



Universidade de Brasília

Faculdade UnB Gama



Sistemas Inteligentes

*Programa de Pós-Graduação em Engenharia
Biomédica*

Sistemas Inteligentes



Redes Neurais Artificiais (RNA)

Profa. Dra. Lourdes Mattos Brasil
lmbrasil@unb.br



Sumário

- ◆ Introdução
- ◆ Histórico
- ◆ Morfologia do Neurônio Biológico
- ◆ Arquiteturas das RNA
 - ◆ Modelos de Neurônios
 - ◆ Topologia
 - ◆ Regras de aprendizado
- ◆ Paradigmas de RNA
- ◆ Aprendizagem
- ◆ Aplicações
- ◆ Conclusões



Introdução

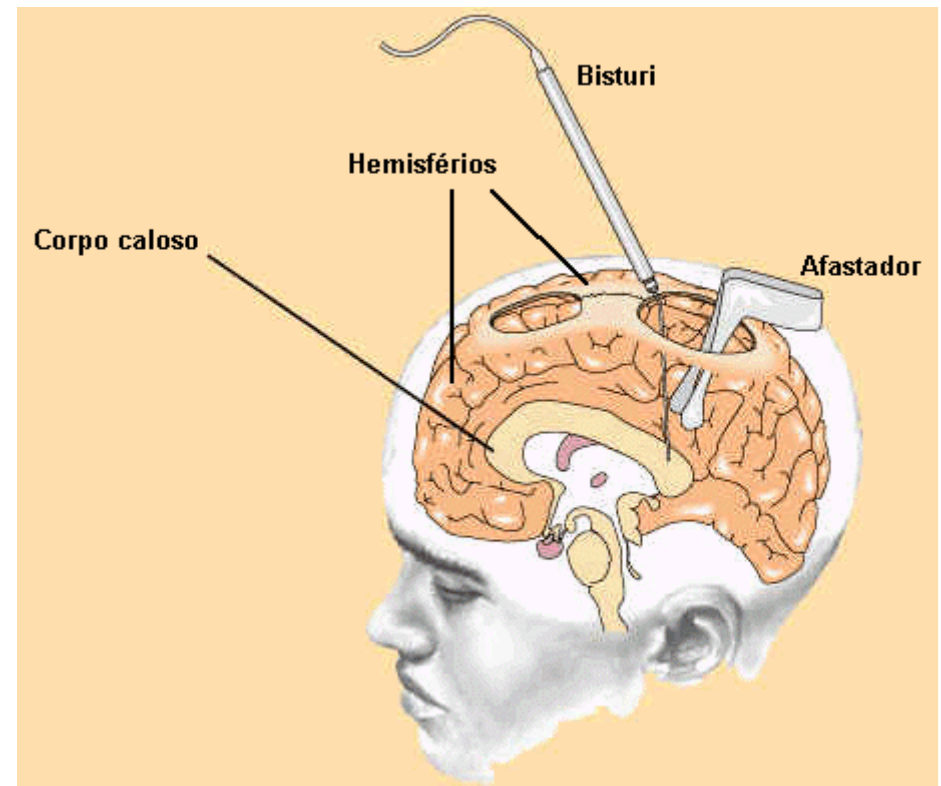
Situação atual

O estímulo inicial que conduziu ao desenvolvimento de RNA foi de entender detalhadamente o funcionamento do cérebro.

O objetivo era construir mecanismos que operassem de modo similar (tomar decisões, processar informações, aprender, lembrar e otimizar) ao cérebro humano.

Este objetivo ainda está longe de ser atingido.

No entanto, continua elevado o interesse na formalização e aplicação de arquiteturas de RNA.





Universidade de Brasília

Faculdade UnB Gama



Sistemas Inteligentes

*Programa de Pós-Graduação em Engenharia
Biomédica*

Introdução

Redes Neurais

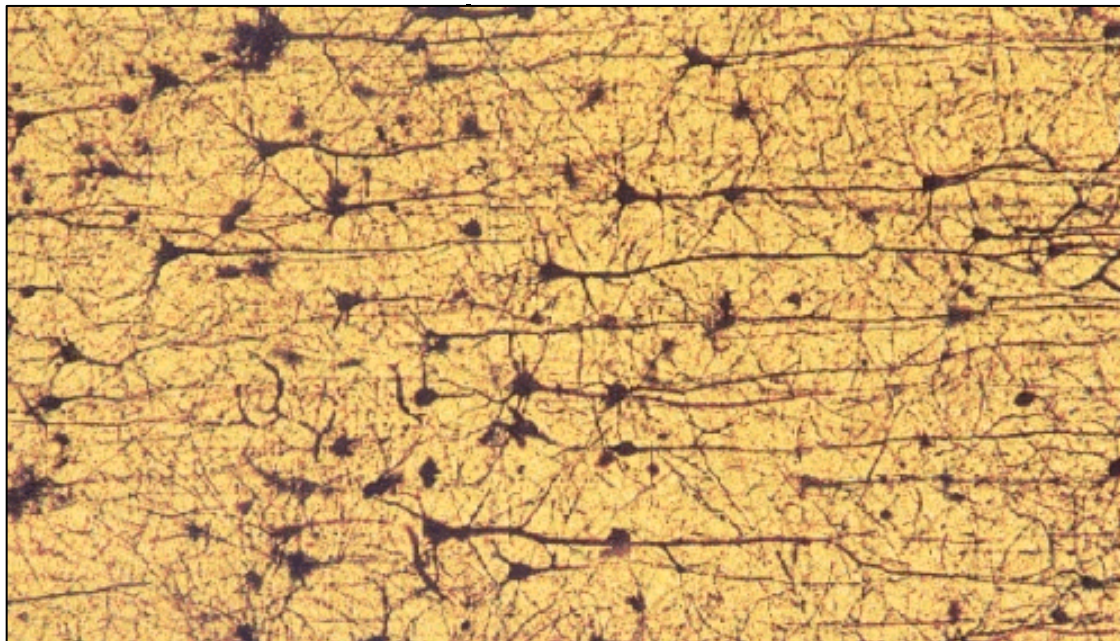




Introdução

Redes neurais naturais

- ◆ O sistema nervoso é formado por um conjunto extremamente complexo de células, os **neurônios**
- ◆ O cérebro humano possui cerca de 10^{11} neurônios e mais de 10^{14} **sinapses**, possibilitando a formação de redes





Introdução

Ramón e Cajál, em 1911, sugeriram que os constituintes básicos do cérebro são os neurônios.

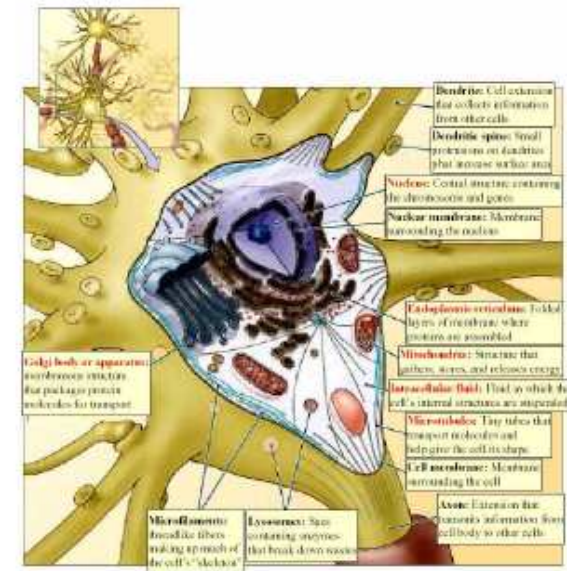
O cérebro humano apresenta aproximadamente 10^{11} neurônios, com aproximadamente 10^{13} sinapses (ou conexões).

Os neurônios são de 5 a 6 ordens de magnitude mais lentos do que portas lógicas de silício (10^{-3} seg. x 10^{-9} seg), porém, sua eficiência energética é muito alta.

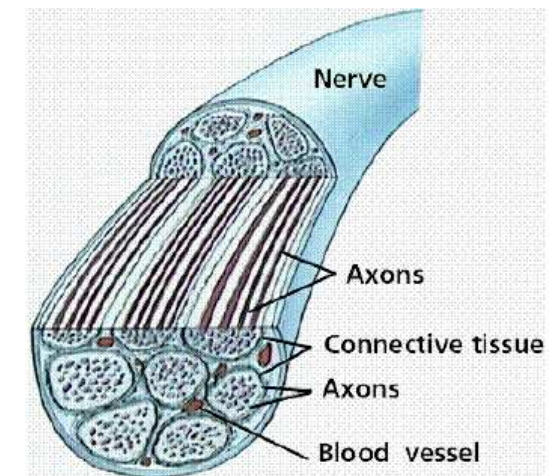
cérebro $\rightarrow 10^{-16}$ Joules/operação/seg

computador $\rightarrow 10^{-6}$ Joules/operação/seg

Como consequência o cérebro realiza reconhecimento de padrões, percepção e controle motor muitíssimo mais rápido e melhor do que qualquer computador já produzido.



Elementos constituintes de um neurônio



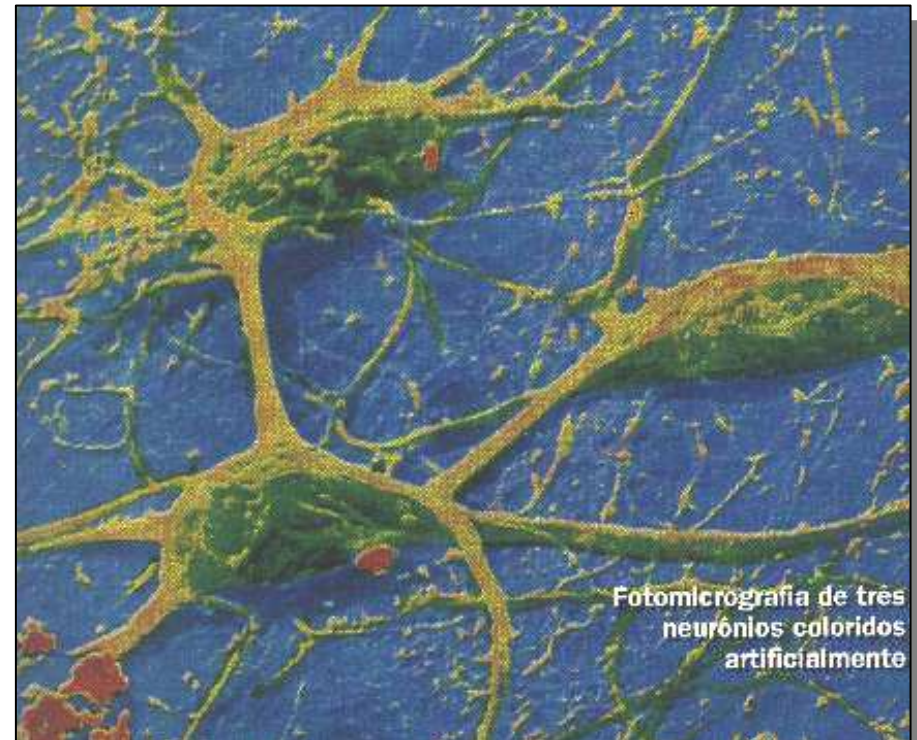
Estrutura de um nervo



Introdução

Neurônio natural

- ◆ Neurônios têm papel essencial na determinação do funcionamento e comportamento do corpo humano e do raciocínio



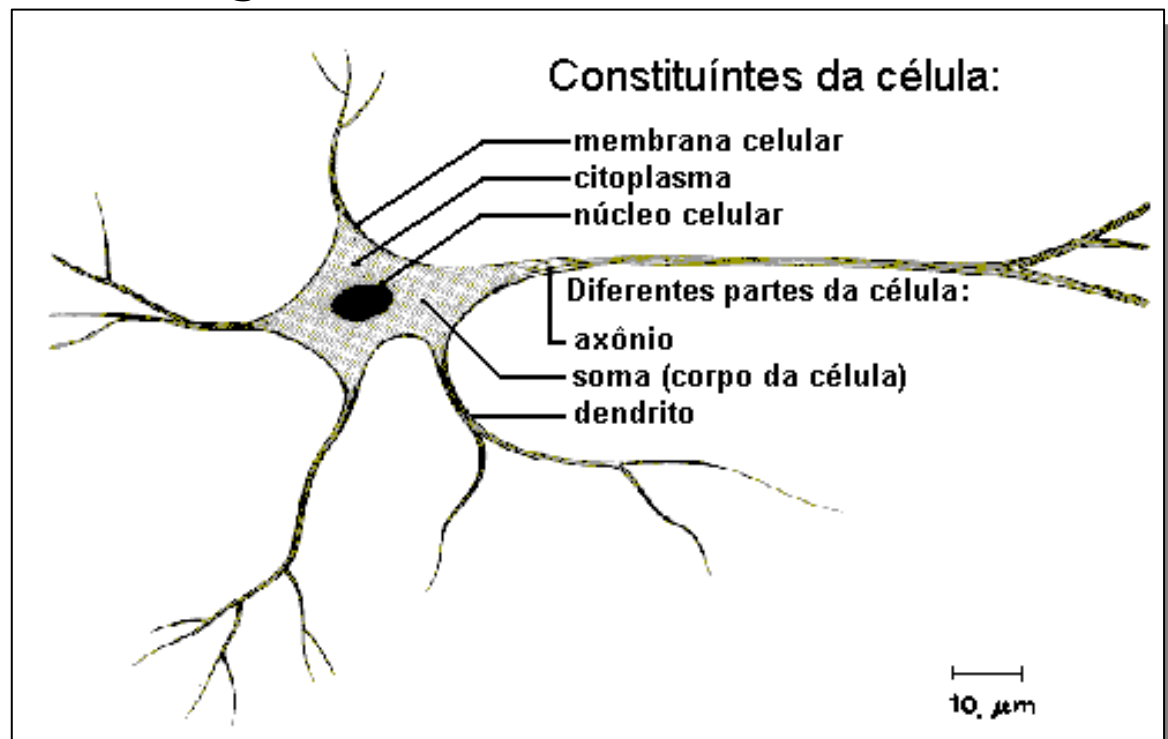
Fotomicrografia de três
neurônios coloridos
artificialmente



Introdução

Neurônio natural

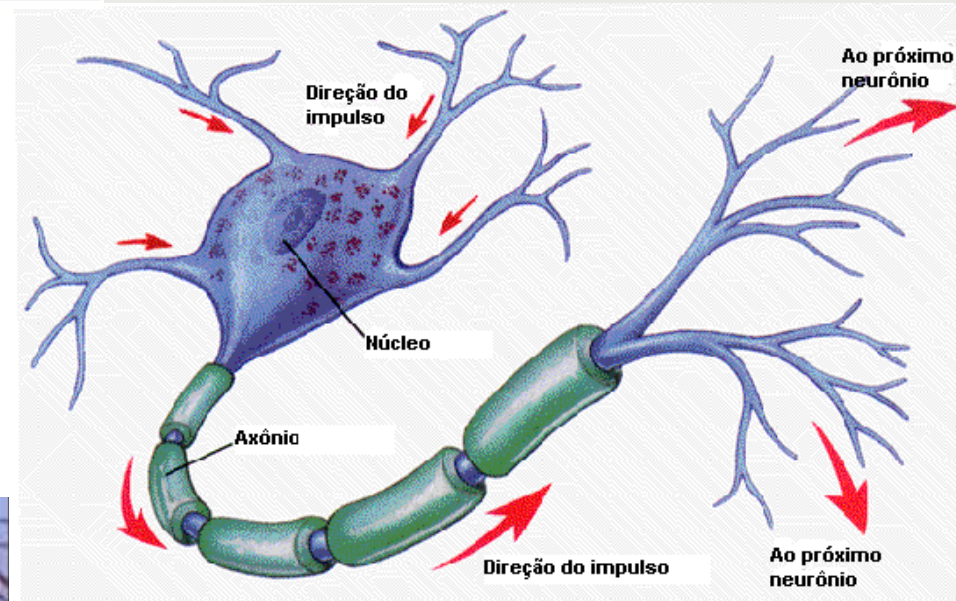
- ◆ Neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo central, e pelos axônios que são longos terminais de saída
- ◆ Neurônios se comunicam através de *sinapses*





Introdução

Neurônio natural



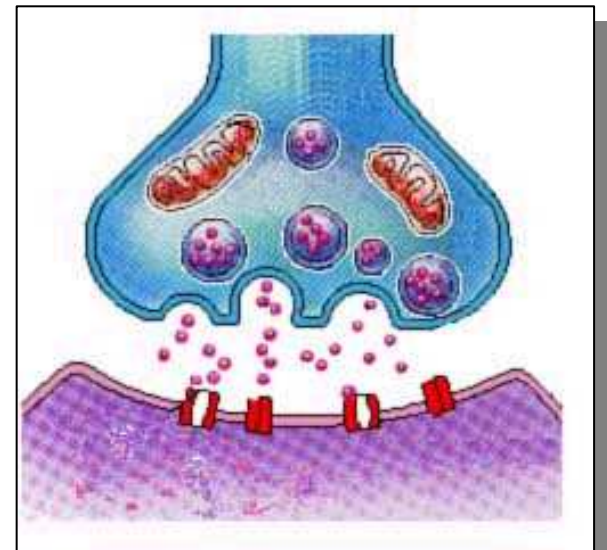


Introdução

Neurônio natural

♦ Sinapse

- É o contato entre dois neurônios através da qual os impulsos nervosos são transmitidos entre eles
- Os impulsos recebidos por um neurônio são processados e, atingindo um limiar de ação, dispara, produzindo uma substância neurotransmissora que flui para o axônio, que pode estar conectado a um dendrito de outro neurônio
- O neurotransmissor pode diminuir ou aumentar a polaridade da membrana pós-sináptica, inibindo ou excitando a geração dos pulsos no outro neurônio
- Este processo depende de fatores como a geometria da sinapse e o tipo de neurotransmissor





Introdução

Por que utilizar Redes Neurais Artificiais (RNA)?

➤ Os atributos de uma RNA, tais como aprender através de exemplos, generalizações redundantes e tolerância à falhas, proporcionam fortes incentivos para a escolha de RNA como uma escolha apropriada a aproximação para a modelagem de sistemas biológicos.

Potencial de uma RNA

- A arquitetura de uma RNA tem muitos neurônios conectados por pesos com capacidade de adaptação que podem ser arranjados em uma estrutura paralela.
 - Por causa deste paralelismo, a falha de alguns neurônios não causam efeitos significantes para a *performance* de todo o sistema, o que é chamado de tolerância à falhas.



Introdução

Por que utilizar Redes Neurais Artificiais (RNA)?

- A principal força na estrutura de RNA reside em sua habilidade de adaptação e aprendizagem.
 - A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de RNA podem lidar com dados imprecisos e situações não totalmente definidas.
 - Uma rede treinada de maneira razoável tem a habilidade de generalizar quando é apresentada à entradas que não estão presentes em dados já conhecidos por ela.
- A característica mais significativa de RNA está em sua habilidade de aproximar qualquer função contínua não-linear de um grau de correção desejado.
 - Esta habilidade das RNA as tem tornado útil para modelar sistemas não-lineares na combinação de controladores não lineares.



Introdução

Redes Neurais Artificiais (RNA)

- ◆ RNA são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência (Haykin, 2001; Russel and Norvig)
- ◆ Uma grande RNA pode ter centenas ou milhares de Unidades de Processamento (UP)



Introdução

Sinônimos:

- ◆ Paradigma conexcionista
- ◆ Paradigma não-simbólico
- ◆ Paradigma subsimbólico
- ◆ Rede neural artificial





Histórico

◆ Anos 40-50

- Primeiro modelo de redes neurais - modelo artificial de um neurônio biológico (McCulloch e Pitts, 1943)
- Modelo básico de rede de auto-organização - aprendizado de redes biológicas e artificiais (Hebb, 1949)
- Modelo Perceptron de aprendizado supervisionado (Rosemblatt, 1958)



Histórico

◆ Anos 60-70

- Widrow e Hoff - Regra Delta (1960)
- Minsky e Papert - Perceptron não resolve problemas não-linearmente separáveis (1969)
- Década onde abordagem conexionista “ficou adormecida” (1970)
 - K. Fukushima (Cognitron e Neocognitron)
 - Steven Grossberg (Sistemas Auto-Adaptativos)
 - Teuvo Kohonen (Memórias Associativas e Auto-organizáveis)
 - Modelos de redes neurais em visão, memória, controle e auto-organização (Amari, Anderson, Cooper, Cowan, Fukushima, Grossberg, Kohonen, von der Malsburg, Werbos e Widrow)



Histórico

◆ Anos 80-90

- Redes simétricas para otimização – Artigo – Propriedades Associativas das RNA (Hopfield, 1982)
- Algoritmo Backpropagation – Algoritmo de Retropropagação (Rumelhart, Hinton e Williams)



RNA

Alternativa aos sistemas simbólicos

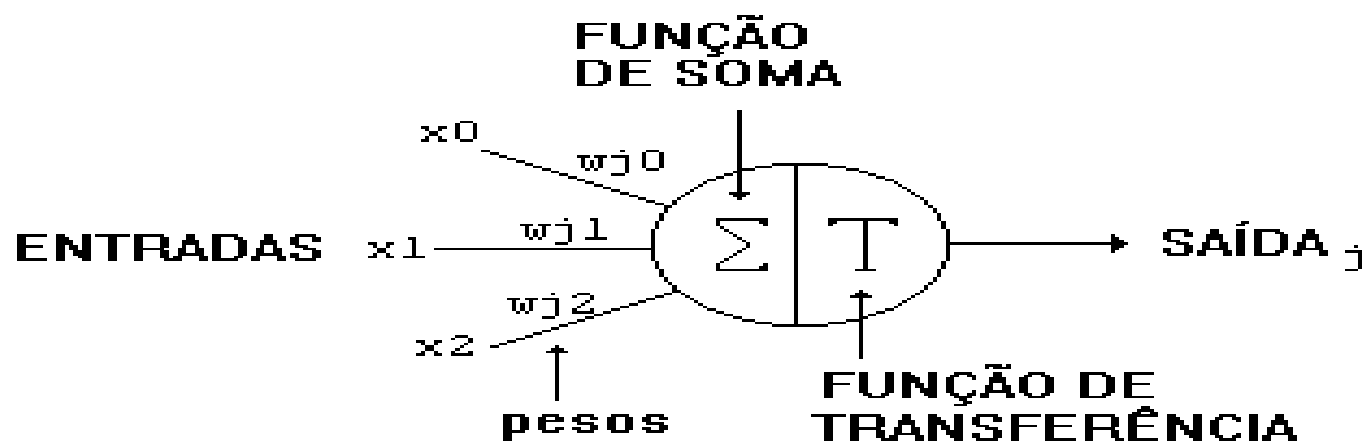
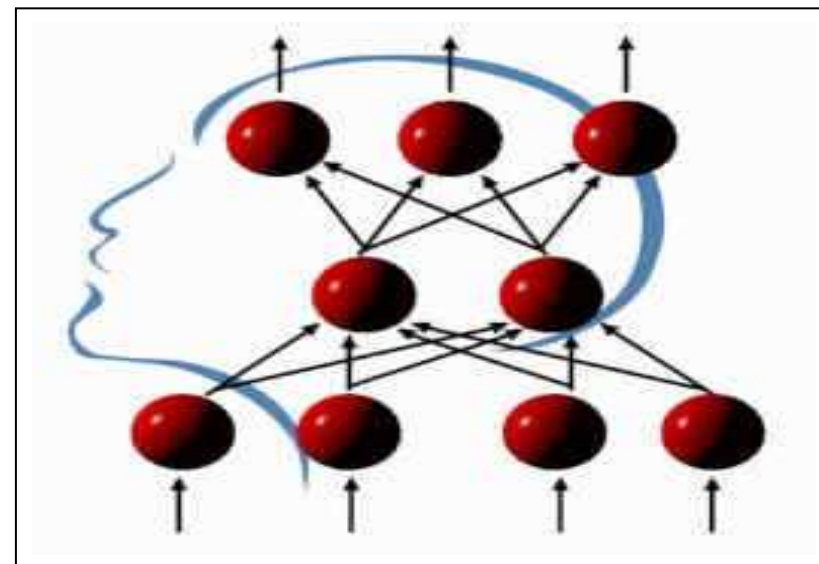
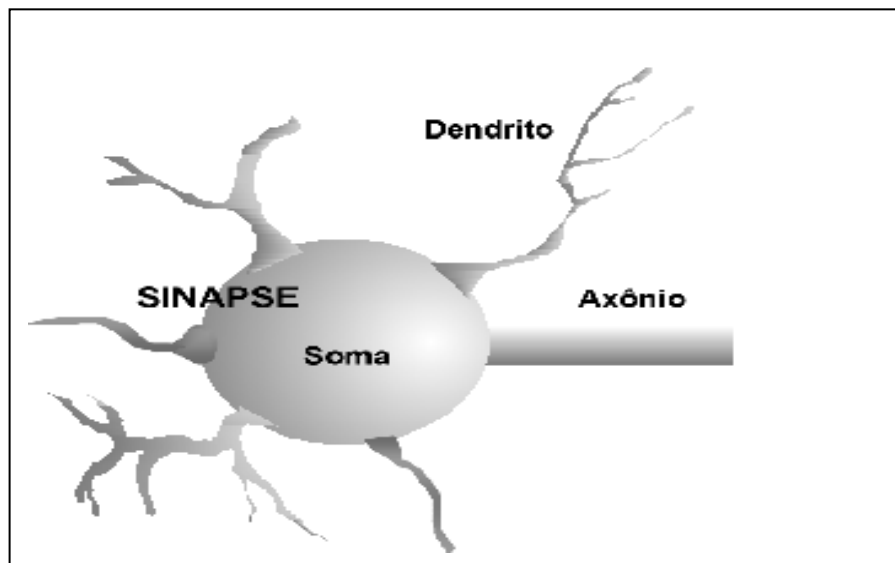
Caracterizado:

- ◆ Modelo do neurônio
- ◆ Topologia
- ◆ Regras de aprendizado

Problemas:

- ◆ Escolha da topologia
- ◆ Mapeamento dos atributos de dados, variáveis ou conceitos

Inspiração Biológica



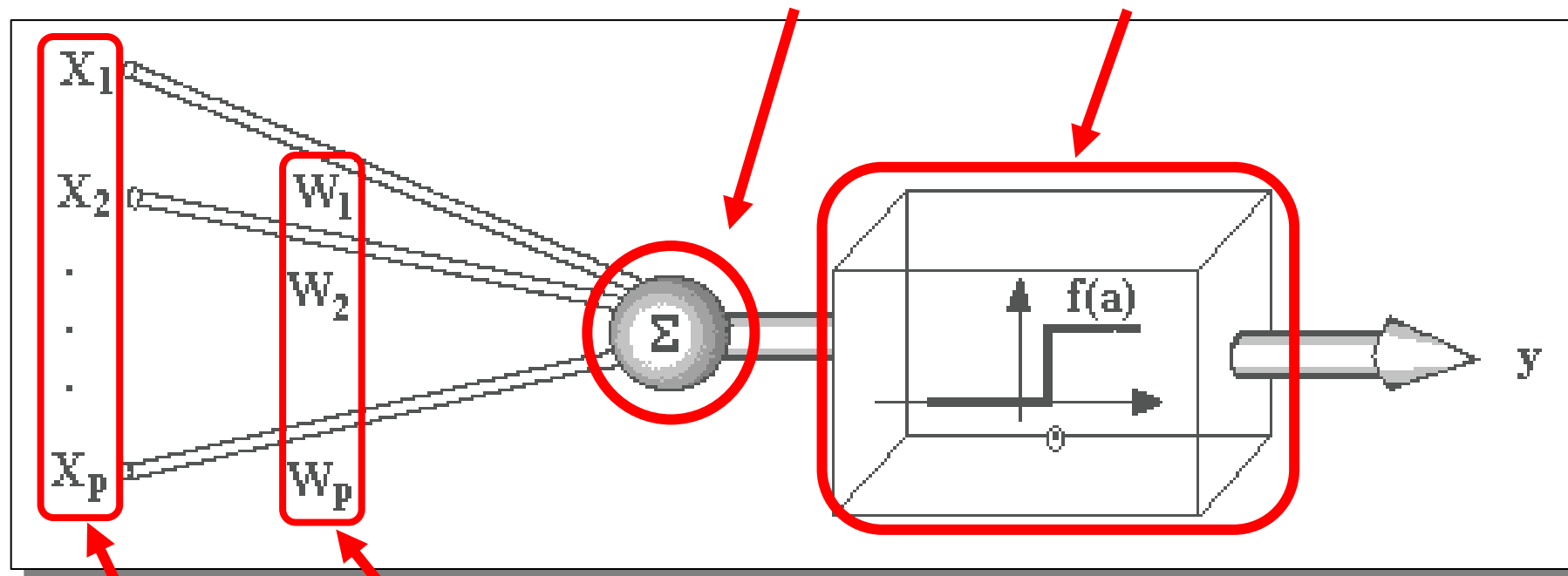


Características

- ◆ Operação de uma UP
- ◆ (McCullock & Pitts, 1943)

$$s = \sum W_i X_i$$

$$\begin{cases} y = 1, & \text{se } s > t \\ y = 0, & \text{se } s \leq t \end{cases}$$



Sinais de entrada pesos



Aprendizagem

- ♦ A maioria das arquiteturas de RNA possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados (elas ***aprendem através de exemplos***)
- ♦ **Lei de aprendizagem de Hebb**
 - “Se um neurônio A é repetidamente estimulado por um outro neurônio B, ao mesmo tempo em que ele está ativo, ele ficará mais sensível ao estímulo de B, e a conexão sináptica de B para A será mais forte. Deste modo, B achará mais fácil estimular A para produzir uma saída.”
- ♦ O conhecimento fica retido nos neurônios
- ♦ Para reter conhecimento, toda RNA passa por um processo de aprendizagem

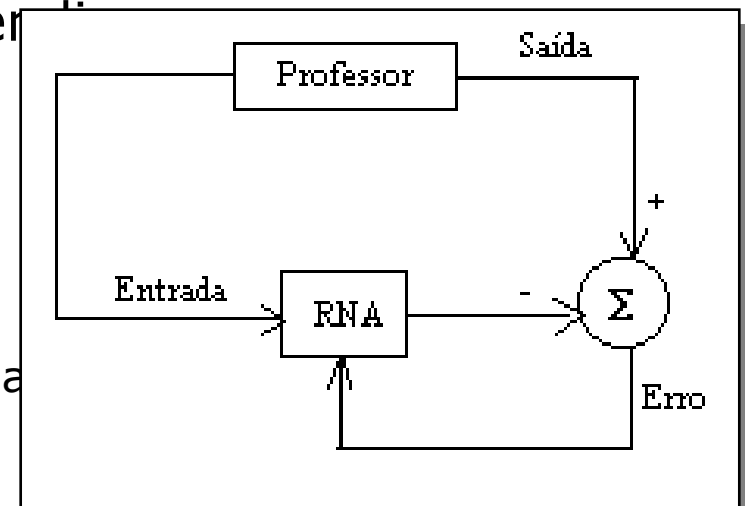


Aprendizagem

◆ Existem vários processos de aprendizagem

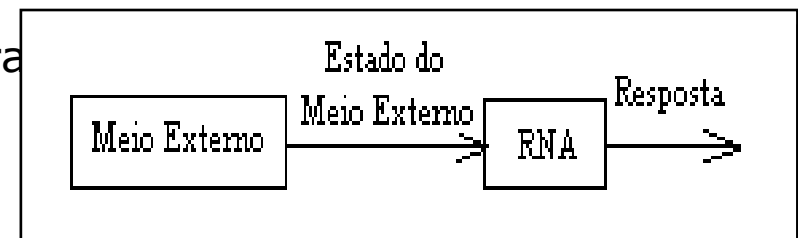
• **Aprendizagem supervisionada**

- A saída desejada é conhecida e informada para que a rede compare com a saída processada
- Se houver erro, a rede tenta corrigir este erro até que a mesma forneça uma saída igual a saída desejada



• **Aprendizagem não supervisionada**

- A saída desejada é obtida através de entradas repetitivas até a rede reter o conhecimento
- Não existe saída informada para comparação





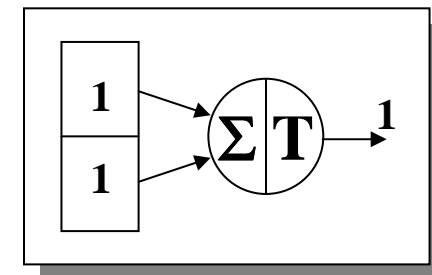
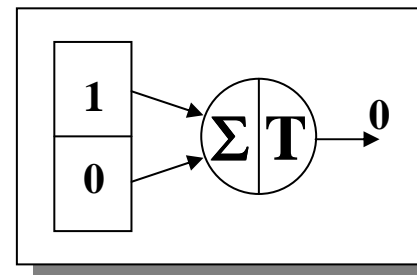
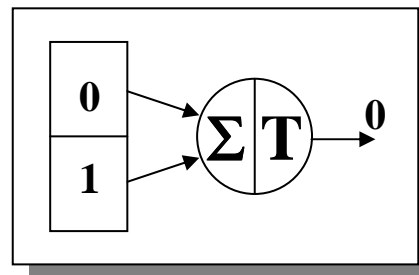
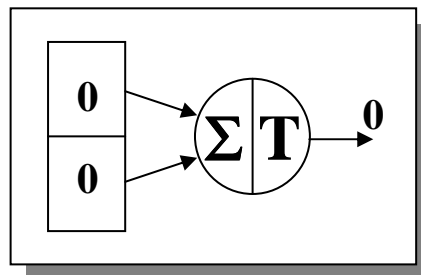
Aprendizagem

Exemplo: Implementação da porta lógica AND

Certificar-se de que todas as respostas estão corretas para cada conjunto de entradas pela tabela-verdade

A RNA possui um único neurônio de duas entradas e uma saída

Tabela Verdade - AND		
Entrada 1	Entrada 2	Saída
1	1	1
1	0	0
0	1	0
0	0	0





Aprendizagem

Exemplo: Implementação da porta lógica AND

Para treinar a rede vamos seguir alguns passos:

Para as entradas [1, 1] ...

Pesos iniciais [0, 0]

Passo 1: Aplicar a função *Soma*

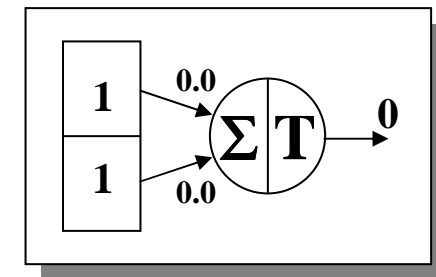
$$Soma = 1*0 + 1*0 = 0$$

Passo 2: Aplicar a função de *Transferência*

$$Soma \leq 0,5 \rightarrow y = 0$$

$$Soma > 0,5 \rightarrow y = 1$$

Transferido 0 para a saída. **Erro!!!!**



$$y = 1, \text{ se } s > \theta$$
$$y = 0, \text{ se } s \leq \theta$$

limiar



Aprendizagem

Exemplo: Implementação da porta lógica AND

Passo 3: Ajuste do peso

Calcular o erro: $E = 1 - 0 = 1$

Calcular o fator de correção:

$$F_1 = c * E * x_1$$

$$F_1 = 0,5 * 1 * 1$$

$$F_1 = 0,5$$

$$F_2 = c * E * x_2$$

$$F_2 = 0,5 * 1 * 1$$

$$F_2 = 0,5$$

Calcular o novo peso:

$$w_{1novo} = w_1 + F_1$$

$$w_{1novo} = 0 + 0,5$$

$$w_{1novo} = 0,5$$

$$w_{2novo} = w_2 + F_2$$

$$w_{2novo} = 0 + 0,5$$

$$w_{2novo} = 0,5$$

Equação do erro:

$$E = S_d - S_o$$

onde

S_d é a saída desejada

S_o é a saída obtida

Fator de correção:

$$F = c * x * E$$

onde

$c = 0,5$ (constante)

x é a entrada

E é o erro

Equação do ajuste:

$$w_{novo} = w + F$$



Aprendizagem

Exemplo: Implementação da porta lógica AND

Para treinar a rede vamos seguir alguns passos:

Para as entradas [1, 1] ...

Pesos iniciais [0,5, 0,5]

Passo 1: Aplicar a função *Soma*

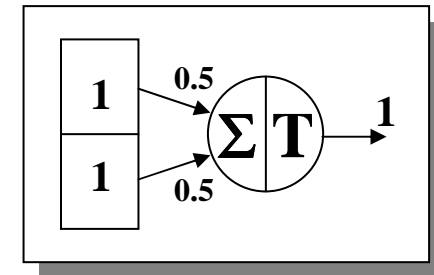
$$Soma = 1 * 0,5 + 1 * 0,5 = 1$$

Passo 2: Aplicar a função de *Transferência*

$$Soma \leq 0,5 \rightarrow y = 0$$

$$Soma > 0,5 \rightarrow y = 1$$

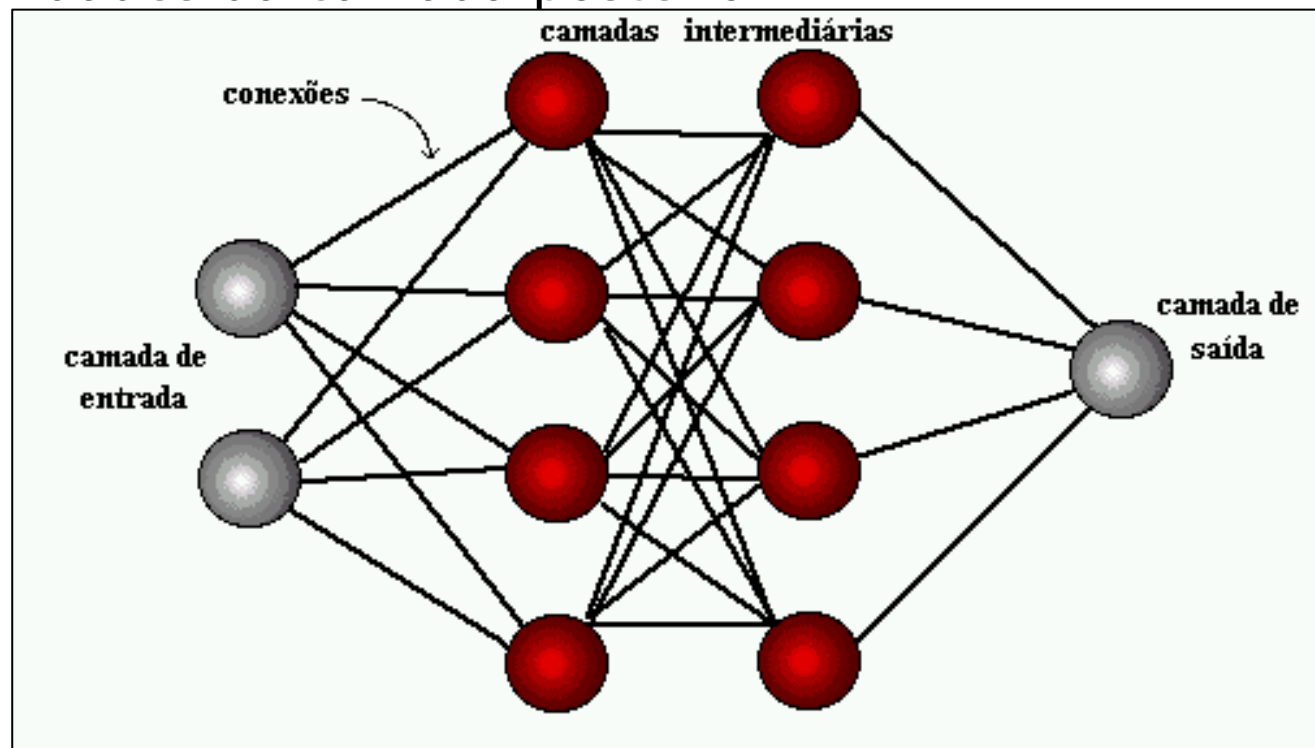
Transferido 1 para a saída. **Correto!!!!**





Arquiteturas de RNA

- ◆ Arquiteturas neurais são tipicamente organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior





Arquiteturas de RNA

- ◆ Usualmente as camadas são classificadas em três grupos:
 - **Camada de Entrada:** onde os padrões são apresentados à rede
 - **Camadas Intermediárias ou Escondidas:** onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características
 - **Camada de Saída:** onde o resultado final é concluído e apresentado

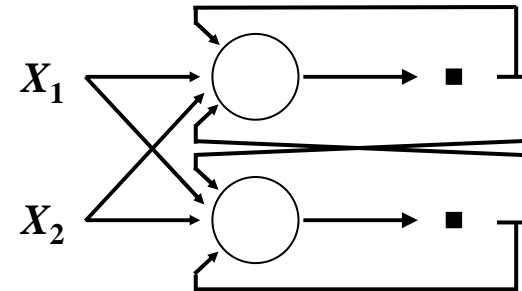
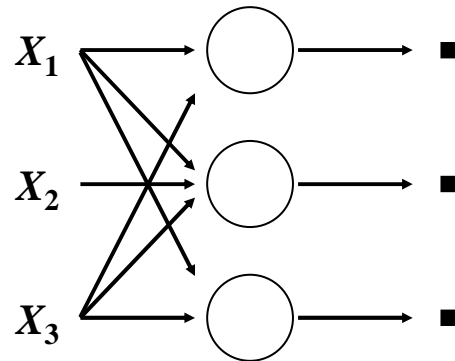


Arquiteturas de RNA

◆ Número de camadas

• Redes de camada única

- Só existe um nó entre qualquer entrada e qualquer saída da rede



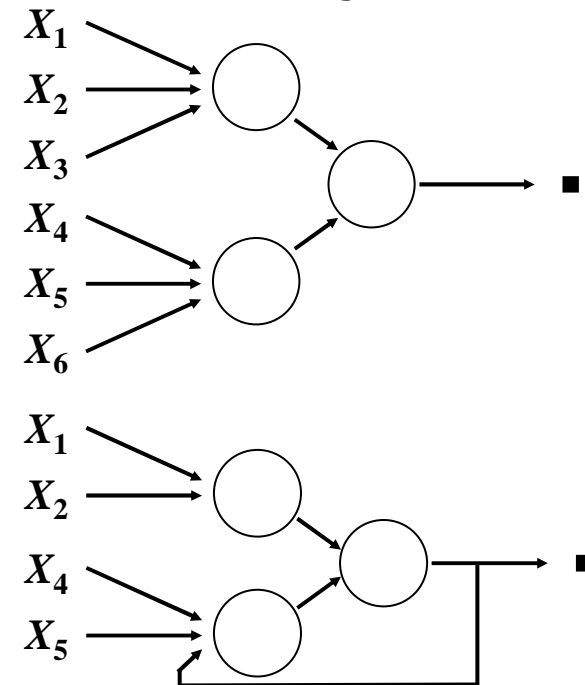
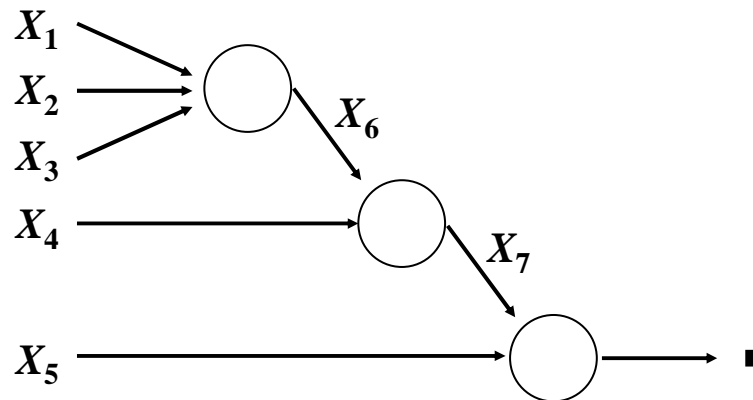


Arquiteturas de RNA

◆ Número de camadas

• Redes de múltiplas camadas

- Existe mais de um neurônio entre alguma entrada e alguma saída da rede



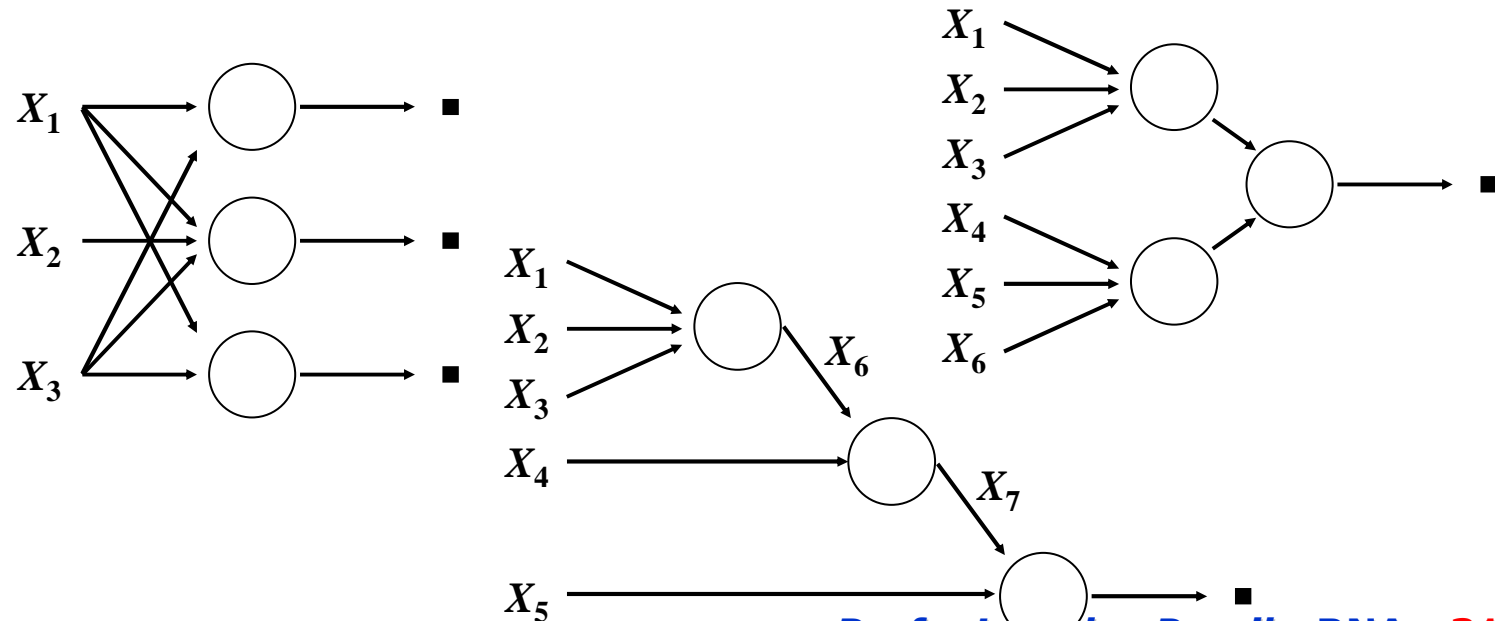


Arquiteturas de RNA

◆ Tipos de conexões dos nós

• **Feedforward, ou acíclica**

- A saída do neurônio na i -ésima camada da rede não pode ser usada como entrada de nodos em camadas de índice menor ou igual a i





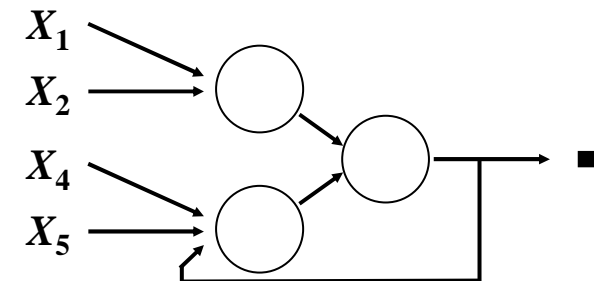
Arquiteturas de RNA

◆ Tipos de conexões dos nós

• **Feedback, ou cíclica**

- A saída do neurônio na i -ésima camada da rede é usada como entrada de nodos em camadas de índice menor ou igual a i

» Redes cuja saída final (única) é ligada às entradas comportam-se como autômatos reconhecedores de cadeias, onde a saída que é realimentada fornece o estado do autômato

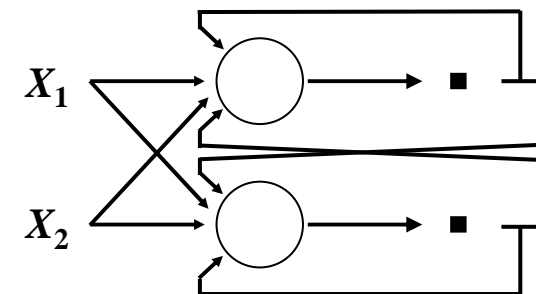


» **Auto-associativa**

Todas as ligações são cíclicas

Associam um padrão de entrada com ele mesmo

São particularmente úteis para recuperação ou regeneração de um padrão de entrada

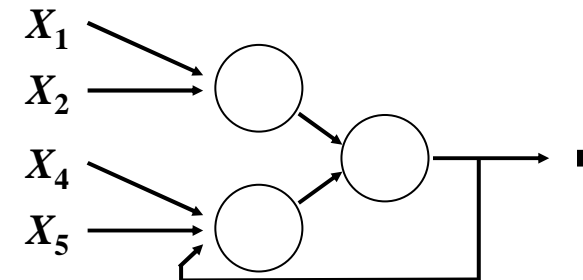
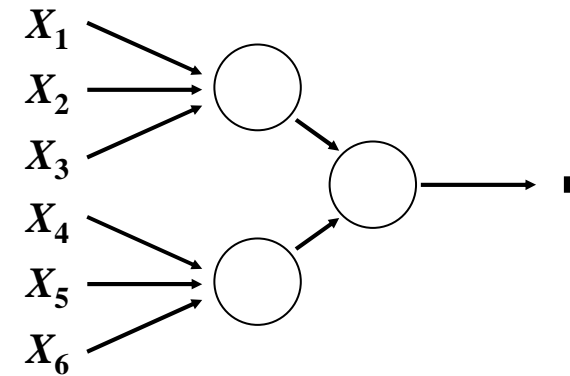
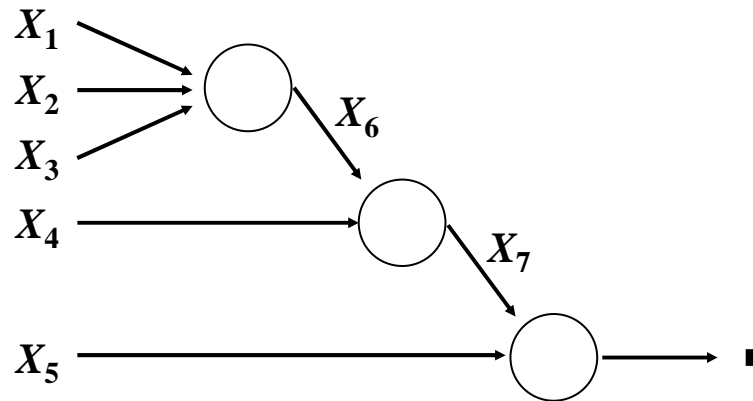




Arquiteturas de RNA

◆ Conectividade

• Fracamente (ou parcialmente) conectada

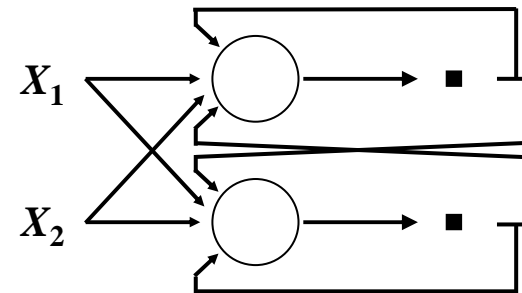
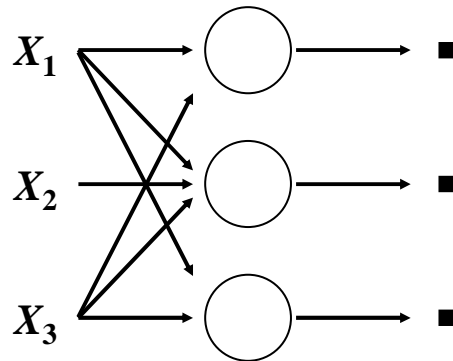




Arquiteturas de RNA

◆ Conectividade

- Completamente conectada





Arquiteturas de RNA

- ♦ Uma rede neural é caracterizada, principalmente
 - ... pela sua **topologia** (*feedforward, feedback*)
 - ... pelas **características dos nós** (*booleano, fuzzy, híbrido*)
 - ... pelas **regras de treinamento** (*Hebb, backpropagation, ...*)



Tipos de RNA

- ◆ Redes lineares
- ◆ **Perceptrons**
- ◆ ADALINE e MADALINE
- ◆ Perceptrons de Múltiplas Camadas
- ◆ Rede Hopfield
- ◆ Mapas de Kohonen
- ◆ Rede Counterpropagation
- ◆ Rede BAM (*Bidirectional Associative Memory*)
- ◆ Rede ART (*Adaptive Resonance Theory*)
- ◆ Rede IAC (*Interactive Activation and Competition*)
- ◆ ...



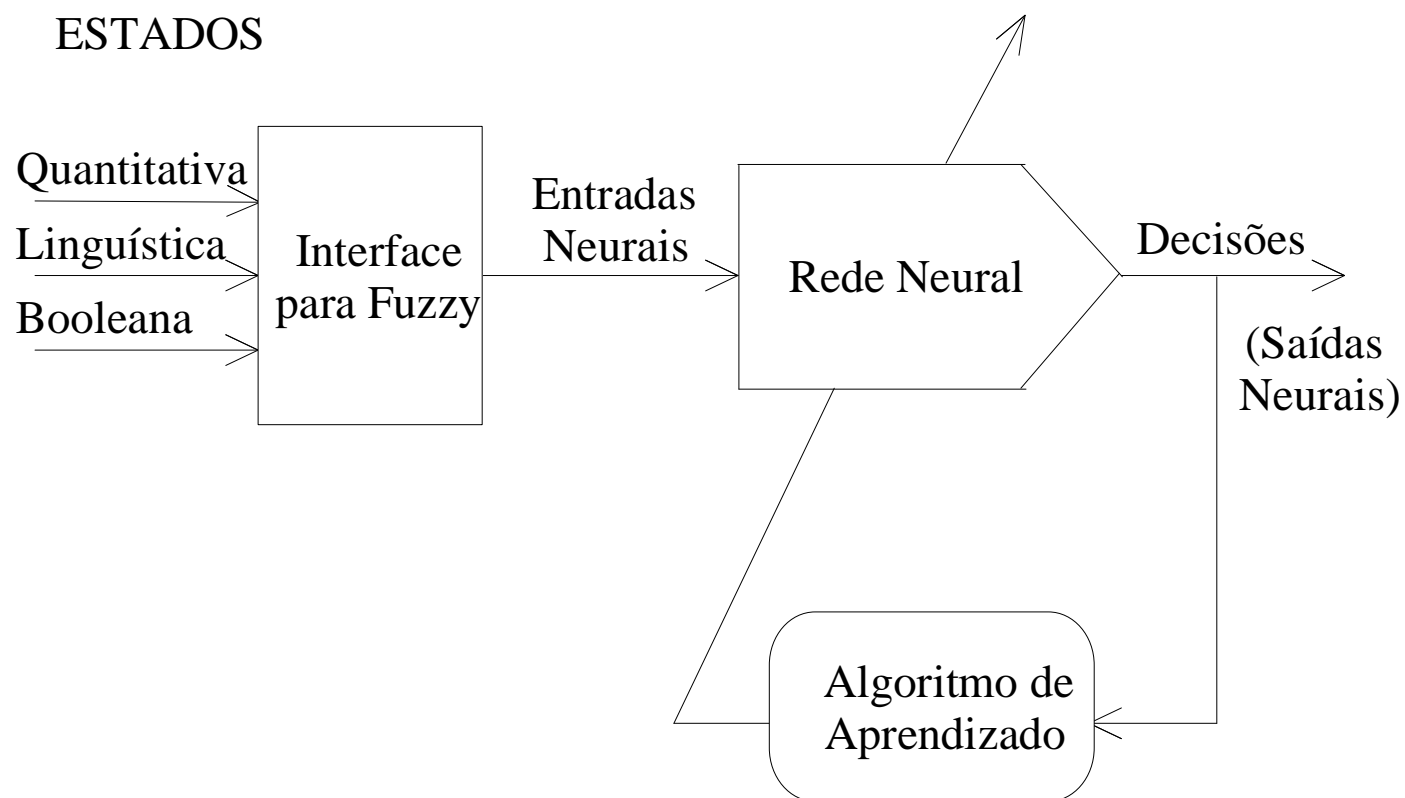
Portas de limiar (*threshold*)

Função	Equação com Polarização	Sem polarização	Com polarização
Degrau	$y = \begin{cases} 1, x > -\theta \\ 0, x < -\theta \end{cases}$		
Degrau Simétrico	$y = \begin{cases} 1, x > -\theta \\ -1, x < -\theta \end{cases}$		
Linear	$y = x + \theta$		
Logística Sigmoidal	$y = \frac{1}{1+e^{-(\mu+\theta)}}$		
Tangente Sigmoidal	$y = \frac{e^{(x+\theta)} - e^{-(x+\theta)}}{e^{(x+\theta)} + e^{-(x+\theta)}}$		

$$y' = y(1 - y)$$



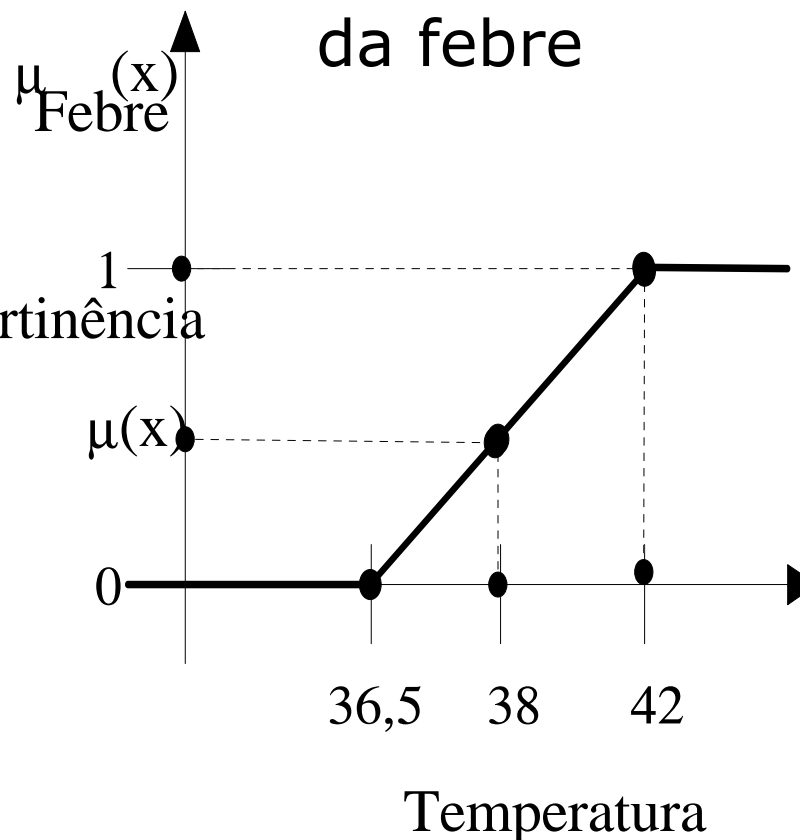
Tratamento das Variáveis





Quantitativas:

◆ Intensidade da febre



Booleanas:

◆ Febre

- Sim = 1
- Não = 0

Lingüísticas:

$$\mu_{\text{FEBRÍCULA}}(x) = 0.2$$

$$\mu_{\text{MODERADA}}(x) = 0.6$$

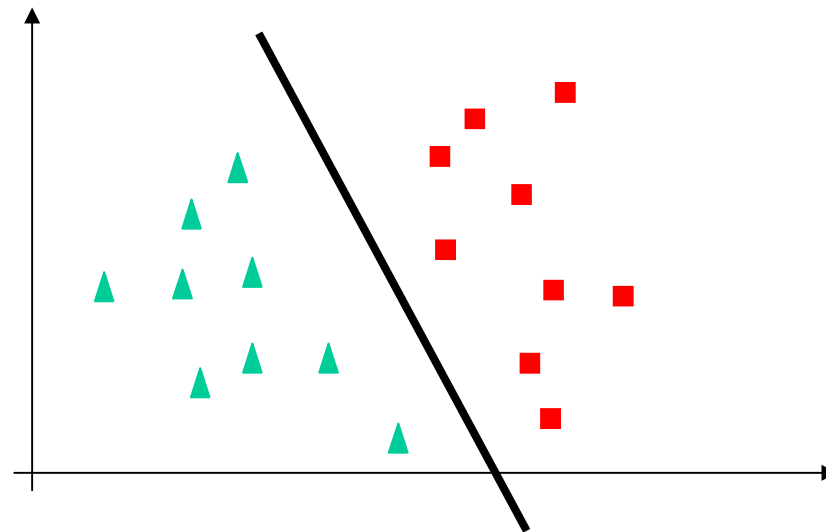
$$\mu_{\text{ALTA}}(x) = 0.9$$



Perceptrons

Introdução

- ◆ Desenvolvido por Rosembat (1958)
- ◆ Rede mais simples que pode ser utilizada para classificação de padrões linearmente separáveis
- ◆ Utiliza modelo de Mculloch-Pitts para o nó (Slide 22)





Perceptrons

Treinamento

- ◆ Supervisionado
- ◆ Correção de erro: $\Delta w_{ij} = \mu \cdot e \cdot x_i$
 - μ = taxa de aprendizagem
 - x_i = valor de entrada
 - $e = (d_j - y_i)$ = erro (valor calculado – valor desejado)
- ◆ Teorema da convergência:
“Se é possível classificar um conjunto de entradas, uma rede *Perceptron* fará a classificação”



Perceptrons

Treinamento

♦ Algoritmo:

Iniciar todas as conexões com $w_i = 0$ (ou aleatórios)

Repita

Para cada padrão de treinamento (X, d)

faça

Calcular a saída y

Se ($d \neq y$)

então atualizar pesos

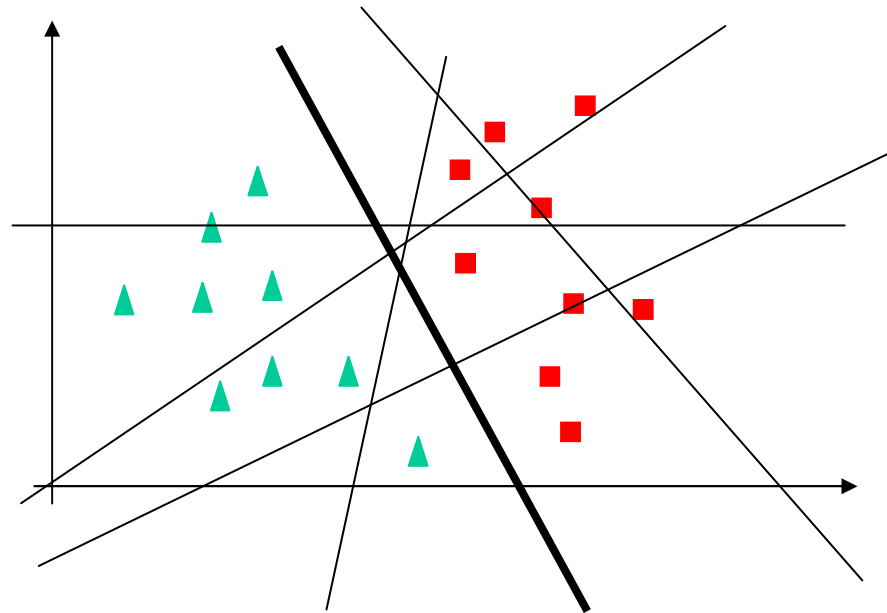
até o erro ser aceitável



Perceptrons

Treinamento

- ♦ Algoritmo:





Perceptrons

Treinamento

- ♦ Algoritmo de teste:

Para cada padrão de 1 a p
faça

Apresentar X_p à entrada da rede

Calcular a saída y

Se $y \geq \theta$

então $X_p \in \text{Classe 1}$

senão $X_p \in \text{Classe 2}$

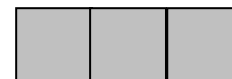
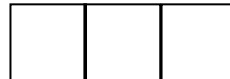


Perceptrons

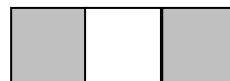
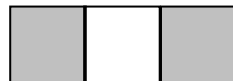
Treinamento

Exemplo:

- Ensinar uma rede *Perceptron* a classificar os seguintes padrões:



- Utilizar a rede treinada para classificar os padrões



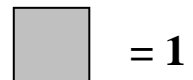


Perceptrons

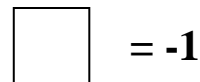
Treinamento

Exemplo:

- **Codificar as entradas**



= 1



= -1

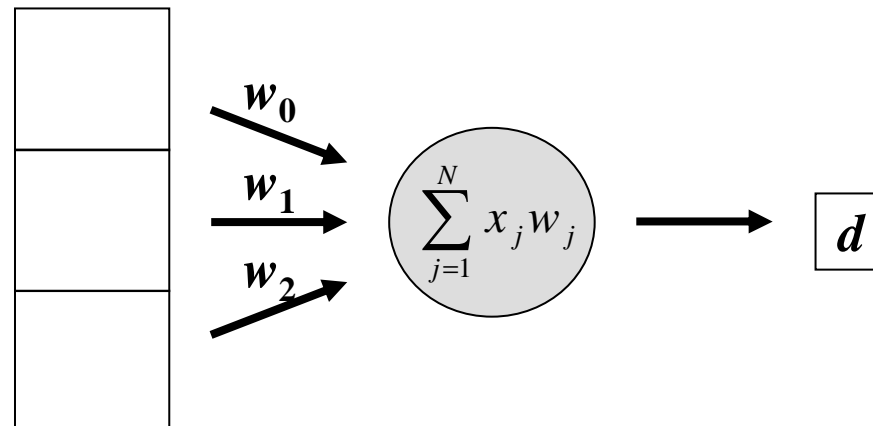


= 1 1 1



= -1 -1 -1

- **Supor** $\mu = 0.2$, $\theta = 0$, $w_0 = 0.4$, $w_1 = -0.8$, $w_2 = 0.3$





Perceptrons

Treinamento

Exemplo:

– Treinar a rede

» Para o padrão

--	--	--

 $\Leftrightarrow -1-1-1$ (**d = -1**)

Passo 1: Definir a saída da rede

$$d = (-1)(0.4) + (-1)(-0.8) + (-1)(0.3) = 0.1$$

$y = +1$ (uma vez que $0.1 \geq 0$)

Como ($d \neq y$), **atualizar pesos**

Passo 2: Atualizar pesos

$$w_0 = 0.4 + 0.2(-1)(-1 - (+1)) = 0.8$$

$$w_1 = -0.8 + 0.2(-1)(-1 - (+1)) = -0.4$$

$$w_2 = 0.3 + 0.2(-1)(-1 - (+1)) = 0.7$$



Perceptrons

Treinamento

Exemplo:

– Treinar a rede

» Para o padrão  $\Leftrightarrow 1 \ 1 \ 1$ (**d = 1**)

Passo 1: Definir a saída da rede

$$d = (1)(0.8) + (1)(-0.4) + (1)(0.7) = 1.1$$

$y = +1$ (uma vez que $1.1 \geq 0$)

Como ($d = y$), **não precisa atualizar pesos**




Perceptrons

Validação

Exemplo:

– Testar a rede

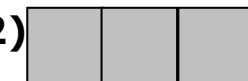
» Para o padrão  $\Leftrightarrow -11-1$

$$d = (-1)(0.8) + (1)(-0,4) + (-1)(0.7) = -1.9 \quad \text{(classe 1)}$$



» Para o padrão  $\Leftrightarrow 1-11$

$$d = (1)(0.8) + (-1)(-0,4) + (1)(0.7) = 1.9 \quad \text{(classe 2)}$$



» Para o padrão  $\Leftrightarrow 1-11$

$$d = (1)(0.8) + (-1)(-0,4) + (-1)(0.7) = 0,5 \quad \text{(classe 2)}$$





Perceptrons

Exercício

Implementar um discriminador de dois caracteres utilizando um único *perceptron* e *regra delta* (regra de aprendizado dos *perceptrons*) por meio de um programa. O neurônio, com n entradas, deverá discriminar os caracteres **T** e **H** descritos na forma de uma matriz $i \times j$, onde $ij = n$. Usar n pelo menos igual a 9 ($i = j = 3$). O neurônio deverá ser treinado para responder com **1** quando o valor de entrada for igual a **T** e **0** quando for igual a **H**. Fornecer os seguintes resultados:

- Curva de erro do neurônio durante o aprendizado (definir uma função de ativação para o neurônio usando uma função do tipo *sigmóide*)
- Respostas do neurônio quando a entrada for igual a **T** e **H**
- Comentar a capacidade de generalização da rede:
 - » Qual a resposta para caracteres não conhecidos?
 - » Testar com os caracteres **T** e **H** distorcidos

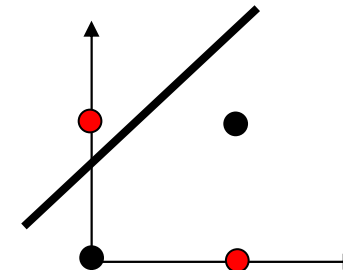
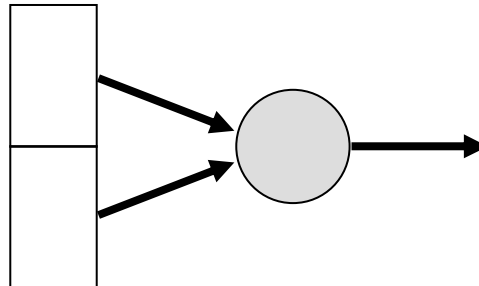


Perceptrons

Problema

- ◆ Redes com uma camada resolvem apenas problemas linearmente separáveis

0	0	→	0
0	1	→	1
1	0	→	1
0	0	→	0

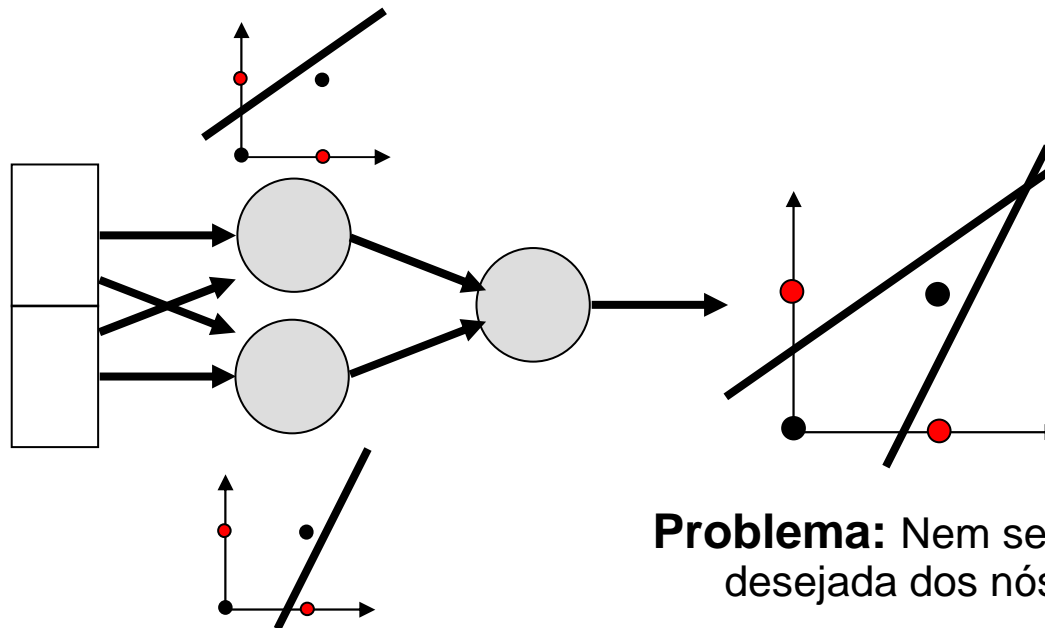




Perceptrons

Problema

- ♦ **Solução:** Utilizar mais de uma camada
 - **Camada 1:** uma rede Perceptron para cada grupo de entradas linearmente separáveis
 - **Camada 2:** uma rede combinando as saídas das redes da 1ª camada, produzindo a classificação final



Problema: Nem sempre se conhece a saída desejada dos nós da camada intermediária



Aprendizagem em Redes Neurais Artificiais

Introdução

- ◆ No aprendizado conexionista, não se procura obter regras como na abordagem simbólica da IA, mas sim determinar a intensidade de conexões entre neurônios
- ◆ Em outras palavras, aprendizagem em RNA é o processo de modificar os valores de pesos e do limiar (*bias*)



Aprendizagem em Redes Neurais Artificiais

Definição

“Aprendizagem é o processo pelo qual os parâmetros de uma RNA são ajustados através de uma forma continuada de estímulo pelo ambiente no qual a rede está operando, sendo o tipo específico de aprendizagem realizada definido pela maneira particular como ocorrem os ajustes realizados nos parâmetros” [MENDEL70].



Aprendizagem em Redes Neurais Artificiais

Regras de aprendizagem em RNA

- ◆ Estabelecer um conjunto de pesos para suas conexões, ativar um conjunto de unidades que correspondam a um padrão de entrada e observar o padrão para o qual a rede converge e em que se estabiliza
- ◆ Se o padrão final não corresponder ao que se deseja associar como resposta ao de entrada, é preciso fazer ajustes nos pesos e ativar novamente o padrão de entrada
- ◆ Por causa de sua semelhança com o aprendizado humano, esse processo de ajustes sucessivos das RNA é chamado de aprendizagem



Aprendizagem em Redes Neurais Artificiais

Regra de Hebb

- ◆ Desenvolvida por Donald Hebb em 1949
- ◆ **Princípio:** a força da conexão entre dois neurônios é aumentada se os neurônios estão simultaneamente excitados

$$\Delta w_{ij} = \mu \cdot y_i \cdot x_j$$



Aprendizagem em Redes Neurais Artificiais

Regra Delta (Widrow-Hoff)

- ♦ A regra Delta é uma variação da regra de Hebb
- ♦ Foi desenvolvida por Bernard Widrow e Ted Hoff (1982), conhecida também como *least mean square* (LMS), por minimizar o erro médio quadrático

$$net_i = \sum_j w_{ij} x_j + \Phi$$

$$\Delta W_{ij} = \eta \cdot (T_i - Y_i) \cdot \frac{d}{dx} (net_i) \cdot X_j$$

$$\Delta w_{ij} = \mu \cdot (d_i - y_i) \cdot x_j$$



Aprendizagem em Redes Neurais Artificiais

Regra Delta generalizada

♦ Algoritmo de retropropagação (*backpropagation*)

- Desenvolvido por Paul Werbos (1974) e redescoberto independentemente por Parker (1982) e Rumelhart (1986)
- Aplicado para RNA feedforward com uma ou mais camadas intermediárias
- Utiliza um método de descida de gradiente por correção de erro: o algoritmo de codificação executa um mapeamento entrada-saída através da minimização de uma função de custo qualquer



Aprendizagem em Redes Neurais Artificiais

Regra Delta generalizada

♦ Algoritmo de retropropagação (*backpropagation*)

- A função de custo é minimizada realizando-se iterativamente ajustes nos pesos sinápticos de acordo com o **erro quadrático** acumulado para todos os padrões do conjunto de treinamento
- Outras funções de custo podem ser utilizadas, mas independentemente disto, o procedimento de ajuste de pesos é realizado através do cálculo da mudança da função de custo com respeito à mudança em cada peso (método do delta)

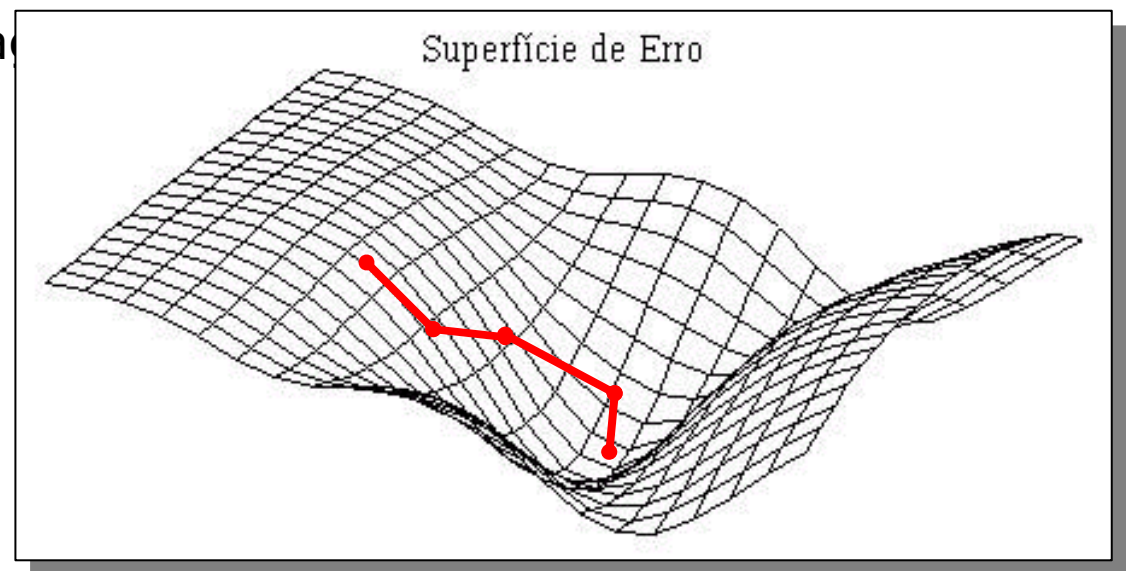


Aprendizagem em Redes Neurais Artificiais

Regra Delta generalizada

♦ Algoritmo de retropropagação (*backpropagation*)

- O processo de redução gradativa de erro que acompanha a minimização se denomina *convergência*
- A medida que a rede aprende, o valor do erro converge para um valor estável, normalmente irreduzível
- O processo de aprendizagem seja estabelecido, como por exemplo, um valor mínimo de erro global, ou uma diferença sucessiva mínima entre erros calculados para cada iteração





Aprendizagem em Redes Neurais Artificiais

Regra Delta generalizada

♦ Cálculo do erro na saída

- *Leans Means Square*

$$E = \frac{1}{2} \sum_i (d_i - y_i)^2$$

- *Root Means Square*

$$E = \sqrt{\sum_i (d_i - y_i)^2}$$



RNA – APLICAÇÃO

Toy-Problem

**Mostrar implementação
XOR ou Ou Exclusivo
Lourdes - Valéria**



Aplicações

Classificação

- ◆ Diagnósticos médicos (tomografias, raios-x, ultrasonografias etc)
- ◆ Análise de estruturas (ultra-som)

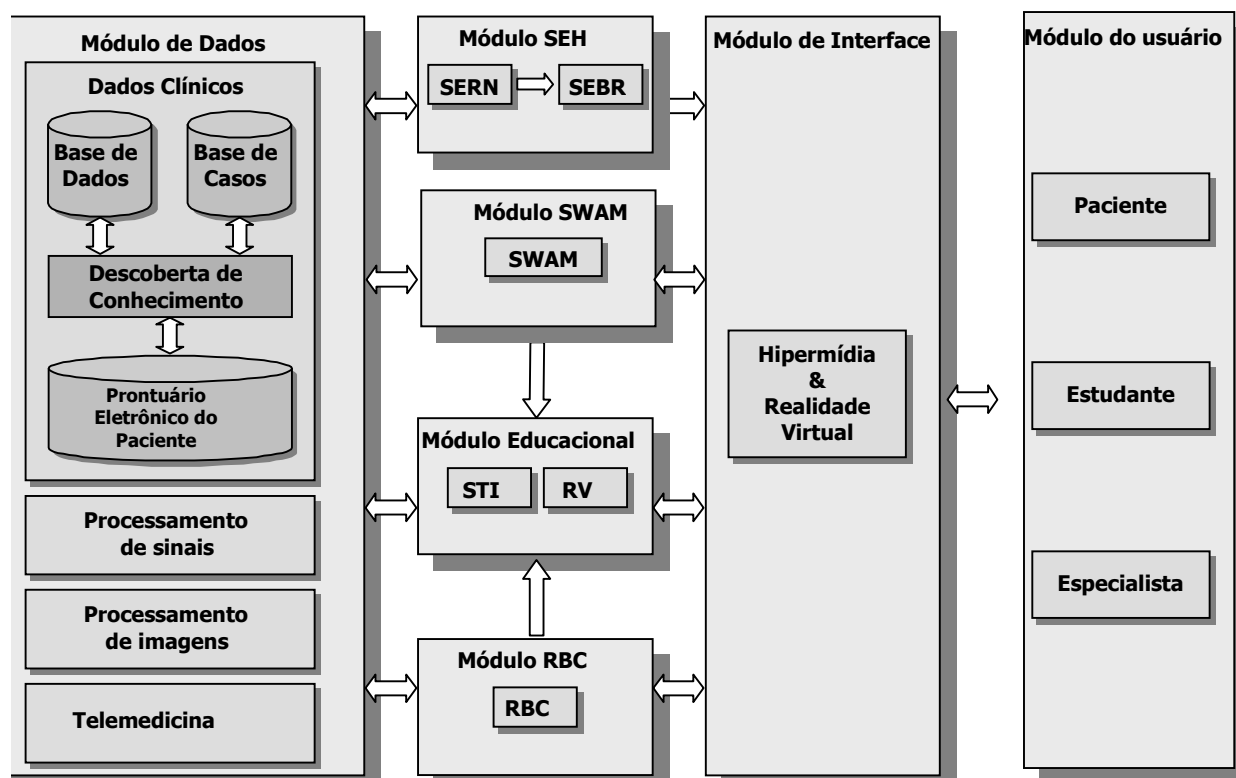


Aplicações

- Modelamento e controle neural
- Predição
 - Simulação de modelos não lineares
 - Previsão de tendências
 - Validação de sinais
- Otimização
- Processamento de Imagens e Sinais
- Processamento de voz
- Visão computacional

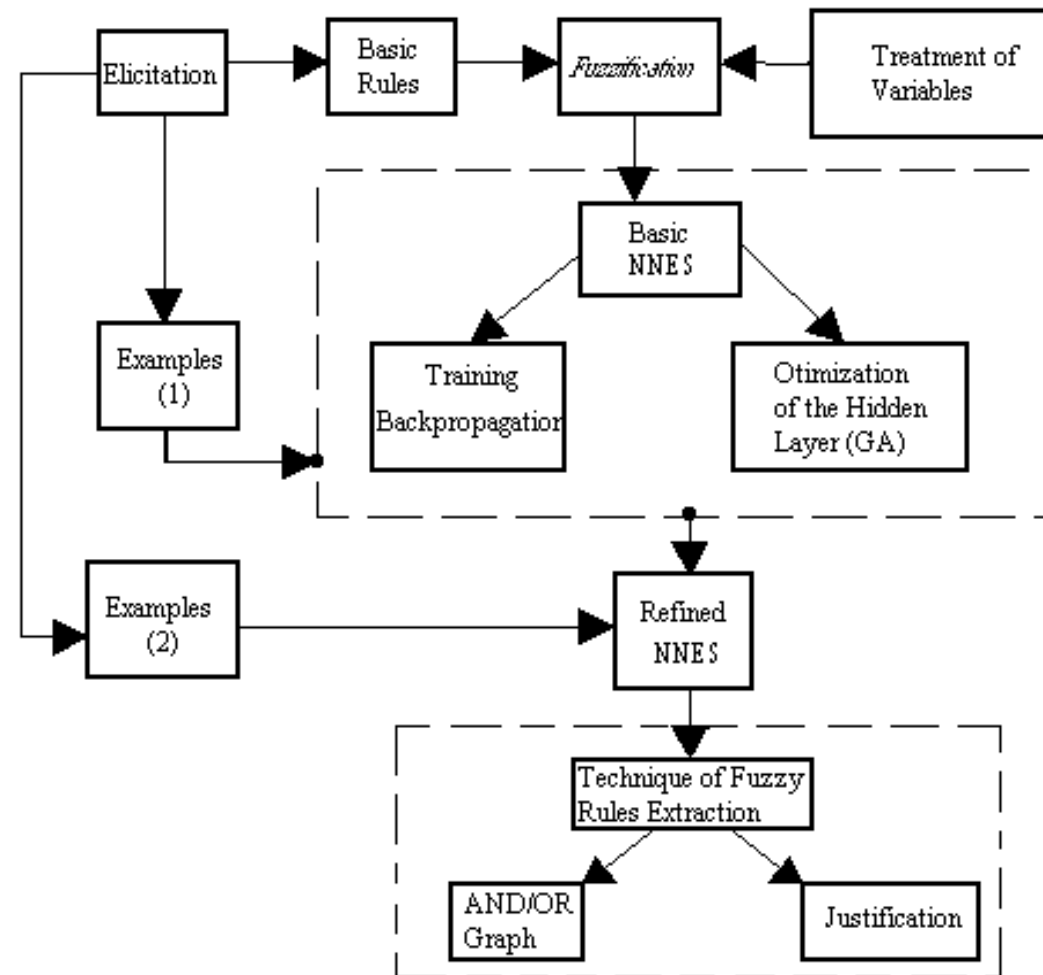


Sistemas Especialistas Híbridos





Sistemas Especialistas Híbridos





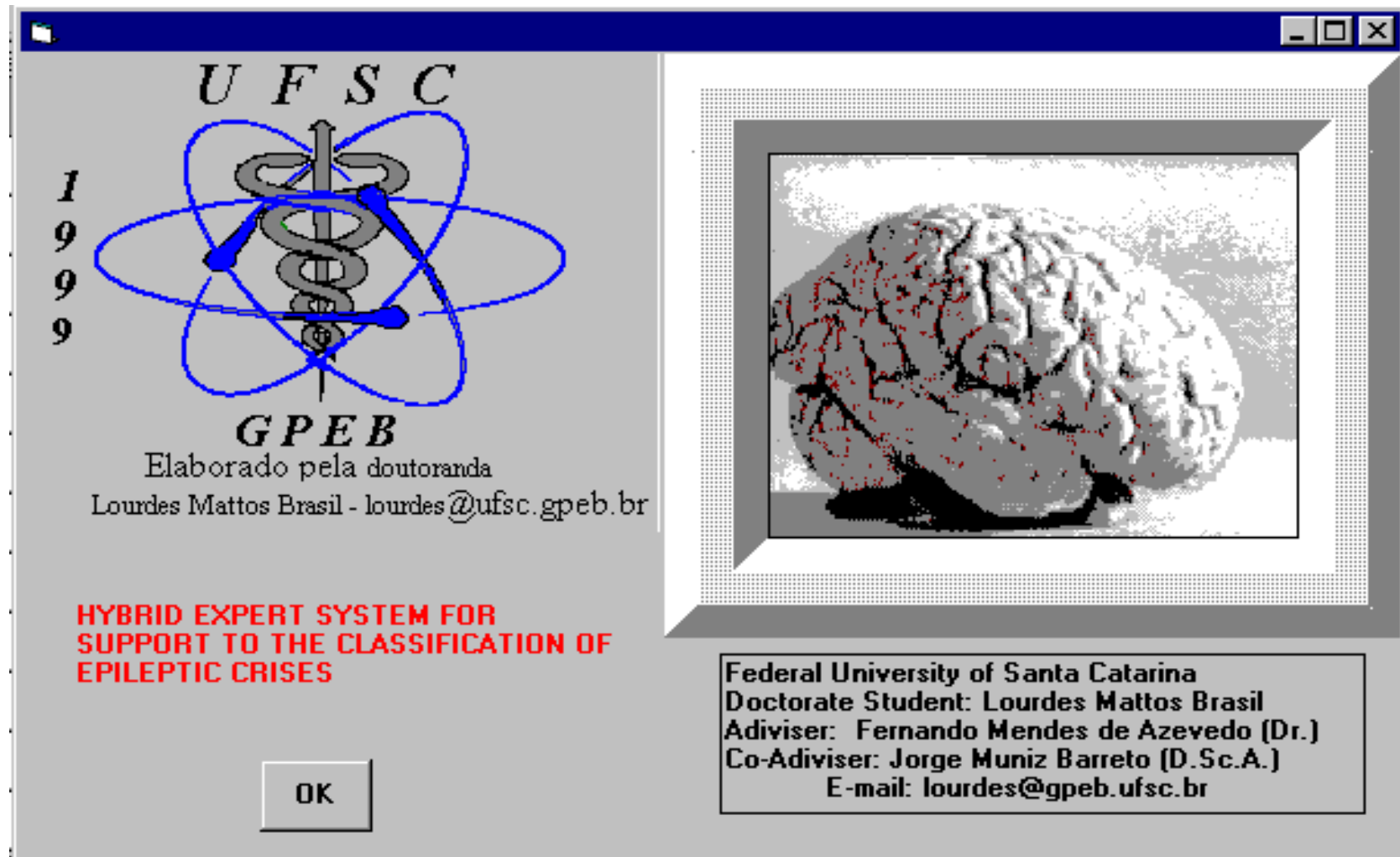
Universidade de Brasília

Faculdade UnB Gama



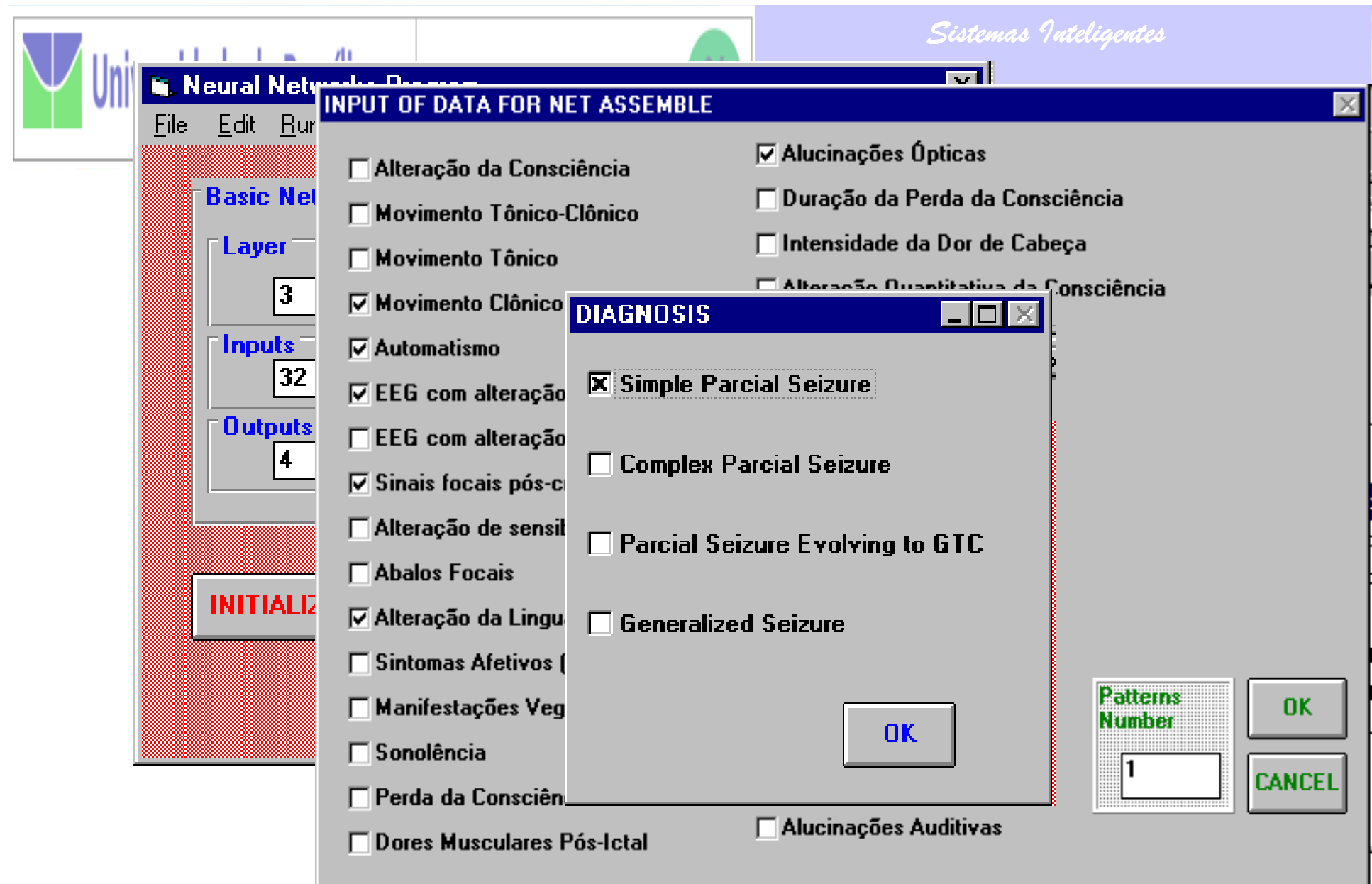
Sistemas Inteligentes

*Programa de Pós-Graduação em Engenharia
Biomédica*



Grupo de Pesquisas em Engenharia Biomédica (GPEB)
UFSC

Profa. Lourdes Brasil - RNA - 67



SOFTWARE PARA CLASSIFICAÇÃO DE CRISES EPILEPTICAS

Generations Number

10

Crossover Rate

0.6

Generation

Initial Population

8

Mutation Rate

0.1

Nhidden

Net_Number

Mean Value

Fitness

Initial Distribution

Uniform

Gauss

The Best Network/Generation

NCHidden

Fitness

Fitness Functions

FF = 1/NCI + ERROR

FF = NCI/ERROR

Results

Fitness

NCHidden

BackPropagation

Learning Rate

0.3

Momentum

0.7

Tolerance

0.01

Max Epoches

100

Total_Error

Total_Epoches

Epochs/Generation

Graphic - Back



Initialization

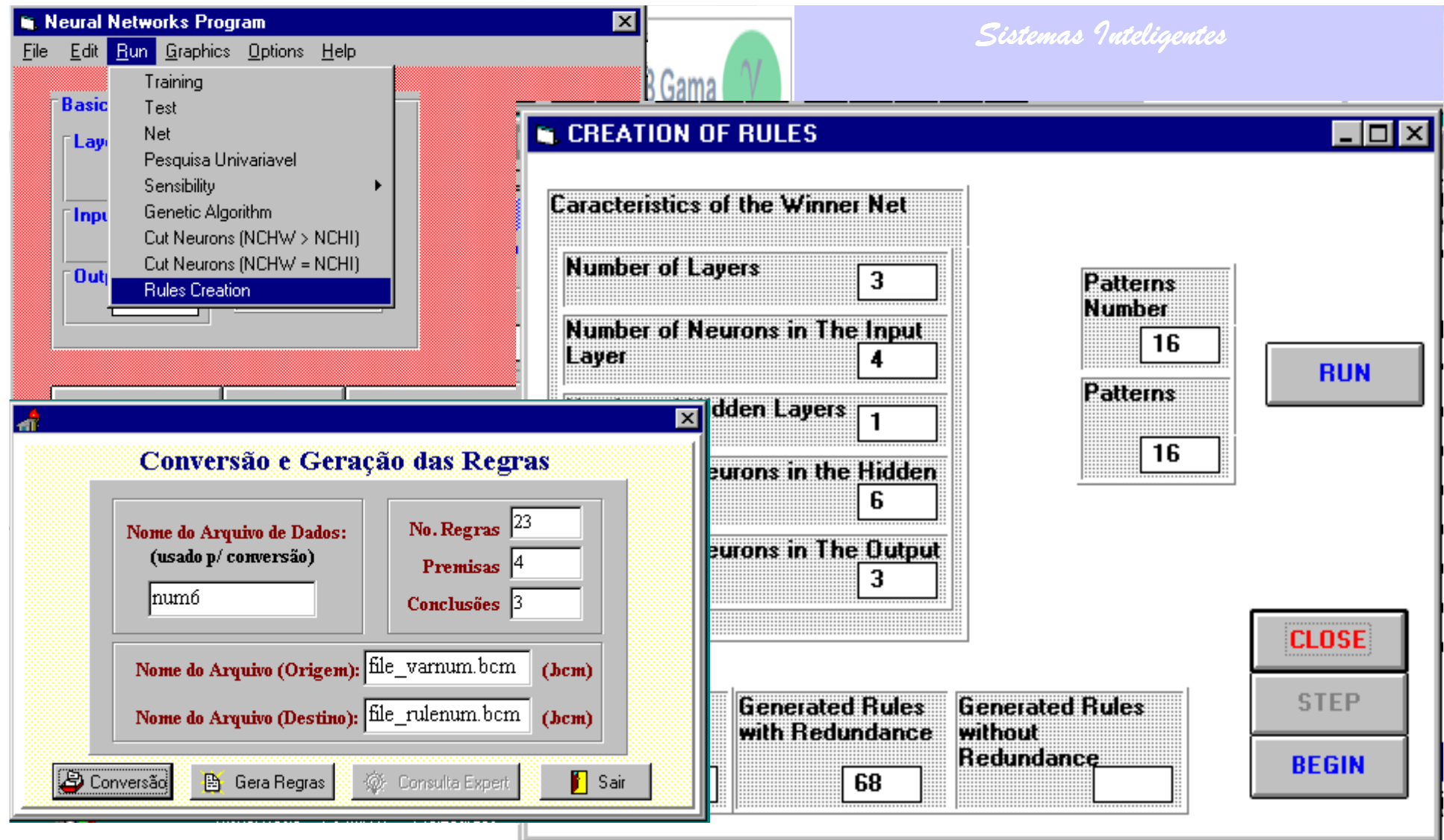
Reset

Continue

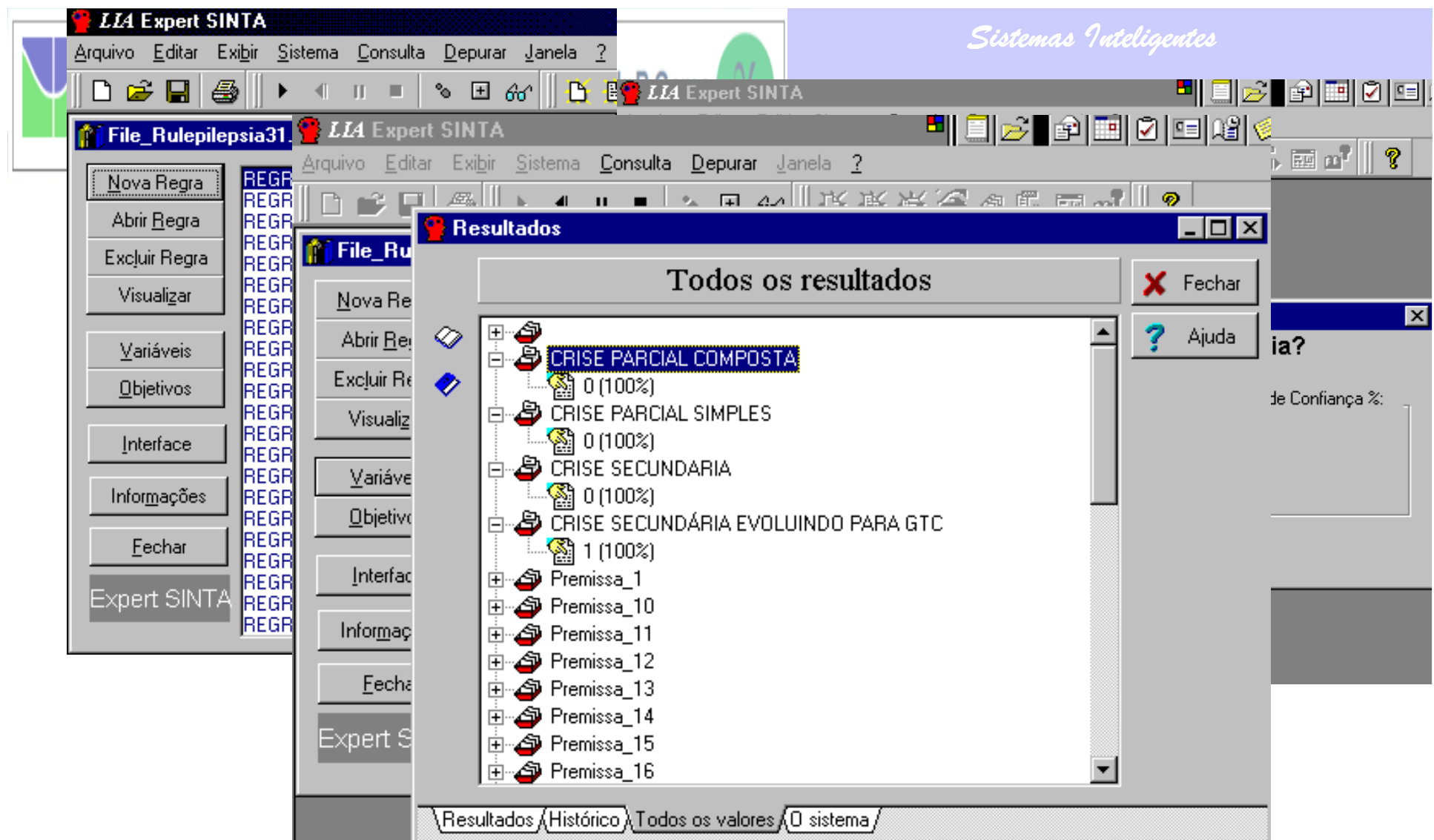
Exit

STOP

TREINAMENTO E ESCOLHA DA MELHOR REDE NEURAL



EXTRAÇÃO DAS REGRAS FUZZY



RESULTADOS FINAIS: CLASSIFICAÇÃO DAS CRISES EPILÉPTICAS



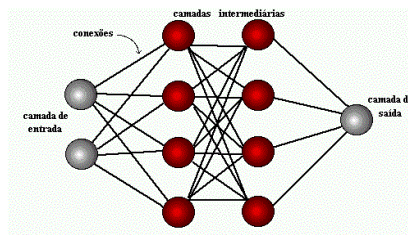
Universidade de Brasília

Faculdade UnB Gama

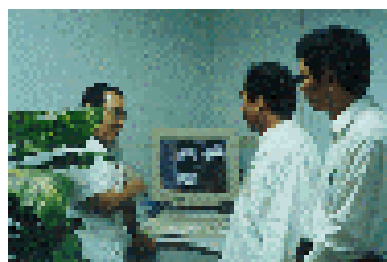
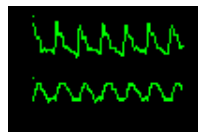


Sistemas Inteligentes

*Programa de Pós-Graduação em Engenharia
Biomédica*



Visual
Prolog



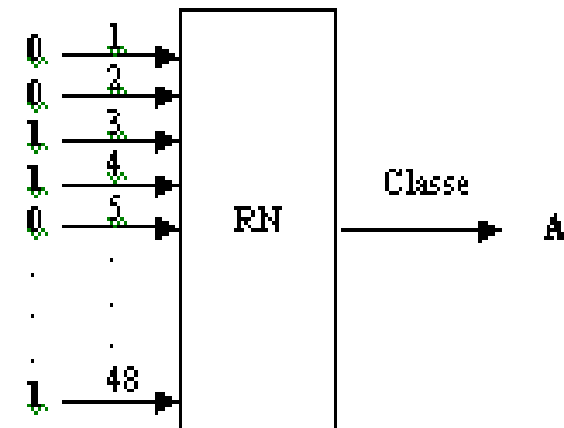
Next



Reconhecimento de Imagens

- Imagens simples podem ser identificadas, fornecendo-se um mapa de *bits* como entrada da RNA, enumerando-se as cores.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48



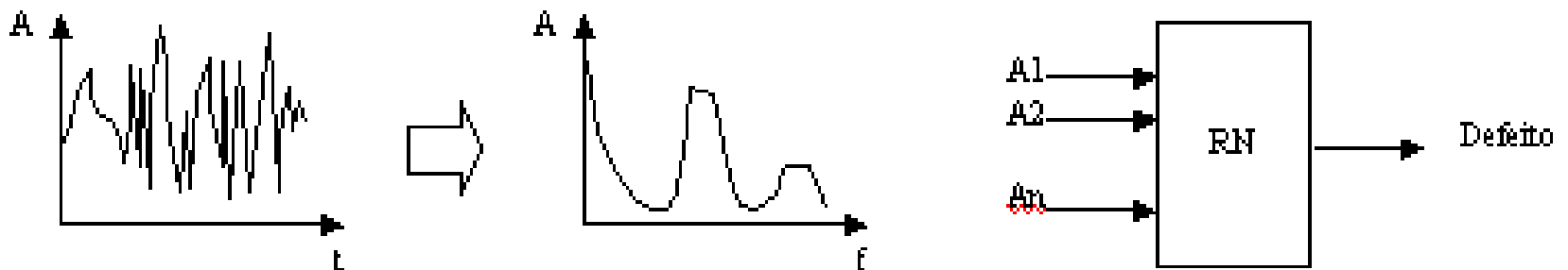


Identificação de Defeitos em Estruturas via Ultra-som

- A identificação de defeitos em estruturas via ultra-som baseia-se na análise das curvas (“assinaturas”) geradas.
- Geralmente são feitos pré-processamentos dos sinais de ultra-som, e é comum utilizar-se transformada de Fourier para mudar para o domínio da frequência. Tal transformação reduz o conjunto de dados a serem aprendidos, visando aumentar a performance da rede.



Identificação de Defeitos em Estruturas via Ultra-som

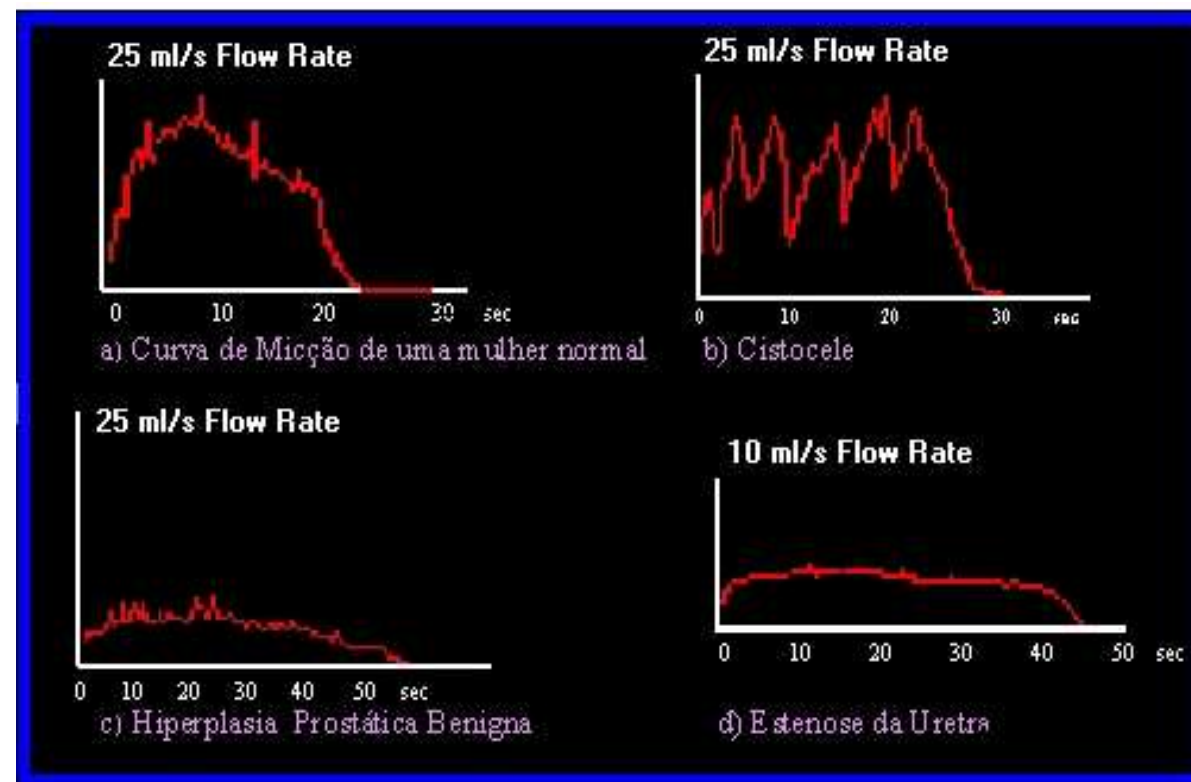


67



Vídeo-Urofluxômetro Computadorizado

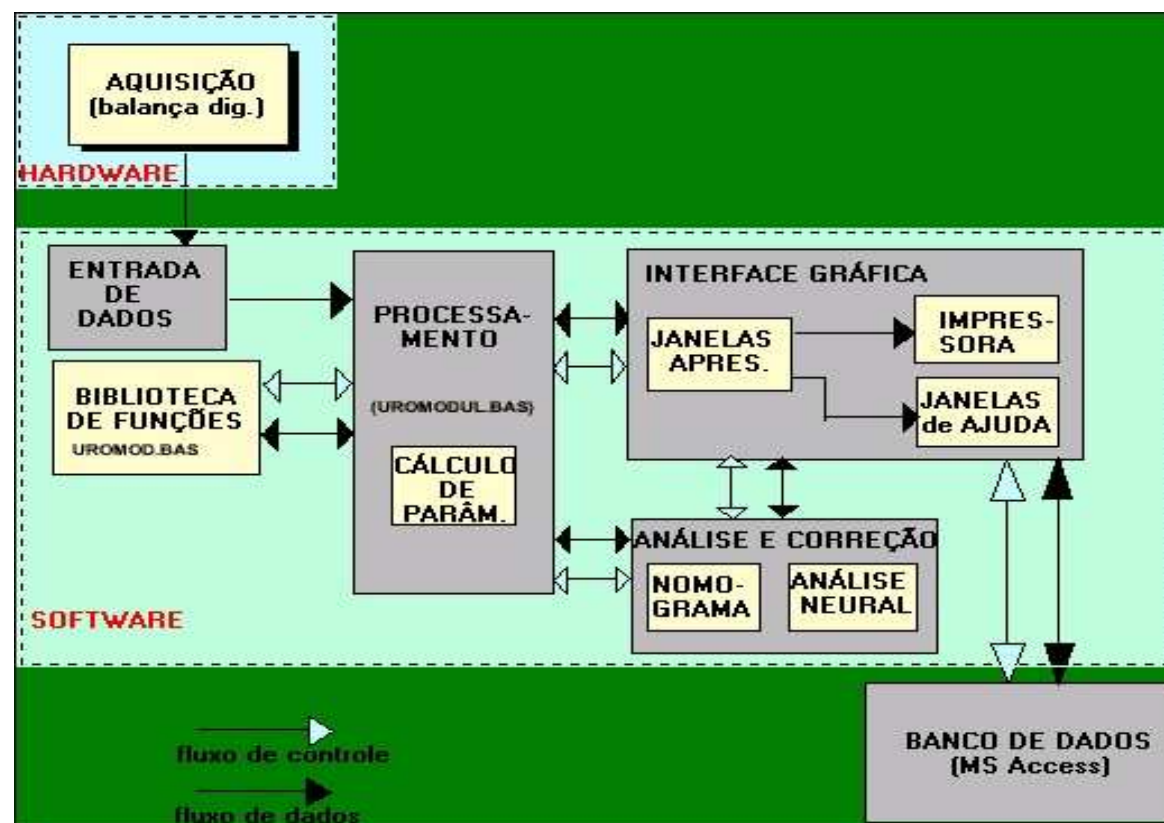
Exame fundamental em Urologia: a curva de Fluxo Urinário





Vídeo-Urofluxômetro Computadorizado

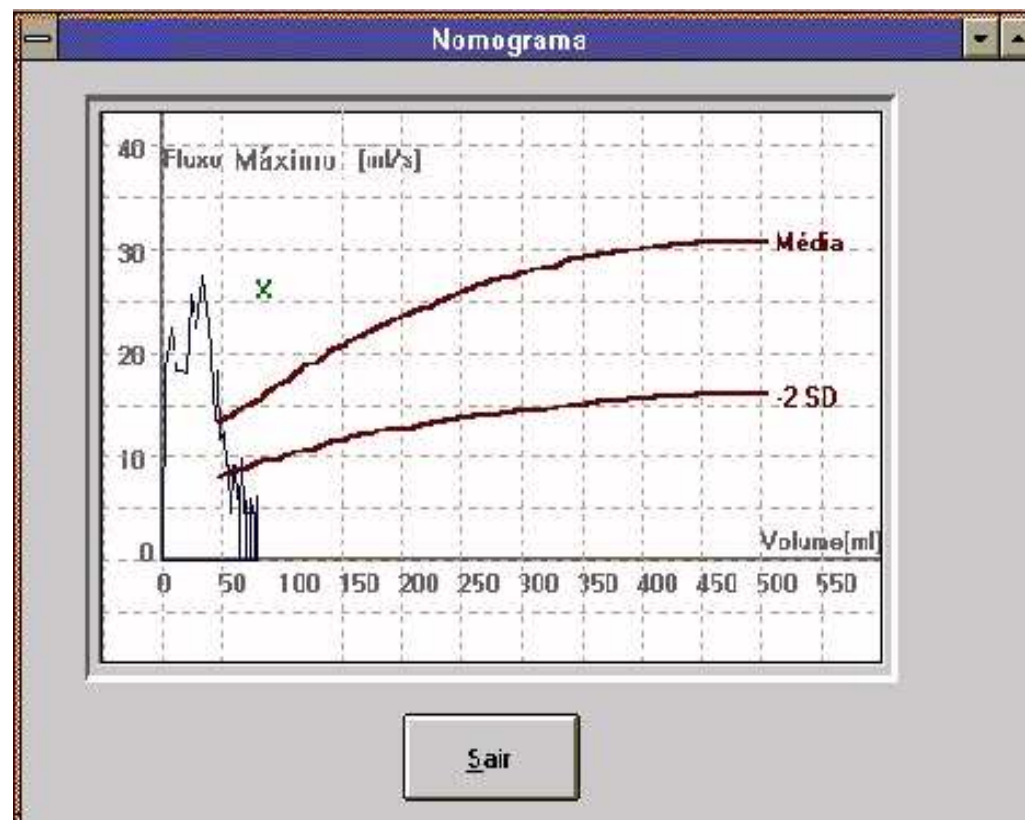
Metodologia do Sistema VUCOM





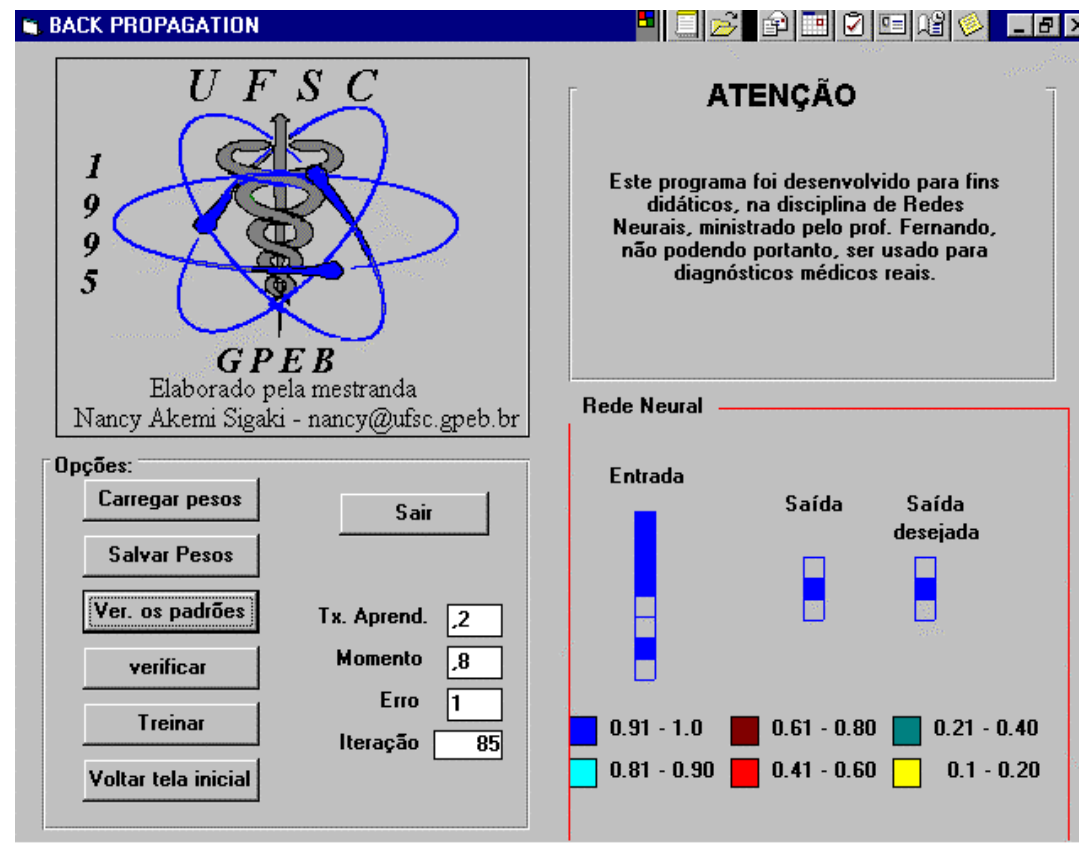
Vídeo-Urofluxômetro Computadorizado

Janelas Principal e de Nomograma





Software para diagnóstico clínico de algumas doenças comuns





Software para diagnóstico clínico de algumas doenças comuns

MEMÓRIA ASSOCIATIVA BIDIRECIONAL

Sintomas	Doenças
<input checked="" type="checkbox"/> Febre	<input type="checkbox"/> Meningite
<input checked="" type="checkbox"/> Vômito	<input checked="" type="checkbox"/> Coqueluche
<input checked="" type="checkbox"/> Tosse	<input type="checkbox"/> Infecção Virótica
<input checked="" type="checkbox"/> Indisposição	
<input type="checkbox"/> Dificuldade em inclinar a cabeça	
<input type="checkbox"/> Sonolência	
<input checked="" type="checkbox"/> Recusa-se a beber líquidos	
<input type="checkbox"/> Secreção nasal	

Dúvida !!!

Verificar doenças Limpar Verificar sintomas

Sair Voltar tela inicial

Meningite

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Coqueluche

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Inf. Virótica

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Diagnóstico: Coqueluche



Software para diagnóstico clínico de algumas doenças comuns

MEMÓRIA ASSOCIATIVA BIDIRECIONAL

Sintomas

- ☒ Febre
- ☐ Vômito
- ☒ Tosse
- ☒ Indisposição
- ☐ Dificuldade em enclinar a cabeça
- ☒ Sonolência
- ☐ Recusa-se a beber líquidos
- ☒ Secreção nasal

Doenças

- ☐ Meningite
- ☐ Coqueluche
- ☒ Infecção Virótica

Dúvida !!!

Verificar doenças Limpar Verificar sintomas

Sair Voltar tela inicial

Meningite

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Coqueluche

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Inf. Virótica

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

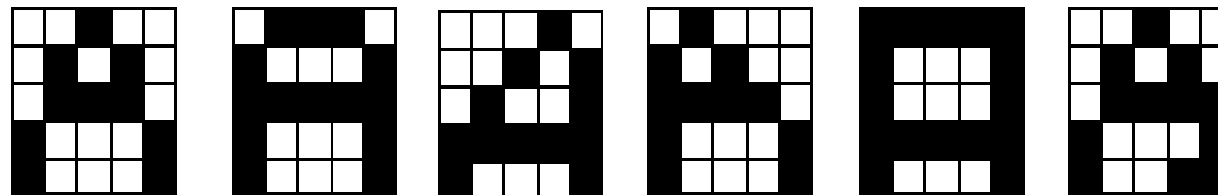
Diagnóstico: Infecção Virótica



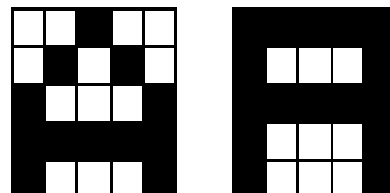
Reconhecimento de caracteres

Letra "A"

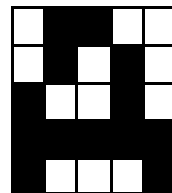
Conjunto de TREINAMENTO:



Conjunto de TESTE



Conjunto de PRODUÇÃO:

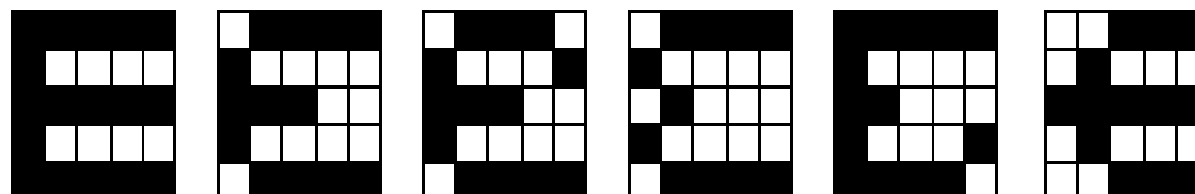




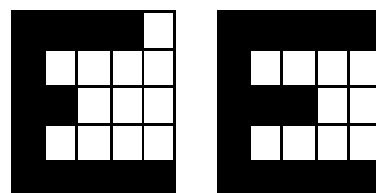
Reconhecimento de caracteres

Letra "E"

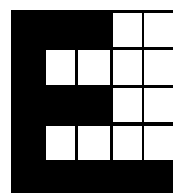
Conjunto de TREINAMENTO:



Conjunto de TESTE



Conjunto de PRODUÇÃO:

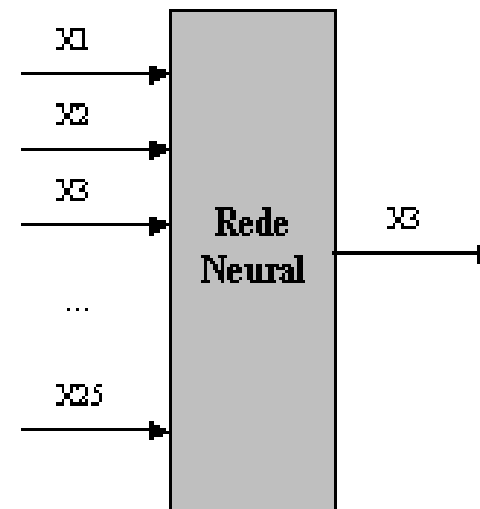
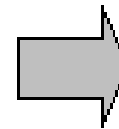




Reconhecimento de caracteres

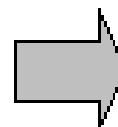
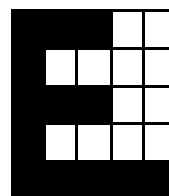
Modelagem da RNA

x_1	x_2	x_3	x_4	x_5
x_6	x_7	x_8	x_9	x_{10}
x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}



Onde BRANCO vale 1 e PRETO vale 0

Ex:

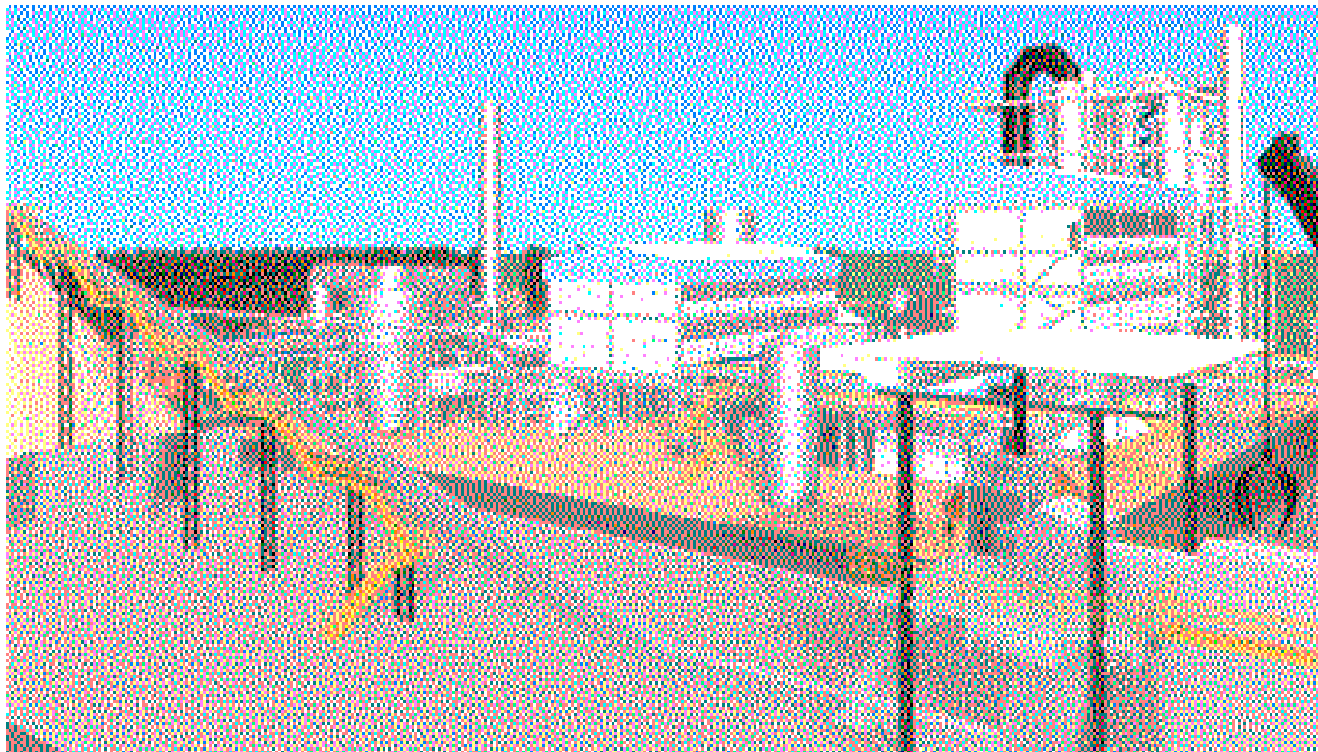


0	0	0	1	1
0	1	1	1	1
0	0	0	1	1
0	1	1	1	1
0	0	0	0	0



Aplicações: automação de sistemas complexos

- Como modelar os componentes do sistema e dar-lhes autonomia?
- Como assegurar boa comunicação e coordenação entre esses componentes?





Aplicações: sistemas de controle

- Como frear o carro sem as rodas deslizarem em função da velocidade, atrito, etc.?
- Como focar a câmera em função de luminosidade, distância, etc.?
- Como ajustar a temperatura em da quantidade de roupa, fluxo de água, etc.?





Aplicações: previsão

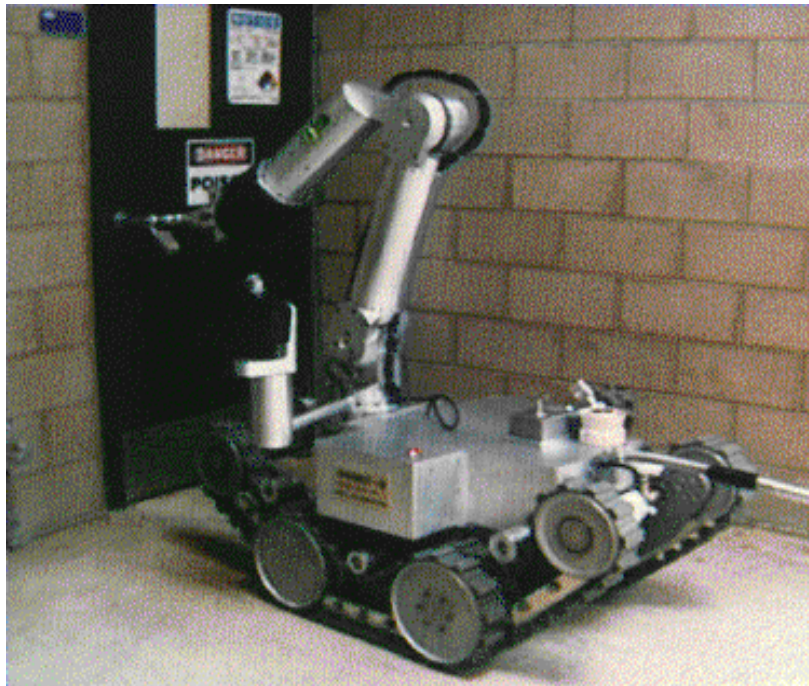
- Como prever o valor do dólar (ou o clima) amanhã?
- Que dados são relevantes? Há comportamentos recorrentes?





Aplicações: controle de robôs

- Como obter navegação segura e eficiente, manipulação fina e versátil, autonomia?
- E no caso de ambientes dinâmicos e imprevisíveis?



HAZBOT: trabalham em ambientes com atmosfera inflamável



APLICAÇÕES

Eletrônica

Chips cada vez menores são o caminho, mas limites da miniaturização estão no horizonte

Pesquisadores estimam que o limite esteja entre 16 e 11 nanômetros. "Nesse ponto não deverá ser mais prático ou econômico fazer as coisas menores, ainda que, em teoria, isso possa ser possível," diz coordenador do projeto europeu NanoCMOS. 11/01/2008



Nanotecnologia

Nanotubos de boro superam desempenho de nanotubos de carbono na eletrônica

Cientistas chineses descobriram que nanotubos construídos com o elemento boro podem ser melhores do que os famosos nanotubos de carbono para a construção de uma nova geração de equipamentos eletrônicos mais rápidos e com menor consumo de energia. 08/01/2008



Eletrônica

Transistor orgânico de alto desempenho é fabricado com moléculas de Carbono 60

Os transistores foram feitos de C_{60} , um fulereno conhecido como *buckyball*. Os resultados poderão encorajar a indústria a expandir o uso dos componentes orgânicos em telas flexíveis e etiquetas RFID. 27/11/2007



Nanotecnologia

Medição na condução elétrica avalia desempenho de nanotubos de carbono

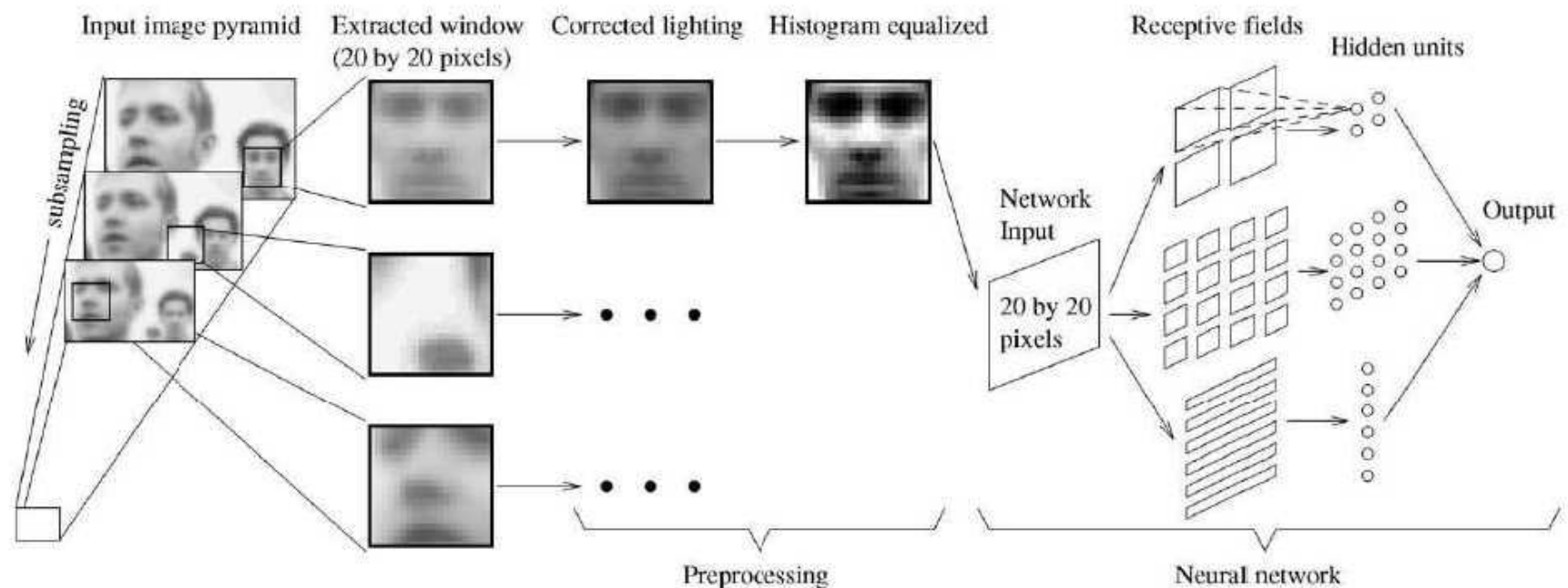
Cientistas da IBM mediram a variação de condução elétrica ao longo de um único nanotubo de carbono, determinando as características necessárias para que eles possam ser utilizados para substituir os tradicionais transistores de silício. 22/10/2007





Aplicações - Reconhecimento de Faces

- ✓ As RNA são empregadas no reconhecimento faces pela sua capacidade de aprendizado que permite tratar eficientemente essa complexa tarefa.
- ✓ Uma desvantagem dessa arquitetura é necessitar de muitos ajustes (número de níveis, número de nós, razões de aprendizado, etc.) para obter bom desempenho.
- ✓ Métodos utilizando RNA.



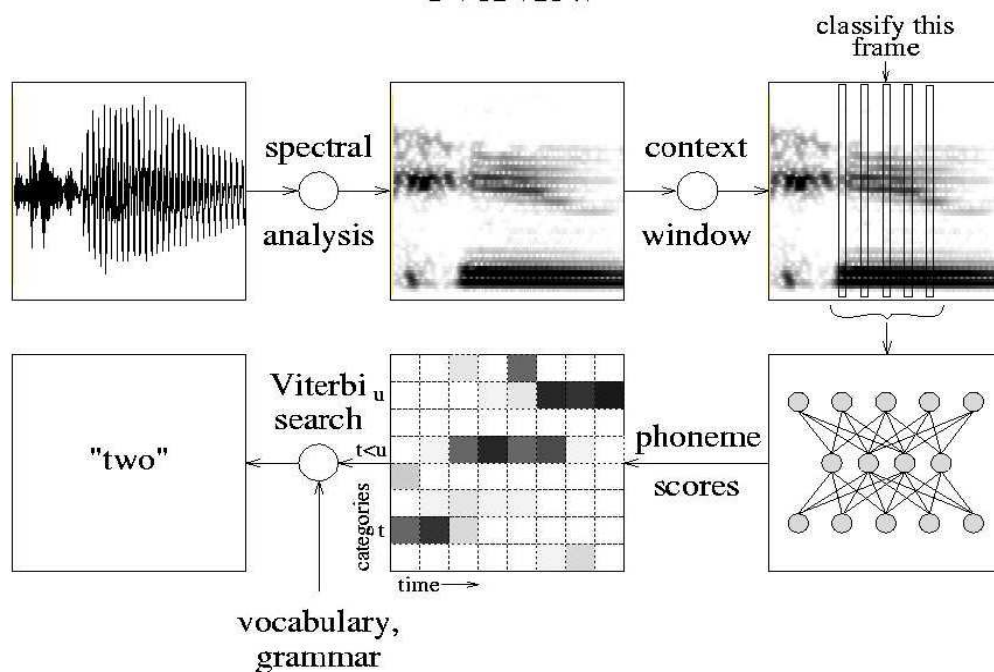
Aplicações - Reconhecimento de Voz

- ✓ Para que seja possível realizar o reconhecimento são seguidos quatro passos básicos.
 - ✓ Primeiramente ocorre a digitalização da fala que se quer reconhecer.
 - ✓ Um segundo passo realiza uma computação das características que representam o domínio espectral contido na fala (regiões de alta energia em frequências particulares). Esse passo é computado a cada 10 ms, sendo que cada seção de 10 ms é chamada de quadro (*frame*).

✓ O terceiro passo consiste do uso de uma rede neural para realizar a classificação de um conjunto de características dentro das características fonéticas básicas para cada *frame*.

✓ O quarto passo aplica um método de busca para associar as saídas da RNA com padrões de palavras a fim de encontrar a palavra com a qual as saídas da rede mais se assemelham.

✓ Um exemplo de método de busca é o *Viterbi search*, o qual foi empregado no exemplo apresentado aqui.

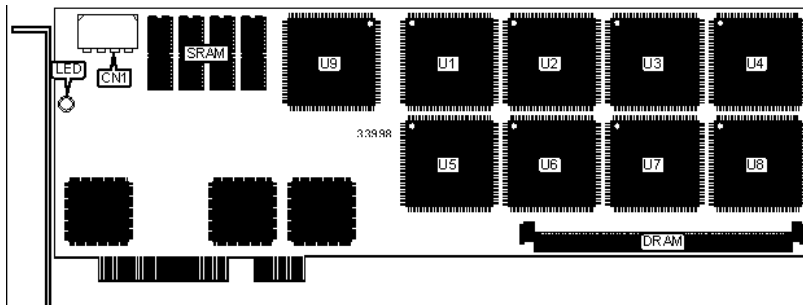




Aplicações - Hardware

- Três sistemas de *hardware* de RNA.
- Duas das iniciativas são comerciais e uma desenvolvida por um grupo de pesquisa universitária.

➤ Soluções Adaptativas CNAPS



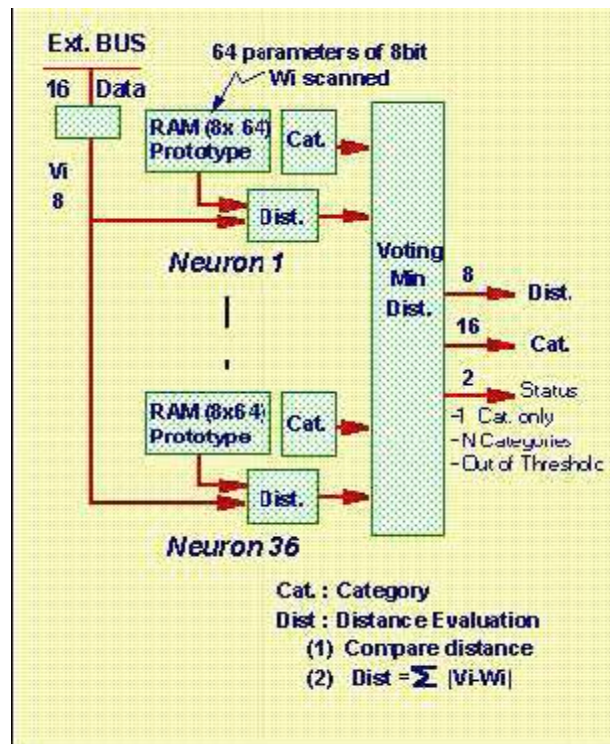
- ✓ CNAPS-1064 é um circuito eletrônico desenvolvido por Dan Hammerstron que teve sua primeira iniciativa nos anos 80 com o circuito Inova N64000.
- ✓ O CNAPS-1064 possui 80 elementos de processamento mas apenas 64 deles estão habilitados, os quais operam de forma SIMD.
- ✓ Cada subprocessador tem 4 *Kbytes* de memória local e uma unidade de aritmética de ponto fixo de 1 *bit* , 8 *bits* ou 16 *bits*.
- ✓ O circuito não apresenta unidade de aritmética de ponto flutuante nem possibilita a realização de divisões.
- ✓ Cada sub-processador pode emular um ou mais neurônios e múltiplos circuitos podem ser integrados.



Aplicações - Hardware

➤ IBM ZISC

- O circuito eletrônico ZISC036 foi desenvolvido pelo Laboratório IBM Essonnes em 1994.
- Cada circuito possui 36 neurônios ou protótipos com 64 entradas de 8 *bits*.
- Esse circuito é montado em três formatos de placas:



- Placa ISA: 16 circuitos com 576 protótipos (é permitido associar outras placas para obter protótipos adicionais);
- SIZM: 6 circuitos em uma placa do tipo SIMM, visando ser um sistema de fácil expansão;
- Placa PCI: 1 circuito com *slots* de expansão para 3 placas SIZM possibilitando 684 protótipos.

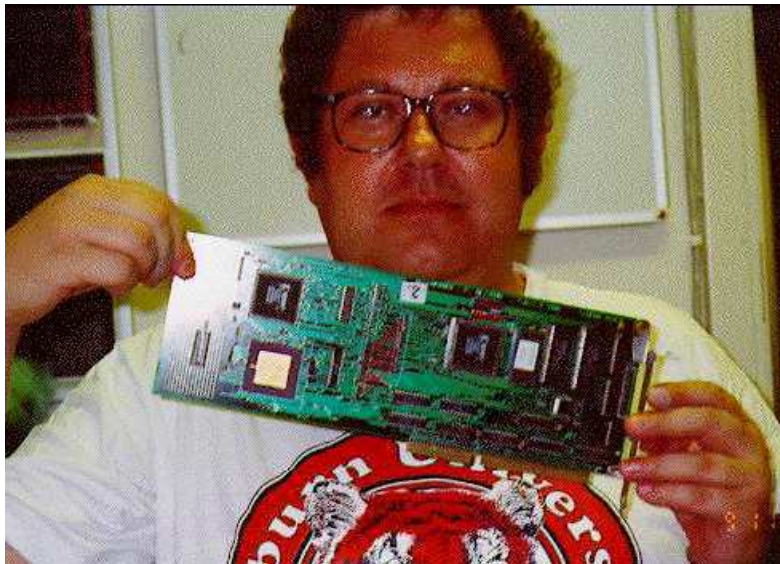




Aplicações - Hardware

➤ TOTEM

- O circuito eletrônico TOTEM foi desenvolvido pela Universidade de Trento, o qual oferece bom desempenho para RNA *feedforward*.
- Esse circuito foi desenvolvido particularmente para a execução dos algoritmos de *Reactive Tabu Search* ou RTS.



- O circuito é composto por 32 processadores simples que trabalha com uma razão de 30 MHz obtendo um desempenho de 1 bilhão de operações por segundo.
- Cada um dos processadores possui uma região de memória de 128 x 8 *bits* e um acumulador de 32 *bits*.
- Múltiplos circuitos podem ser combinados para construir grandes redes.



Aplicações

Fazer com que um robô possa
“pensar”, “tomar decisões”,
etc., por ele próprio...



Robô da série de TV “Perdidos no Espaço” dos anos 60's.



Aplicações

Exemplo de aplicação da Robótica

O cirurgião fica sentado atrás do painel manipulando os instrumentos no paciente. Um cirurgião assistente monitora e assiste através dos procedimentos.





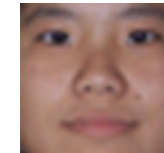
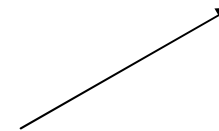
Classificação de padrões

A tarefa de classificação de padrões é atribuir a um padrão de entrada (como uma imagem facial, forma de onda vocal, ou símbolo manuscrito) representado por um vetor de fatores, a uma das classes pré-especificadas.

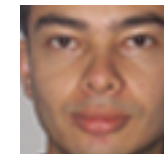
Aplicações conhecidas incluem reconhecimento de faces, caracteres, reconhecimento vocal ou da fala, classificação de formas de onda de eletroencefalograma (EEG), classificação de células sanguíneas, e inspeção de cartões de circuito impresso.



Imagem de entrada



1



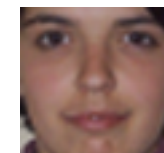
2



3



4



5



6

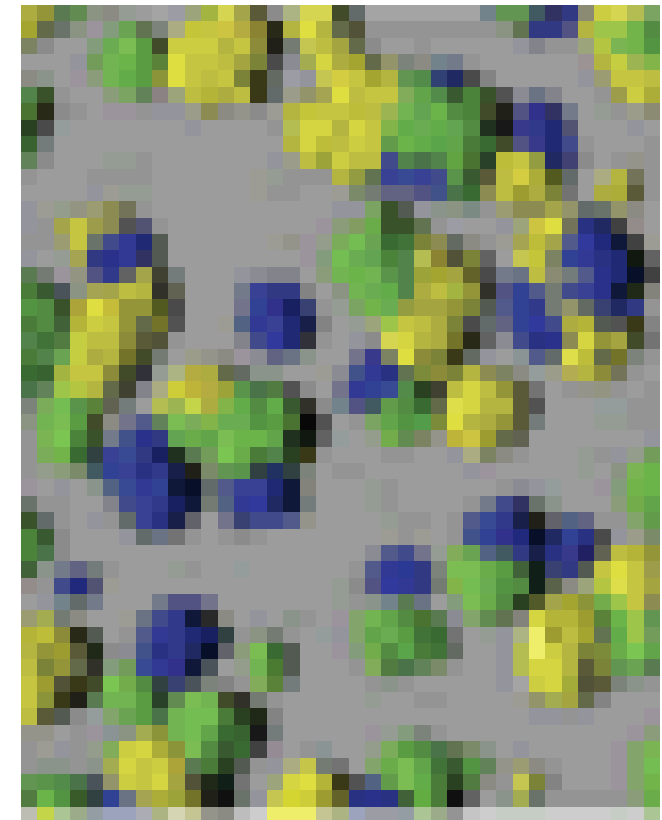


Clustering / categorização

Em *clustering* ou agrupamento, também conhecido como classificação de padrões não-supervisionada, não existem dados para treinamento com classes conhecidas.

Um algoritmo de clustering explora a similaridade entre os padrões e coloca os padrões similares num grupo.

Aplicações conhecidas de clustering incluem mineração de dados, compressão de dados, e análise exploratória de dados.

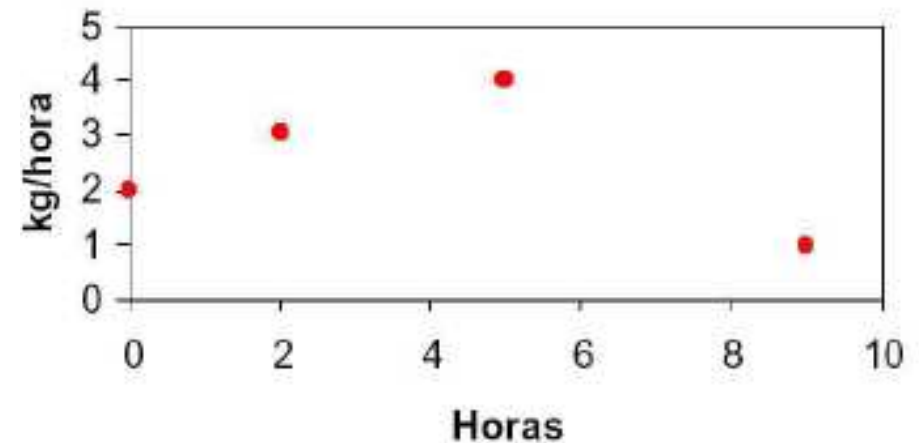




Aproximação de funções

A tarefa da aproximação de função é achar uma estimativa da função desconhecida.

Vários problemas de modelamento em engenharia e ciência requerem aproximações de funções.





Previsão(*forecasting*)

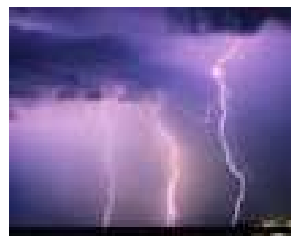
Dado um conjunto de n amostras ($y(t_1), y(t_2), \dots, y(t_n)$) em uma sequência de tempo t_1, t_2, \dots, t_n , a tarefa é de prever a amostra $y(t_{n+1})$ num futuro t_{n+1} .

A previsão (*forecasting*) tem um impacto significativo na tomada de decisões em negócios, ciência, e engenharia.

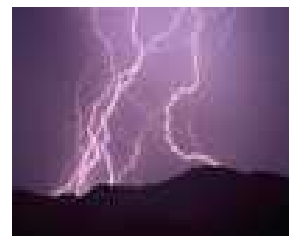
Previsão de estoques e de tempo são aplicações típicas.



t_1



t_2

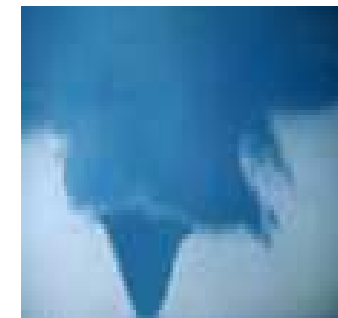


.....



t_n

t_{n+1}
previsão





Otimização

Uma grande variedade de problemas em matemática, estatística, engenharia, ciência, medicina e economia podem ser classificadas como problemas de otimização.

A meta de um algoritmo de otimização é achar uma solução que consiste numa “função objetiva” gerada por um conjunto de restrições que seja maximizada ou minimizada.

O problema do vendedor ambulante (*traveling salesman problem*) é um exemplo clássico.



Memória endereçável por conteúdo

No modelo de von Neumann de computação, uma entrada na memória é acessada somente através do endereço, e não em função do conteúdo na memória.

Assim, se um pequeno erro ocorre quando é feito o cálculo de endereço, é obtido um item completamente diferente.

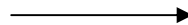
A memória associativa ou memória endereçável por conteúdo, é acessada pelo conteúdo.

O conteúdo na memória pode ser recuperado mesmo usando uma entrada parcial ou distorcida do conteúdo.

A memória associativa é extremamente desejável na construção de base de dados de informações multimídia.



entrada parcial

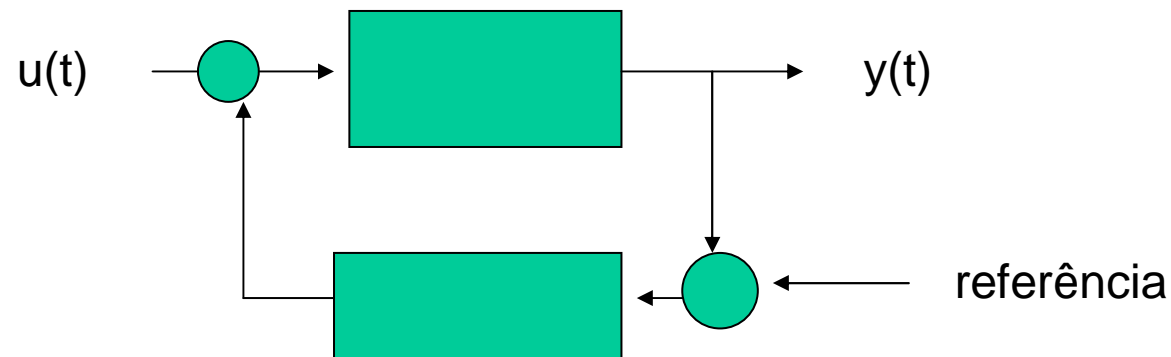


conteúdo recuperado



Controle

Considerando um sistema dinâmico definido por um $(u(t), y(t))$, onde $u(t)$ é a entrada do sistema de controle e $y(t)$ é a saída, no tempo t .



No controle adaptativo referenciado por padrão, a meta é gerar uma entrada $u(t)$ tal que o sistema siga uma trajetória desejada determinada por um padrão de referência.



processamento essencialmente paralelo

- ◆ os neurônios no cérebro humano são muito mais lentos que os *chips* usados em computadores convencionais.
- ◆ mas o cérebro humano trabalha velozmente em muitas tarefas de processamento de informação.
- ◆ isso acontece porque no cérebro as unidades trabalham essencialmente em paralelo, e cada unidade é capaz de realizar várias funções simultaneamente a outras unidades.
- ◆ os projetos baseados numa CPU essencialmente requerem ações realizadas uma a uma, sequencialmente e serialmente.



Até que ponto as redes neurais artificiais devem imitar o cérebro humano?

- ◆ Comparando aves (naturais) e aviões (artificiais), ambos tem asas, porém, enquanto as aves voam agitando as suas asas, os aviões voam sem agitar as suas asas.
- ◆ Até que ponto os cérebros artificiais devem ser projetados baseados na estrutura dos cérebros reais?
- ◆ Atualmente os cientistas ainda não tem certeza dos princípios essenciais que devem ser incorporados num cérebro artificial.
- ◆ Em relação ao avião, sabe-se que a agitação das asas não é um princípio essencial, porém, ter asas é essencial. Similarmente, ter interconexões entre os processadores é um princípio essencial, mas usar processos químicos e elétricos para as interações entre os processadores, não é um princípio essencial.



Resumo

- Importância das RNA em nosso convívio diário
- Habilidade de treinamento e generalização
- Variedades de arquiteturas conexionistas
- Problema de separabilidade linear (funções não linearmente separáveis)
- Aplicabilidade
- Dificuldades



Bibliografia

- **BÁSICA**

- HAYKIN, S. *Redes Neurais – Princípios e Prática*. 2 ed. Porto Alegre: Bookman, 2001.
- RUSSEL, S. e NORVIG, P.. *Inteligência Artificial*. 2a ed. Rio de Janeiro: Elsevier, 2004.
- AZEVEDO, F.M., BRASIL, L.M. e OLIVEIRA, R.C.L. *Redes Neurais com Aplicações em Controle e em Sistemas Especialistas*, Florianópolis: Visual Books, S.C., 2000.
- BRAGA, A.P., LUDERMIR, T.B. e CARVALHO, A.C.P.L.F. *Redes Neurais Artificiais: Teoria e Aplicações*. Rio de Janeiro: LTC, 2000.
- RICH, E. e KNIGHT, K.. *Inteligência Artificial*, Segunda Edição, Makron Books do Brasil Editora: São Paulo, 1994.



Bibliografia

- **Complementar**

- **BARRETO, J.M. *Inteligência Artificial: No Limiar do Século XXI - Abordagem Híbrida*, Terceira Edição, Florianópolis: Duplic Prestação de Serviços, 2001.**
- **KOVACS, Z.L. *Redes Neurais Artificiais: Fundamentos e Aplicações*, São Paulo: Edição Acadêmica, 1996.**
- **NASCIMENTO, C.L. e YONEYAMA, T. *Inteligência Artificial em Controle e Automação*, São Paulo: Editora Edgard Blücher Ltda., 2000.**
- **ZURADA, J.M. *Introduction to Artificial Neural Systems*, St. Paul: West Publishing Company, Inc., 1992.**



Bibliografia

- **Sites Relevantes**
 - **Simuladores**
 - » **EASYN** (<http://www.easynn.com>)
 - » **Neuroimitator**
(www.din.uem.br/ia/sombra/novas/redesneurais/download.html)
 - » **NeuroSolutions**
(terrabrasil.softonic.com/ie/12895/NeuroSolution)
 - **Aplicativos**
 - » www.unicamp.br
 - » www.inf.ufsc.br/~13c
 - » www.inf.ufsc.br/~barreto
 - » www.lia.ufc.br
 - » www.eps.ufsc.br
 - » www.gpeb.ufsc.br/~lim
 - » www.ele.ita.br/cnrn/links.html