

# Redes Neurais Artificiais

Hélio Pio

# Programação das Aulas

---

Tópico 1: Introdução a Inteligência Artificial

Tópico 2: Agentes Inteligentes

Tópico 3: Fundamentos de Aprendizagem de Máquina

Tópico 4: Redes Neurais Artificiais

Tópico 5: Atividade em Aula – Primeira Avaliação

Tópico 6: Representação da Incerteza e Lógica Fuzzy

Tópico 7: Redes Bayesianas

Tópico 8: Support Vector Machines

Tópico 9: Atividade em Aula – Segunda Avaliação

Tópico 10: Resolução de Problemas por Meio de Busca e Otimização

Tópico 11: Técnicas de Ensemble

Tópico 12: Atividade em Aula – Terceira Avaliação

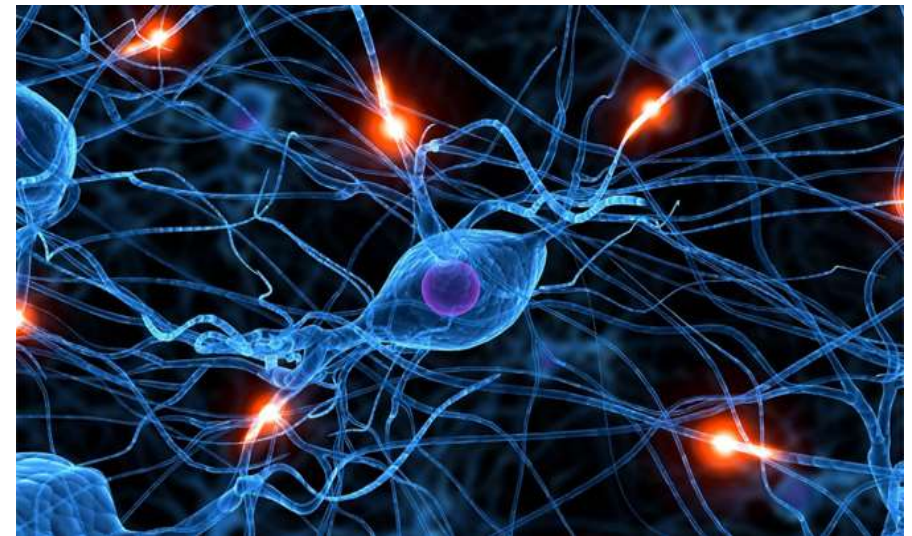
O que é uma rede  
neural artificial?

# Redes Neurais Artificiais

---

O que é uma rede neural artificial?

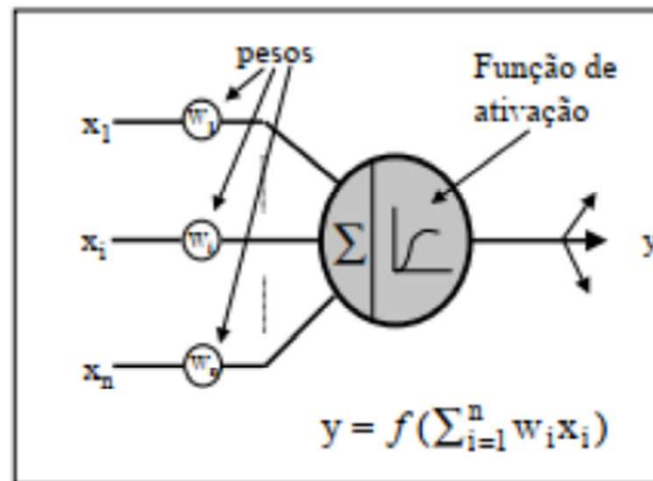
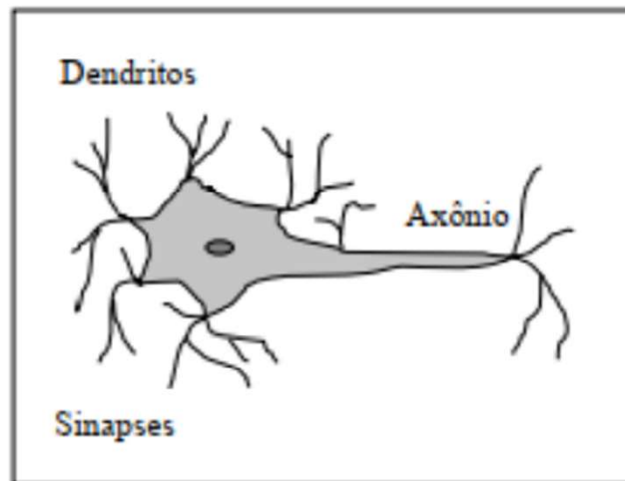
- Inspirada no córtex cerebral humano
  - 100 bilhões de neurônios ( $10^{11}$ ).
  - Cada neurônio com 1.000 dendritos.
  - Logo: 100 trilhões ( $10^{14}$ ) de sinapses.
  - Sinapse: opera a 100 Hz.
  - Velocidade da operação: 10 quatrilhões.
  - ( $10^{16}$ ) de conexões por segundo.



# Redes Neurais Artificiais

---

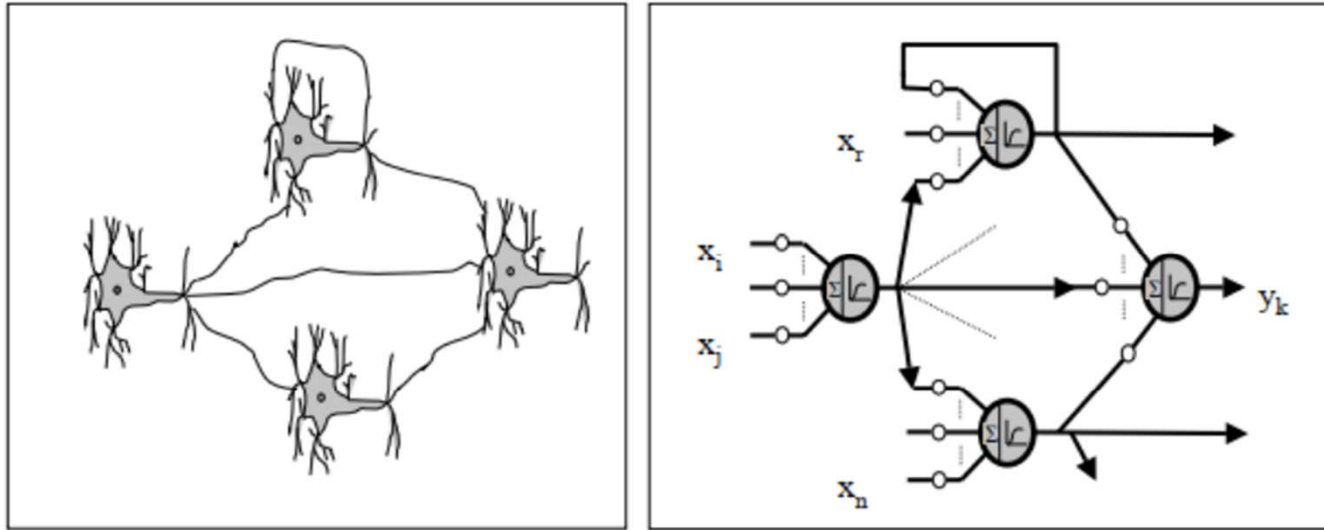
O que é uma rede neural artificial?



# Redes Neurais Artificiais

---

O que é uma rede neural artificial?



# Redes Neurais Artificiais

---

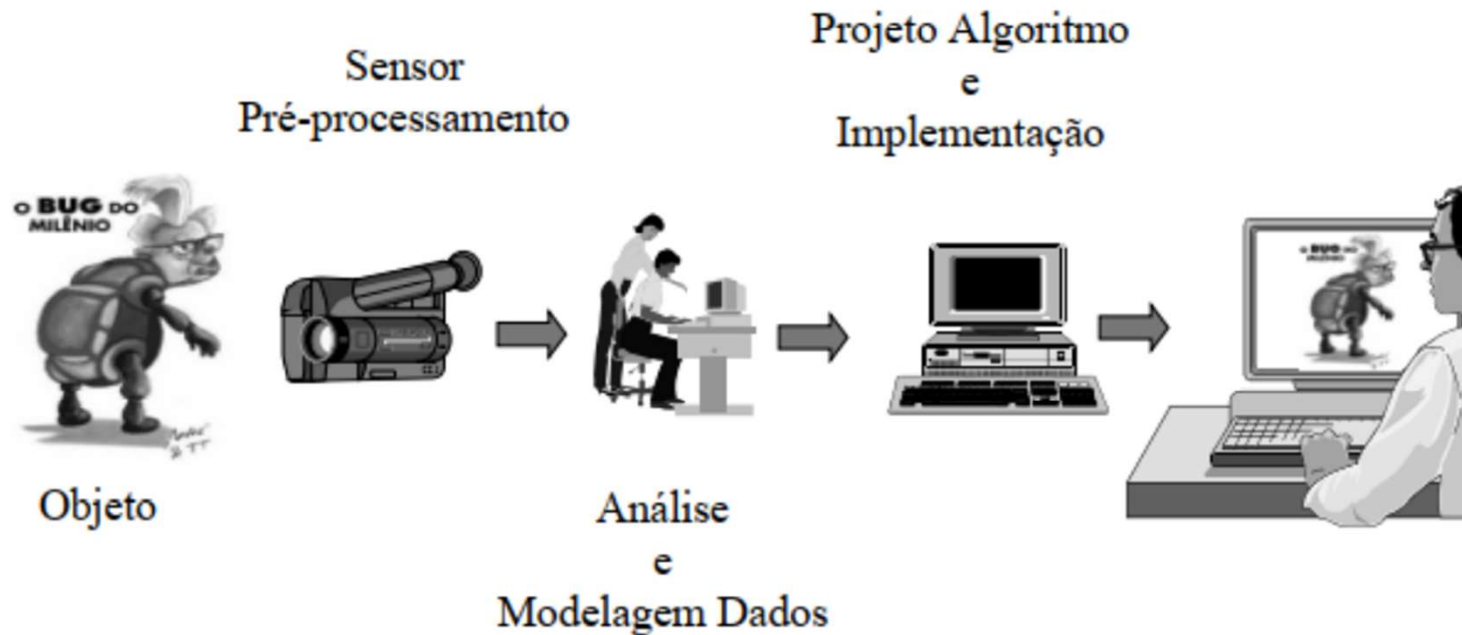
O que é uma rede neural artificial?

- Definições:
  - Redes neurais artificiais são modelos computacionais inspirados no funcionamento do cérebro humano, compostos por camadas de neurônios artificiais interconectados que recebem sinais de entrada, processam esses sinais e geram saídas.
  - Elas possuem conexões ajustáveis entre os neurônios, permitindo que aprendam a mapear entradas para saídas desejadas.

# Redes Neurais Artificiais

---

## Processamento convencional

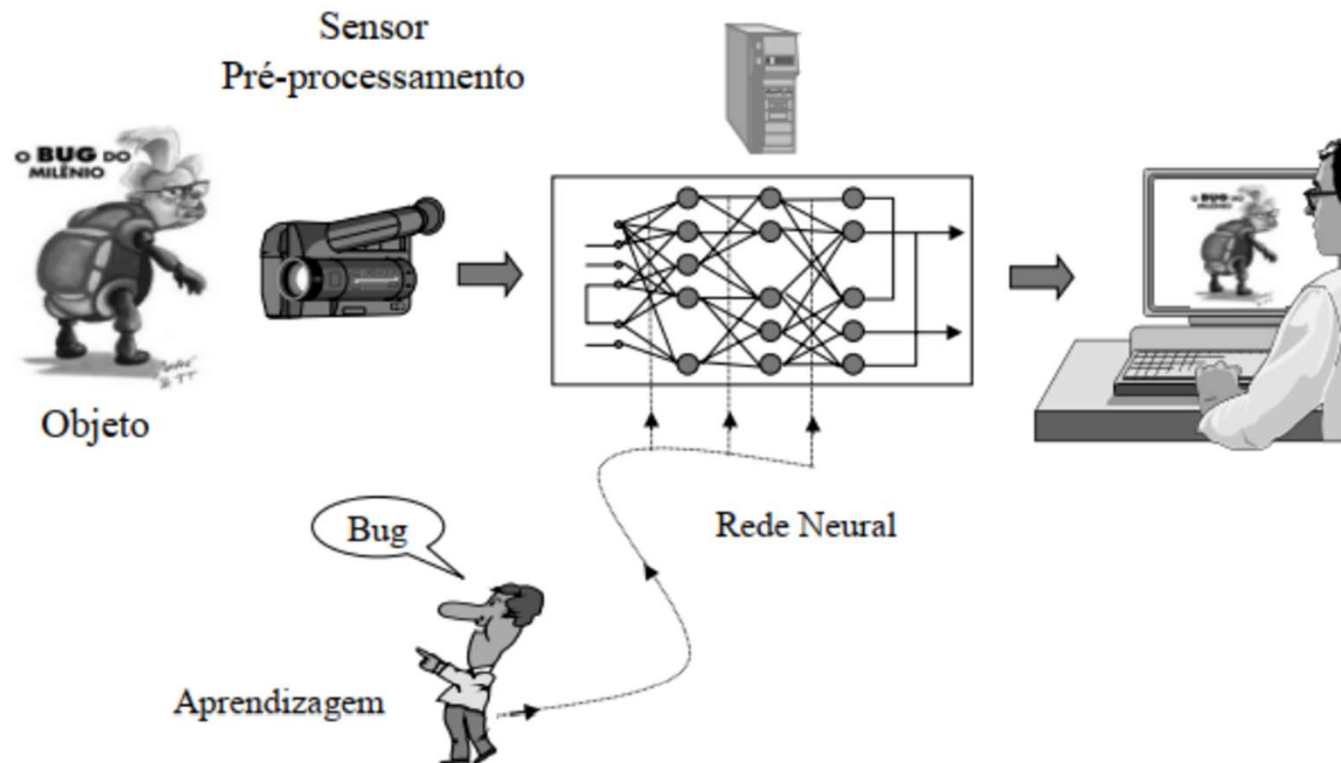




# Redes Neurais Artificiais

---

## Processamento neural



# Redes Neurais Artificiais

---

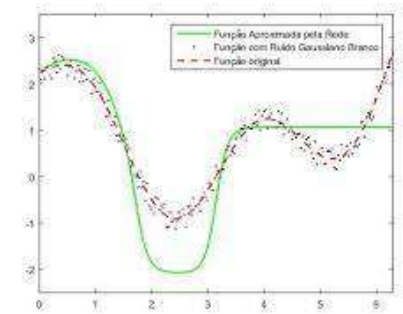
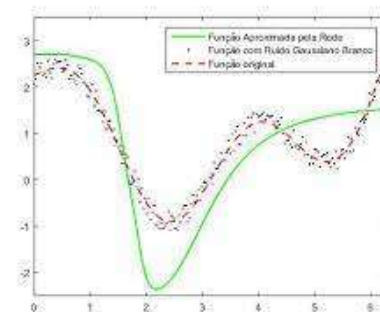
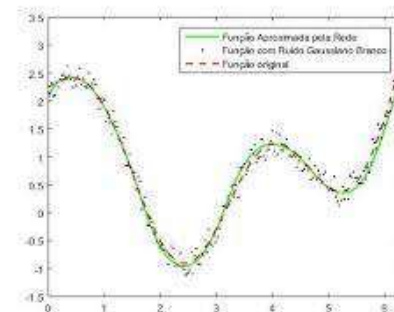
Rede neural artificial (RNA) – Capacidade de generalização

- A rede aprende a lidar com um certo problema.
- Em seguida, é apresentado um problema similar, mas não exatamente o mesmo, nunca antes visto.
- A rede reconhece esse novo problema, oferecendo a mesma solução.

# Redes Neurais Artificiais

## Rede neural artificial (RNA) – Aproximador de funções

- A característica mais significativa de redes neurais está em sua habilidade de aproximar qualquer função contínua ou não contínua com um grau de correção desejado.
- Esta habilidade das redes neurais as tem tornado útil para modelar sistemas não lineares.



# Redes Neurais Artificiais

---

## Rede neural artificial (RNA)

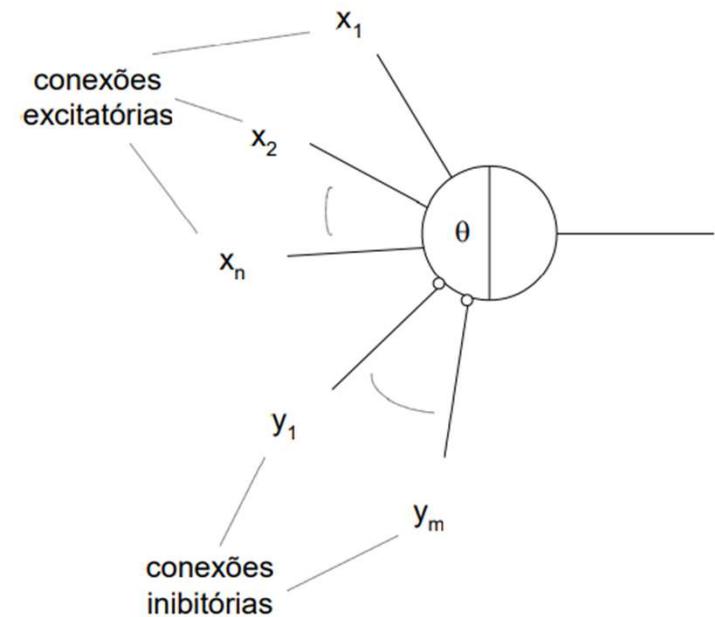
- Mapeamento Entrada-Saída para Aproximação de Funções
  - Objetivo da Aprendizagem: descobrir a função  $f(x)$  dado um número finito (desejável pequeno) de pares entrada-saída  $(x, d)$ .
- Devido à sua estrutura, as Redes Neurais Artificiais são bastante efetivas no aprendizado de padrões a partir de dados:
  - Não-lineares
  - Incompletos
  - Com ruído
  - Até mesmo compostos de exemplos contraditórios.

# Redes Neurais Artificiais

---

## O neurônio de McCulloch e Pitts

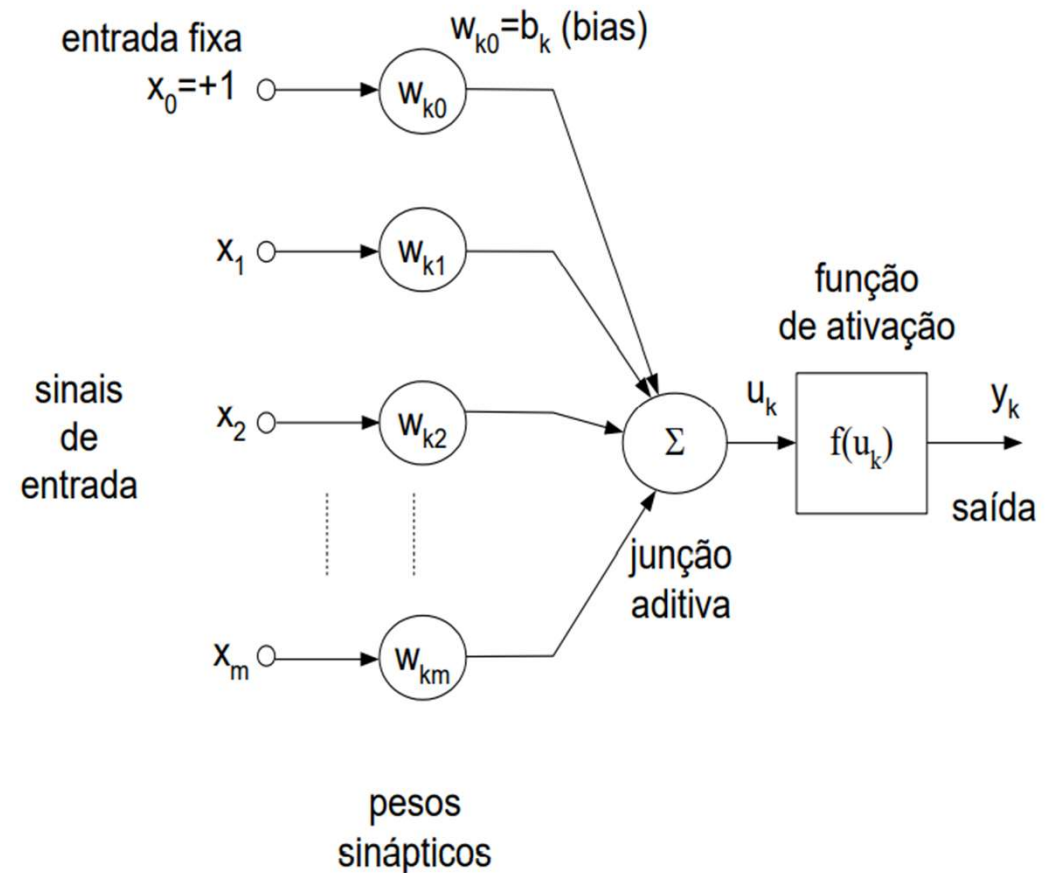
- Consiste basicamente de um neurônio que executa uma função lógica.
- Os nós produzem somente resultados binários e as conexões transmitem exclusivamente zeros e uns.
- As redes são compostas de conexões sem peso, de tipos excitatórios e inibitórios.
- Cada unidade é caracterizada por um certo limiar (*threshold*)  $q$ .



# Redes Neurais Artificiais

## Componentes do neurônio artificial

- As sinapses (entradas), com seus pesos associados.
- A junção somadora.
- A função de ativação.

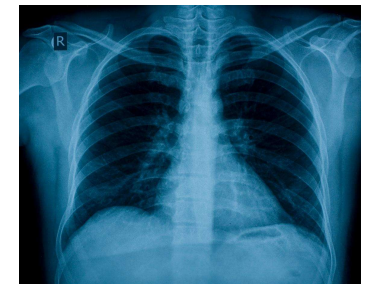
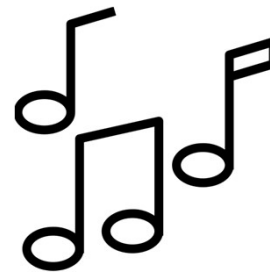
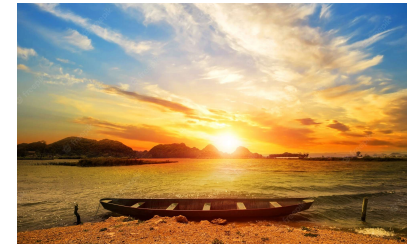


# Redes Neurais Artificiais

---

Componentes do neurônio artificial (entradas e saídas):

$[x_0, x_1, x_2, \dots, x_n]$

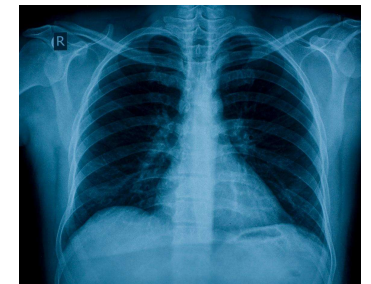
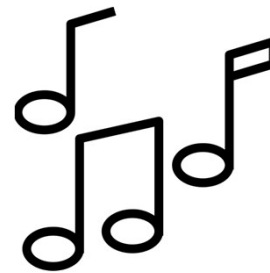
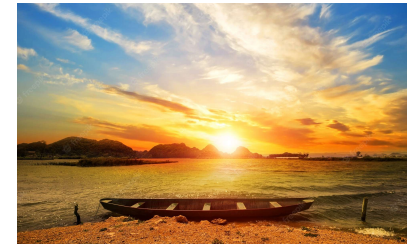


# Redes Neurais Artificiais

---

Componentes do neurônio artificial (entradas e saídas):

$[y_0, y_1, y_2, \dots, y_n]$

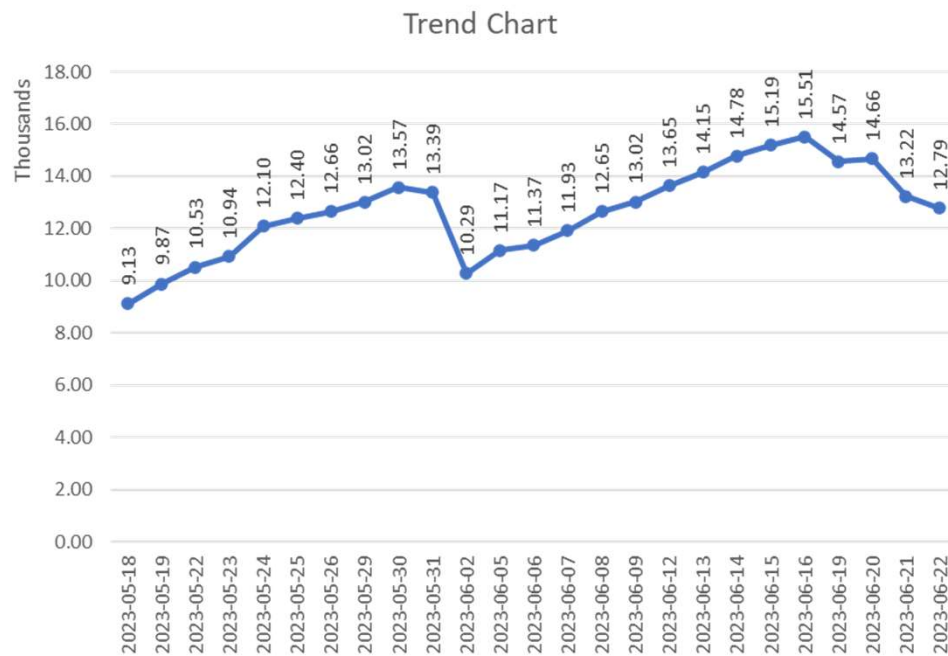




# Redes Neurais Artificiais

Componentes do neurônio artificial (entradas e saídas):

$[x_0, x_1, x_2, \dots, x_n]$



$X_0 =$  [9.13, 9.87, 10.53, 10.94, 12.10, 12.40, 12.66, 13.02]

$Y_0 =$  [13.57, 13.39, 10.29]

$X_1 =$  [9.87, 10.53, 10.94, 12.10, 12.40, 12.66, 13.02, 13.57]

$Y_1 =$  [13.39, 10.29, 11.17]

# Redes Neurais Artificiais

---

## RNA – Princípio de funcionamento

- Sinais são apresentados à entrada ( $x_1$  à  $x_m$ ).
- Cada sinal é multiplicado por um peso que indica sua influência na saída da unidade ( $w_k$ ).
- É feita a soma ponderada dos sinais que produz um nível de atividade ( $u_k$ ).
- A função de ativação  $f(u_k)$  assume o papel de limitar a saída e introduzir não-linearidade ao modelo.
- O viés (ou bias)  $b_k$  tem o papel de aumentar ou diminuir a influência do valor das entradas.
- É possível considerar o bias como uma entrada de valor constante 1, multiplicado por um peso igual a  $b_k$ .

# Redes Neurais Artificiais

---

RNA – Expressão matemática do neurônio artificial

- Matematicamente a saída pode ser expressa por:

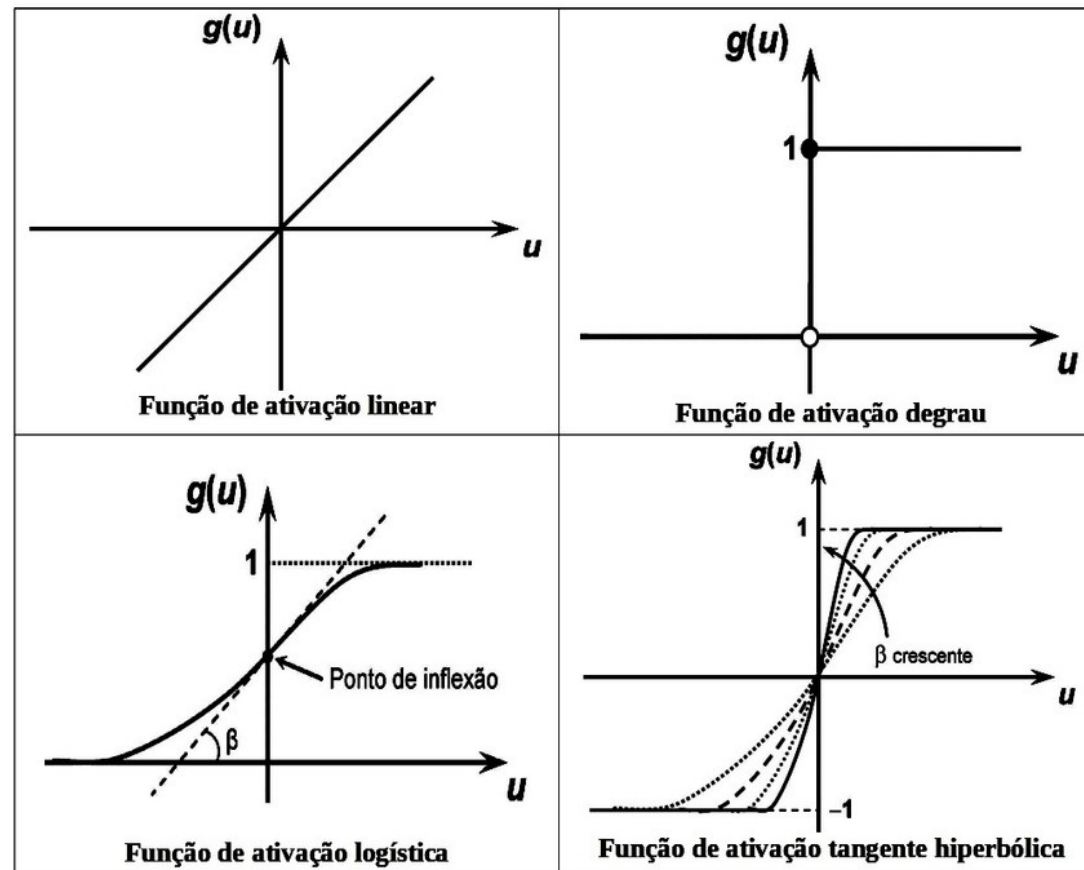
$$y_k = f(u_k) = f\left(\sum_{j=0}^m w_{kj} x_j\right)$$

ou considerando o bias como entrada de valor  $x_0=1$  e peso  $w_{k0}=b_k$ ,

$$y_k = f(u_k) = f\left(\sum_{j=1}^m w_{kj} x_j + b_k\right)$$

# Redes Neurais Artificiais

## RNA – Funções de ativação



# Redes Neurais Artificiais

---

## RNA – Separação linear

- Sabe-se que se formarmos uma combinação linear de duas variáveis, e igualá-la a um número, então os pontos no espaço bidimensional podem ser divididos em três categorias:

a) pontos pertencentes à linha com coordenadas tais que:  $w_1 * x_1 + w_2 * x_2 = q$

b) pontos em um lado da linha tem coordenadas tais que:  $w_1 * x_1 + w_2 * x_2 < q$

c) pontos no outro lado da linha tem coordenadas tais que:  $w_1 * x_1 + w_2 * x_2 > q$

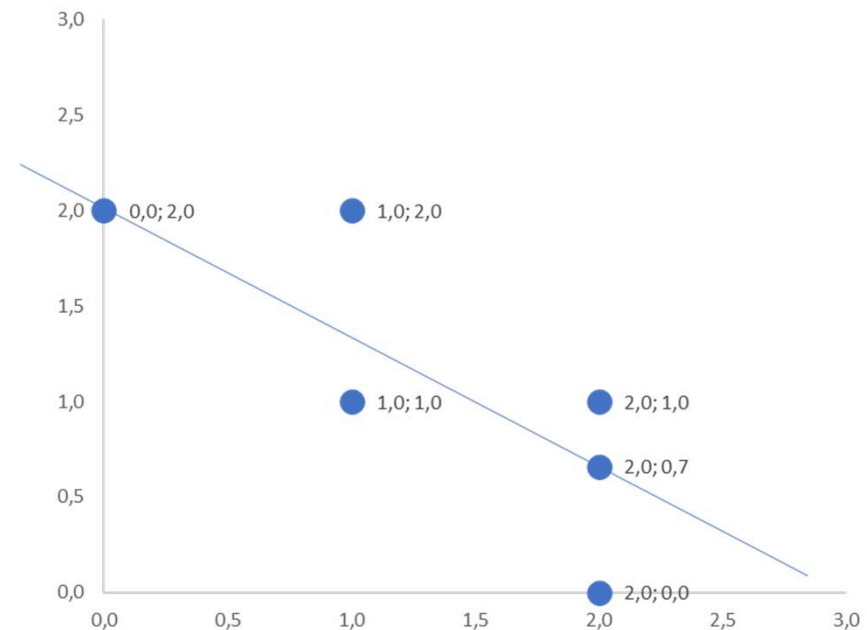
d) bias  $b = -q = w_0$ :  $y = f(w_1 * x_1 + w_2 * x_2 + w_0)$

# Redes Neurais Artificiais

## RNA – Separação linear - Exemplo

- Pontos:  $2 * x_1 + 3 * x_2$
- Posição dos pontos em função da linha de delimitação:  $2 * x_1 + 3 * x_2 = 6$
- Linha:  $2 * x_1 + 3 * x_2 = 6$
- Acima  $2 * x_1 + 3 * x_2 > 6$
- Abaixo  $2 * x_1 + 3 * x_2 < 6$

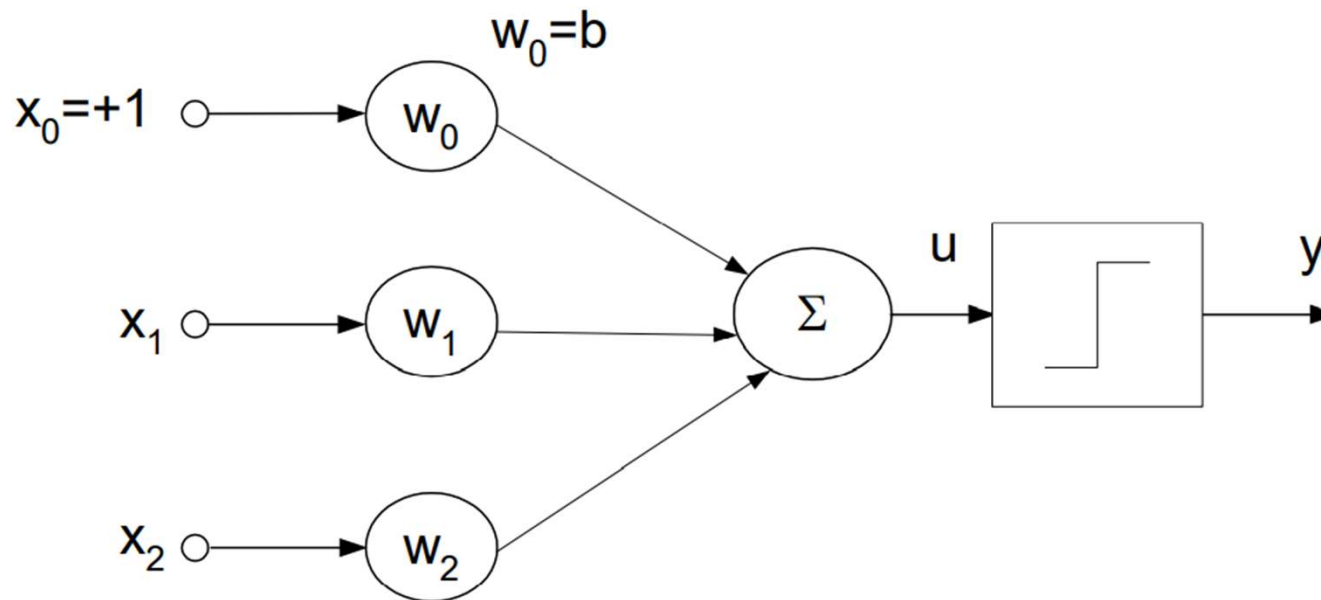
X1	X2	Pontos	
0,0	2,0	6,0	linha
1,0	1,0	5,0	abaixo
1,0	2,0	8,0	acima
2,0	0,0	4,0	abaixo
2,0	0,7	6,0	linha
2,0	1,0	7,0	acima



# Redes Neurais Artificiais

## RNA – Separação linear

- Com os parâmetros  $w_0$ ,  $w_1$  e  $w_2$ , a função  $f(u)$  separa o espaço de entradas em duas regiões, usando uma linha reta dada por:  $w_1 * x_1 + w_2 * x_2 + w_0 = 0$



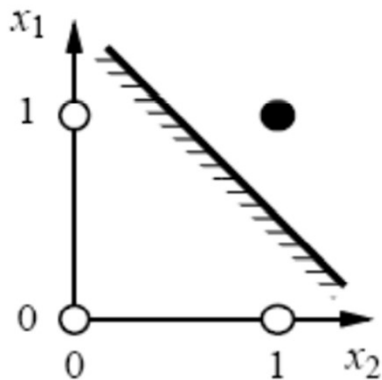
$$y = f(w_1x_1 + w_2x_2 + w_0), \text{ sendo } \begin{cases} f(u) = 1 & \text{se } u \geq 0 \\ f(u) = 0 & \text{se } u < 0 \end{cases}$$

# Redes Neurais Artificiais

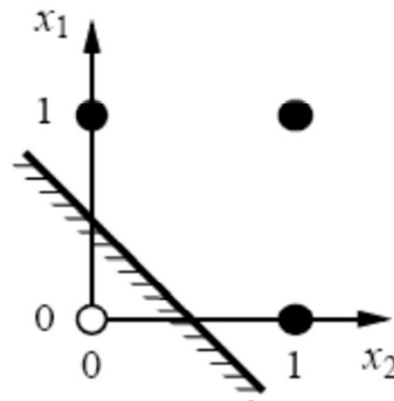
---

## RNA – *Perceptron* simples

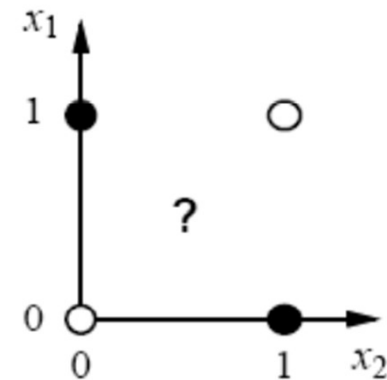
- O *Perceptron* simples é um separador linear que traça um plano entre os pontos de entrada onde a saída é zero ou um.



(a)  $x_1$  and  $x_2$



(b)  $x_1$  or  $x_2$



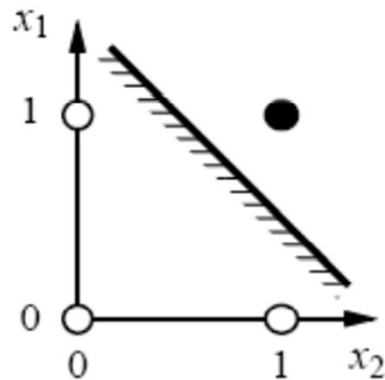
(c)  $x_1$  xor  $x_2$



# Redes Neurais Artificiais

## RNA – Treinamento de neurônio (Exemplo)

- Iniciamos o neurônio com os pesos 0.5 e 0.6 para as duas entradas, e -0.3 para o limiar ( $w_0$ ). Isso é equivalente à equação:  $u = 0.5 * x_1 + 0.6 * x_2 - 0.3$
- $x_0$  é a entrada estável sempre igual a 1.
- Assim, a saída  $y$  deve disparar quando  $u \geq 0$ , para a função AND.

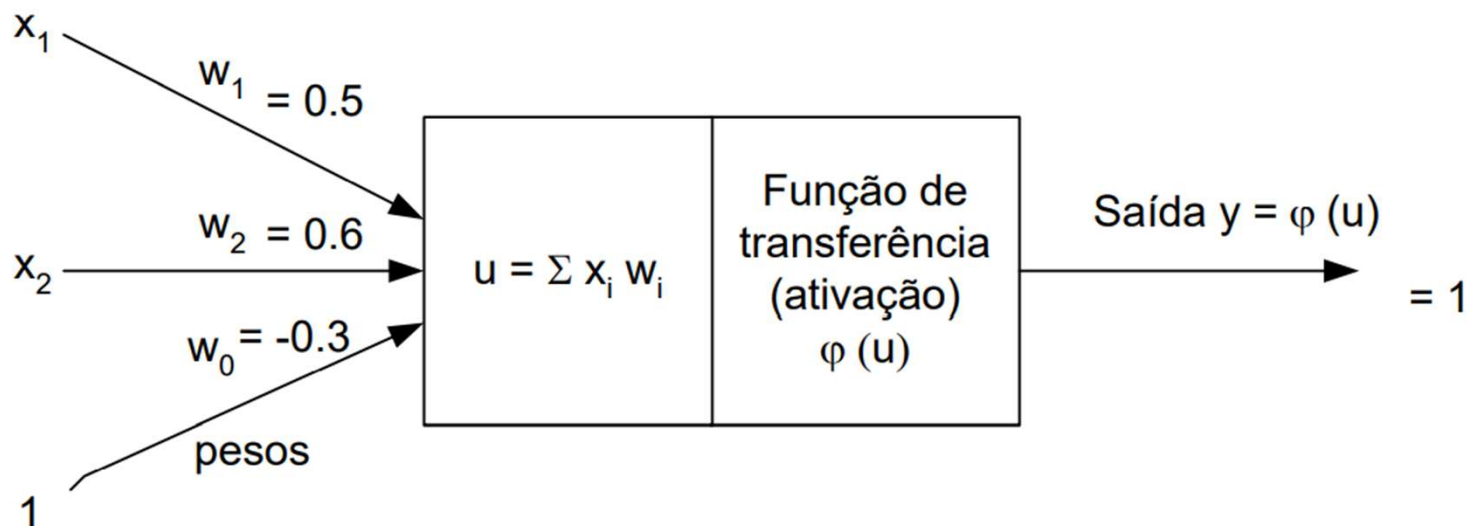


(a)  $x_1$  and  $x_2$

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

# Redes Neurais Artificiais

## RNA – Treinamento de neurônio (Exemplo)

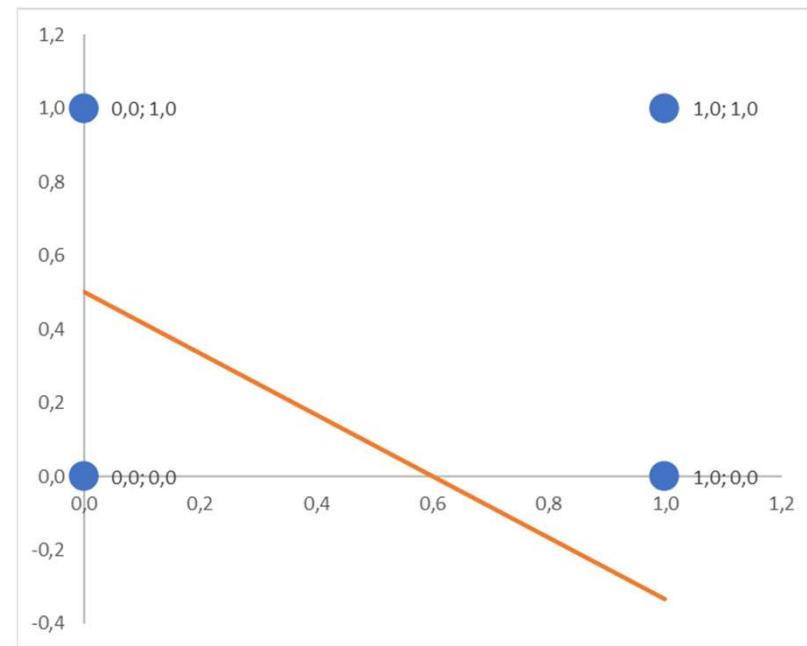


	X1	X2	W1	W2	W3	U	Y	DESEJADO
1a	0	0	0,5	0,6	-0,3	-0,3	0	0
2a	0	1	0,5	0,6	-0,3	0,3	1	0
3a	1	0	0,5	0,6	-0,3	0,2	1	0
4a	1	1	0,5	0,6	-0,3	0,8	1	1

# Redes Neurais Artificiais

## RNA – Treinamento de neurônio (Exemplo)

			w1	w2	w0			
			0,5	0,6	-0,3			
	X1	X2	W1	W2	W3	U	Y	DESEJADO
1a	0	0	0,5	0,6	-0,3	-0,3	0	0
2a	0	1	0,5	0,6	-0,3	0,3	1	0
3a	1	0	0,5	0,6	-0,3	0,2	1	0
4a	1	1	0,5	0,6	-0,3	0,8	1	1



# Redes Neurais Artificiais

---

## RNA – Treinamento de neurônio (Exemplo)

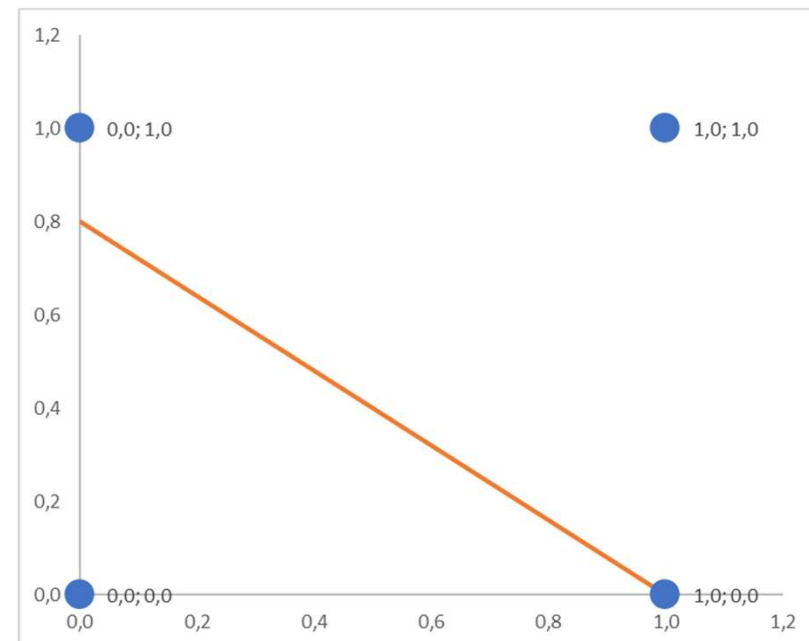
	X1	X2	W1	W2	W0	U	Y	DESEJADO	
1a	0	0	0,5	0,6	-0,3	-0,3	0	0	Ok
2a	0	1	0,5	0,6	-0,3	0,3	1	0	Not Ok
									Ajuste -0,1
3a	1	0	0,4	0,5	-0,4	0	1	0	Not Ok
									Ajuste -0,1
4a	1	1	0,3	0,4	-0,5	0,2	1	1	Ok
									Nova rodada
5a	0	0	0,3	0,4	-0,5	-0,5	0	0	Ok
6a	0	1	0,3	0,4	-0,5	-0,1	0	0	Ok
7a	1	0	0,3	0,4	-0,5	-0,2	0	0	Ok
8a	1	1	0,3	0,4	-0,5	0,2	1	1	Ok

# Redes Neurais Artificiais

## RNA – Treinamento de neurônio (Exemplo)

	X1	X2	u	y	Desejado
1a	0	0	-0,4	0	0
2a	0	1	0,1	1	0
3a	1	0	0	1	0
4a	1	1	0,5	1	1

w1	w2	w0
0,4	0,5	-0,4

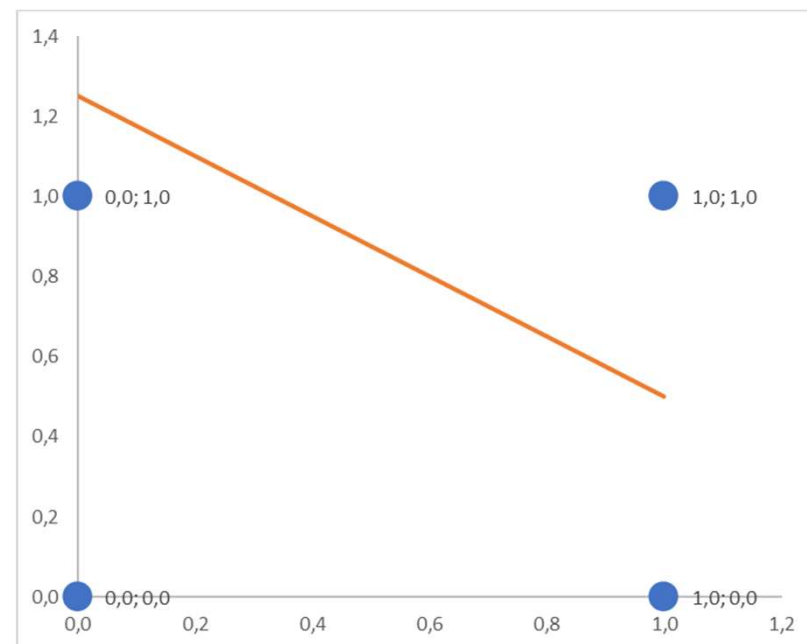


# Redes Neurais Artificiais

## RNA – Treinamento de neurônio (Exemplo)

	X1	X2	u	y	Desejado
1a	0	0	-0,5	0	0
2a	0	1	-0,1	0	0
3a	1	0	-0,2	0	0
4a	1	1	0,2	1	1

w1	w2	w0
0,3	0,4	-0,5



# Redes Neurais Artificiais

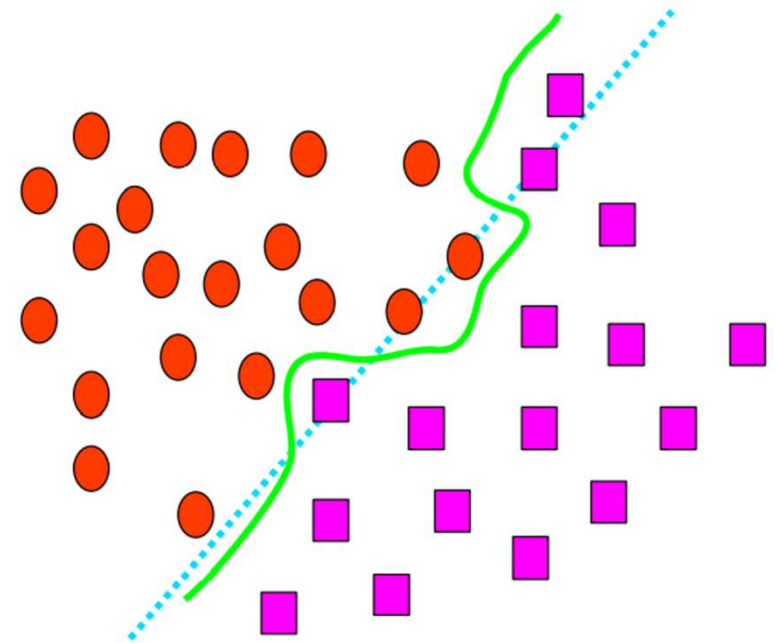
---

## RNA – Algoritmo de treinamento *Perceptron*

- Para classificação padrões de entrada como pertencentes ou não a uma dada classe, considere o conjunto de treinamento formado por  $N$  amostras  $\{x_1, d_1\}, \{x_2, d_2\}, \dots, \{x_N, d_N\}$ , onde  $x_j$  é o vetor de entradas e  $d_j$  a saída desejada (classe), que em notação vetorial tem-se  $\{X, d\}$ , onde:

$$X \in \mathbb{R}^{mxN}$$

$$d \in \mathbb{R}^{1 \times N}$$



# Redes Neurais Artificiais

---

## RNA – Algoritmo de treinamento *Perceptron*

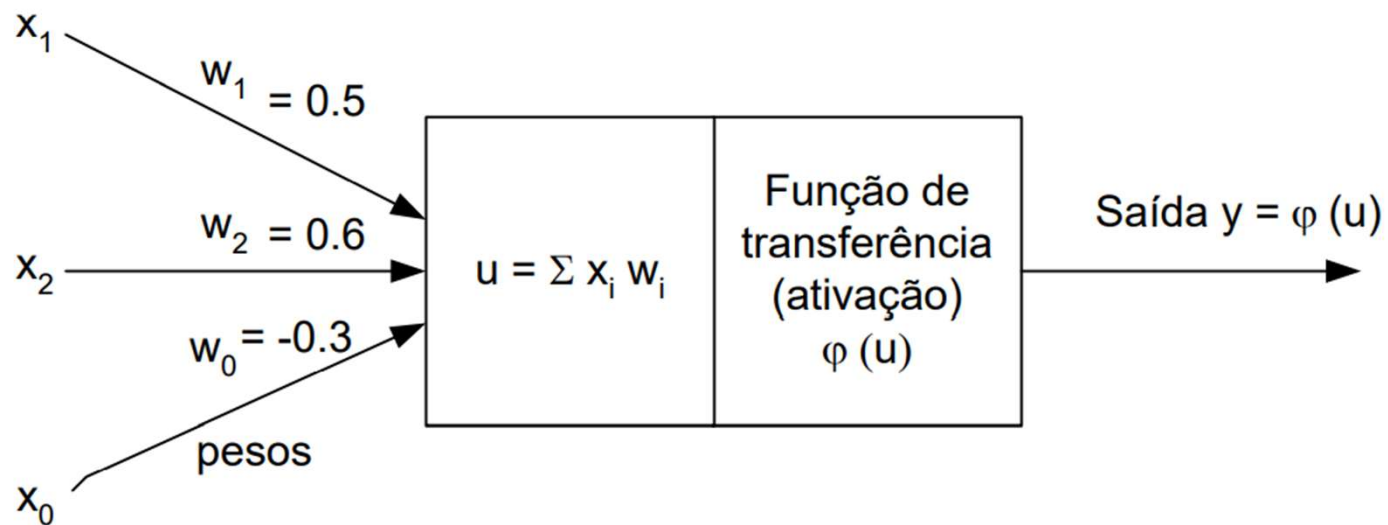
- Executa os exemplos de treinamento através da rede.
- Ajusta os pesos depois de cada exemplo para reduzir o erro.
- Cada ciclo através dos exemplos é chamado de época.
- As épocas são repetidas até que se alcance algum critério de parada.
- Em geral, quando as mudanças nos pesos forem pequenas.



# Redes Neurais Artificiais

---

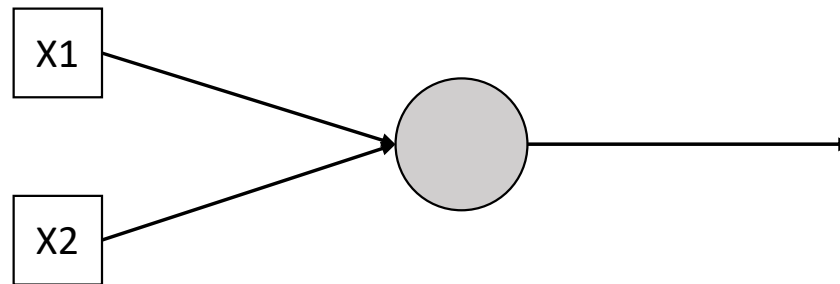
## RNA – Arquitetura



# Redes Neurais Artificiais

---

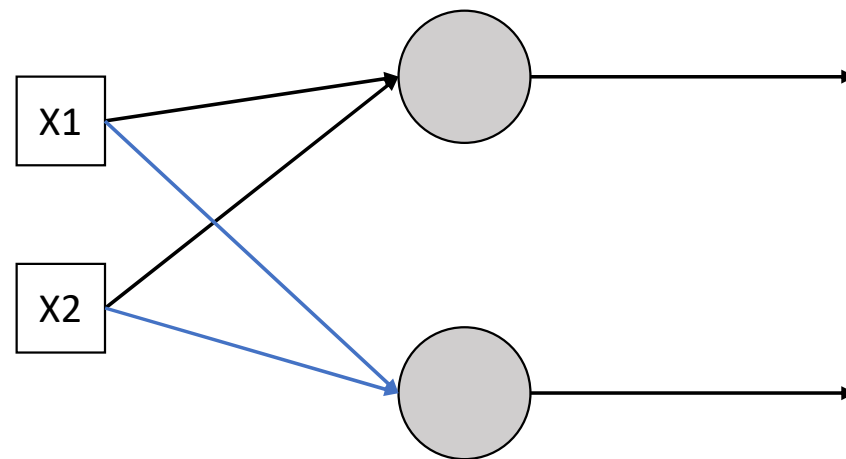
## RNA – Arquitetura



# Redes Neurais Artificiais

---

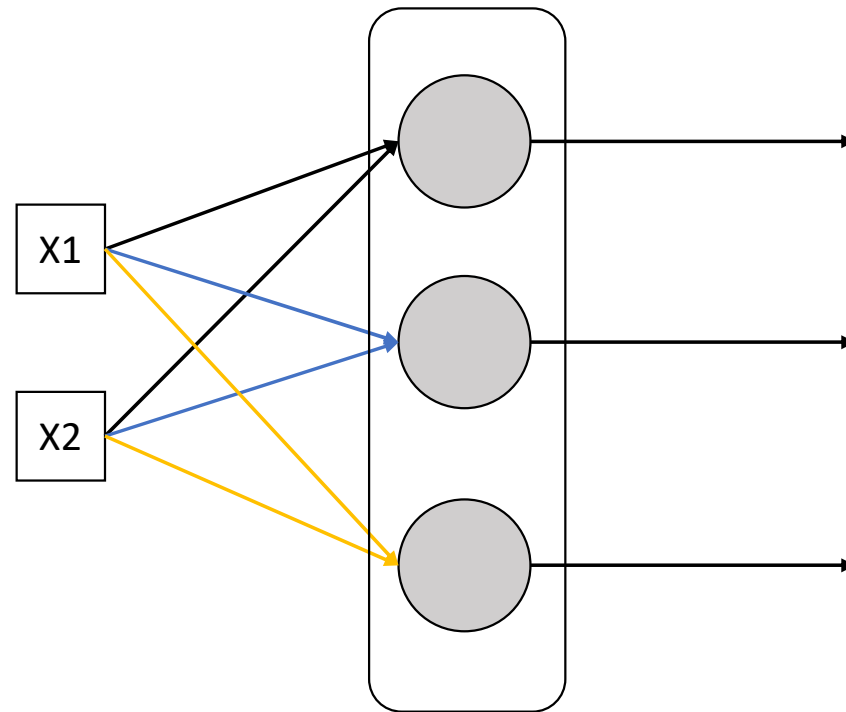
## RNA – Arquitetura



# Redes Neurais Artificiais

---

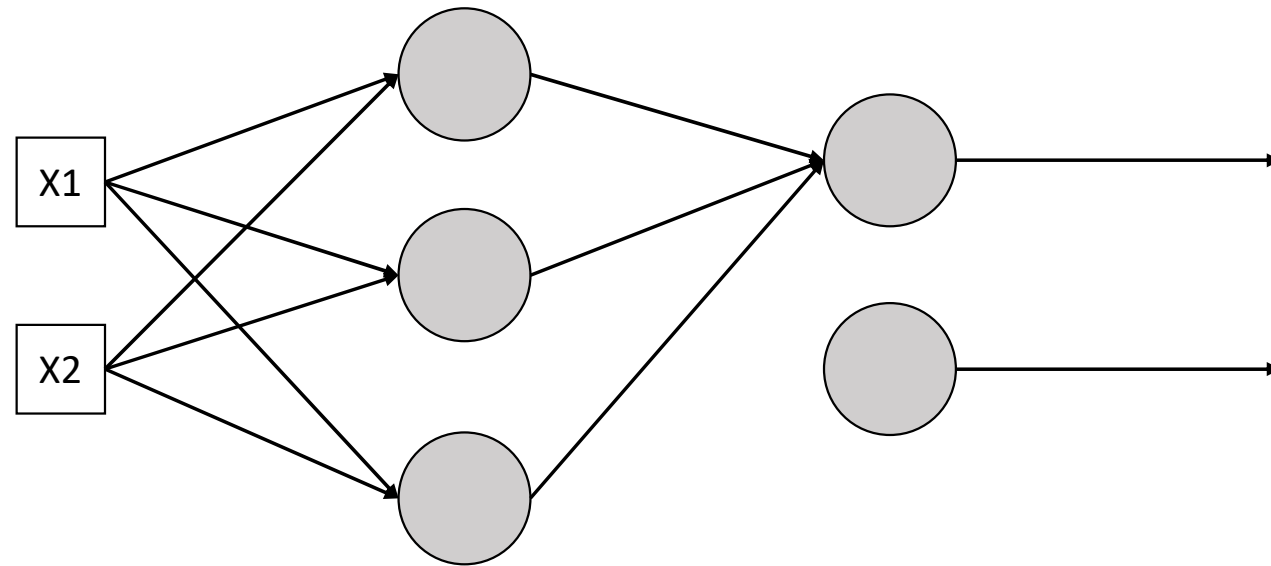
## RNA – Arquitetura



# Redes Neurais Artificiais

---

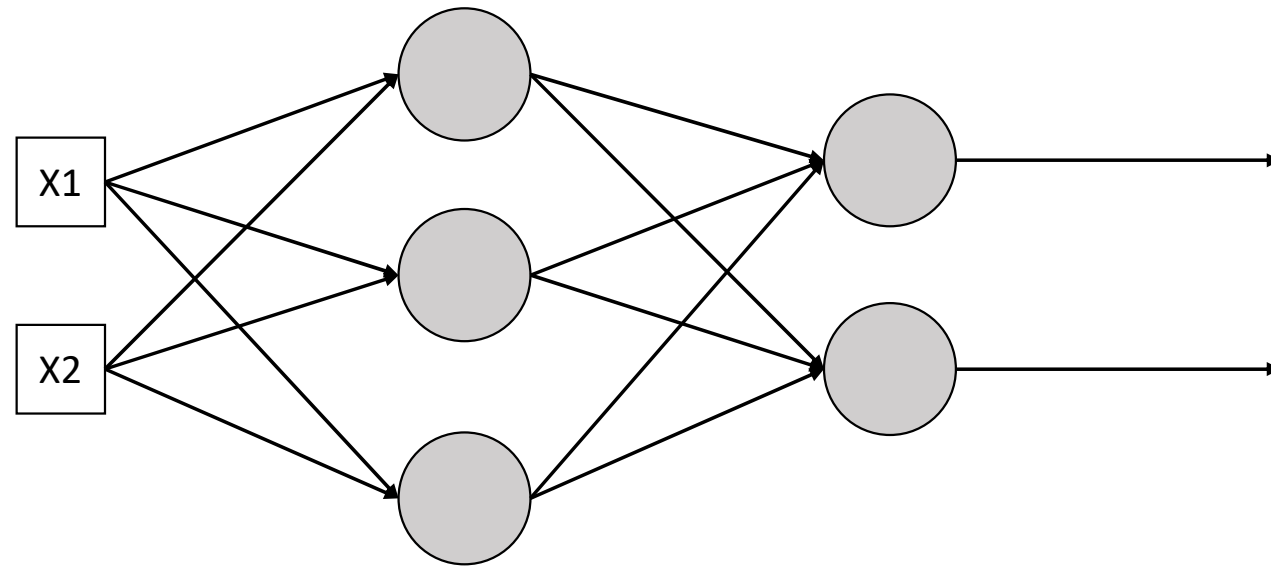
## RNA – Arquitetura



# Redes Neurais Artificiais

---

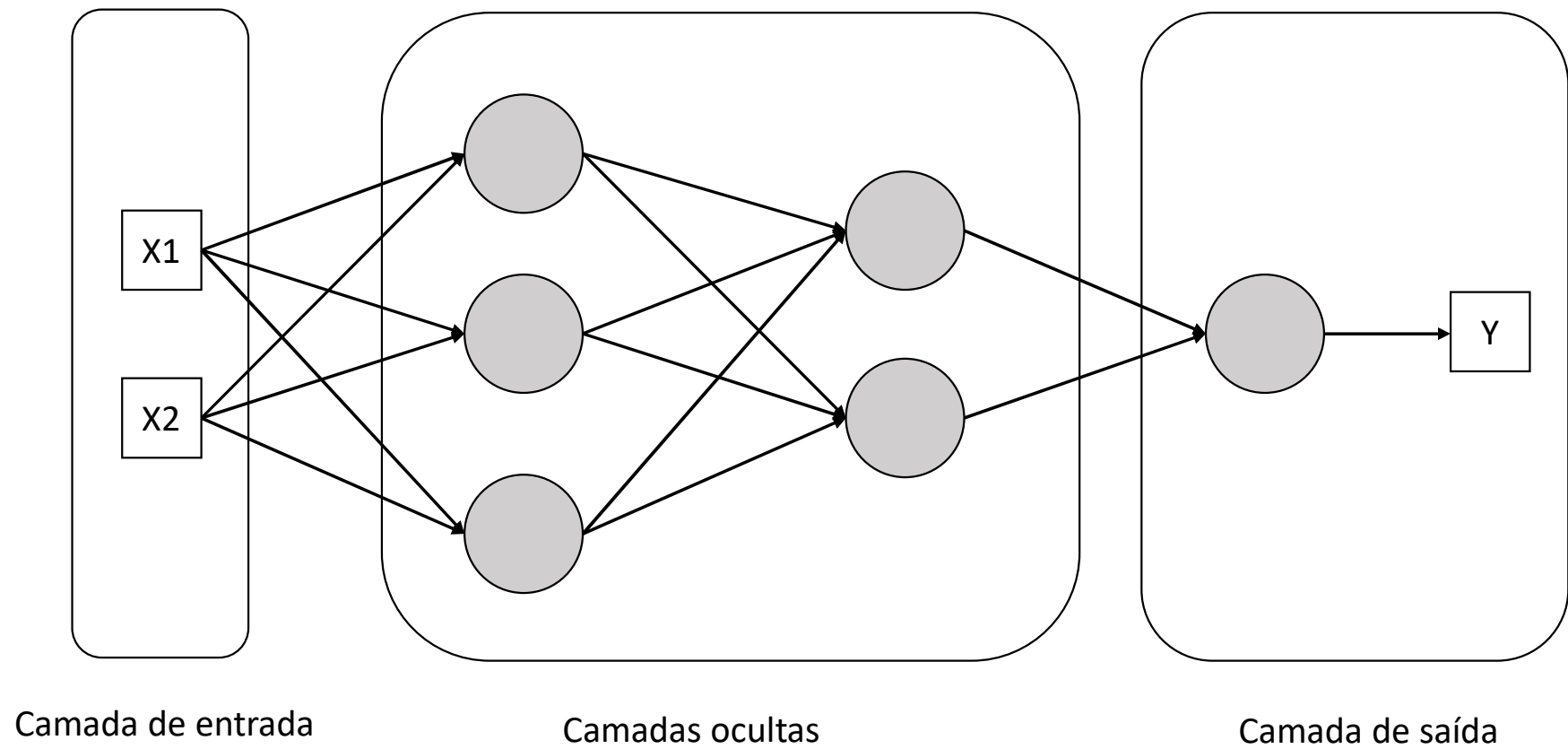
## RNA – Arquitetura



# Redes Neurais Artificiais

---

## RNA – Arquitetura



# Redes Neurais Artificiais

---

## RNA – Arquitetura

- Uma rede neural artificial é composta por várias unidades de processamento (neurônios), cujo funcionamento é bastante simples.
- Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso.
- As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões.
- O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.



# Redes Neurais Artificiais

---

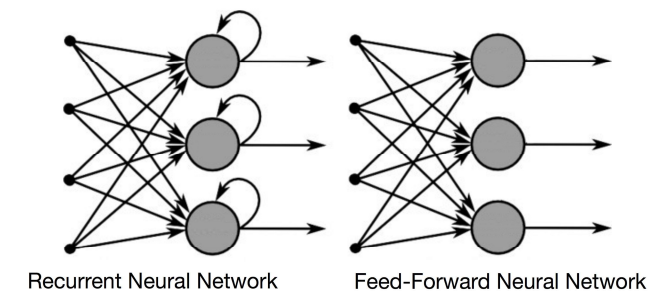
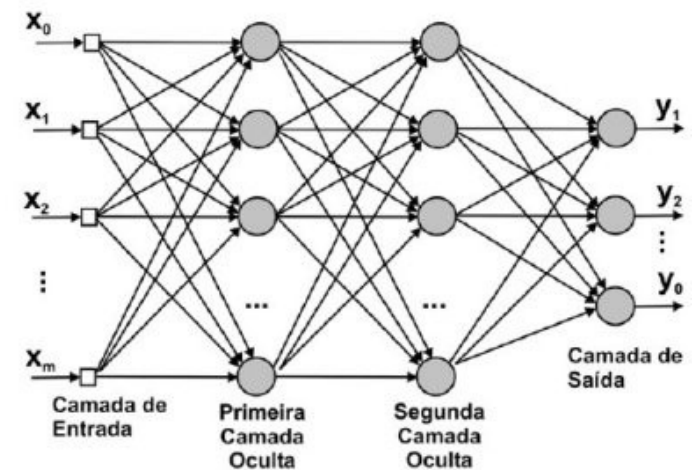
## RNA – Características gerais

- São modelos adaptativos treináveis.
- Podem representar domínios complexos (não lineares).
- São capazes de generalização diante de informação incompleta.
- Robustos.
- São capazes de fazer armazenamento associativo de informações.
- Processam informações Espaço/temporais.
- Possuem grande paralelismo, o que lhe conferem rapidez de processamento.

# Redes Neurais Artificiais

## RNA – Tipos

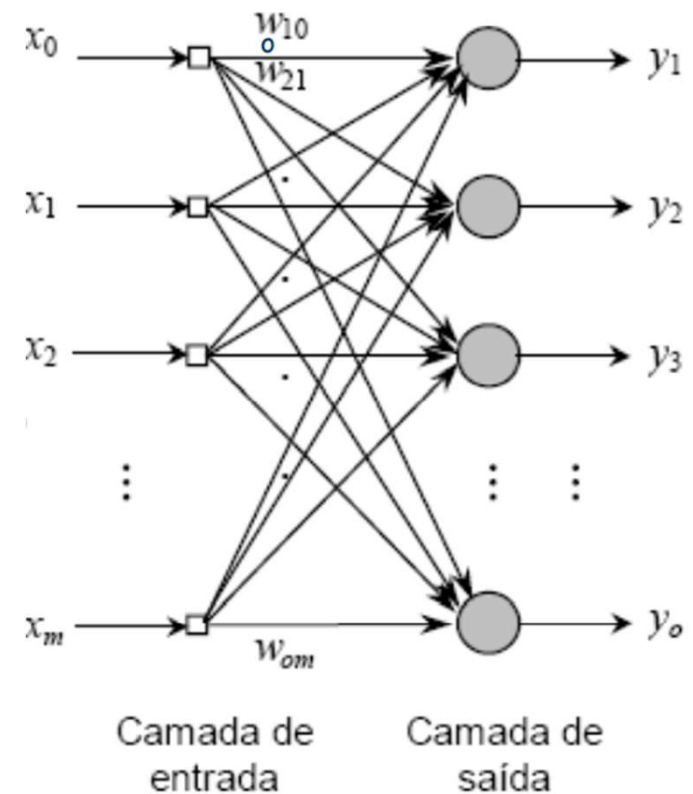
- Existem basicamente 3 tipos básicos de arquitetura de RNAs:
  - *Feedforward* de uma única camada
  - *Feedforward* de múltiplas camadas
  - Redes recorrentes.
- Classificação das camadas:
  - Entrada: padrões são apresentados à rede.
  - Intermediárias ou Ocultas: maior parte do processamento, através das conexões ponderadas; extratoras de características.
  - Saída: resultado final é concluído e apresentado.



# Redes Neurais Artificiais

## RNA – *Feedforward* única camada

- Os neurônios da camada de entrada correspondem aos neurônios sensoriais que possibilitam a entrada de sinais na rede (não fazem processamento).
- Os neurônios da camada de saída fazem processamento.

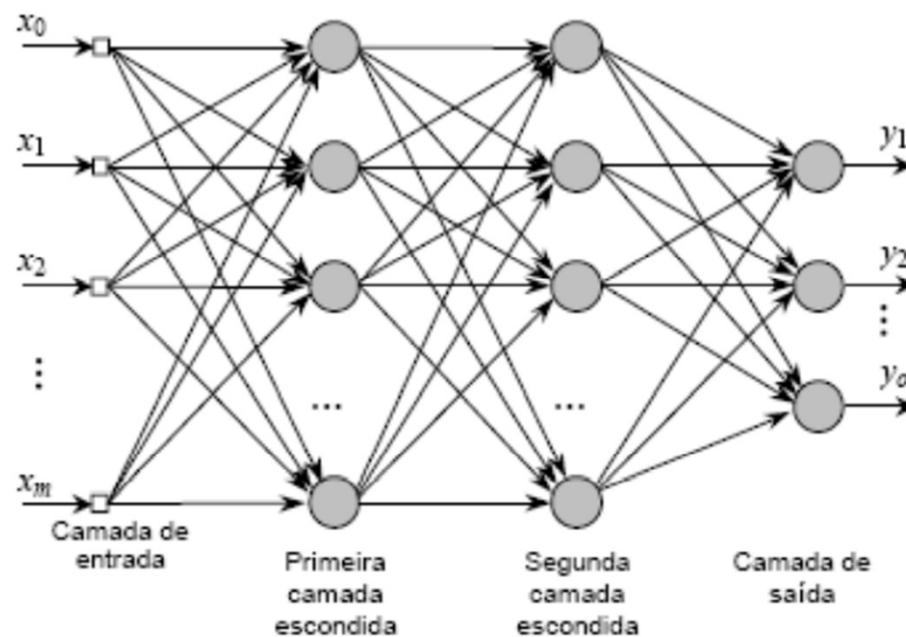


# Redes Neurais Artificiais

---

RNA – *Feedforward* múltiplas camadas – *Multi Layer Perceptron* (MLP)

- A rede possui múltiplas camadas.

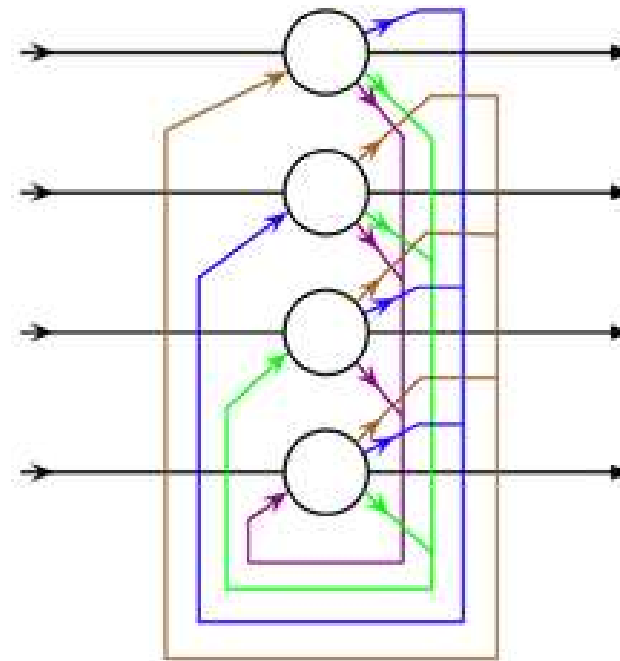


# Redes Neurais Artificiais

---

## RNA – Redes recorrentes

- Possuem pelo menos uma interconexão realimentando a saída de neurônios para outros neurônios da rede (conexão cíclica).



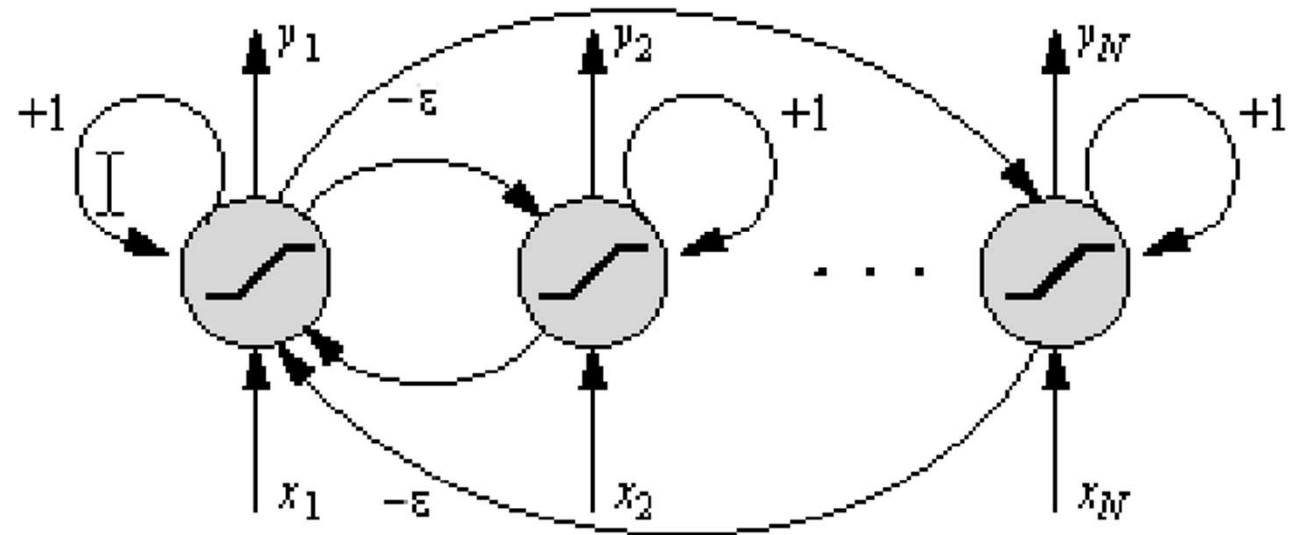
Exemplo: Rede de Hopfield com quatro nós

# Redes Neurais Artificiais

---

## RNA – Redes de Kohonen

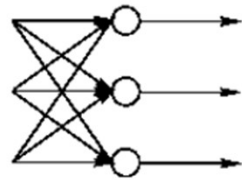
- Mapa auto-organizável, *Self-organized map* (SOM), capaz de diminuir a dimensão de um grupo de dados, conseguindo manter a representação real com relação as propriedades relevantes dos vetores de entrada, tendo-se como resultado um conjunto das características do espaço de entrada.



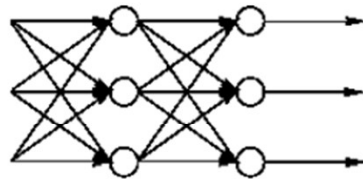
# Redes Neurais Artificiais

---

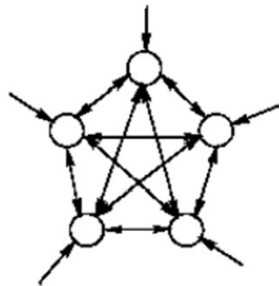
## RNA – Diversas arquiteturas



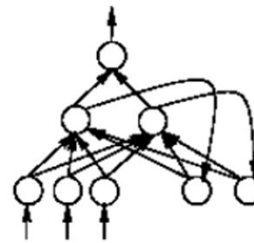
a) Single-Layer Perceptron



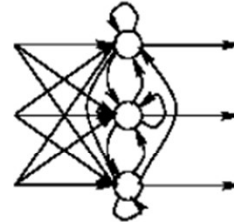
b) Multi-Layer Perceptron



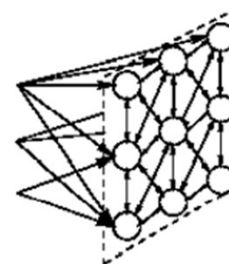
c) Hopfield network



d) Elman recurrent network



e) Competitive networks



f) Self-Organizing Maps

# Redes Neurais Artificiais

---

## Treinamento

- A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.
- Algoritmo de aprendizado é um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais. Eles diferem entre si principalmente pelo modo como os pesos são modificados.



# Redes Neurais Artificiais

---

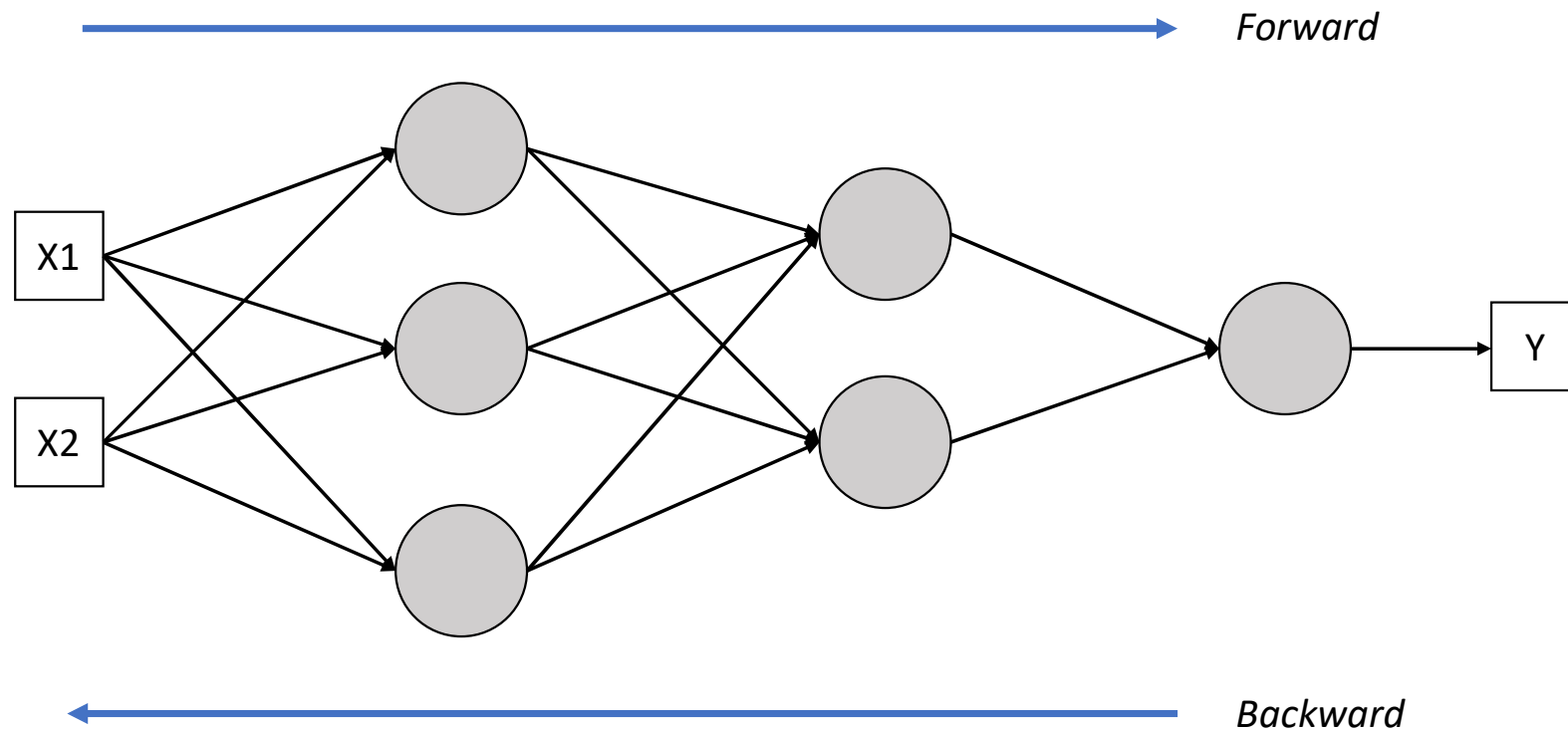
## Treinamento da MLP

- Algoritmo de aprendizagem – *backpropagation*
  - Regra de aprendizagem baseada na correção do erro pelo método do Gradiente.
  - O algoritmo de aprendizagem é composto de duas fases:
    - ✓ – Cálculo do erro (*forward*)
    - ✓ – Correção dos pesos sinápticos (*backward*)

# Redes Neurais Artificiais

---

Algoritmo de retro propagação (*backpropagation*)



# Redes Neurais Artificiais

---

Algoritmo de retro propagação (*backpropagation*)

- Desvantagens do algoritmo de aprendizagem *backpropagation*:
  - Normalmente o tempo de processamento é elevado
  - A arquitetura da rede deve ser fixada a priori
  - Precisa haver convergência

# Redes Neurais Artificiais

---

## Aspectos do treinamento de redes MLP

- O aprendizado é resultado de apresentação repetitiva de todas as amostras do conjunto de treinamento.
- Cada apresentação de todo o conjunto de treinamento é denominada época.
- O processo de aprendizagem é repetido época após época, até que um critério de parada seja satisfeito.
- É recomendável que a ordem de apresentação das amostras seja aleatória de uma época para outra. Isso tende a fazer com que o ajuste de pesos tenha um caráter estocástico ao longo do treinamento.

# Redes Neurais Artificiais

---

## Atualização local ou por lote

- Local: a atualização é feita imediatamente após a apresentação de cada amostra de treinamento.
  - É também chamado de método de atualização on-line ou padrão a padrão.
- Lote: a atualização dos pesos só é feita após a apresentação de todas as amostras de treinamento que constituem uma época.
  - É também conhecido como método de atualização off-line ou batch.
  - O ajuste relativo a cada apresentação de uma amostra é acumulado.
  - Fornece uma melhor estimativa do vetor gradiente.
  - Requer um menor armazenamento para cada conexão, e apresenta menos possibilidade de convergência para um mínimo local.

# Redes Neurais Artificiais

---

## Critérios de parada

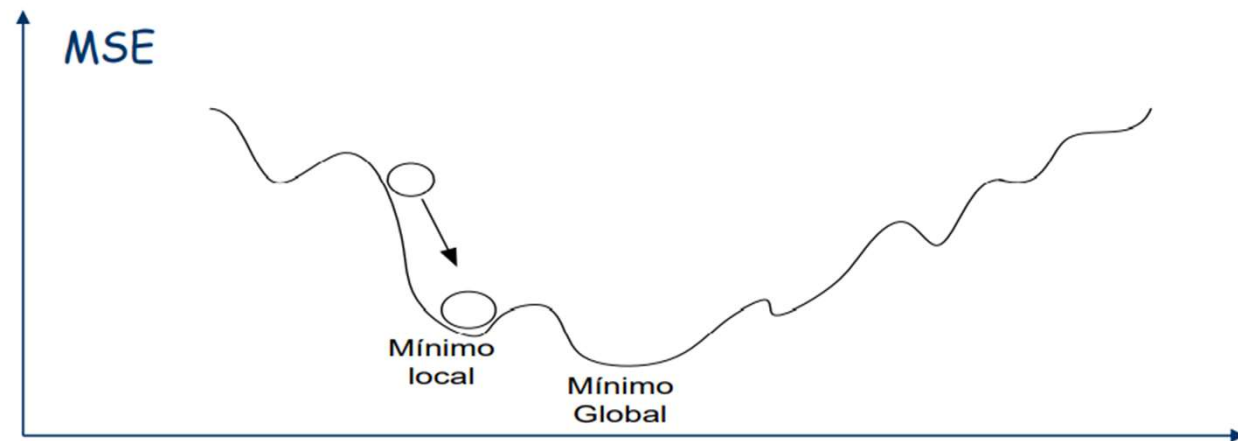
- O processo de minimização do MSE (função custo) não apresenta convergência garantida e não possui um critério de parada bem definido.
- Um critério de parada não muito recomendável, que não leva em conta o estado do processo iterativo é o da pré definição do número total de iterações.

# Redes Neurais Artificiais

---

## Cr terios de parada

- Nota-se que se o crit rio   de valor m nimo de MSE  nt o n o se garante que o algoritmo ir  atingir esse valor.
- Por outro lado, se o crit rio   o m nimo valor do vetor gradiente deve-se considerar que o algoritmo termina no m nimo local mais pr ximo.

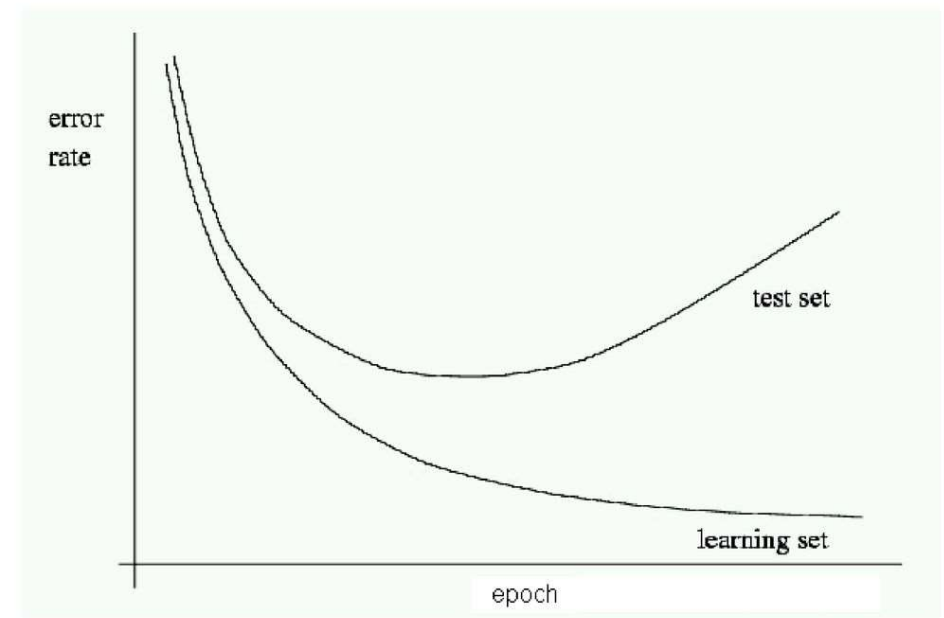


# Redes Neurais Artificiais

---

## Cr terios de parada

- Outro crit rio de parada que pode ser usado em conjunto com um dos crit rios anteriores   a avalia  o da capacidade de generaliza  o da rede ap s cada  poca de treinamento.
- O processo de treinamento   interrompido antes que a capacidade de generaliza  o da rede fique restrita.

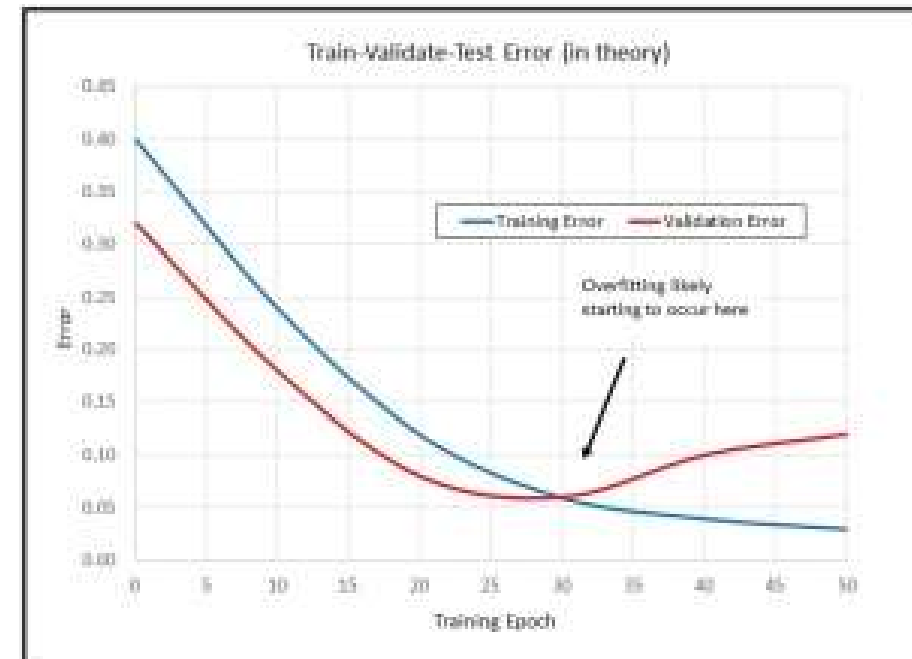




# Redes Neurais Artificiais

## Critérios de parada

- *Overfitting*
  - Depois de um certo ponto do treinamento, a rede piora ao invés de melhorar
  - Memoriza padrões de treinamento, incluindo todas as suas peculiaridades (ruído)
- Solução:
  - Encerrar treinamento cedo
  - Adoção das técnicas de *pruning* (eliminação de pesos e nodos irrelevantes)



# Redes Neurais Artificiais

---

## Definição da arquitetura da rede

- A quantidade de neurônios na camada de entrada e saída é dada pelo problema a ser abordado.
- No entanto, a quantidade de neurônios nas camadas de processamento são características do projeto.
- Aumentando-se o número de neurônios na camada escondida aumenta-se a capacidade de mapeamento não-linear da rede.
- No entanto, quando esse número for muito grande, o modelo pode se sobre ajustar aos dados, na presença de ruído nas amostras de treinamento. Diz-se que a rede está sujeito ao sobre treinamento (*overfitting*).

# Redes Neurais Artificiais

---

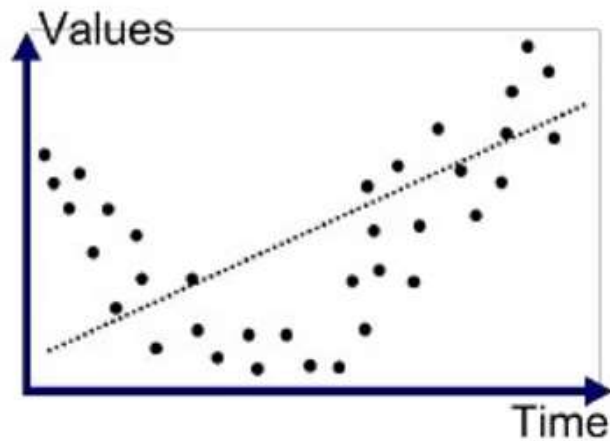
## Definição da arquitetura da rede

- Por outro lado, uma rede com poucos neurônios na camada escondida pode não ser capaz de realizar o mapeamento desejado, o que é denominado de *underfitting*.
- O *underfitting* também pode ser causado quando o treinamento é interrompido de forma prematura.

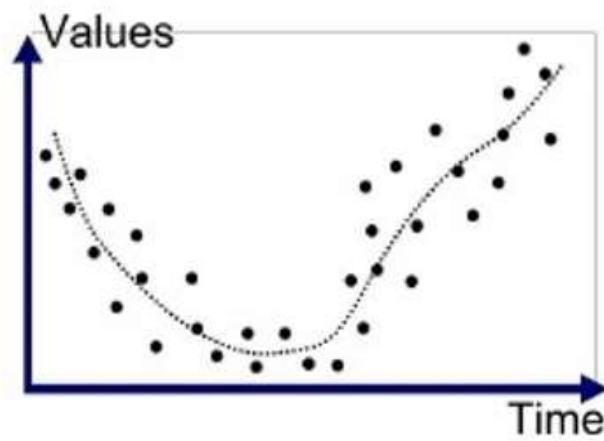
# Redes Neurais Artificiais

---

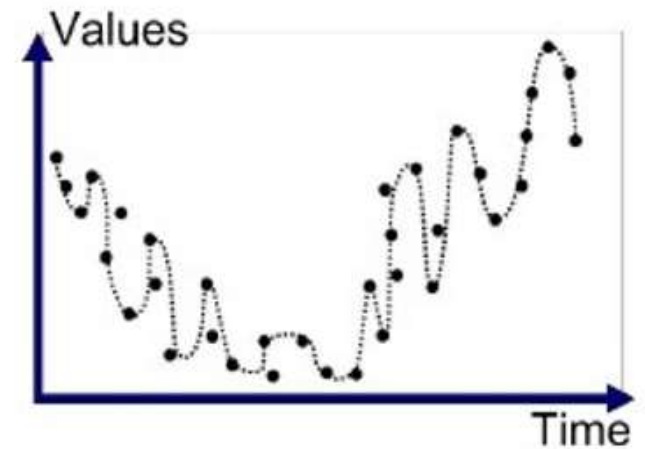
## *Overfitting vs Underfitting*



Underfitted



Good Fit/Robust

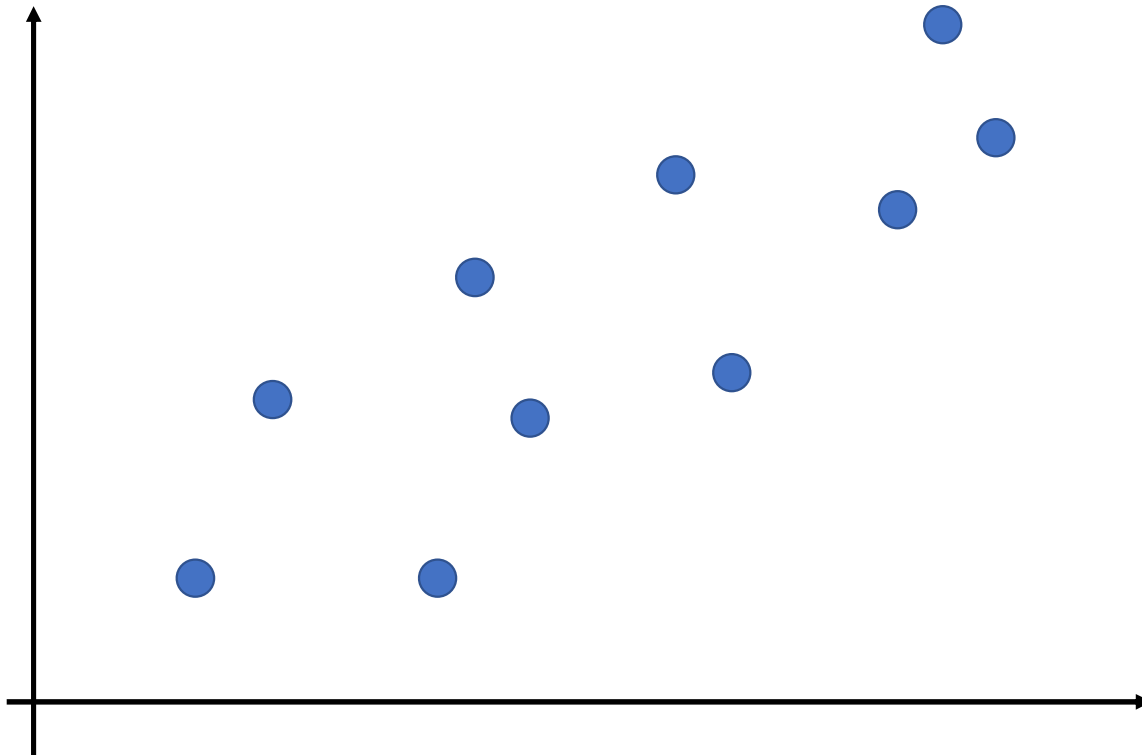


Overfitted

# Redes Neurais Artificiais

---

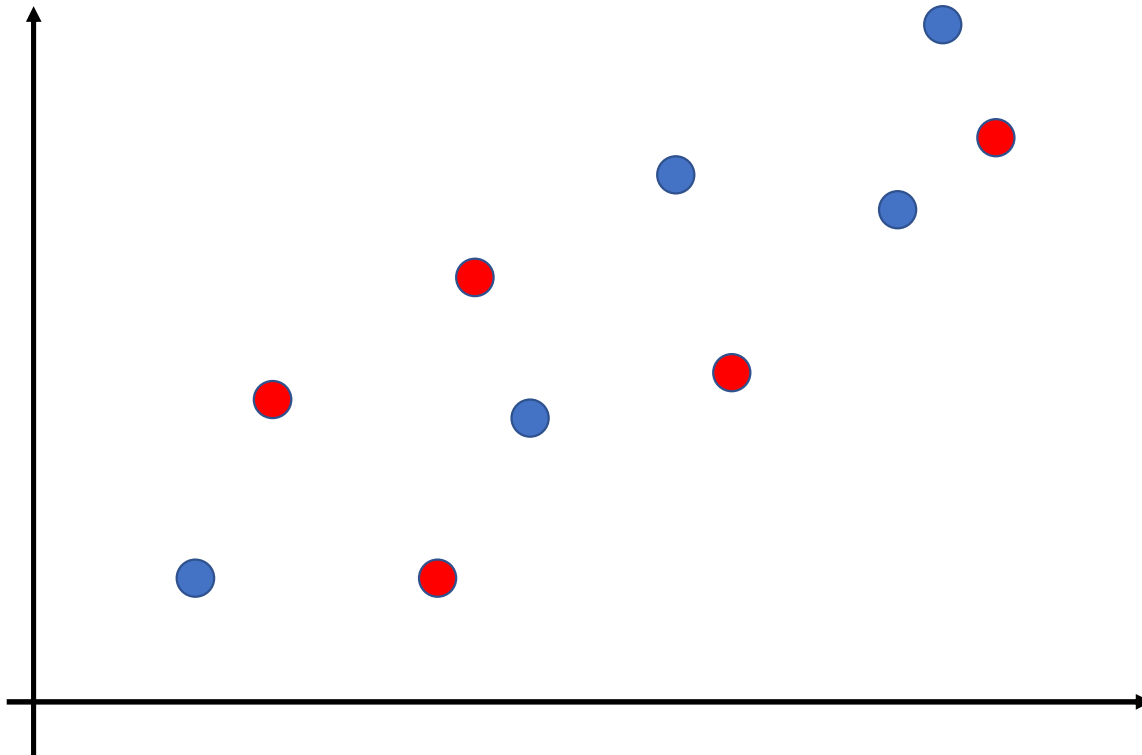
*Overfitting*



# Redes Neurais Artificiais

---

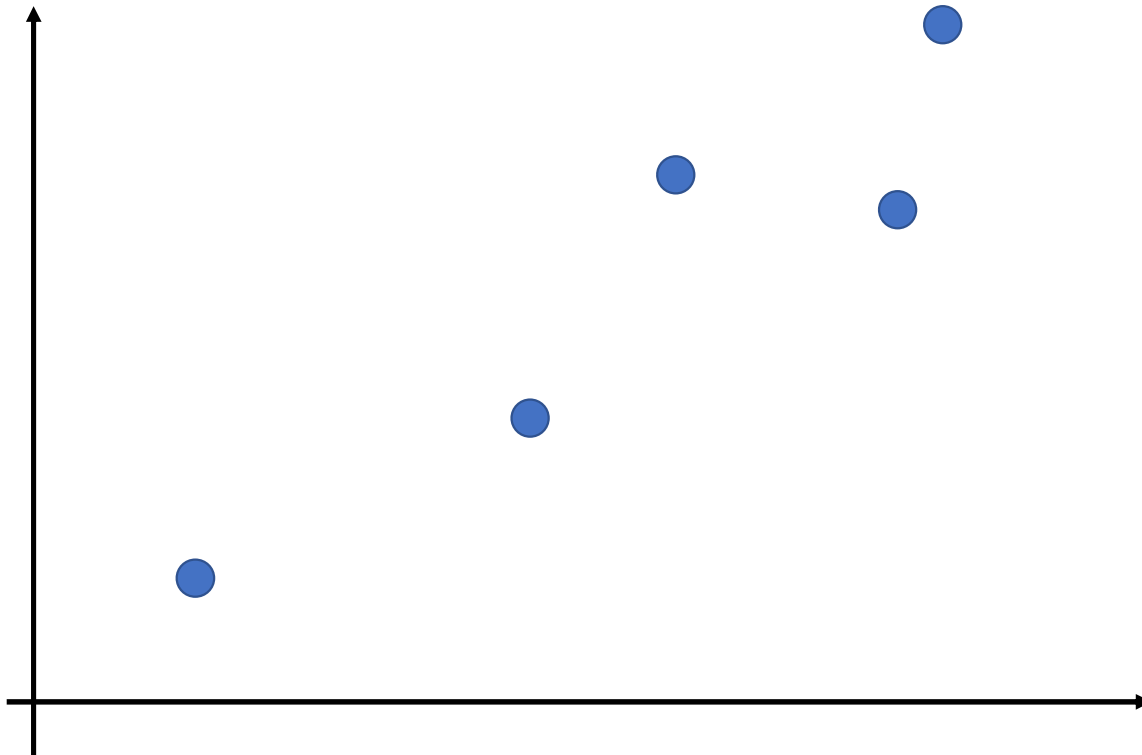
*Overfitting*



# Redes Neurais Artificiais

---

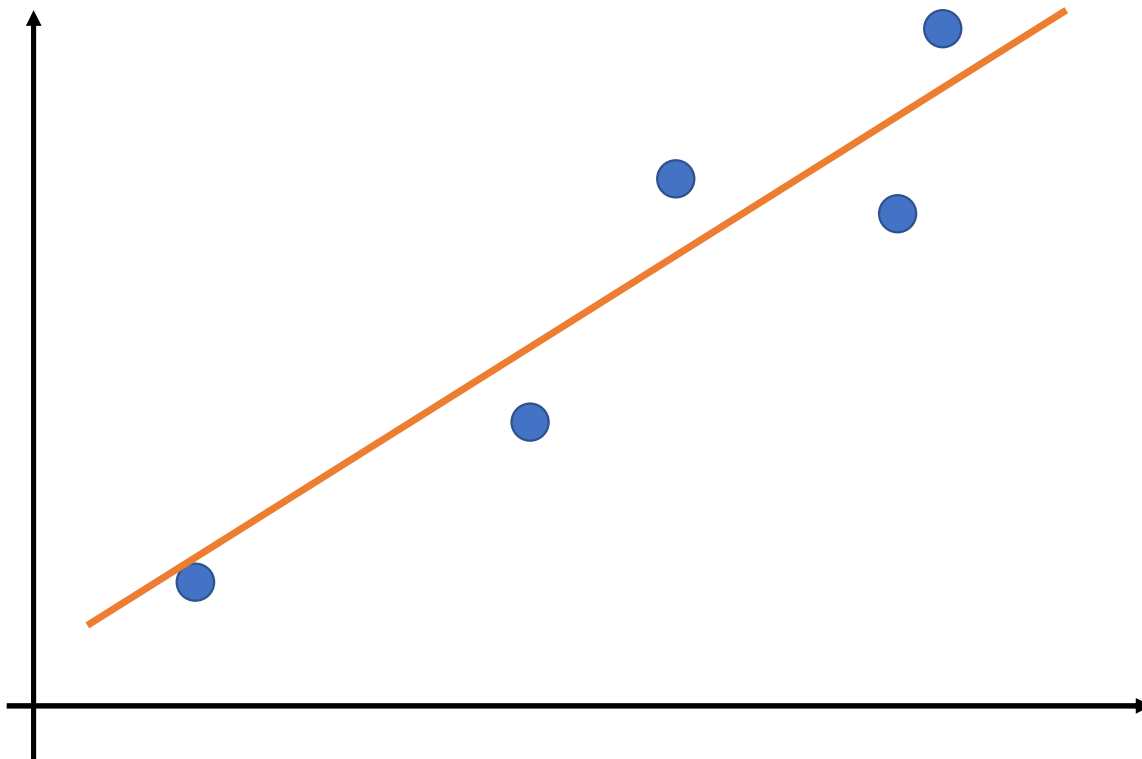
*Overfitting*



# Redes Neurais Artificiais

---

*Overfitting*

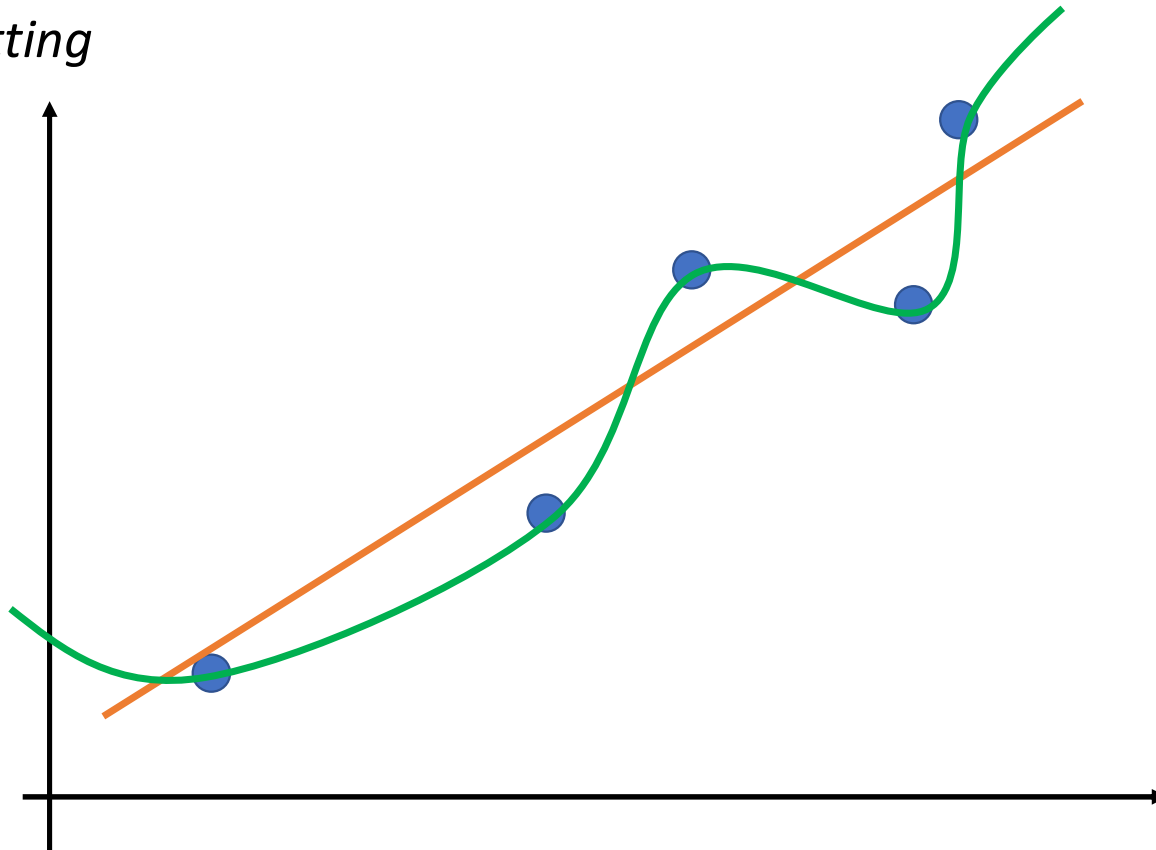




# Redes Neurais Artificiais

---

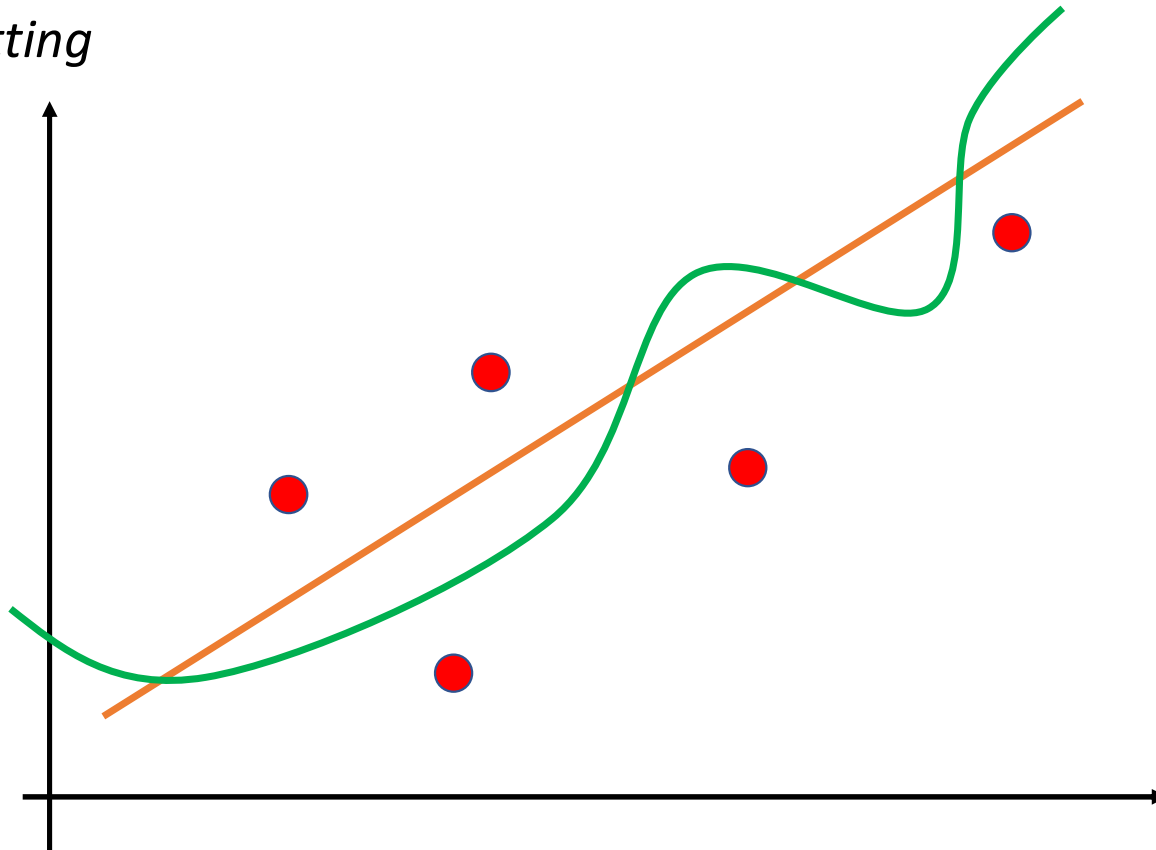
*Overfitting*



# Redes Neurais Artificiais

---

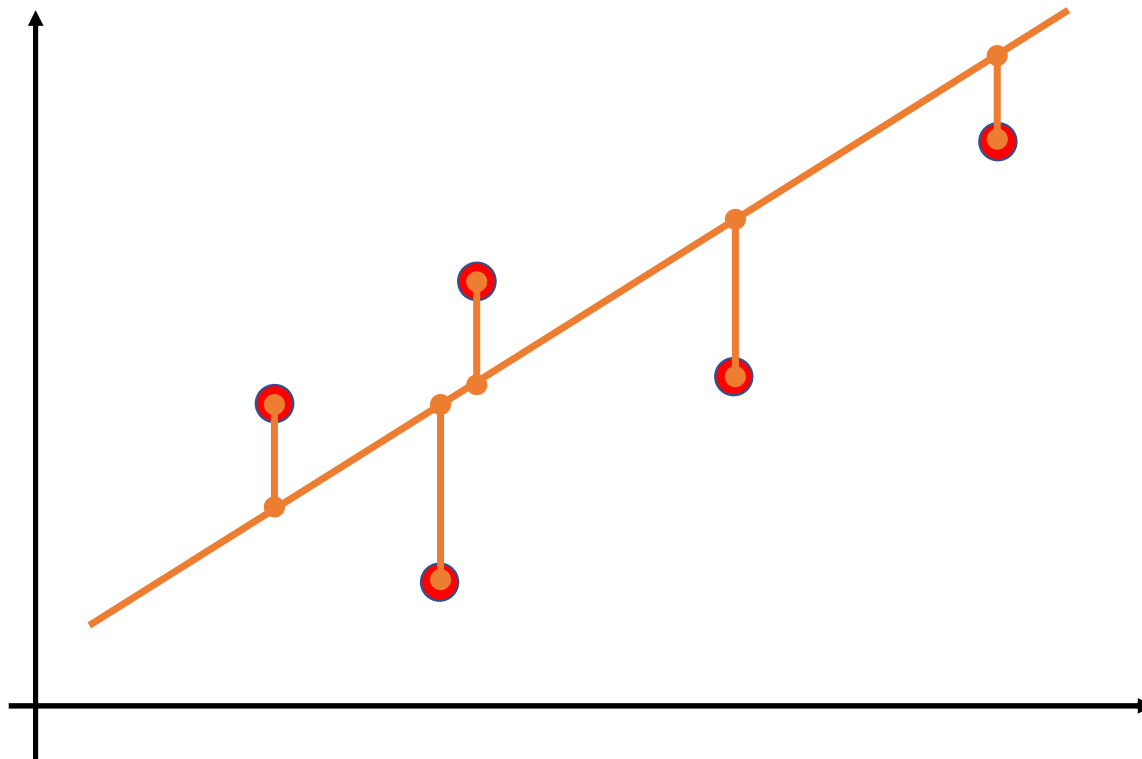
*Overfitting*



# Redes Neurais Artificiais

---

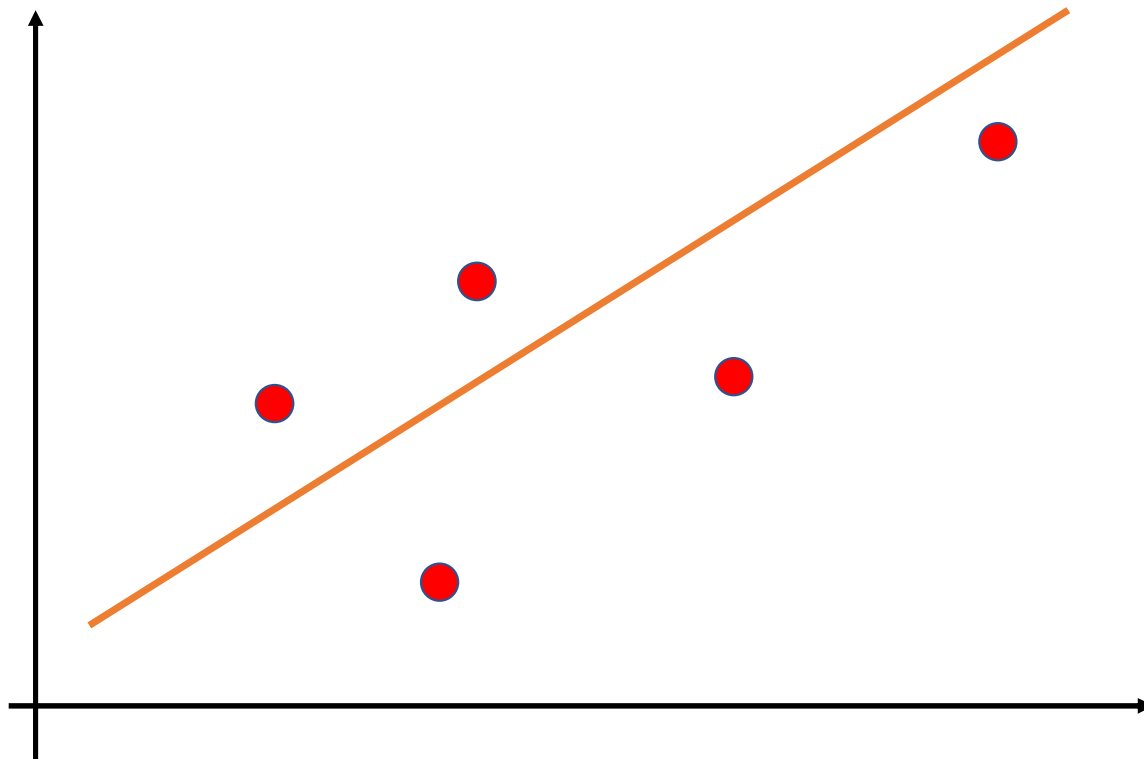
*Overfitting*



# Redes Neurais Artificiais

---

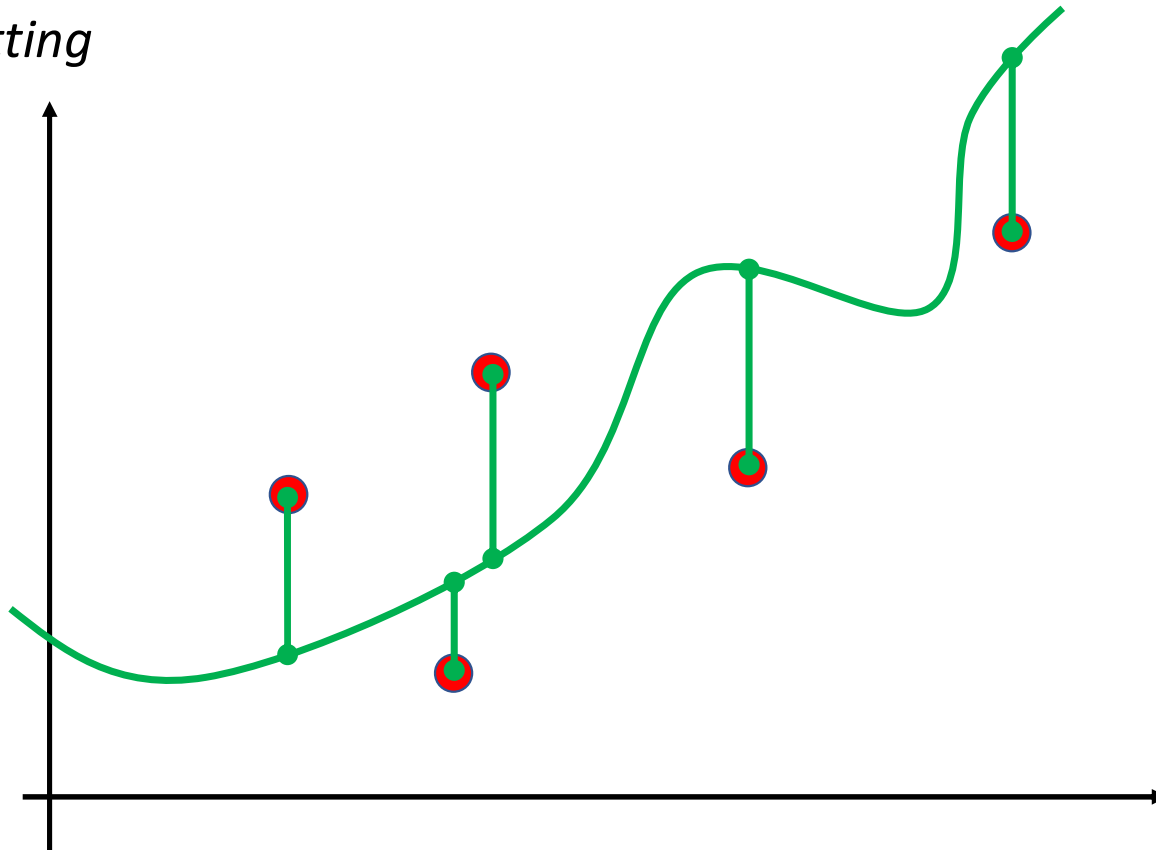
*Overfitting*



# Redes Neurais Artificiais

---

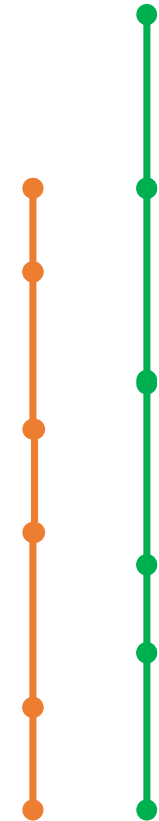
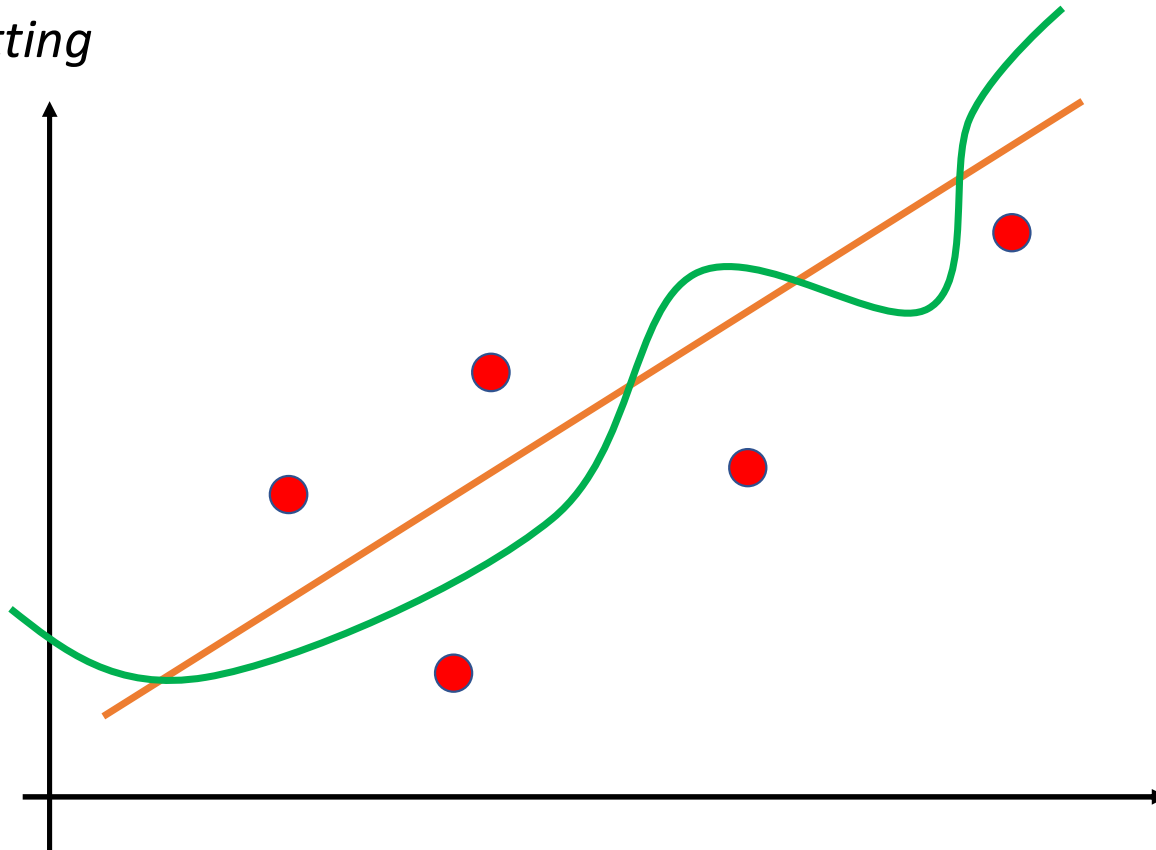
*Overfitting*



# Redes Neurais Artificiais

---

*Overfitting*

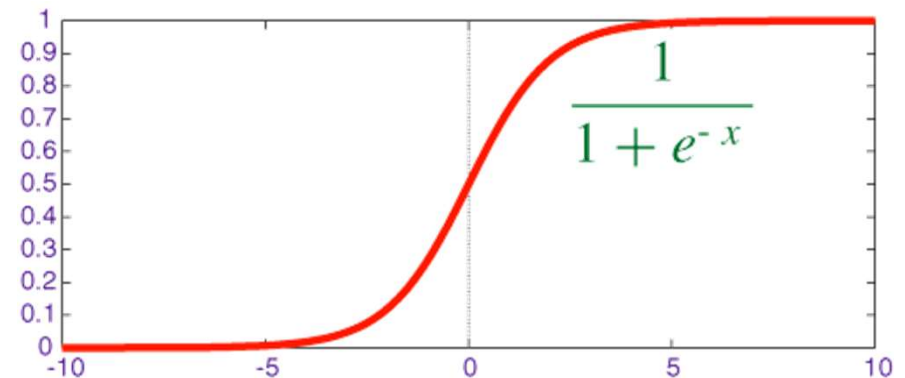


# Redes Neurais Artificiais

---

## Normalização dos dados de entrada

- Uma característica das funções sigmoidais é a saturação, ou seja, para valores grandes de argumento, a função opera numa região de saturação.
- É importante portanto trabalhar com valores de entrada que estejam contidos num intervalo que não atinjam a saturação, por exemplo:  $[0,1]$ .



# Redes Neurais Artificiais

---

## Inicialização dos vetores de pesos e bias

- A eficiência do aprendizado em redes multicamadas depende da:
  - Especificação de arquitetura da rede.
  - Função de ativação.
  - Regra de aprendizagem.
  - Valores iniciais dos vetores de pesos e bias.
- Considerando-se que os três primeiros itens já foram definidos, verifica-se agora a inicialização dos vetores de pesos e bias.



# Redes Neurais Artificiais

---

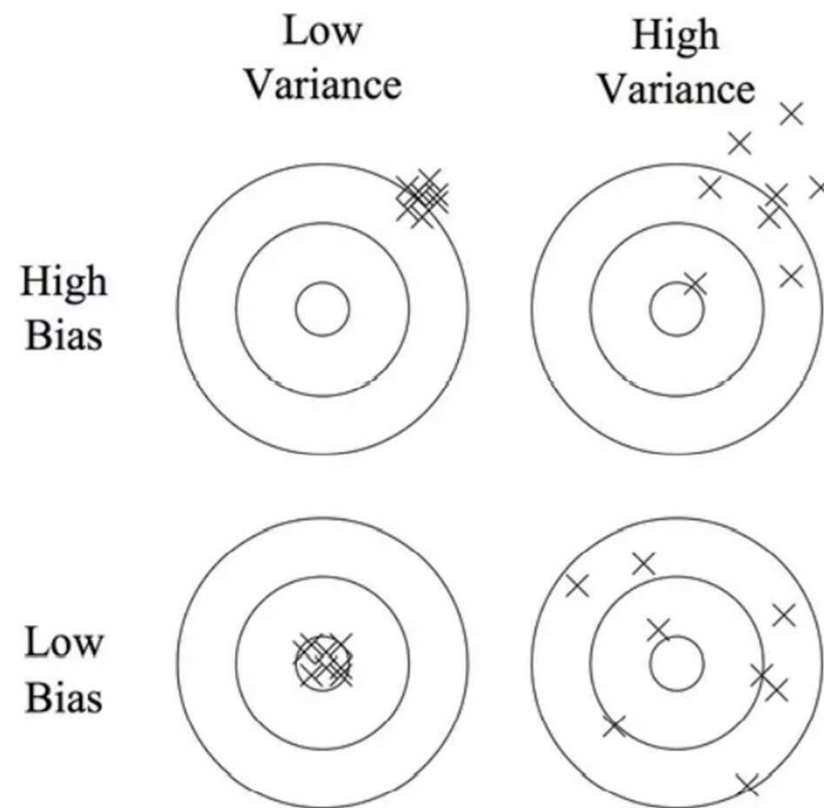
## Dilema Viés - Variância

- Com poucos parâmetros o desempenho no conjunto de treinamento (e de teste) é ruim, pois as superfícies de separação não são adequadamente colocadas.
- Com muitos parâmetros, há um ajuste exato (memorização) ao conjunto de treinamento, mas o resultado no conjunto de teste não é aceitável.
- A diferença entre o desempenho sobre os conjuntos de treinamento e teste é uma medida da variância do modelo.
- O objetivo da aprendizagem não deve ser o erro igual a zero no conjunto de treinamento.

# Redes Neurais Artificiais

---

## Dilema Viés - Variância



# Redes Neurais Artificiais

---

## Exercício

- Implementar um *perceptron* simples para criar um classificador da função AND.
- Utilizar Python, preferencialmente.
- Apresentar graficamente o classificador no estado inicial (fase de atribuição dos pesos) e no estado final (após o treinamento), calculando a acurácia final.
- Meta: separar totalmente ambas as classes.

# Redes Neurais Artificiais

---

E agora?

## Comentários