

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Conhecimento e Raciocínio não Simbólico

Sistemas de Representação de Conhecimento e Raciocínio
MiEI/3º – 2º Semestre

- *"Every aspect of learning or any other feature of intelligence can principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves."*

John McCarthy, Dartmouth Conference, 1956



- Cada aspecto da aprendizagem ou qualquer outra característica da inteligência pode, em princípio, ser descrito de forma tão precisa que será possível construir uma máquina para o simular. Serão feitas tentativas para descobrir como fazer com que as máquinas usem a linguagem, formem abstrações e conceitos, resolvam tipos de problemas até agora reservados para os humanos, e sejam capazes de se melhorarem a si próprias.

Inteligência Artificial Contextualização

- A abordagem da Inteligência Artificial à representação de conhecimento expõe-se através de dois paradigmas:
 - Simbólico
 - Baseia-se na lógica para representar conhecimento;
 - Fundamenta o raciocínio na construção de sistemas de inferência;
 - Não simbólico, ou conexionista
 - Baseia o funcionamento do sistema na capacidade de aprender, generalizando;
 - Resolve problemas com base em conhecimento passado ou dados sobre a resolução de outros problemas;



Inteligência Artificial Contextualização

- **Vantagens da abordagem simbólica:**

- O problema da representação de conhecimento é simples;
- Grande poder expressivo das linguagens de representação;
- Mecanismos de inferência solidificados;

- **Vantagens da abordagem não simbólica:**

- Capacidade de aprender a resolver problemas;
- Capacidade de generalizar a resolução de problemas;
- ~ Tolerante a falhas



Soft Computing vs Hard Computing

Contextualização

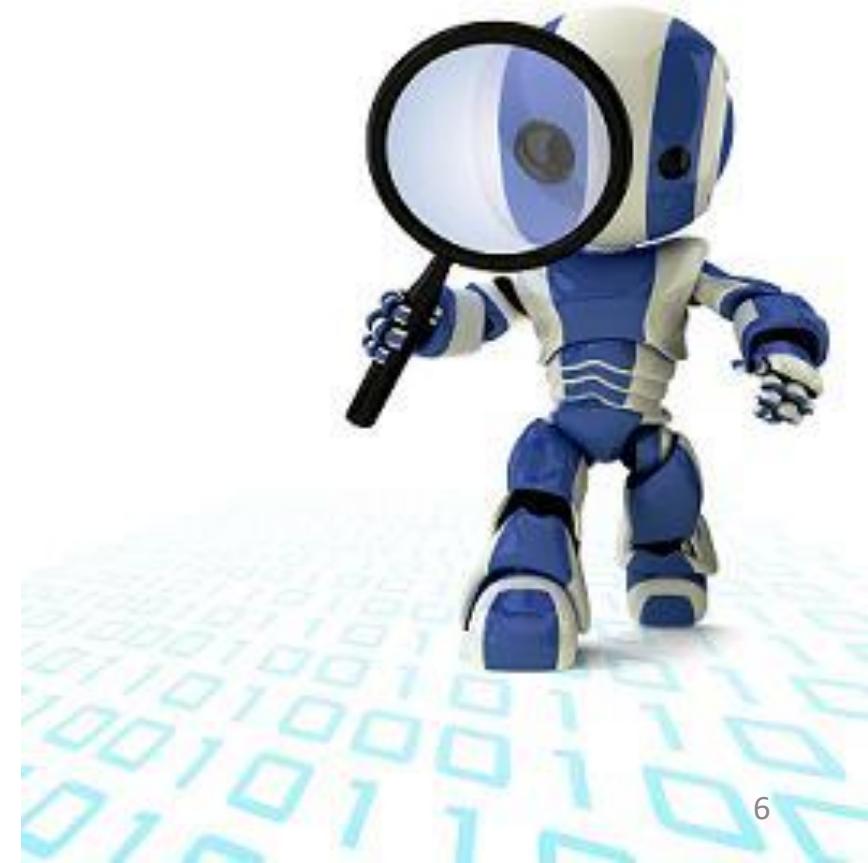
- **Soft Computing**
 - Computational Intelligence / Inteligência Computacional;
 - Natural Computing / Computação Natural;
- O modelo subjacente a “Soft Computing” é a mente humana;
- “Soft Computing” difere de “Hard Computing” no sentido em que é tolerante à imprecisão, à incerteza, à verdade parcial e à aproximação, explorando estas capacidades para alcançar, por estes meios, a exequibilidade em tempo oportuno.

<http://www.soft-computing.de/def.html>



- *Fuzzy Logic*
- Redes Neuronais Artificiais
- Computação Genética e Evolutiva
- Raciocínio Probabilístico
- *Particle Swarm Optimization*
- *Ant Colony*

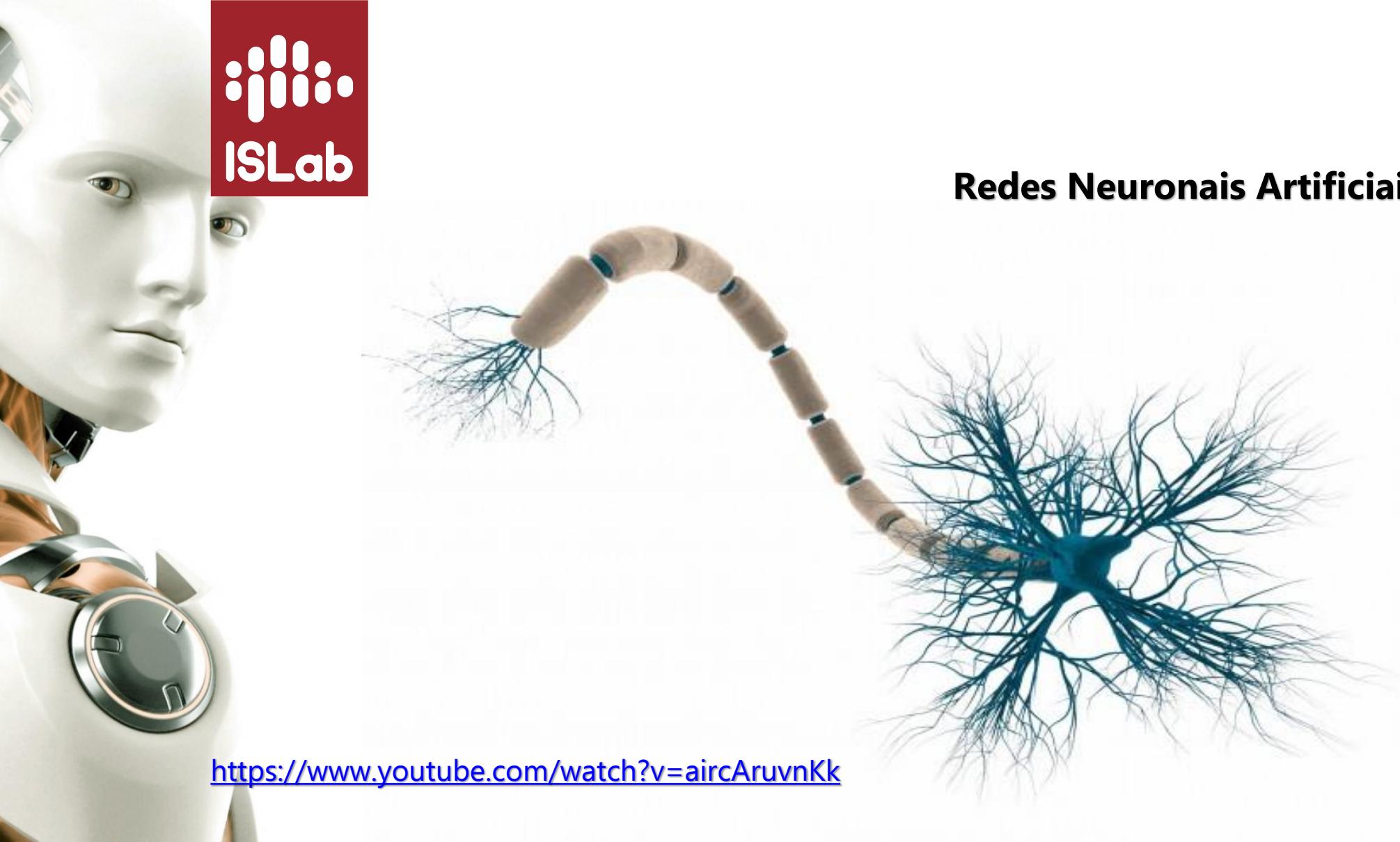
Sistemas de Soft Computing Contextualização





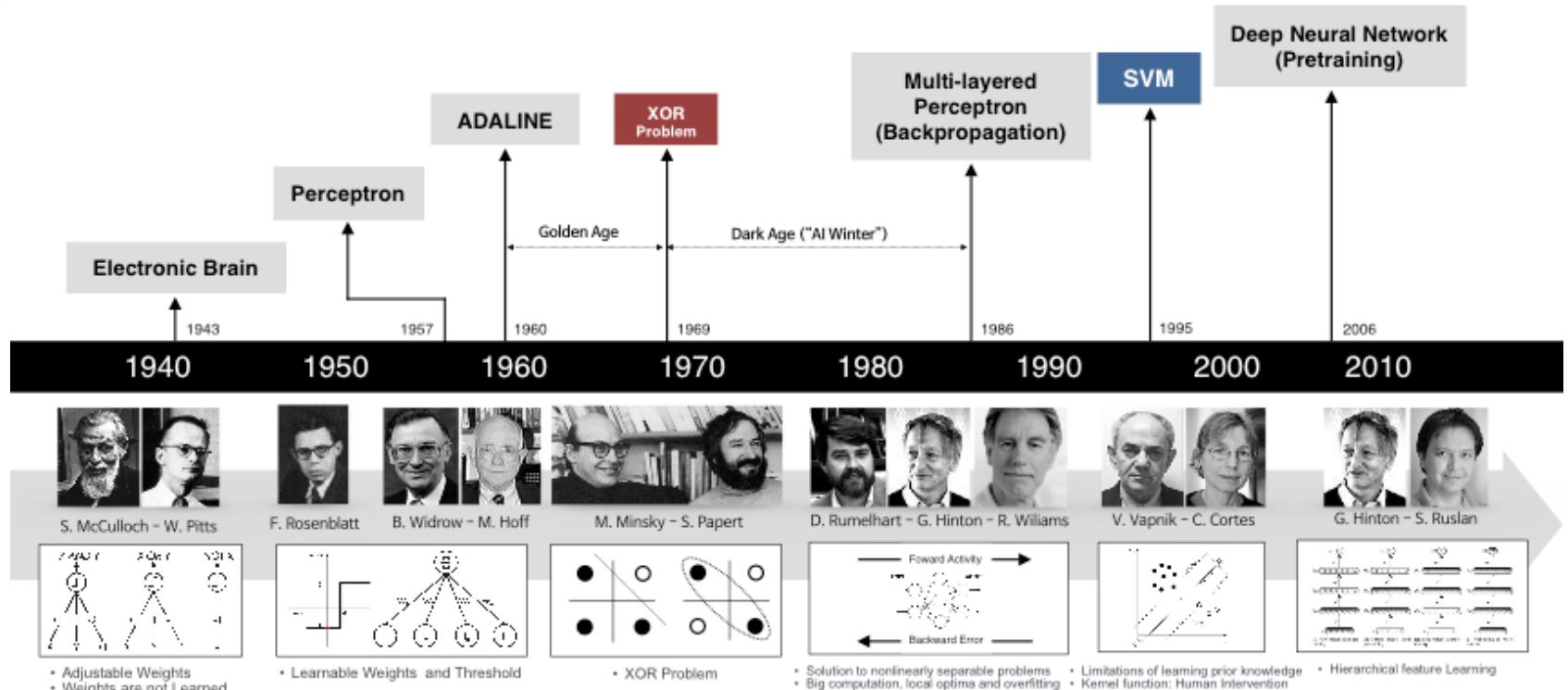
ISLab

Redes Neuronais Artificiais



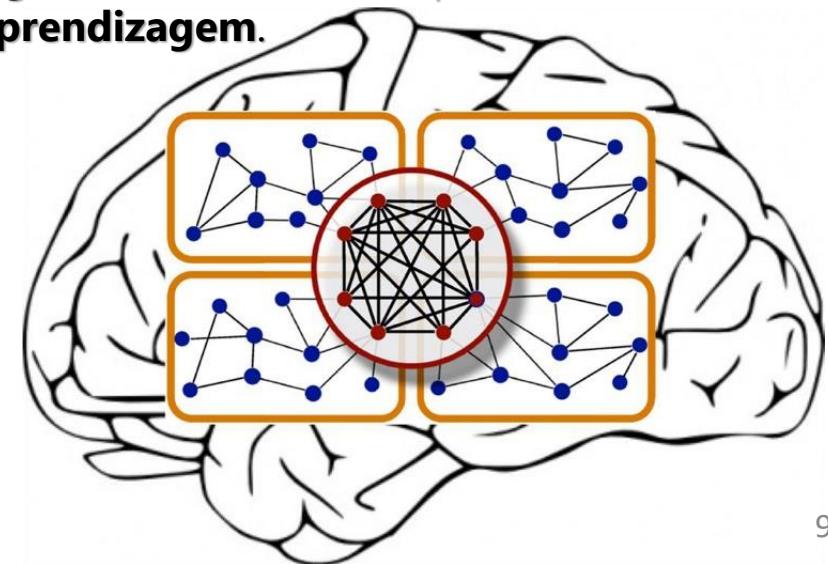
<https://www.youtube.com/watch?v=aircAruvnKk>

Evolução Redes Neuronais Artificiais



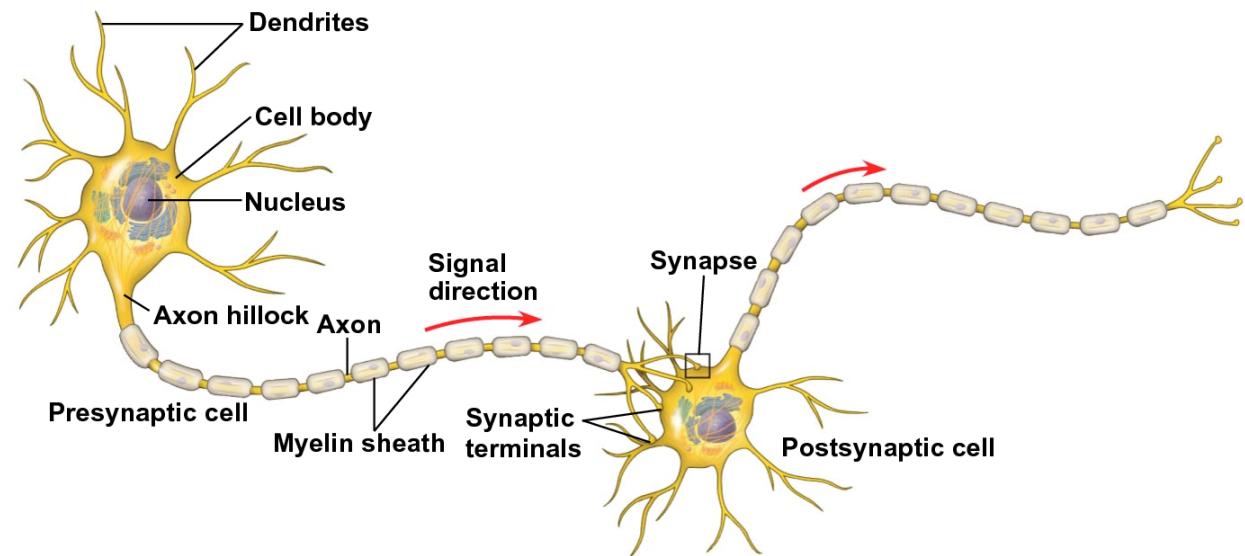
Definição Redes Neuronais Artificiais

- Uma **Rede Neuronal Artificial** (RNA) é um sistema computacional de base conexionista para a resolução de problemas.
- Uma RNA é concebida com base num **modelo simplificado do sistema nervoso central** dos seres humanos.
- Uma RNA é definida por uma estrutura interligada de unidades computacionais, designadas **neurónios**, com capacidade de **aprendizagem**.



Definição Redes Neuronais Artificiais

- Uma **Rede Neuronal Artificial** (RNA) é um sistema computacional de base conexionista para a resolução de problemas.
- Uma RNA é concebida com base num **modelo simplificado do sistema nervoso central** dos seres humanos.
- Uma RNA é definida designadas **neurónios**

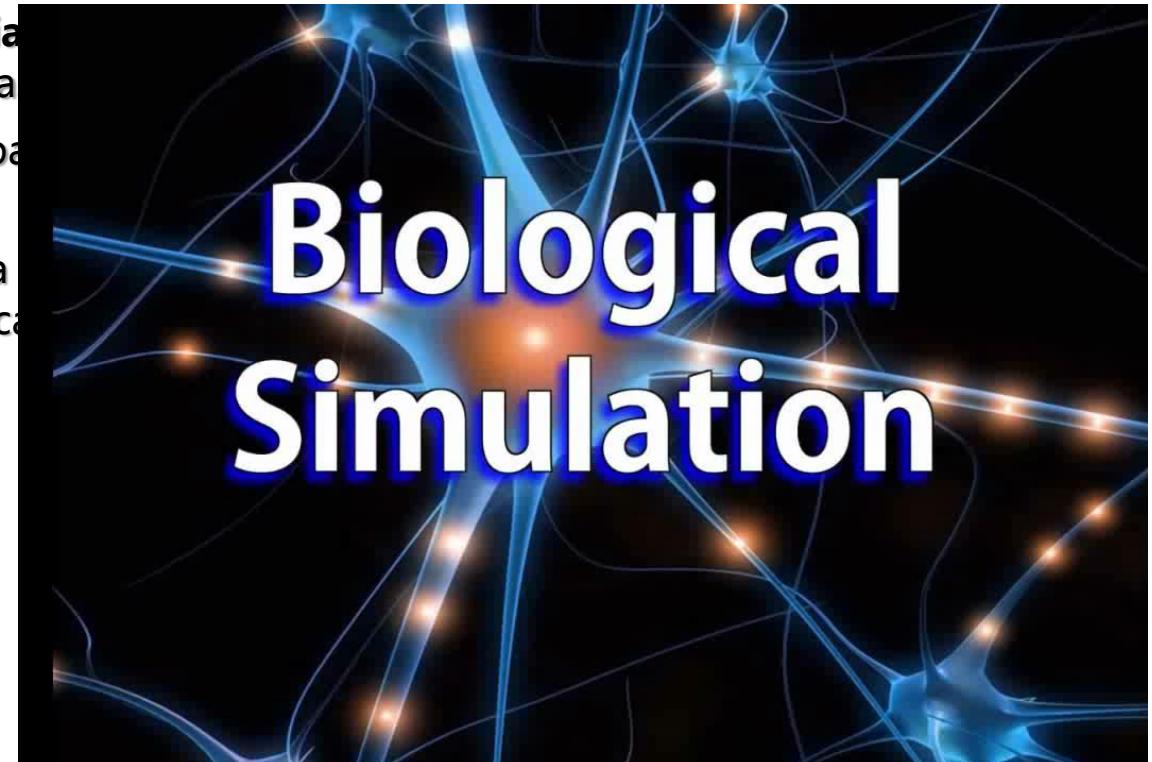




ISLab

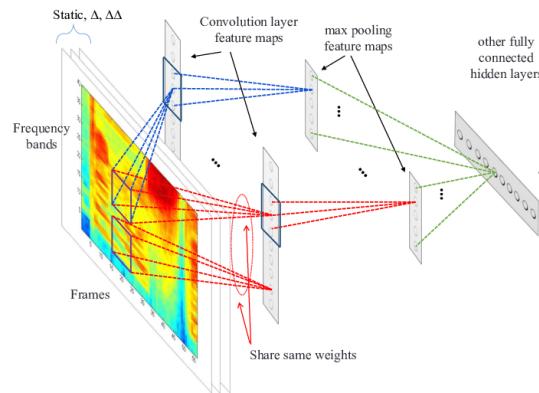
- Uma **Rede Neuronal Artificial** é uma estrutura de processamento de informação para a resolução de problemas.
- Uma RNA é concebida com base no funcionamento das estruturas nervosas dos seres humanos.
- Uma RNA é definida por uma estrutura de processamento de informação designadas **neurónios**, com capacidade de processar e transmitir informação.

Definição Redes Neuronais Artificiais

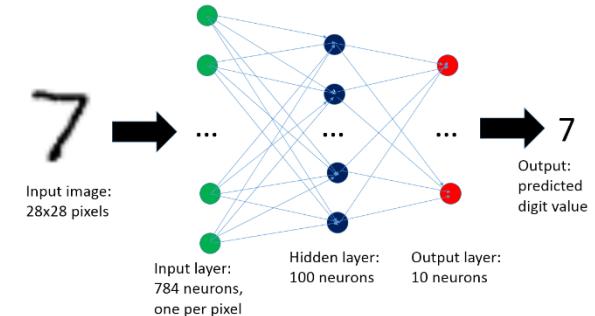


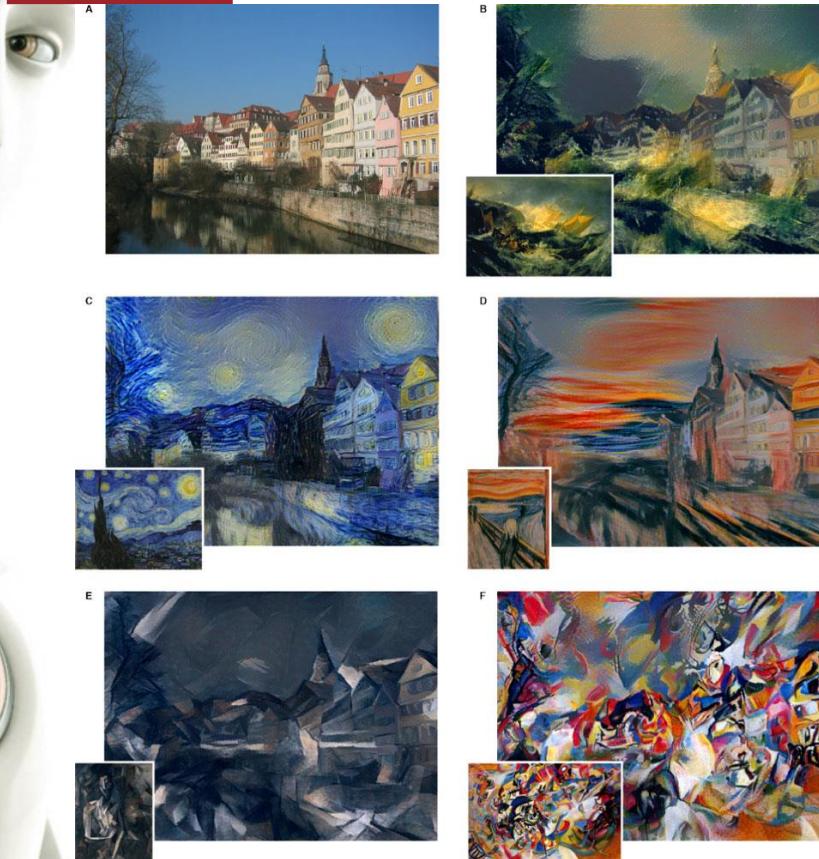
https://www.youtube.com/watch?v=gCK_5x2KsLA

- Reconhecimento de carateres
- Deteção de fraudes
- Reconhecimento de áudio/vídeo



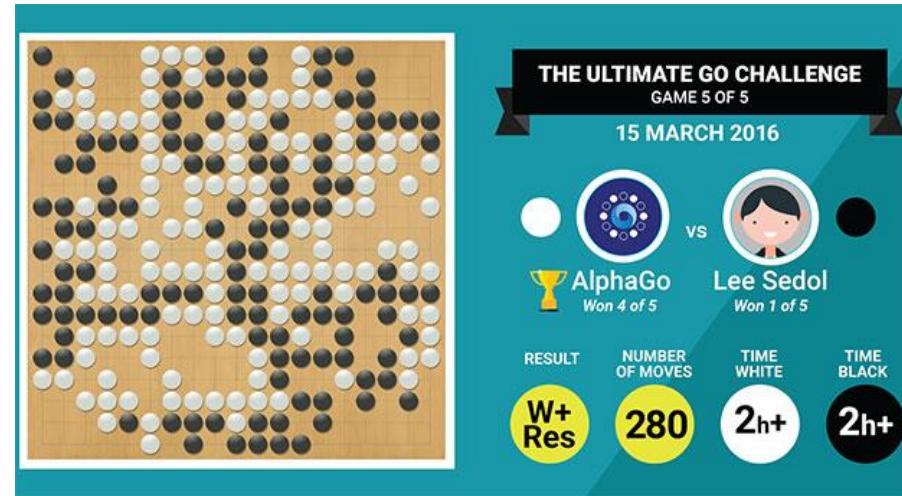
Por onde andam? Redes Neuronais Artificiais



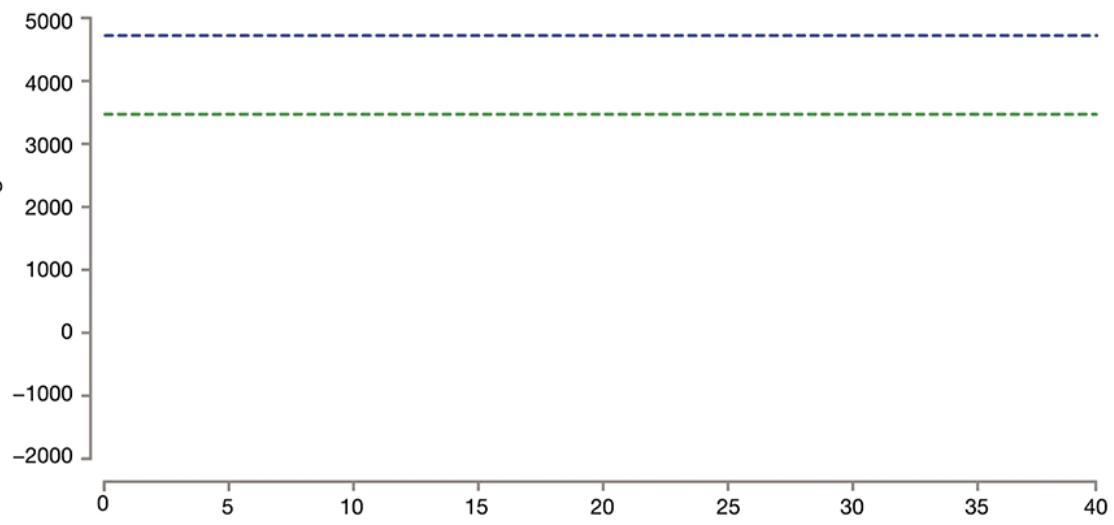


[https://www.boredpanda.es/algoritmo-aprendizaje-profundo-imita-pintura-maestros/...](https://www.boredpanda.es/algoritmo-aprendizaje-profundo-imita-pintura-maestros/)

Por onde andam? Redes Neuronais Artificiais



<https://deepmind.com/blog/alphago-zero-learning-scratch>



Por onde andam? Redes Neuronais Artificiais

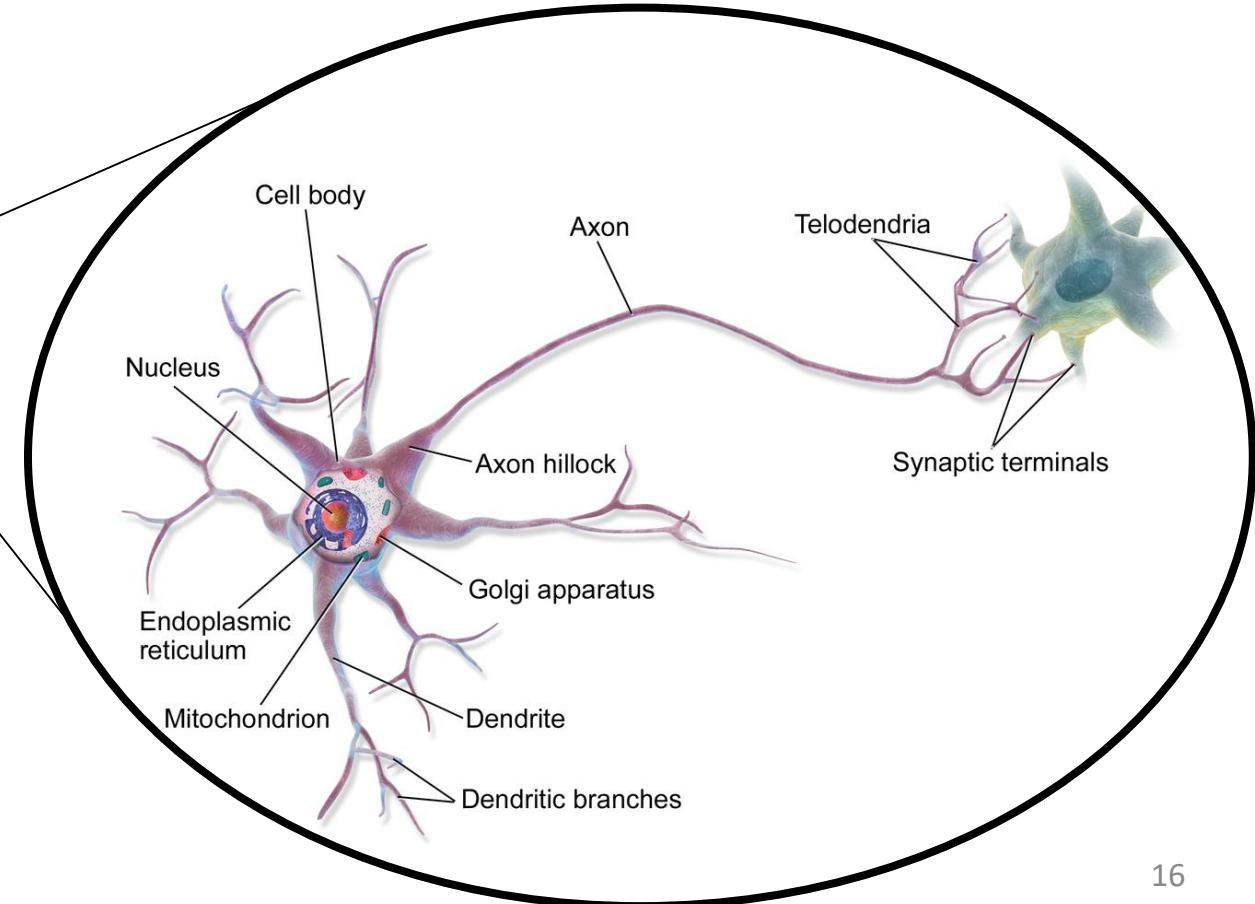
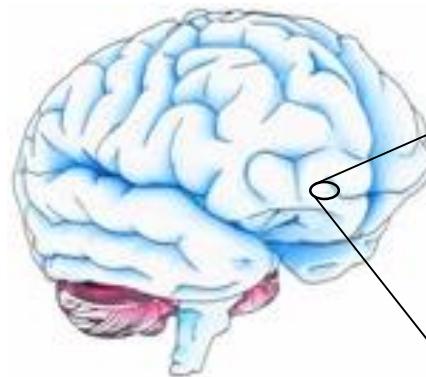
[https://www.boredpanda.es/algoritmo-aprendizaje-profundo-imita-pinturas-mas-1000/...](https://www.boredpanda.es/algoritmo-aprendizaje-profundo-imita-pinturas-mas-1000/)

O cérebro humano

- 100.000.000.000 neurónios
- 10.000 entradas por neurónio
- 1 sinal eletroquímico em cada neurónio
- Neurónios são conectados através de neurotransmissores químicos (dopamina, serotonina, glutamato ↑, gama-aminobutírico ↓)

- Representa 2% da massa do corpo humano
- Recebe 25% do sangue bombeado pelo coração

Conceitos e definições



Conceitos e definições Neurónio

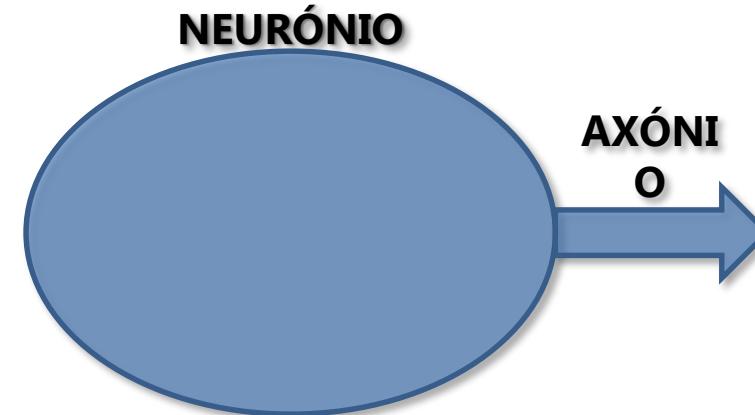
- **Unidade computacional** de composição da RNA.
- **Identificado** pela sua **posição** na rede.
- Caracterizado pelo **valor do estado**.



NEURÓNIO

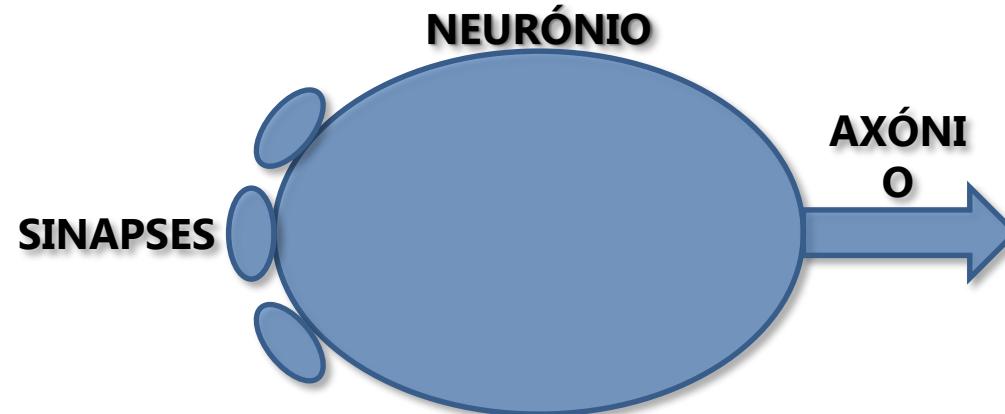
Conceitos e definições Axónio

- **Via de comunicação** entre os neurónios.
- Pode **ligar qualquer neurónio**, incluindo o próprio.
- As ligações podem **variar** ao longo do **tempo**.
- A informação circula em **um só sentido**.



Conceitos e definições Sinapses

- **Ponto de ligação** entre axónios e neurónios.
- O **valor da sinapse** determina o **peso** (importância) do sinal a entrar no neurónio: excitativo, inibidor ou nulo.
- A **variação no tempo** determina a **aprendizagem** da RNA.



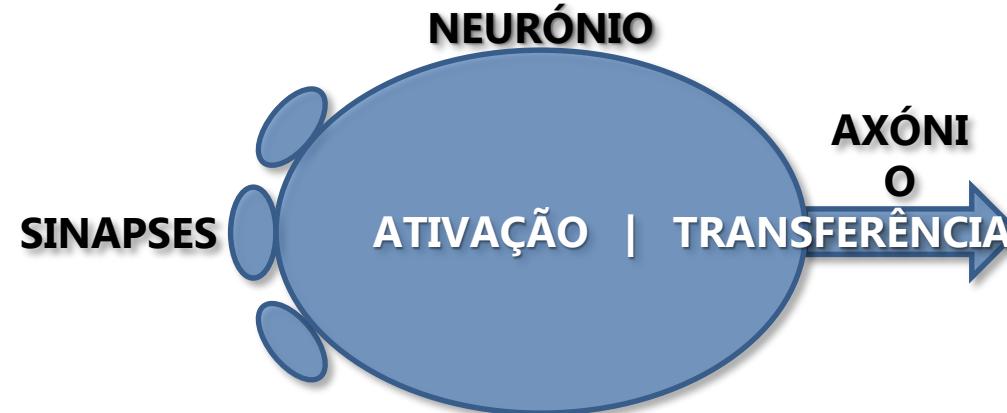
Conceitos e definições Ativação

- O valor de ativação é representado por **um único valor**.
- O valor de ativação **varia com o tempo**.
- A gama de valores varia com o modelo adotado
(normalmente está dependente das entradas e de algum efeito de memória).



Conceitos e definições Transferência

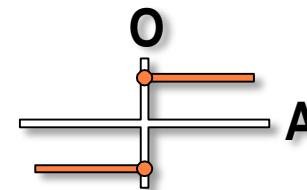
- O valor de transferência de um neurónio determina **o valor** que é **colocado na saída** (transferido através do axónio).
- É calculado como uma função do valor de ativação (eventualmente com algum efeito de memória).



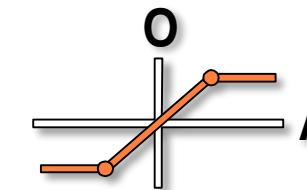
Tarefas dos neurónios

- Cálculo do valor de saída (output = O_i), função do valor de ativação, por uma função de transferência (f_T):

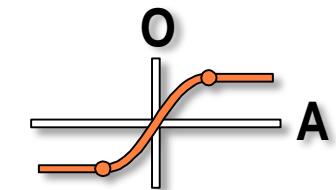
$$O_i = f_T(A_i)$$



Binária ou Escada



Linear



Sigmoid

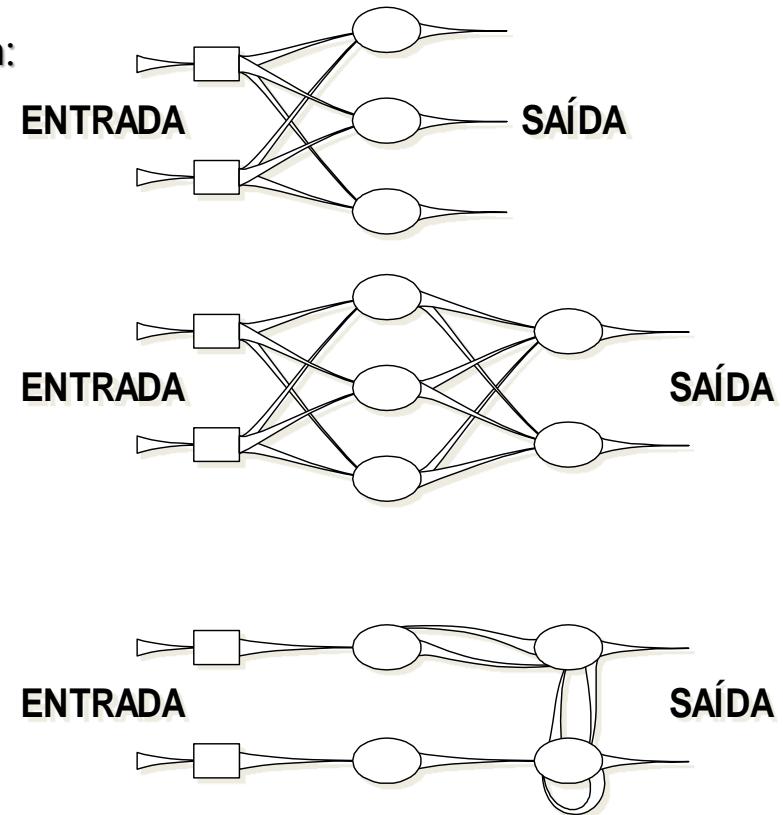
- Cálculo do valor de ativação (A_j).
- Varia no tempo com o seu próprio valor e o de outras entradas (w_i ; I):

$$A_j = F(A_{j-1}; I_j; \sum w_{ij} \times O_i)$$

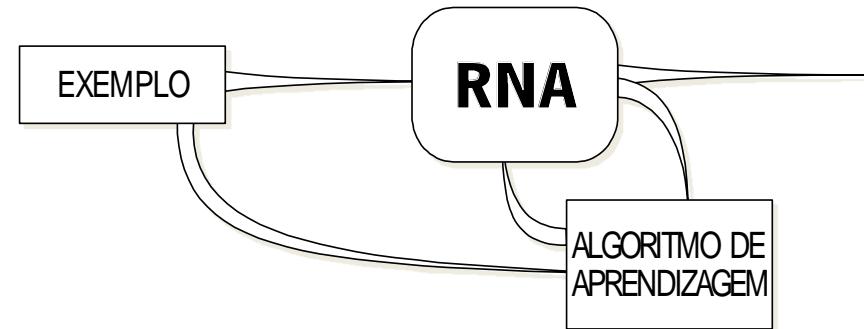
- **Aprendizagem:** regras de modificação dos pesos (w_i).

- Arquitetura *Feed forward*, de uma só camada:
(Perceptron)
- Arquitetura *Feed forward*, multi-camada:
(Multi-layer Perceptron)
- Arquitetura Recorrente

Organização dos neurónios

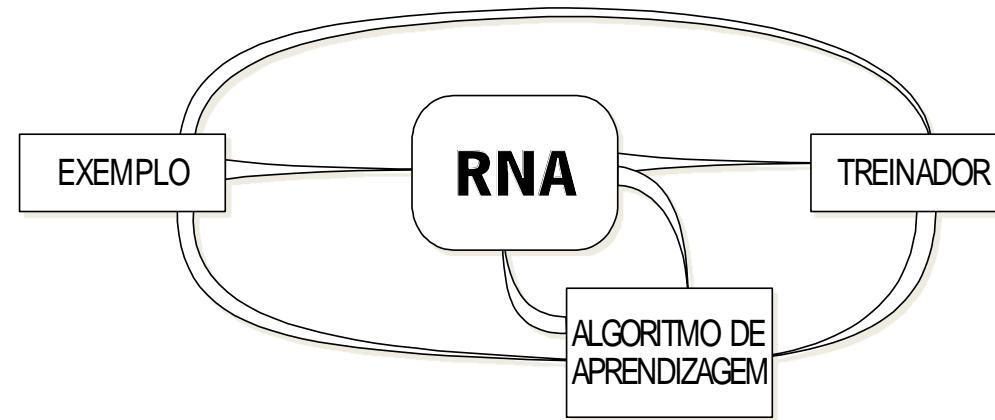


- Sem supervisão:



(p.ex., quando dois neurónios adjacentes têm variações da ativação no mesmo sentido, então o peso da ligação deve ser progressivamente aumentado.)

- Com supervisão:



(p.ex., os ajustes nos pesos das ligações são efetuados por forma a minimizar o erro produzido pelos resultados da RNA.)

- De reforço: o exemplo contém, apenas, uma indicação sobre a correção do resultado.

Regras de aprendizagem

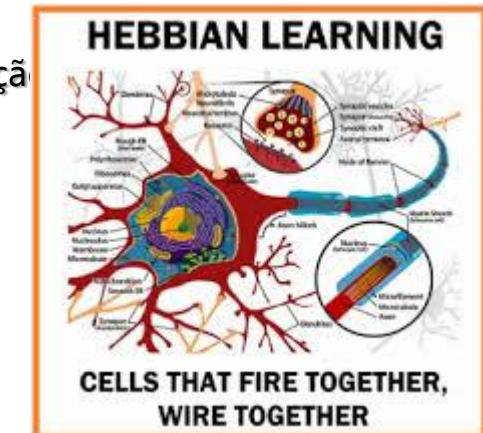
- O treino de uma RNA corresponde à aplicação de regras de aprendizagem, por forma a fazer variar os pesos das ligações (sinapses);



Regras de aprendizagem

- O treino de uma RNA corresponde à aplicação de regras de aprendizagem, por forma a fazer variar os pesos das ligações (sinapses);
 - Regras de aprendizagem mais comuns:
 - *Hebbian Learning Rule*
 - Desenvolvida por Donald Hebb em 1949 para o treino não supervisionado de RNAs;
 - Se dois neurónios adjacentes sofrem variações no mesmo sentido, o peso da ligação deve aumentar;
 - Se as variações acontecem em sentido oposto, o peso da ligação diminuir;
 - Não havendo variação, o peso deve manter-se inalterado;
 - Os pesos são inicializados a zero;

$$W_{ij} = x_i^* x_j$$



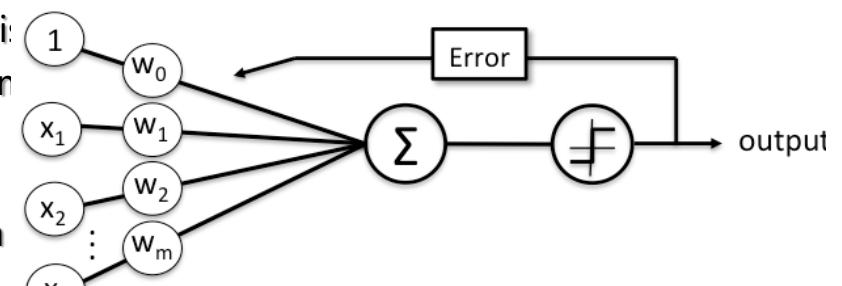
Regras de aprendizagem

- O treino de uma RNA corresponde à aplicação de regras de aprendizagem, por forma a fazer variar os pesos das ligações (sinapses);
- Regras de aprendizagem mais comuns:

- *Hebbian Learning Rule*
- *Perceptron Learning Rule*

- Desenvolvida para aprendizagem supervisada;
- Os pesos iniciais são atribuídos aleatoriamente;
- Os *inputs* são processados pela rede e comparados com o *output* desejado;
- Calcula-se o erro produzido pela rede na

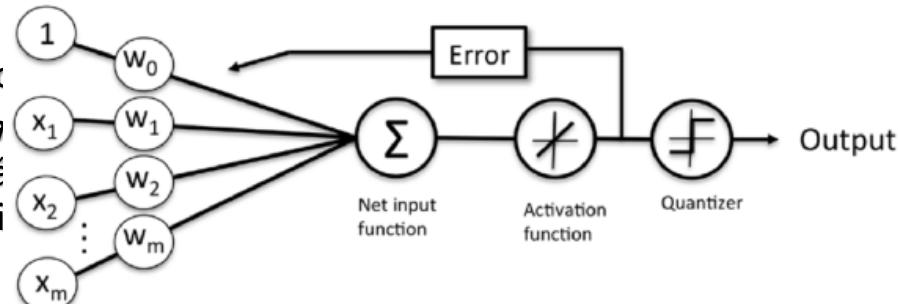
$$\sum_i \sum_j (E_{ij} - O_{ij})^2$$



- A função de alteração dos pesos usa este erro para calcular a atualização dos seus valores;

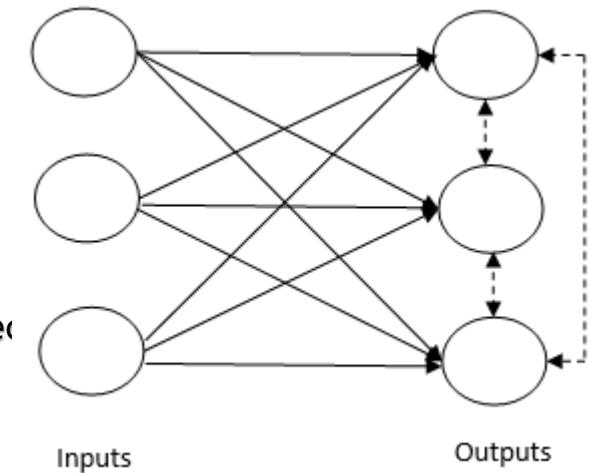
Regras de aprendizagem

- O treino de uma RNA corresponde à aplicação de regras de aprendizagem, por forma a fazer variar os pesos das ligações (sinapses);
- Regras de aprendizagem mais comuns:
 - *Hebbian Learning Rule*
 - *Perceptron Learning Rule*
 - *Widrow-Hoff Learning Rule*
 - Desenvolvida por Bernard Widrow e Marc
 - A principal diferença para *Perceptron Lea* de que é usado um sinal linear e não biná para cálculo do erro e consequente atuali dos pesos;



Regras de aprendizagem

- O treino de uma RNA corresponde à aplicação de regras de aprendizagem, por forma a fazer variar os pesos das ligações (sinapses);
- Regras de aprendizagem mais comuns:
 - *Hebbian Learning Rule*
 - *Perceptron Learning Rule*
 - *Widrow-Hoff Learning Rule*
 - *Competitive Learning Rule*
 - Desenvolvida para aprendizagem não supervisionada;
 - Os neurónios de *output* competem entre si para representarem o padrão do *input*,
 - O neurónio com maior *output* para um dado *input* é vencedor, sendo o único a alterar os pesos;



Regras de aprendizagem

- O treino de uma RNA corresponde à aplicação de regras de aprendizagem, por forma a fazer variar os pesos das ligações (sinapses);
- Regras de aprendizagem mais comuns:
 - *Hebbian Learning Rule*
 - *Perceptron Learning Rule*
 - *Widrow-Hoff Learning Rule (Delta Rule)*
 - *Competitive Learning Rule (Winner-takes-it-all Rule)*
 - *etc*



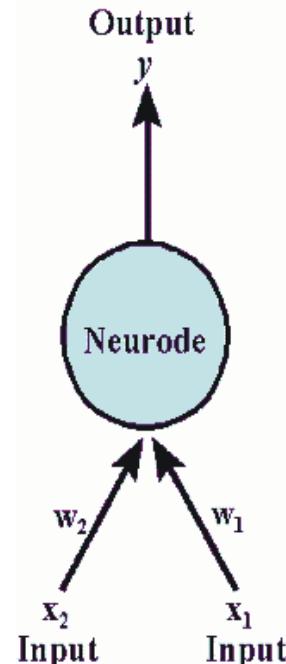
Afinação de parâmetros

- Quantidade de neurónios:
 - na camada de entrada;
 - na camada de saída;
 - nas camadas intermédias;
- Níveis (ou camadas) da RNA;
- Ligações entre neurónios;
- Topologia das ligações;
- Esquema de atribuição e atualização dos pesos;
- Funções:
 - de transferência;
 - de ativação;
 - de aprendizagem;
- Métodos de treino.



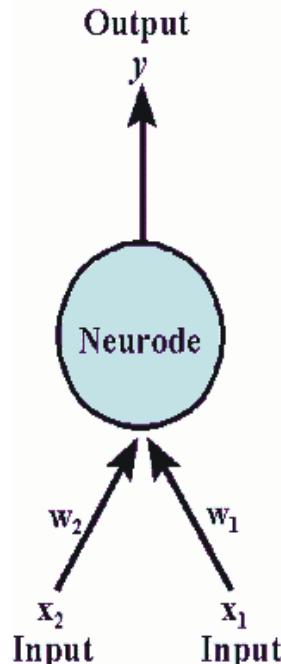
O “cérebro” artificial

- Perceptron (por volta de 1960)



<https://web.csulb.edu/~cwallis/artificialn/History.htm>

- Perceptron (por volta de 1960)

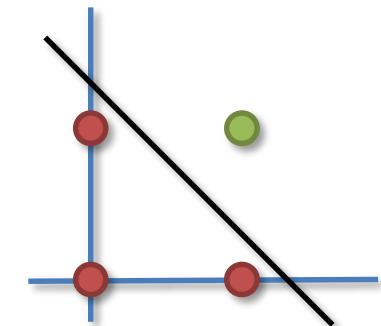


<https://web.csulb.edu/~cwallis/artificialn/History.htm>

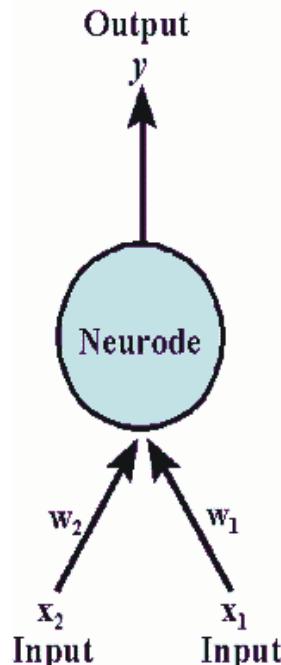
O “cérebro” artificial

- Função linear

p	q	and
1	1	1
1	0	0
0	1	0
0	0	0



- Perceptron (por volta de 1960)

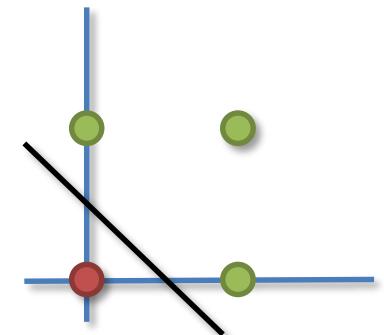


<https://web.csulb.edu/~cwallis/artificialn/History.htm>

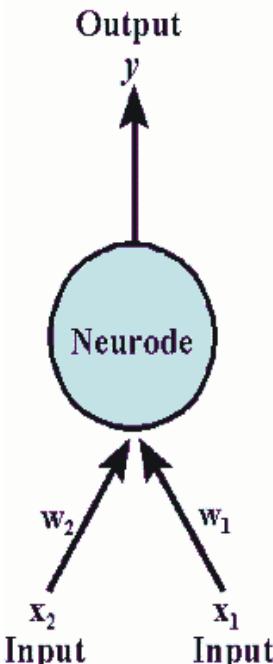
O “cérebro” artificial

- Função linear

p	q	and
1	1	1
1	0	0
0	1	0
0	0	0



- Perceptron (por volta de 1960)

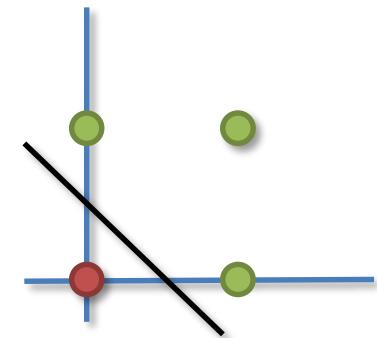


<https://web.csulb.edu/~cwallis/artificialn/History.htm>

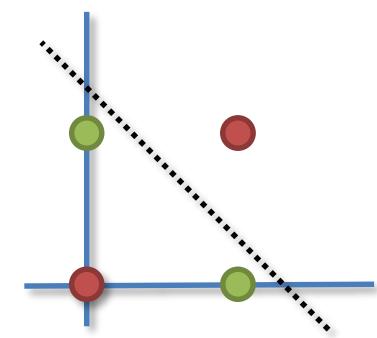
O “cérebro” artificial

- Função linear

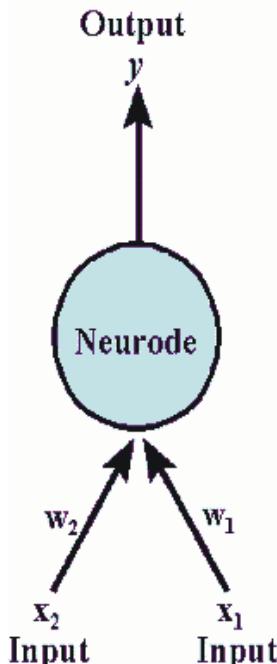
p	q	and
1	1	1
1	0	0
0	1	0
0	0	0



p	q	xor
1	1	0
1	0	1
0	1	1
0	0	0



- Perceptron (por volta de 1960)

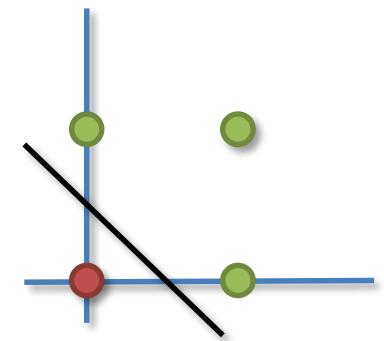


<https://web.csulb.edu/~cwallis/artificialn/History.htm>

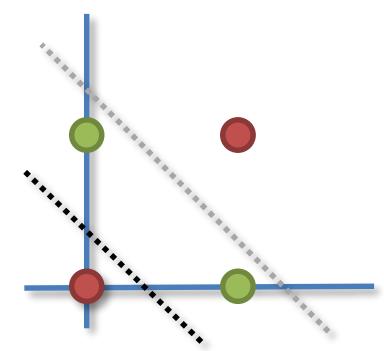
O “cérebro” artificial

- Função linear

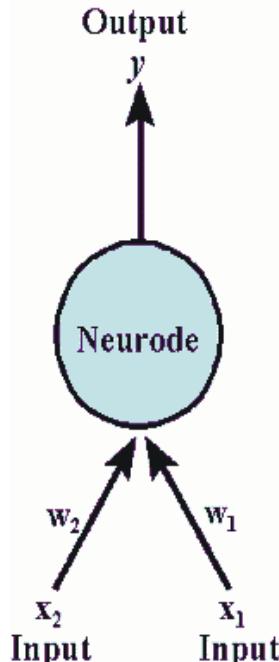
p	q	and
1	1	1
1	0	0
0	1	0
0	0	0



p	q	xor
1	1	0
1	0	1
0	1	1
0	0	0



- Perceptron (por volta de 1960)

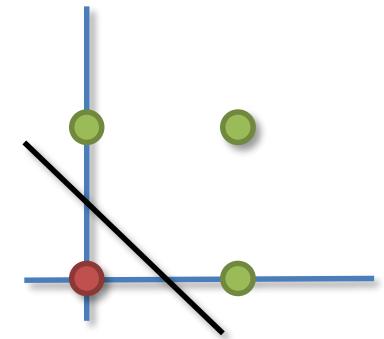


<https://web.csulb.edu/~cwallis/artificialn/History.htm>

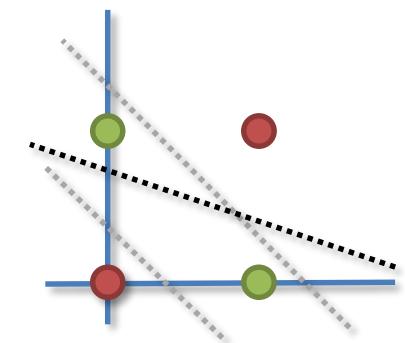
O “cérebro” artificial

- Função linear

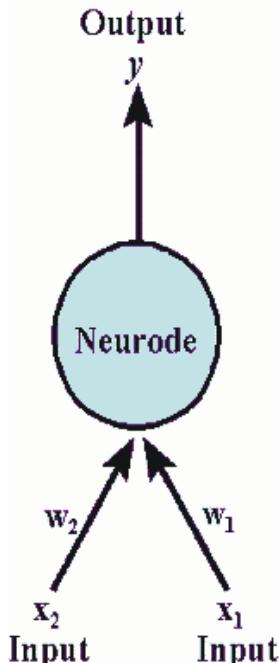
p	q	and
1	1	1
1	0	0
0	1	0
0	0	0



p	q	xor
1	1	0
1	0	1
0	1	1
0	0	0



- Perceptron (por volta de 1960)

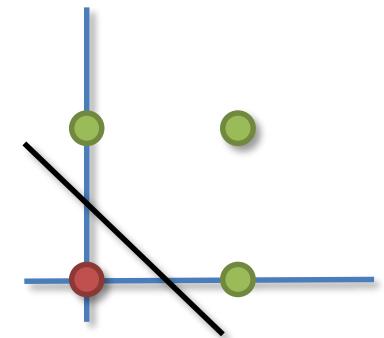


<https://web.csulb.edu/~cwallis/artificialn/History.htm>

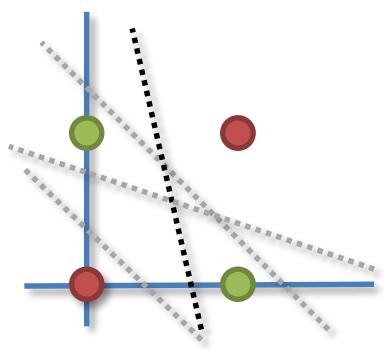
O “cérebro” artificial

- Função linear

p	q	and
1	1	1
1	0	0
0	1	0
0	0	0



p	q	xor
1	1	0
1	0	1
0	1	1
0	0	0





Resolução de problemas

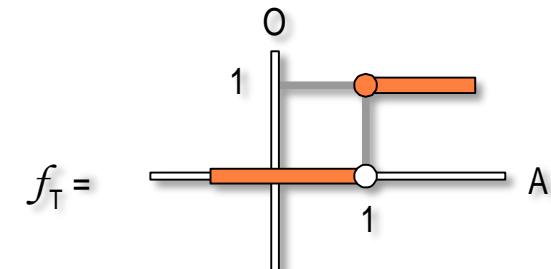


Problema: XOR

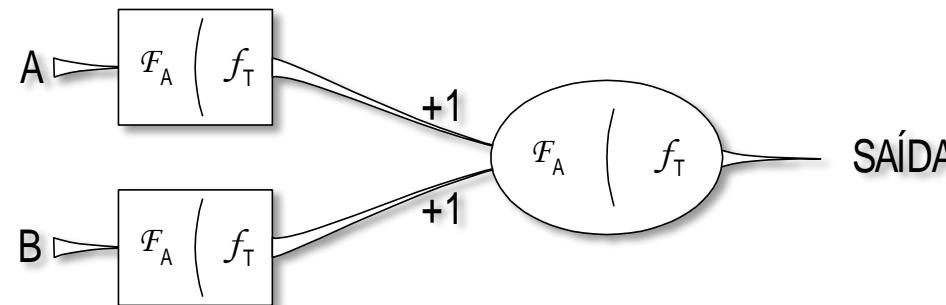
p	q	xor
1	1	0
1	0	1
0	1	1
0	0	0

- Função de ativação:
 $F_A = \sum$ entradas x pesos

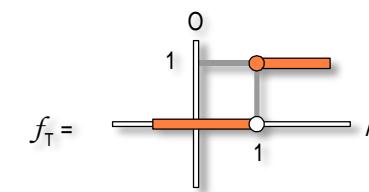
- Função de transferência:



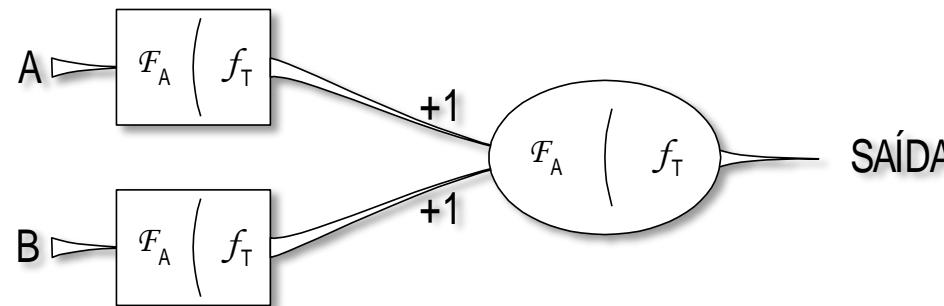
Problema: XOR Perceptron



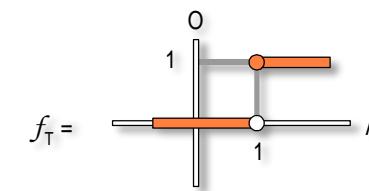
$$\mathcal{F}_A = \sum \text{entradas} \times \text{pesos}$$



Problema: XOR Perceptron

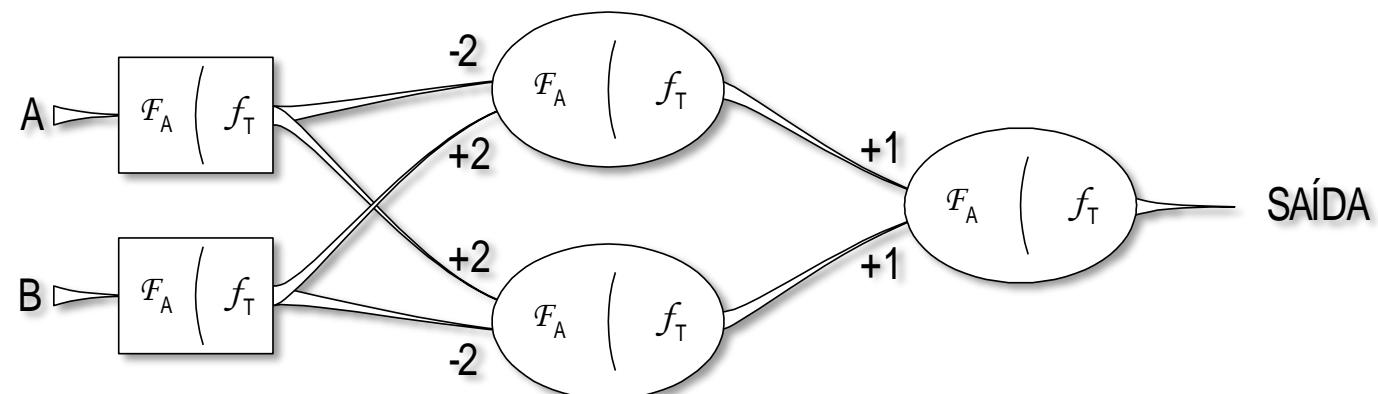


$$F_A = \sum \text{entradas} \times \text{pesos}$$

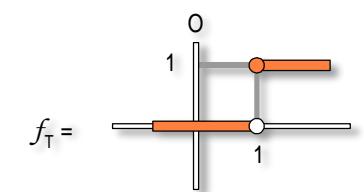


p	q	xor
1	1	1
1	0	1
0	1	1
0	0	0

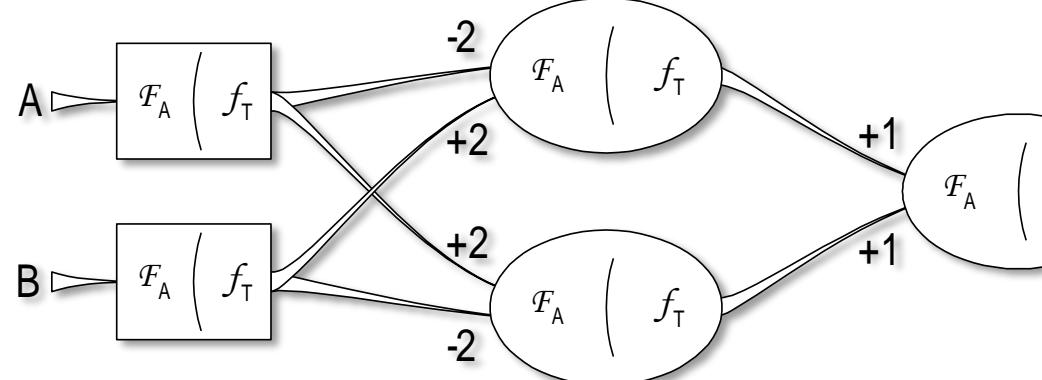
Problema: XOR Multi-layer Perceptron



$$\mathcal{F}_A = \sum \text{entradas} \times \text{pesos}$$

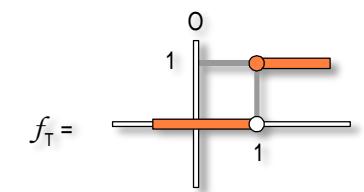


Problema: XOR Multi-layer Perceptron



p	q	xor
1	1	0
1	0	1
0	1	1
0	0	0

$$F_A = \sum \text{entradas} \times \text{pesos}$$





- Considere-se uma Rede Neuronal Artificial...

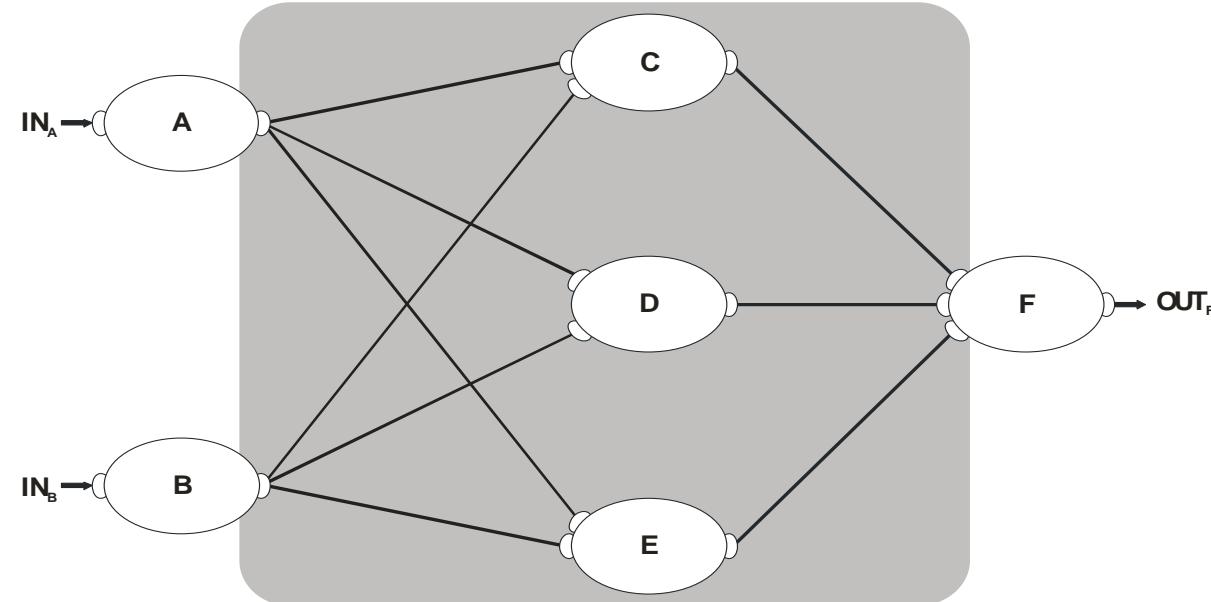


RNA

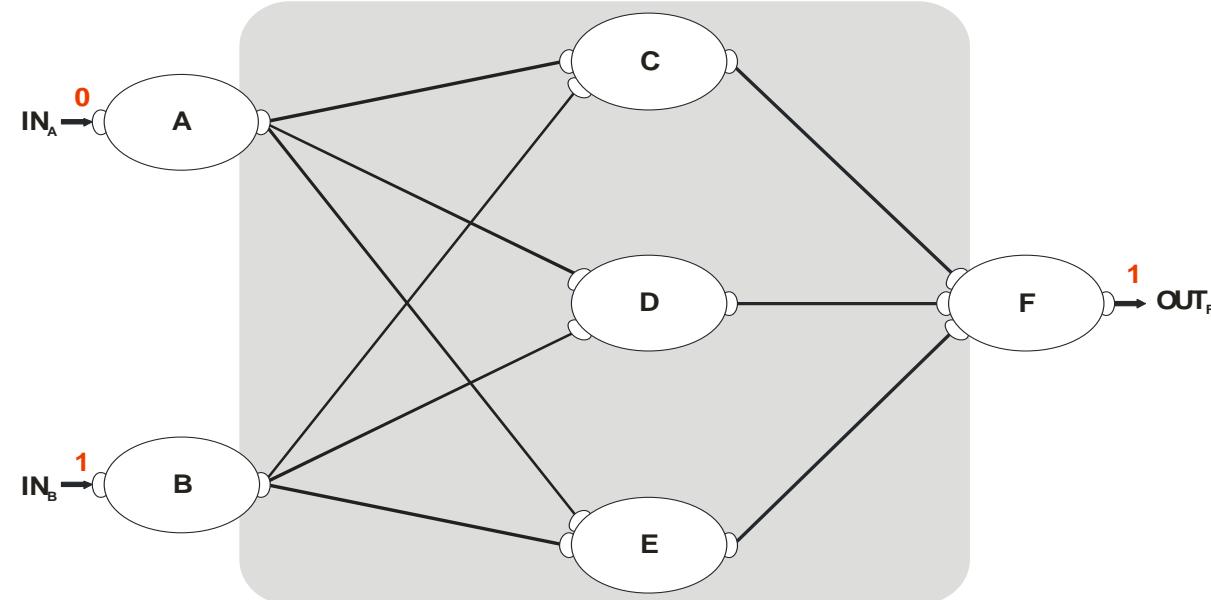
- ... composta por 2 neurónios à entrada e 1 à saída...



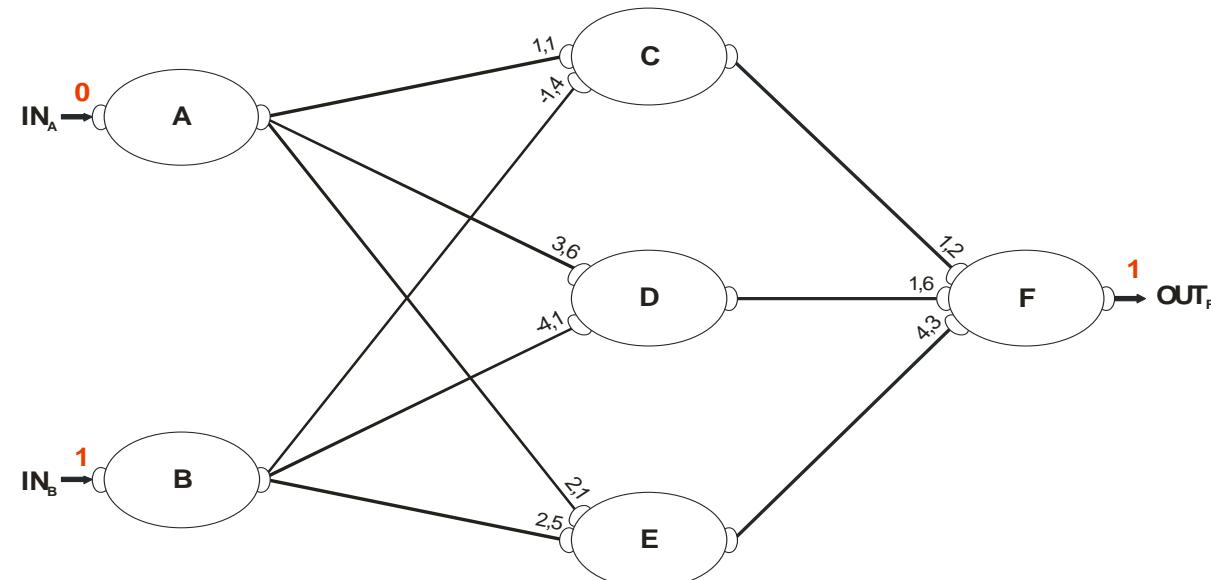
- ... *feed forward*, completamente ligada.



- Os exemplos de treino contêm os resultados pretendidos.



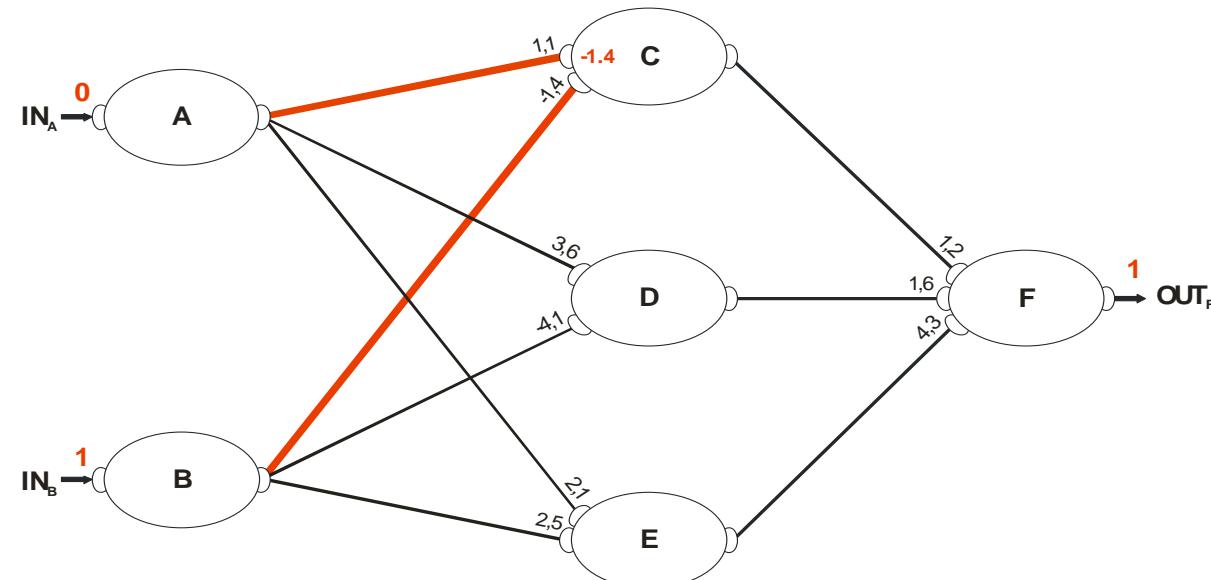
- Atribuição aleatória dos pesos às sinapses.



$$f_A(P, E) = \sum P \times E$$

$$f_T(A) = A$$

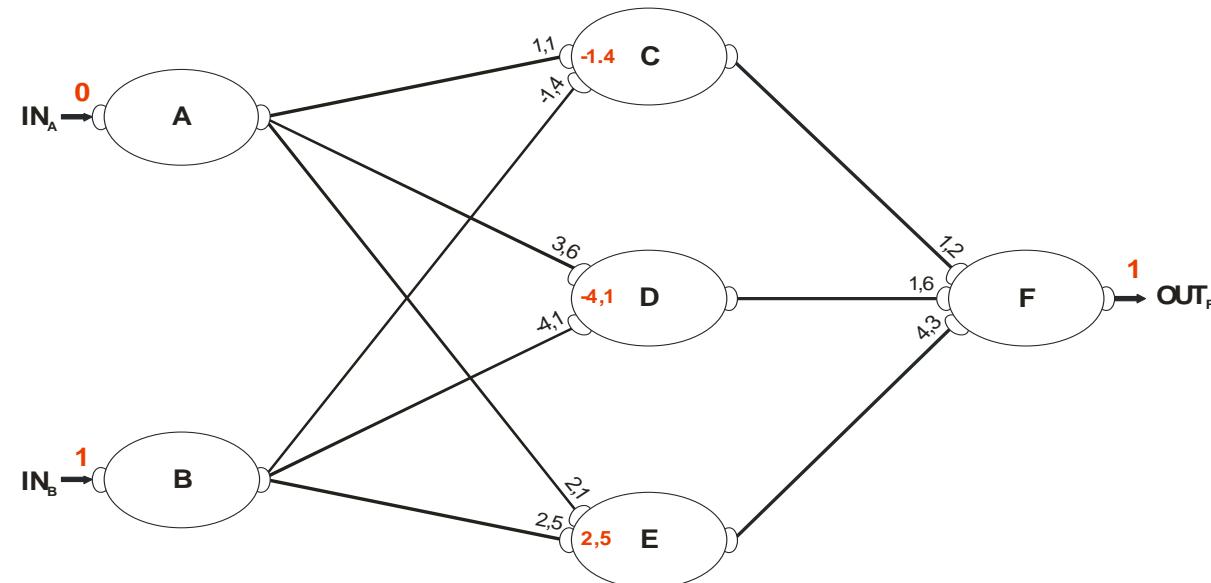
- Cálculo do valor de ativação...



$$f_A(P, E) = \sum P \times E$$

$$f_T(A) = A$$

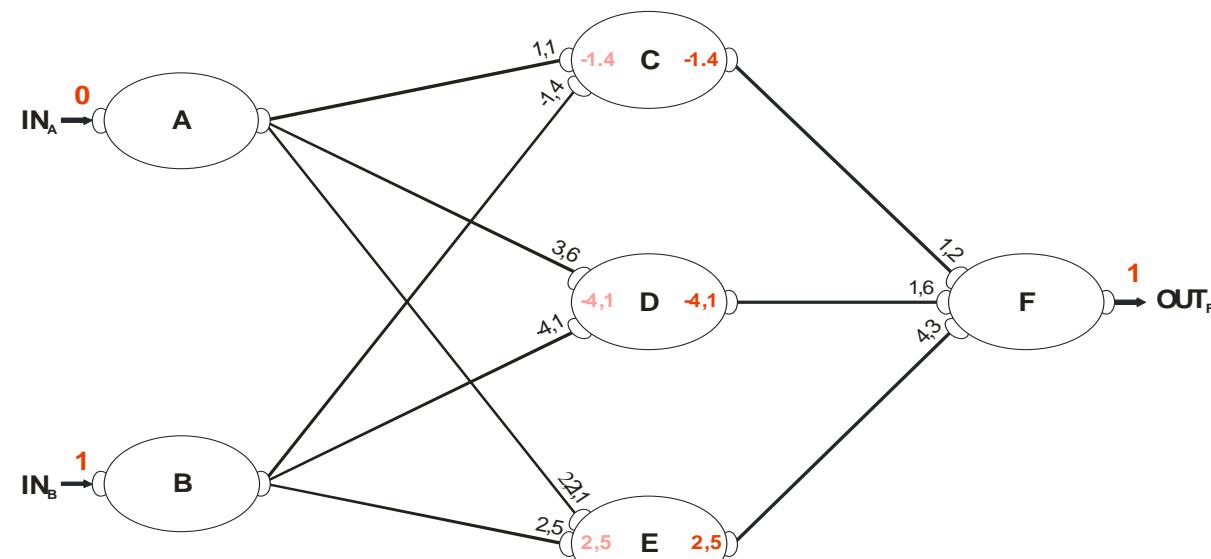
- ... para todos os neurónios da camada intermédia.



$$f_A(P, E) = \sum P \times E$$

$$f_T(A) = A$$

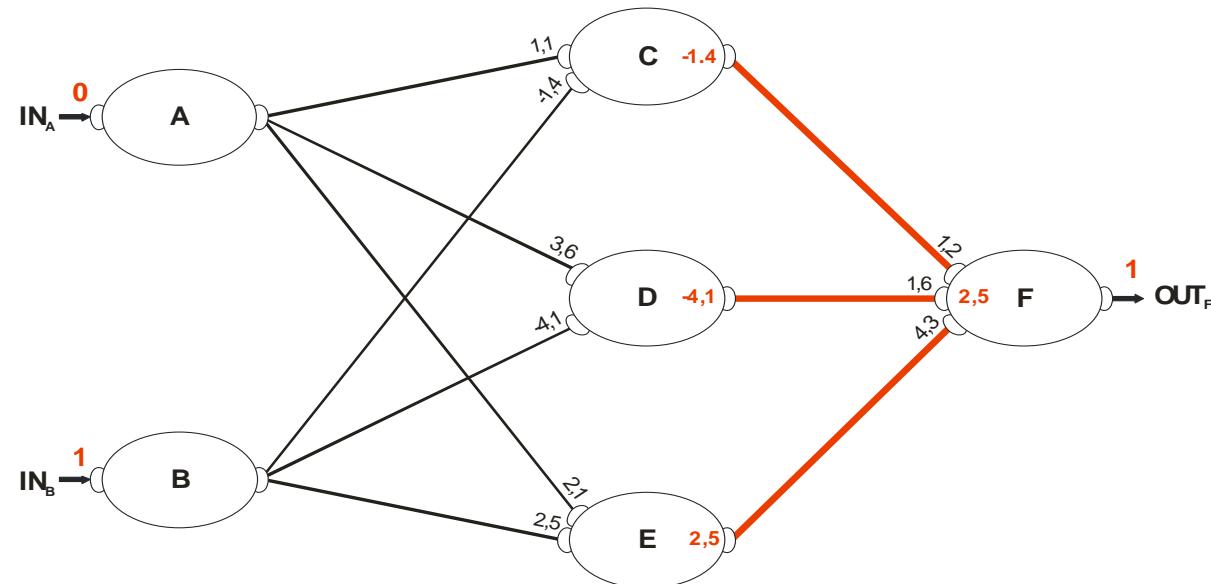
- Cálculo do valor de transferência.



$$f_A(P, E) = \sum P \times E$$

$$f_t(A) = A$$

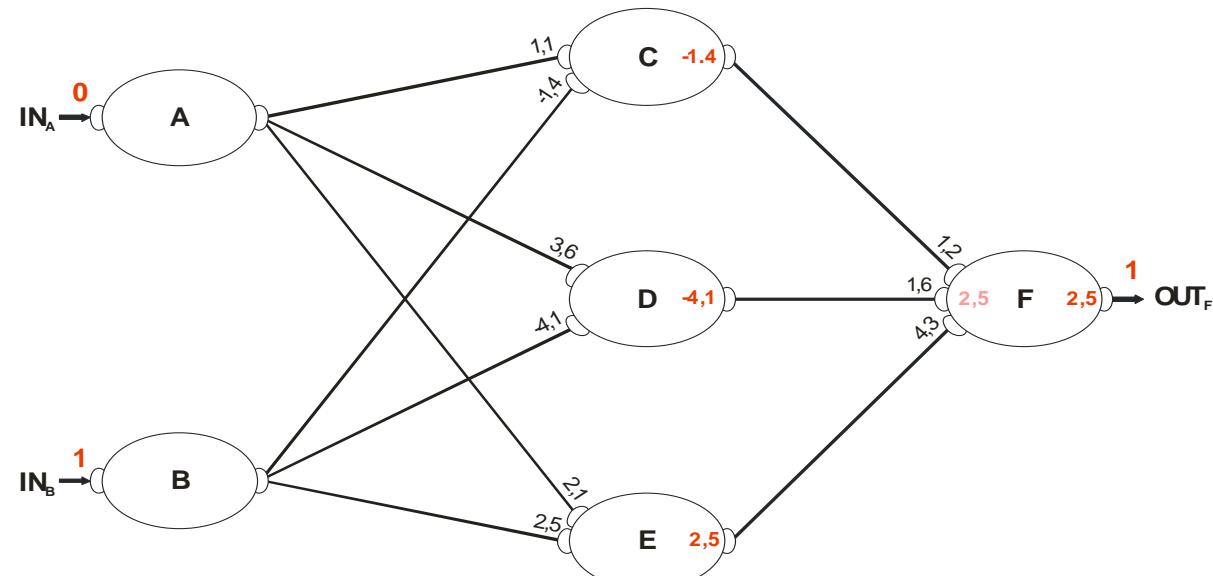
- Valor de ativação na camada de saída...



$$f_A(P, E) = \sum P \times E$$

$$f_T(A) = A$$

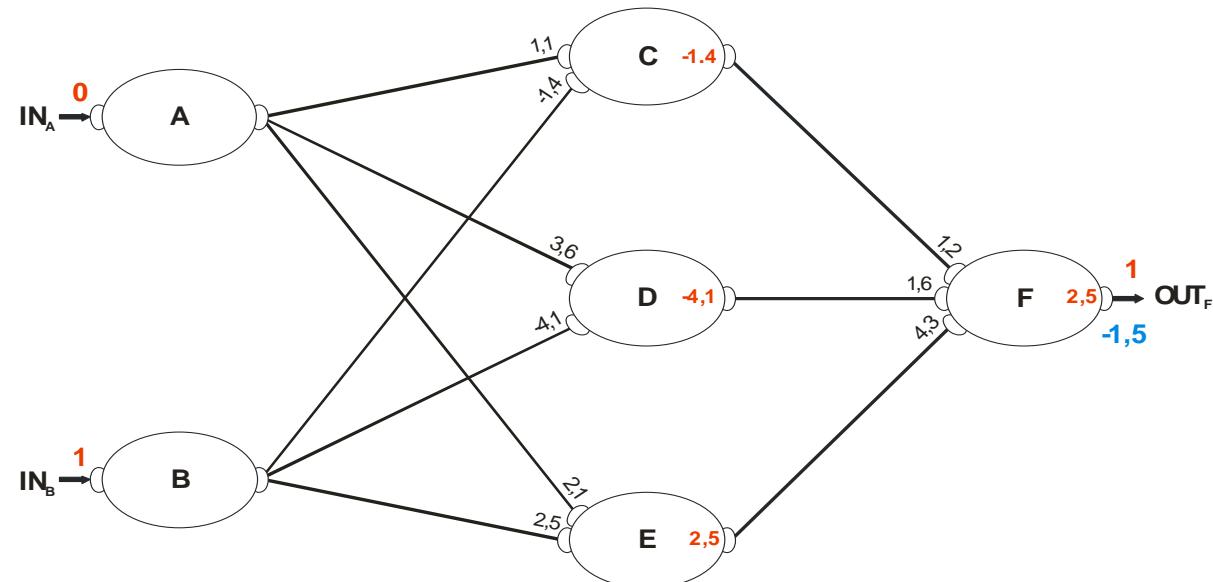
- ... e respetivo valor de transferência.



$$f_A(P, E) = \sum P \times E$$

$$f_T(A) = A$$

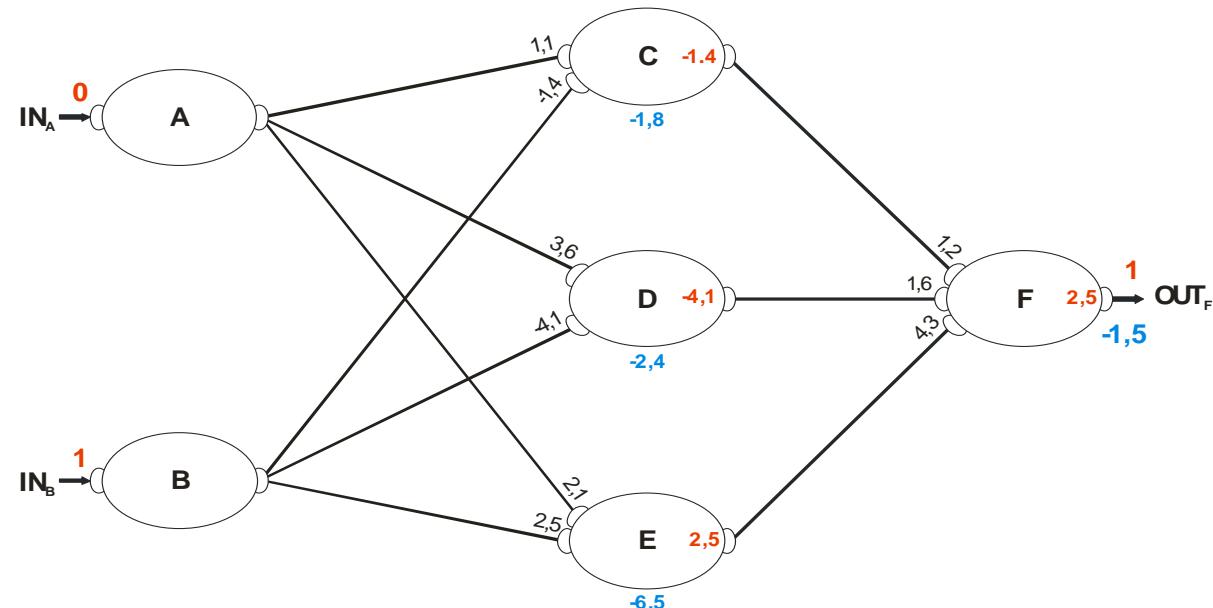
- Cálculo do erro na camada de saída...



$$\epsilon = OUT_D - OUT_C$$

$$\epsilon_{\leftarrow} = \epsilon \times P$$

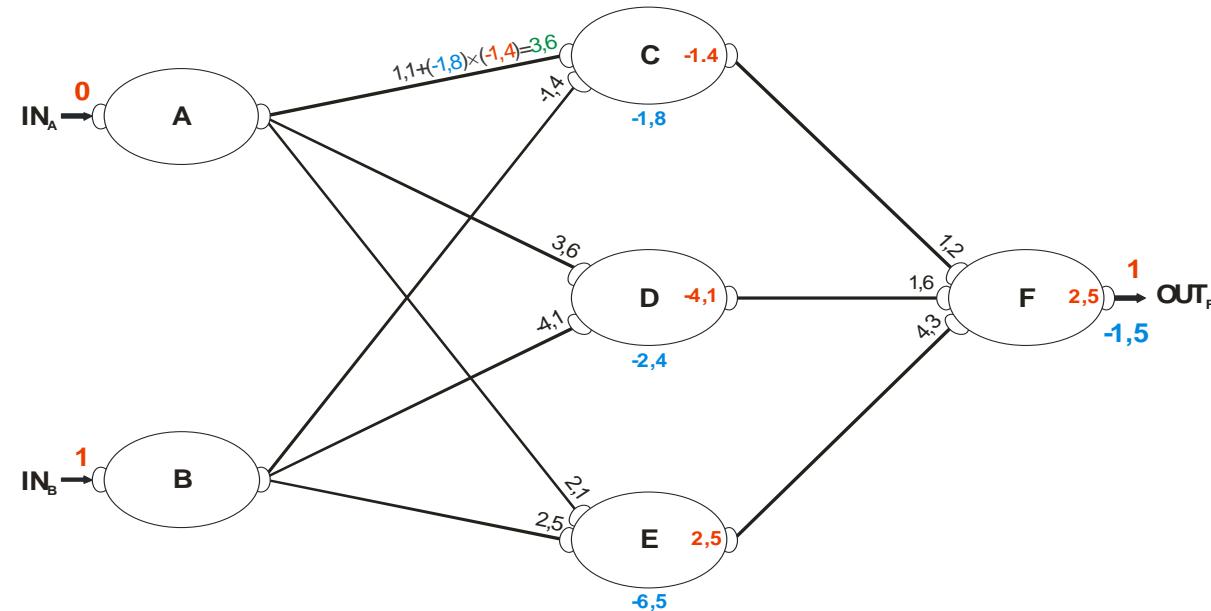
- ... e cálculo do valor estimado do erro na camada intermédia.



$$\varepsilon = \text{OUT}_D - \text{OUT}_C$$

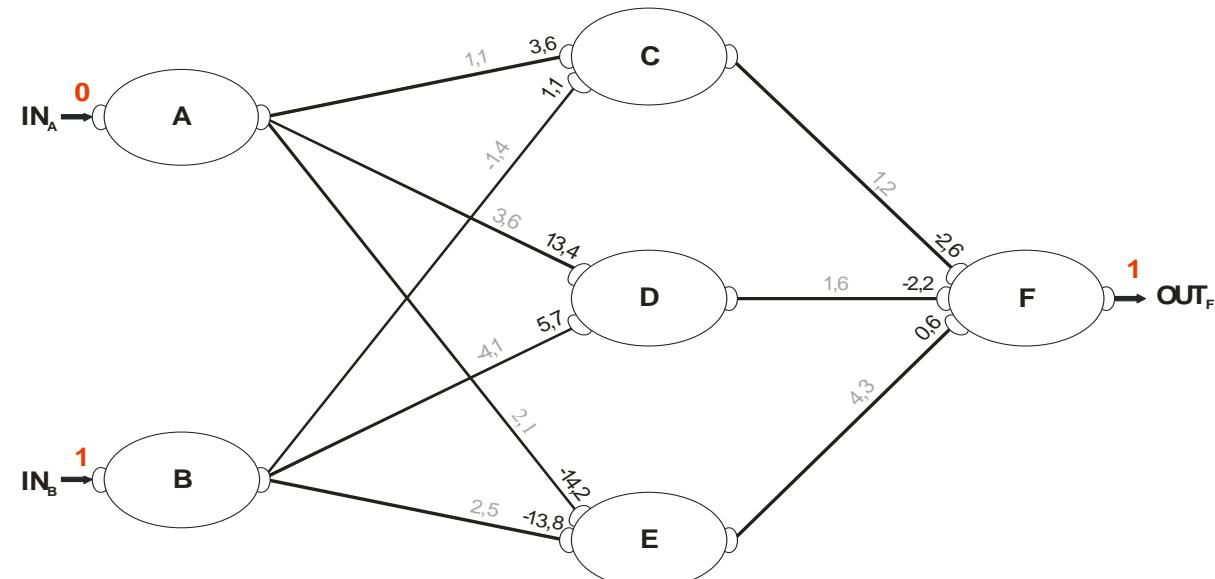
$$\varepsilon_{\leftarrow} = \varepsilon \times P$$

- Aplicação de uma regra de atualização dos pesos das sinapses...



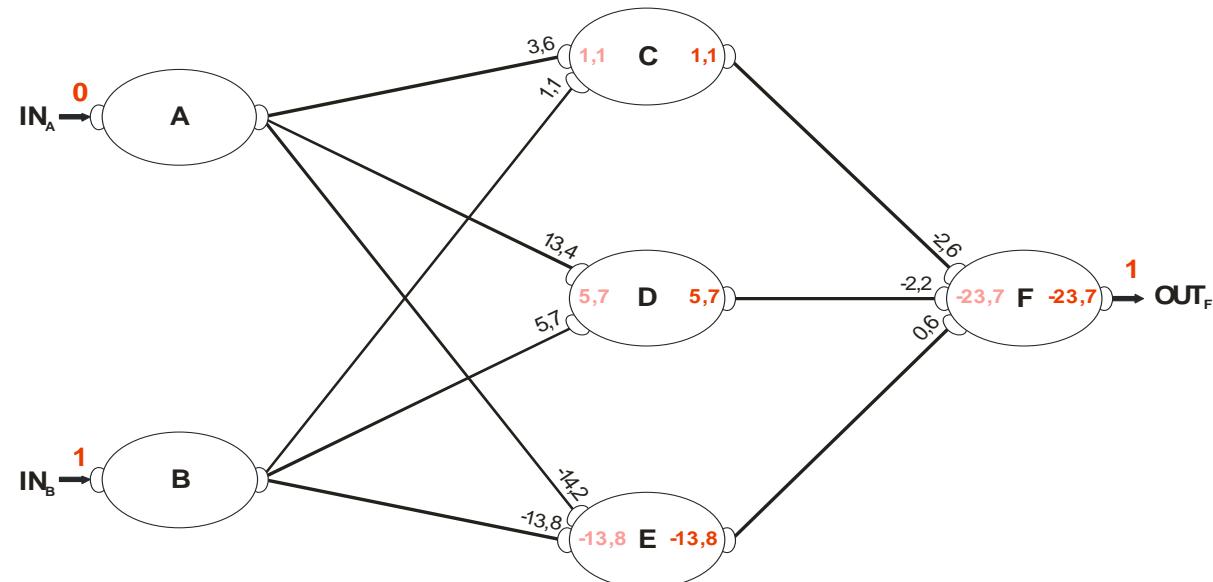
$$P_{i+1} = P_i + \mathcal{E} \times f_T$$

- ... para atualizar os valores das sinapses de todos os neurónios.



$$P_{i+1} = P_i + \mathcal{E} \times f_t$$

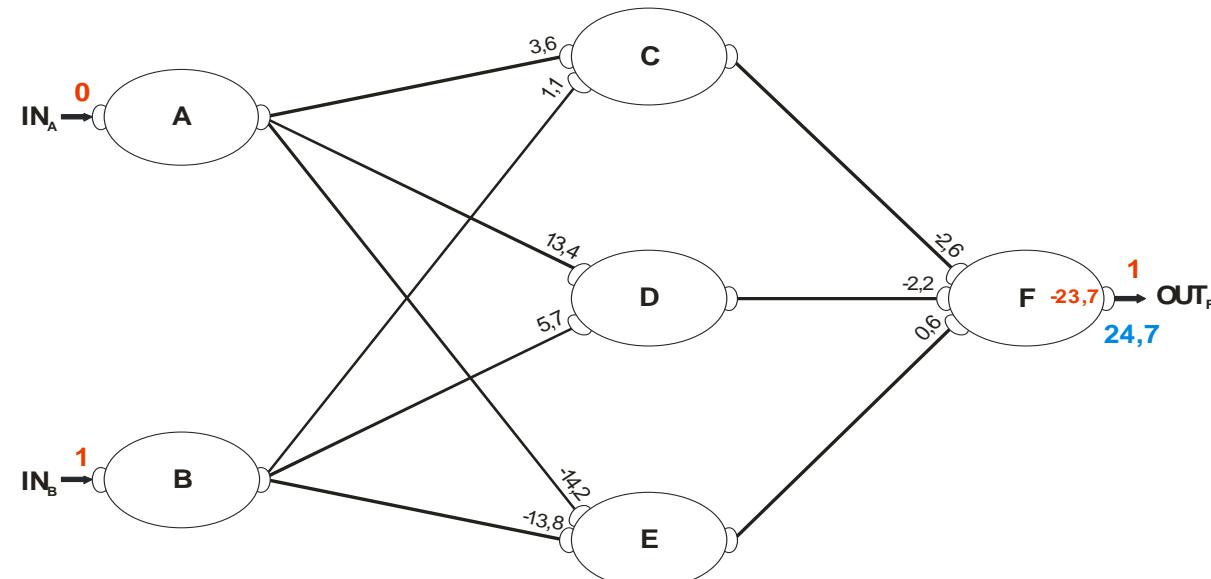
- Segunda iteração da propagação do caso de treino...



$$f_A(P, E) = \sum P \times E$$

$$f_T(A) = A$$

- ... e cálculo do erro produzido pela RNA na segunda iteração.



Referências bibliográficas

- Cortez, P., Neves, J., "Redes Neuronais Artificiais", Unidade de Ensino, Departamento de Informática, Universidade do Minho, 2000;
- Haykin, S., "Neural Networks - A Comprehensive Foundation", Prentice-Hall, New Jersey, 2nd Edition, 1999.

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Sistemas de Representação de Conhecimento e Raciocínio

MiEI/3º – 2º Semestre