

# Conteúdo

- 1 **Introdução**
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

# Conteúdo

- 1 **Introdução**
  - Motivação Biológica
  - Um Breve Histórico
- 2 **Perceptrons**
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

# Conteúdo

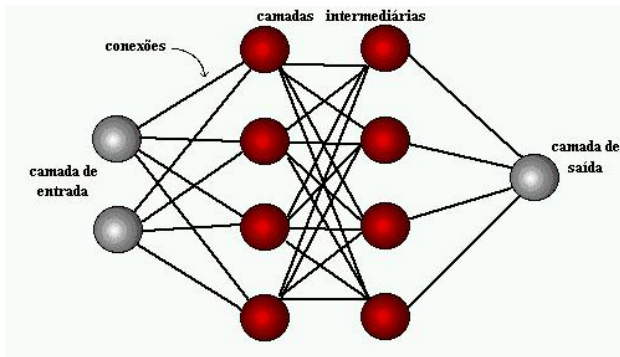
- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

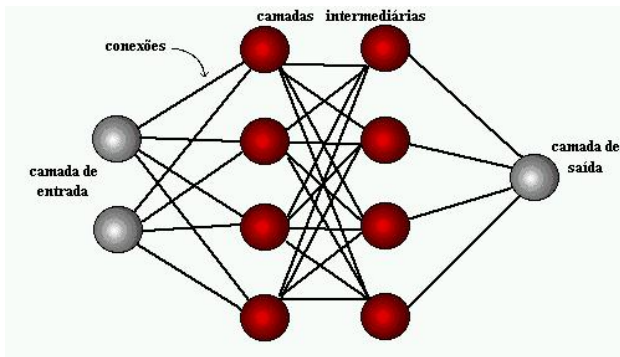
# Rede Neural Artificial

- sistema paralelo e distribuído.
- composto por unidades de processamento simples.
- unidades calculam funções matemáticas.



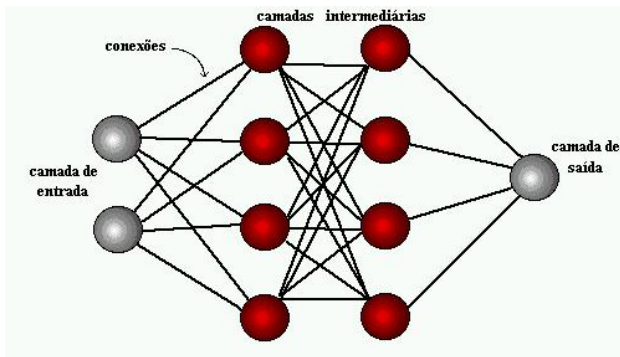
# Rede Neural Artificial

- sistema paralelo e distribuído.
- composto por unidades de processamento simples.
- unidades calculam funções matemáticas.



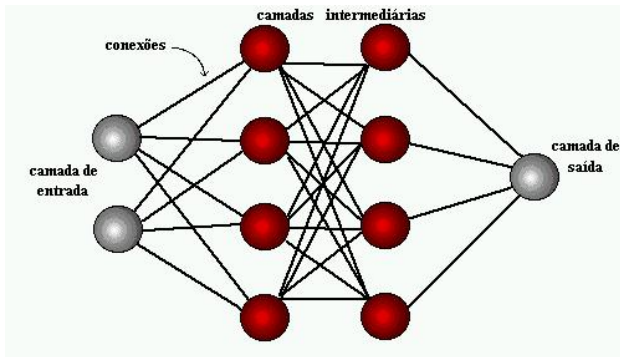
# Rede Neural Artificial

- sistema paralelo e distribuído.
- composto por unidades de processamento simples.
- unidades calculam funções matemáticas.



# Rede Neural Artificial

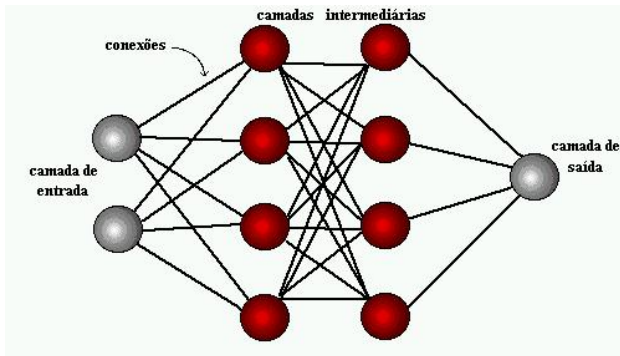
- unidades dispostas em uma ou mais camadas.
- interligadas por um grande número de conexões.
- conexões geralmente unidirecionais.





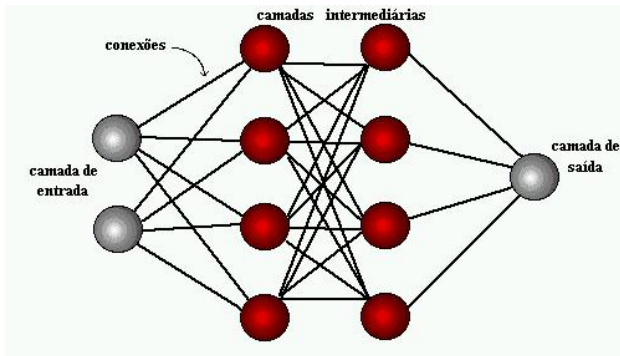
# Rede Neural Artificial

- unidades dispostas em uma ou mais camadas.
- interligadas por um grande número de conexões.
- conexões geralmente unidirecionais.



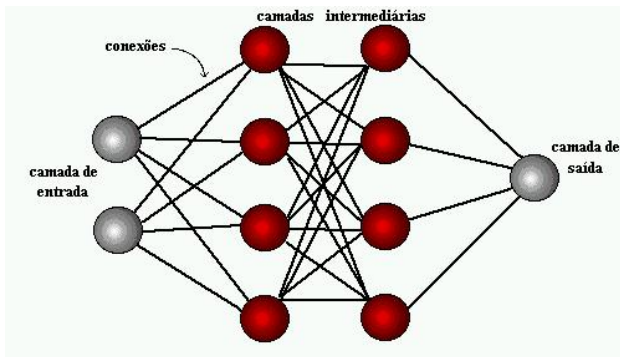
# Rede Neural Artificial

- unidades dispostas em uma ou mais camadas.
- interligadas por um grande número de conexões.
- **conexões geralmente unidirecionais.**



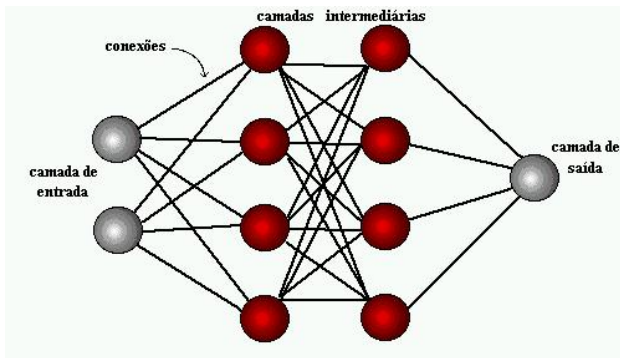
# Rede Neural Artificial

- pesos associados às conexões.
- pesos armazenam o conhecimento representado.
- pesos ponderam a entrada recebida por cada neurônio.



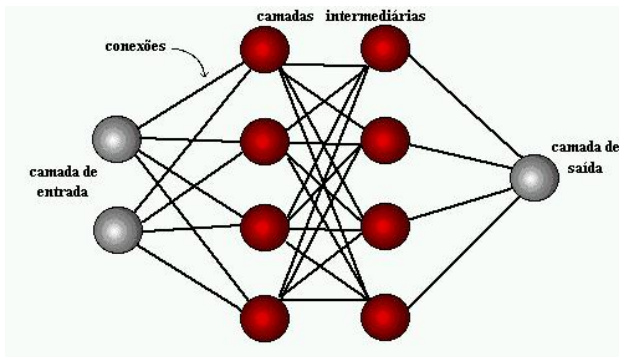
# Rede Neural Artificial

- pesos associados às conexões.
- pesos armazenam o conhecimento representado.
- pesos ponderam a entrada recebida por cada neurônio.



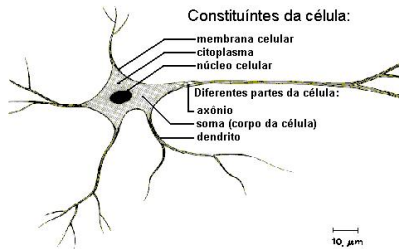
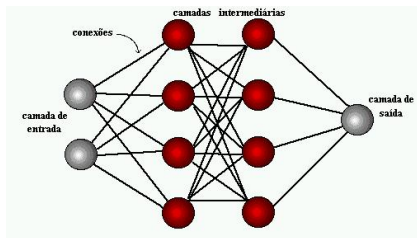
# Rede Neural Artificial

- pesos associados às conexões.
- pesos armazenam o conhecimento representado.
- pesos ponderam a entrada recebida por cada neurônio.



# Rede Neural Artificial

inspirada no cérebro humano.



# Conteúdo

## 1 Introdução

- **Motivação Biológica**
- Um Breve Histórico

## 2 Perceptrons

- Unidade de McCulloch-Pitts e Neurônio Artificial
- Funções de Ativação

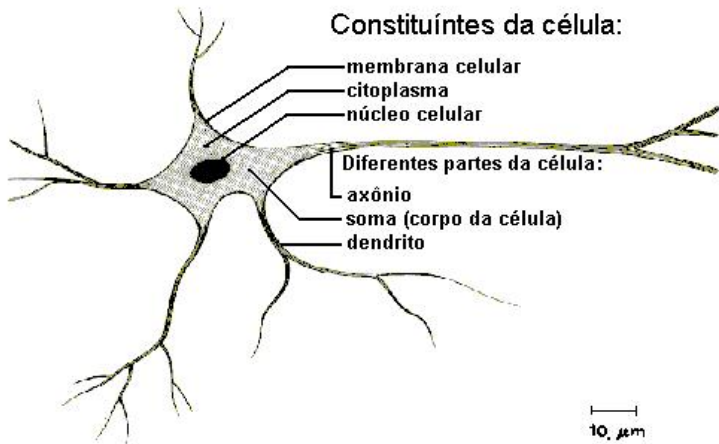
## 3 RNA's MLP e o Algoritmo Backpropagation

- Rede Perceptron Multicamada - MLP
- Algoritmo Backpropagation

## 4 Algumas Considerações

- Exemplos de Aplicações
- Considerações Finais
- Resumo
- Mais Informações

# O Neurônio Biológico

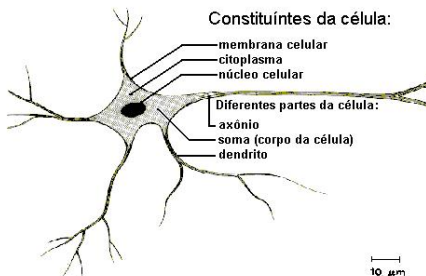




# Neurônios Biológicos

## Alguns números

- **aproximadamente  $10^{11}$  neurônios.**
- cada neurônio conectado a  $10^4$  outros, em média.
- $10^{-3}$ s é a ordem de tempo de chaveamento mais rápido.
- $10^{-1}$ s para reconhecimento da mãe.

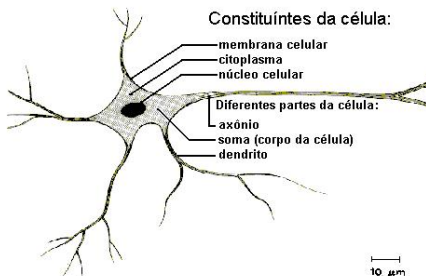


- $10^{-10}$ s - computador.
- centenas ou milhares de unidades.
- complexidades não modeladas.

# Neurônios Biológicos

## Alguns números

- aproximadamente  $10^{11}$  neurônios.
- cada neurônio conectado a  $10^4$  outros, em média.
- $10^{-3}$ s é a ordem de tempo de chaveamento mais rápido.
- $10^{-1}$ s para reconhecimento da mãe.

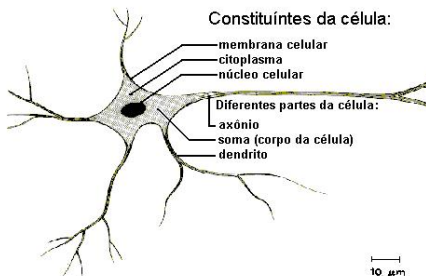


- $10^{-10}$ s - computador.
- centenas ou milhares de unidades.
- complexidades não modeladas.

# Neurônios Biológicos

## Alguns números

- aproximadamente  $10^{11}$  neurônios.
- cada neurônio conectado a  $10^4$  outros, em média.
- $10^{-3}s$  é a ordem de tempo de chaveamento mais rápido.
- $10^{-1}s$  para reconhecimento da mãe.

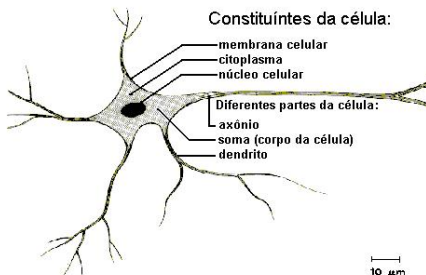


- $10^{-10}s$  - computador.
- centenas ou milhares de unidades.
- complexidades não modeladas.

# Neurônios Biológicos

## Alguns números

- aproximadamente  $10^{11}$  neurônios.
- cada neurônio conectado a  $10^4$  outros, em média.
- $10^{-3}s$  é a ordem de tempo de chaveamento mais rápido.
- $10^{-1}s$  para reconhecimento da mãe.

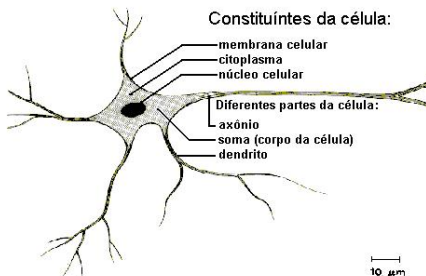


- $10^{-10}s$  - computador.
- centenas ou milhares de unidades.
- complexidades não modeladas.

# Neurônios Biológicos

## Alguns números

- aproximadamente  $10^{11}$  neurônios.
- cada neurônio conectado a  $10^4$  outros, em média.
- $10^{-3}s$  é a ordem de tempo de chaveamento mais rápido.
- $10^{-1}s$  para reconhecimento da mãe.



- $10^{-10}s$  - computador.
- centenas ou milhares de unidades.
- complexidades não modeladas.

# Conteúdo

## 1 Introdução

- Motivação Biológica
- **Um Breve Histórico**

## 2 Perceptrons

- Unidade de McCulloch-Pitts e Neurônio Artificial
- Funções de Ativação

## 3 RNA's MLP e o Algoritmo Backpropagation

- Rede Perceptron Multicamada - MLP
- Algoritmo Backpropagation

## 4 Algumas Considerações

- Exemplos de Aplicações
- Considerações Finais
- Resumo
- Mais Informações

# Breve Histórico

1943 McCulloch e Pitts estabelecem as bases da neurocomputação, com modelos matemáticos.

1949 Hebb traduz matematicamente a sinapse dos neurônios biológicos.

1951 Minski constrói o primeiro neurocomputador com capacidade de aprendizado (ajustava automaticamente os pesos entre as sinapses).

1958 Roseblatt concebe o “perceptron”, uma rede neural usada no reconhecimento de caracteres.

# Breve Histórico

1943 McCulloch e Pitts estabelecem as bases da neurocomputação, com modelos matemáticos.

1949 Hebb traduz matematicamente a sinapse dos neurônios biológicos.

1951 Minski constrói o primeiro neurocomputador com capacidade de aprendizado (ajustava automaticamente os pesos entre as sinapses).

1958 Roseblatt concebe o “perceptron”, uma rede neural usada no reconhecimento de caracteres.



# Breve Histórico

- 1943 McCulloch e Pitts estabelecem as bases da neurocomputação, com modelos matemáticos.
- 1949 Hebb traduz matematicamente a sinapse dos neurônios biológicos.
- 1951 Minski constrói o primeiro neurocomputador com capacidade de aprendizado (ajustava automaticamente os pesos entre as sinapses).
- 1958 Roseblatt concebe o “perceptron”, uma rede neural usada no reconhecimento de caracteres.

# Breve Histórico

- 1943 McCulloch e Pitts estabelecem as bases da neurocomputação, com modelos matemáticos.
- 1949 Hebb traduz matematicamente a sinapse dos neurônios biológicos.
- 1951 Minski constrói o primeiro neurocomputador com capacidade de aprendizado (ajustava automaticamente os pesos entre as sinapses).
- 1958 Roseblatt concebe o “perceptron”, uma rede neural usada no reconhecimento de caracteres.

# Breve Histórico

## continuação

1960 Widrow e Hoff sugerem a regra delta.

1969 Minsky e Papert chamam a atenção para tarefas que o perceptron não consegue executar.

1974 Werbos lança bases para o algoritmo backpropagation.

1982 Hopfield publica artigo que promove parte da retomada das pesquisas na área.

1986 Rumelhart, Hinton e Williams introduzem o backpropagation.

# Breve Histórico

## continuação

1960 Widrow e Hoff sugerem a regra delta.

1969 Minsky e Papert chamam a atenção para tarefas que o perceptron não consegue executar.

1974 Werbos lança bases para o algoritmo backpropagation.

1982 Hopfield publica artigo que promove parte da retomada das pesquisas na área.

1986 Rumelhart, Hinton e Williams introduzem o backpropagation.

# Breve Histórico

## continuação

1960 Widrow e Hoff sugerem a regra delta.

1969 Minsky e Papert chamam a atenção para tarefas que o perceptron não consegue executar.

1974 Werbos lança bases para o algoritmo backpropagation.

1982 Hopfield publica artigo que promove parte da retomada das pesquisas na área.

1986 Rumelhart, Hinton e Williams introduzem o backpropagation.

# Breve Histórico

## continuação

1960 Widrow e Hoff sugerem a regra delta.

1969 Minsky e Papert chamam a atenção para tarefas que o perceptron não consegue executar.

1974 Werbos lança bases para o algoritmo backpropagation.

1982 Hopfield publica artigo que promove parte da retomada das pesquisas na área.

1986 Rumelhart, Hinton e Williams introduzem o backpropagation.

# Breve Histórico

## continuação

- 1960 Widrow e Hoff sugerem a regra delta.
- 1969 Minsky e Papert chamam a atenção para tarefas que o perceptron não consegue executar.
- 1974 Werbos lança bases para o algoritmo backpropagation.
- 1982 Hopfield publica artigo que promove parte da retomada das pesquisas na área.
- 1986 Rumelhart, Hinton e Williams introduzem o backpropagation.

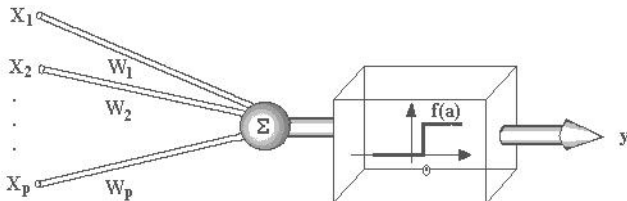
# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 **Perceptrons**
  - **Unidade de McCulloch-Pitts e Neurônio Artificial**
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações



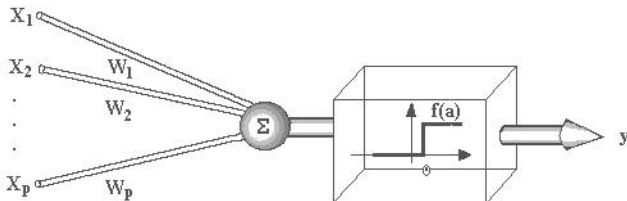
# Unidade de McCulloch-Pitts

- Sinais são apresentados à entrada.
- Cada sinal é multiplicado por um número (peso).
- É feita a soma ponderada que produz um nível de atividade.
- Se este nível de atividade exceder um limite, a unidade produz uma determinada resposta de saída.



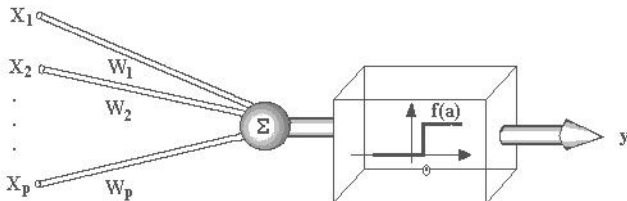
# Unidade de McCulloch-Pitts

- Sinais são apresentados à entrada.
- Cada sinal é multiplicado por um número (peso).
- É feita a soma ponderada que produz um nível de atividade.
- Se este nível de atividade exceder um limite, a unidade produz uma determinada resposta de saída.



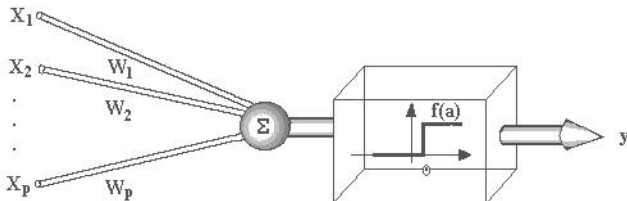
# Unidade de McCulloch-Pitts

- Sinais são apresentados à entrada.
- Cada sinal é multiplicado por um número (peso).
- É feita a soma ponderada que produz um nível de atividade.
- Se este nível de atividade exceder um limite, a unidade produz uma determinada resposta de saída.

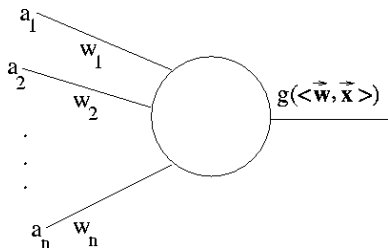


# Unidade de McCulloch-Pitts

- Sinais são apresentados à entrada.
- Cada sinal é multiplicado por um número (peso).
- É feita a soma ponderada que produz um nível de atividade.
- Se este nível de atividade exceder um limite, a unidade produz uma determinada resposta de saída.



# Neurônio Artificial



$$\vec{x} = [a_1, a_2, \dots, a_n]$$

Uma dupla ordenada  $U = (\vec{w}, g)$   
vetor de pesos

$$\vec{w} = [w_1, w_2, \dots, w_n]$$

função de ativação

$$g : \langle \vec{w}, \vec{x} \rangle \mapsto \mathbb{R}$$

$$\langle \vec{w}, \vec{x} \rangle = w_1 a_1 + w_2 a_2 + \dots + w_n a_n$$

# Conteúdo

## 1 Introdução

- Motivação Biológica
- Um Breve Histórico

## 2 Perceptrons

- Unidade de McCulloch-Pitts e Neurônio Artificial
- **Funções de Ativação**

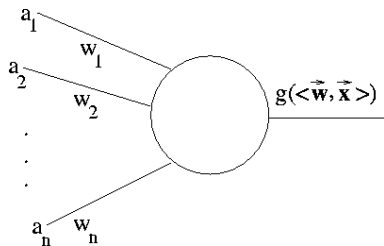
## 3 RNA's MLP e o Algoritmo Backpropagation

- Rede Perceptron Multicamada - MLP
- Algoritmo Backpropagation

## 4 Algumas Considerações

- Exemplos de Aplicações
- Considerações Finais
- Resumo
- Mais Informações

# Funções de Ativação - Neurônio Linear



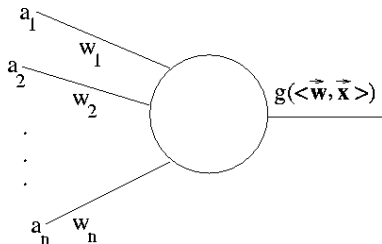
$$g: \mathbb{R} \mapsto \mathbb{R}$$

$$g(\langle \vec{w}, \vec{x} \rangle) = w_1 a_1 + w_2 a_2 + \cdots + w_n a_n$$

$$g(\langle \vec{w}, \vec{x} \rangle) = \langle \vec{w}, \vec{x} \rangle$$

# Funções de Ativação

continuação



$$g : \mathbb{R} \mapsto \{1, 0\}$$

$$g(\langle \vec{w}, \vec{x} \rangle) = \begin{cases} 1, & \text{se } w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq \theta \\ 0, & \text{se } w_1 a_1 + w_2 a_2 + \cdots + w_n a_n < \theta \end{cases}$$



# Funções de Ativação

## continuação

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq \theta$$

$$-\theta + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$(-\theta).1 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 a_0 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 = -\theta \quad a_0 = 1$$

# Funções de Ativação

## continuação

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq \theta$$

$$-\theta + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$(-\theta).1 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 a_0 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 = -\theta \quad a_0 = 1$$

# Funções de Ativação

## continuação

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq \theta$$

$$-\theta + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$(-\theta).1 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 a_0 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 = -\theta \quad a_0 = 1$$

# Funções de Ativação

## continuação

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq \theta$$

$$-\theta + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$(-\theta).1 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 a_0 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 = -\theta \quad a_0 = 1$$

# Funções de Ativação

## continuação

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq \theta$$

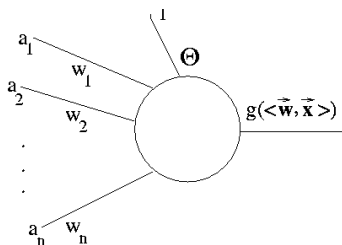
$$-\theta + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$(-\theta).1 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 a_0 + w_1 a_1 + w_2 a_2 + \cdots + w_n a_n \geq 0$$

$$w_0 = -\theta \quad a_0 = 1$$

# Função de Ativação



$$w_0 = -\theta \quad a_0 = 1$$

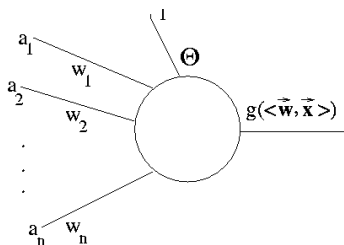
$$\vec{x} = [a_0, a_1, a_2, \dots, a_n]$$

$$\vec{w} = [w_0, w_1, w_2, \dots, w_n]$$

$$g : \mathbb{R} \mapsto \{1, 0\}$$

$$g(< \vec{w}, \vec{x} >) = \begin{cases} 1, & \text{se } w_0 a_0 + w_1 a_1 + w_2 a_2 + \dots + w_n a_n \geq 0 \\ 0, & \text{se } w_0 a_0 + w_1 a_1 + w_2 a_2 + \dots + w_n a_n < 0 \end{cases}$$

# Funções de Ativação



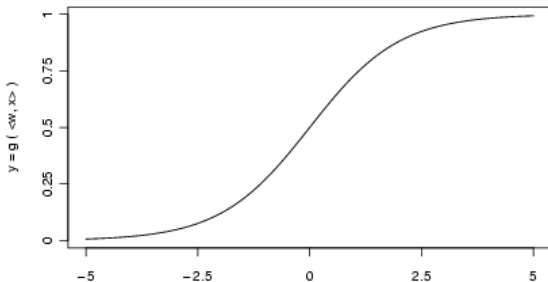
$$g : \mathbb{R} \mapsto \{1, -1\}$$

$$g(< \vec{w}, \vec{x} >) = \begin{cases} 1, & \text{se } w_0 a_0 + w_1 a_1 + \dots + w_n a_n \geq 0 \\ -1, & \text{se } w_0 a_0 + w_1 a_1 + \dots + w_n a_n < 0 \end{cases}$$

# Função Logística

$$g : \mathbb{R} \longrightarrow [0..1]$$

$$g(\langle \vec{w}, \vec{x} \rangle) = \frac{1}{1 + e^{-\langle \vec{w}, \vec{x} \rangle}}$$

Valores de  $\langle w, x \rangle$



# Função Logística

Forma mais geral

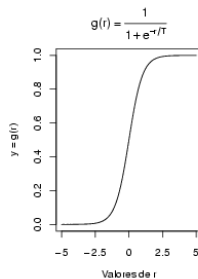
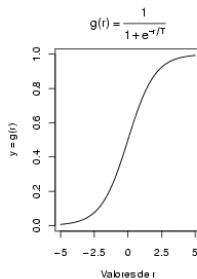
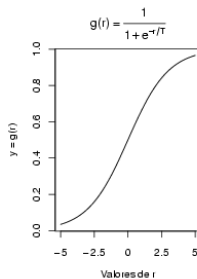
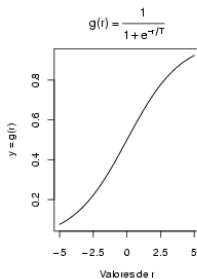
$$g : \mathbb{R} \longrightarrow [0..1]$$

$$g(r) = \frac{1}{1 + e^{-r/T}}$$

o parâmetro  $T$  oferece curvas suaves a íngremes à medida que seu valor é decrementado.

# Função Logística

Valores de  $T=2$ ,  $T=3/2$ ,  $T=1$  e  $T=1/2$

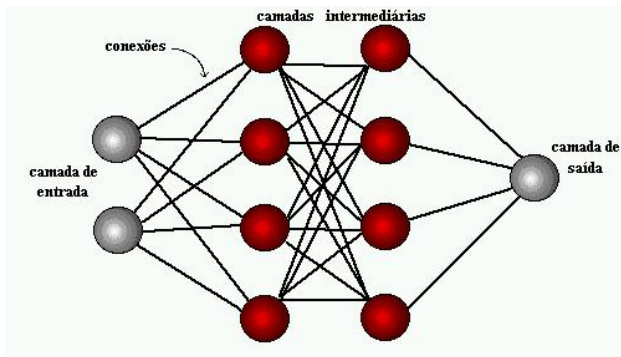


# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

# RNA MLP

## Perceptron Multicamada



## Questão:

Como se dá o processo de aprendizado?

## Resposta:

- obter valores para os pesos  $\vec{w} = [w_0, \dots, w_n]$  dos neurônios...
- satisfazendo determinada condição geralmente a minimização de alguma função de erro.
- A hipótese numa rede neural é um conjunto de vetores de pesos.

## Questão:

Como se dá o processo de aprendizado?

## Resposta:

- obter valores para os pesos  $\vec{w} = [w_0, \dots, w_n]$  dos neurônios...
- satisfazendo determinada condição geralmente a minimização de alguma função de erro.
- A hipótese numa rede neural é um conjunto de vetores de pesos.

## Aprendizado de uma rede com um único neurônio

Seja  $\mathcal{T} = \left\{ \left( \vec{x}_i, f(\vec{x}_i) \right) \right\}_{i=1}^m$  uma amostra de treinamento e

$\vec{w} = [w_1, w_2, \dots, w_n]$  um vetor de pesos.

O **erro de treinamento** de  $\vec{w}$ , relativo a amostra de treinamento  $\mathcal{T}$ , é definido pela função

$$E(\vec{w}) = \frac{1}{2} \sum_{\vec{x} \in \mathcal{T}} (f(\vec{x}) - s(\vec{x}))^2$$

onde  $f(\vec{x})$  é o valor da função alvo para um exemplo  $\vec{x}$  e  $s(\vec{x})$  é a saída obtida para um exemplo  $\vec{x}$ .

# Gradiente

Seja  $\vec{w}$  um vetor de pesos e  $E(\vec{w})$  uma função erro de treinamento relativa a uma amostra de treinamento  $\mathcal{T}$ .

$$\nabla E(\vec{w}) = \nabla E([w_0, w_1, \dots, w_n]) = \left[ \frac{\partial E(\vec{w})}{\partial w_0}, \frac{\partial E(\vec{w})}{\partial w_1}, \dots, \frac{\partial E(\vec{w})}{\partial w_n} \right]$$



# Regra de treinamento para o gradiente descendente

$$w_i \leftarrow w_i + \Delta w_i$$

sendo

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

onde  $\eta$  é a taxa de aprendizado.

$$w_i = w_i + \eta \sum_{\vec{x} \in \mathcal{T}} (f(\vec{x}) - s(\vec{x})) a_{i\vec{x}}$$

# Gradiente Descendente Incremental

$$\Delta w_i = \eta(f(\vec{x}) - s(\vec{x}))a_i$$

onde  $f(\vec{x})$  é o valor da função alvo,  $s(\vec{x})$  é o valor de saída e  $a_i$  o  $i$ -ésimo valor de entrada do exemplo considerado.

Alternativa com menor custo computacional.

A função de erro é definida para cada exemplo da seguinte forma:

$$E_{\vec{x}}(\vec{w}) = \frac{1}{2}(f(\vec{x}) - s(\vec{x}))^2$$

onde  $f(\vec{x})$  é o valor da função alvo e  $s(\vec{x})$  é o valor de saída para o exemplo  $\vec{x}$ .

## Função erro para rede com mais de um neurônio

a função erro é a soma dos erros sobre todas as unidades de saída da rede:

$$E(\vec{w}) = \frac{1}{2} \sum_{\vec{x} \in \mathcal{T}} \sum_{k \in \text{saídas}} (f_k(\vec{x}) - s_k(\vec{x}))^2$$

onde

- saídas é o conjunto de neurônios cujos valores de saída são os valores de saída da rede
- $f_k(\vec{x})$  e  $s_k(\vec{x})$  são o valor alvo e valor de saída associados ao  $k$ -ésimo neurônio e exemplo  $\vec{x}$ .

# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 **RNA's MLP e o Algoritmo Backpropagation**
  - Rede Perceptron Multicamada - MLP
  - **Algoritmo Backpropagation**
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

Backpropagation( $\mathcal{T}, \eta, n_{entrada}, n_{saida}, n_{oculta}$ )

Cada exemplo de treinamento é um par da forma  $(\vec{x}, f(\vec{x}))$ , onde

$\vec{x}$  é o vetor de valores de entrada da rede e  $f(\vec{x})$  é o vetor de valores de saída.

$\eta$  é a taxa de aprendizado (por exemplo 0,05),  $n_{entrada}$  é o número de entradas da rede,

$n_{oculta}$  é o número de neurônios da camada oculta e

$n_{saida}$  é o número de neurônios de saída.

A entrada do nodo  $i$  no nodo  $j$  é denotada por  $x_{ji}$  e o peso do neurônio  $i$

no neurônio  $j$  é denotado por  $w_{ji}$ .

- Criar uma rede acíclica com  $n_{entrada}$ ,  $n_{oculta}$  neurônios ocultos e  $n_{saida}$  neurônios de saída.
- Inicializar todos os pesos da rede com pequenos números aleatórios (ex., entre  $-0,05$  e  $0,05$ ).

- Até que a condição de término seja satisfeita faça
  - para cada  $(\vec{x}, f(\vec{x}))$  nos exemplos de treinamento faça
    1. Entre com a instância  $\vec{x}$  na rede e calcule a saída  $s_u$  do neurônio  $u$  na rede
    2. Para cada neurônio de saída  $k$ , calcule o seu termo de erro  $\delta_k$

$$\delta_k \leftarrow s_k(1 - s_k)(f_k - s_k)$$

- 3. Para cada neurônio oculto  $o$ , calcule o seu termo de erro  $\sigma_o$

$$\delta_o \leftarrow s_o(1 - s_o) \sum_{k \in \text{saídas}} w_{ko} \delta_k$$

- 4. Atualize cada peso  $w_{ji}$

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

onde

$$\Delta w_{ji} = \eta \delta_j a_{ji}$$

# Algumas Condições de Término

- 1 encerrar após um número máximo de iterações.
- 2 encerrar quando  $E(\vec{w})$  for menor que um determinado valor.
- 3 encerrar quando a proporção de classificações corretas numa amostra de teste estiver acima de um determinado valor.



# Weight Decay

$$E(\vec{w}) = \frac{1}{2} \sum_{\vec{x} \in \mathcal{T}} \sum_{k \in \text{saídas}} (f_k(\vec{x}) - s_k(\vec{x}))^2 + \frac{1}{2} \lambda \|\vec{w}\|^2$$

- para minimizar *overfitting*.
- formalmente gera vetores de pesos de norma mínima.
- $\lambda$  dita a taxa de decremento.

# Momentum

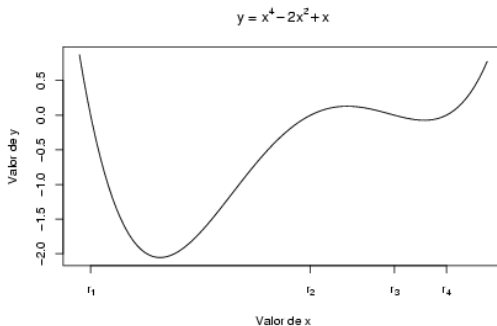
$$\psi = \alpha(w_{ji}(t) - w_{ji}(t - 1))$$

- para evitar lentidão.
- para evitar convergência para mínimo local.
- $\lambda$  dita a taxa de decremento.

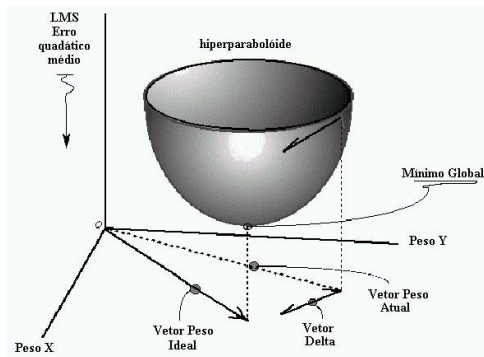
O ajuste de pesos do algoritmo *backpropagation* passa então a ser dado por

$$\Delta w_{ji} = \eta \delta_j a_{ji} + \alpha(w_{ji}(t) - w_{ji}(t - 1))$$

# Mínimo Global



# Mínimo Global



# Poder Representacional

“Qual o conjunto de funções que podem ser representadas?”

## Funções Booleanas

“Toda função booleana pode ser representada exatamente por alguma rede com duas camadas de unidades”.

# Poder Representacional

## continuação

“Qual o conjunto de funções que podem ser representadas?”

### Funções Contínuas

“Toda função contínua limitada pode ser aproximada com pequeno erro arbitrário por uma rede com duas camadas de unidades” (Cybenko 1989; Hornik et al. 1989).

# Poder Representacional

## continuação

“Qual o conjunto de funções que podem ser representadas?”

### Funções Arbitrárias

“Qualquer função pode ser aproximada com uma precisão arbitrária por uma rede com três camadas de unidades” (Cybenko 1988).

# Poder Representacional

## continuação

Pode-se então perceber que...

Redes com arquitetura não extremamente complexa oferecem um espaço de hipótese expressivo.

Contudo...

os vetores de peso obtidos por gradiente descendente a partir dos pesos iniciais podem não incluir todos os vetores de pesos possíveis.



# Poder Representacional

## continuação

Pode-se então perceber que...

Redes com arquitetura não extremamente complexa oferecem um espaço de hipótese expressivo.

Contudo...

os vetores de peso obtidos por gradiente descendente a partir dos pesos iniciais podem não incluir todos os vetores de pesos possíveis.

# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - **Exemplos de Aplicações**
  - Considerações Finais
  - Resumo
  - Mais Informações

# Exemplos de Aplicações

- **Reconhecimento de imagens**
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - **classificação de caracteres**
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - **reconhecimento de assinaturas**
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - **reconhecimento de faces**
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - **classificação de fonemas**
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas



# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - **reconhecimento de comandos**
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - **análise de crédito**
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - previsão e falência de empresas

# Exemplos de Aplicações

- Reconhecimento de imagens
  - classificação de caracteres
  - reconhecimento de assinaturas
  - reconhecimento de faces
- Reconhecimento de sons
  - classificação de fonemas
  - reconhecimento de comandos
- Classificação financeira
  - análise de crédito
  - análise de investimentos
  - **previsão e falência de empresas**

# Exemplos de Aplicações

continuação

- Monitoramento (tráfego aéreo)

## Novas aplicações

- Pen PC's (reconhecimento de escrita)
- Brinquedos (comandos como Vá! Pare! - lógica fuzzy com redes neurais artificiais)

# Exemplos de Aplicações

## continuação

- Monitoramento (tráfego aéreo)

### Novas aplicações

- Pen PC's (reconhecimento de escrita)
- Brinquedos (comandos como Vá! Pare! - lógica fuzzy com redes neurais artificiais)

# Exemplos de Aplicações

continuação

- Monitoramento (tráfego aéreo)

## Novas aplicações

- Pen PC's (reconhecimento de escrita)
- Brinquedos (comandos como Vá! Pare! - lógica fuzzy com redes neurais artificiais)



# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - **Considerações Finais**
  - Resumo
  - Mais Informações

# Considerações Finais

- Área de Redes Neurais Artificiais é **vasta**.
- Tem se mostrada **atrativa** para profissionais de domínios distintos.
- Psicólogos, neurofisiologistas, engenheiros, **cientistas da computação** etc.
- É **interdisciplinar**.
- Buscas por **novos caminhos**, cada um em sua área.

# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - **Resumo**
  - Mais Informações

# Resumo

- Motivação inicial **biológica**.
- Período de **agitação, diminuição e retomada** na pesquisa.
- **Rede Neural: Arquitetura + Algoritmo de Aprendizado**.
- RNA's MLP com Backpropagation são **bem utilizadas na prática**.

# Conteúdo

- 1 Introdução
  - Motivação Biológica
  - Um Breve Histórico
- 2 Perceptrons
  - Unidade de McCulloch-Pitts e Neurônio Artificial
  - Funções de Ativação
- 3 RNA's MLP e o Algoritmo Backpropagation
  - Rede Perceptron Multicamada - MLP
  - Algoritmo Backpropagation
- 4 Algumas Considerações
  - Exemplos de Aplicações
  - Considerações Finais
  - Resumo
  - Mais Informações

# Para Mais Informações



A. Braga, A. Carvalho e T. Ludermir.

*Redes Neurais Artificiais: Teoria e Aplicações.*

LTC, 2000.



Z. Kovács.

*Redes Neurais Artificiais: Fundamentos e Aplicações.*

Livraria da Física, 2002.



T. Mitchell.

*Machine Learning.*

McGraw-Hill, 1997.

# Para Mais Informações



A. Braga, A. Carvalho e T. Ludermir.

*Redes Neurais Artificiais: Teoria e Aplicações.*

LTC, 2000.



Z. Kovács.

*Redes Neurais Artificiais: Fundamentos e Aplicações.*

Livraria da Física, 2002.



T. Mitchell.

*Machine Learning.*

McGraw-Hill, 1997.

# Para Mais Informações



A. Braga, A. Carvalho e T. Ludermir.

*Redes Neurais Artificiais: Teoria e Aplicações.*

LTC, 2000.



Z. Kovács.

*Redes Neurais Artificiais: Fundamentos e Aplicações.*

Livraria da Física, 2002.



T. Mitchell.

*Machine Learning.*

McGraw-Hill, 1997.



# Conteúdo do Apêndice

- 5 Apêndice
  - Outros Tipos de RNA's

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes



# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes

# Outros Tipos de RNA's

- Memórias Matriciais
- Redes Self-Organizing
  - SOM
  - ART
- Sistemas Neurais Híbridos
  - RNA's e Algoritmos Genéticos
  - RNA's com Raciocínio Baseado em Casos
- Redes Neurais sem Pesos
- Redes RBF
- Redes Construtivas
- Redes Recorrentes