

**PIP/CA - Programa Interdisciplinar de Pós-Graduação
Mestrado em Computação Aplicada da UNISINOS**

2000/2 - 3o. Trimestre - AULA 02 / FSO

**SISTEMAS
ADAPTATIVOS
INTELIGENTES**

• Professor Responsável:

Prof. Dr. Fernando Osório

E-Mail: osorio@exatas.unisinos.br

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

TEMAS DE ESTUDO: REDES NEURAIS ARTIFICIAIS

Histórico, Princípios Básicos e Características

- Psychon [McCulloch & Pitts 43]
- Adaline, Madaline, Perceptron [Widrow 62, Rosenblatt 59]
- Multi-Layer Perceptron e Back-Propagation [Rumelhart 86]
- Outros modelos: Hopfield, Kohonen, ART, BAM, BSB, RBF, ...

Modelos de Redes Neurais

- Características e Classificação
- Topologia
- Aprendizado e Generalização
- Vantagens e limitações do uso das redes neurais artificiais

Aplicações das RNA (ANN=Artificial Neural Nets)

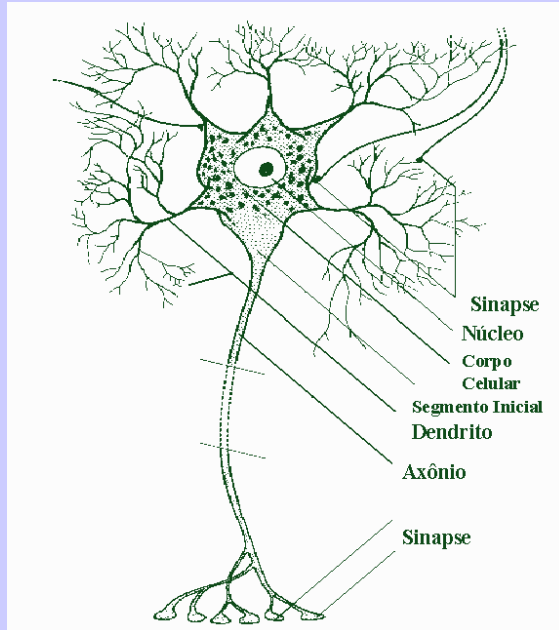
- Aproximação de Funções
- Classificação
- Exemplos: previsão, séries temporais, diagnóstico, etc

MLP - BP = Multi-Layer Perceptron com Back-Propagation

- Descrição do modelo
- Algoritmo de Aprendizado
- Características e Propriedades

REDES NEURAIS:

Histórico e Princípios



Redes Neurais Artificiais:

Neurônio...
Modelo Simulado
Modelo SIMPLIFICADO

Características Básicas:

Adaptação
Aprendizado
Autômato

Representação de Conhecimentos:

Baseada em Conexões

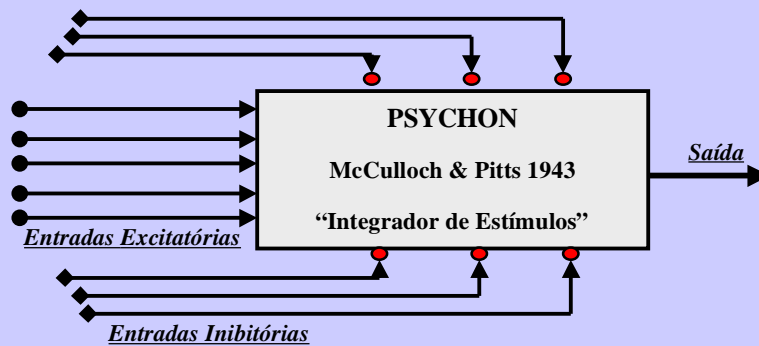
SIMULAR REDES NEURAIS

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Psychon** - McCulloch & Pitts 1943
Dispositivo Lógico de 2 estados - Modelo de conexões não adaptáveis
Não previa o aprendizado!



- **Regra de Hebb** - 1949 (Neurofisiologia => Aprendizado)
Livro “The Organization of Behavior” - Plasticidade e adaptação
“Dois neurônios conectados que são ativados ao mesmo tempo devem fortalecer esta conexão, e o comportamento inverso (ativação isolada, enfraquecer a conexão) também deve ser considerado” - Adaptação da força das conexões

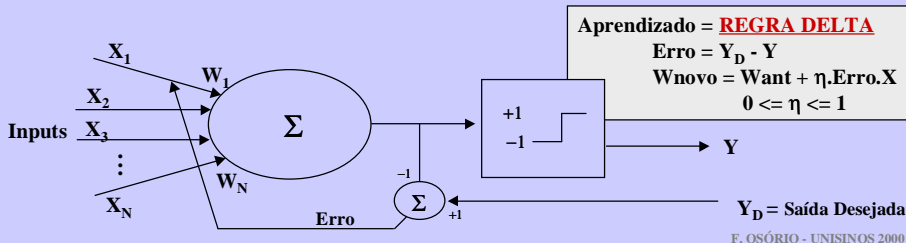
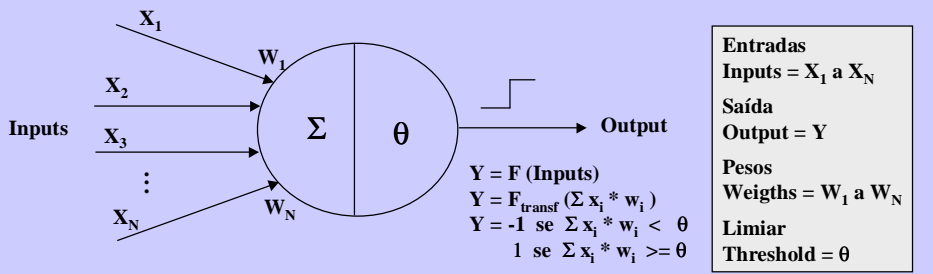
F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Adaline** - B. Widrow & M. Hoff 1960 - **Adaptive Linear Element (Linear Neuron)**
Elementos com saídas binárias {0,1} ou {-1,1}

Conexões com adaptação - Aprendizado - *Correção do Erro de Saída*



F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Adaline** - B. Widrow & M. Hoff 1960 - **Adaptive Linear Element**
Exemplo de uso...

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	Pneumonia
S	N	S	S	S	N	N	S	Pneumonia
S	N	S	N	S	N	S	N	Saudável
S	N	N	S	S	N	S	N	Saudável

Realizar o diagnóstico baseado nos conhecimentos prévios

Consultas ao sistema:

S,N,S,N,S,N,S,N => Diagnóstico ?

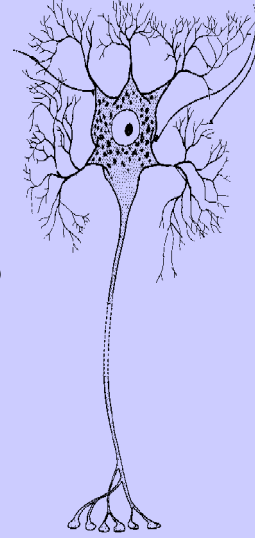
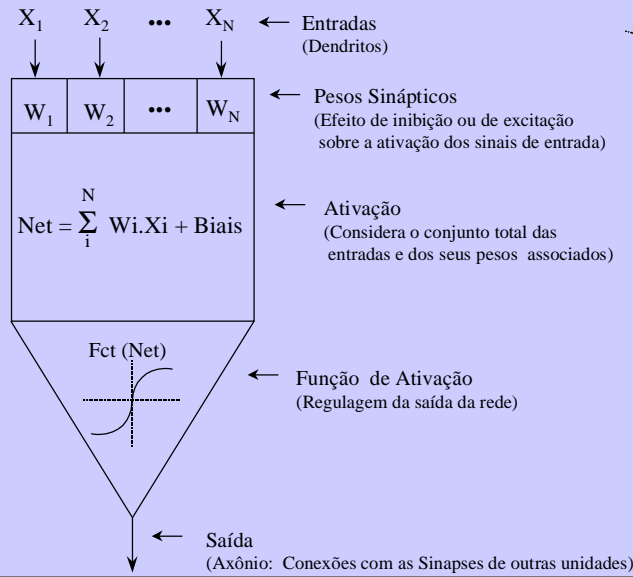
S,N,?, ?,S,N,S,N => Diagnóstico ?

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **PERCEPTRON** - Frank Rosenblatt 1958-1962 (“Principles of Neurodynamics”)
Modelo básico de neurônio adotado por muitos dos modelos atuais!

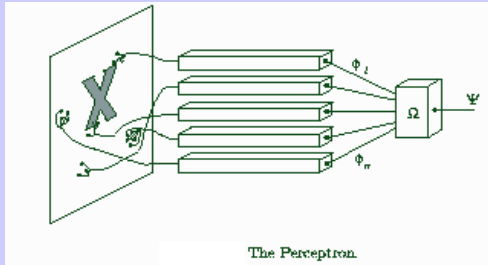


F. OSÓRIO - UNISINOS 2000

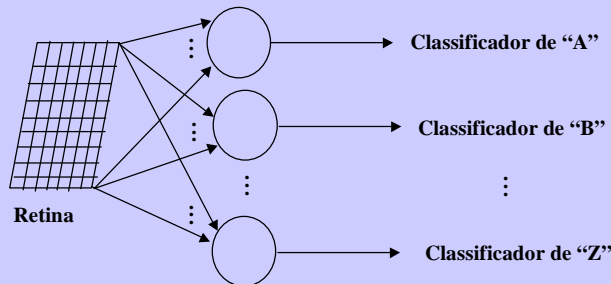
REDES NEURAIS:

Histórico e Princípios

- **Perceptron** - Exemplos de uso:



Reconhecimento de Padrões em Imagens



Saída Binária
x
Contínua

Individual
x
Competição
(winner take all)

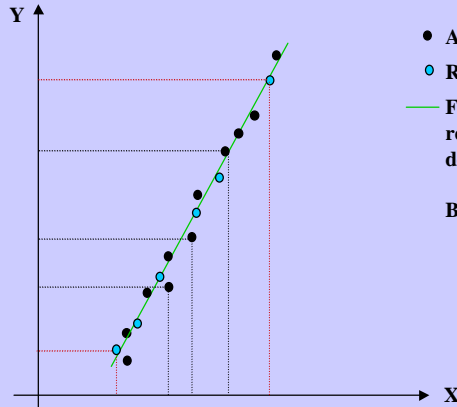
F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

• **Perceptron** - Exemplos de uso:

Aprendizado de uma função - aproximação / regressão / interpolação



- **Aprendizado** : Entra X, Saida desejada Y
- **Resposta da rede**: Entra X, Sai Y
- **Função Y=F(X)** “desconhecida” representada através de exemplos de pares de coord. X,Y

Base de Aprendizado: $Y = 1.2 * X$

X	Y
1.0	1.2
1.5	1.8
2.0	2.4
2.5	3.0
3.0	3.6

Saída com valores numéricos (variável quantitativa)

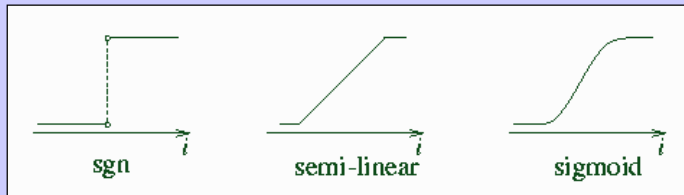
F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

• **Perceptron** - Funções de Transferência: “normaliza saída”

Objetivo - adaptar o resultado da soma ponderada em um intervalo de saída adotado



$F(X) = \text{Sgn}(X)$
If $X \geq 0$
Then $Y = 1$
Else $Y = 0$ (ou -1)
ou
If $X \geq \text{Limiar}$
Then $Y = 1$
Else $Y = 0$ (ou -1)

$F(X) = \text{Linear}(X, \text{Min}, \text{Max})$
If $X < \text{Min}$
Then $Y = 0$
If $X \geq \text{Min}$ and $X \leq \text{Max}$
Then $Y = X$
If $X > \text{Max}$
Then $Y = 1$
Obs.: $Y = X$ ou $Y = \text{Normaliza}(X)$

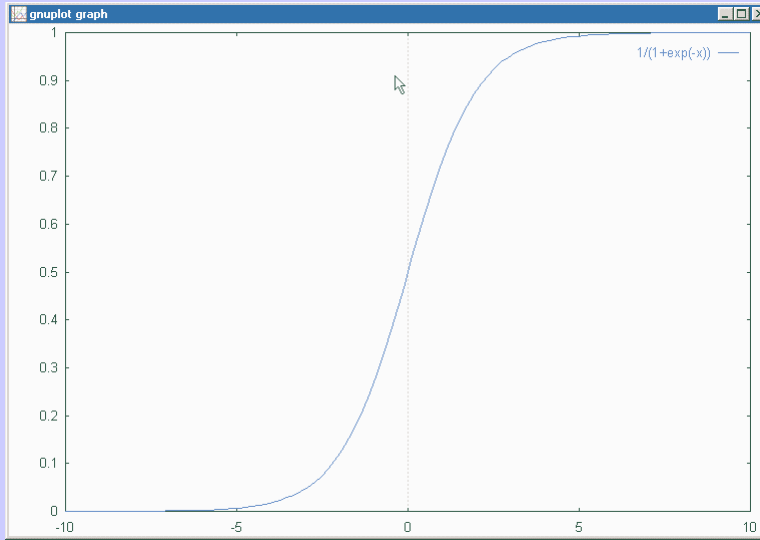
$F(X) = \text{Sigmoid}(X)$
Assimétrica
 $Y = \frac{1}{1 + \text{Exp}(-x)}$
Simétrica
 $Y = \text{TanHip}(X)$

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Perceptron** - Funções de Transferência: “normaliza saída”
Sigmoid assimétrica

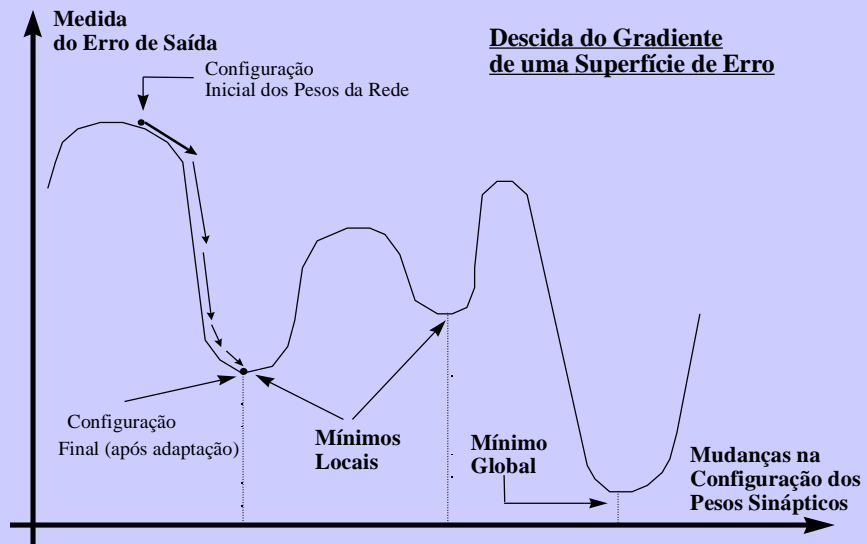


F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Perceptron** - Aprendizado = Adaptação das conexões (pesos sinápticos)



F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Perceptron** - Aprendizado = Adaptação das conexões (pesos sinápticos)

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)

$$\text{Peso_Novo}(i) = \text{Peso_Antigo}(i) + \frac{\beta * \text{Erro}(i) * \text{Entrada}(i)}{|\text{Entrada}(i)|}$$

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

β = Fator de ajuste aplicado aos pesos (valor entre 0 e 1)

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Perceptron** - Aprendizado = Adaptação das conexões (pesos sinápticos)

- Método da Descida do Gradiente

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

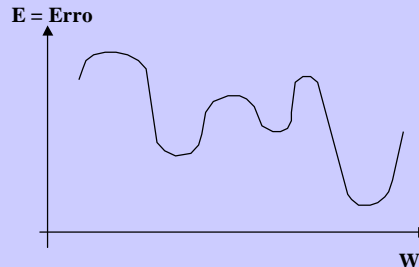
Derivação da regra de reajuste dos pesos (saída linear):

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

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

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

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



Dada uma entrada X_i e o peso W_i , o neurônio fornece uma saída A_i . Queremos treiná-lo para responder D_i

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

• **Perceptron** - Aprendizado = Adaptação das conexões (pesos sinápticos)

- Método da Descida do Gradiente
- Derivação da regra de reajuste dos pesos: Neurônios usando a sigmoíde

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

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

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

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

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

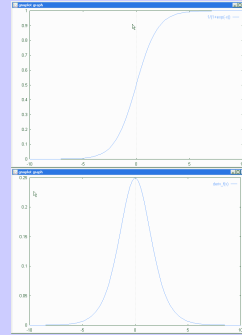
$$Fa(x) = \frac{1}{1 + e^{-x}} \quad \therefore \quad Fa'(x) = Fa(x) \cdot (1 - Fa(x))$$

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

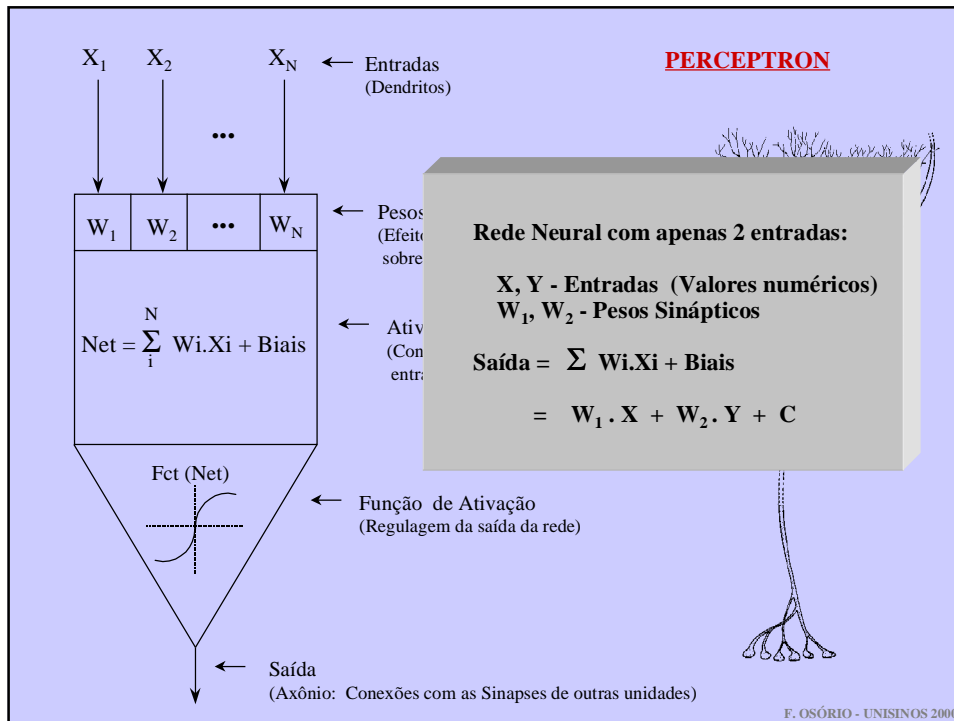
$$Fa'(x) = (1 - Fa(x)) \cdot Fa(x)$$

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

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



F. OSÓRIO - UNISINOS 2000



PERCEPTRON

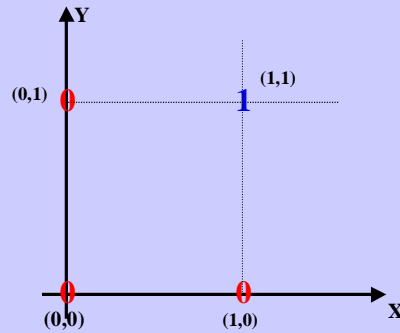
Rede Neural com apenas 2 entradas:

X, Y - Entradas (Valores numéricos)
 W_1, W_2 - Pesos Sinápticos

$$\begin{aligned} \text{Saída} &= \sum W_i \cdot X_i + \text{Biais} \\ &= W_1 \cdot X + W_2 \cdot Y + C \end{aligned}$$

CLASSIFICAÇÃO

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



Representação Geométrica do Problema

F. OSÓRIO - UNISINOS 2000

PERCEPTRON

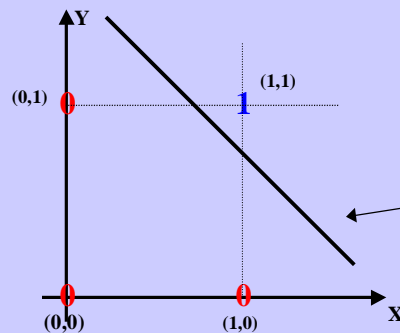
Rede Neural com apenas 2 entradas:

X, Y - Entradas (Valores numéricos)
 W_1, W_2 - Pesos Sinápticos

$$\begin{aligned} \text{Saída} &= \sum W_i \cdot X_i + \text{Biais} \\ &= W_1 \cdot X + W_2 \cdot Y + C \end{aligned}$$

CLASSIFICAÇÃO

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



Representação Geométrica do Problema

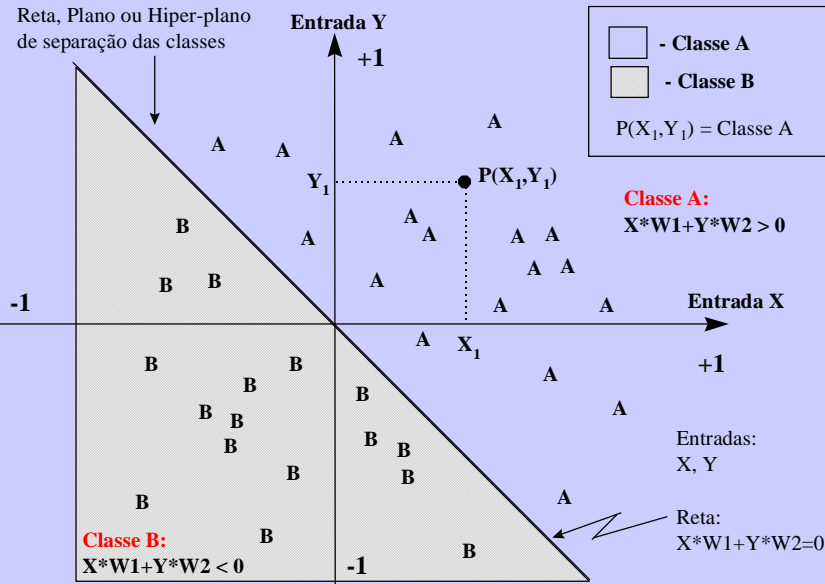
Como classificar?
Separar as classes

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Perceptron** - Aprendizado = Adaptação dos (hiper)planos de separação



F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

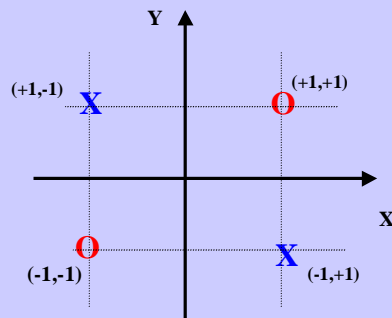
Histórico e Princípios

- **Perceptron** - **Problema do Aprendizado do XOR**
Minsky & Papert 1969 ("Perceptrons") - Problema não linearmente separável!

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

← Problema do XOR ———→
 OU Exclusivo

X	Y	XOR
-1	-1	0
-1	+1	X
+1	-1	X
+1	+1	0



Sistema de Equações:

$$A.X + B.Y = S$$

$$-1.X + -1.Y = -1$$

$$-1.X + +1.Y = +1$$

$$+1.X + -1.Y = +1$$

$$+1.X + +1.Y = -1$$

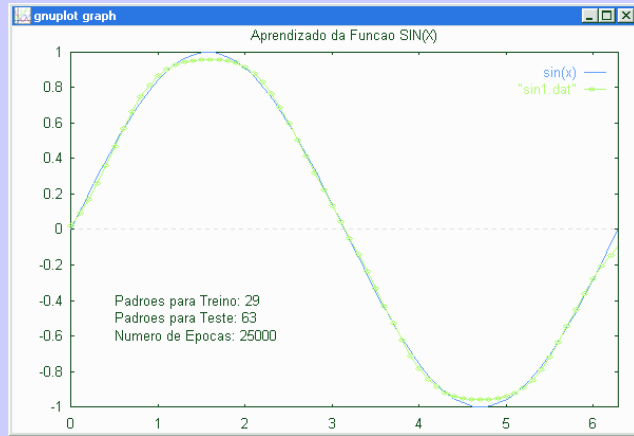
Sem solução!!!

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

- **Perceptron - Aprendizado:**
Aprendizado de uma função não linear - aproximação / regressão



0.0	0.000000
0.3	0.147760
0.6	0.282321
0.9	0.391663
1.05	0.433712
1.2	0.466020
1.35	0.487862
1.5	0.498747
1.65	0.498433
1.8	0.486924
1.95	0.464480
2.1	0.431605
2.4	0.337732
2.7	0.213690
3.0	0.070560
3.3	-0.078873
3.6	-0.221260
3.9	-0.343883
4.2	-0.435788
4.35	-0.467526
4.5	-0.488765
4.65	-0.499027
4.8	-0.498082
4.95	-0.485952
5.1	-0.462907
5.25	-0.429470
5.4	-0.386382
5.7	-0.275343
6.0	-0.139708



REDES NEURAIS:

Histórico e Princípios

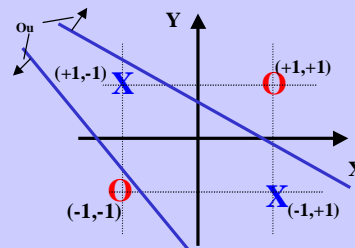
- **Perceptron:**
- Quais as soluções existentes para o aprendizado de problemas não lineares?

- **MADALINE - Many Adaline - Bernard Widrow**

* Combinando múltiplos Adalines

* Modos de combinação:

- And
- Or
- Majority



* Problemas: falta de um método automático de aprendizado e combinação dos múltiplos Adalines

- **Redes Multi-Nível:** “*década perdida*”

* Mais de 10 anos buscando uma solução... 1969 (Minsky) => 1986 (Rumelhart)

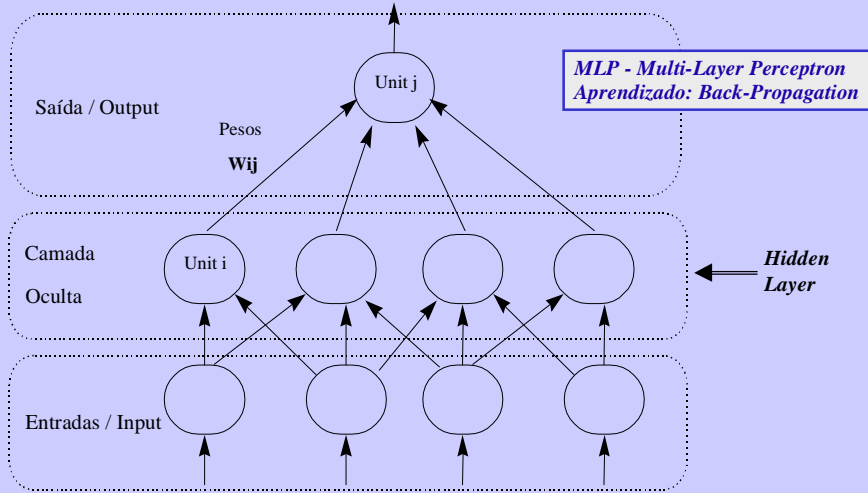
* Problema enfrentado:

- Cálculo do erro na saída: $S_{desejada} - S_{obtida} = \text{Erro}$
- Cálculo do erro de um neurônio interno da rede:
Como fazer a atribuição da “culpa” em relação ao erro final na saída ???

REDES NEURAIS:

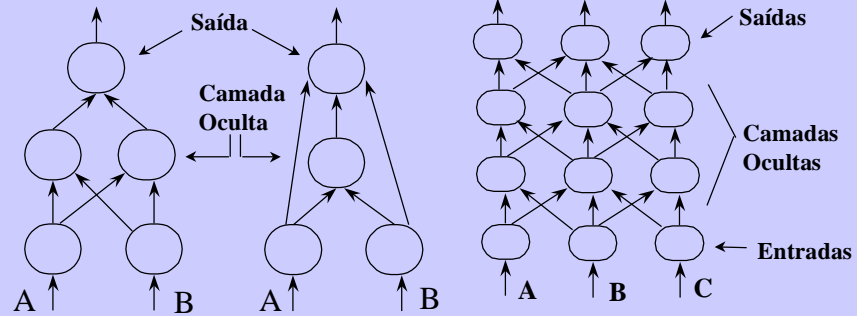
Histórico e Princípios

- **Redes Multi-Nível:** Aprendizado de problemas - Classes não linearmente separáveis
Rumelhart, Hinton & Williams 1986 - Livro "PDP - Parallel Distributed Processing"



F. OSÓRIO - UNISINOS 2000

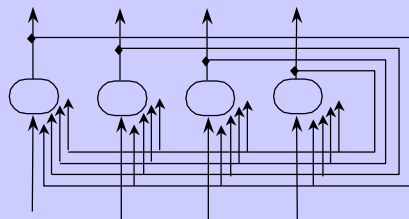
REDES NEURAIS: *Redes Multi-Nível - Anos 80 surgem vários modelos e arquiteturas*



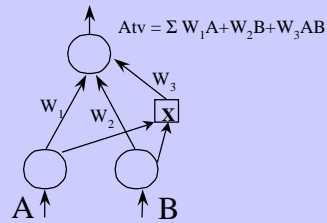
(a) Rede de três camadas

(b) Rede com atalhos

(c) Rede com múltiplas camadas



(d) Rede recorrente



(e) Rede de ordem superior

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Histórico e Princípios

• **Redes Multi-Nível: Multi-Layer Perceptron com Back-Propagation**

• 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*

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

* Cálculo do erro de um neurônio: (hidden)

Como fazer a atribuição da “culpa” em relação ao erro final na saída ???

Resposta:

- A culpa (erro) de um neurônio é proporcional a sua influência no erro da camada seguinte, ou seja, depende do peso da conexão entre a saída deste neurônio e a entrada na camada seguinte.

- O processo de ajuste dos pesos passa por uma etapa *forward* de ativação da rede e uma etapa *backward* de propagação do erro para as camadas superiores (partindo da saída em direção a camada de entrada)

REDES NEURAIS:

Histórico e Princípios

• **Redes Multi-Nível: Multi-Layer Perceptron com Back-Propagation**

• 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*

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

Camada intermediária (oculta)

$$\begin{aligned} \frac{\partial E}{\partial W_{ij}} &= \frac{\partial E}{\partial S_i} \frac{\partial S_i}{\partial W_{ij}} = \delta_i X_j \\ \frac{\partial E}{\partial S_i} &= \frac{\partial E}{\partial A_i} \frac{\partial A_i}{\partial S_i} = -(D_i - A_i) \cdot Fa'(S_i) = \delta_i \\ \frac{\partial E}{\partial A_i} &= -(D_i - A_i) \\ \frac{\partial A_i}{\partial S_i} &= Fa'(S_i) \\ \frac{\partial S_i}{\partial W_{ij}} &= X_j \end{aligned}$$

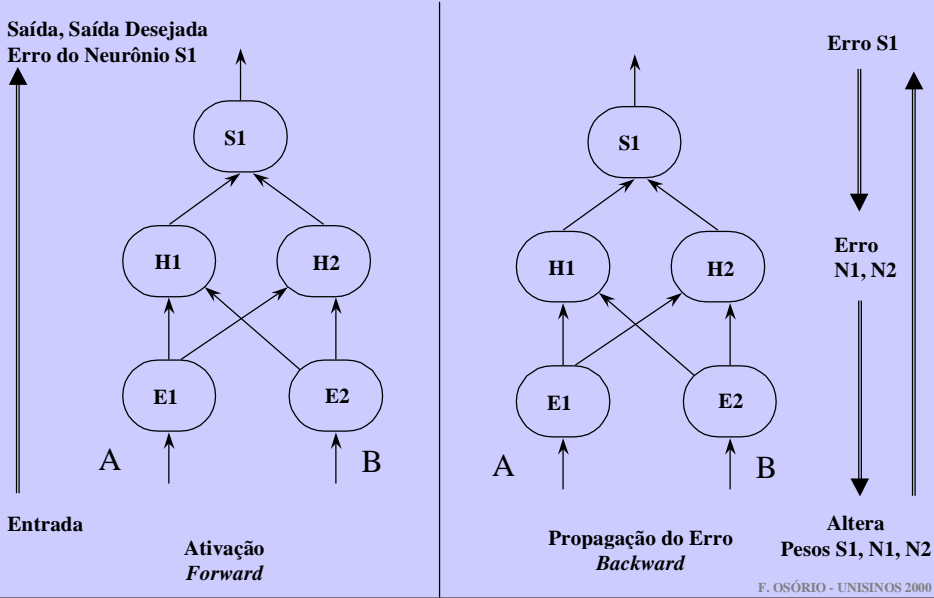
$$\Delta W_{ij} = -\alpha \cdot \delta_i \cdot X_j = \alpha \cdot (D_i - A_i) \cdot Fa'(S_i) \cdot X_j$$

Camada de saída com sigmoide

REDES NEURAIS:

Histórico e Princípios

- **Redes Multi-Nível:** *Multi-Layer Perceptron com Back-Propagation*



REDES NEURAIS:

Histórico e Princípios

- **Resumo da evolução das Redes Neurais Artificiais:** [Widrow & Lehr; Braga; Kovacs]

- 1943 - Psychon - McCulloch & Pits (Excitação / Inibição - Sem aprendizado)
- 1949 - Regra de Hebb - D. O. Hebb (Aprendizado / Adaptação)
- 1960 - Adaline - Widrow & Hoff (Valores discretos / binários / Regra Delta)
- 1962 - Perceptron - Frank Rosenblatt (Valores contínuos)
- 1962 - Madaline - Bernard Widrow (Combinando Adalines / Combinação manual)
- 1969 - Problema do XOR - Minsky & Papert (Livro "Perceptrons")
- 1970 - 1980 => Década perdida..
- 1982 - Modelo de Hopfield (Redes recorrentes - Memórias Auto-Associativas)
- 1982 - Modelo de Kohonen - SOFM (Redes recorrentes - Clustering)
- 1983 - Modelo ART - Carpenter & Grossberg (Cria protótipos - não supervisionado)
- 1986 - MLP Back-Propagation - Rumelhart, Hinton & Williams (Multi-nível)
- 1980-1990 => Década das aplicações
 - Nettalk, Alvin, Dupla espiral, Jogos, Robótica, Visão,
 - Reconhecimento de Imagens e Padrões (OCR, Digitais, Assinaturas),
 - Reconhecimento de Voz (Comandos e Fonemas)
 - Previsão de séries temporais (Ações, tempo, consumo, etc).
- 1990-2000 => Revendo conceitos e limitações. Propondo novos modelos (ver FAQ).

FAQ - <http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/top.html>

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Características e Classificação das Redes

• **Classificando as Redes Neurais Artificiais:**

⇒ Em relação as unidades da rede: conhecimento local / distribuído, função de transf., ...

- * Redes baseadas em Perceptrons (MLP - *Multi-Layer Perceptron*)
- * Redes baseadas em Protótipos (RBF - *Radial Basis Function*, ART, SOFM Kohonen)

⇒ Em relação a estrutura da rede (topologia):

- * Redes de uma única camada
- * Redes de múltiplas camadas
- * Redes modulares

- * Redes do tipo uni-direcional (*Feed-Forward na ativação*)
- * Redes do tipo recorrentes (*Feed-Back na ativação*)

- * Redes com estrutura estática (não altera sua estrutura)
- * Redes com estrutura dinâmica (altera a estrutura - cresce / diminui)

- * Redes com atalhos (*short-cuts*)
- * Redes com conexões de ordem superior (*sigma-pi*)

⇒ Em relação a saída de rede / tipo de aplicação: valores *binários, discretos e contínuos*

- * Redes binárias - entradas e saídas binárias
- * Redes com saídas discretas (valores binários) usadas na classificação
- * Redes com saídas contínuas (valores numéricos) usadas na aproximação de funções

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Características e Classificação das Redes

• **Aprendizado nas Redes Neurais Artificiais:**

⇒ Em relação ao aprendizado:

- * **Aprendizado supervisionado** (correção de erros)
- * Aprendizado semi-supervisionado (*reinforcement learning*)
- * Aprendizado não supervisionado (*self-organizing, clustering, competitive*)

- * Aprendizado instantâneo (*one shot learning*)
- * Aprendizado por pacotes (*batch learning*)
- * Aprendizado contínuo (*on-line learning*)
- * Aprendizado ativo (*interactive*)

- * Aprendizado: tarefa de aproximação de funções
- * Aprendizado: tarefa de classificação
 - Pertence a classe (sim/não)
 - Qual é a classe que pertence (A, B, C, D - nro. de classes limitado)
 - Qual é a classe que pertence (A, B, C, D, ... - nenhuma destas)
 - Qual é a classe que pertence (uma única, múltiplas classes)
- * Aprendizado: tarefa envolvendo aspectos temporais e contexto
- * Aprendizado: memória associativa (auto-associativa, bidirecional)

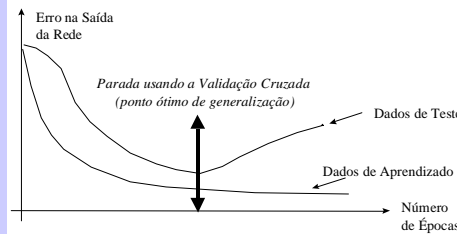
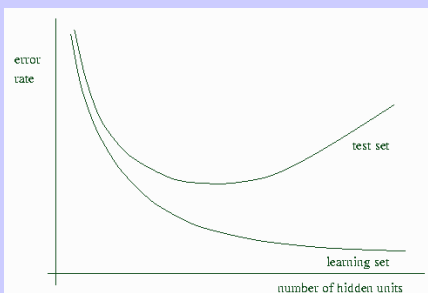
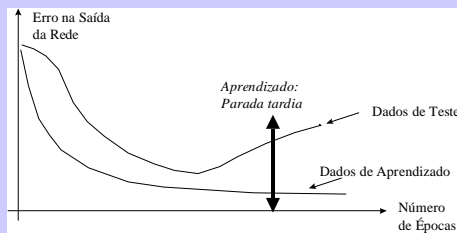
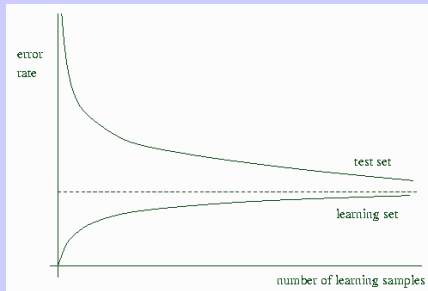
- * Uso apenas uma base de exemplos de aprendizado (decorar / generalizar)
- * Uso de uma base de aprendizado e uma base de teste de generalização

F. OSÓRIO - UNISINOS 2000

REDES NEURAIAS:

Características e Classificação das Redes

Aprendizado nas Redes Neurais Artificiais: Curvas de Aprendizado



F. OSÓRIO - UNISINOS 2000

REDES NEURAIAS:

Características e Classificação das Redes

Vantagens e Limitações das Redes Neurais Artificiais:

- Aplicações de Machine Learning e Sistemas Adaptativos;
- Aplicadas em tarefas onde temos bases de exemplos disponíveis sobre um determinado problema, realizando a aquisição automática de conhecimentos;
- Associação de padrões de entradas e saída;
- Classificação de padrões de forma supervisionada ou não;
- Aproximação de funções desconhecidas através de amostras destas funções;
- Trabalhar com dados aproximados, incompletos e inexatos;
- Paralelismo, generalização, robustez;
- “Tarefas complexas realizadas por seres humanos”;

Limitações:

- Trabalhar com conhecimentos simbólicos de alto nível;
- Composição e construção de conhecimentos estruturados;
- Dificuldade de explicitação dos conhecimentos adquiridos;
- Dificuldade para definir a estrutura da rede, seus parâmetros e a base de dados;
- Falta de garantia de uma convergência do algoritmo para uma solução ótima;

F. OSÓRIO - UNISINOS 2000

REDES NEURAIS:

Aplicações

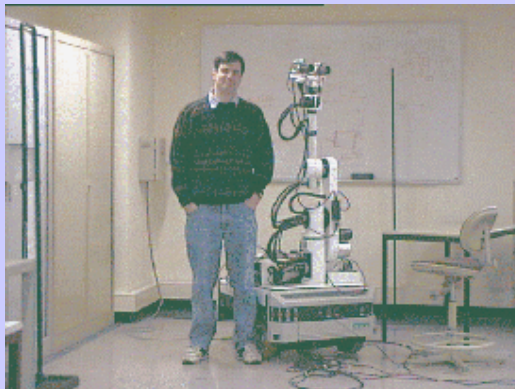
- Sistemas de auxílio ao Diagnóstico: Médico, Falhas de Sistemas, etc;
- Previsão de Séries Temporais: Cotações da Bolsa de Valores, Dados Econômicos, Consumo de Energia Elétrica, Meteorologia, etc;
- Processamento de Linguagem Natural - PLN (Textos e Web);
- Data Mining & KDD (Knowledge Data Discovery);
- Robótica Inteligente;
- Sistemas de Controle e Automação;
- Reconhecimento e Síntese de Voz;
- Processamento de Sinais e Imagens: Radar, Sensores, Imagens de satélite, etc.
- Clássicos: Nettek, Alvin, Dupla espiral, Reconhecimento de faces de pessoas, Reconhecimento de Imagens e Padrões (OCR, Digitais, Assinaturas), Reconhecimento de Voz (Pessoa, Comandos e Fonemas) Previsão de séries temporais (Ações, meteorologia, consumo, etc).

UCI-ML - University of California Irvine - Machine Learning Repository
<http://www.ics.uci.edu/~mlern/MLRepository.html>

F. OSÓRIO - UNISINOS 2000

Redes Neurais Artificiais

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



Grupo de Inteligência Artificial @



F. OSÓRIO - UNISINOS 2000

TEMAS DE PESQUISA SOBRE REDES NEURAIS ARTIFICIAIS:

*** PAPERS / DOCUMENTAÇÃO:**

- **FAQ:** [Http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/top.html](http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/top.html)
[Http://www.faqs.org/faqs/ai-faq/neural-nets/](http://www.faqs.org/faqs/ai-faq/neural-nets/)
[Ftp://ftp.sas.com/pub/neural/FAQ.html](ftp://ftp.sas.com/pub/neural/FAQ.html)
- **Osorio - Neural:** [Http://www.inf.unisinos.br/~osorio/neural.html](http://www.inf.unisinos.br/~osorio/neural.html)
- **Livro On-Line:** [Http://www.inf.unisinos.br/~osorio/neural/Neuro-book.html](http://www.inf.unisinos.br/~osorio/neural/Neuro-book.html)
- **UCI-ML:** <http://www.ics.uci.edu/~mlern/MLRepository.html>
- **Neuroprose**

*** ASSOCIAÇÃO:**

- **Connectionist List**
- **Comp.ai.neural-nets**

• SOFTWARES:

- **PDP++** <http://www.cnbc.cmu.edu/PDP++/PDP++.html>
- **SNNS** <http://www-ra.informatik.uni-tuebingen.de/SNNS/>
- **NevProp** Nevada Propagation Software
- **INSS** Contactar Osório...
- **Outros** <http://www.inf.unisinos.br/~osorio/neural/software.html>