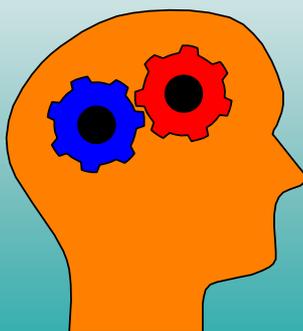


# Paradigma Conexionista

## Introdução às Redes Neurais



Maria C. Monard  
Solange O. Rezende  
Roseli F. Romero  
Thiago A. S. Pardo

1/68

## Introdução

- **Modelos inspirados no cérebro humano**
  - ◆ Compostas por várias **unidades de processamento** (“neurônios”)
  - ◆ Interligadas por um grande número de **conexões** (“sinapses”)
- Eficientes onde métodos tradicionais têm se mostrado inadequados
  - ◆ Reconhecimento de caracteres escritos à mão, fala, rostos, etc.

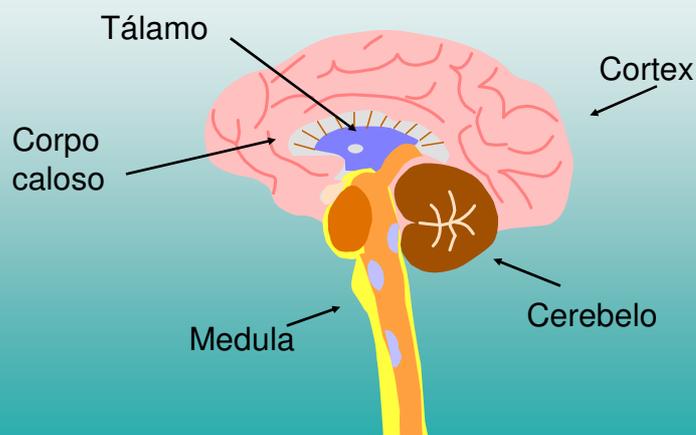
2/68

## Sistema Nervoso

- Conjunto de células extremamente complexo que tem papel essencial na determinação do funcionamento e comportamento dos seres vivos
- Divide-se em
  - ◆ Sistema nervoso central (**SNC**)
  - ◆ Sistema nervoso periférico (**SNP**)

3/68

## Sistema Nervoso Central



4/68

## Cérebro Humano vs. Computadores

- Funciona de forma **inteiramente diferente** dos computadores convencionais
  - ◆ Neurônios são **100 mil a 1 milhão de vezes mais lentos** que portas lógicas de silício
  - ◆ Lentidão compensada por grande número de **neurônios massivamente conectados**
  - ◆ Para certas operações, muito mais rápido que computadores convencionais
    - Visão, audição, controle, previsão

5/68

## Características do Cérebro

- Paralelismo massivo
- Representação e computação distribuída
- Capacidade de aprender
- Capacidade de generalizar
- Adaptatividade
- Processamento de informação contextual
- Tolerância a falha
- Baixo consumo de energia



6/68

## Redes Neurais Artificiais

- **Redes Neurais Artificiais** (RNAs) são tentativas de produzir sistemas de aprendizado biologicamente realistas
  - ◆ São baseadas em modelos abstratos de como pensamos que o cérebro (e os neurônios) funcionam
  - ◆ RNAs aprendem por exemplo
    - **RNA = arquitetura + aprendizado**

7/68

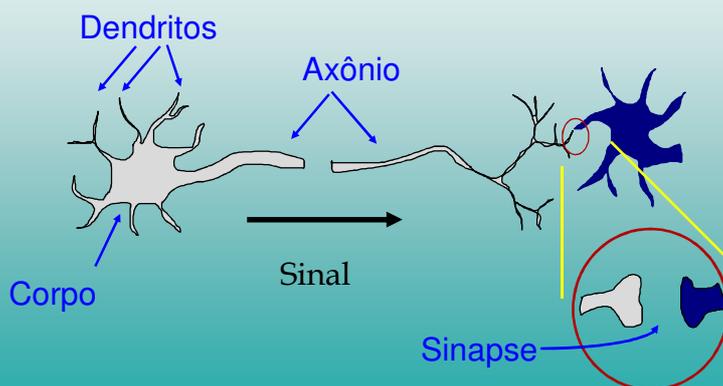
## Estudos de RNA

- **Dois grupos famosos** de trabalho em RNAs
  - ◆ Interesses em modelagem da **capacidade cognitiva humana**
  - ◆ Interesses em **aprendizado de máquina**
    - Predominante atualmente
    - Não totalmente fiel à biologia

8/68

## Neurônio Natural

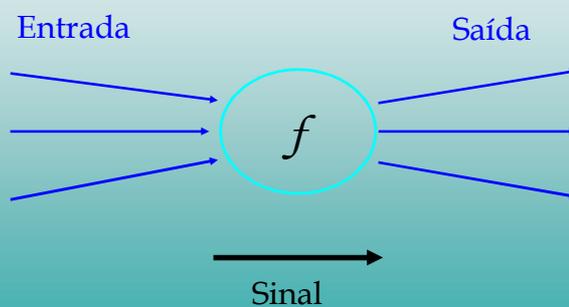
- Um neurônio simplificado



9/68

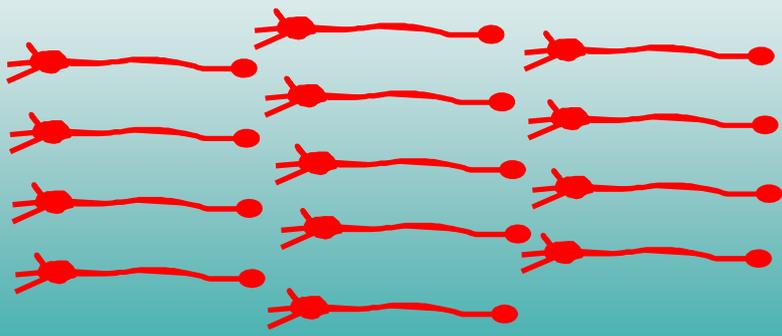
## Neurônio Artificial

- Modelo de um neurônio (nó) abstrato

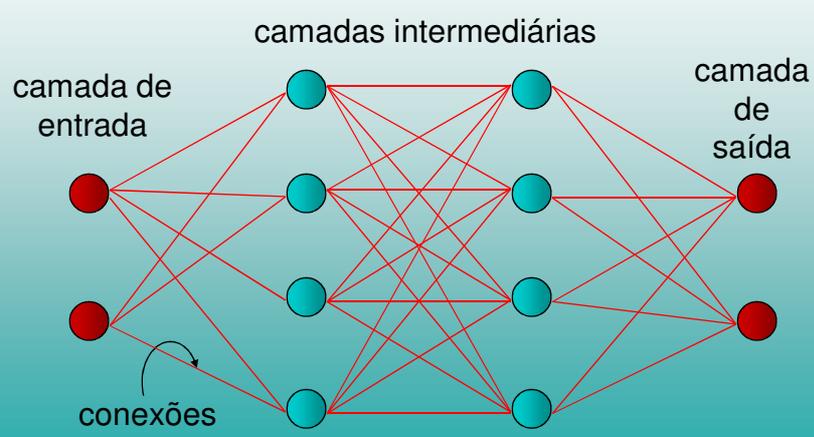


10/68

# Redes Neurais Naturais



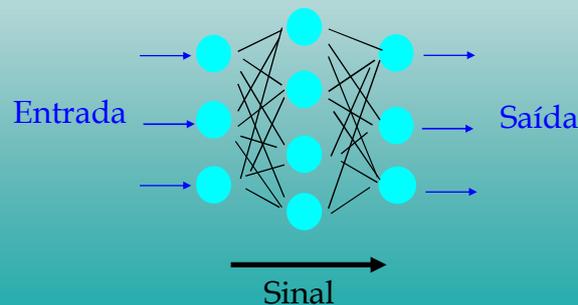
# Redes Neurais Artificiais



## Redes Neurais Artificiais

### Estrutura das RNAs

- Nós equivalem aos neurônios
- Conexões são semelhantes às sinapses



13/68

## Redes Neurais Artificiais

- Em geral, as **redes formam grafos** de várias formas
  - ◆ Direcionados ou não
  - ◆ Acíclicos ou não
- RNAs mais comuns: **direcionadas e acíclicas**
- **Aprendizado** consiste em aprender **pesos para as arestas**

14/68

## Exemplo

- Sistema ALVINN (1993)
  - ◆ Direção de um veículo em estradas

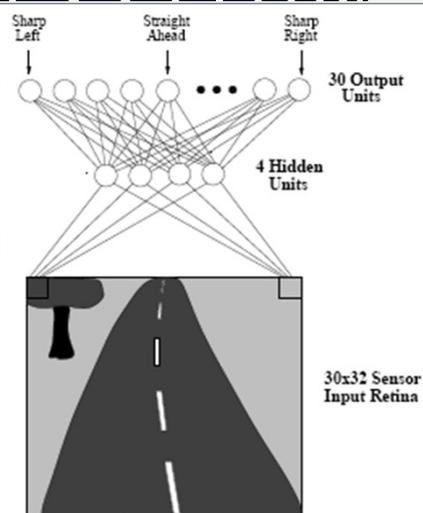


Hoje há muitos assim!

15/68

## Exemplo

- Grade de pixels (30 x 32) como entrada (960 entradas)
- Camada escondida: não se tem acesso à saída dos neurônios
- Treinamento a partir de 5 minutos de direção humana
- Direção de 112km/h, por 144km !!!



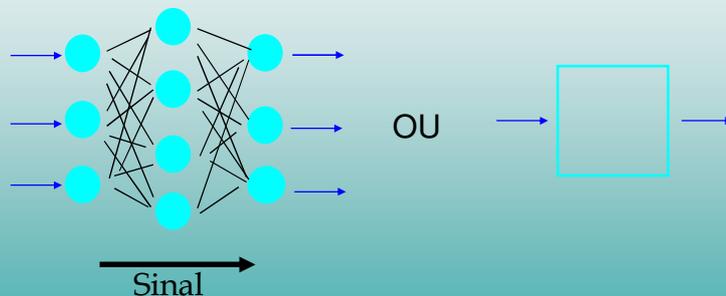
## Aplicações e características

- Problemas que envolvem dados complexos de sensores, com possíveis ruídos, como microfones e câmeras
- Instâncias no formato atributo-valor
- Função aprendida pode produzir valores discretos, contínuos ou vetores
- Podem haver erros nos dados
- O tempo de treinamento pode ser longo
- Hipótese aprendida não é legível por humanos

17/68

## Redes *Feedforward*

- Sinais seguem em uma única direção



- Tipo mais comum

18/68

## Redes Recorrentes

- Possuem conexões ligando saída da rede a sua entrada

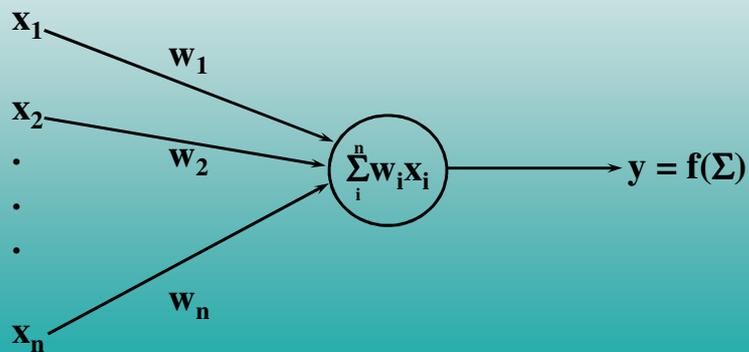


- Podem lembrar entradas passadas e, conseqüentemente, processar sequência de informações (no tempo ou espaço)

19/68

## Um Neurônio Artificial

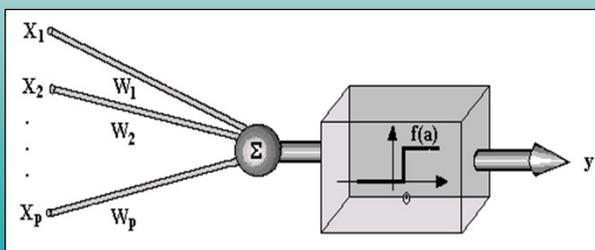
- Modelo simples e muito utilizado



20/68

## Nomenclatura

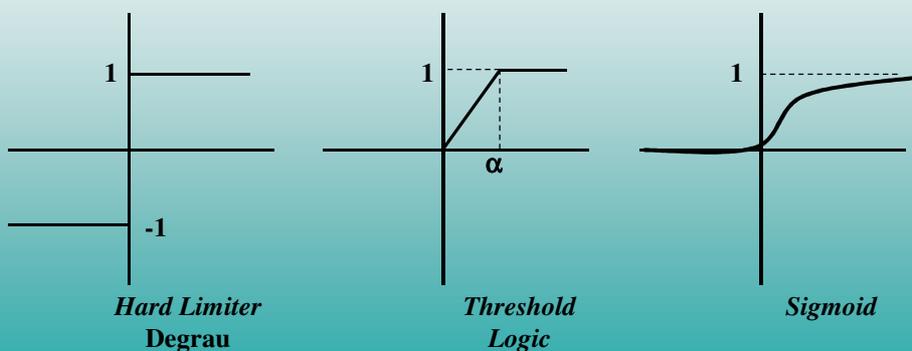
- Neurônios (nós): unidades de processamento
- Função de ativação  $f$
- Saída  $y$
- Conexões  $w_i$
- Aprendizado: estimativa das conexões  $w_i$
- Topologia da rede: uma única camada



21/68

## Funções de ativação

- Possíveis funções de ativação

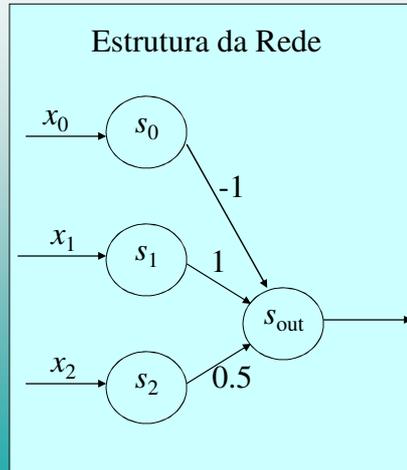


22/68

## Exemplo

Assumindo a **função degrau**, calcule as saídas para cada uma das entradas abaixo

	$x_0$	$x_1$	$x_2$
Entrada 1:	1	0	0
Entrada 2:	1	0	1
Entrada 3:	1	1	0
Entrada 4:	1	1	1



23/68

## Aprendizado

- Uma rede neural **deve produzir** para cada conjunto de entradas apresentado o **conjunto de saídas desejado**
- Quando a saída produzida é diferente da desejada, os **pesos da rede** são modificados
  - ◆  $w(t+1) = w(t) + \text{fator\_de\_correção}$

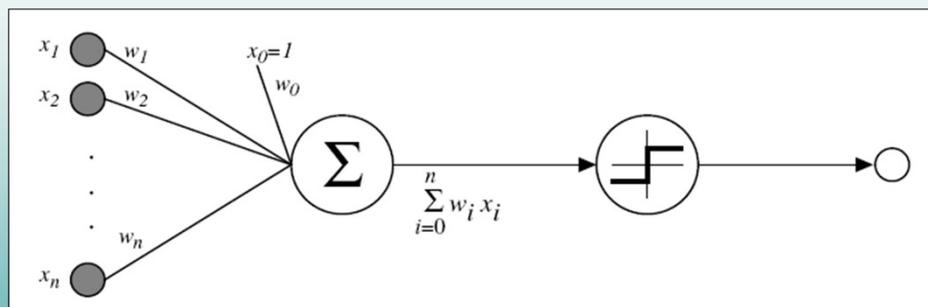
24/68

# PERCEPTRON

- Modelo de Neurônio
- Características Básicas
- Estrutura da Rede
- Algoritmo de Aprendizado

25/68

# MODELO DO NEURÔNIO



saída=1, se  $w_0+w_1x_1+\dots+w_nx_n>0$   
saída=0, caso contrário

$w_0$  é o *bias/threshold*

26/68

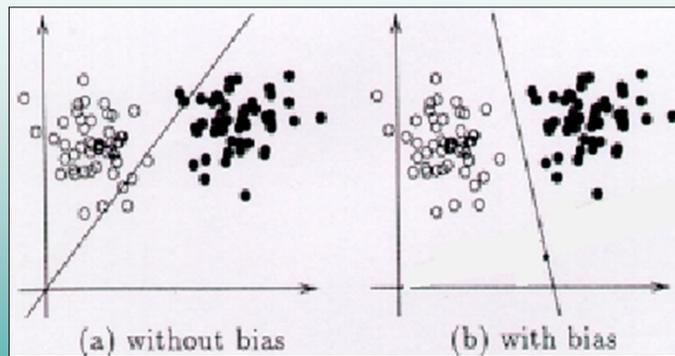
## CARACTERÍSTICAS BÁSICAS

- Regra de propagação:  $rede_j = \sum_i x_i w_{ij} + w_0$
- Função de ativação: degrau, normalmente
- Topologia: uma única camada de processadores

27/68

## PERCEPTRON

Finalidade do *threshold*



$\sum_i x_i w_{ij} = 0$  Define um hiperplano passando pela origem

$\sum_i x_i w_{ij} + w_0 = 0$  Desloca-se o hiperplano da origem

28/68

## ALGORITMO DE APRENDIZADO

- 1) Inicializam-se os pesos sinápticos com valores randômicos, pequenos ou iguais a zero
- 2) Aplica-se um padrão com seu respectivo valor desejado de saída ( $t_j$ ) e verifica-se a saída da rede ( $s_j$ )
- 3) Calcula-se o erro na saída:  $E_j = t_j - s_j$
- 4) Se  $E_j = 0$ , volta ao passo 2; se  $E_j \neq 0$ , atualiza os pesos com o fator de correção  $\Delta w_{ij}$
- 5) Volta ao passo 2

29/68

## ALGORITMO DE APRENDIZADO

- Saída correta: peso mantido
- Caso contrário, cada peso é incrementado quando a saída é menor que o valor-alvo e decrementado quando a saída é maior que o valor-alvo
- Regra de treinamento do perceptron:

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

$$\Delta w_{ij} = \eta x_i E_j$$

$\eta$  é a **taxa de aprendizado**, que regula a aproximação da rede da hipótese desejada: deve ser um valor pequeno

30/68

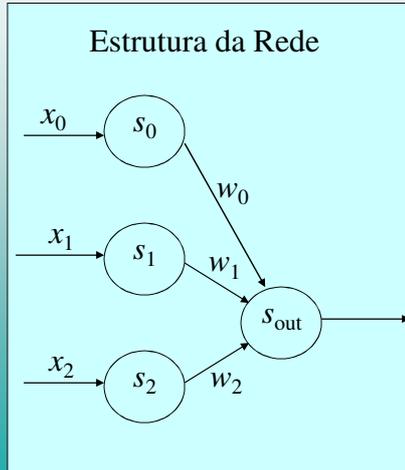
## EXEMPLO

### Simulação do Operador Lógico AND

AND	$x_0$	$x_1$	$x_2$	$t$
Entrada 1:	1	0	0	0
Entrada 2:	1	0	1	0
Entrada 3:	1	1	0	0
Entrada 4:	1	1	1	1

Peso inicial:  $w_0=0, w_1=0, w_2=0$

Taxa de aprendizado:  $\eta = 0.5$



31/68

## EXEMPLO

### 1º. Ciclo

$$\begin{aligned} \text{Entrada 1: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(0 \times 1 + 0 \times 0 + 0 \times 0) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 2: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(0 \times 1 + 0 \times 0 + 0 \times 1) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 3: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(0 \times 1 + 0 \times 1 + 0 \times 0) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 4: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(0 \times 1 + 0 \times 1 + 0 \times 1) = f(0) = 0 \implies s_{\text{out}} \neq t \end{aligned}$$

$$w_0 = w_0 + \eta(t - s_{\text{out}})x_0 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$$

$$w_1 = w_1 + \eta(t - s_{\text{out}})x_1 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$$

$$w_2 = w_2 + \eta(t - s_{\text{out}})x_2 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$$

32/68

## EXEMPLO

2º. Ciclo



???

Continuem: façam mais um ciclo!

33/68

## EXEMPLO

2º. Ciclo



$$\text{Entrada 1: } s_{\text{out}} = f(w_0x_0 + w_1x_1 + w_2x_2)$$

$$= f(0.5 \times 1 + 0.5 \times 0 + 0.5 \times 0) = f(0.5) = 1 \implies s_{\text{out}} \neq t$$

$$w_0 = w_0 + \eta(t - s_{\text{out}})x_0 = 0.5 + 0.5 \times (0 - 1) \times 1 = 0$$

$$w_1 = w_1 + \eta(t - s_{\text{out}})x_1 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$$

$$w_2 = w_2 + \eta(t - s_{\text{out}})x_2 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$$

$$\text{Entrada 2: } s_{\text{out}} = f(w_0x_0 + w_1x_1 + w_2x_2)$$

$$= f(0 \times 1 + 0.5 \times 0 + 0.5 \times 1) = f(0.5) = 1 \implies s_{\text{out}} \neq t$$

$$w_0 = w_0 + \eta(t - s_{\text{out}})x_0 = 0 + 0.5 \times (0 - 1) \times 1 = -0.5$$

$$w_1 = w_1 + \eta(t - s_{\text{out}})x_1 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$$

$$w_2 = w_2 + \eta(t - s_{\text{out}})x_2 = 0.5 + 0.5 \times (0 - 1) \times 1 = 0$$

34/68

## EXEMPLO

### 2º. Ciclo

$$\begin{aligned} \text{Entrada 3: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-0.5 \times 1 + 0.5 \times 1 + 0 \times 0) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 4: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-0.5 \times 1 + 0.5 \times 1 + 0 \times 1) = f(0) = 0 \implies s_{\text{out}} \neq t \end{aligned}$$

$$w_0 = w_0 + \eta(t - s_{\text{out}})x_0 = -0.5 + 0.5 \times (1 - 0) \times 1 = 0$$

$$w_1 = w_1 + \eta(t - s_{\text{out}})x_1 = 0.5 + 0.5 \times (1 - 0) \times 1 = 1$$

$$w_2 = w_2 + \eta(t - s_{\text{out}})x_2 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$$

35/68

## EXEMPLO

### 3º. Ciclo

$$\begin{aligned} \text{Entrada 1: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(0 \times 1 + 1 \times 0 + 0.5 \times 0) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 2: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(0 \times 1 + 1 \times 0 + 0.5 \times 1) = f(0.5) = 1 \implies s_{\text{out}} \neq t \end{aligned}$$

$$w_0 = w_0 + \eta(t - s_{\text{out}})x_0 = -0.5 + 0.5 \times (0 - 1) \times 1 = -1$$

$$w_1 = w_1 + \eta(t - s_{\text{out}})x_1 = 1 + 0.5 \times (0 - 1) \times 0 = 1$$

$$w_2 = w_2 + \eta(t - s_{\text{out}})x_2 = 0.5 + 0.5 \times (0 - 1) \times 1 = 0$$

36/68

## EXEMPLO

### 3º. Ciclo

$$\begin{aligned} \text{Entrada 3: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 1 + 0 \times 0) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 4: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 1 + 0 \times 1) = f(0) = 0 \implies s_{\text{out}} \neq t \\ w_0 &= w_0 + \eta(t - s_{\text{out}})x_0 = -1 + 0.5 \times (1 - 0) \times 1 = -0.5 \\ w_1 &= w_1 + \eta(t - s_{\text{out}})x_1 = 1 + 0.5 \times (1 - 0) \times 1 = 1.5 \\ w_2 &= w_2 + \eta(t - s_{\text{out}})x_2 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5 \end{aligned}$$

37/68

## EXEMPLO

### 4º. Ciclo

$$\begin{aligned} \text{Entrada 1: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-0.5 \times 1 + 1.5 \times 0 + 0.5 \times 0) = f(-0.5) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 2: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-0.5 \times 1 + 1.5 \times 0 + 0.5 \times 1) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 3: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-0.5 \times 1 + 1.5 \times 1 + 0.5 \times 0) = f(1) = 1 \implies s_{\text{out}} \neq t \\ w_0 &= w_0 + \eta(t - s_{\text{out}})x_0 = -0.5 + 0.5 \times (0 - 1) \times 1 = -1 \\ w_1 &= w_1 + \eta(t - s_{\text{out}})x_1 = 1.5 + 0.5 \times (0 - 1) \times 1 = 1 \\ w_2 &= w_2 + \eta(t - s_{\text{out}})x_2 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5 \end{aligned}$$

$$\begin{aligned} \text{Entrada 4: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 1 + 0.5 \times 1) = f(0.5) = 1 \implies s_{\text{out}} = t \end{aligned}$$

38/68

## EXEMPLO

### 5º. Ciclo

$$\begin{aligned} \text{Entrada 1: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 0 + 0.5 \times 0) = f(-1) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 2: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 0 + 0.5 \times 1) = f(-0.5) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 3: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 1 + 0.5 \times 0) = f(0) = 0 \implies s_{\text{out}} = t \end{aligned}$$

$$\begin{aligned} \text{Entrada 4: } s_{\text{out}} &= f(w_0x_0 + w_1x_1 + w_2x_2) \\ &= f(-1 \times 1 + 1 \times 1 + 0.5 \times 1) = f(0.5) = 1 \implies s_{\text{out}} = t \end{aligned}$$

$$w_0 = -1, w_1 = 1, w_2 = 0.5$$

39/68

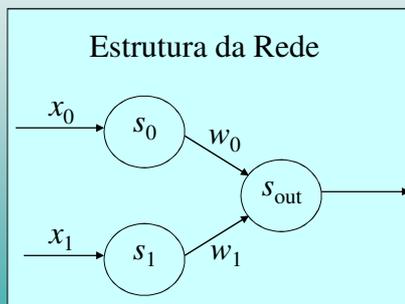
## EXERCÍCIO

### Simulação do Operador Lógico OR

OR	$x_0$	$x_1$	$t$
Entrada 1:	0	0	0
Entrada 2:	0	1	1
Entrada 3:	1	0	1
Entrada 4:	1	1	1

Peso inicial:  $w_0 = 0, w_1 = 0$

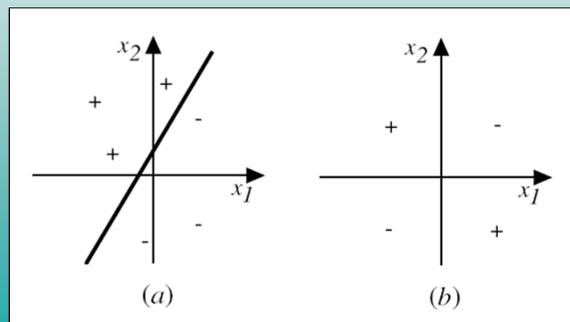
Taxa de aprendizado:  $\eta = 0.5$



40/68

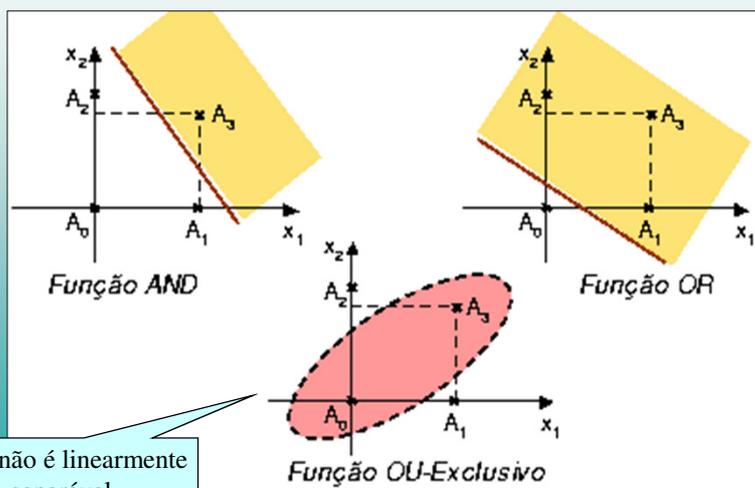
## Limitações

- A rede de uma única camada funciona para **exemplos linearmente separados**



41/68

## O PROBLEMA DO OU-EXCLUSIVO (XOR)



XOR não é linearmente separável

42/68

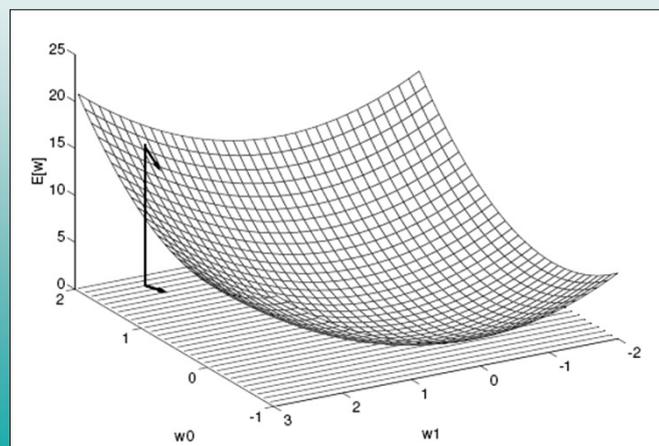
## Treinamento do perceptron

- A regra de treinamento do perceptron **não converge** quando o **problema não é linearmente separável**
- Solução: **regra delta**
  - ◆ Usa o *gradient descent*
    - A cada iteração, diminui o erro
  - ◆ Encontra a melhor aproximação para o problema

43/68

## Regra delta

- Em busca do erro mínimo (mínimo global)



44/68

## Regra delta

- Também conhecida por
  - ◆ Regra LMS (*least-mean-square*)
  - ◆ Regra Adaline
  - ◆ Regra de Widrow-Hoff
- Não pode haver *threshold*: interfere no processo de aprendizado
  - ◆ Agora, em vez de perceptron, chama-se o neurônio de “**unidade linear**”
- Todos os erros de todas as entradas são somados e, somente depois, os pesos são atualizados

$$\Delta w_i = \eta \sum_j (t_j - s_j) x_{ij}$$

45/68

## Regra delta

- **Dificuldades** dessa abordagem
  - ◆ A convergência da rede pode demorar muito
  - ◆ Computacionalmente caro
  - ◆ Se houver vários mínimos locais possíveis na superfície de erro, então não há garantia de que o mínimo global será encontrado
- **Solução**
  - ◆ **Aproximação estocástica**: atualização imediata dos pesos para cada entrada vista
    - Não se soma tudo antes
    - Diferencia-se da regra do perceptron pelo fato de não haver *threshold*

46/68

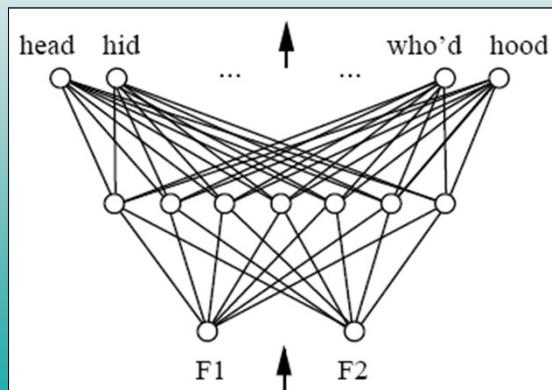
## Redes multicamadas

- Na prática, não se usam redes de uma só camada, mas de **várias camadas**
- Redes capazes de fazer **separações não lineares**
- Possui uma ou mais camadas intermediárias/escondidas de nós
- Geralmente utiliza função de ativação sigmóide
- Em geral, treinamento pelo algoritmo *backpropagation*

47/68

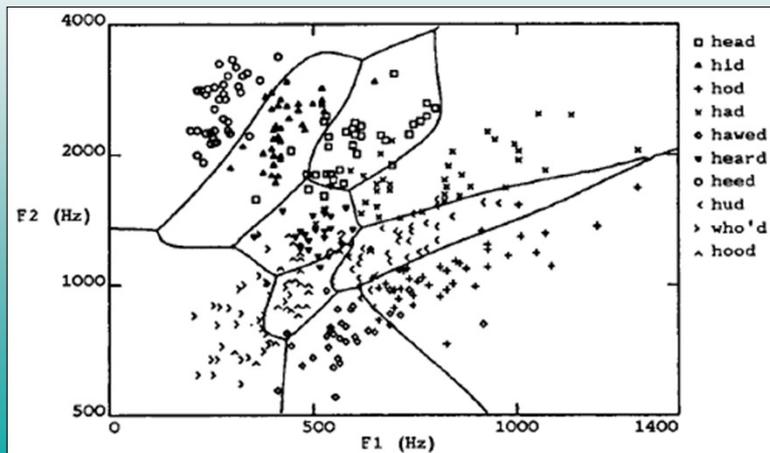
## Exemplo: identificação de sons

- Entrada: características espectrais do som de uma vogal
- Saída: possível som da vogal



## Exemplo: identificação de sons

- Separação não linear

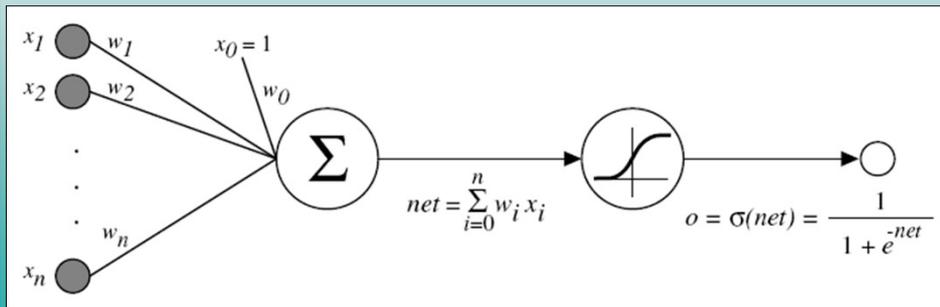


## Unidade de processamento

- Se for um perceptron, tem-se a rede chamada *multilayer perceptron*
  - ◆ Mas o *threshold* e a função “degrau” tornam o perceptron inadequado para o método de aprendizado mais comumente utilizado
- Também se pode ter uma rede multicamada de unidades lineares
  - ◆ Mas ainda apenas capaz de fazer separações lineares

## Unidade de processamento

- Costuma-se utilizar a “unidade sigmóide”, cuja função de ativação é a sigmóide



51/68

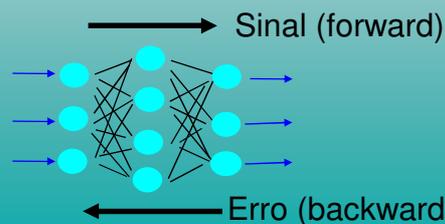
## Técnica de Treinamento

### *Backpropagation*

- Rede é treinada com pares entrada-saída
- Cada entrada de treinamento está associada a uma saída desejada
- Treinamento em duas fases, cada uma percorrendo a rede em um sentido

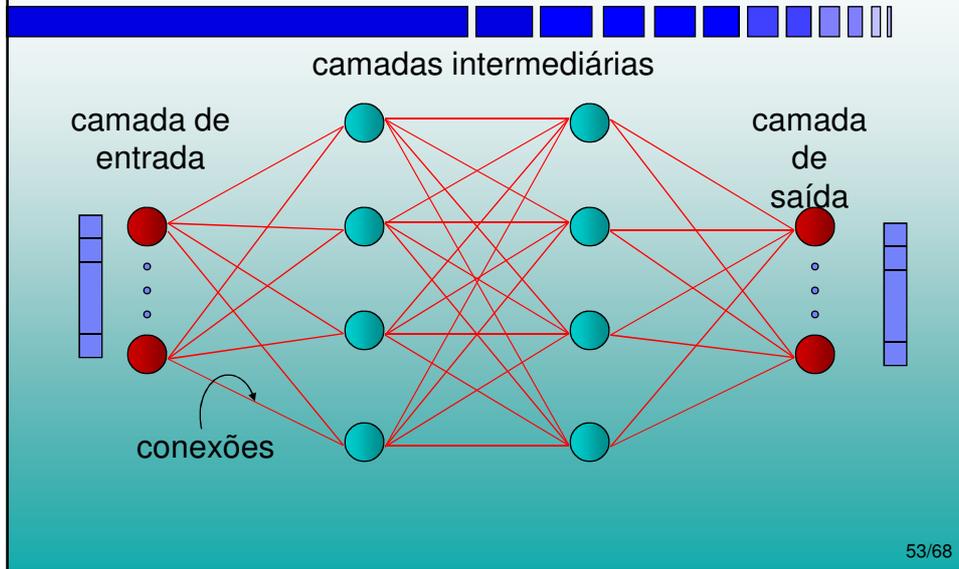
- ◆ Fase **forward**

- ◆ Fase **backward**

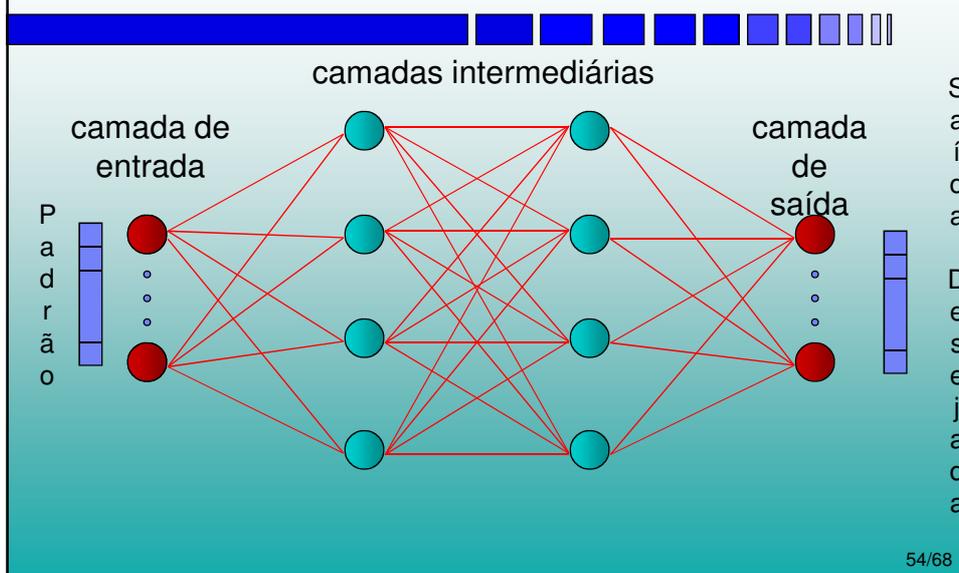


52/68

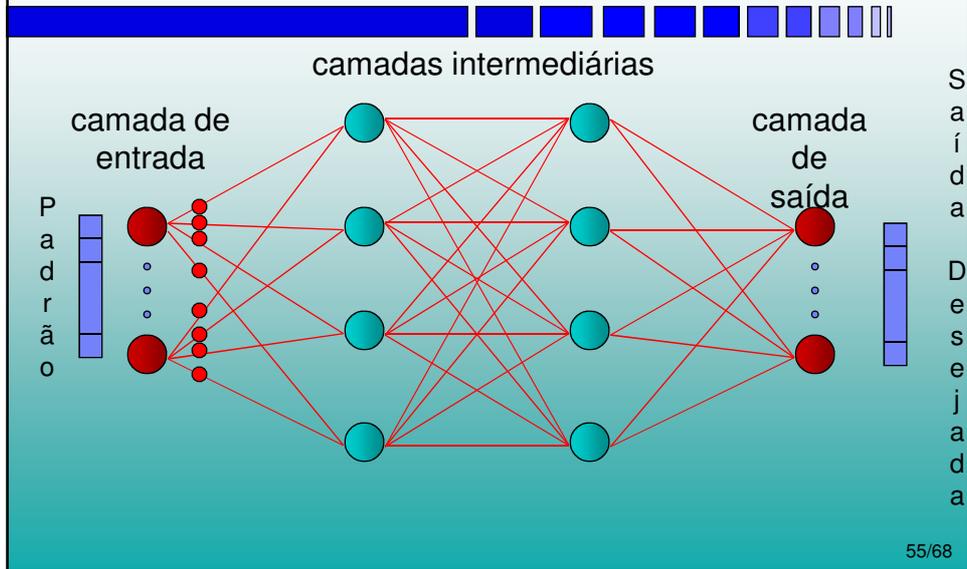
# Redes Neurais Artificiais



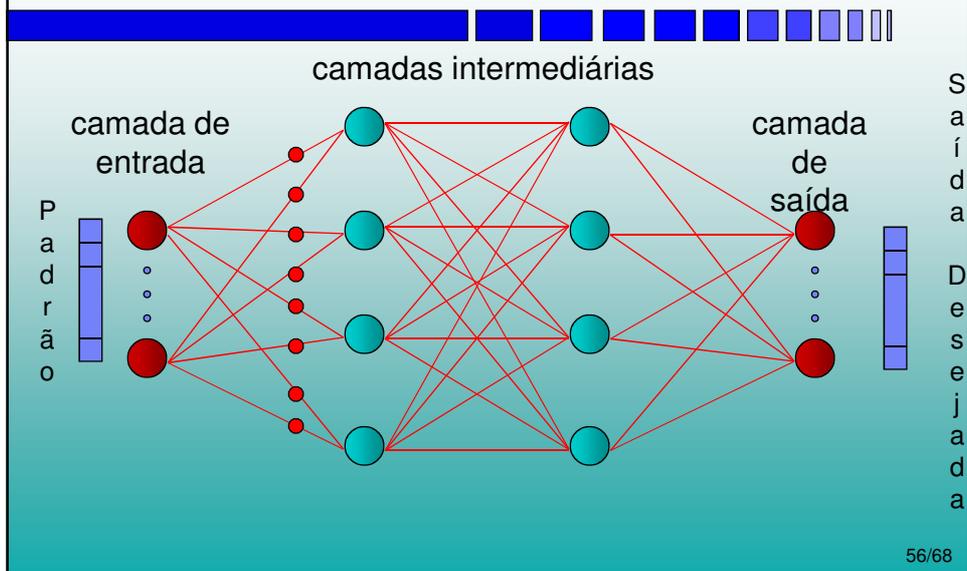
# RNA - Aprendizado



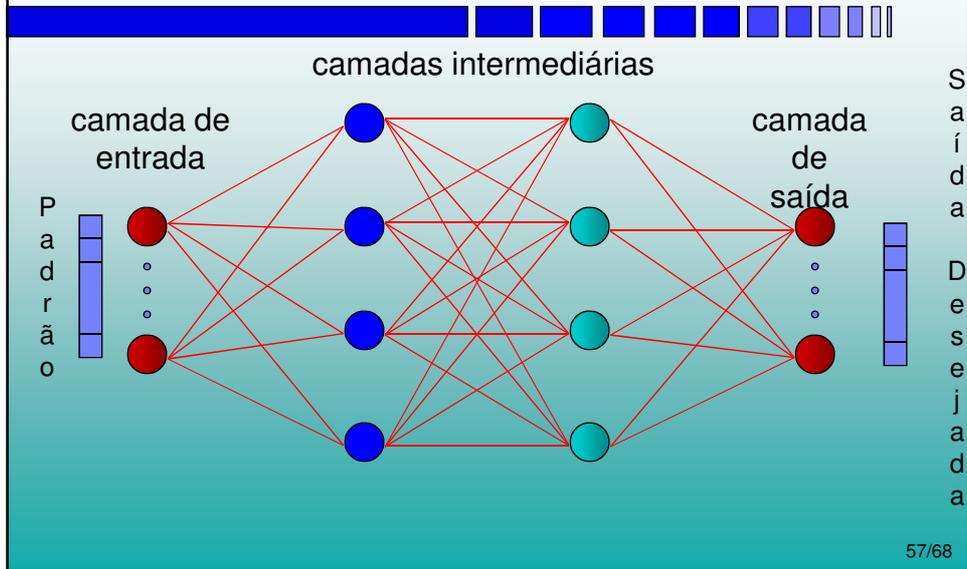
# RNA - Aprendizado



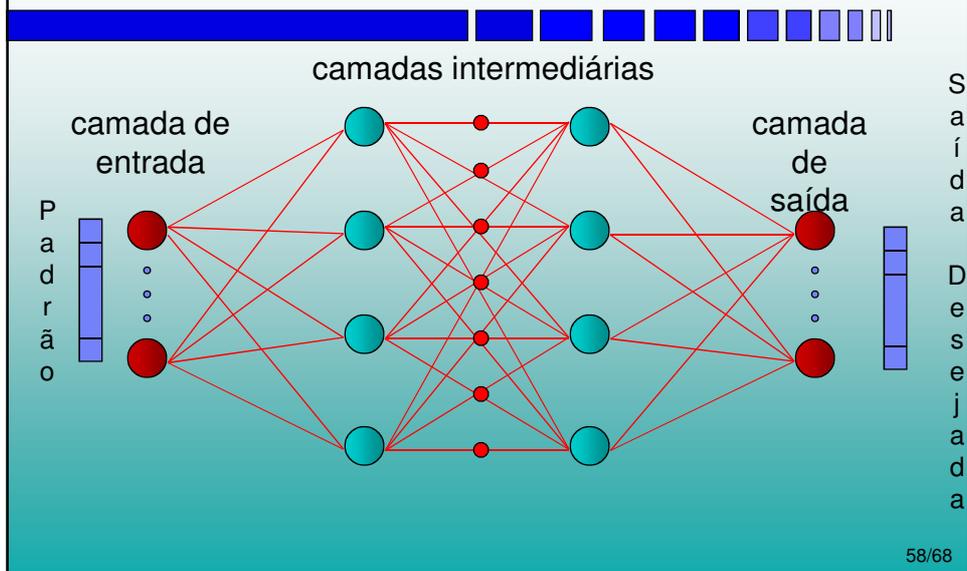
# RNA - Aprendizado



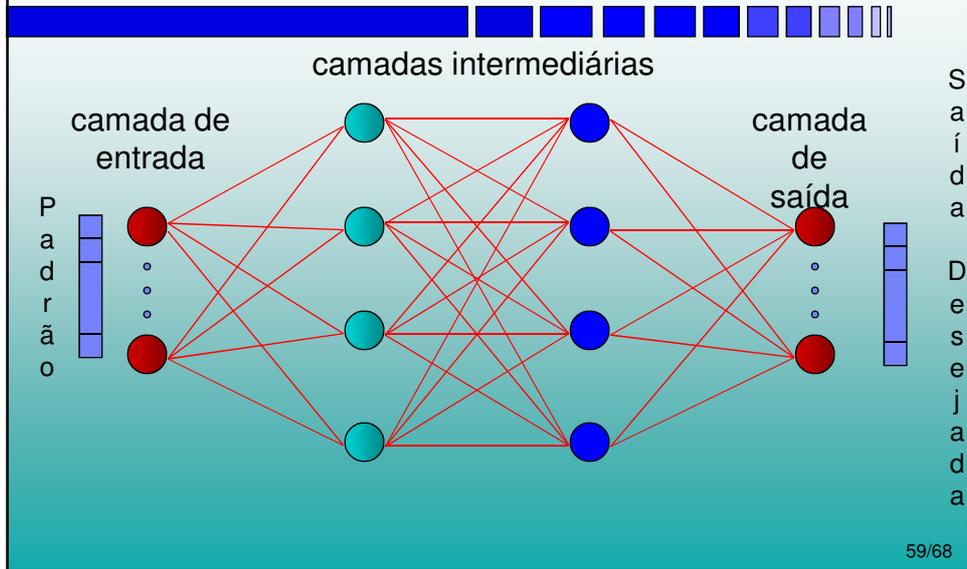
# RNA - Aprendizado



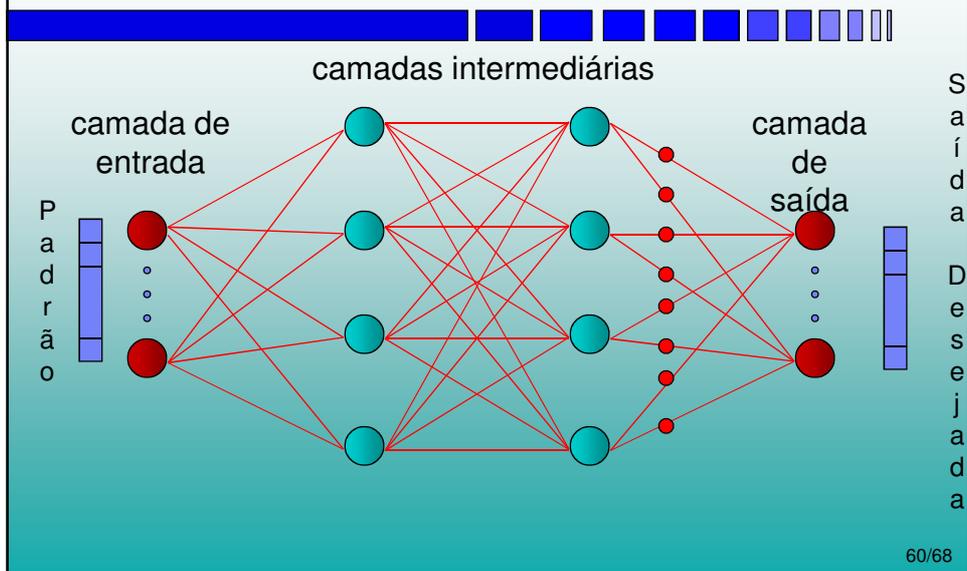
# RNA - Aprendizado



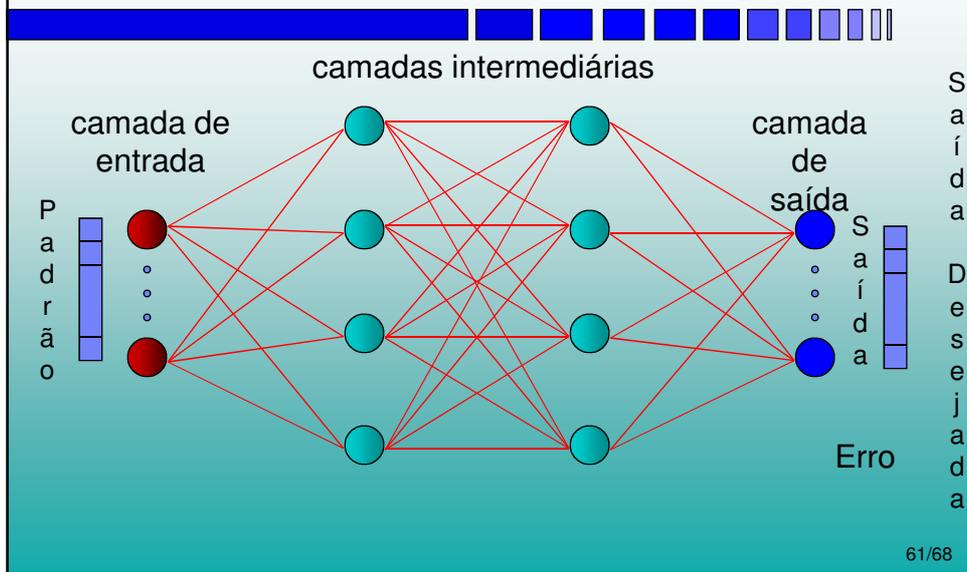
# RNA - Aprendizado



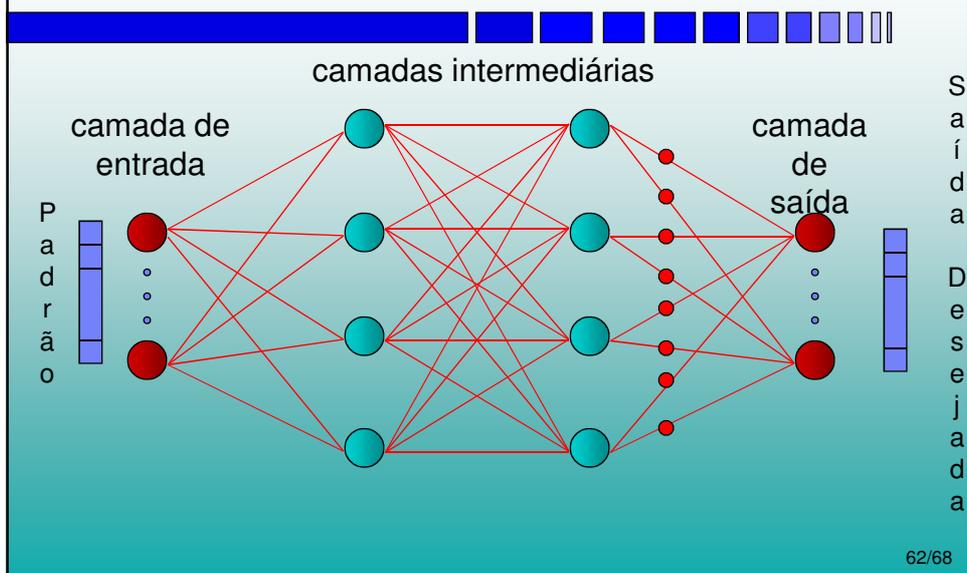
# RNA - Aprendizado



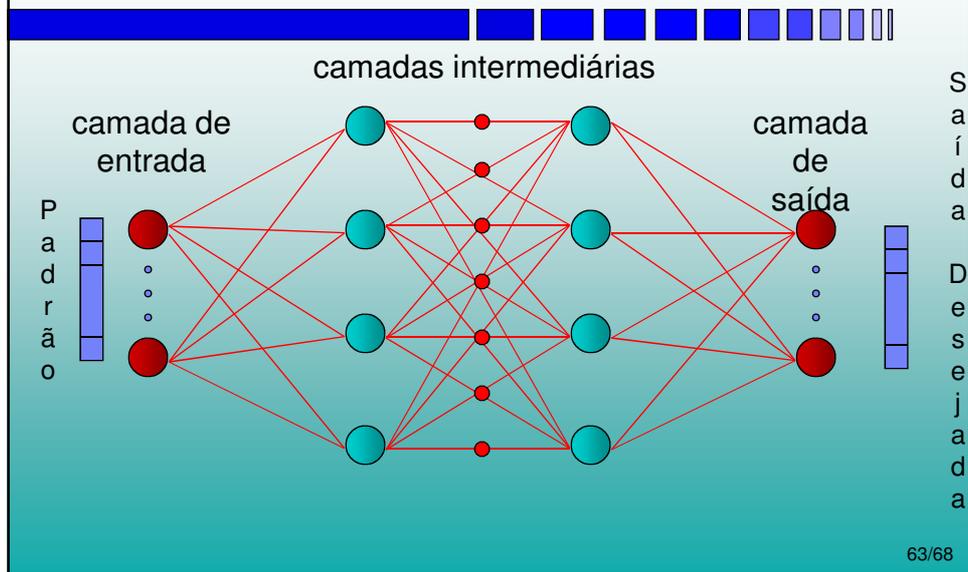
# RNA - Aprendizado



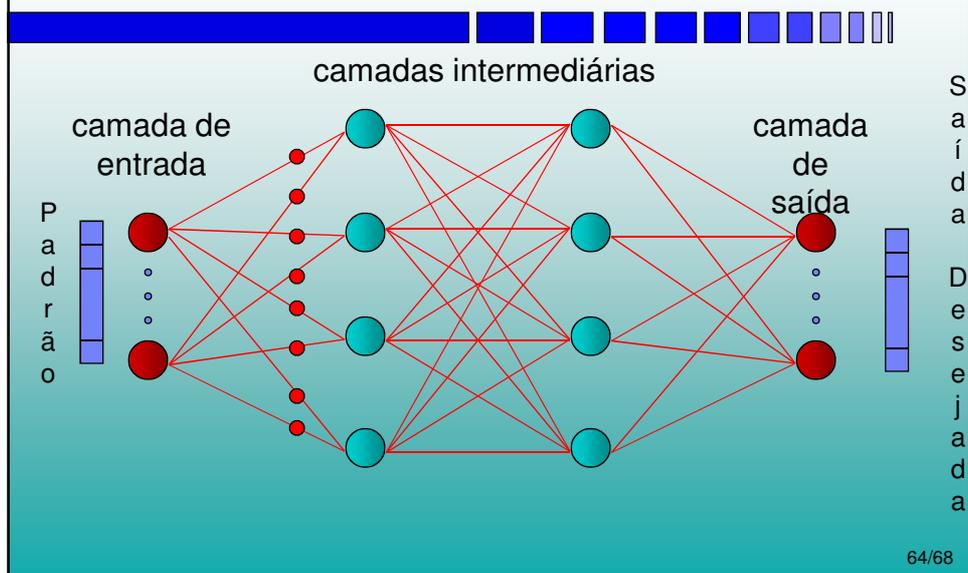
# RNA - Aprendizado



# RNA - Aprendizado



# RNA - Aprendizado



## Fase *forward*

- Entrada é apresentada à primeira camada da rede
  - ◆ Após os neurônios da camada  $i$  calcularem seus sinais de saída, os neurônios da camada  $i + 1$  calculam seus sinais de saída
  - ◆ Saídas produzidas pelos neurônios da última camada são comparadas com as saídas desejadas
  - ◆ Erro para cada neurônio de saída é calculado

65/68

## Fase *backward*

- A partir da última camada, o erro de cada nó é dividido entre suas conexões
  - ◆ O nó ajusta seu peso de modo a reduzir o erro que ele produzirá no futuro
  - ◆ Nós das camadas anteriores têm seus erros definidos pelos erros dos nós da camada seguinte conectados a eles ponderados pelos pesos das conexões entre eles

66/68

## Backpropagation

- O algoritmo é repetido até que se alcance um **limiar aceitável para o erro**
  - ◆ Dependente da tarefa e experiência prévia

67/68

## Problemas das RNAs

- Definição dos **parâmetros**
  - ◆ “Magia negra”
- Extração de **conhecimento**
  - ◆ Caixa preta
- Existem **vários tipos** de RNAs diferentes
  - ◆ Cada tipo tem propósitos diferentes
  - ◆ Alguns tipos são mais adequados para resolver classes particulares de problemas



68/68