



Redes Neurais Artificiais

UFPI

PROF. ME. FILIPE FONTINELE DE ALMEIDA

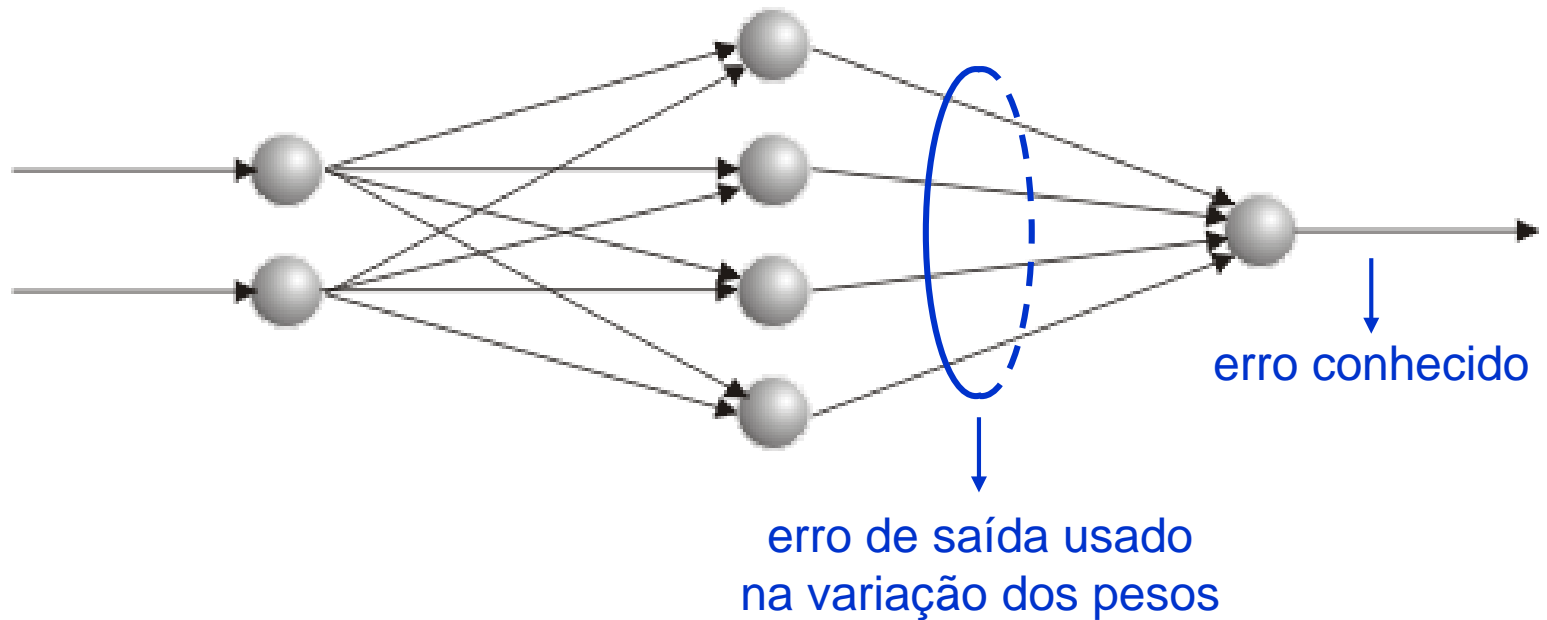
FILIFEFONTINELI@GMAIL.COM

Treinamento BP

- ▶ Objetivo:
 - ▶ providenciar valores corretos para os pesos visando uma correta saída da rede neural
- ▶ Requisito:
 - ▶ dados de entrada-saída conhecidos

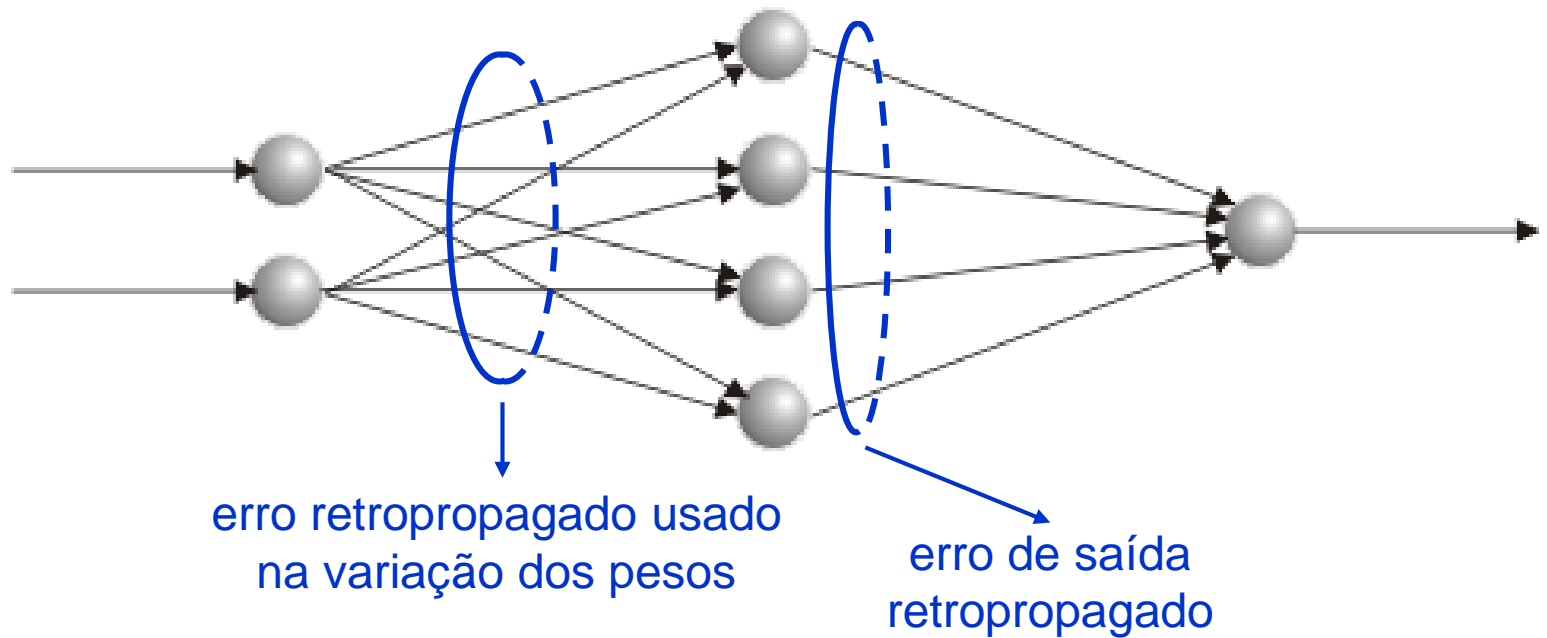
Treinamento BP

- Retro propagação do erro:



Treinamento BP

- Retro propagação do erro:



Treinamento BP

- ▶ Cálculo dos novos pesos:

- ▶ entre camadas de saída e escondida:

$$\Delta W_{ij} = \eta \cdot e_i \cdot \dot{f}_i \cdot out_j$$

i – neurônio de saída j – neurônio escondido

- ▶ entre camadas escondida e de entrada

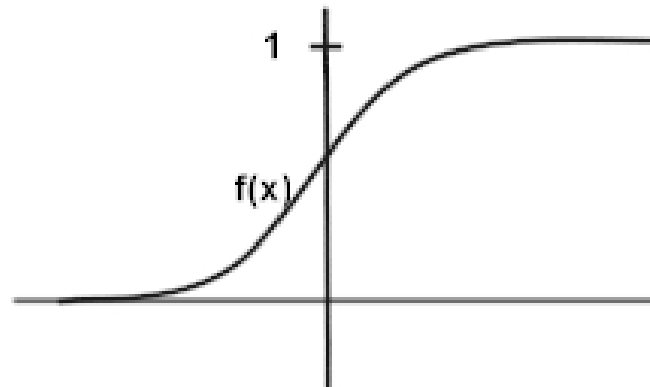
$$\Delta W_{jk} = -\eta \cdot e_j \cdot \dot{f}_j \cdot u_k$$

j – neurônio escondido k – neurônio de entrada

$$e_j = \sum_{i=1}^n e_i \cdot \dot{f}_i \cdot W_{ij}$$

Treinamento BP

- ▶ Função de ativação:
 - ▶ sigmoide



$$\frac{d\text{sig}(x)}{dx} = \text{sig}(x) \cdot [1 - \text{sig}(x)]$$

$$\frac{d\tanh(x)}{dx} = [1 + \tanh(x)] \cdot [1 - \tanh(x)]$$

Treinamento BP

► Algoritmo:

1. Inicializar pesos

```
Whi = rand(H,I) - 0.5;  bias_hi = rand(H,1) - 0.5;  
Woh = rand(O,H) - 0.5; bias_oh = rand(O,1) - 0.5;
```

2. Calcular entrada da camada escondida

```
net_h = Whi * X + bias_hi * ones(1, size(X,2));
```

3. Calcular a saída da camada escondida

```
Yh = logsig(net_h);
```


Treinamento BP

- ▶ Algoritmo:

- ▶ Calcular entrada da camada de saída

$$\text{net_o} = \text{Woh} * \text{Yh} + \text{bias_oh} * \text{ones}(1, \text{size}(\text{Yh}, 2));$$

- ▶ Calcular a saída da rede neural

$$\text{Y} = k * \text{net_o};$$

- ▶ Calcular erro de saída

$$\text{E} = \text{D} - \text{Y};$$

Treinamento BP

► Algoritmo:

7. Calcular variação dos pesos entre as camadas de saída e escondida

```
df = k * ones(size(net_o));  
delta_bias_oh = eta * sum((E.* df)');  
delta_Woh = eta * (E.* df) * Yh';
```

8. Calcular erro retro propagado

$$E_h = W_{oh}' * (E.* df);$$

Treinamento BP

► Algoritmo:

9. Calcular variação dos pesos entre as camadas escondida e de entrada
$$df = \text{logsig}(\text{net_h}) - (\text{logsig}(\text{net_h}))^2;$$
$$\text{delta_bias_hi} = \eta * \text{sum}((E_h * df)')$$
$$\text{delta_Whi} = \eta * (E_h * df) * X'$$
10. Calcular novos valores dos pesos
11. Calcular erro quadrático médio E_{av}
12. Se $E_{av} > E_m$, voltar para passo 2

Treinamento BP

- ▶ Observações:
 - ▶ Definido pelo projetista:
 - ▶ Taxa de aprendizado
 - ▶ valores altos aceleram aprendizado
 - ▶ valores baixos evitam oscilações
 - ▶ Número de neurônios
 - ▶ Número de camadas
 - ▶ Além de erro médio baixo, a rede deve generalizar: boa saída para dados não usados no treinamento

Treinamento BP

► Observações:

- Pesos podem ser inicializados com outros valores, por exemplo: 0
- Para taxa de aprendizado alta, pode-se usar o termo de momento:

$$\Delta W_{ij}(t+1) = \eta \cdot e_i \cdot \dot{f}_i \cdot out_j + \alpha \cdot \Delta W_{ij}(t)$$

$$\Delta W_{jk}(t+1) = -\eta \cdot e_j \cdot \dot{f}_j \cdot u_k + \alpha \cdot \Delta W_{jk}(t)$$

$\alpha \rightarrow \text{cte} > 0$ definida pelo projetista

$t \rightarrow$ número da iteração