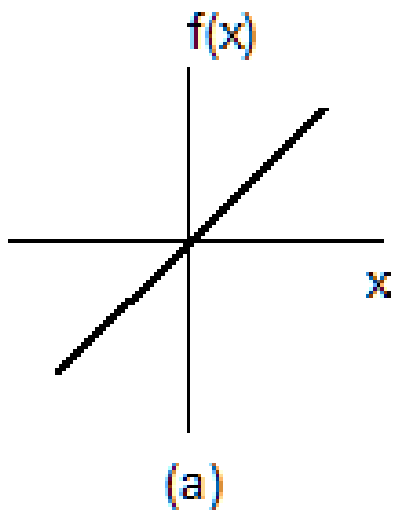
- Se tomarmos como exemplo  $x_1 = 0$  e  $x_2 = 1$ , a soma ponderada será  $Soma = (0 \times 1) + (1 \times 1) = 1$  e quando comparada com T, o resultado será  $1 (1 < 0.5) \rightarrow Y = 1$ .

EXERCÍCIO NO MOODLE



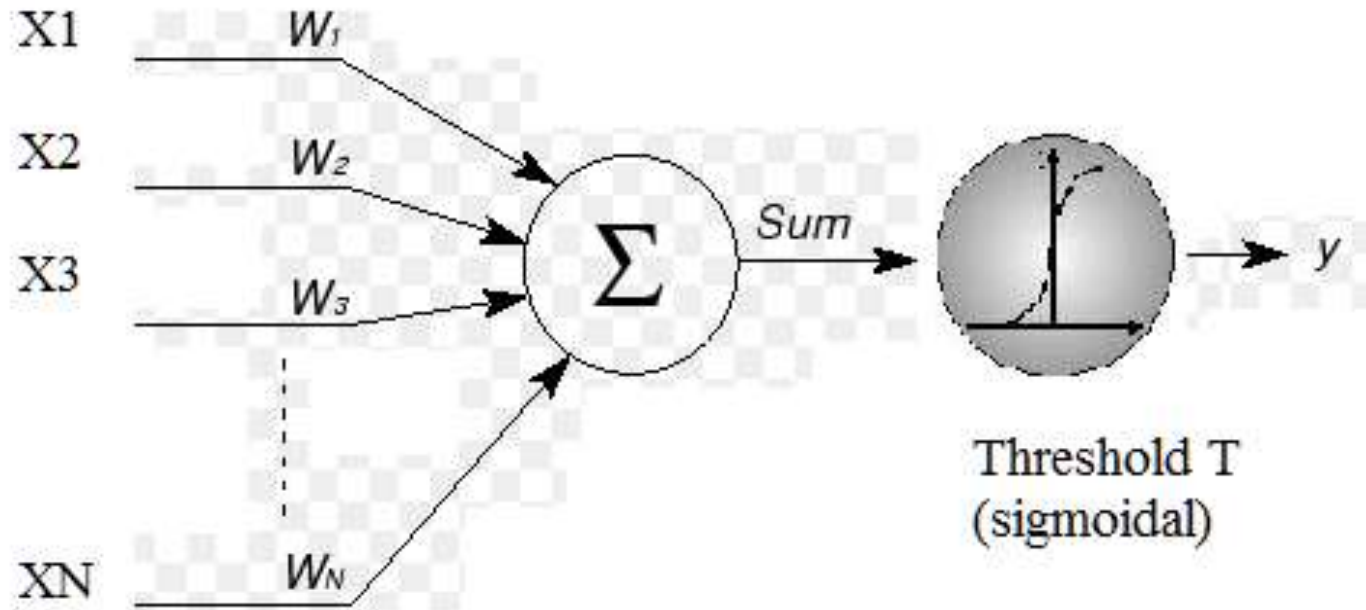
# FUNÇÕES DE ATIVAÇÃO

- A partir do modelo proposto por McCulloch e Pitts foram derivados outros modelos que permitem a produção de uma saída qualquer, não necessariamente zero ou um usando iferentes funções de ativação.



**(a) Função Linear; (b) Função semilinear; (c) Função degrau e (d) Função Sigmoideal**

# NEURÔNIO COM FUNÇÃO DE ATIVAÇÃO SIGMOIDAL

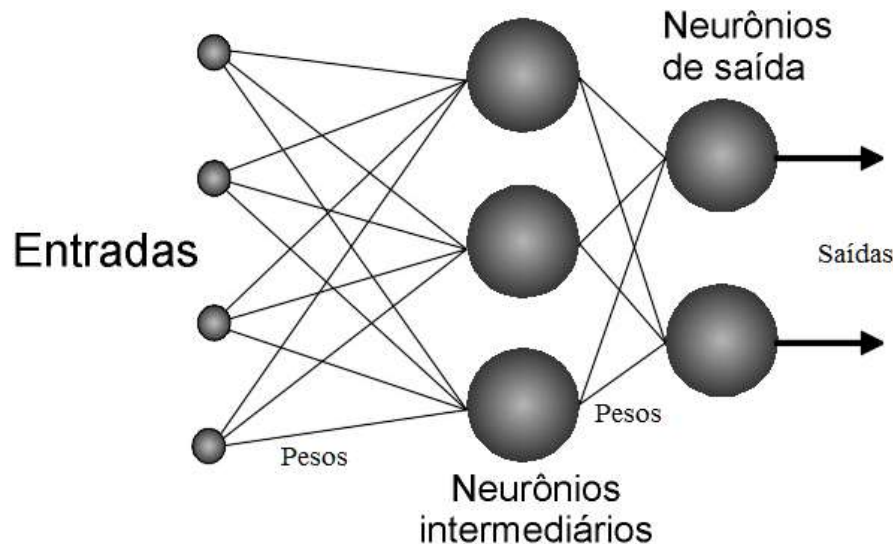


$$y = \frac{1}{1 + e^{-Sum / K}}$$

$$Sum = \sum_{i=1}^n x_i w_i$$

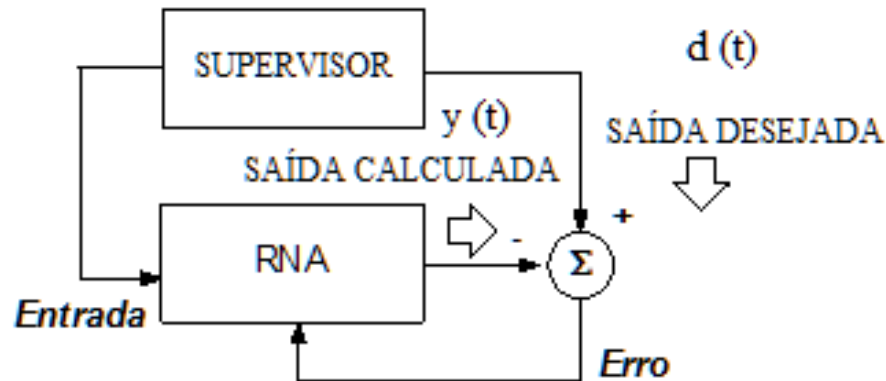
# APRENDIZADO

- As RNAs possuem a capacidade de aprender através exemplos.
- Um ALGORITMO DE APRENDIZADO é um conjunto de procedimentos para adaptar os pesos das conexões entre as unidades de processamento, a fim de que a mesma possa *aprender* uma determinada função.
- A utilização de uma RNA na solução de uma tarefa passa, inicialmente, por uma fase de aprendizagem, um processo iterativo de ajuste dos pesos realizados por meio de um algoritmo de aprendizado.



# APRENDIZADO SUPERVISIONADO

- A saída desejada que a rede deve resultar é fornecida por um supervisor (professor) externo.
- O supervisor indica, explicitamente, um comportamento bom ou ruim para a rede, visando direcionar o processo de treinamento.
- A rede tem sua saída corrente (calculada) comparada com a saída desejada, recebendo informações do supervisor sobre o erro da resposta atual.
- A cada padrão de entrada submetido à rede compara-se a resposta desejada com a resposta calculada, e os pesos das conexões são ajustados para minimizar o erro.
- Pequenos ajustes são feitos nos pesos à cada etapa de treinamento, de tal forma que estes caminhem, se possível, para uma solução.

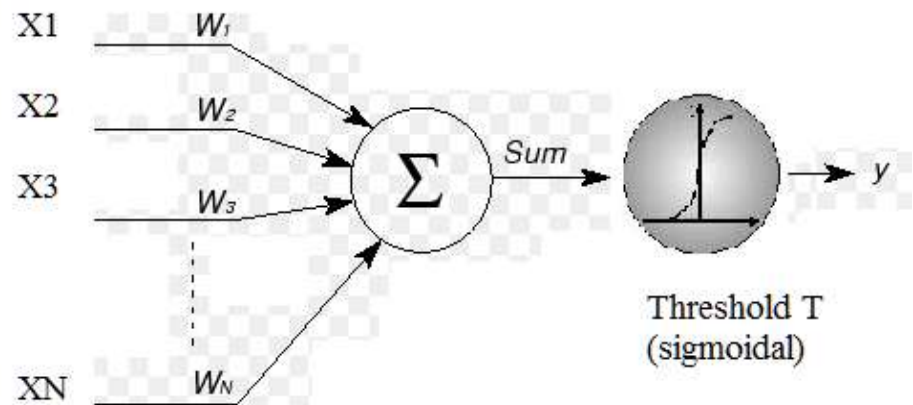
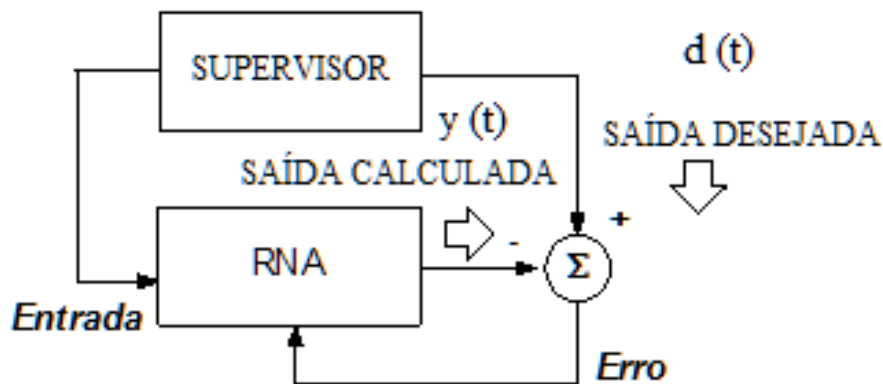


# CORREÇÃO DOS ERROS

- O processo de aprendizado por correção de erros procura minimizar a diferença entre a saída calculada pela rede e a saída desejada, ou seja, o erro da resposta atual da rede.
- O termo  $e(t)$  denota o erro e deve ser calculado através da seguinte expressão:  $e(t) = d(t) - y(t)$ , onde  $d(t)$  é a saída desejada e  $y(t)$  é a resposta atual (calculada) no instante de tempo  $t$ .
- A forma genérica para alteração dos pesos por correção de erros é definida por:

$$w_i(t+1) = w_i(t) + \eta e(t) x_i(t)$$

onde  $\eta$  é a taxa de aprendizado e  $x_i(t)$  é a entrada para o neurônio  $i$  no tempo  $t$ .

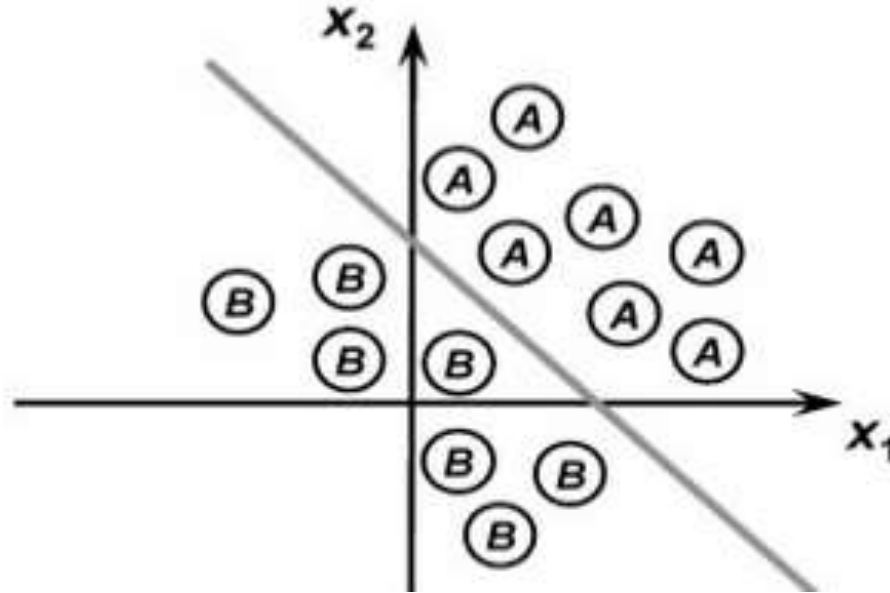


## PERCEPTRON DE UMA CAMADA E SEU ALGORITMO DE APRENDIZADO

- O modelo conhecido como perceptron, era composto por uma estrutura de rede tendo como unidades básicas nodos MCP e uma regra de aprendizado que utiliza o algoritmo de correção de erros

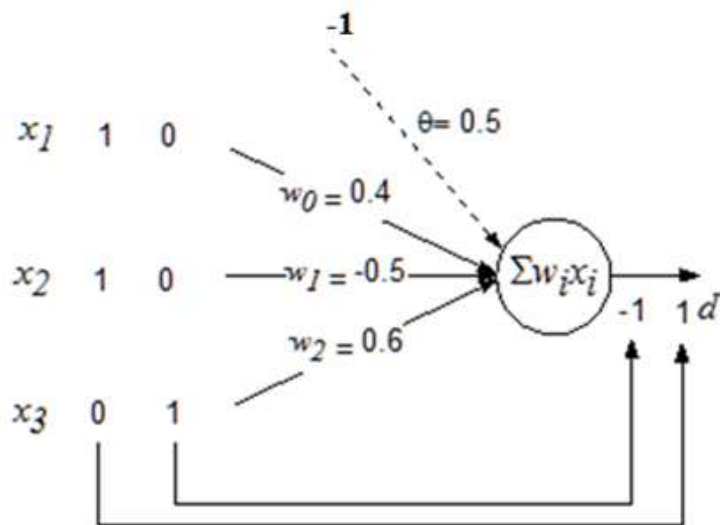
$$w_i(t+1) = w_i(t) + \eta e(t) x_i(t)$$

- Ele só converge no caso de problemas linearmente separáveis ( a operação XOR não é linearmente separável)



## PERCEPTRON DE UMA CAMADA (EXEMPLO)

- Dada uma rede do tipo perceptron formada por 3 terminais de entrada utilizando pesos iniciais de  $w_0 = 0.4$ ,  $w_1 = -0.5$ ,  $w_2 = 0.6$ ; limiar de ativação ( $T$ ) =  $0.5$  e a taxa de aprendizado  $\eta = 0.4$ .
- TREINAR a rede:
  - Gerar a saída  $-1$  para o padrão 001 | Gerar saída de  $1$  para o padrão 110.
- Após a rede treinada, determinar a que classe pertence os padrões 111, 000, 100, 011.
- Solução:  $x_i$  = padrões de entrada. |  $d$  = valor desejada para saída após processamento.



Função de ativação

$$y = \begin{cases} +1 & \text{se } SOMA > 0 \\ -1 & \text{se } SOMA \leq 0 \end{cases}$$

## PERCEPTRON DE UMA CAMADA (EXEMPLO)

Processamento para o padrão 001:

$$SOMA = (-1 \cdot 0,5) + (0 \cdot 0,4) + (0 \cdot -0,5) + (1 \cdot 0,6)$$

$$SOMA = 0,1 \quad [\text{aplicar função de ativação}]$$

$y = 1$ , onde  $y$  representa a saída do padrão processado

$$SOMA = 0,1 > 0 \Rightarrow Y=1$$

Como a saída desejada é diferente da saída calculada, aplica-se o algoritmo de correção de erro

$$\Delta w_i = w_i + \eta e x_i \quad | \quad e = (d - y)$$

$$\Delta w_3 = 0,5 + 0,4 \cdot (-1 - (1)) \cdot -1$$

$$\Delta w_3 = 1,3$$

$$\Delta w_0 = 0,4 + 0,4 \cdot (-1 - (1)) \cdot 0$$

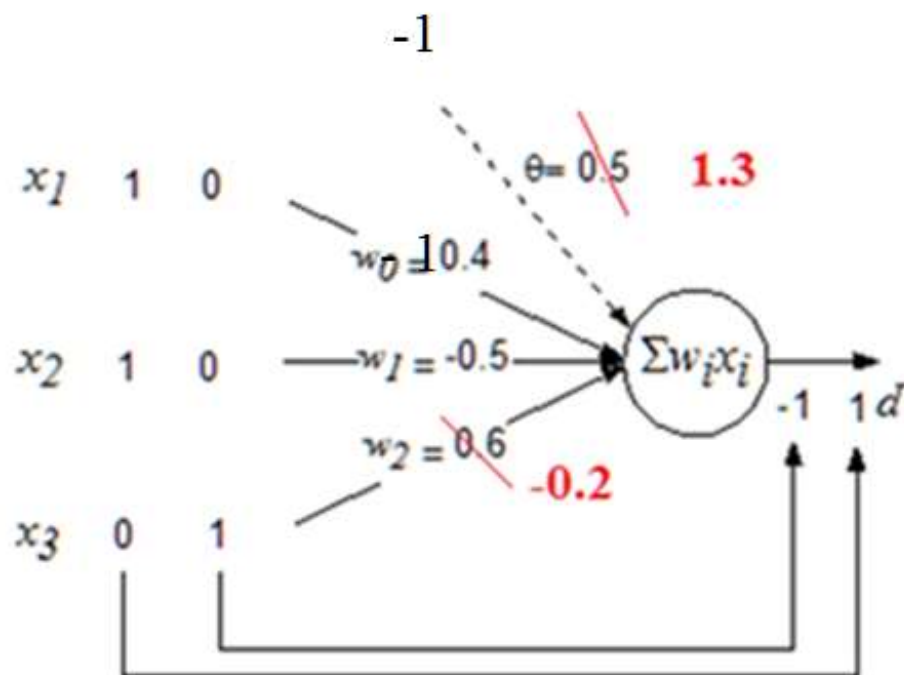
$$\Delta w_0 = 0,4$$

$$\Delta w_1 = -0,5 + 0,4 \cdot (-1 - (1)) \cdot 0$$

$$\Delta w_1 = -0,5$$

$$\Delta w_2 = 0,6 + 0,4 \cdot (-1 - (1)) \cdot 1$$

$$\Delta w_2 = -0,2$$





## PERCEPTRON DE UMA CAMADA (EXEMPLO)

Com os pesos atualizados, processa-se o padrão de entrada novamente

**Processamento para o padrão 001:**

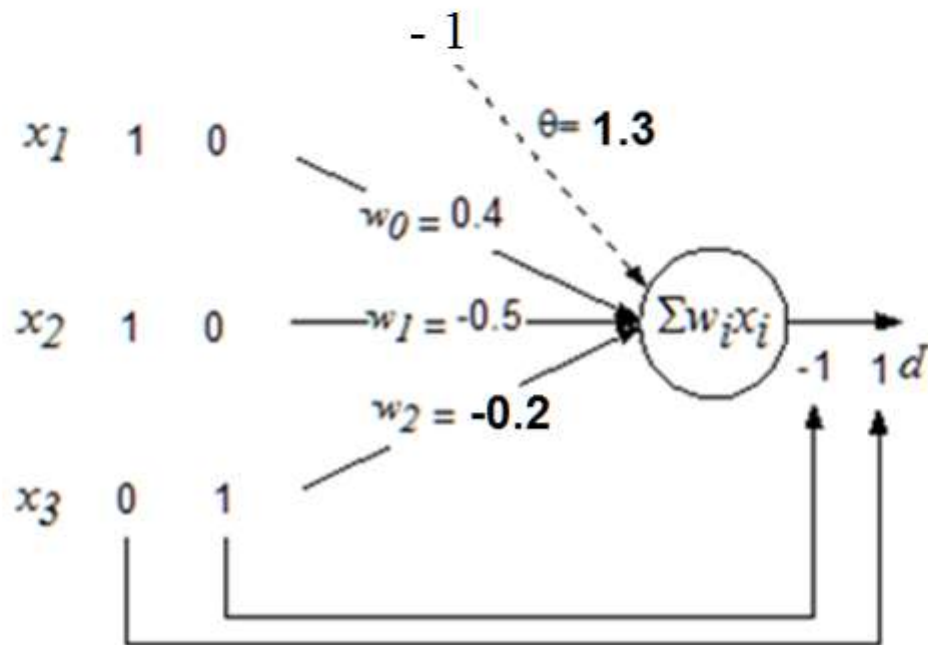
$$SOMA = (-1 \cdot 1,3) + (0 \cdot 0,4) + (0 \cdot -0,5) + (1 \cdot -0,2)$$

$$SOMA = -1,5 \text{ [aplicar a função de ativação]}$$

$$SOMA = -1,5 < 0 \Rightarrow Y = -1$$

$$y = -1$$

Como a saída desejada é igual a obtida pela rede, não é mais necessário a atualização dos pesos



### Processamento para o padrão 110:

$$SOMA = (-1 \cdot 1,3) + (1 \cdot 0,4) + (1 \cdot -0,5) + (0 \cdot -0,2)$$

$$SOMA = -1,4 \text{ [aplicar a função de ativação]}$$

$$y = -1 \quad \boxed{SOMA = -1,4 < 0 \Rightarrow Y = -1}$$

A saída está incorreta, portanto deve-se aplicar o algoritmo de treinamento

$$\Delta w_i = w_i + \eta e x_i \mid e = (d - y)$$

$$\Delta w_3 = 1,3 + 0,4 \cdot (1 - (-1)) \cdot -1$$

$$\Delta w_3 = 0,5$$

$$\Delta w_0 = 0,4 + 0,4 \cdot (1 - (-1)) \cdot 1$$

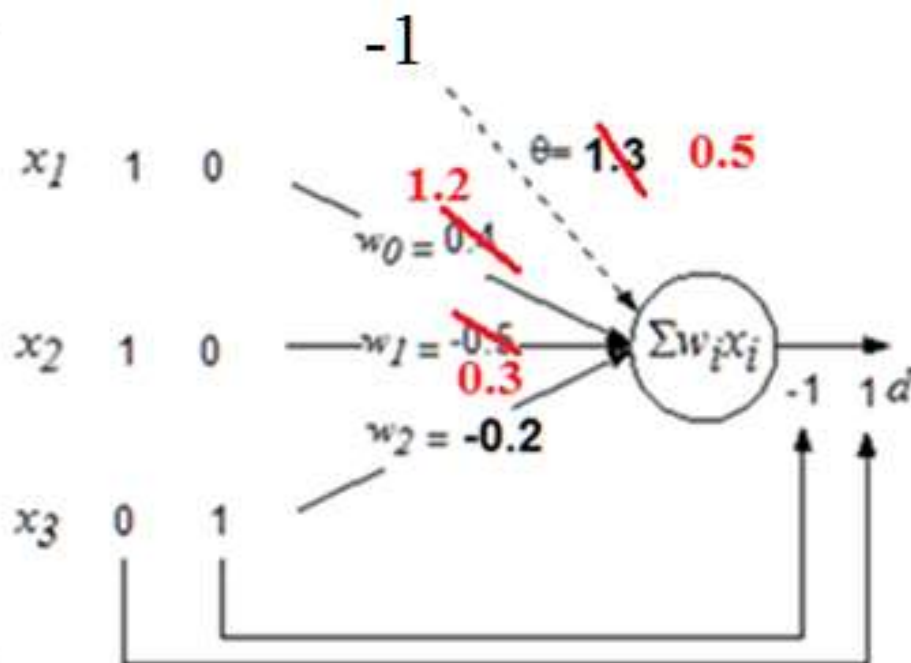
$$\Delta w_0 = 1,2$$

$$\Delta w_1 = -0,5 + 0,4 \cdot (1 - (-1)) \cdot 1$$

$$\Delta w_1 = 0,3$$

$$\Delta w_2 = -0,2 + 0,4 \cdot (-1 - (-1)) \cdot 0$$

$$\Delta w_2 = -0,2$$



Com os pesos atualizados processar o padrão de entrada novamente

Processamento para o padrão 110:

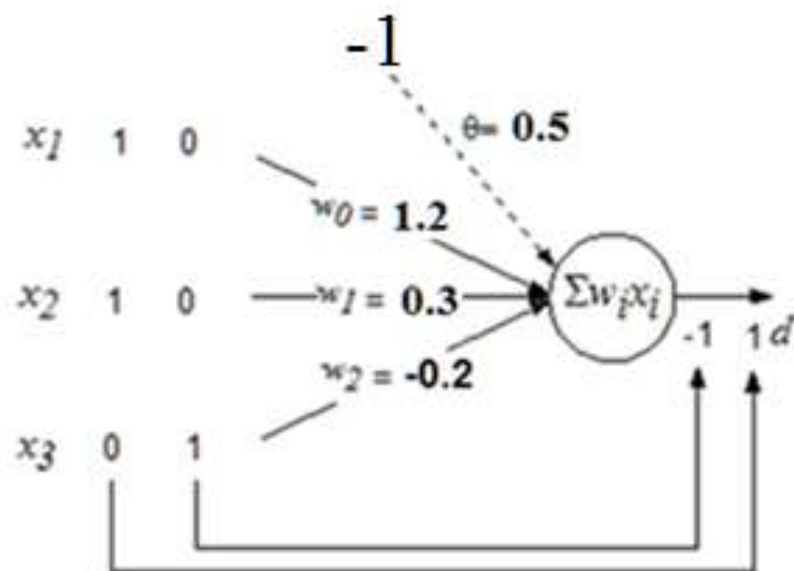
$$SOMA = (-1 \cdot 0,5) + (1 \cdot 1,2) + (1 \cdot 0,3) + (0 \cdot -0,2)$$

$$SOMA = 1 \quad [\text{aplicar a função de ativação}]$$

$$y = 1$$

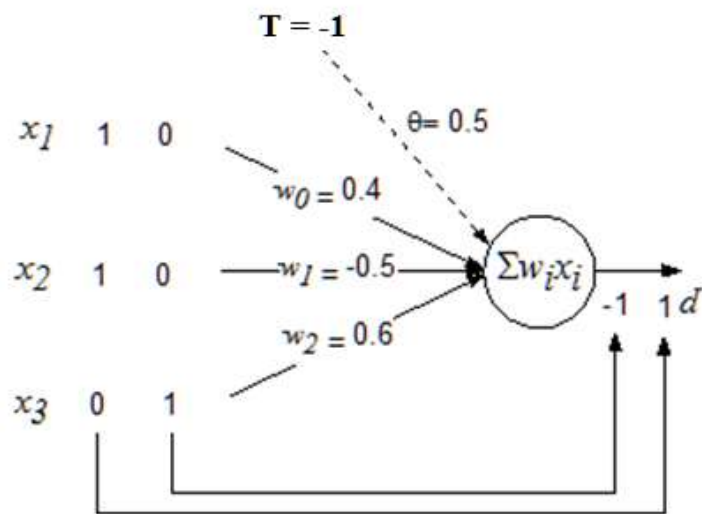
$$SOMA = 1 > 0 \implies Y = 1$$

Como a saída obtida pela rede é igual ao desejado, não é necessário atualização dos pesos

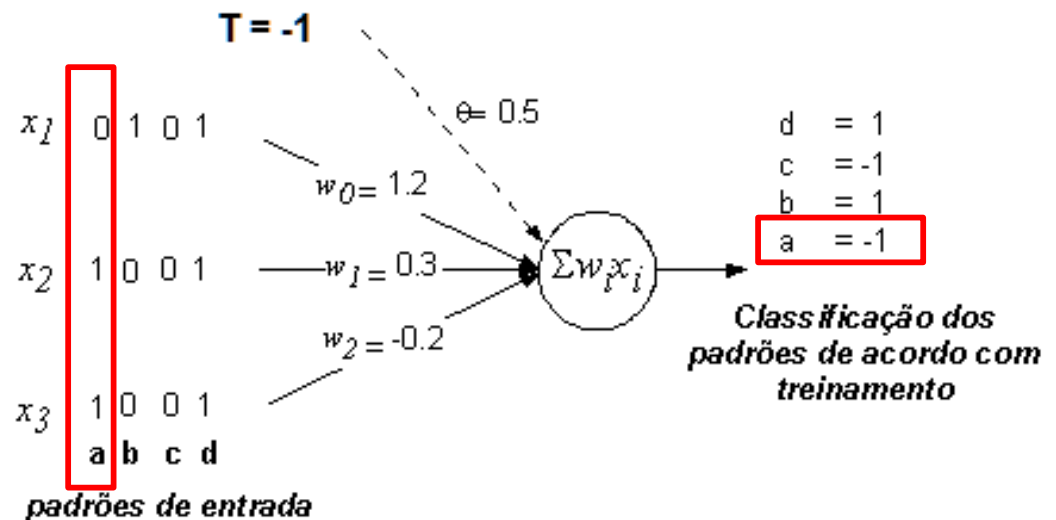


# OBTENÇÃO DE SAÍDAS

- Após o treinamento realizado, pode-se então classificar outros padrões de entrada (111, 000, 100, 011), obtendo as saídas 1, -1, 1, -1 respectivamente.



REDE INICIAL



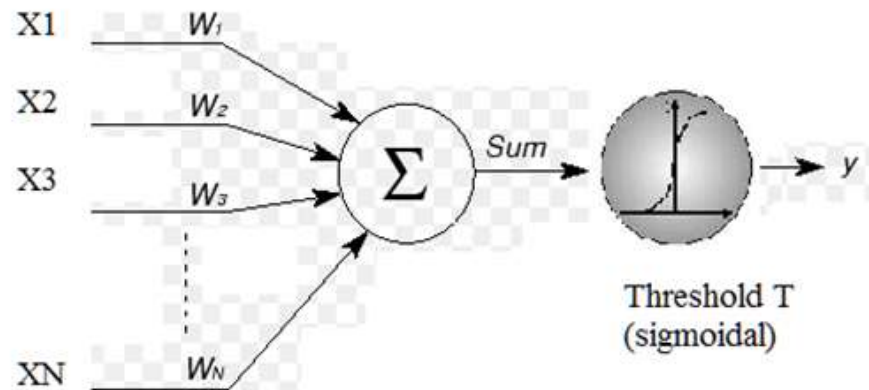
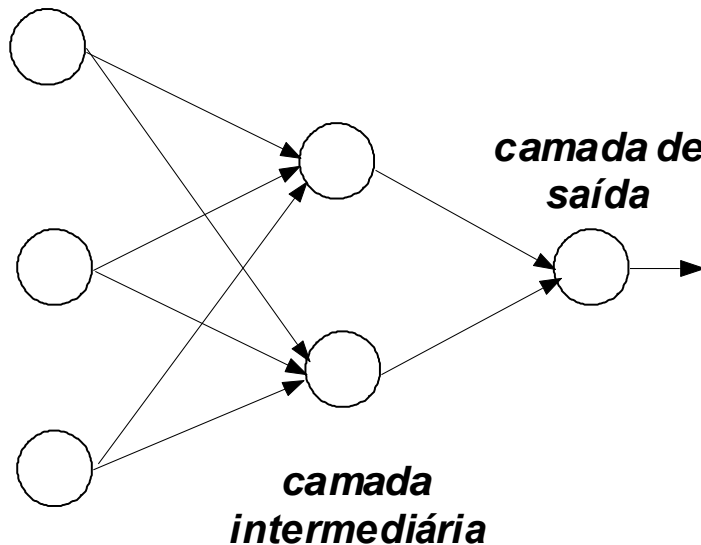
REDE TREINADA

EXERCÍCIO NO MOODLE

# REDE MULTIPLAYER PERCEPTRON

- Algoritmo de treinamento BACKPROPAGATION
- A função de ativação dos neurônios SIGMOIDAL

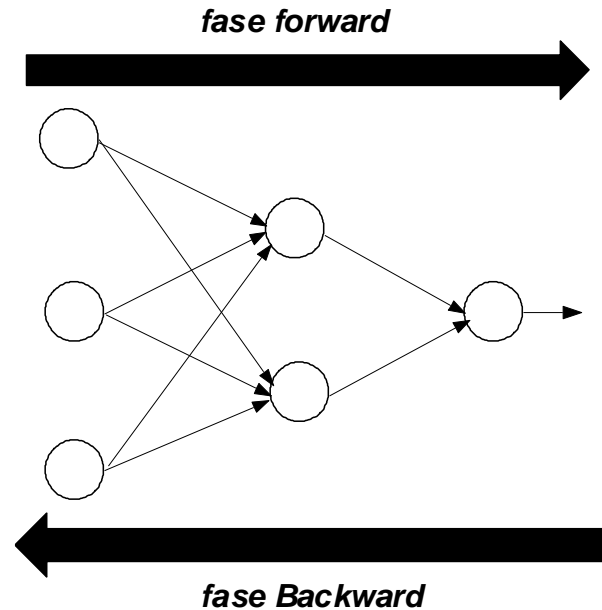
*camada de entrada*



$$Sum = \sum_{i=1}^N x_i w_i \Rightarrow y = \frac{1}{1 + e^{-Sum / K}}$$

# ALGORITMO BACKPROPAGATION

- É um algoritmo supervisionado, que utiliza pares (entrada, saída desejada) para, através de um mecanismo de correção de erros, ajustar os pesos da rede. O treinamento ocorre em duas fases, onde cada fase percorre a rede em um sentido. Estas duas fases são chamadas de fase *forward* e fase *backward*.



# ALGORITMO BACKPROPAGATION

O algoritmo *back-propagation*, que faz uso destas duas fases, é apresentado a seguir:

1 – Inicializar pesos e parâmetros

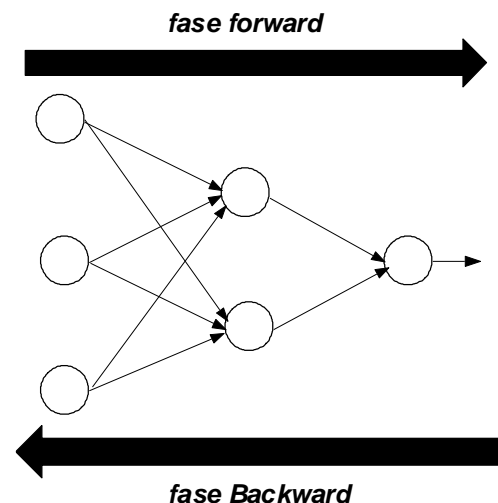
2 – Repita até o erro ser mínimo ou a realização de um dado número de ciclos:

2.1 – Para cada padrão de treinamento  $X$

2.1.1 – Definir saída da rede através de fase *forward*

2.1.2 – Comparar saídas produzidas com as saídas desejadas

2.1.3 – Atualizar pesos dos nodos através da fase *backward*





# PRÁTICA EM C++ USANDO REDE NEURAL MLP