

Redes Neurais Artificiais[†]

Prof. Carlos H. C. Ribeiro
ITA – Divisão de Ciência da Computação

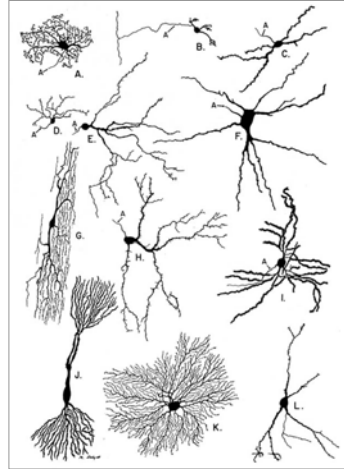
[†] Parcialmente baseado em material fornecido pelo prof. Reinaldo Bianchi (FEI), a quem o autor é grato.

Objetivos

- Descrever as características básicas das Redes Neurais Artificiais (RNAs).
- Descrever o funcionamento de uma RNA.
- Descrever algumas das principais arquiteturas existentes para RNAs.
- Exemplificar o uso de RNAs em Reconhecimento de Padrões e Controle.

Motivação Biológica

- Idéia central:
 - utilizar neurônios biológicos como modelos para neurônios artificiais.
- Neurônio Biológico:
 - Elemento fundamental do sistema nervoso.
 - Diversos tipos.
- Cérebro humano: relativamente lento, muito paralelismo (10^{11} processadores de 1 KHz)



Motivação de Engenharia

- IA tradicional (simbólica):
 - Adequada para problemas expressos em uma lógica apropriada (*ad hoc*).
 - Pouca ênfase na comunicação entre o agente e o mundo (problema de *grounding*).
 - Pouca ênfase em aprendizagem ou adaptação.
 - Ênfase em processamento sequencial.
- Redes Neurais Artificiais:
 - Adequada para tarefas “simples e diretas” como andar, falar, ouvir, ver → ênfase em processos que envolvem interação entre agente e mundo.
 - Ênfase em aprendizagem e adaptação.
 - Ênfase em processamento paralelo.

Características das Redes Neurais Naturais

- Grande número de elementos de processamento. (cérebro = 10^{11})
- Grande número de conexões ponderadas entre os elementos.
 - cada neurônio se conecta com outros 10^4 .
- Controle altamente distribuído e paralelo: é mais robusta a falhas de neurônios individuais.
- Relação entrada-saída em cada neurônio é não-linear.
- **Aprendizagem automática!**
- As RNAs são uma aproximação grosseira de um “cérebro”.

Um pouco de História

1950s & 1960s: Infância

1943 - Neurônio de McCulloch-Pitts
1959 - Perceptrons (Rosenblatt)
1960 - Adaline (Widrow & Hoff)

1970s: Os anos negros...

1969 - Minsky and Papert (MIT) publicam um livro que põe em dúvida a capacidade de aprendizado dos perceptrons.
1972 - Redes auto-organizáveis (Kohonen)
1976 - Teoria de Ressonância Adaptativa (Grossberg & Carpenter)

1980s: Entusiasmo renovado

1986 - Surgimento do algoritmo *backpropagation* (Werbos)
1987 - Redes de Hopfield (Hopfield & Tank)
1988 - Redes Neocognitron (Fukushima), RBF (Broomhead e Lowe)

1990s: Amadurecimento

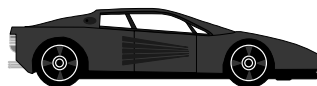
- Aplicações no mundo real
- Teoria computacional da aprendizagem: *support vector machines* (Vapnik), aprendizado PAC (Anthony e Biggs).

Aplicações



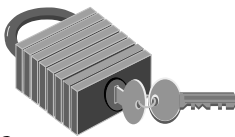
Identificação de alvos militares:
B-52, Boeing 747 e Space Shuttle

Reconhecimento
de Faces



Controle de Navegação Autônoma de Veículos:
ALVINN at CMU

2002



Autenticação de
usuário.

Exploração de petróleo:
Determinação e Litologia

Predição no
Mercado
Financeiro



CTC15 / CT215

7

Referências Bibliográficas

- A Bíblia:
 - Simon Haykin. Neural Networks: A Comprehensive Foundation.
- Livros de IA com bom capítulo de RNAs:
 - Russel e Norvig. Artificial Intelligence: A Modern Approach.
- Livro em Português:
 - Braga, Ludermir e Carvalho. Redes Neurais Artificiais. LTC.

Referências adicionais (CT215)

- Enfoque alternativo:
 - Chris Bishop. Neural Networks for Pattern Recognition. *Ênfase no enfoque de processamento probabilístico de RNAs, muito bom para quem quer se aprofundar em reconhecimento de padrões.*
- Enfoque biológico:
 - Michael Arbib (ed.). The Handbook of Brain Theory and Neural Networks.
- Principal referência histórica (provavelmente):
 - Rumelhart e McClelland (eds.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vols. 1 e 2.
- Uma crítica (entre muitas):
 - Fodor e Pylyshyn. "Connectionism and cognitive architecture: a critical analysis", Cognition, vol. 28, pp. 3-72.

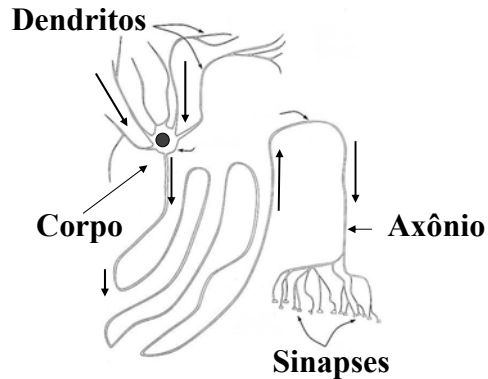
2002

CTC15 / CT215

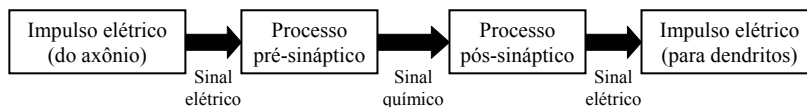
8

Estrutura simplificada de neurônio biológico

- Um neurônio biológico é composto de 4 partes principais:
 - Corpo.
 - Axônio: envia impulsos elétricos.
 - Dendritos: recebem informação (impulsos elétricos) provenientes de outros neurônios).
 - Sinapse: ponto de contato entre neurônios.



Ação sináptica:



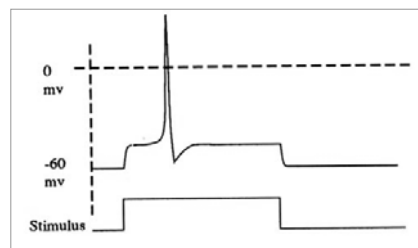
2002

CTC15 / CT215

9

Funcionamento de um Neurônio Biológico

- Neurônios podem estar em 2 estados:
 - Ativo ou excitado: envia sinais para outros neurônios por meio do axônio e sinapses.
 - Inativo ou inibido: não envia sinais.
- Sinapses podem ser de 2 tipos:
 - Excitatórias (excitam o neurônio receptor).
 - Inibitórias (inibem o neurônio receptor).
- Quando o efeito cumulativo das várias sinapses que chegam a um neurônio excedem um valor limite, o neurônio dispara (fica ativo por um período) e envia um sinal para outros neurônios.



2002

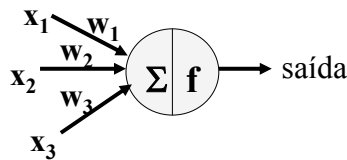
CTC15 / CT215

10

Modelo de Neurônio Artificial

■ Um Neurônio Artificial é composto por:

- X_i : o conjunto de entradas.
- W_{ji} : o conjunto de pesos (um para cada sinapse).
- Nível de Ativação: soma ponderada do vetor de entradas.
- Função de Ativação: função não-linear aplicada sobre o nível de ativação para produzir o sinal de saída. Se a saída for diferente de zero, ela é transmitida para outros neurônios.

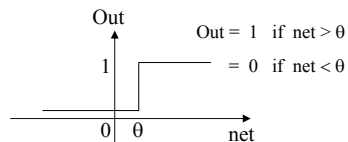


w = Vetor de pesos.
 x = Vetor de entradas.
 Σ = Soma ponderada das entradas
 f = função de ativação

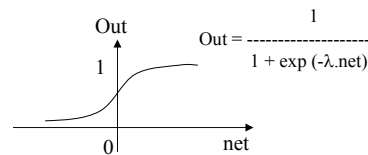
$$\text{saída} = f(\mathbf{w}^t \cdot \mathbf{x}) = f[(w_1 \cdot x_1) + (w_2 \cdot x_2) + (w_3 \cdot x_3)]$$

Tipos de Função de Ativação

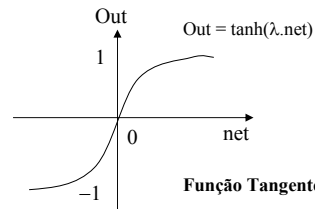
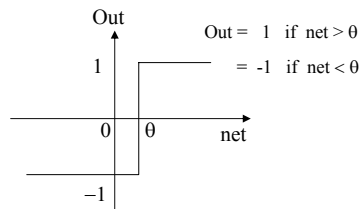
Função Limiar



Função Sigmoide



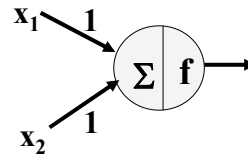
Função Logística



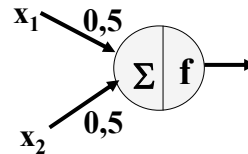
Função Tangente Hiperbólica

Exemplos de neurônio

- $x_1, x_2 \in \{0, 1\}$
- $w_1 = w_2 = 1$
- $f(\mathbf{w}^t \cdot \mathbf{x}) = 1$ se nível de ativ. ≥ 2 ,
0 caso contrário.



- $x_1, x_2 \in \{0, 1\}$
- $w_1 = w_2 = 0,5$
- $f(\mathbf{w}^t \cdot \mathbf{x}) = 1$ se nível de ativ. $\geq 0,5$,
0 caso contrário.



O que é uma RNA?

- É um sistema de processamento de informações desenvolvido a partir de modelos matemáticos simplificados dos neurônios biológicos.
- É constituída por um grande número de unidades de processamento (chamadas neurônios artificiais) interconectadas.
- Reconhecem, classificam, convertem e aprendem padrões.

Processamento Neural

- O processo de calcular a saída de uma rede neural dada uma entrada é chamado **recall**.
- O objetivo do **recall** é recuperar a informação armazenada na rede.
- Exemplos de processamento realizados por uma rede neural:
 - Associação de **padrões**
 - Classificação de **padrões**.
 - Reconhecimento de **padrões**.
 - Generalização.

2002

CTC15 / CT215

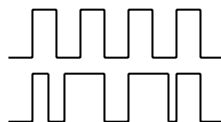
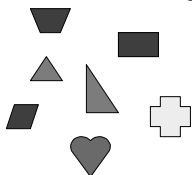
15

Padrões, atributos e classes

- Um **Padrão** é uma descrição quantitativa de um objeto.
- Cada padrão é definido por um conjunto de valores de **atributos**.
- Um atributo é uma característica mensurável de um padrão.
- Uma **Classe** contém um conjunto de padrões com propriedades comuns.



001011101011
000000101110
101011111111



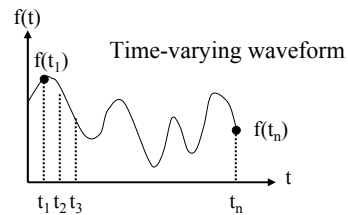
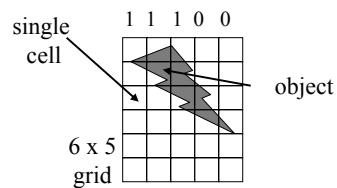
2002

CTC15 / CT215

16

Tipos de Padrões

- Padrões estáticos ou espaciais:
 - não envolvem o tempo.
- Padrões dinâmicos ou temporais:
 - envolve o tempo.



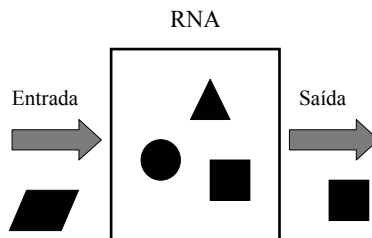
2002

CTC15 / CT215

17

Autoassociação

- Um conjunto de padrões de referência é armazenado na rede.
- Um padrão similar é apresentado e a rede associa esta entrada com o padrão armazenado mais próximo.
- O padrão de entrada pode possuir ruído ou distorções em relação ao seu modelo armazenado na rede.
- Corresponde a um processo de reconhecimento de padrões.



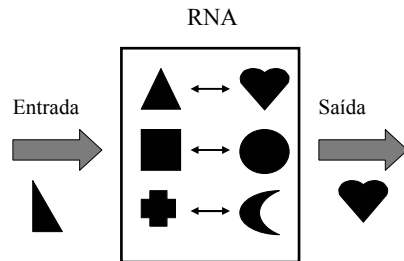
2002

CTC15 / CT215

18

Heteroassociação

- Um conjunto de pares de padrões é armazenado na rede.
- Um padrão é apresentado e a rede associa esta entrada com o seu par.
- Memória associativa: corresponde a um processo de associação de padrões.



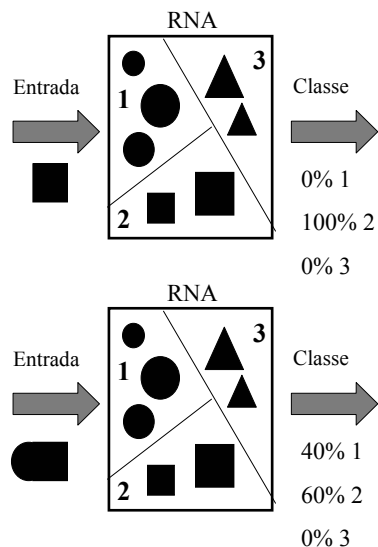
2002

CTC15 / CT215

19

Classificação

- Um conjunto de padrões é particionado em classes.
- Cada classe é definida por uma série de valores de atributos.
- Em resposta a uma entrada, a rede retorna informação sobre a pertinência do padrão apresentado a uma classe.
- A rede é treinada para classificar.



2002

CTC15 / CT215

20

Exemplo de Memória Associativa

- Memória associativa: associa o conceito A ao conceito B. (dado A, “lembra-se” de B)

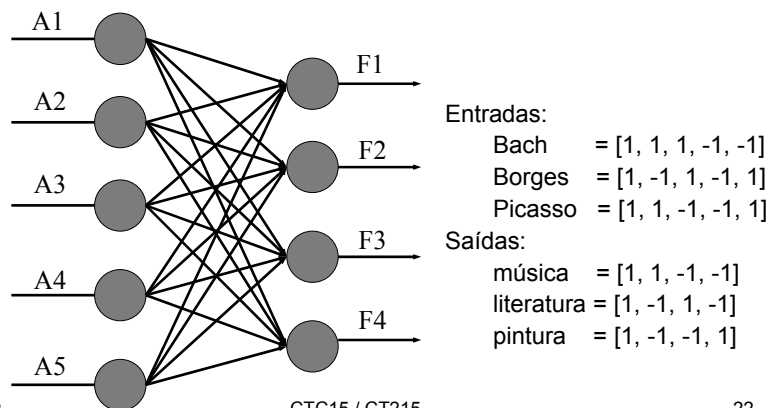
- Pares de associação:



Exemplo de RNA: Memória Associativa

- 5 neurônios de entrada e 4 de saída.
- Neurônio binário: ativado = 1, inibido = -1
- Função de ativação:

$\Sigma > 0 \rightarrow$ saída = 1, $\Sigma < 0 \rightarrow$ saída = -1, $\Sigma = 0 \rightarrow$ saída inalterada



Exemplo de RNA: Memória Associativa

■ Tabela de sinapses:

	A1	A2	A3	A4	A5
F1	3	1	1	-3	1
F2	-1	1	1	1	-3
F3	-1	-3	1	1	1
F4	-1	1	-3	1	1

■ Operação da RNA:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = f \left(\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \right) = f \left(\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \\ w_{41} & w_{42} & w_{43} & w_{44} & w_{45} \end{bmatrix} \times \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{bmatrix} \right) = f \left(\begin{bmatrix} 3 & 1 & 1 & -3 & 1 \\ -1 & 1 & 1 & 1 & -3 \\ -1 & -3 & 1 & 1 & 1 \\ -1 & 1 & -3 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{bmatrix} \right)$$

Exemplo de RNA: Memória Associativa

Operação

– Apresentando Bach:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = f \left(\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \right) = \begin{bmatrix} 3 & 1 & 1 & -3 & 1 \\ -1 & 1 & 1 & 1 & -3 \\ -1 & -3 & 1 & 1 & 1 \\ -1 & 1 & -3 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = f \left(\begin{bmatrix} 7 \\ 3 \\ -5 \\ -5 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

– Apresentando Bach “ruidoso”:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = f \left(\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \right) = \begin{bmatrix} 3 & 1 & 1 & -3 & 1 \\ -1 & 1 & 1 & 1 & -3 \\ -1 & -3 & 1 & 1 & 1 \\ -1 & 1 & -3 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.9 \\ 1 \\ 1.1 \\ -1.2 \\ -1 \end{bmatrix} = f \left(\begin{bmatrix} 7.4 \\ 2.9 \\ -5 \\ -5.4 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

CAPACIDADE DE GENERALIZAÇÃO!



Projeto de uma RNA

- O projeto de uma RNA envolve a determinação dos seguintes parâmetros:
 - Neurônios: função de ativação.
 - Conexões e disposição dos neurônios: topologia (arquitetura da rede).
 - Pesos sinápticos: valores ou (no caso de pesos aprendidos), o algoritmo a ser utilizado e seus parâmetros particulares.
 - *Recall*: procedimento a ser utilizado – como a rede calcula a saída para uma dada entrada?

Infelizmente, não existe uma “receita” única...

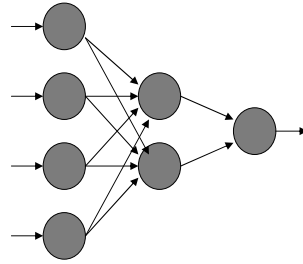
Arquiteturas das RNAs

- Os neurônios de uma RNAs devem estar conectados entre si.
- Os neurônios são dispostos em **camadas** (*layers*):
 - neurônios de uma mesma camada normalmente se comportam da mesma maneira.
- A disposição dos neurônios nas camadas e o padrão de conexão entre estas define a **arquitetura** da RNA:
 - Redes sem realimentação (*feedforward*)
 - Redes com realimentação ou recorrentes (*recurrent*)

Arquiteturas de RNAs

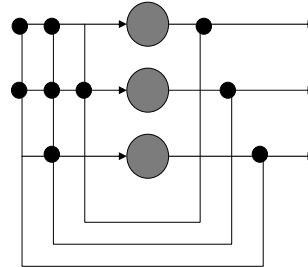
Sem realimentação:

- Neurônios agrupados em camadas.
- Sinal percorre a rede em uma única direção, da entrada para a saída.
- Neurônios da mesma camada não são conectados.
- Representante típico: Perceptron.



Recorrentes:

- Saída de alguns neurônios alimentam neurônios da mesma camada (inclusive o próprio) ou de camadas anteriores.
- Sinal percorre a rede em duas direções.
- Memória dinâmica: capacidade de representar estados em sistemas dinâmicos.
- Representante típico: rede de Hopfield



2002

CTC15 / CT215

27

Aprendizado (ou treinamento)

- O aprendizado consiste na modificação dos pesos das conexões entre os neurônios.
- Os pesos iniciais (sinapses) são modificados de forma iterativa, por um algoritmo que segue um dos seguintes paradigmas de aprendizado:
 - Aprendizado Supervisionado: é apresentado um conjunto de treino, consistindo de entradas e correspondentes saídas desejadas.
 - Aprendizado por Reforço: para cada entrada apresentada, é produzida uma indicação (reforço) sobre a adequação das saídas correspondentes produzidas pela rede.
 - Aprendizado Não-supervisionado: A rede atualiza seus pesos sem o uso de pares entrada-saídas desejadas e sem indicações sobre a adequação das saídas produzidas.

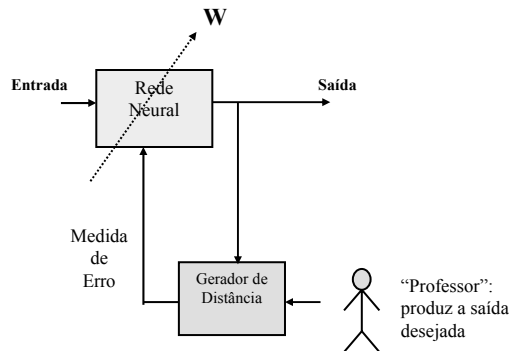
2002

CTC15 / CT215

28

Aprendizado Supervisionado

- Define-se um conjunto de treinamento (in,out) de entradas e correspondentes saídas desejadas.
- A distância entre a resposta atual e a desejada serve como uma **medida de erro** que é usada para corrigir os parâmetros da rede.
- Pesos da rede são ajustados para minimizar este erro.



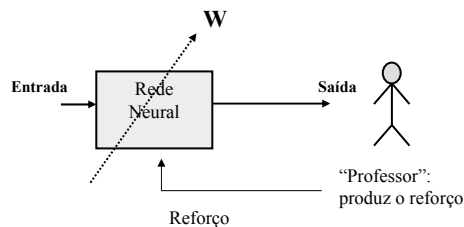
2002

CTC15 / CT215

29

Aprendizado por Reforço

- Para cada entrada apresentada, a rede calcula a saída e um "professor" indica a sua conveniência ou não.
- Não há medida explícita da distância entre resposta atual e a "desejada".
- Pesos da rede são ajustados para maximizar os reforços recebidos.
- Frequentemente, a entrada é o estado de um sistema dinâmico, a saída é a ação produzida para o estado, e o "professor" é o próprio sistema que se deseja controlar: reforço positivo = ação "boa", reforço negativo (punição) = ação "ruim"



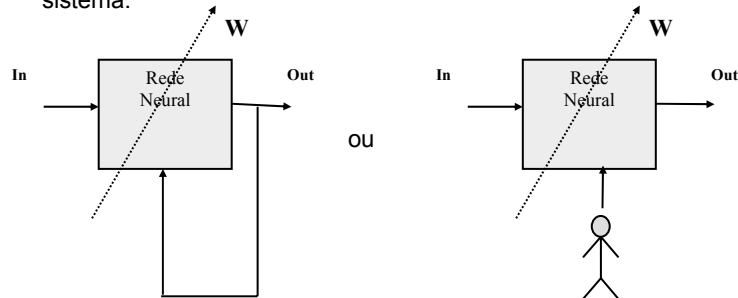
2002

CTC15 / CT215

30

Aprendizado Não-Supervisionado

- Não existe comparação entre a saída obtida pela rede e a saída “desejada”.
- Aprendizado baseado na **observação** das respostas obtidas ou da saída desejada (e não dos dois simultaneamente).
- A Rede descobre por si mesmo propriedades e regularidades do sistema.



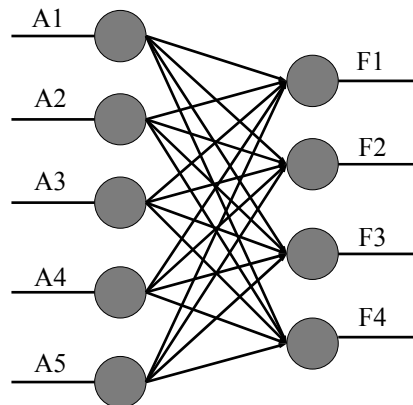
2002

CTC15 / CT215

31

Exemplo: Aprendizado em Memória Associativa

- Regra de Aprendizado de Hebb:
 - 1. Tabela W de pesos w_{ji} inicialmente zerada.
 - 2. Se F_j e A_i estiverem excitados, soma 1 em w_{ji} .
 - 3. Se F_j e A_i estiverem inibidos, soma 1 em w_{ji} .
 - 4. Se F_j estiver inibido e A_i excitado, subtrai 1 de w_{ji} .
 - 5. Se F_j estiver excitado e A_i inibido, subtrai 1 de w_{ji} .
- Pares de treinamento são apresentados um a um, uma única vez.



2002

CTC15 / CT215

32

Exemplo: Aprendizado em Memória Associativa

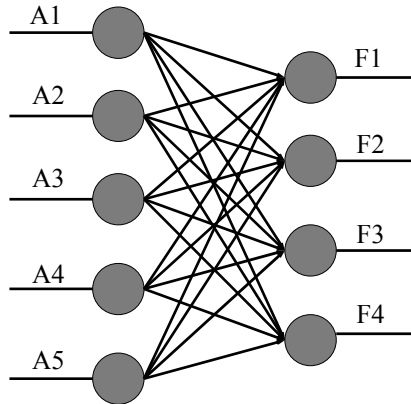
■ Tabela inicial:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

■ A) Apresento par

Borges = [1, -1, 1, -1, 1]
 literatura = [1, -1, 1, -1]

$A_1=1, F_1=1 \rightarrow w_{11}=0+1=1$
 $A_1=1, F_2=-1 \rightarrow w_{21}=0-1=-1$
 $A_1=1, F_3=1 \rightarrow w_{31}=0+1=1$
 $A_1=1, F_4=-1 \rightarrow w_{41}=0-1=-1$
 Etc...



■ Tabela após A):

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

■ Tabela após B):

$$\begin{bmatrix} 2 & 0 & 2 & -2 & 0 \\ 0 & 2 & 0 & 0 & -2 \\ 0 & -2 & 0 & 0 & 2 \\ -2 & 0 & -2 & 2 & 0 \end{bmatrix}$$

B) Apresento par

Bach = [1, 1, 1, -1, -1]
 música = [1, 1, -1, -1]

$A_1=1, F_1=1 \rightarrow w_{11}=1+1=2$
 $A_1=1, F_2=1 \rightarrow w_{21}=-1+1=0$
 $A_1=1, F_3=-1 \rightarrow w_{31}=1-1=0$
 $A_1=1, F_4=-1 \rightarrow w_{41}=-1-1=-2$
 Etc...

C) Apresento par

Picasso = [1, 1, -1, -1, 1]
 pintura = [1, -1, -1, 1]

$A_1=1, F_1=1 \rightarrow w_{11}=2+1=3$
 $A_1=1, F_2=-1 \rightarrow w_{21}=0-1=-1$
 $A_1=1, F_3=-1 \rightarrow w_{31}=0-1=-1$
 $A_1=1, F_4=1 \rightarrow w_{41}=-2+1=-1$
 Etc...

Exemplo: Aprendizado em Memória Associativa

- Pesos finais:

$$\begin{bmatrix} 3 & 1 & 1 & -3 & 1 \\ -1 & 1 & 1 & 1 & -3 \\ -1 & -3 & 1 & 1 & 1 \\ -1 & 1 & -3 & 1 & 1 \end{bmatrix}$$

- Regra de Hebb:

- Realiza aprendizado não-supervisionado

- Equação:

$$w_{ji}(t+1) = w_{ji}(t) + \alpha OUT_j OUT_i$$

- O que acontece se treinar com pares “ruidosos”?
- O que acontece se alterar a ordem de apresentação dos pares?
- O que acontece se treinar mais de uma vez com um par?

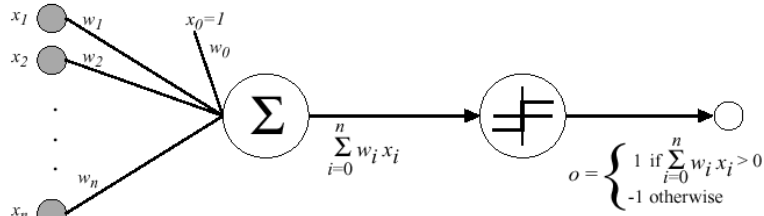
Principais arquiteturas de RNAs

- Dependem da estrutura, tipo de neurônio, regra de aprendizado, ...
 - **Perceptron**,
 - **Perceptron Multi-Camada (MLP)**,
 - **Rede de Hopfield**,
 - ADALINE e MADALINE,
 - Máquinas de Boltzmann,
 - **Rede de Kohonen**,
 - dezenas de outras...

Perceptron (McCulloch e Pitts, 1947)

- Arquitetura baseada em um neurônio discreto com função de ativação limiar:

$$f(\mathbf{w}^t \cdot \mathbf{x}) = \begin{cases} 1 & \text{se nível de ativação} > 0, \\ -1 & \text{caso contrário.} \end{cases}$$

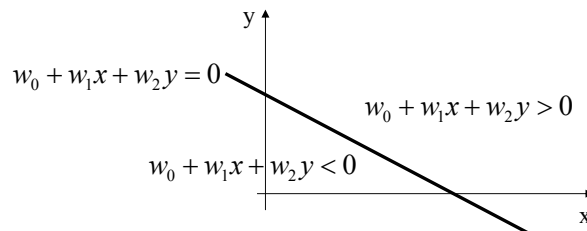
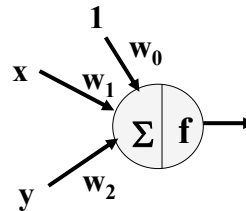


- Pode produzir saída vetorial: vários neurônios em paralelo

Operação do Perceptron: Exemplo

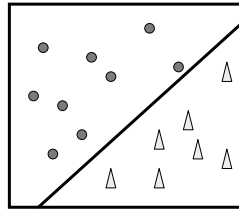
- Duas entradas: x, y
- Dois pesos: w_1, w_2
- Uma entrada de *bias*

$$o = f[w_0 + w_1 x + w_2 y]$$



Poder de representação dos Perceptrons

Conseguem representar qualquer função linearmente separável

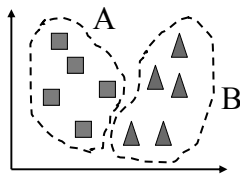


Função linearmente separável: consigo separar as classes com retas (ou planos, ou hiperplanos).

1 neurônio: classes +1 e -1

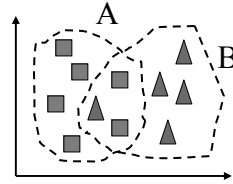
2 neurônios: classes +1+1,+1-1,-1+1,-1-1

N neurônios: 2^N classes



← OK

Não OK →

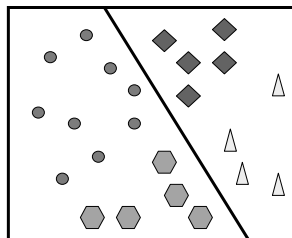


2002

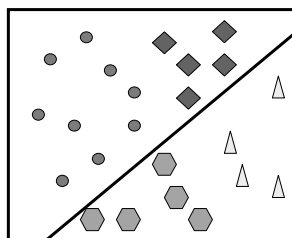
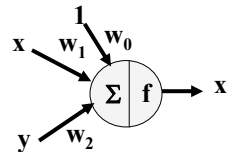
CTC15 / CT215

39

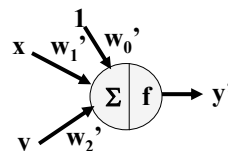
Separação em 4 classes: Exemplo



Neurônio 1: separa em classes C1 e C2



Neurônio 2: separa em classes C1' e C2'

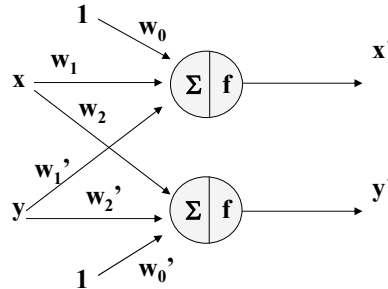
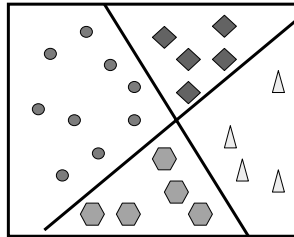


2002

CTC15 / CT215

40

Separação em 4 classes: Exemplo



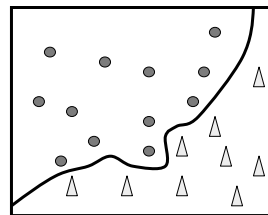
2002

CTC15 / CT215

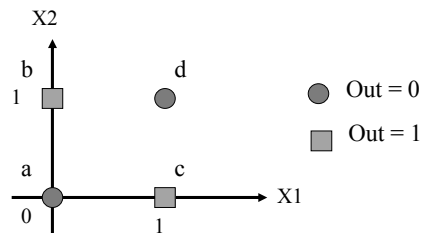
41

Limitações do Perceptron

- Infelizmente, várias funções de interesse não são linearmente separáveis.
- Por exemplo, o Perceptron não pode representar o XOR (ou exclusivo).



	X1	X2	Out
a	0	0	0
b	0	1	1
c	1	0	1
d	1	1	0



2002

CTC15 / CT215

42

Aprendizado no Perceptron: Regra Delta

- A regra de atualização dos pesos é:

$$w_i = w_i + \alpha \delta x_i$$

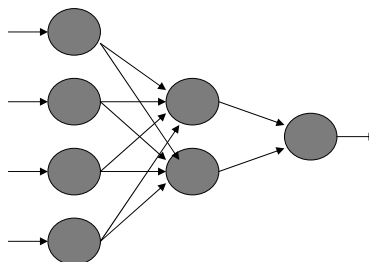
Onde $\delta = (t - o)$

- x_i é a entrada,
- t é o valor desejado,
- o é a saída do perceptron e,
- α é uma constante pequena (ex: 0.1), chamada taxa de aprendizado.

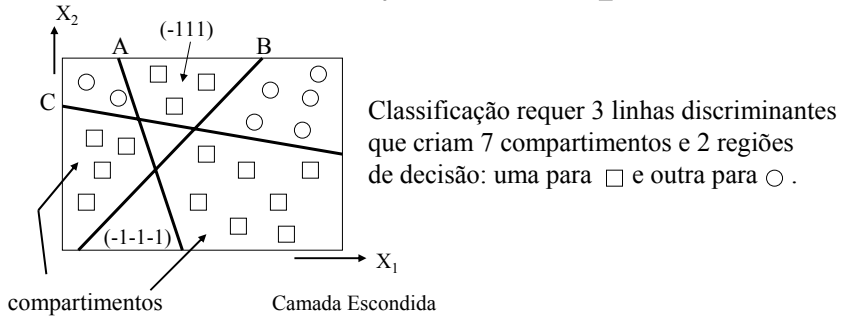
Funciona para entradas/saídas discretas ou contínuas.

MLP: Multi-layer Perceptrons

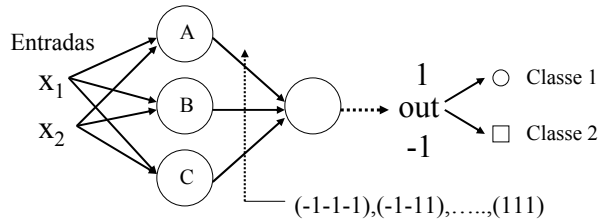
- Conjuntos de Perceptrons arranjados em diversas camadas.
- Pelo menos uma camada escondida.
- Soluciona problema de classificação não-linear: consegue formar regiões de decisão mais complexas



MLP: Multi-layer Perceptrons



Solução:



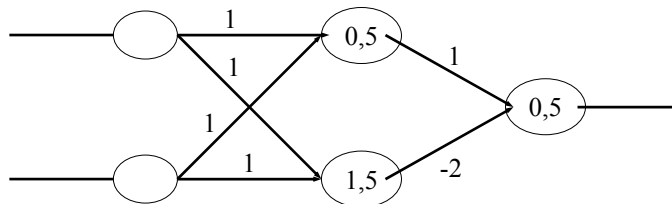
2002

CTC15 / CT215

45

Exemplo de Rede MLP: XOR

- $x_1 = x_2 =$ entradas binárias.
- $w_1 = w_2 = w_3 = w_4 = 1, w_5 = 1$ e $w_6 = -2$
- $f_1(\mathbf{w}^t \cdot \mathbf{x}) = 1$ se nível de ativação $\geq 1,5$,
0 caso contrário.
- $f_2(\mathbf{w}^t \cdot \mathbf{x}) = 1$ se nível de ativação $\geq 1,5$,
0 caso contrário.



Preciso agora de um algoritmo para treinar os pesos...

2002

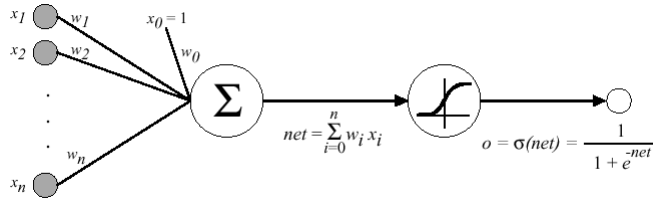
CTC15 / CT215

46

MLP Sigmóide

- Neurônio com função de ativação limiar sigmóide:

$$\sigma(net) = \frac{1}{1 + e^{-net}} \quad \text{ou} \quad \sigma(net) = \tanh(net)$$



$$\sigma(x) = \frac{1}{1 + e^{-x}} \Rightarrow \sigma'(x) = \sigma(x)(1 - \sigma(x))$$

$$\sigma(x) = \tanh(x) \Rightarrow \sigma'(x) = \frac{1 - \sigma^2(x)}{2}$$

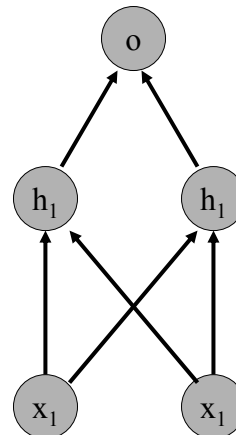
Propriedade Interessante...

Como treinar um MLP?

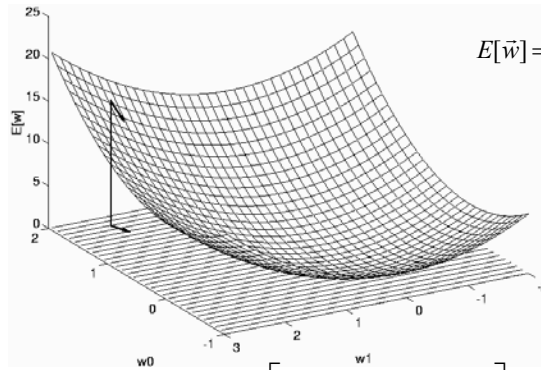
- Descida do gradiente: Aprenda w_i 's que minimizam erro quadrático:

$$E[\vec{w}] = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

$D = \text{dados de treinamento}$



Descida do Gradiente



$$E[\vec{w}] = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

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

Regra: $\Delta \vec{w} = -\eta \nabla E[\vec{w}] \quad \Delta w_i = -\eta \frac{\partial E}{\partial w_i}$

Descida do Gradiente (uma camada)

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_d (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_d \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_d 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\ &= \sum_d (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - \sigma(\vec{w} \cdot \vec{x}_d)) \\ &= \sum_d (t_d - o_d) \left(-\frac{\partial}{\partial w_i} \sigma(\vec{w} \cdot \vec{x}_d) \right) \\ &= -\sum_d (t_d - o_d) \sigma(\vec{w} \cdot \vec{x}_d) (1 - \sigma(\vec{w} \cdot \vec{x}_d)) x_{i,d} \end{aligned}$$

- Inicialize w_i 's (pequeno valor aleatório)
- Repita até condição de término:

$$\Delta w_i = 0$$

Para cada exemplo d faça

$$o_d \leftarrow \sigma(\sum_i w_i x_{i,d})$$

$$\Delta w_i \leftarrow \Delta w_i + \eta (t_d - o_d) o_d (1 - o_d) x_{i,d}$$

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

Aprendizado Batch (em lotes)

- Inicialize w_i 's (pequeno valor aleatório)
- Repita até condição de término:

Para cada exemplo d faça

$$\Delta w_i = 0$$

$$o_d \leftarrow \sum_i w_i x_{i,d}$$

$$\Delta w_i \leftarrow \Delta w_i + \eta (t_d - o_d) o_d (1 - o_d) x_{i,d}$$

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

Aprendizado Online (incremental)

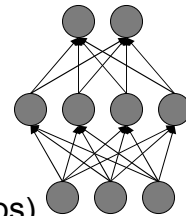
2002

CTC15 / CT215

51

Algoritmo Backpropagation

Generalização da descida do gradiente: várias camadas



- Inic. pesos (valores aleatórios pequenos)
- Para cada exemplo de treinamento:

– Para cada neurônio escondido h :
$$o_h = \sigma\left(\sum_i w_{hi} x_i\right)$$

– Para cada neurônio de saída k :
$$o_k = \sigma\left(\sum_k w_{kh} x_h\right)$$

– Para cada neurônio de saída k :
$$\delta_k = o_k (1 - o_k) (t_k - o_k)$$

– Para cada neurônio escondido h :
$$\delta_h = o_h (1 - o_h) \sum_k w_{hk} \delta_k$$

- Atualize cada peso da rede w_{ij} :

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \quad \text{onde} \quad \Delta w_{ij} = \eta \delta_j x_{ij}$$

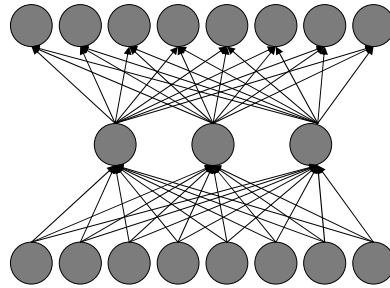
2002

CTC15 / CT215

52

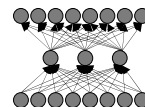
Exemplo

Entrada		Saída
10000000	→	10000000
01000000	→	01000000
00100000	→	00100000
00010000	→	00010000
00001000	→	00001000
00000100	→	00000100
00000010	→	00000010
00000001	→	00000001

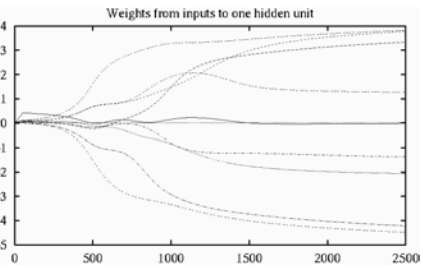
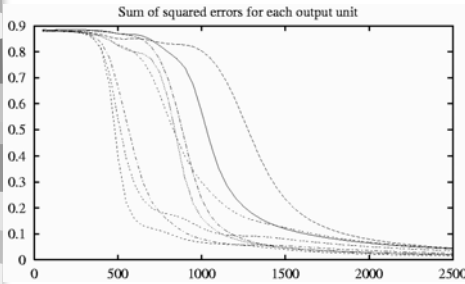
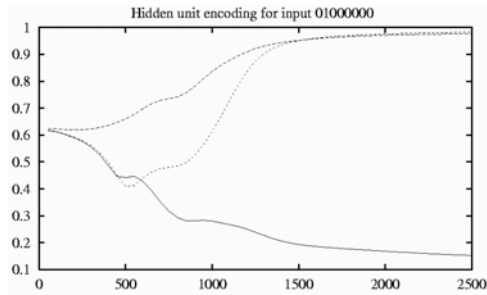


Aprendizado da representação (camada interna)

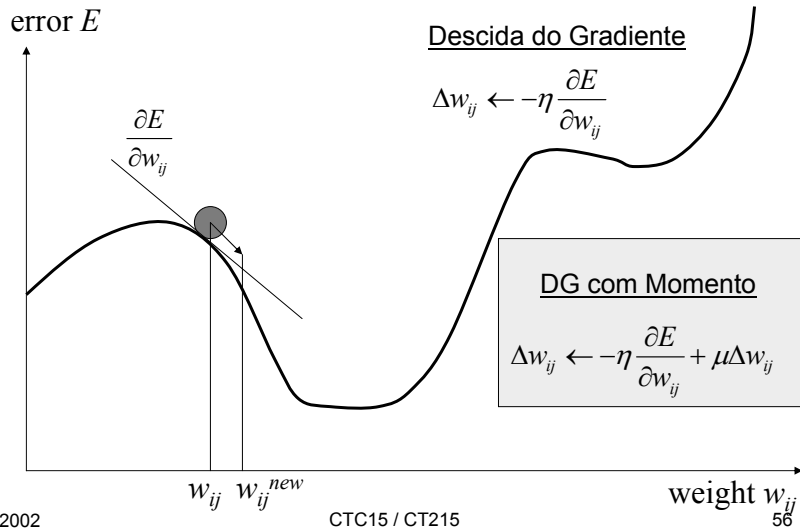
Entrada				Saída		
10000000	→	.89	.04	.08	→	10000000
01000000	→	.01	.11	.88	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.22	.99	.99	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001



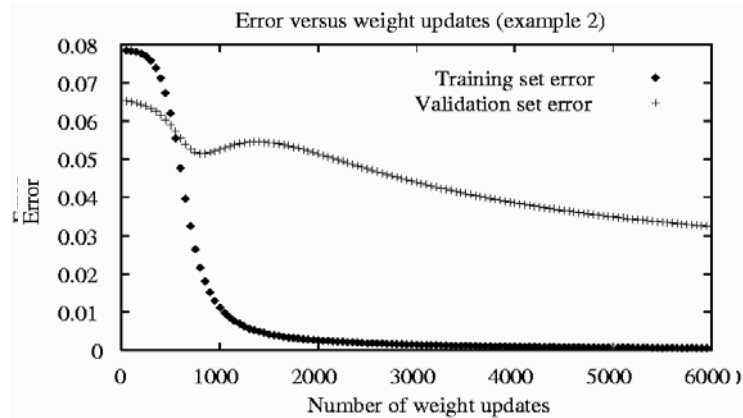
Aprendizado: Representação Interna, Erro e Pesos



Aumento da velocidade de aprendizado: Momento



Overfitting em RNAs



2002

CTC15 / CT215

57

Convergência e Critério de Parada

- Pode ficar preso em mínimo local
- Pesos podem divergir

...mas na prática, funciona bem!

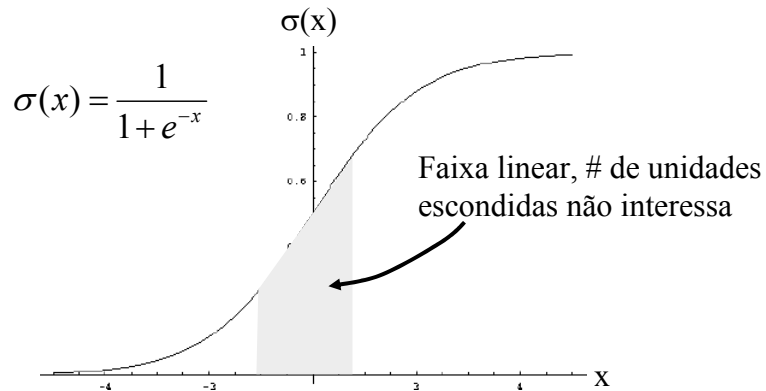
Critério de parada (para evitar *overfitting*):
parar assim que erro no conjunto de
validação começar a aumentar

2002

CTC15 / CT215

58

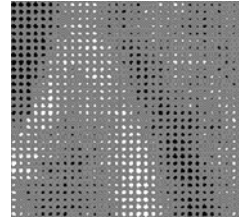
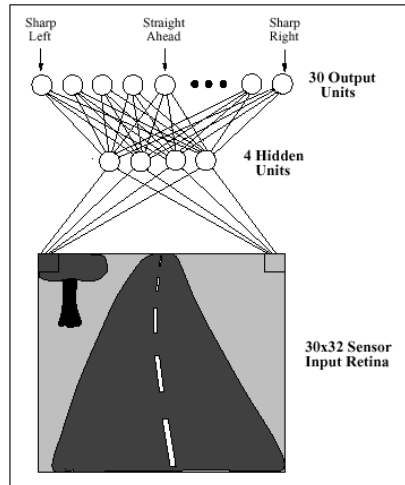
Faixa de operação



Capacidade de uma RNA MLP

- Qualquer função booleana pode ser representada por uma RNA com uma camada escondida.
- Qualquer função contínua limitada pode ser aproximada por uma RNA com uma camada escondida.
- Qualquer função contínua pode ser aproximada por uma RNA com duas camadas escondidas.

Exemplo: ALVINN



Dirige um carro a 100 Km/h.

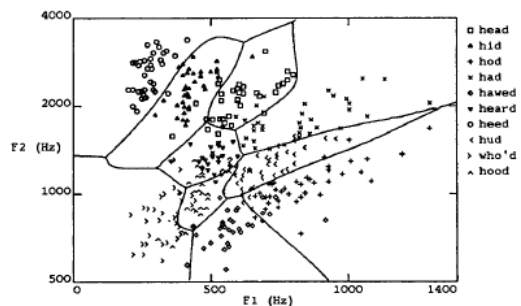
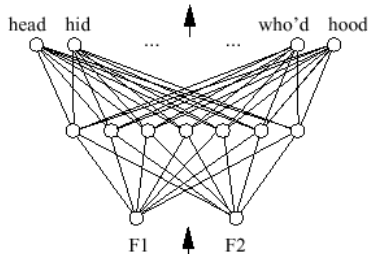
2002

CTC15 / CT215

61

Exemplo: MLP Sigmóide

- Reconhecimento de fala - separar 10 vogais entre h_d (em inglês)



2002

CTC15 / CT215

62