

# *I WorkShop de Inteligência Artificial - UNISC*

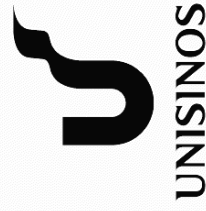
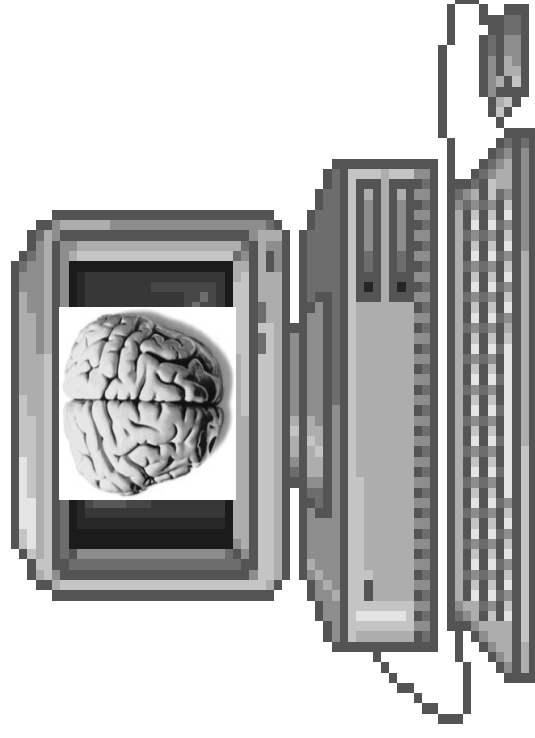
---

## **Sistemas Inteligentes baseados em Redes Neurais Artificiais aplicados ao Processamento de Imagens**

***Coordenador do Projeto:  
Prof. Dr. Fernando Osório***

***Bolsista CNPq :  
João Ricardo Bittencourt***

***UNISC - Santa Cruz do Sul  
Junho de 2000***



**UNISINOS - Centro de Ciências Exatas e Tecnológicas (C6/6)  
Curso de Informática - Mestrado em Computação Aplicada  
E-mail: [osorio@exatas.unisinós.br](mailto:osorio@exatas.unisinós.br)  
Web: <http://www.inf.unisinós.br/~osorio/>**



## *Tópicos abordados*

### **1. Introdução & Conceitos Básicos**

#### **1.1. Projeto HMLT**

**1.1. Inteligência: Humana e Artificial**

**1.2. Aprendizado de máquinas**

### **2. Redes Neurais Artificiais - RNAs**

**2.1. Conceitos Básicos**

**2.2. Representação de Conhecimentos**

**2.3. Modelos de RNAs**

**2.4. Aprendizado Neural**

**2.5. Discussão: vantagens / desvantagens**

### **3. Processamento de Imagens**

**3.1. Conceitos e Aplicações**

**3.1.1. Tratamento de Imagens**

**3.1.2. Reconhecimento de Padrões**

**3.2. Processamento de Imagens Convencional**

**3.3. Processamento de Imagens Neural**

**3.4. Exemplos de Aplicações: OCR, Filtros, etc.**

### **6. Conclusões e Perspectivas**

# **Introdução: Contexto da Pesquisa**

---

**Inteligência Humana  $\Rightarrow$  Inteligência Artificial**

**Aprendizado Humano  $\Rightarrow$  Aprendizado de Máquinas**

**Métodos de Raciocínio e Aquisição de Conhecimentos Múltiplos  $\Rightarrow$  Sistemas Híbridos**

---

---

# Introdução

---

Inteligência Humana ⇒ Inteligência Artificial

Aprendizado Humano ⇒ Aprendizado de Máquinas

Métodos de Raciocínio e Aquisição de Conhecimentos Múltiplos ⇒ Sistemas Híbridos

---

---

## Sistemas Inteligentes Híbridos

Projeto de Pesquisa HMLT - *Hybrid Machine Learning Tools*

‘Ferramentas Híbridas de Aprendizado para o Máquinas’

Coordenador ..... Prof. Fernando Osório

Bolsistas de Iniciação Científica.... Carla Medeiros Barros

João Ricardo de Bittencourt Menezes

Rafael Guterres Jeffman

Mestrando..... Farlei Heinen

Cooperação: Laboratoire LEIBNIZ - Grenoble, França e LRI/UFRGS



# INTELIGÊNCIA: Humana e Artificial

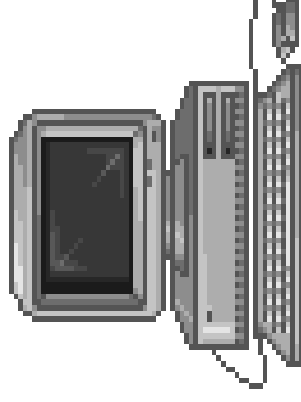
---



REPRODUZIR A



INTELIGÊNCIA HUMANA



- O que é Inteligência ?

- O que é um ser Inteligente ?

# INTELIGÊNCIA: Humano e Artificial

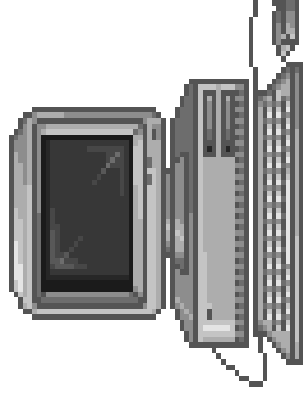
---



REPRODUZIR A



INTELIGÊNCIA HUMANA



- O que é Inteligência ?

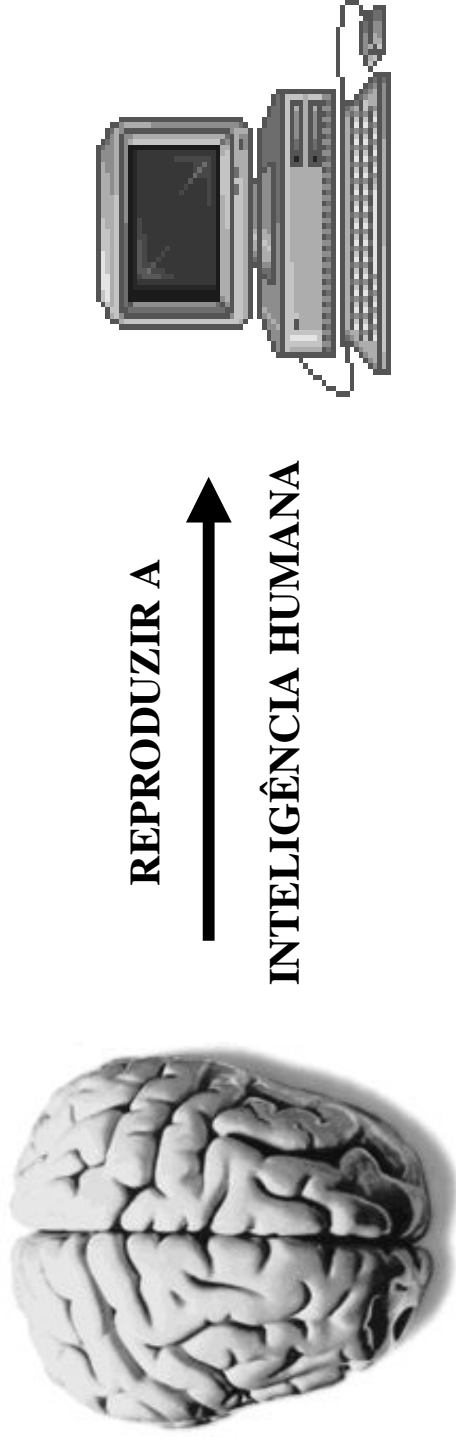
- O que é um ser Inteligente ?

- \* Associação de idéias e conceitos
- \* Concluir coisas
- \* Capacidade de aprendizado
- \* Acúmulo de conhecimentos
- \* Raciocínio: lógico, abstrato, dedução, analogia, indução, inferência, síntese, análise

- \* Uso prático de experiências e conhecimentos passados
- \* Tomada de decisões
- \* Criar coisas novas (criatividade)
- \* Saber o que eu sei (saber explicar)
- \* Interação
- \* Comunicação

# Inteligência Artificial: Conceitos Básicos

---



- **Conceito de Inteligência:** CAPACIDADE DE RESOLVER PROBLEMAS  
CAPACIDADE DE APRENDER  
CAPACIDADE DE SE ADAPTAR / MELHORAR
- **Realizar Atividades Inteligentes**
- **Sistemas Inteligentes:**
  - \* **Sistemas Especialistas**
  - \* **Sistemas de Apoio ao Diagnóstico e a Decisão**
  - \* **Reprodução de atividades típicas dos seres humanos:**  
Fala, Audição, Visão, Deslocamento, Manipulação de Objetos, etc.
  - \* **Jogos: jogo da velha, xadrez, jogos de ação**

# Inteligência Artificial: Conceitos Básicos

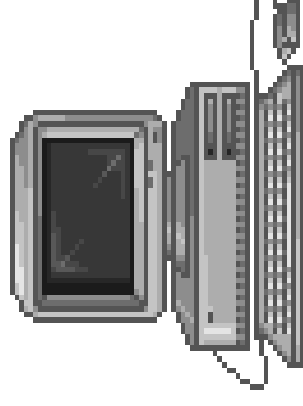
---



REPRODUZIR A



INTELIGÊNCIA HUMANA



- **Sistemas Inteligentes:**

**CAPACIDADE DE RESOLVER PROBLEMAS**  
**CAPACIDADE DE APRENDER**  
**CAPACIDADE DE SE ADAPTAR / MELHORAR**

⇒ Grandes Desafios:

- Linguagem / PLN
- Visão Artificial
- Robótica Autônoma

« *Sentidos Humanos* «

\* Problema escolhido:

Processamento de Imagens

\* Ferramentas Usadas:

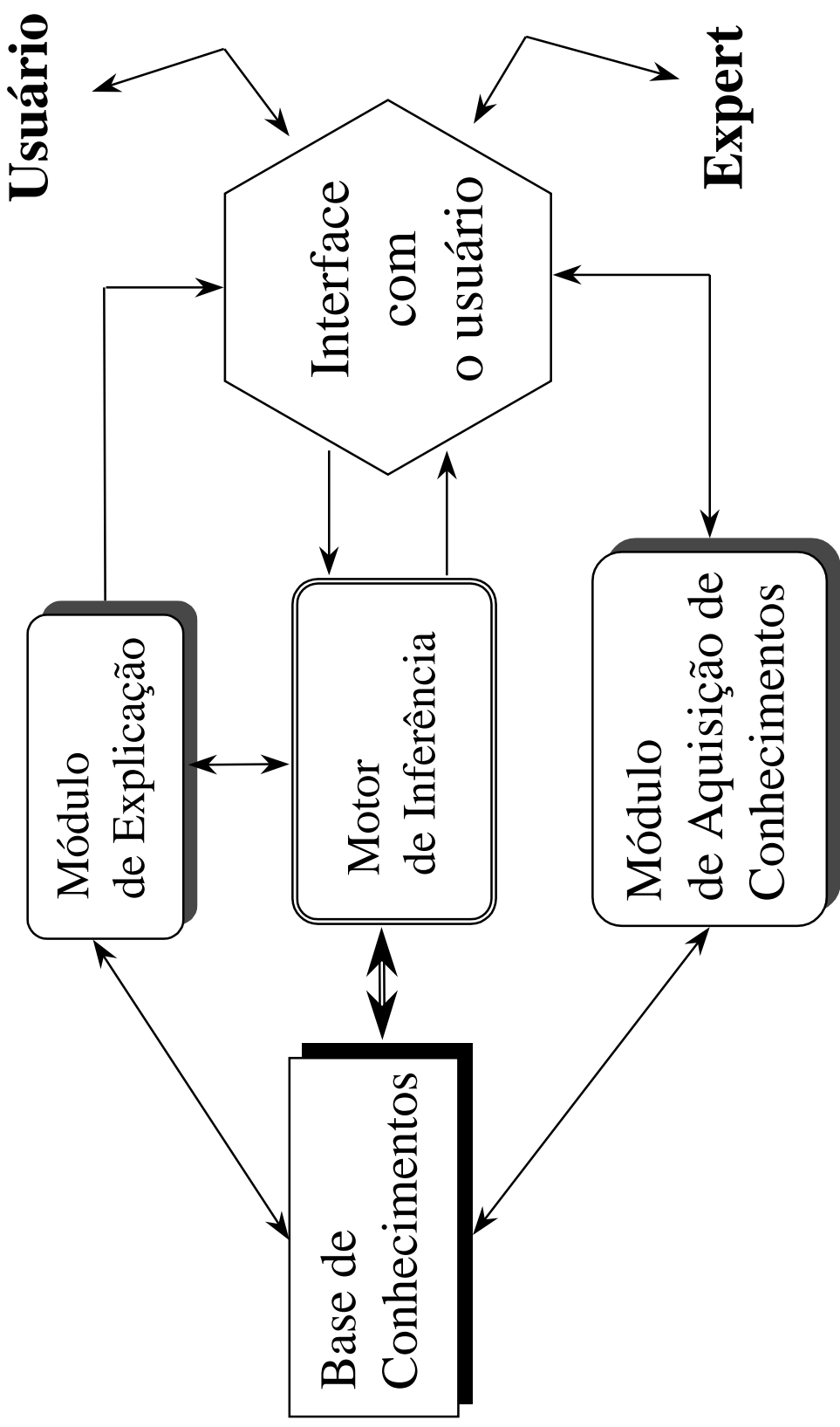
Redes Neurais & Sistemas Híbridos



# Inteligência Artificial: Adquirindo e usando conhecimentos

---

## Sistemas Especialistas



## Sistema Especialista: Exemplo Básico e Introdutório:

---

Sistema de auxílio ao diagnóstico de pacientes

Médico: 8 perguntas sobre os sintomas do paciente (resposta: ‘S’/‘N’)

Sintomas: 1 = Dor de Cabeça, 2 = Febre, 3 = Problemas digestivos, ...

### Base de Conhecimentos do Médico

1	2	3	4	5	6	7	8	Diagnóstico
S	S	N	S	N	S	S	S	Gripe
S	N	S	S	S	N	N	S	Sem problemas
S	N	S	N	S	N	S	N	Morte certa
S	N	N	S	S	N	S	N	Morte certa

Criar uma árvore binária de decisão baseada na tabela de conhecimentos

Consultas ao sistema:

S,N,S,N,S,N,S,N => Diagnóstico ?

S,N,?,?,S,N,S,N => Diagnóstico ?



# Inteligência Artificial: Sistemas Especialistas

## Diagnóstico Médico

<i>Sintomas</i>		<i>Gerais</i>	
Variável	Tipo	Valores	
1	Sexo – SEXE	Discreto	{feminino, masculino} - 0/1
2	Temperatura – TEMP	Contínuo	{ ... }
	Temp_Low	Discreto	0/1
	Temp_Normal	Discreto	0/1
	Temp_High	Discreto	0/1
<i>Sintomas</i>			
<i>Neurológicos</i>			
3	Estado – CALME	Discreto	{calmo, agitado} - 0/1
4	Foto-motores – PHOTO (reação das pupilas)	Discreto	{presente, ausente} - 0/1
5	Olhar – REGARD	Discreto	{normal, anormal} - 0/1
6	Estado das Pupilas - PUPIL	Discreto	{myosis, intermediário, mydriase}
	Pupil_myosis	Discreto	0/1
	Pupil_intermed	Discreto	0/1
	Pupil_mydriase	Discreto	0/1
7	Tonus muscular - TONUS	Discreto	{hipertonia, hipotonia}
8	Reflexos dos tendões - ROT	Discreto	{vivos, diminuídos}
<i>Sintomas</i>			
<i>cardiológicos</i>			
9	Pressão Arterial - PAS	Contínuo	{...}
	PAS_Low	Discreto	0/1
	PAS_Normal	Discreto	0/1
	PAS_High	Discreto	0/1
10	Frequência Cardíaca - FC	Contínuo	{...}
	FC_Low	Discreto	0/1
	FC_Normal	Discreto	0/1
	FC_High	Discreto	0/1
11	Medida ECG – QRS	Contínuo	{...}
	QRS_Normal	Discreto	0/1
	QRS_Prolongado	Discreto	0/1
12	Intervalo QT – QT	Discreto	{normal, prolongado} - 0/1
<i>Urina</i>			
13	Bexiga - GLOBE_VESICAL	Discreto	{sim, não} - 0/1

# Inteligência Artificial: Sistemas Especialistas

## Diagnóstico: Substância tóxica ingerida

	<i>Abreviações</i>	<i>Substância Tóxica</i>	<i>Ocorrência</i>
1	adt, a	Anti-depressores tri-cíclicos	265
2	B	Barbitúricos	86
3	ben, b	Benzodiazepina	414
4	C	Carbamato	68
5	P	Fenotiazina	130
6	M	Morfina	13
7	E	Alcool	137
<b><i>Combinações</i></b>			
a (25), aBb (12), aBbp (9), ab (104), abc (11), abm (5), abp (38), ap (5), B (8), Bb (23), Bbcp (5), Bcp (13), ben (37), bc (24), bm (8), bp (31), c (5), p (5), Ea (12), Eab (36), Eabp (8), E (12), EB (6), EBb (5), EBbp (5), Eb (32), Ebc (10), Ebp (11)			

Tabela - Os diferentes tipos de substâncias tóxicas da base de dados

# Inteligência Artificial: Aprendizado de Máquinas

---

## \* Sistemas Especialistas de 1a. Geração:

- Aquisição manual de conhecimentos
- Problemas: Base de Conhecimentos (regras e fatos)

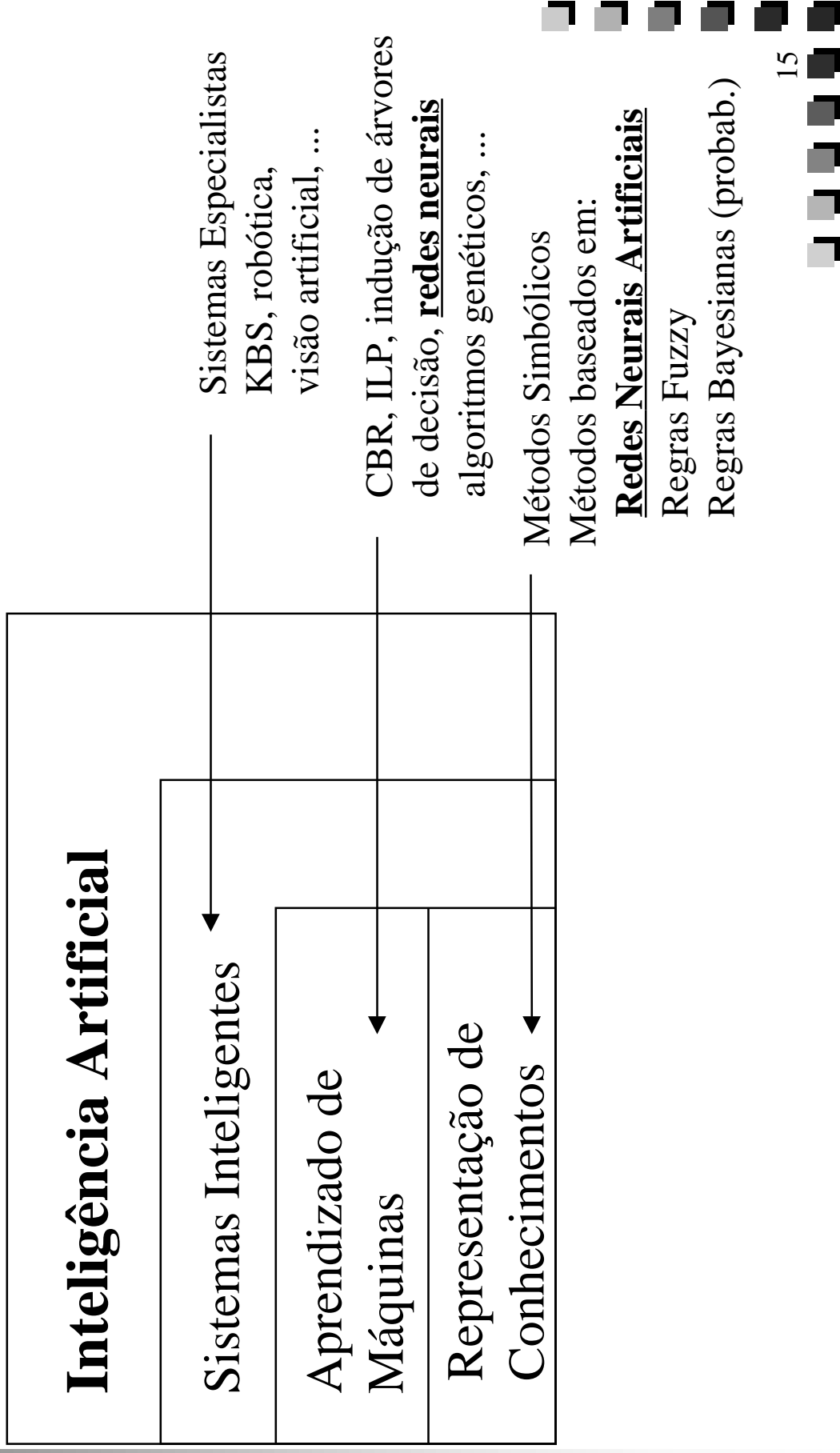
## \* Sistemas Especialistas de 2a. Geração:

- Aquisição automática de conhecimentos
- Integração de diferentes métodos da I.A.

“Sistemas Híbridos”

# Inteligência Artificial: Aprendizado de Máquinas

---



# Inteligência Artificial: Aprendizado de Máquinas

---

- O que é o aprendizado?

- \* Adaptação do comportamento (melhoria)
- \* Correção dos erros cometidos no passado
- \* Otimização da performance do sistema (melhoria)
- \* Interação com o meio, experimentação e descoberta
- \* Representação do conhecimento adquirido  
Memória e compressão dos conhecimentos



# Aprendizado de Máquinas / Machine Learning

---

⇒ **Conhecimentos Teóricos / Simbólicos**

$$\text{XOR} = ( A \text{ Or } B ) \text{ And } \text{Not} ( A \text{ And } B )$$

*ou*

$$\text{XOR} = ( A \text{ And } \text{Not} ( B ) ) \text{ Or } ( \text{Not} ( A ) \text{ And } B )$$

⇒ **Conhecimentos Empíricos / Dados**

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

# Aprendizado de Máquinas / Machine Learning

---

⇒ **Conhecimentos Teóricos / Simbólicos**

Se existem 2 casas na horizontal, vertical ou diagonal  
com uma marca do jogador oponente  
e a terceira casa está livre  
Então jogar nesta casa!

⇒ **Conhecimentos Empíricos / Dados**

$X_2$		$X$
$O_1$	$O_3$	$X_4$
$O_5$		

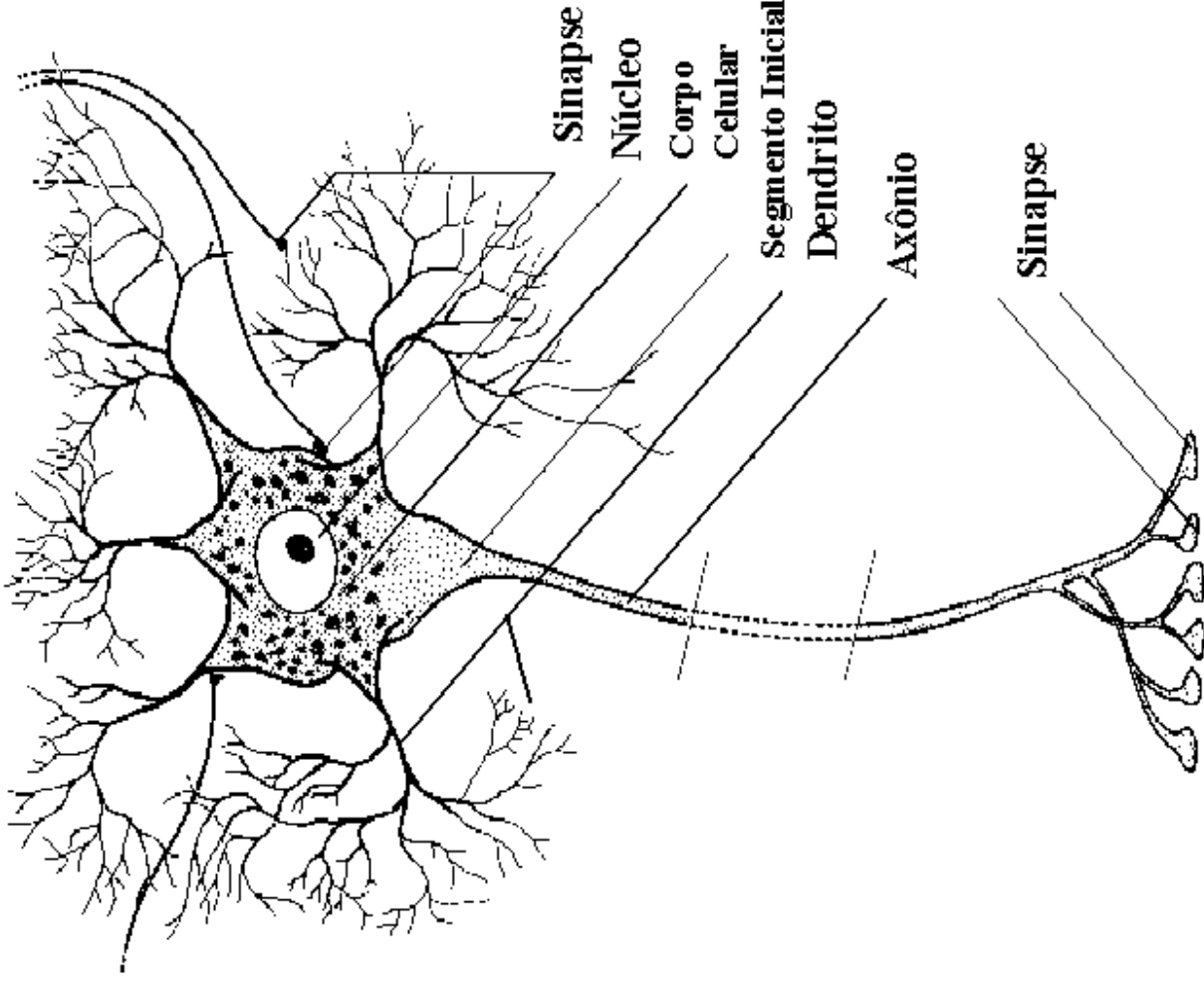
## Métodos de Aprendizado de Máquinas

---

- Aprendizado por analogia / por instâncias  
Sistemas baseados em casos  
*CBR - Case Based Reasoning*
- Aprendizado por Indução  
Indução de Árvores de Decisão  
ID3, C4.5, CN2 - *Induction of Decision Trees*  
ILP - *Inductive Logic Programming (Prolog)*
- Aprendizado por evolução/seleção  
Algoritmos Genéticos  
GA e GP - *Genetic Algorithms / Genetic Programming*
- Aprendizado por reforço (*reinforcement learning*)
- Aprendizado Bayesiano (probabilista)
- Aprendizado Neural  
MLP Back-Propagation - Artificial Neural Networks

# Redes Neurais Artificiais: Aprendizado de Máquinas

---



**Redes Neurais Artificiais:**

**Neurônio...  
Modelo Simulado  
Modelo SIMPLIFICADO**

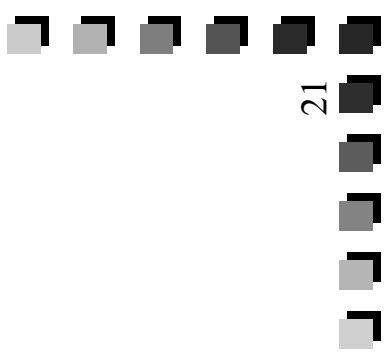
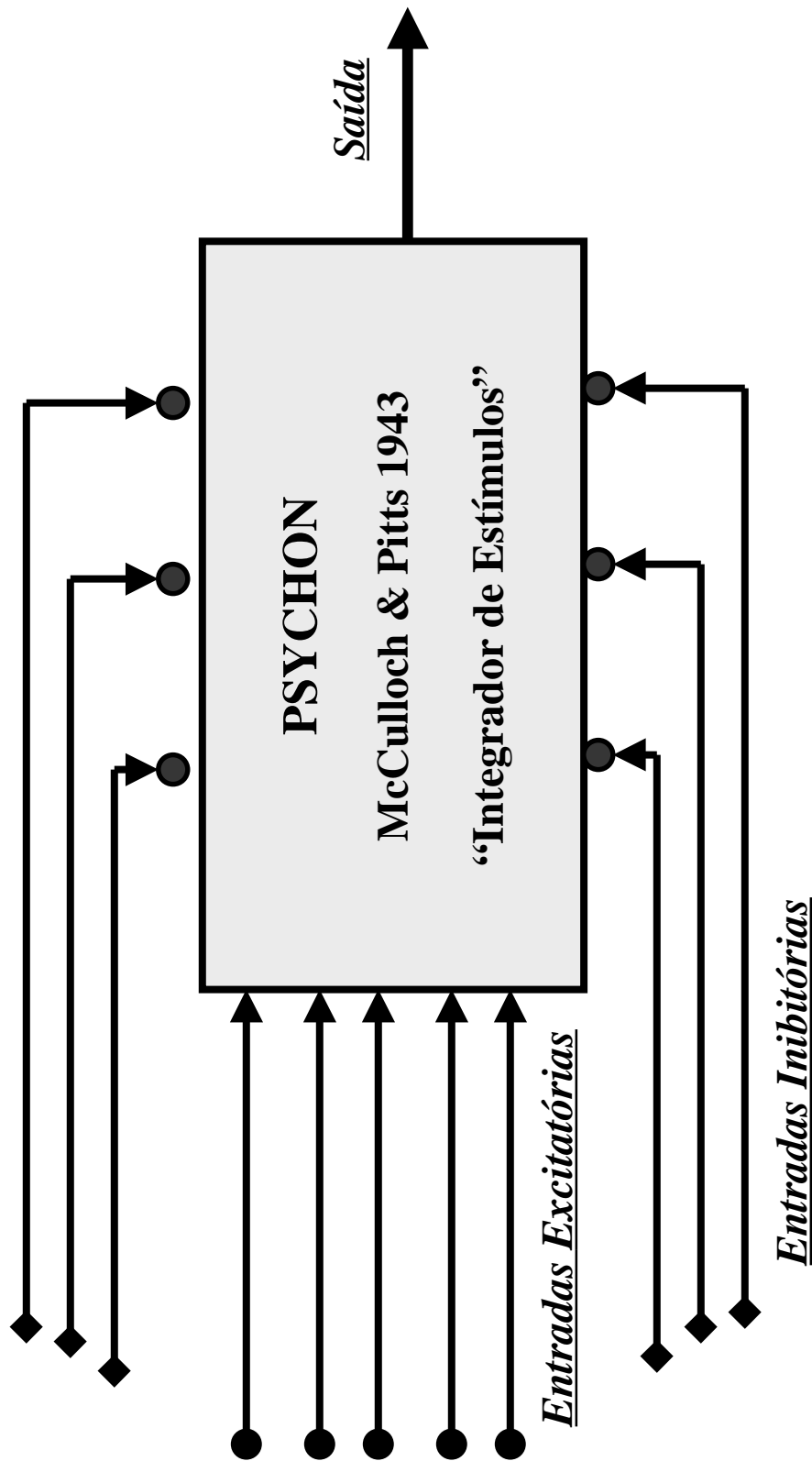
**Características Básicas:  
Adaptação  
Aprendizado  
Autômato**

**Representação de  
Conhecimentos:**

**Baseada em Conexões**

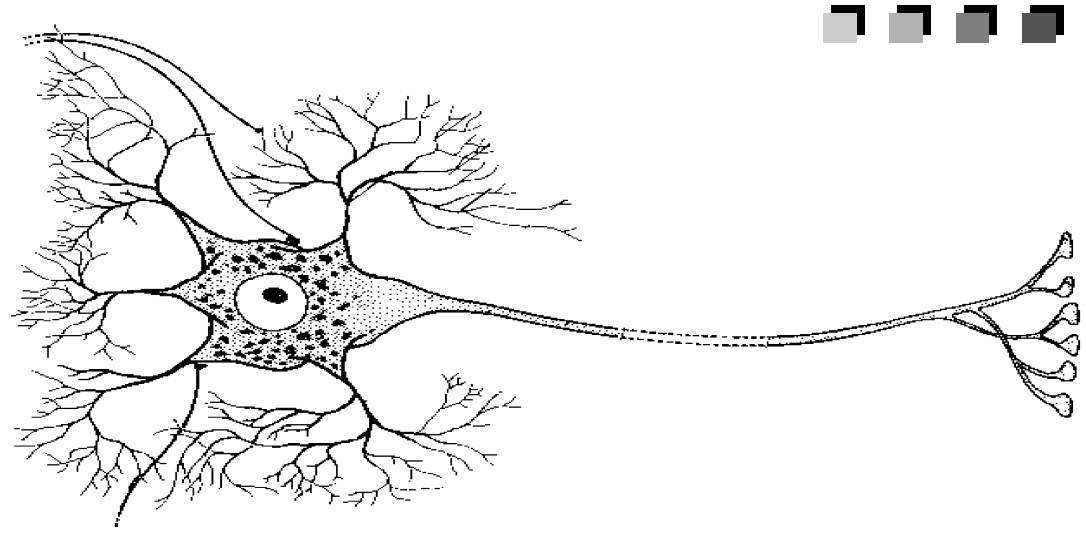
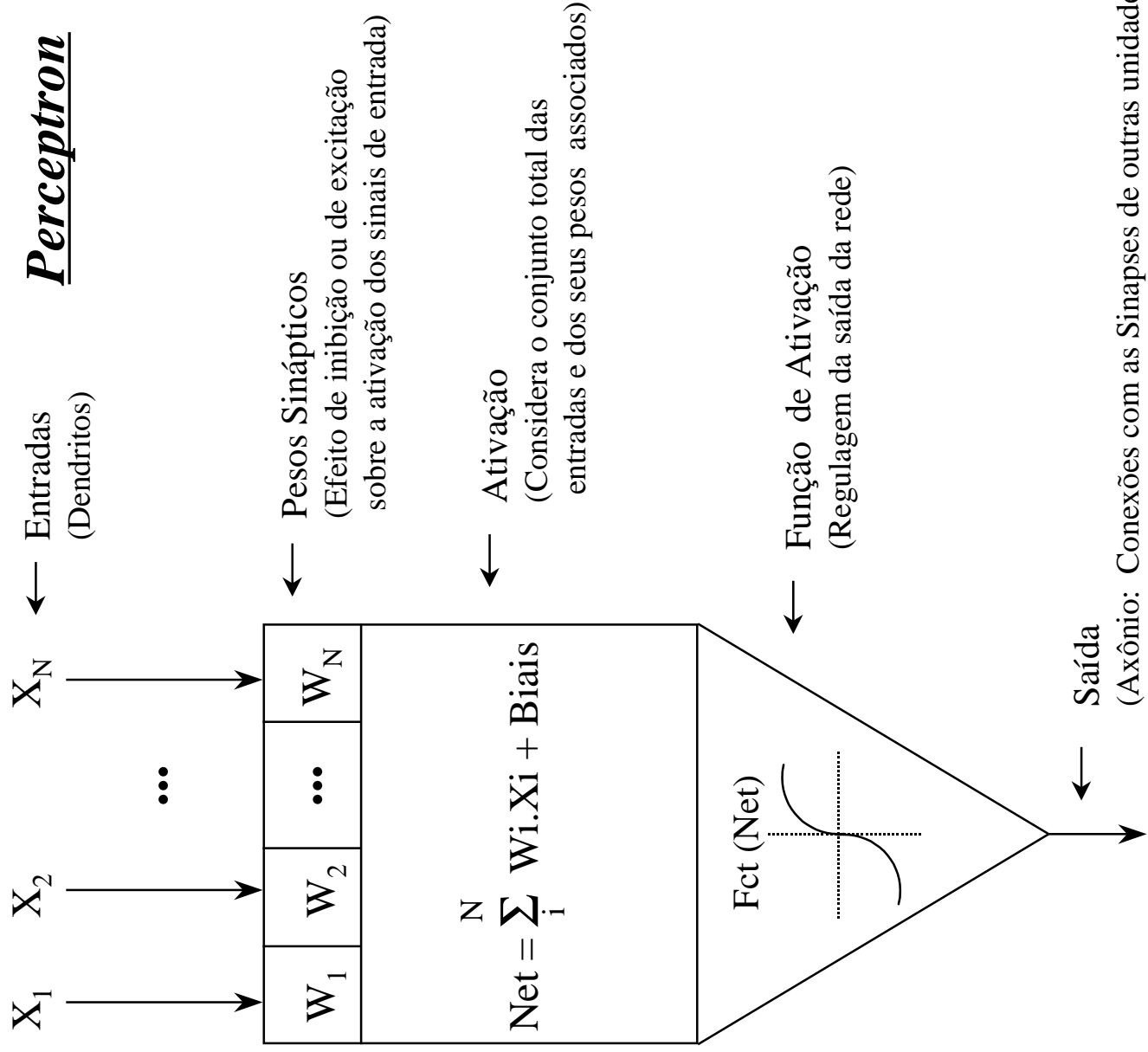
# Redes Neurais Artificiais: Origem

---



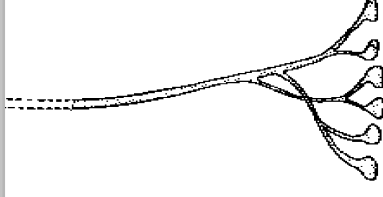
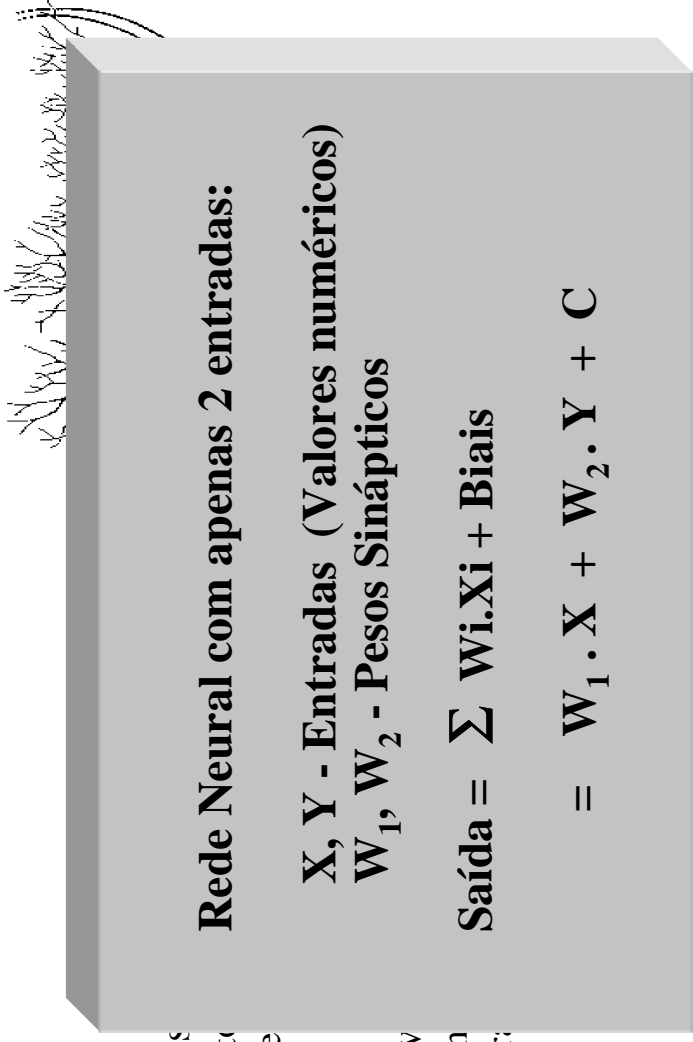
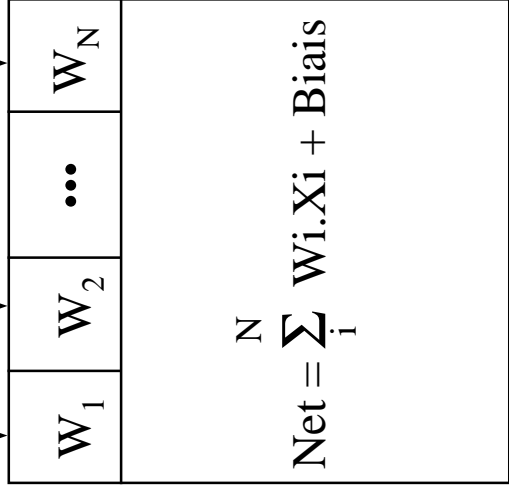
# NEURAL

## Perceptron



# NEURAL

$X_1$   $X_2$  ...  $X_N$  ← Entradas (Dendritos)



23

Axônio: Conexões com as Sinapses de outras unidades)

### Rede Neural com apenas 2 entradas:

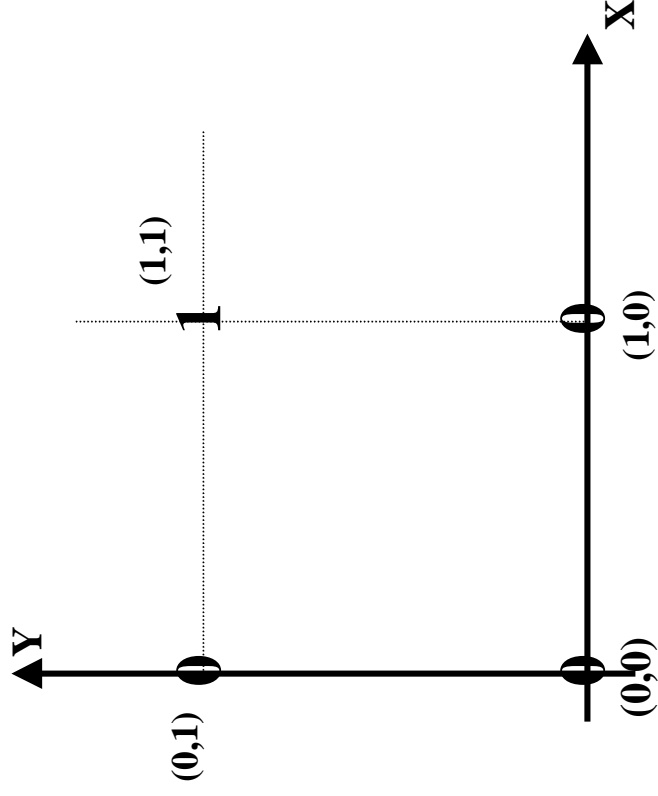
**X, Y - Entradas (Valores numéricos)**

**$W_1, W_2$  - Pesos Sinápticos**

**Saída =  $\sum W_i \cdot X_i + \text{Biais}$**

$$= W_1 \cdot X + W_2 \cdot Y + C$$

X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1



**Representação Geométrica do Problema**



## Rede Neural com apenas 2 entradas:

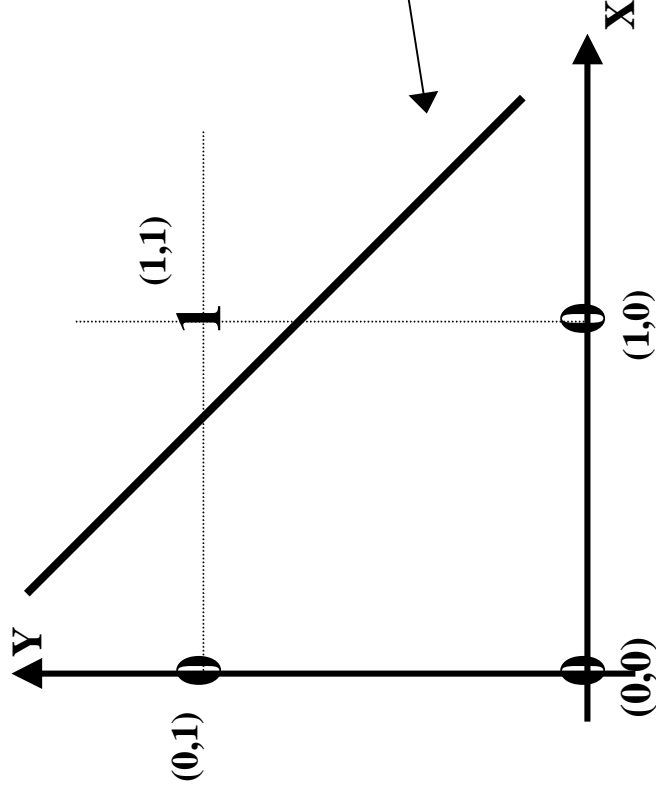
$X, Y$  - Entradas (Valores numéricos)

$W_1, W_2$  - Pesos Sinápticos

Saída =  $\sum W_i \cdot X_i + \text{Biais}$

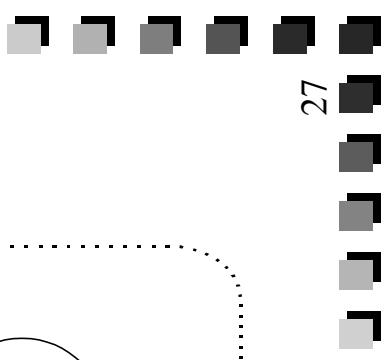
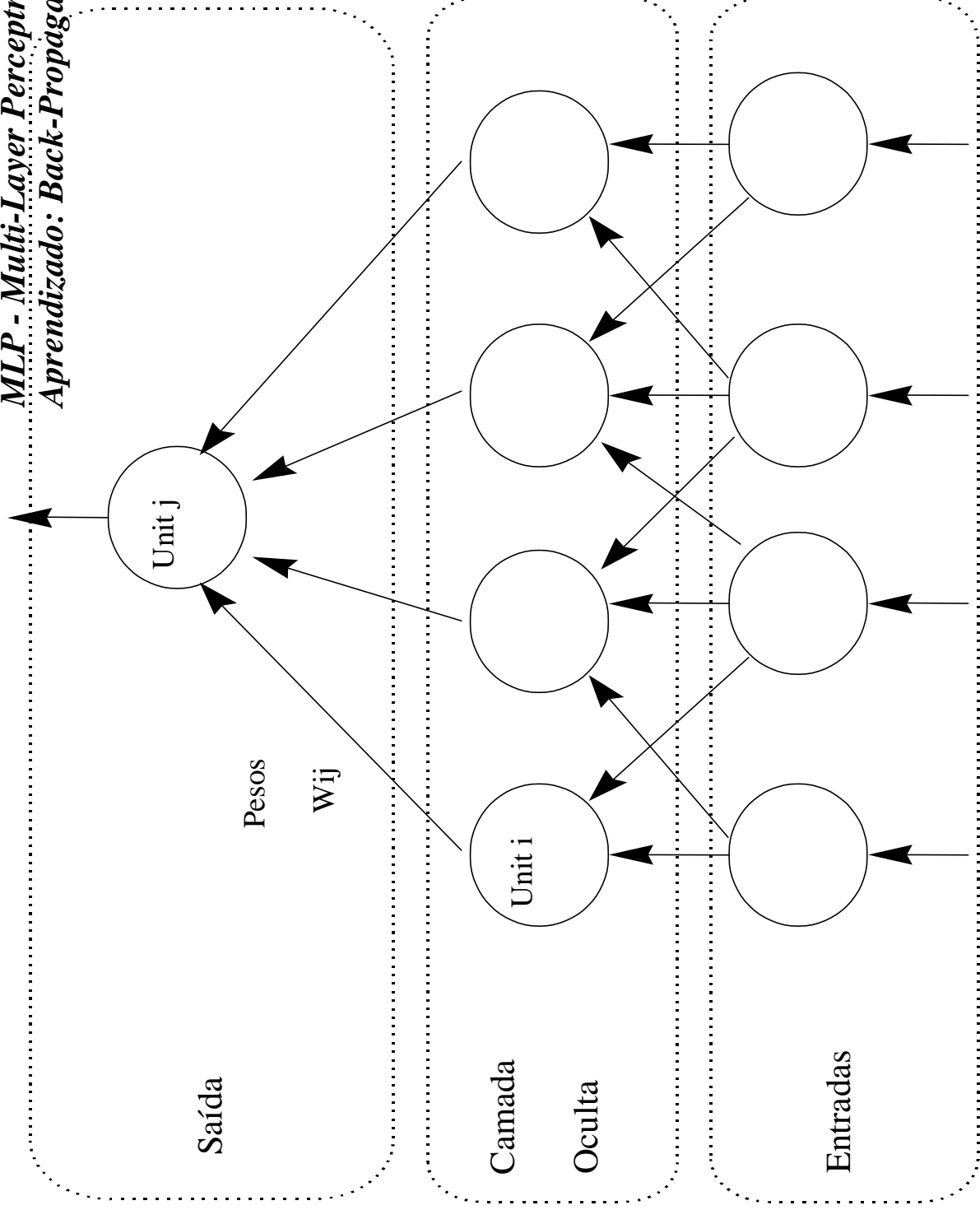
$$= W_1 \cdot X + W_2 \cdot Y + C$$

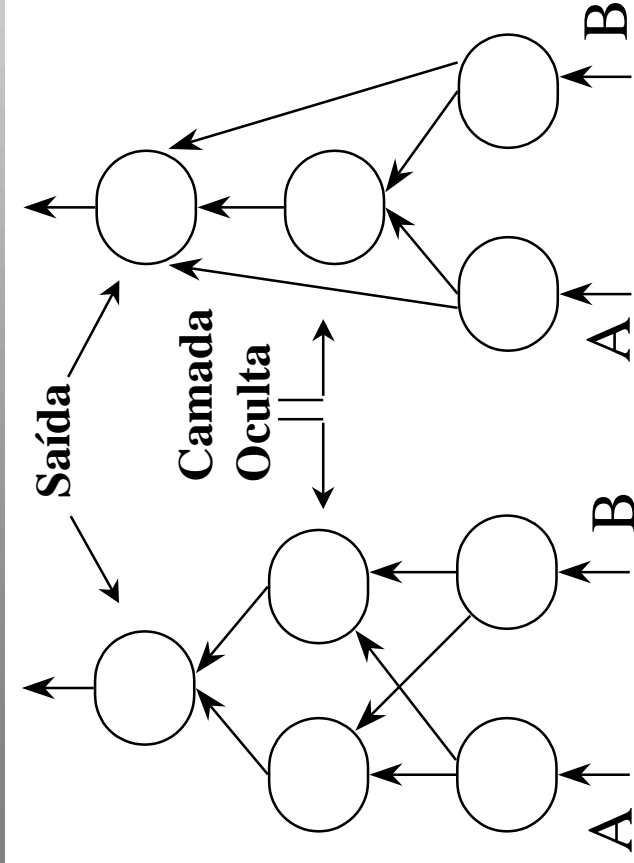
X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1





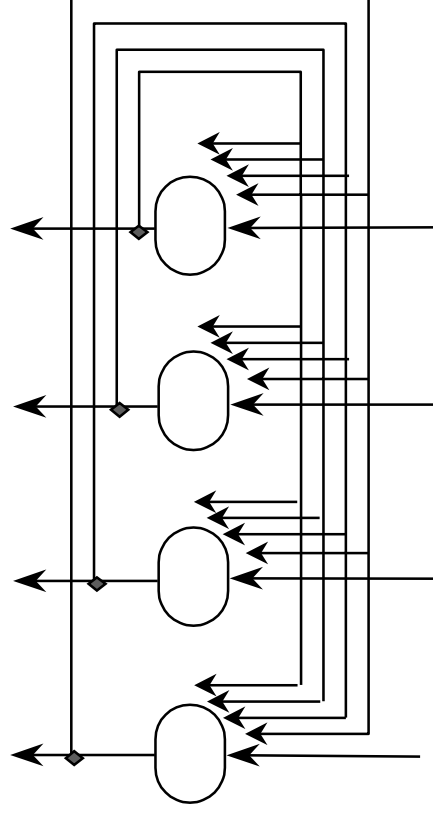
*MLP - Multi-Layer Perceptron*  
*Aprendizado: Back-Propagation*



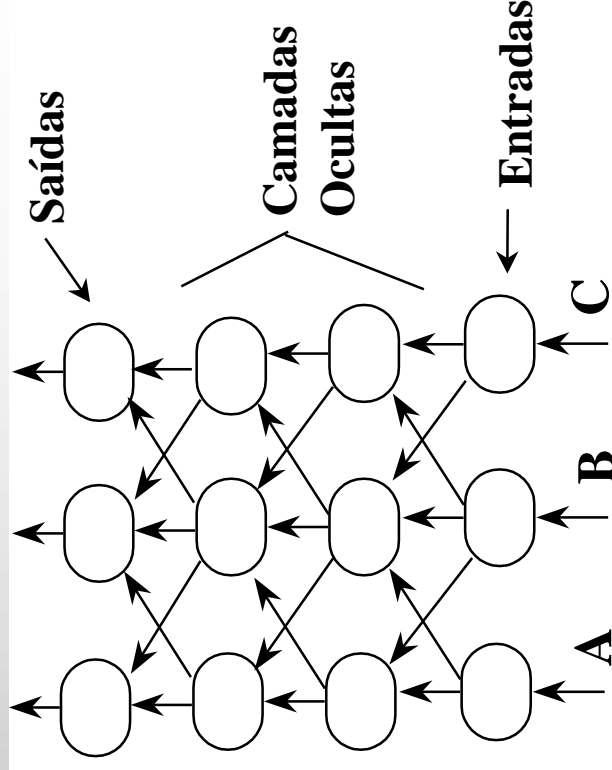


(a) Rede de três camadas

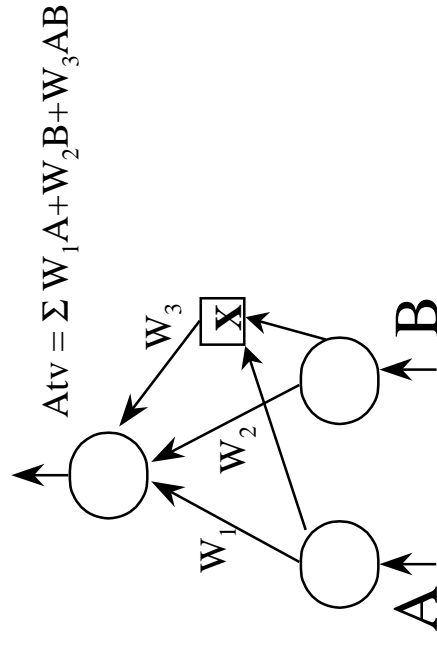
(b) Rede com atalhos



(d) Rede recorrente



(c) Rede com múltiplas camadas

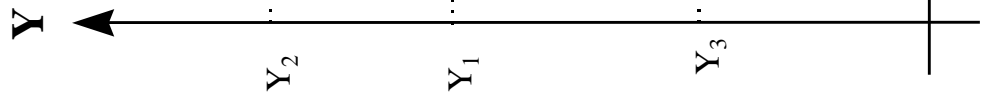


(e) Rede de ordem superior

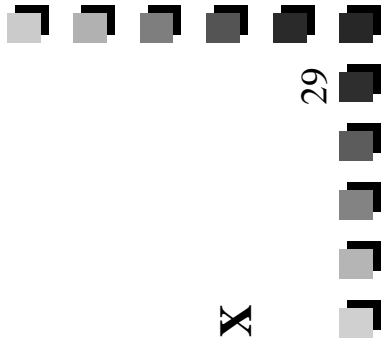
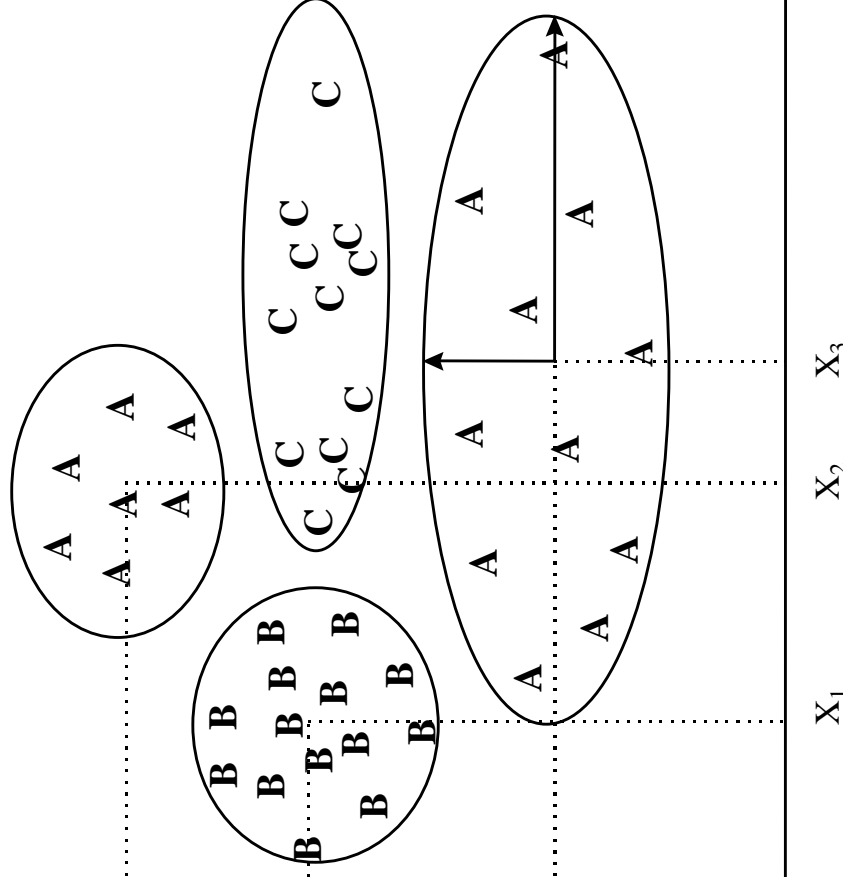
# Redes Neurais: Modelos Conexionistas

## Redes à base de Protótipos

### Entradas - X, Y Saída - Classes (A, B ou C)



- A: Exemplos da classe A
- B: Exemplos da classe B
- C: Exemplos da classe C
  
- $X_1, Y_1$  - Protótipo da classe B
- $X_2, Y_2$  - Protótipo da classe A
- $X_3, Y_3$  - Protótipo da classe A
  
- Protótipos:
  - \* Centro de Gravidade
  - \* Raio de influência (x,y)
  
- Teste de similaridade:
  - \* Distância Euclidiana



## Redes Neurais: Modelos Conexionistas

---

⇒ Em relação as unidades da rede:

- \* Redes baseadas em Perceptrons (MLP - *Multi-Layer Perceptron*)
- \* Redes baseadas em Protótipos (RBF - *Radial Basis Function*)

⇒ Em relação a estrutura da rede:

- \* Redes de uma única camada
- \* Redes de múltiplas camadas
- \* Redes do tipo uni-direcional (*Feed-Forward*)
- \* Redes do tipo recorrentes (*Feed-Back*)
- \* Redes com estrutura estática (não altera sua estrutura)
- \* Redes com estrutura dinâmica (altera a estrutura)
- \* Redes com conexões de ordem superior

# Redes Neurais: Aprendizado

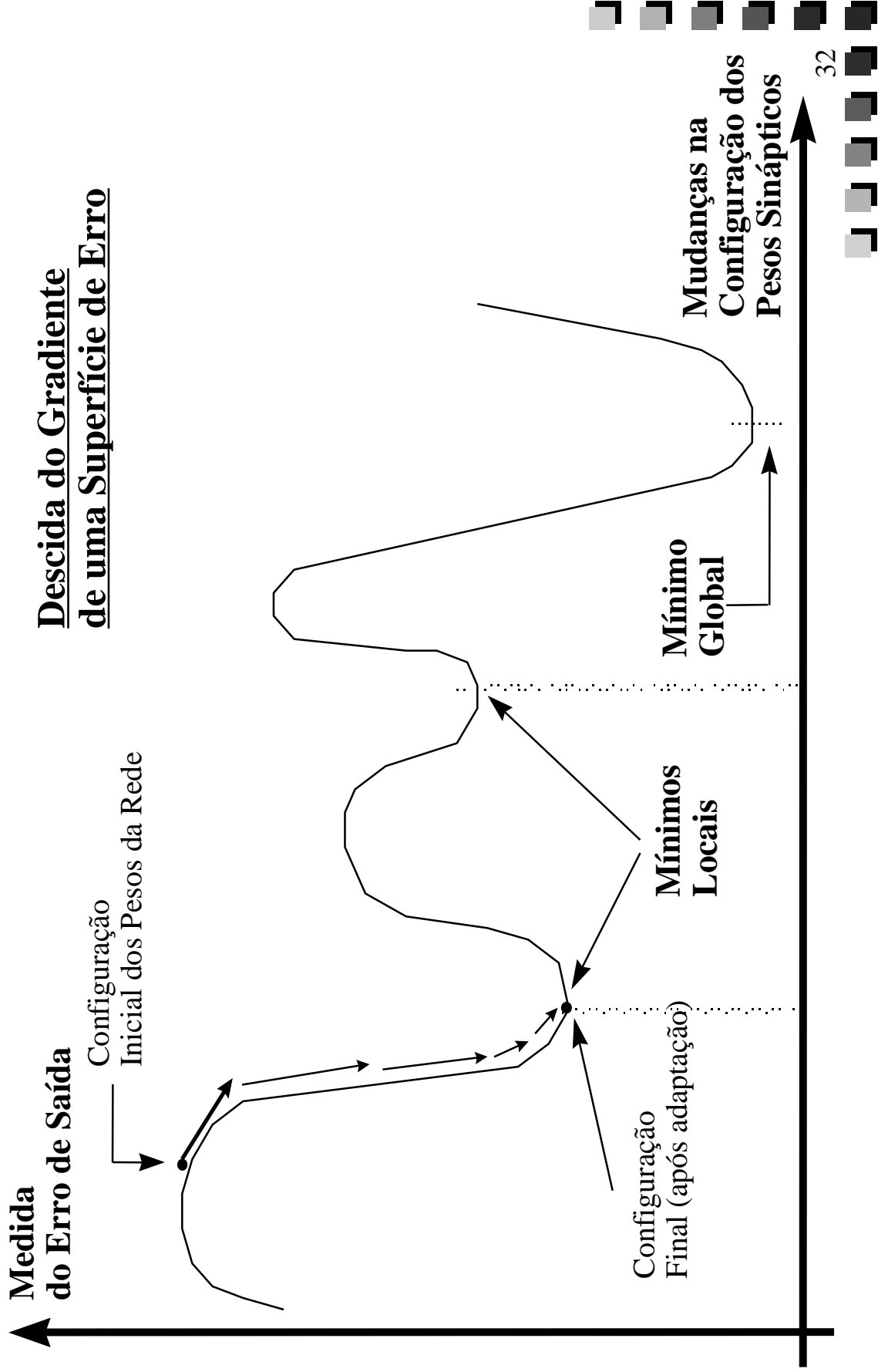
---

⇒ Em relação ao aprendizado:

- \* Aprendizado supervisionado
- \* Aprendizado semi-supervisionado (reinforcement learning)
- \* Aprendizado não supervisionado (self-organizing, clustering)
  
- \* Aprendizado instantâneo
- \* Aprendizado por pacotes
- \* Aprendizado contínuo
- \* Aprendizado ativo
  
- \* Aprendizado: aproximação de funções
- \* Aprendizado: classificação
  
- \* Usar apenas uma base de exemplos de aprendizado
- \* Usar uma base de aprendizado e uma base de teste de generalização

# Redes Neurais: Aprendizado Neural

APRENDIZADO = Adaptação das conexões (pesos sinápticos)





## Redes Neurais: Aprendizado Neural

---

**REGRA DELTA**: Perceptron [Rosenblatt] / Adaline [Widrow]

$$\text{Erro} = \text{SD} - \text{SN}$$

**Erro** = Erro estimado na saída de um neurônio

**SD** = Saída Desejada (valor desejado de saída do aprendizado supervisionado)

**SN** = Saída Rede (valor de saída que foi obtido pela ativação do neurônio)

$$\text{Peso\_Novo}(i) = \text{Peso\_Antigo}(i) + \beta * \text{Erro}(i) * \text{Entrada}(i)$$

-----  
| Entrada(i) |

**Peso\_Novo(i)** = Peso da entrada 'i' de um neurônio, após a adaptação

**Peso\_Antigo(i)** = Peso da entrada 'i' de um neurônio, antes da adaptação

**Entrada(i)** = Valor da entrada 'i' de um neurônio

**$\beta$**  = Fator de ajuste aplicado aos pesos (valor entre 0 e 1)

# Redes Neurais: Aprendizado Neural

---

- Método da Descida do Gradiente - Multi-Layer Perceptron (MLP) BackPropagation

Erro quadrático:  $E = \frac{1}{2} \sum_i (D_i - A_i)^2$       Ajuste dos Pesos:  $\Delta W_{ij} = -\alpha \frac{\partial E}{\partial W_{ij}}$

Derivação da regra de reajuste dos pesos  
(Neurônios da camada de saída - saída linear)

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial A_i} \frac{\partial A_i}{\partial W_{ij}} = \delta_i X_j$$

$$\frac{\partial E}{\partial A_i} = -(D_i - A_i) = \delta_i$$

$$\frac{\partial A_i}{\partial W_{ij}} = X_j$$

$$\Delta W_{ij} = -\alpha \cdot \delta_i \cdot X_j = \alpha \cdot (D_i - A_i) \cdot X_j$$

# Redes Neurais: Aprendizado Neural

---

- Método da Descida do Gradiente - Multi-Layer Perceptron (MLP) BackPropagation

**Erro quadrático:**  $E = \frac{1}{2} \sum_i (D_i - A_i)^2$       **Ajuste dos Pesos:**  $\Delta W_{ij} = -\alpha \frac{\partial E}{\partial W_{ij}}$

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial S_i} \frac{\partial S_i}{\partial A_i} \frac{\partial A_i}{\partial W_{ij}} = \delta_i X_j$$

$$\frac{\partial E}{\partial S_i} = \frac{\partial E}{\partial A_i} \frac{\partial A_i}{\partial S_i} = -(D_i - A_i) \cdot Fa'(S_i) = \delta_i$$

$$\frac{\partial E}{\partial A_i} = -(D_i - A_i)$$

$$\frac{\partial A_i}{\partial S_i} = Fa'(S_i)$$

$$\frac{\partial S_i}{\partial W_{ij}} = X_j$$

$$Fa(x) = \frac{1}{1 + e^{-x}} \quad \therefore \quad Fa'(x) = Fa(x) \cdot (1 - Fa(x))$$

$$Fa'(S_i) = Fa(S_i) \cdot (1 - Fa(S_i)) = A_i \cdot (1 - A_i)$$

$$Fa(x) = \tanh(x) \quad \therefore \quad Fa'(x) = (1 - Fa(x)) \cdot Fa(x)$$

$$Fa'(S_i) = (1 - Fa(S_i)) \cdot Fa(S_i) = (1 - A_i) \cdot A_i$$

$$\Delta W_{ij} = -\alpha \cdot \delta_i \cdot X_j = \alpha \cdot (D_i - A_i) \cdot Fa'(S_i) \cdot X_j$$

# Redes Neurais: Aprendizado Neural

---

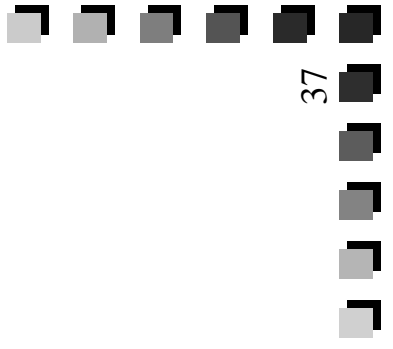
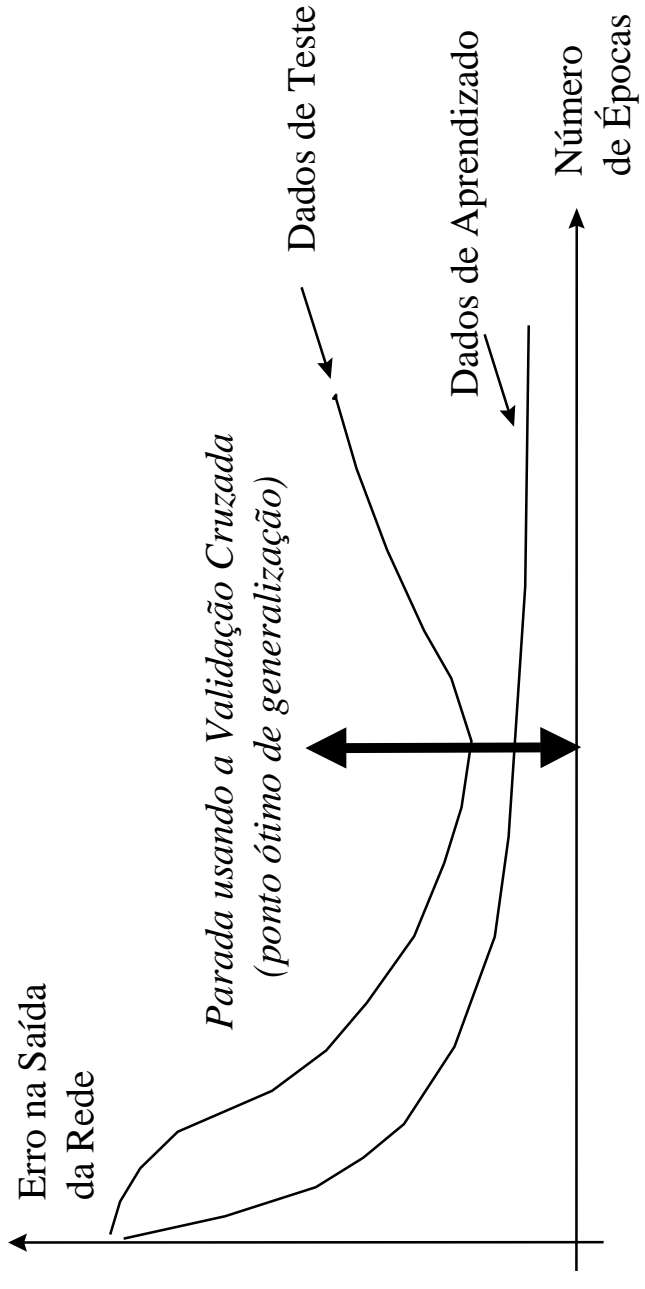
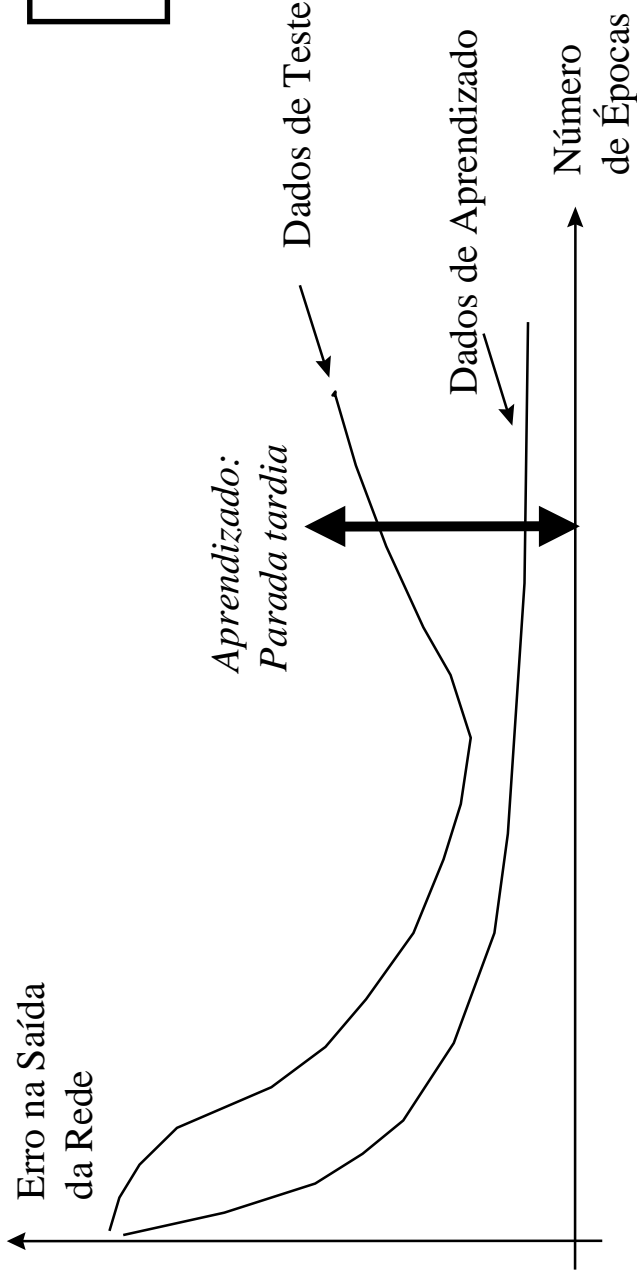
- Método da Descida do Gradiente - Multi-Layer Perceptron (MLP) BackPropagation

**Erro quadrático:**  $E = \frac{1}{2} \sum_i (D_i - A_i)^2$       **Ajuste dos Pesos:**  $\Delta W_{ij} = -\alpha \frac{\partial E}{\partial W_{ij}}$

**Derivação da regra de reajuste dos pesos**  
(Camada intermediária da rede = Hidden Layer)

$$\begin{aligned} \frac{\partial E}{\partial W_{ij}} &= \frac{\partial E}{\partial S_i} \frac{\partial S_i}{\partial A_i} \frac{\partial A_i}{\partial W_{ij}} = \delta_i X_j \\ \frac{\partial E}{\partial S_i} &= \frac{\partial E}{\partial A_i} \frac{\partial A_i}{\partial S_i} = -Fa'(S_i) \cdot \sum_k \delta_k W_{ki} = \delta_i \\ \frac{\partial E}{\partial A_i} &= \sum_k \frac{\partial E}{\partial S_k} \frac{\partial S_k}{\partial A_i} = \sum_k \frac{\partial E}{\partial S_k} \frac{\partial}{\partial A_i} \sum_h W_{hk} A_h = \dots \\ &= \sum_k \frac{\partial E}{\partial S_k} W_{ki} = - \sum_k \delta_k W_{ki} \\ \frac{\partial A_i}{\partial S_i} &= Fa'(S_i) \\ \frac{\partial S_i}{\partial W_{ij}} &= X_j \\ \Delta W_{ij} &= -\alpha \cdot \delta_i \cdot X_j = \alpha \cdot X_j \cdot Fa'(S_i) \cdot \sum_k \delta_k W_{ki} \end{aligned}$$

**APRENDIZADO:  
GENERALIZAÇÃO**



## Redes Neurais Artificiais: Aplicações Práticas

---

- **Sistemas de auxílio ao Diagnóstico: Médico, Falhas de Sistemas, etc;**
- **Previsão de Séries Temporais: Cotações da Bolsa de Valores, Dados Econômicos, Consumo de Energia Elétrica, Meteorologia, etc;**
- **Processamento de Linguagem Natural - PLN (Textos e Web);**
- **Data Mining & KDD (Knowledge Data Discovery);**
- **Robótica Inteligente;**
- **Sistemas de Controle e Automação;**
- **Reconhecimento e Síntese de Voz;**
- **Processamento de Sinais: Radar, Sensores, etc.**

**UCI-ML - University of California Irvine - Machine Learning Repository**

*<http://www.ics.uci.edu/~mllearn/MLRepository.html>*

# **Redes Neurais Artificiais: Aplicações Práticas**

---

## **PROJETO HMLT**

### **APLICAÇÕES DAS REDES NEURAIS**

**EM**

**\* PROCESSAMENTO DE IMAGENS**

**\* ROBÓTICA AUTÔNOMA**

**\* DATA MINING**



# Conclusão e Perspectivas

---

- \* **Importância do Aprendizado nos Sistemas Inteligentes**
- \* **Aprendizado Neural: propriedades interessantes...  
Robustez, Paralelismo, Generalização, Entradas/Saídas Quantitativas**
- \* **Processamento de Imagens: área de estudos de grande interesse para a I.A.  
Visão Artificial - Desafio**
- \* **Processamento de Imagens usando Sistemas Inteligentes:  
Redes Neurais - Ferramenta bastante adequada para se tratar imagens**
- \* **Redes Neurais aplicadas ao processamento de imagens:  
Tratamento de Imagens / Reconhecimento e Classificação de Imagens**
- \* **Resultados são bastante promissores**
- \* **Tendência: *Sistemas Híbridos*  
Integrar => Pré-Processamento, Pós-Processamento  
Cooperação - Multi-agente, Modular  
Tarefas complexas e de alto nível**

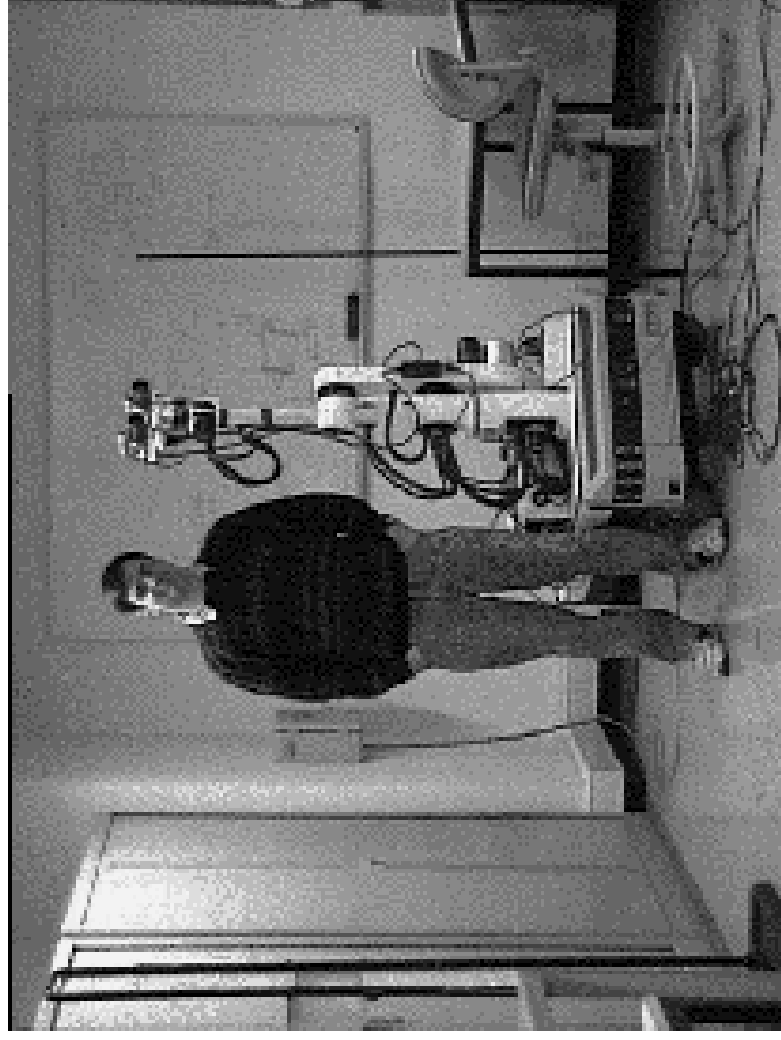


# Inteligência Artificial: Perspectivas

---

*Hybrid Machine Learning Tools*

<http://www.inf.unisinos.br/~osorio/>



Grupo de Inteligência Artificial @



UNISINOS