

**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

**ADI**

# **Redes Neurais Artificiais**

Licenciatura em Engenharia Informática, 3º ano

Mestrado integrado em Engenharia Informática, 4º ano



**ISLab**

Synthetic Intelligence Lab

# Inteligência Artificial



(recolha de opiniões em aula)

## Inteligência Artificial

### Definição

- *“Every aspect of learning or any other feature of intelligence can in 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

<http://jmc.stanford.edu>



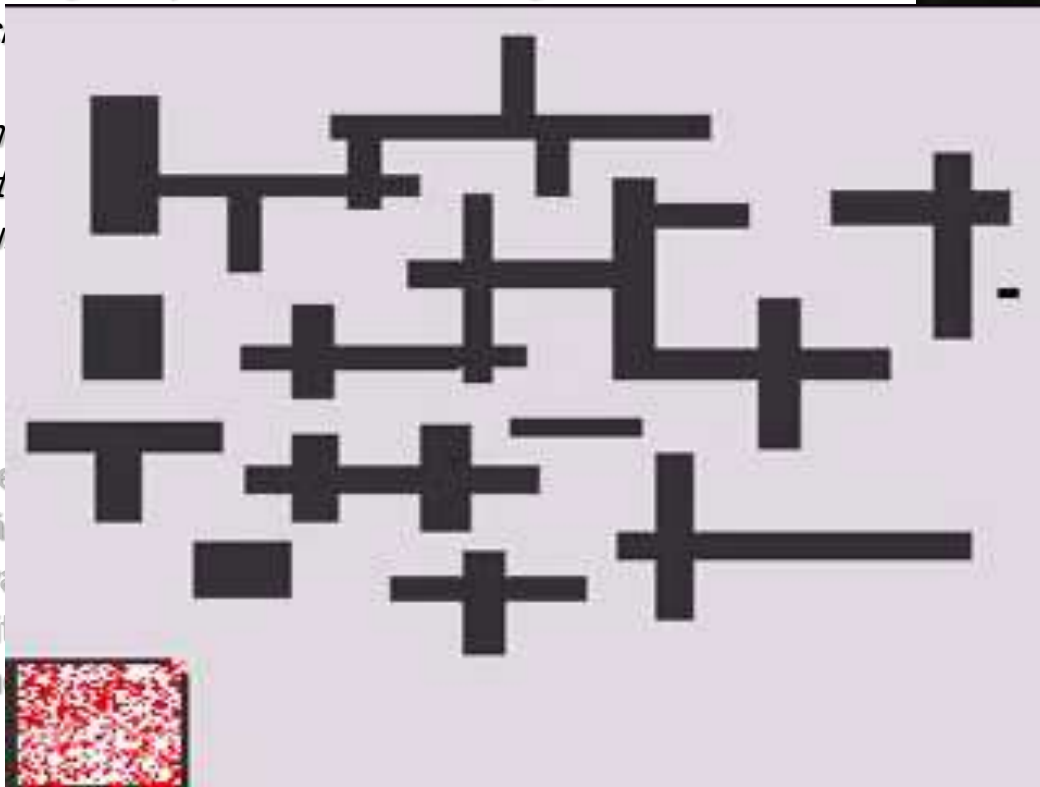
- Cada aspeto da aprendizagem ou qualquer outra característica da inteligência pode, em princípio, ser descrito de forma tão precisa que será possível construir uma máquina para o simular. Serão feitas tentativas para descobrir como fazer com que as máquinas usem a linguagem, formem abstrações e conceitos,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

## Inteligência Artificial

### Definição

- *“Every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to express in a language, form abstract enough to be amenable to analysis, the tentative theory of intelligence now reserved for humans.”*

- Cada aspecto da aprendizagem pode ser descrito de forma tão precisa que uma máquina pode ser feita para simulá-lo. Serão feitas tentativas de expressar em uma linguagem, forma abstrata o suficiente para ser amigável à análise, a tentativa de teoria da inteligência agora reservada para humanos.



o princípio, ser  
gem, formem  
anos, e sejam

- Gerar conhecimento a partir de dados
- Gerar conhecimento para tomar decisões

## ***Machine Learning***

Aprendizagem  
de decisão





## ***Machine Learning*** **Definição**

- *“Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.”*

*Machine learning focuses on developing computer programs that can access data and use it to learn for themselves.”*

Marco Varone et al.

[www.expertsystem.com/machine-learning-definition](http://www.expertsystem.com/machine-learning-definition)



- Aprendizagem computacional é uma aplicação de inteligência artificial que permite que, de forma automática, os sistemas aprendam e melhorem a partir da experiência, sem necessidade de serem explicitamente programados.

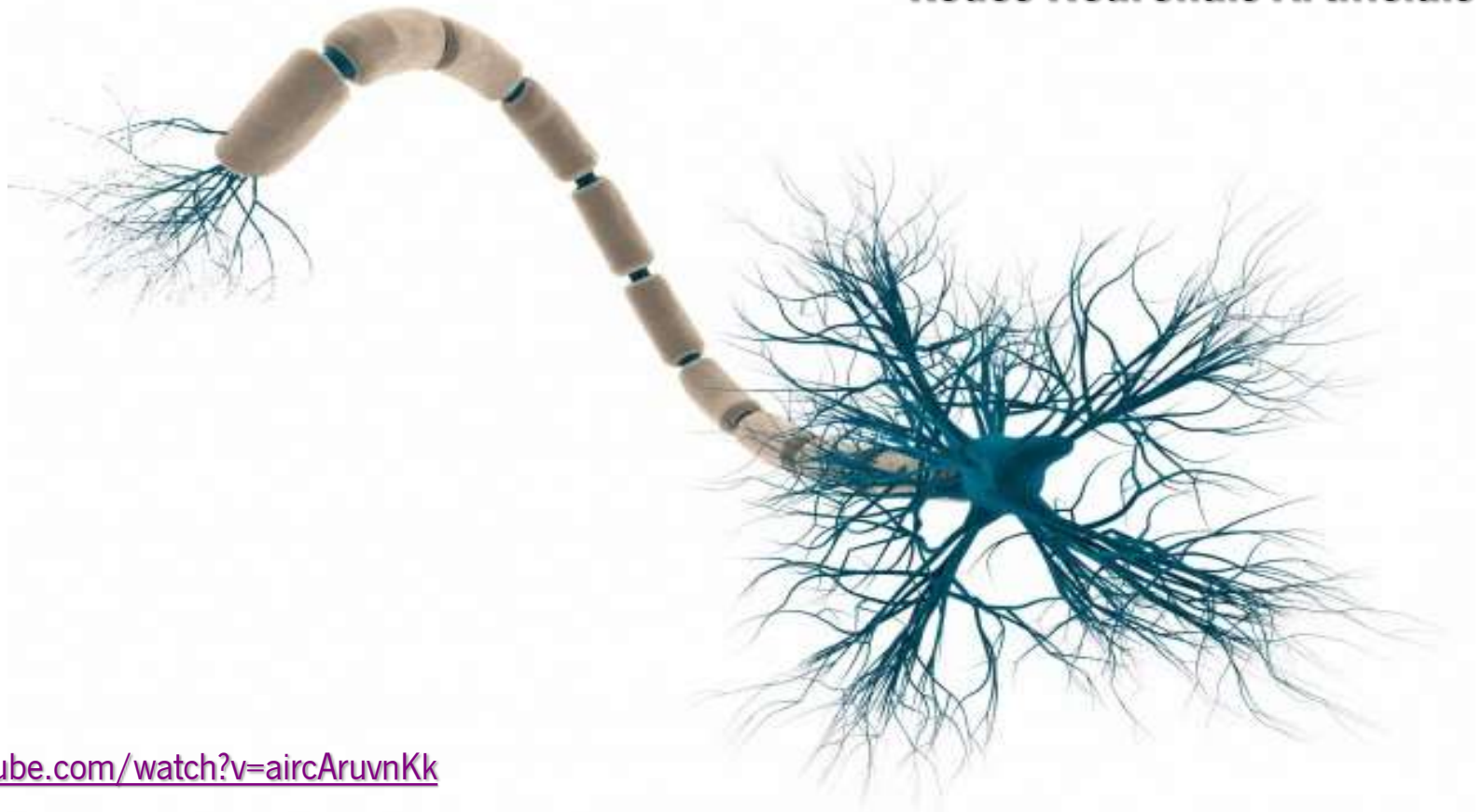
Esta aprendizagem foca-se no desenvolvimento de programas que usam dados para aprenderem por si próprios.



# ISLab

Synthetic Intelligence Lab

## Redes Neurais Artificiais



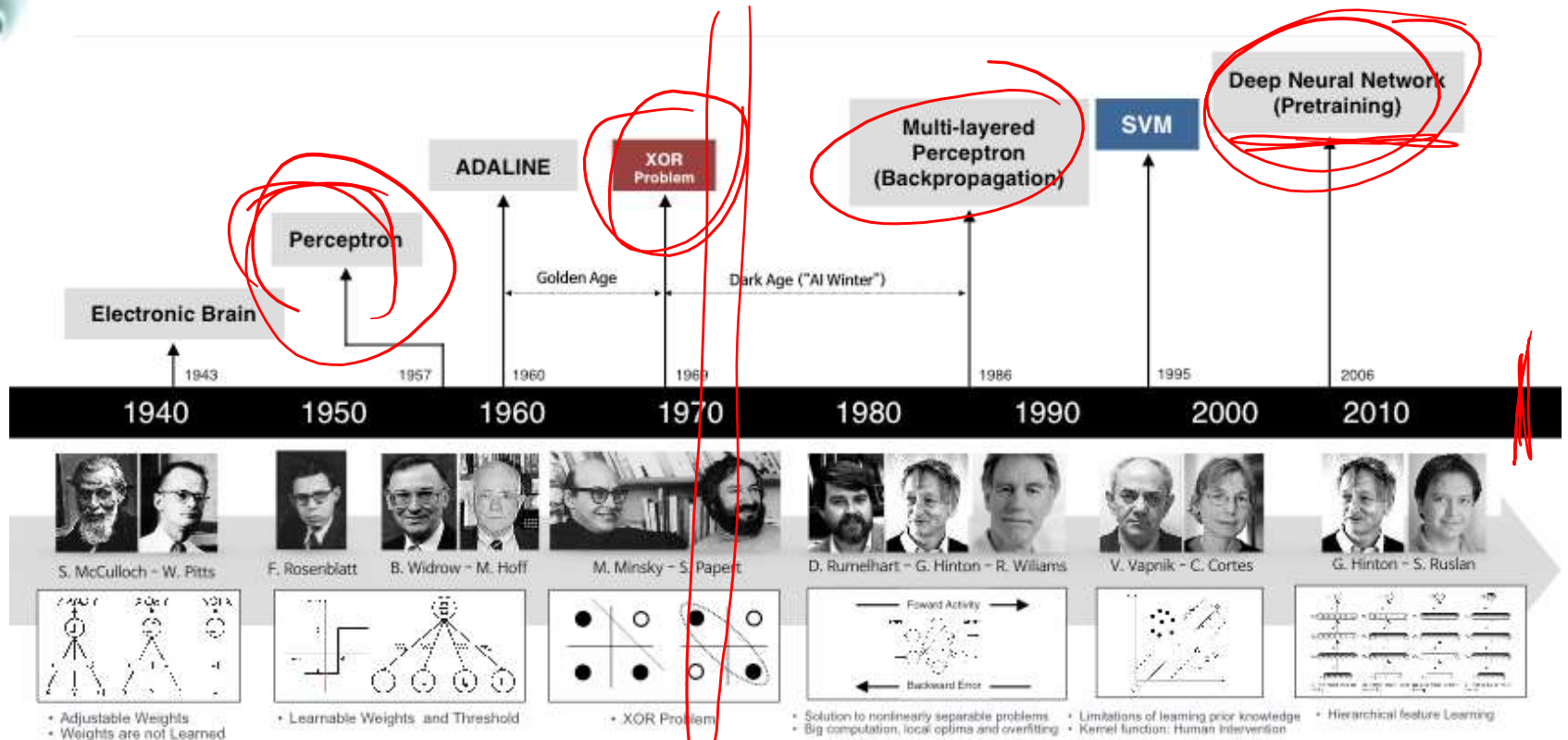
[www.youtube.com/watch?v=aircAruvnKk](https://www.youtube.com/watch?v=aircAruvnKk)



# ISLab

Synthetic Intelligence Lab

## Evolução Redes Neuronais Artificiais

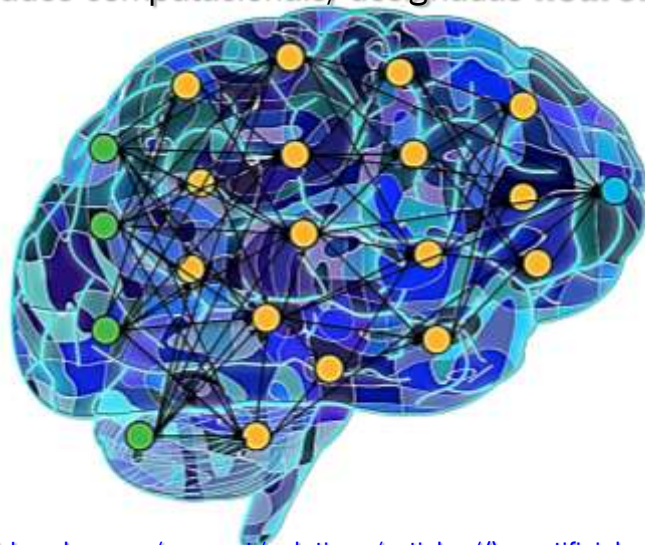




## Definição

### Redes Neurais Artificiais

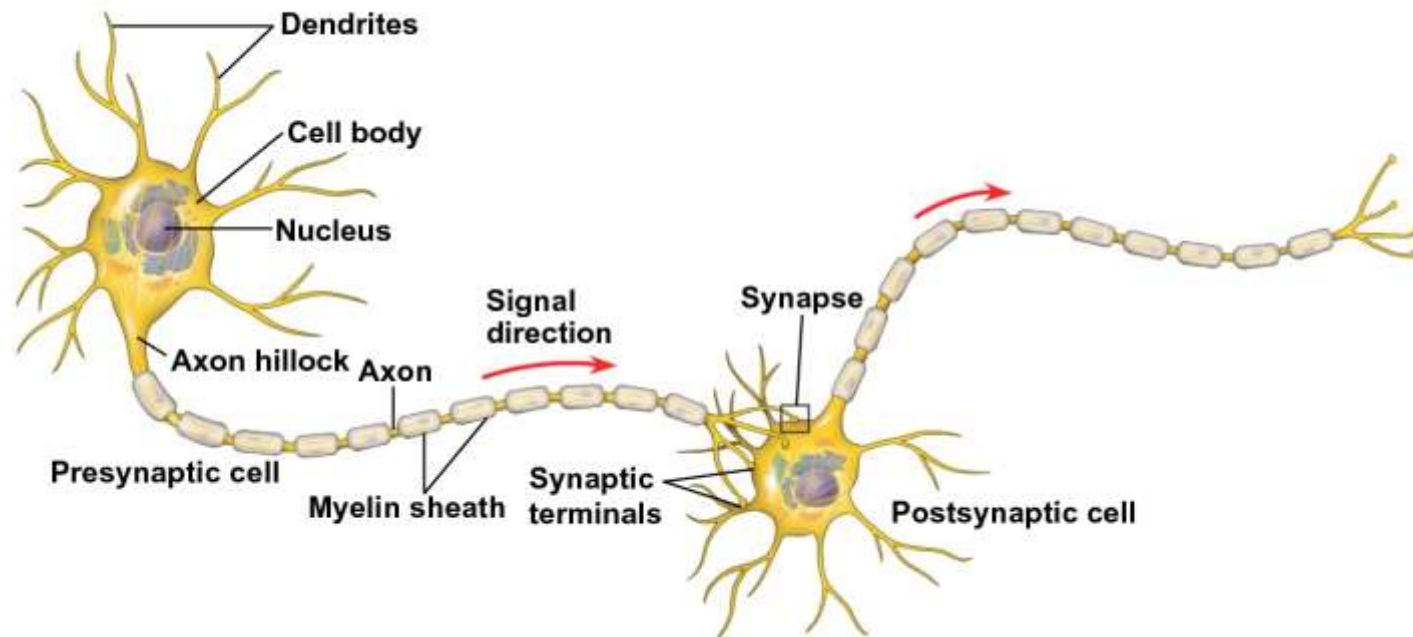
- Uma **Rede Neuronal Artificial** (RNA) é um sistema computacional de base conexionista para a resolução de problemas.
- Uma RNA é concebida com base num **modelo** simplificado **do sistema nervoso central** dos seres humanos.
- Uma RNA é definida por uma estrutura interligada de unidades computacionais,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



## Definição

### Redes Neurais Artificiais

- Uma **Rede Neuronal Artificial** (RNA) é um sistema computacional de base conexionista para a resolução de problema
- Uma RNA é humanos.
- Uma RNA é com capacic



## Definição

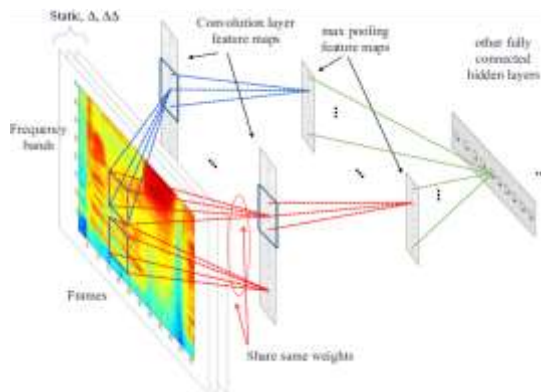
### Redes Neuronais Artificiais

- Uma **Rede Neuronal Artificial** (RNA) é concebida com base na estrutura de problemas.
- Uma RNA é concebida com base na estrutura de problemas humanos.
- Uma RNA é definida por uma estrutura com capacidade de **aprendizagem**.

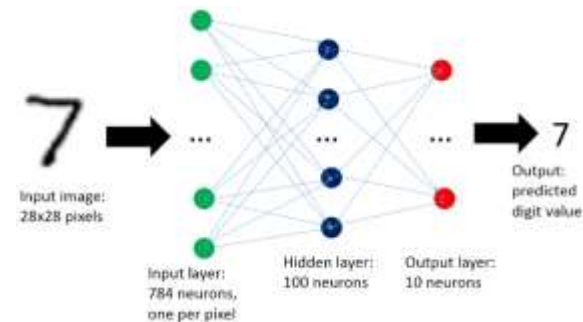


**Biological  
Simulation**

- Reconhecimento de caracteres
- Detecção de fraudes
- Reconhecimento de áudio/vídeo



## Por onde andam? Redes Neurais Artificiais








# ISLab

Synthetic Intelligence Lab



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

## Por onde andam? Redes Neurais Artificiais



**THE ULTIMATE GO CHALLENGE**  
GAME 5 OF 5  
15 MARCH 2016

AlphaGo vs Lee Sedol

AlphaGo Won 4 of 5  
Lee Sedol Won 1 of 5

RESULT	NUMBER OF MOVES	TIME WHITE	TIME BLACK
W+ Res	280	2h+	2h+

[www.deepmind.com/blog/alphago-zero-starting-from-scratch](http://www.deepmind.com/blog/alphago-zero-starting-from-scratch)



## Por onde andam? Redes Neuronais Artificiais

### ▪ AlphaGO (Google DeepMind)

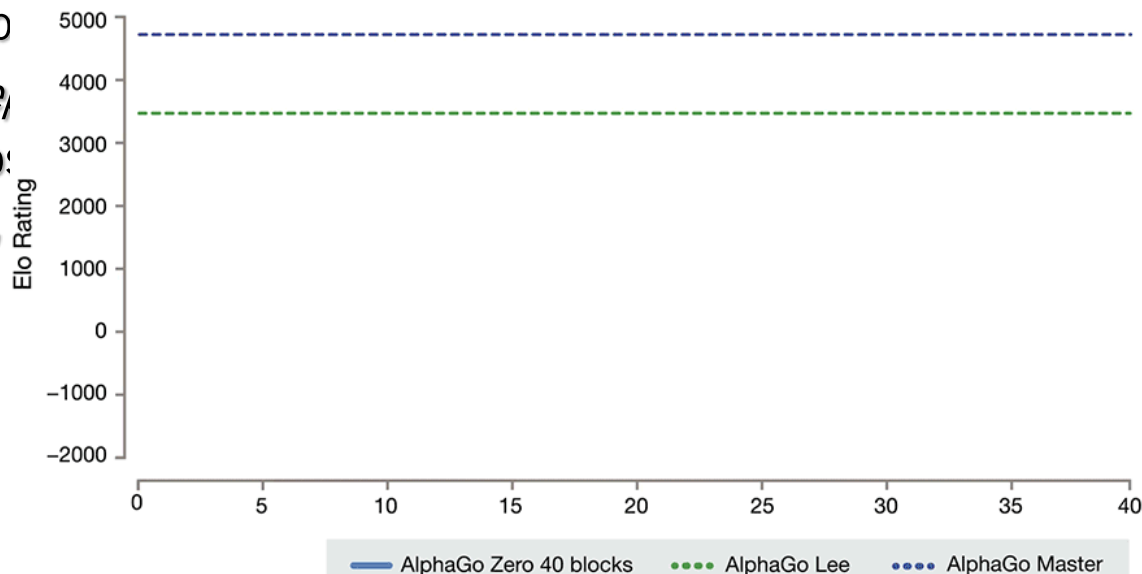
- A partida entre Lee Sedol (o campeão do mundo de GO) e a máquina não correu bem (para o ele!);
- GO é um jogo para 2 participantes, similar ao xadrez mas bastante mais complexo;
- AlphaGO combina *deep neural networks* (avaliação) e algoritmos de procura Monte Carlo (seleção), numa combinação de paradigmas de aprendizagem supervisionada e por reforço.



## Por onde andam? Redes Neuronais Artificiais

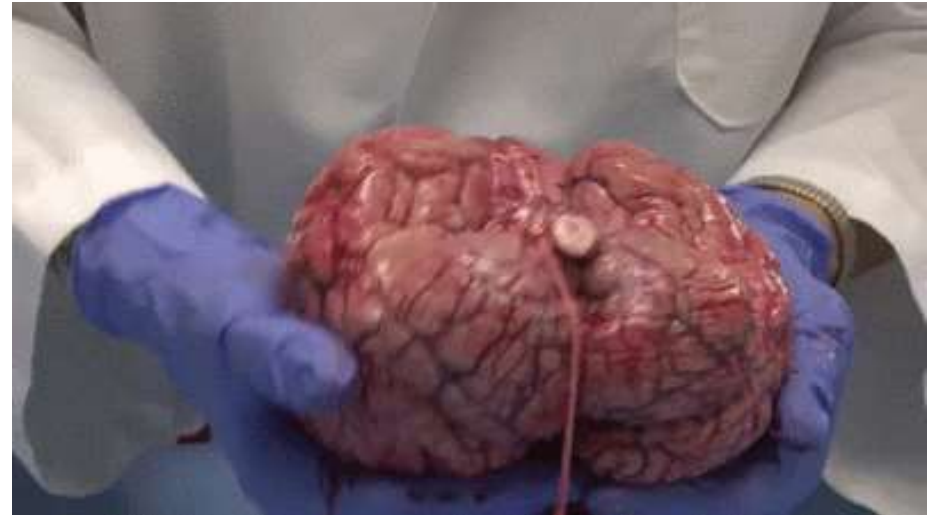
### ▪ AlphaGO (Google DeepMind)

- A partida entre Lee Sedol (o campeão do mundo de GO) e a máquina não correu bem (para o ele!);
- GO é um jogo para 2 p
- AlphaGO combina *dee* (avaliação) e algoritmo Monte Carlo (seleção), de paradigmas de aprendizagem supervisionada e por reforço.



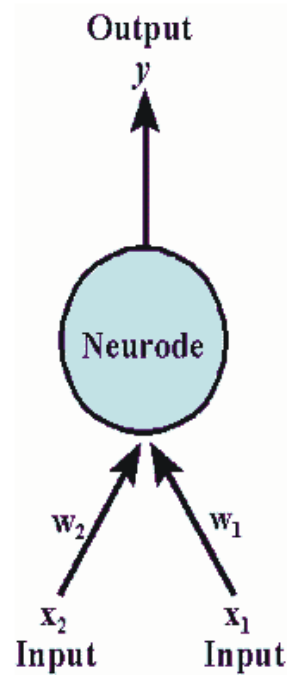
## O cérebro humano

- 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



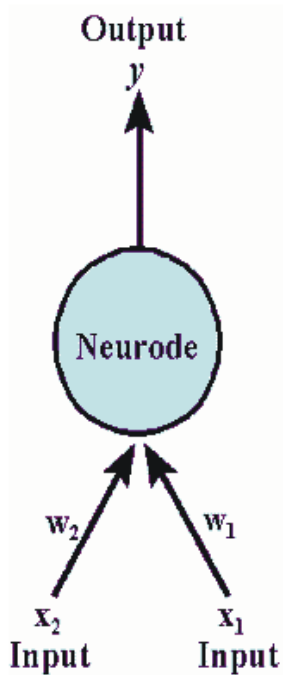
## O “cérebro” artificial

- Perceptron (por volta de 1960)



## O “cérebro” artificial

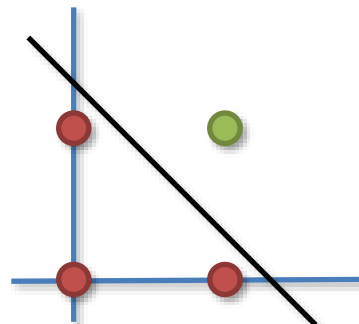
- Perceptron (por volta de 1960)



[web.csulb.edu/~cwallis/artificialn/History.htm](http://web.csulb.edu/~cwallis/artificialn/History.htm)

- Função linear

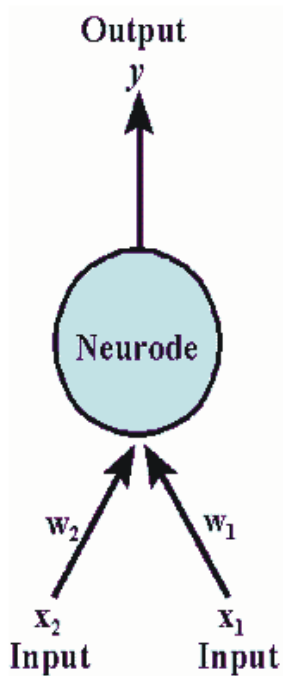
p	q	and
V	V	V
V	F	F
F	V	F
F	F	F





## O “cérebro” artificial

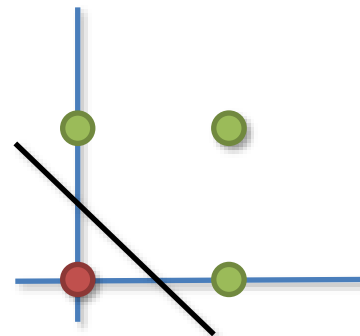
- Perceptron (por volta de 1960)



[web.csulb.edu/~cwallis/artificialn/History.htm](http://web.csulb.edu/~cwallis/artificialn/History.htm)

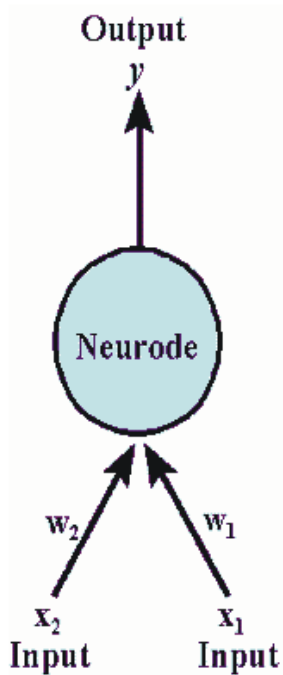
- Função linear

p	q	or
V	V	V
V	F	V
F	V	V
F	F	F



## O “cérebro” artificial

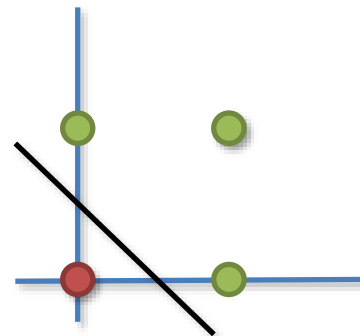
- Perceptron (por volta de 1960)



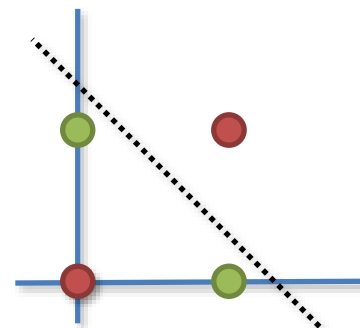
[web.csulb.edu/~cwallis/artificialn/History.htm](http://web.csulb.edu/~cwallis/artificialn/History.htm)

- Função linear

p	q	or
∇	∇	∇
∇	⊞	∇
⊞	∇	∇
⊞	⊞	⊞

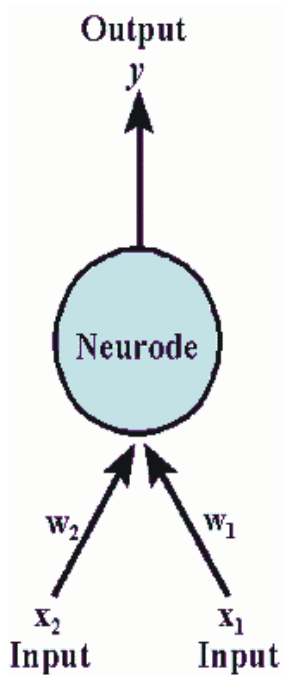


p	q	xor
∇	∇	⊞
∇	⊞	∇
⊞	∇	∇
⊞	⊞	⊞



## O “cérebro” artificial

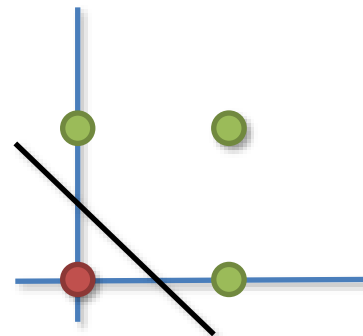
- Perceptron (por volta de 1960)



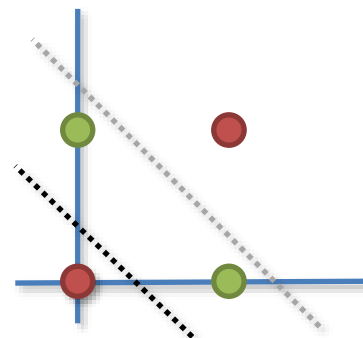
[web.csulb.edu/~cwallis/artificialn/History.htm](http://web.csulb.edu/~cwallis/artificialn/History.htm)

- Função linear

p	q	or
∇	∇	∇
∇	⊞	∇
⊞	∇	∇
⊞	⊞	⊞

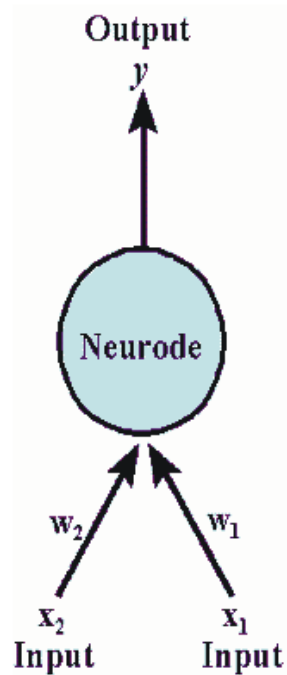


p	q	xor
∇	∇	⊞
∇	⊞	∇
⊞	∇	∇
⊞	⊞	⊞



## O “cérebro” artificial

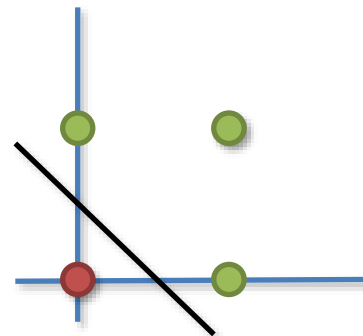
- Perceptron (por volta de 1960)



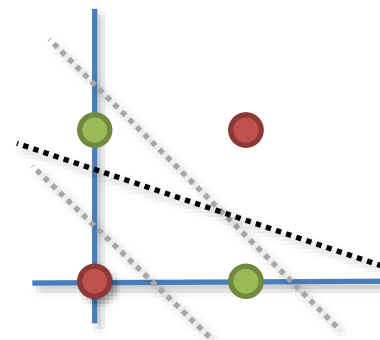
[web.csulb.edu/~cwallis/artificialn/History.htm](http://web.csulb.edu/~cwallis/artificialn/History.htm)

- Função linear

p	q	or
V	V	V
V	F	V
F	V	V
F	F	F

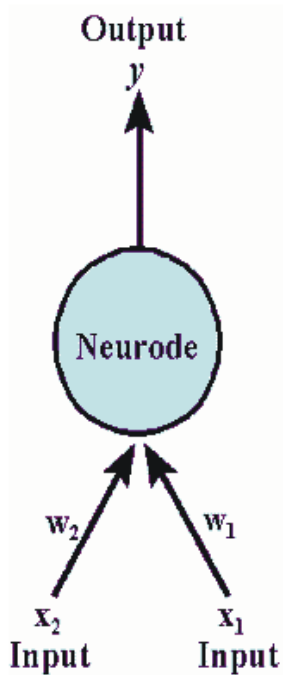


p	q	xor
V	V	F
V	F	V
F	V	V
F	F	F



## 0 “cérebro” artificial

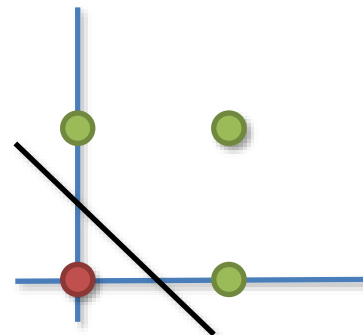
- Perceptron (por volta de 1960)



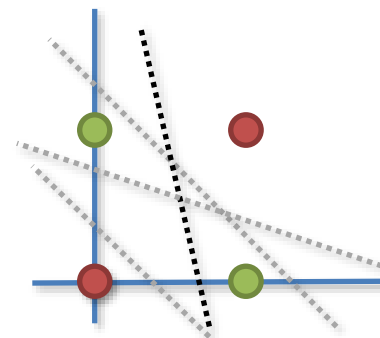
[web.csulb.edu/~cwallis/artificialn/History.htm](http://web.csulb.edu/~cwallis/artificialn/History.htm)

- Função linear

p	q	or
V	V	V
V	F	V
F	V	V
F	F	F



p	q	xor
V	V	F
V	F	V
F	V	V
F	F	F







**ISLab**

Synthetic Intelligence Lab

## O “cérebro” artificial

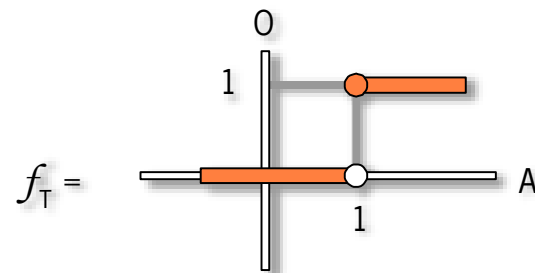


## Problema: XOR

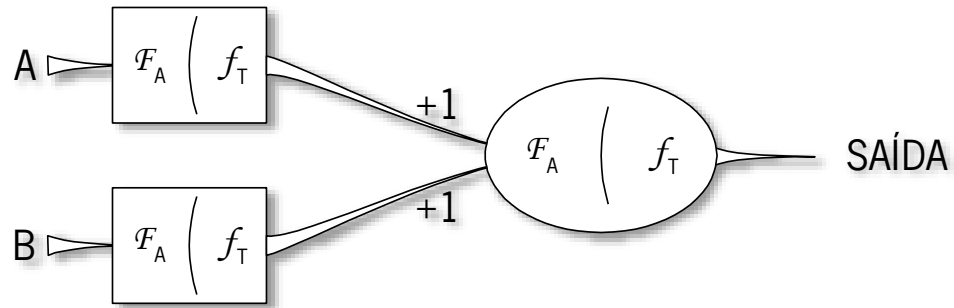
- Função de ativação:  
 $F_A = \sum \text{entradas} \times \text{pesos}$

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

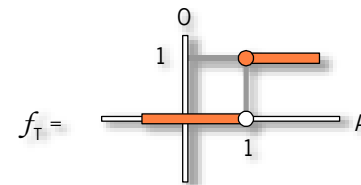
- Função de transferência:



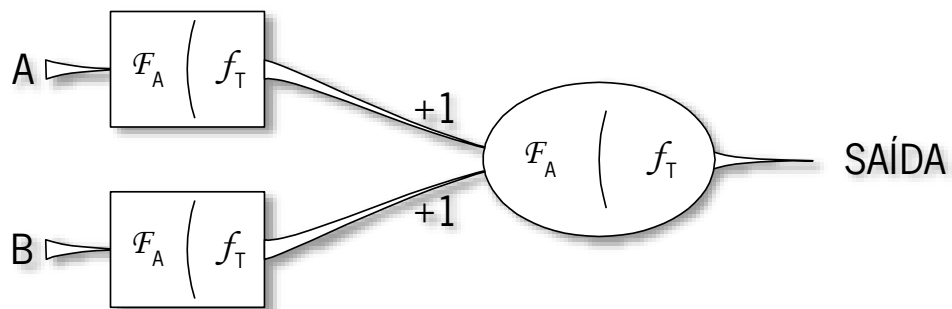
## Problema: XOR Perceptron



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

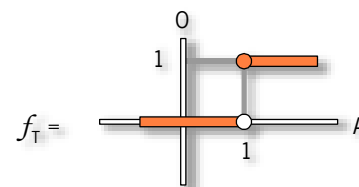


## Problema: XOR Perceptron



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

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

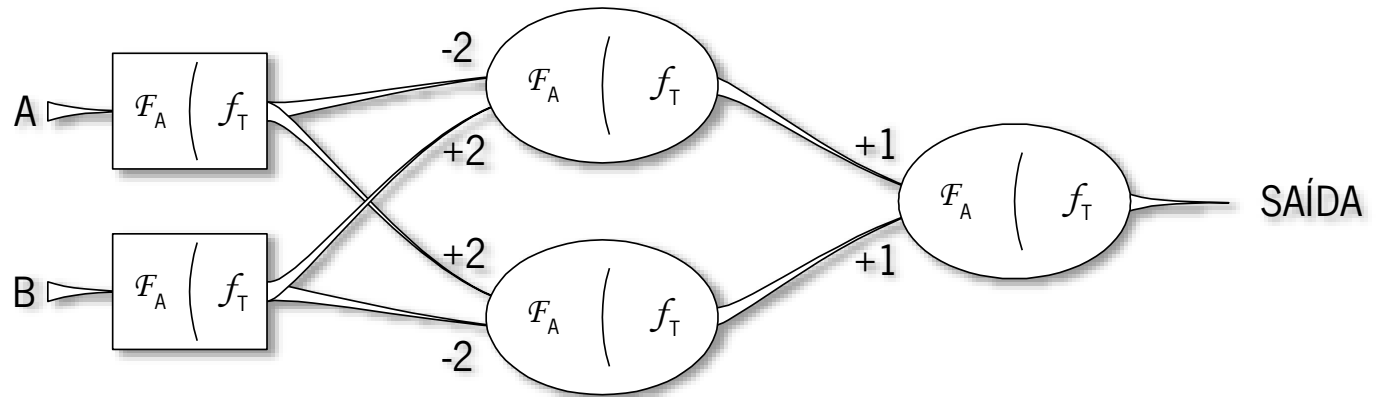




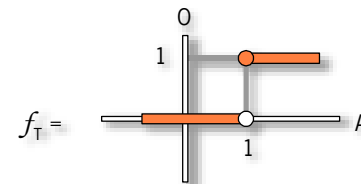
# ISLab

Synthetic Intelligence Lab

## Problema: XOR Multi-layer Perceptron



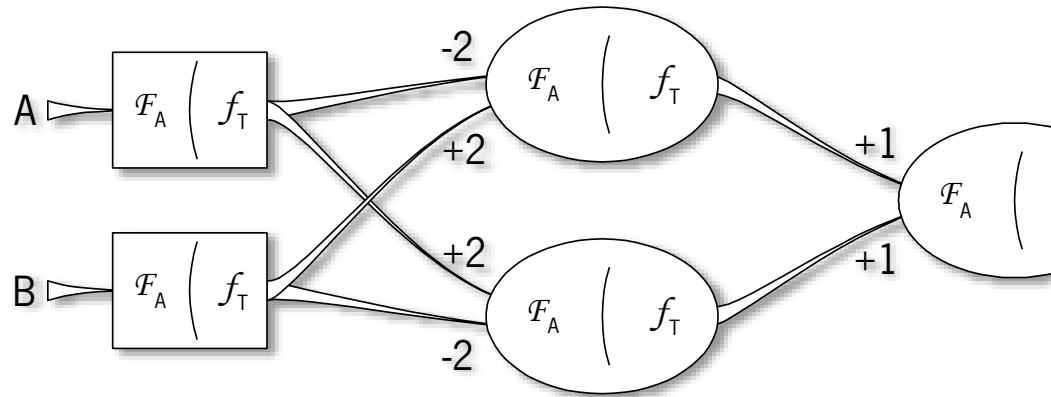
$$\mathcal{F}_A = \sum \text{entradas} \times \text{pesos}$$





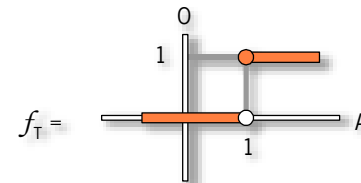


## Problema: XOR Multi-layer Perceptron



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

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

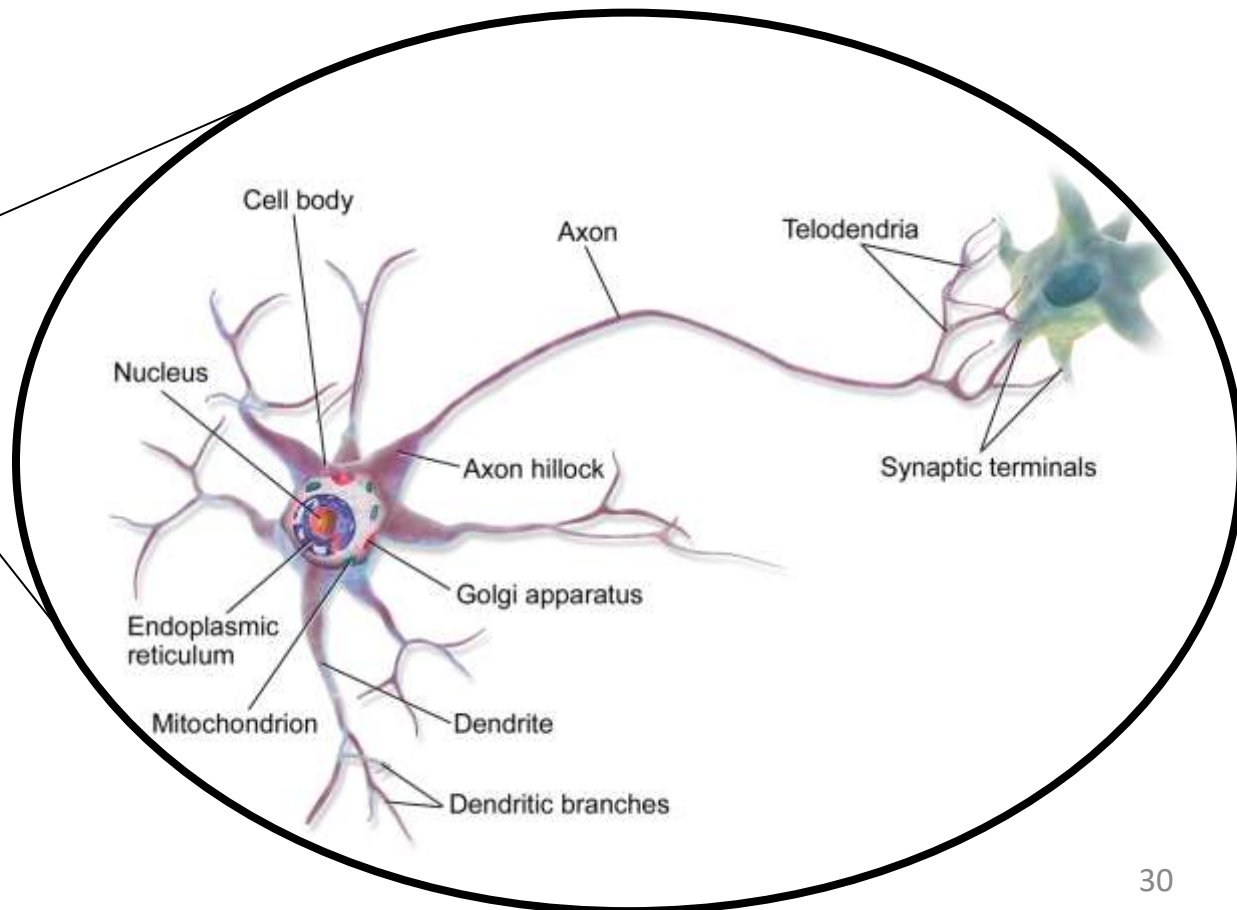
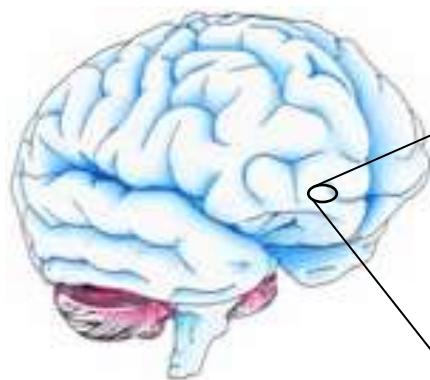




# ISLab

Synthetic Intelligence Lab

## Conceitos e definições

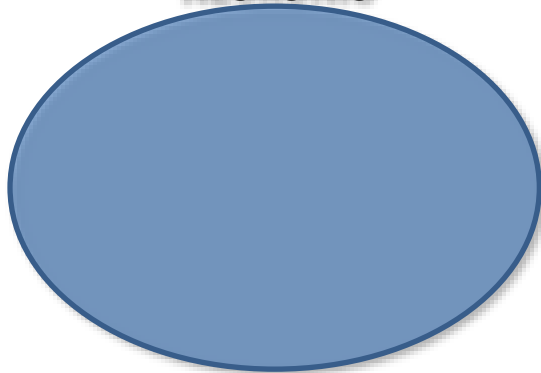


## Conceitos e definições

### Neurónio

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

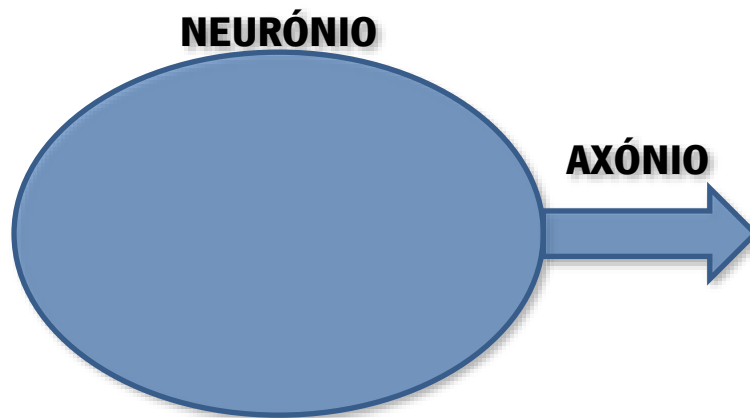
**NEURÓNIO**



## Conceitos e definições

### Axónio

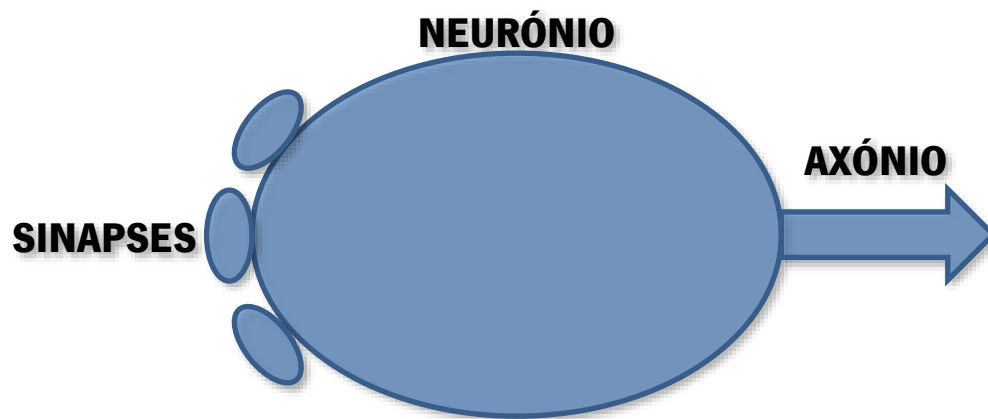
- **Via de comunicação** entre os neurónios.
- Pode **ligar qualquer neurónio**, incluindo o próprio.
- As ligações podem **variar** ao longo do **tempo**.
- A informação circula em **um só sentido**.



## 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).

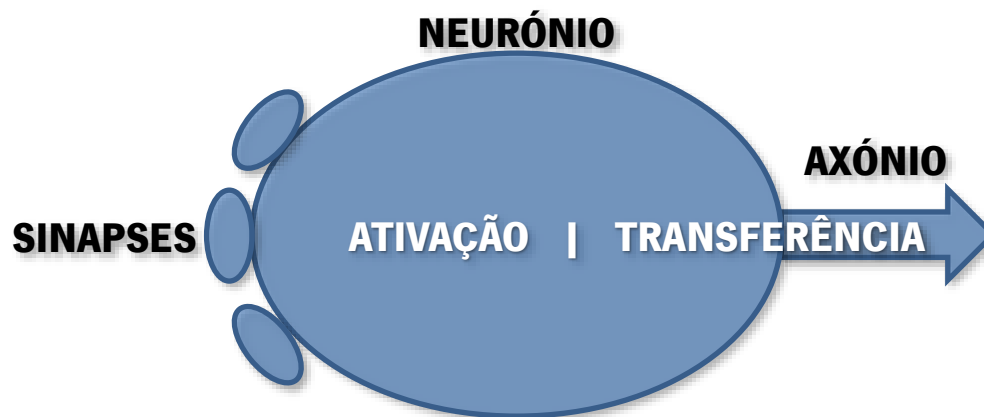




## Conceitos e definições

### Transferência

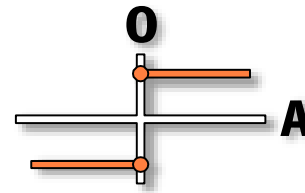
- O valor de transferência de um neurónio determina **o valor** que é **colocado na saída** (transferido através do axónio).
- É calculado como uma função do valor de ativação (eventualmente com algum efeito de memória).



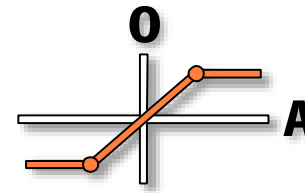
## Tarefas dos neurónios

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

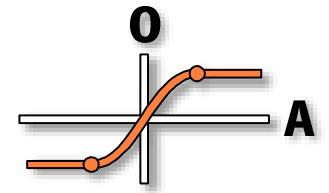
$$O_i = f_T ( A_i )$$



Binária ou Escada



Linear



Sigmoid

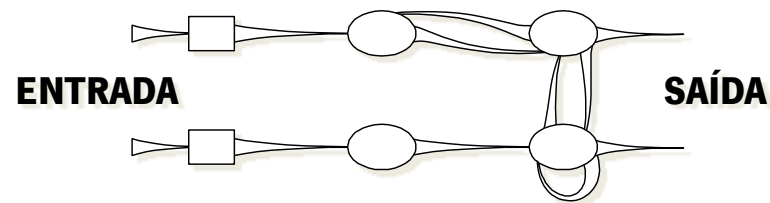
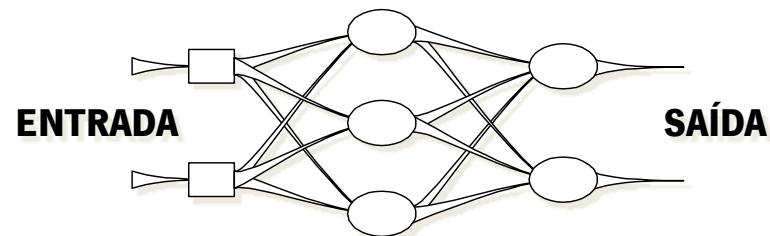
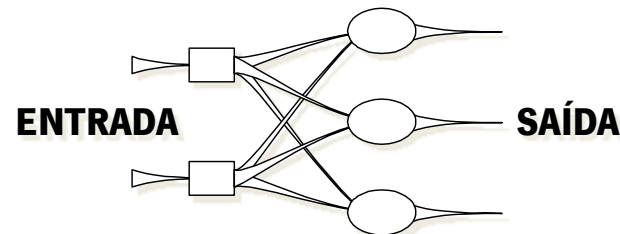
- Cálculo do valor de ativação (  $A_j$  ).
- Varia no tempo com o seu próprio valor e o de outras entradas (  $w_i$  ;  $I$  ):

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

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

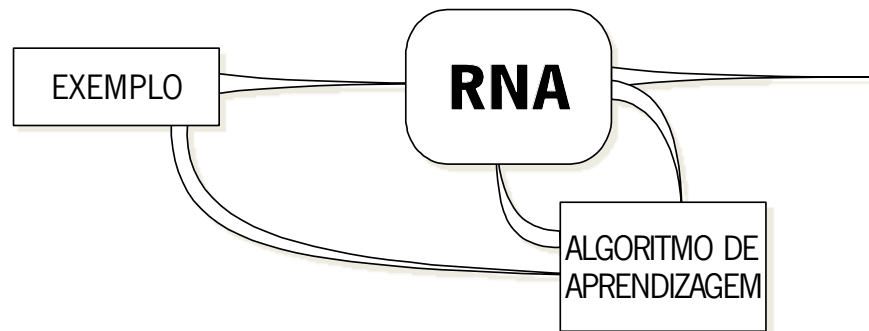
- Arquitetura *Feed forward*, de uma só camada:  
(*Perceptron*)
- Arquitetura *Feed forward*, multi-camada:  
(*Multi-layer Perceptron*)
- Arquitetura Recorrente

## Organização dos neurónios



## Aprendizagem (treino)

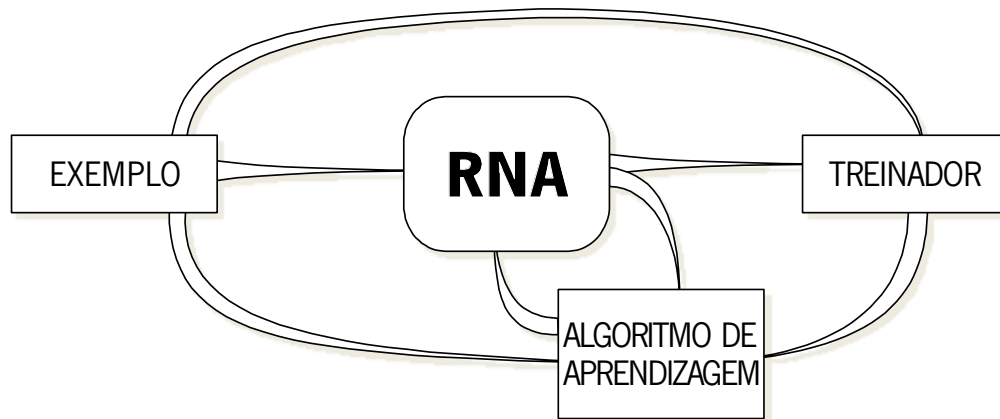
- Sem supervisão:



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

## Aprendizagem (treino)

- 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 deve 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:

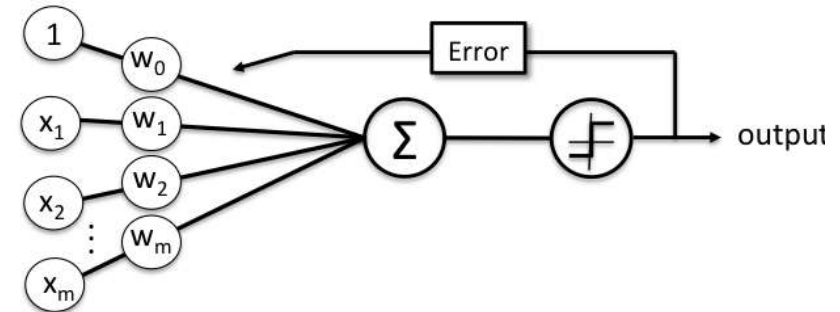
- *Hebbian Learning Rule*

- *Perceptron Learning Rule*

- Desenvolvida para aprendizagem supervisionada;
- 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 forma:

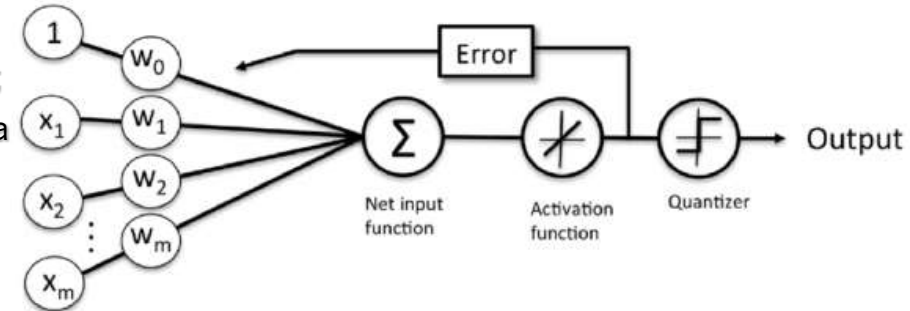
$$\sum_i \sum_j (E_{ij} - O_{ij})^2$$

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



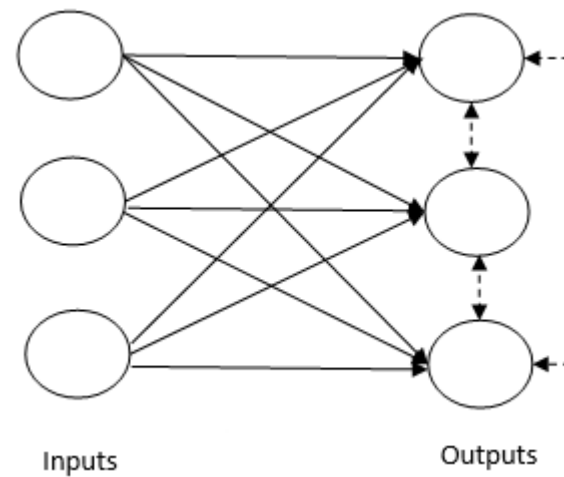
## Regras de aprendizagem

- 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 Marcian Hoff;
    - A principal diferença para *Perceptron Learning* é a de que é usado um sinal linear e não binário para cálculo do erro e consequente atualização 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 declarado 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)*
  - *Correlation Learning Rule*
  - *Outstar Learning Rule (Grossberg Rule)*



## Treino de uma RNA (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.





**ISLab**

Synthetic Intelligence Lab

## Treino de RNAs



# TRAINING

- Considere-se uma Rede Neuronal Artificial...

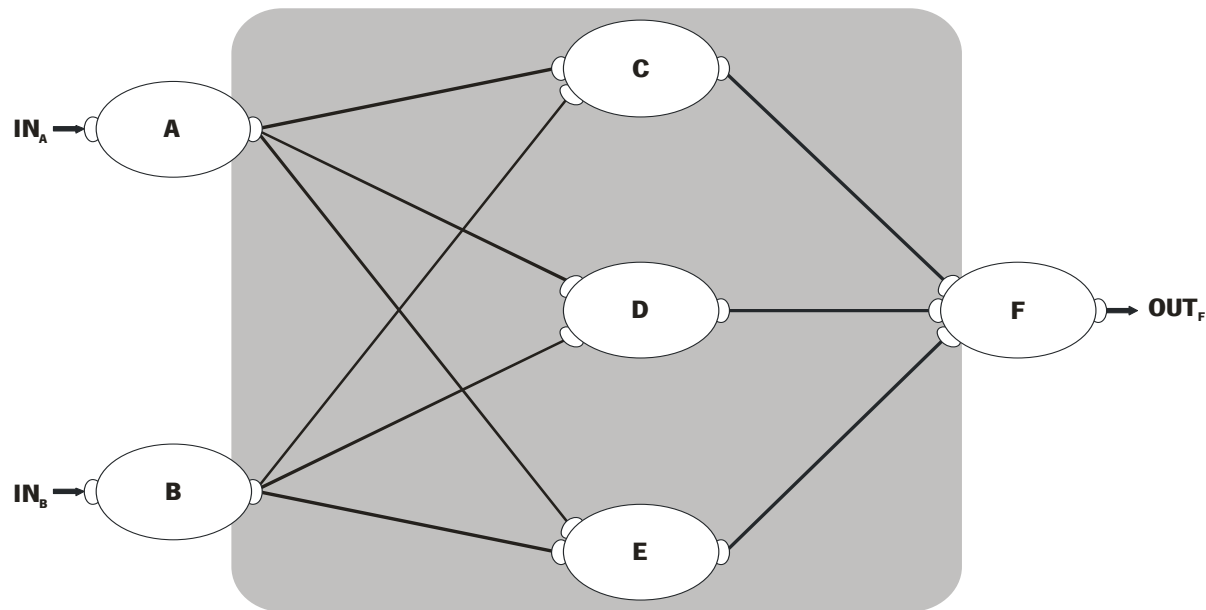
**RNA**

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

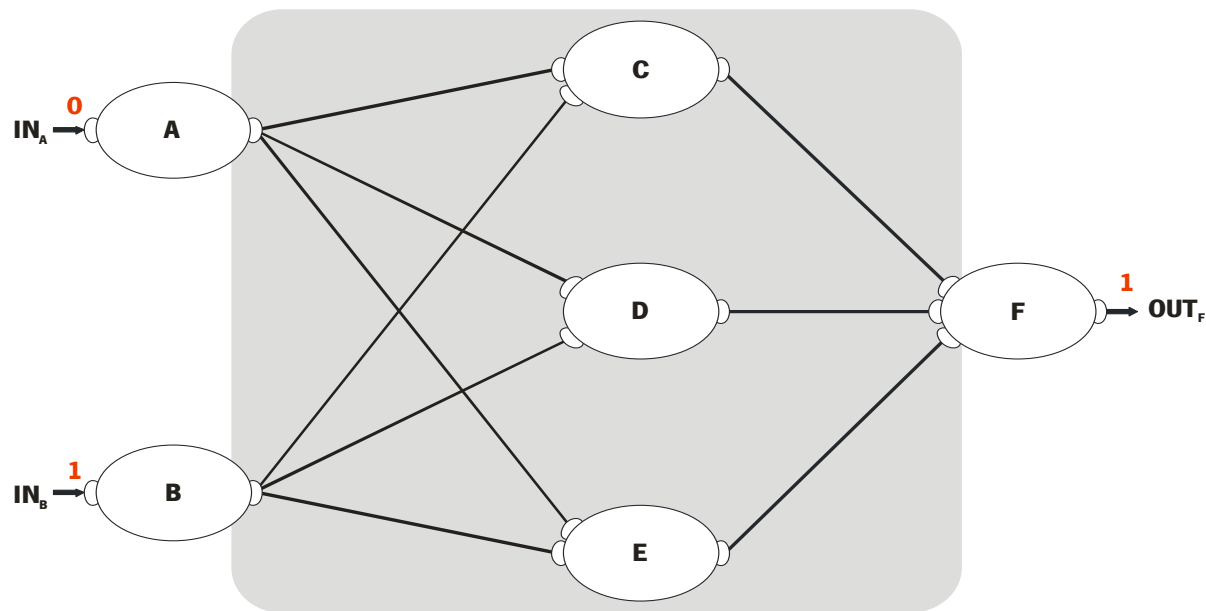




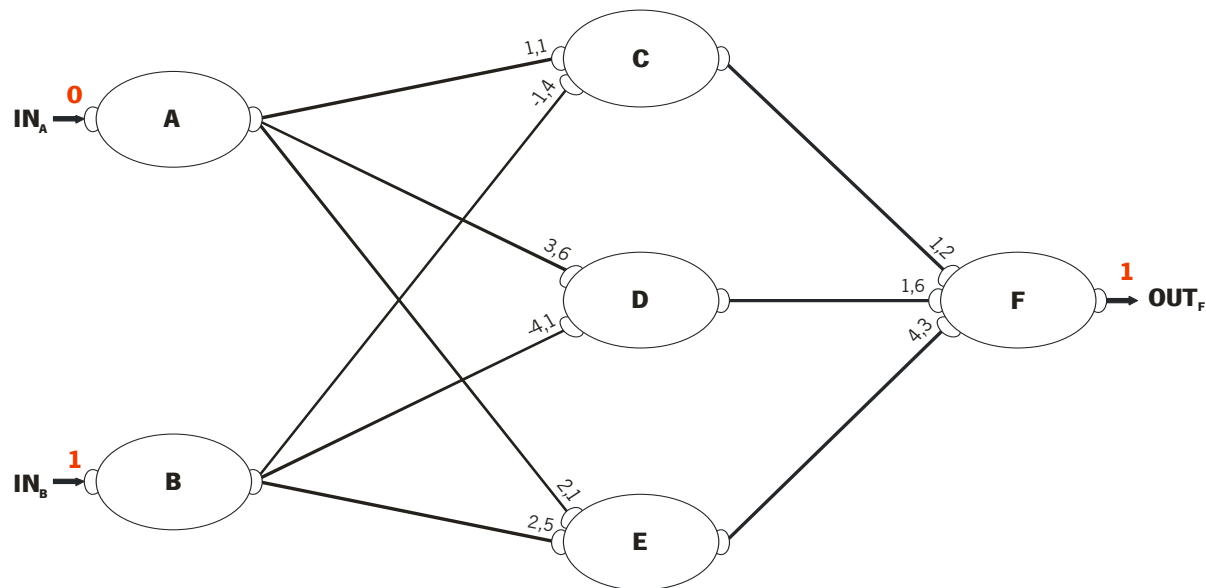
- ... *feed forward*, completamente ligada, com 1 camada intermédia.



- Os exemplos de treino contêm os resultados pretendidos, pelo que a aprendizagem será supervisionada.



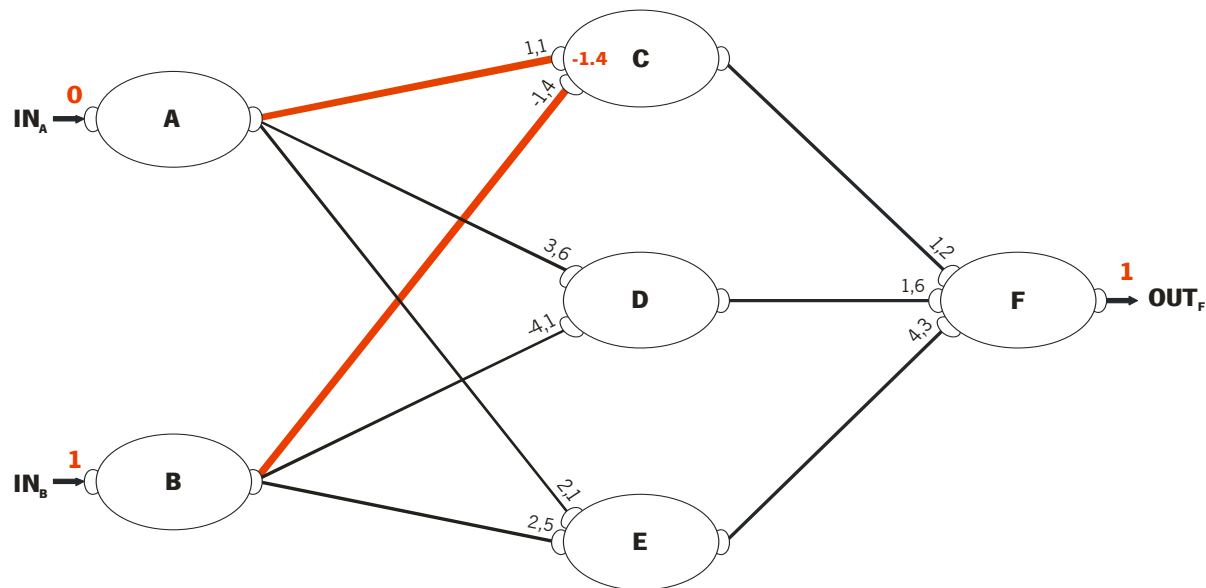
- Atribuição aleatória dos pesos às sinapses.



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

$$f_r(A) = A$$

- Fluxo dos dados pela rede, calculando os valores de ativação...



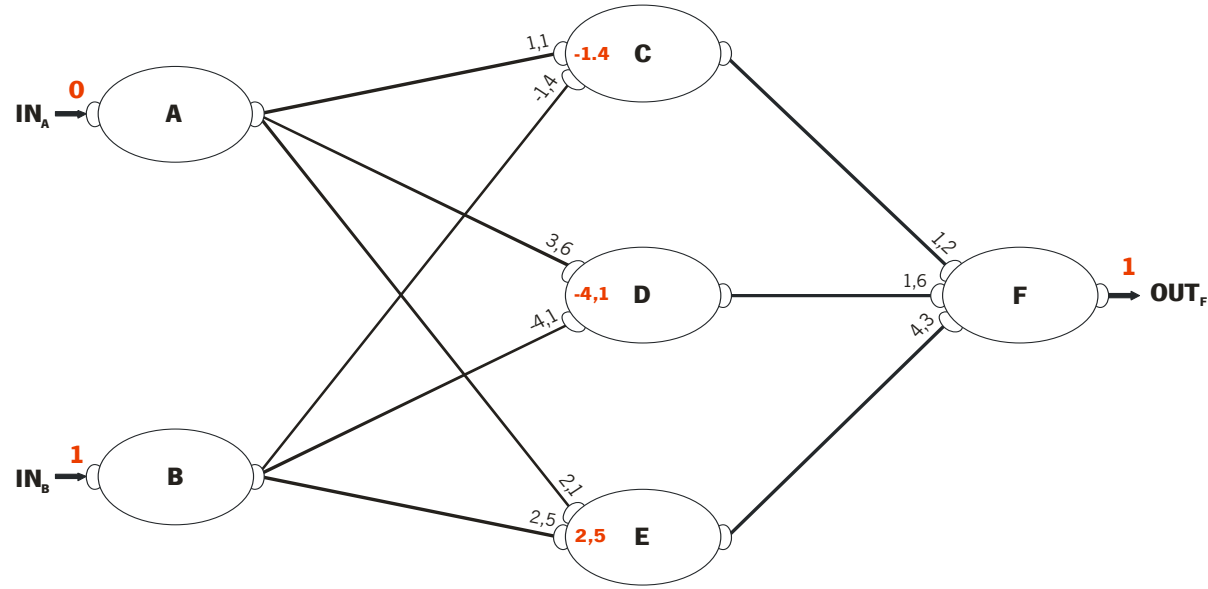
$$f_A(P, E) = \sum P \times E$$

$$f_r(A) = A$$



## Treino de RNAs

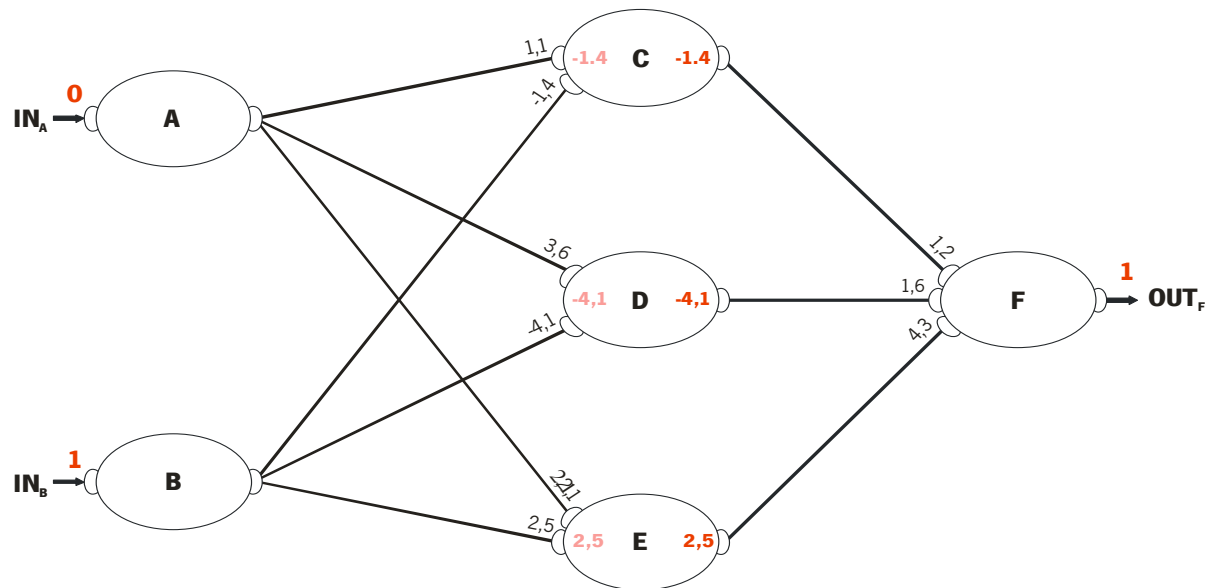
- ... para todos os neurónios da camada intermédia.



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

$$f_r(A) = A$$

- Cálculo do valor de transferência (dado pela função identidade por facilidade de demonstração!).

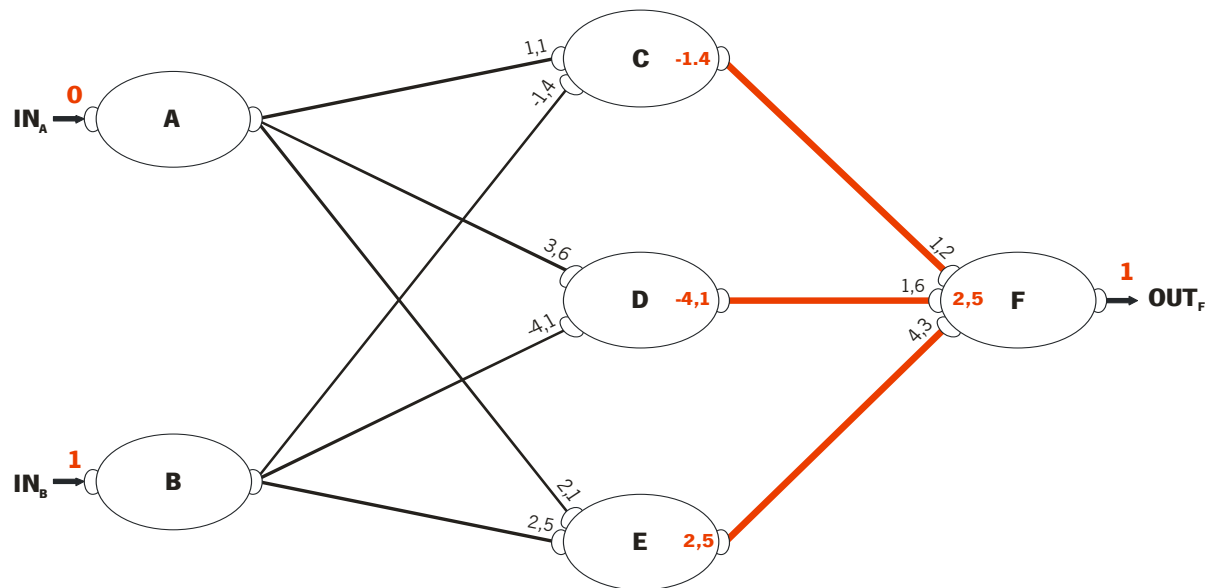


$$f_A(P, E) = \sum P \times E$$

$$f_t(A) = A$$



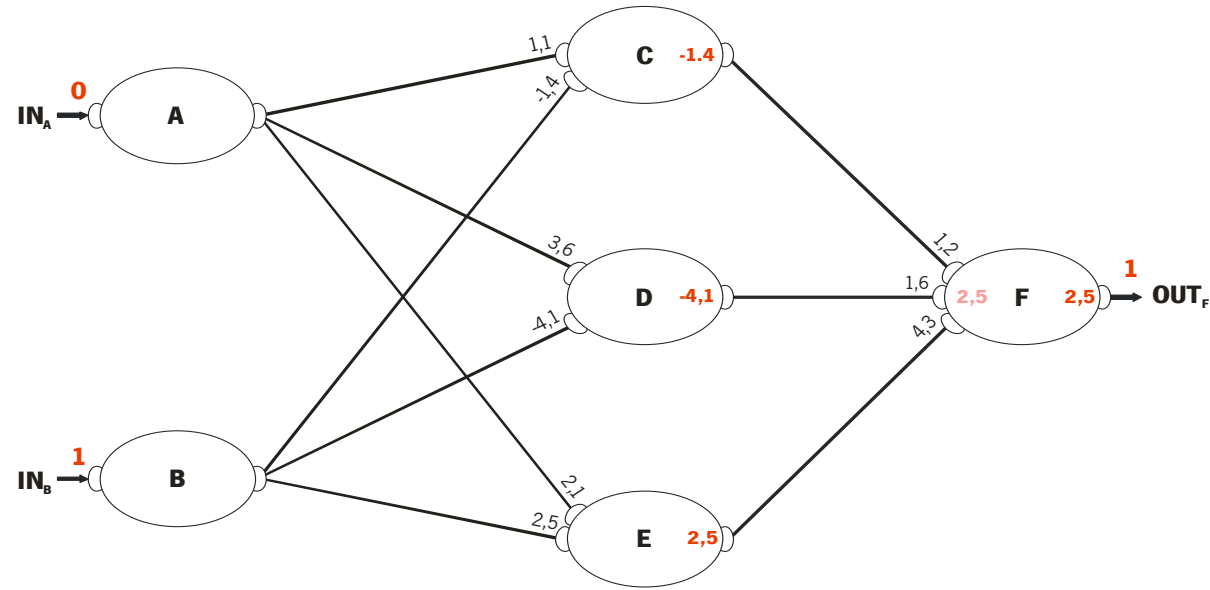
- Fluxo da informação e cálculo do valor de ativação na camada de saída...



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

$$f_r(A) = A$$

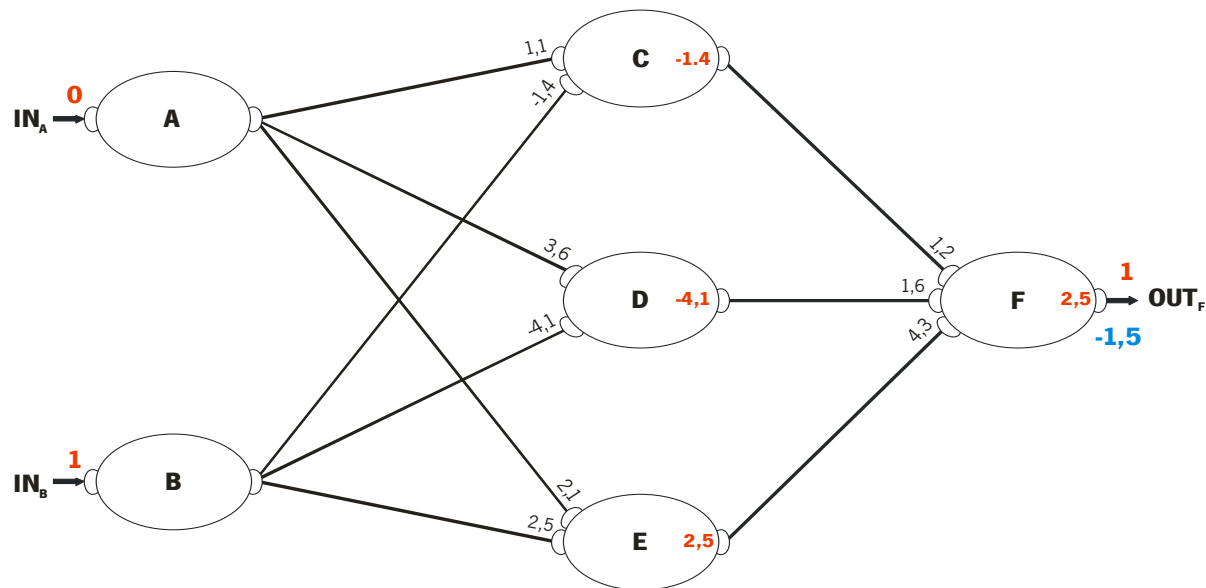
- ... e respetivo valor de transferência.



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

$$f_t(A) = A$$

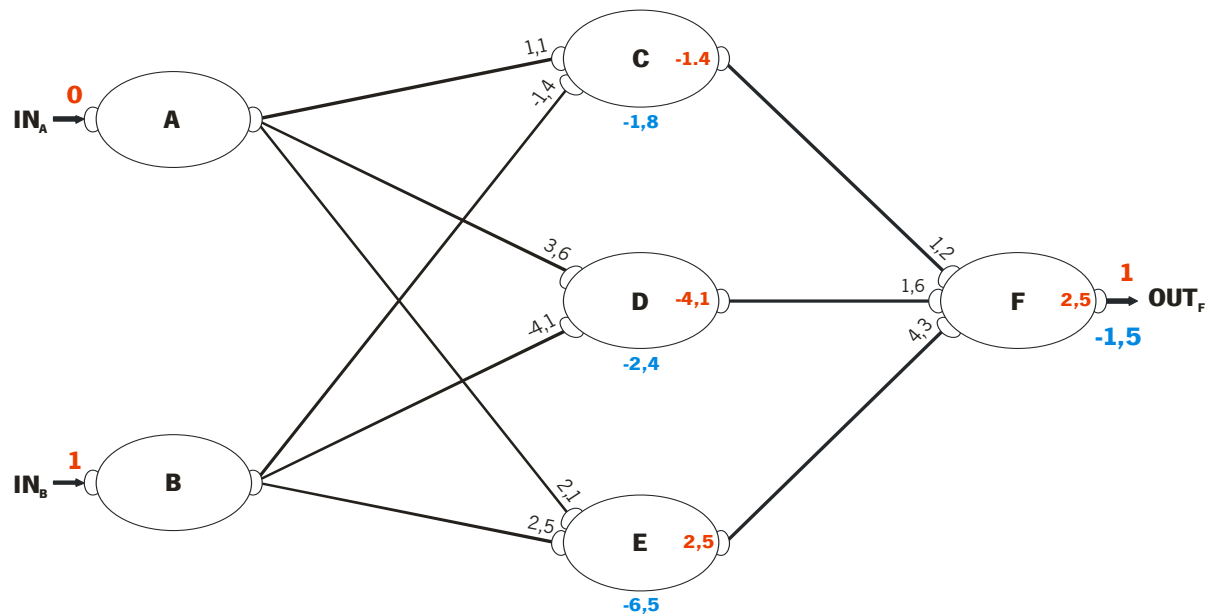
- Cálculo do erro na camada de saída...



$$\mathcal{E} = OUT_d - OUT_c$$

$$\mathcal{E}_c = \mathcal{E} \times P$$

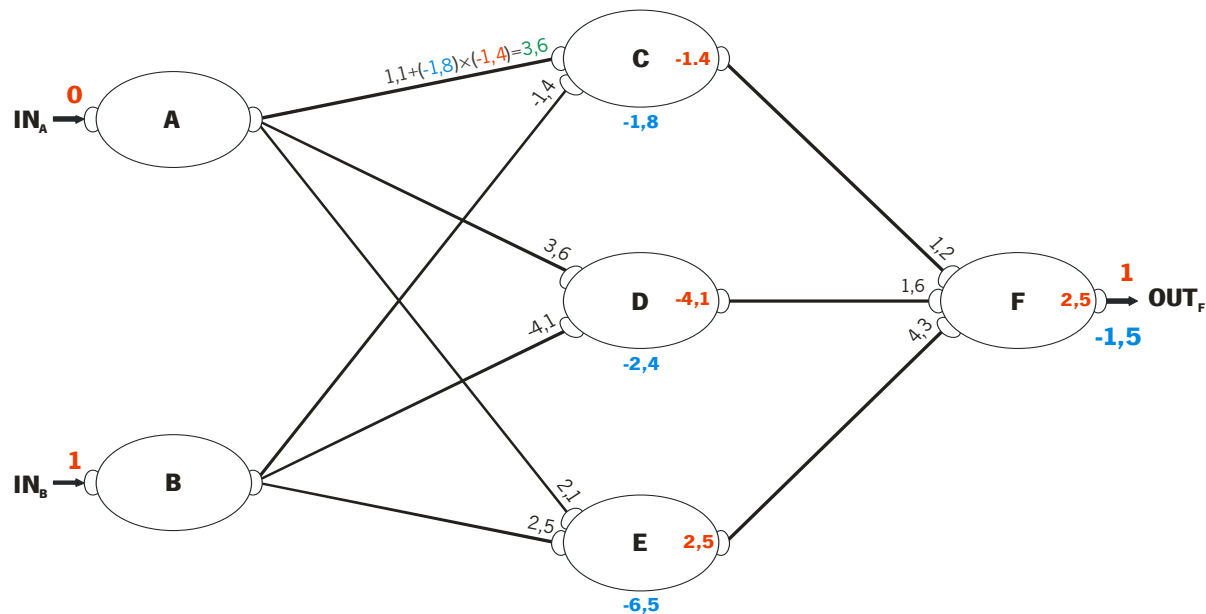
- ... e cálculo do valor estimado do erro na camada intermédia.



$$\mathcal{E} = OUT_D - OUT_C$$

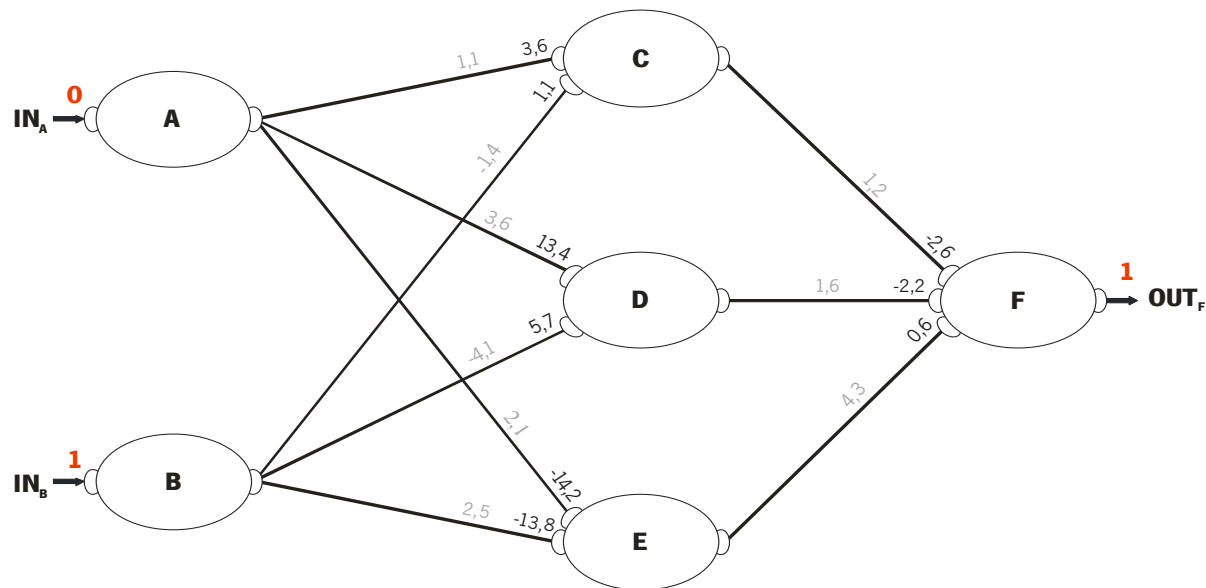
$$\mathcal{E}_{\leftarrow} = \mathcal{E} \times P$$

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



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

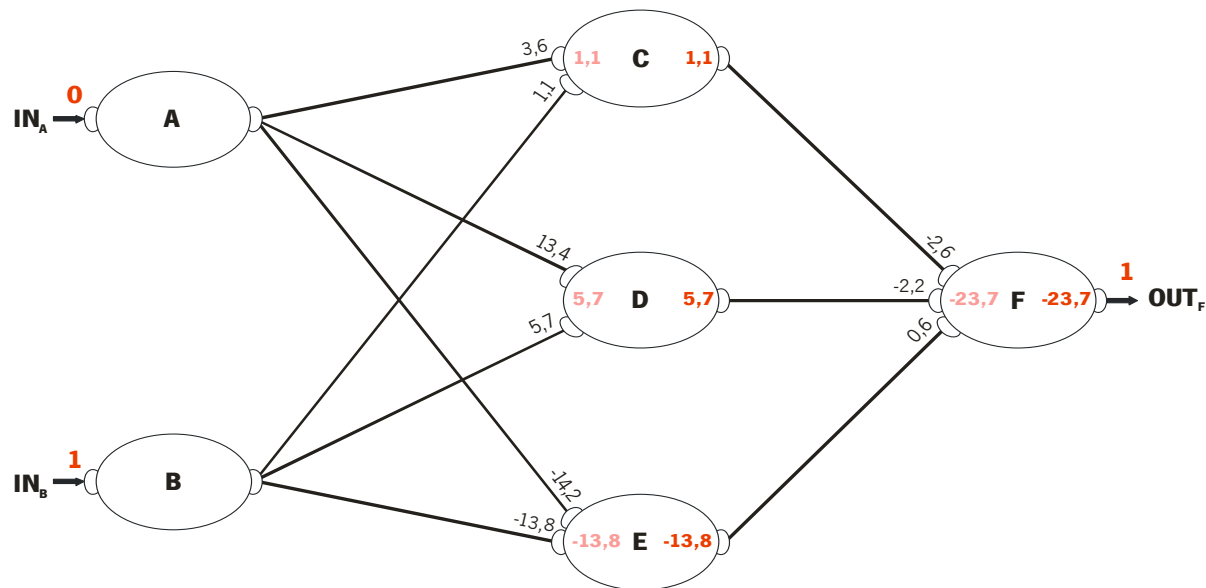
- ... para atualizar os valores das sinapses de todos os neurónios.



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



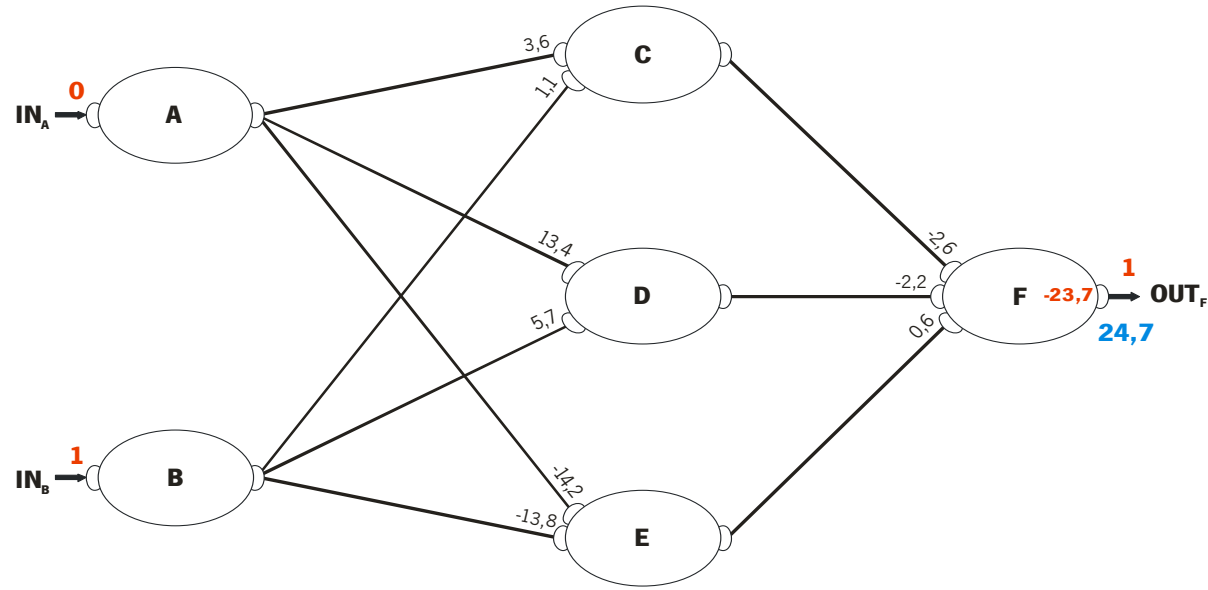
- Segunda iteração da propagação do caso de treino...



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

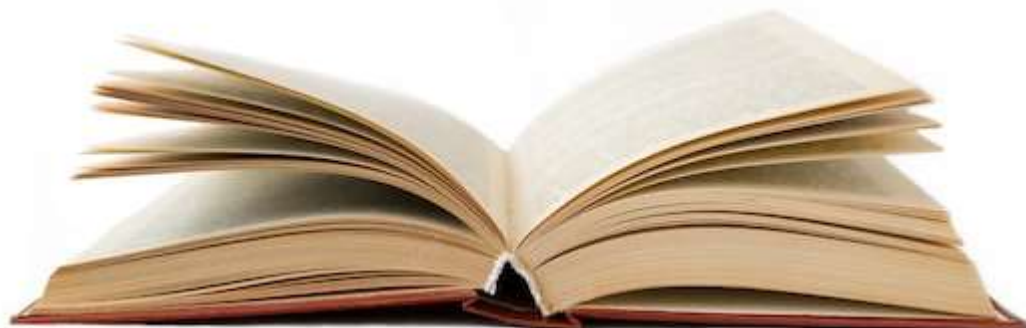
$$f_r(A) = A$$

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



## Referências bibliográficas

- Cortez, P., Neves, J., “Redes Neurais Artificiais”, Unidade de Ensino, Departamento de Informática, Universidade do Minho, 2000;
- Haykin, S., “Neural Networks - A Comprehensive Foundation”, Prentice-Hall, New Jersey, 1999, ISBN 978-0-13-273350-2;
- Bishop, Christopher M., “Neural networks for pattern recognition”, Clarendon Press, 1995, ISBN 978-0-19-853849-3;
- Charu C. Aggarwal, “Neural Networks and Deep Learning”, Springer, 2018, ISBN 978-3-319-94463-0.



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

**ADI**

# **Redes Neurais Artificiais**

Licenciatura em Engenharia Informática, 3º ano

Mestrado integrado em Engenharia Informática, 4º ano