Multi-Layer Perceptron (MLP)- Parte I

Profa. Dra. Roseli Aparecida Francelin Romero
SCC - ICMC - USP

2022

Sumário

Perceptron multicamadas

- Redes de apenas uma camada representam somente funções linearmente separáveis.
- Redes de múltiplas camadas solucionam essa restrição.
- O desenvolvimento do algoritmo *backpropagation* foi um dos motivos para o ressurgimento da área de redes neurais [Rumelhart *et. al*, 1986].

Sumário

Modelo de rede neural com múltiplas camadas.

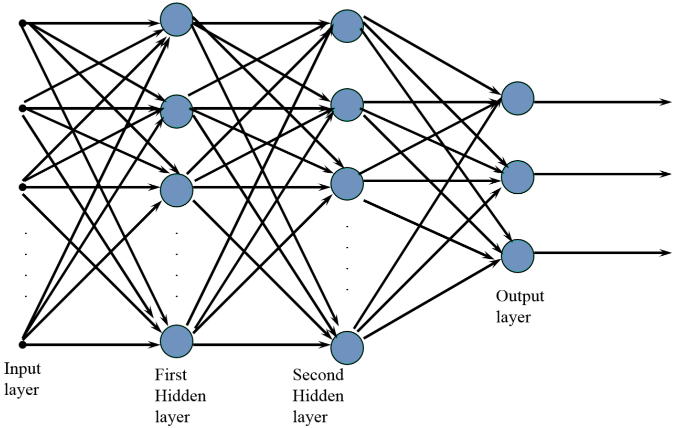

Figura 1: Rede neural *feed-forward* com múltiplas camadas.

Sumário

Sumário

Aprendizado da rede

- O esquema de aprendizado da rede pode ser descrito do seguinte modo:

Vetor entrada → vetor saída (-) / vetor desejado (+) $\overset{?}{=}$ → aprendizado ocorreu

Aprendizado da rede

- Caso contrário, os pesos são modificados para minimizar o erro:

$$E(w) = \sum_{p=1}^{N} E_p(w)$$

onde N é o no. total de padrões e E_p é o erro quadrático referente a cada par p apresentado à rede, sendo dado por:

$$E_p = \frac{1}{2} \sum_{j} (t_{pj} - y_{pj})^2$$

onde:

  - t_{pj}: j-ésima componente do vetor saída desejada.
  - y_{pj}: j-ésima componente do vetor obtido pela rede.

Aprendizado da rede

Pesos (*Gradient Descent Method*)

$$w_{ji}(k+1) = w_{ji}(k) - \eta \frac{\partial E_p(w)}{\partial w_{ji}}\Big|_{w(k)}$$

Onde η é uma constante positiva (velocidade de aprendizado).

- Calculando a derivada parcial do E_p, tem-se:

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial y_{pj}} \cdot \frac{\partial y_{pj}}{\partial v_{pj}} \cdot \frac{\partial v_{pj}}{\partial w_{ji}}$$

- Para se calcular $\frac{\partial E_p}{\partial y_{pj}}$, dois casos devem ser considerados:

Aprendizado da rede

1. Neurônio j está na camada de saída.

$$\frac{\partial E_p}{\partial y_{pj}} = -(t_{pj} - y_{pj})$$

$$\therefore \frac{\partial E_p}{\partial w_{ji}} = \underbrace{-(t_{pj} - y_{pj}) \cdot \overbrace{y_{pj}(1 - y_{pj})}^{\frac{\partial y_{pj}}{\partial v_{pj}}}}_{\delta_{pj}} \cdot y_{pi}$$

$$\boxed{\frac{\partial E_p}{\partial w_{ji}} = -\delta_{pj} \cdot y_{pi}} \rightarrow \text{erro na camada de saída}$$

onde $-\delta_{pj} = \frac{\partial E_p}{\partial v_{pj}}$

Aprendizado da rede

2. Neurônio j está na camada oculta (escondida).
   - Nesse caso, não se conhece a expressão do erro.
   - Para obtermos $\frac{\partial E_p}{\partial y_{pj}}$, usamos mais uma vez a **regra da cadeia**.

$$\frac{\partial E_p}{\partial y_{pj}} = \sum_k \frac{\partial E_p}{\partial v_{pk}} \cdot \frac{\partial v_{pk}}{\partial y_{pj}} = \sum_k \frac{\partial E_p}{\partial v_{pk}} \cdot \frac{\partial \left(\sum_j w_{kj} y_{pj} \right)}{\partial y_{pj}}$$

$$= \sum_k \frac{\partial E_p}{\partial v_{pk}} \cdot w_{kj} = \sum_k (-\delta_{pk} \cdot w_{kj})$$

$$\therefore \boxed{\frac{\partial E_p}{\partial w_{ji}} = \left(\sum_k (-\delta_{pk} w_{kj}) \right) \cdot y_{pj}(1 - y_{pj}) \cdot y_{pi}}$$

erro na camada oculta

Aprendizado da rede

- **Observação:** os erros são computados no sentido *backward*. O erro foi chamado de *back-propagado* $\rightarrow$ algoritmo de aprendizado **backpropagation** (BP).

Algoritmo *Backpropagation*

- **Inicialização:** pesos iniciados com valores aleatórios e pequenos ($[-1, +1]$).
- **Treinamento - Repita:**
  - Considere um novo padrão de entrada x_i e seu respectivo vetor de saída t_i desejado do conjunto de treinamento.
  - **Repita:**
    - Apresentar o par (x_i, t_i). **(modo padrão)**
    - Calcular as saídas dos processadores, começando da primeira camada escondida até a camada de saída.
    - Calcular o erro na camada de saída.
    - Atualizar os pesos de cada processador, começando pela camada de saída, até a camada de entrada.
  - **Até que o erro quadrático médio para esse padrão seja** $<= tol1$.
- **Até que o erro quadrático médio seja** $<= tol2$ **para todos os padrões do conjunto de treinamento.**

Sumário

Processo de aprendizado

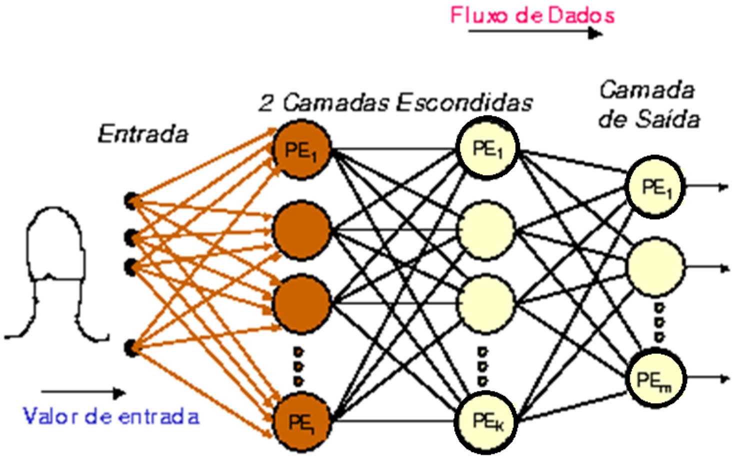

Figura 2: *Feed-forward* (fase 1), primeira camada escondida.

Processo de aprendizado

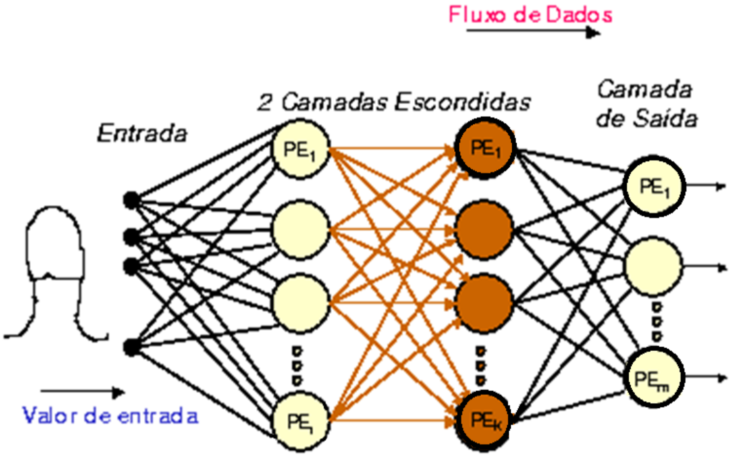

Figura 3: *Feed-forward* (fase 1), segunda camada escondida.

Processo de aprendizado

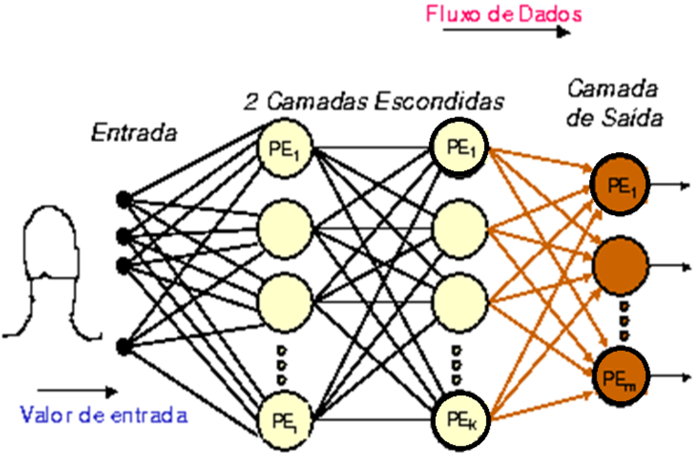

Figura 4: *Feed-forward* (fase 1), camada de saída.

Processo de aprendizado

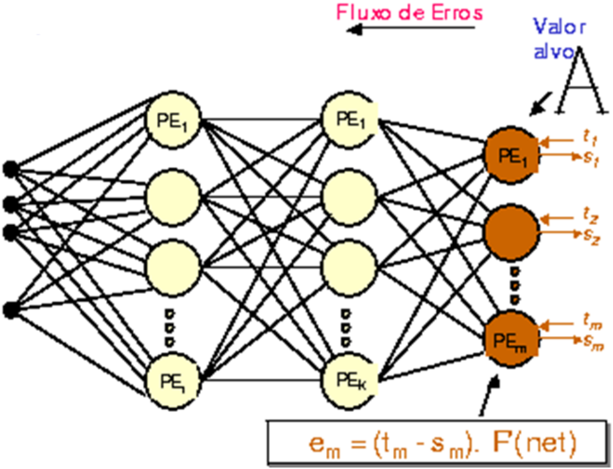

Figura 5: *Feed-backward* (fase 2), cálculo do erro da camada de saída.

Processo de aprendizado

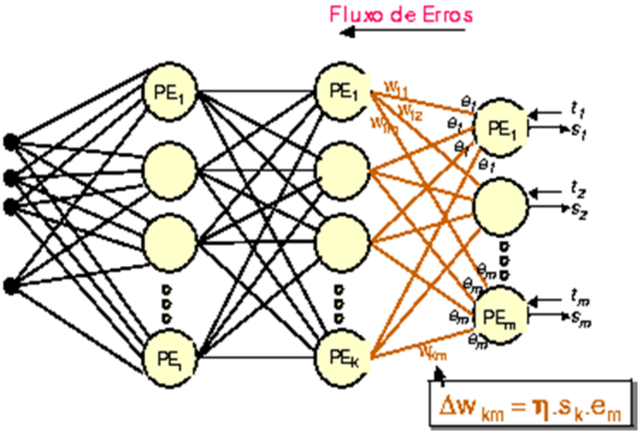

Figura 6: *Feed-backward* (fase 2), atualização dos pesos da camada de saída.

Processo de aprendizado

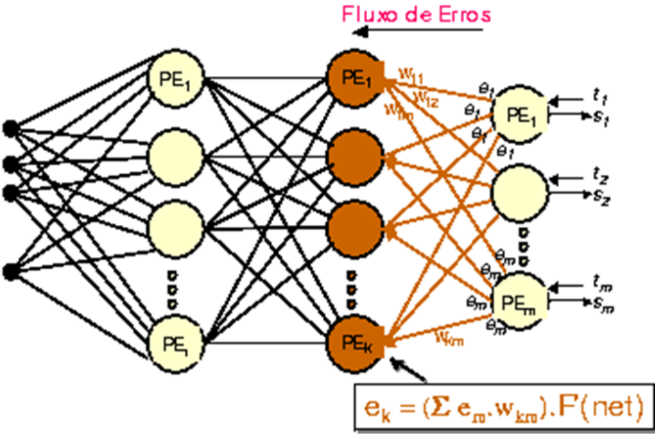

Figura 7: *Feed-backward* (fase 2), cálculo do erro da segunda camada escondida.

Processo de aprendizado

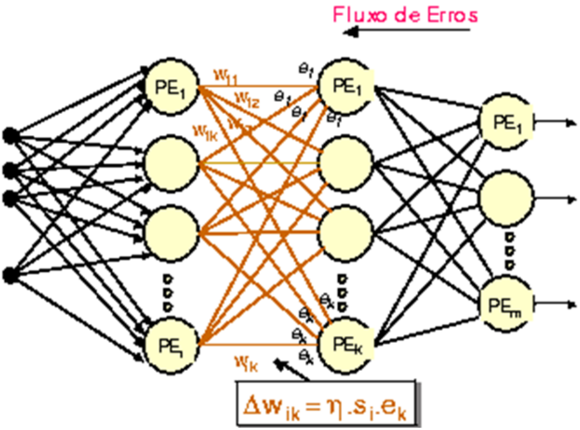

Figura 8: *Feed-backward* (fase 2), atualização dos pesos da segunda camada escondida.

Processo de aprendizado

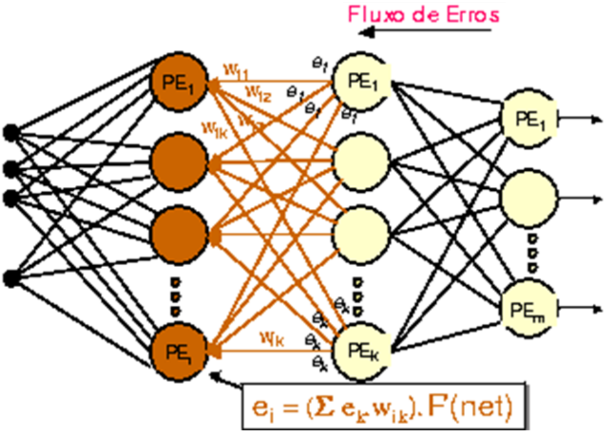

Figura 9: *Feed-backward* (fase 2), cálculo do erro da primeira camada escondida.

Processo de aprendizado

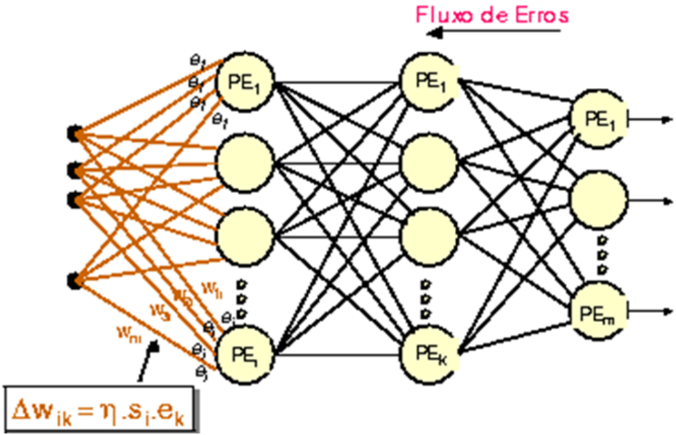

Figura 10: *Feed-backward* (fase 2), atualização dos pesos da primeira camada escondida.

Processo de aprendizado

- Este procedimento de aprendizado é repetido diversas vezes, até que, **para todos processadores de camada de saída e para todos padrões de treinamento**, o erro seja menor do que o especificado.

Observações

- Notem que para a atualização do gradiente local das camadas escondidas leva-se em consideração o gradiente local da camada posterior, e não diretamente o erro da rede.
- Este é um ponto crucial do algoritmo backpropagation.
- Utilizar o erro final durante o ajuste das camadas escondidas seria o equivalente a não estar realizando a retro-propagação do erro.

Sumário

Exemplo - XOR

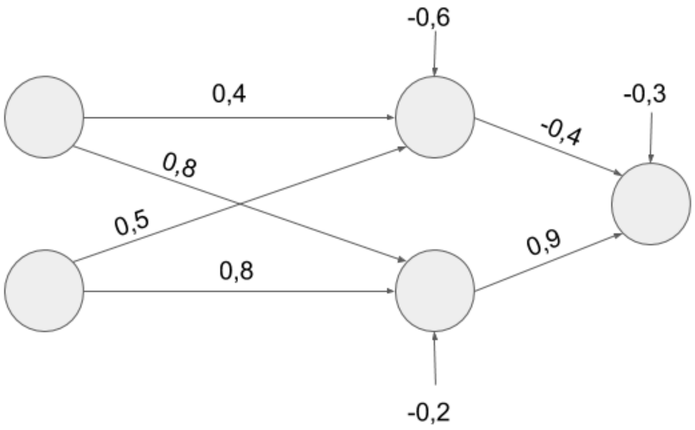

Figura 11: Rede neural inicial. Atualizar os pesos.

Exemplo - XOR

Taxa aprendizado	0,5					
	t	0	1	2	3	4
Entrada	x_1	1	0	0	1	
	x_2	1	0 Saída desejada	y	0	0
1	1					
Pesos	$w^{h1}_{\theta 1}$	-0,6				
	w^{h1}_{11}	0,4				
	w^{h1}_{21}	0,5				
	$w^{h1}_{\theta 2}$	-0,2				
	w^{h1}_{12}	0,8				
	w^{h1}_{22}	0,8				
	$w^{out}_{\theta 1}$	-0,3				
	w^{out}_{11}	-0,4				
	w^{out}_{21}	0,9				
Camada h_1	$v^{h1}_1(x)$					
	$v^{h1}_2(x)$					
	$f[v^{h1}_1(x)]$					
	$f[v^{h1}_2(x)]$					
Camada *out* (saída)	v^{out}_1					
	$y' = f[v^{out}_1]$					

Camada oculta h1 - forward

$$\begin{aligned}
v_1^{h1}(\mathbf{x}_{t=0}) &= 1 \cdot w_{\theta 1}^{h1}(0) + x_1(0) \cdot w_{11}^{h1}(0) + x_2(0) \cdot w_{21}^{h1}(0) \\
&= 1 \cdot -0.6 + 1 \cdot 0.4 + 1 \cdot 0.5 = \mathbf{0.3} \\
v_2^{h1}(\mathbf{x}_{t=0}) &= 1 \cdot w_{\theta}^{h1}(0) + x_1(0) \cdot w_{12}^{h1}(0) + x_2(0) \cdot w_{22}^{h1}(0) \\
&= 1 \cdot -0.2 + 1 \cdot 0.8 + 1 \cdot 0.8 = \mathbf{1.4} \\
f[v_1^{h1}(\mathbf{x}_{t=0})] &= \frac{1}{1 + e^{-v_1^{h1}(\mathbf{x}_{t=0})}} = \frac{1}{1 + e^{0.3}} = \mathbf{0.5744} \\
f[v_2^{h1}(\mathbf{x}_{t=0})] &= \frac{1}{1 + e^{-v_2^{h1}(\mathbf{x}_{t=0})}} = \frac{1}{1 + e^{1.4}} = \mathbf{0.8022}
\end{aligned}$$

Camada de saída - forward

$$v_1^{out}(\mathbf{x}_{t=0}) = 1 \cdot w_{\theta 1}^{out}(0) + f[v_1^{h1}(\mathbf{x}_{t=0})] \cdot w_{11}^{out}(0) + f[v_2^{h1}(\mathbf{x}_{t=0})] \cdot w_{21}^{out}(0)$$

$$= 1 \cdot -0.3 + 0.5744 \cdot (-0.4) + 0.8022 \cdot 0.9 = \mathbf{0.1922}$$

$$y' = f[v_1^{out}(h1)](\mathbf{x}_{t=0}) = \frac{1}{1 + e^{-v_1^{out}(\mathbf{x}_{t=0})}} = \frac{1}{1 + e^{0.1922}} = \mathbf{0.5479}$$

Exemplo - XOR

Taxa aprendizado	0,5					
	t	0	1	2	3	4
Entrada	x_1	1	0	0	1	
	x_2	1	0	1	0	
Saída desejada	y	0	0	1	1	
Pesos	$w_{\theta 1}^{h1}$	-0,6				
	w_{11}^{h1}	0,4				
	w_{21}^{h1}	0,5				
	$w_{\theta 2}^{h1}$	-0,2				
	w_{12}^{h1}	0,8				
	w_{22}^{h1}	0,8				
	$w_{\theta 1}^{out}$	-0,3				
	w_{11}^{out}	-0,4				
	w_{21}^{out}	0,9				
Camada h_1	$v_1^{h1}(x)$	0,3				
	$v_2^{h1}(x)$	1,4				
	$f[v_1^{h1}(x)]$	0,5744				
	$f[v_2^{h1}(x)]$	0,8022				
Camada *out* (saída)	v_1^{out}	0,1922				
	$y' = f[v_1^{out}]$	0,5479				

Backpropagation

Camada de saída

$$w_{ji}(t) = w_{ji}(t-1) - \eta \cdot (t_{pj} - y_{pj}) \cdot y_{pj}(1 - y_{pj}) \cdot y_{pi}$$

$$w_{\theta 1}^{out}(t=1) = -0.3 + \overbrace{0.5}^{\eta} \cdot \overbrace{(0 - 0.5479) \cdot 0.5479(1 - 0.5479)}^{-\delta_{pj} = -0.1357} \cdot 1 = -\mathbf{0.3679}$$

$$w_{11}^{out}(t=1) = -0.4 + 0.5 \cdot (-0.1357) \cdot \overbrace{0.5744}^{y_{pi} = f[v_1^{h1}(x)]} = -\mathbf{0.4390}$$

$$w_{21}^{out}(t=1) = 0.9 + 0.5 \cdot (-0.1357) \cdot \overbrace{0.8022}^{y_{pi} = f[v_2^{h1}(x)]} = \mathbf{0,8456}$$

Exemplo - XOR

Taxa aprendizado	0,5					
	t	0	1	2	3	4
Entrada	x_1	1	0	0	1	
	x_2	1	0	1	0	
Saída desejada	y	0	0	1	1	
Pesos	$w_{\theta 1}^{h1}$	-0,6				
	w_{11}^{h1}	0,4				
	w_{21}^{h1}	0,5				
	$w_{\theta 2}^{h1}$	-0,2				
	w_{12}^{h1}	0,8				
	w_{22}^{h1}	0,8				
	$w_{\theta 1}^{out}$	-0,3	-0.3679			
	w_{11}^{out}	-0,4	-0.4390			
	w_{21}^{out}	0,9	0,8456			
Camada h_1	$v_1^{h1}(x)$	0,3				
	$v_2^{h1}(x)$	1,4				
	$f[v_1^{h1}(x)]$	0,5744				
	$f[v_2^{h1}(x)]$	0,8022				
Camada *out* (saída)	v_1^{out}	0,1922				
	$y' = f[v_1^{out}]$	0,5479				

Backpropagation

Camada oculta

$$w_{ji}(t) = w_{ji}(t-1) - \eta \cdot \left(\sum_k (-\delta_{pk} w_{kj}) \right) \cdot y_{pj}(1 - y_{pj}) \cdot y_{pi}$$

$$w^{h1}_{\theta 1} = -0.6 + 0.5 \cdot \overbrace{(-0.1357)}^{-\delta_{pj}} \cdot \overbrace{(-0.4)}^{w^{out}_{11}(t=0)} \cdot \overbrace{0.5744}^{y_{pj}=f[v^{h1}_1(x)]} \cdot (1 - 0.5744) \cdot 1 = \mathbf{-0.5934}$$

$$w^{h1}_{11} = 0.4 + 0.5 \cdot (-0.1357) \cdot (-0.4) \cdot 0.5744 \cdot (1 - 0.5744) \cdot \overbrace{1}^{y_{pi}=x_1} = \mathbf{0.4066}$$

$$w^{h1}_{21} = 0.5 + 0.5 \cdot (-0.1357) \cdot (-0.4) \cdot 0.5744 \cdot (1 - 0.5744) \cdot \overbrace{1}^{y_{pi}=x_2} = \mathbf{0.5066}$$

Backpropagation

$$w_{\theta 2}^{h1} = -0.2 + 0.5 \cdot (-0.1357) \cdot \overbrace{0.9}^{w_{12}^{out}(t=0)} \cdot \overbrace{0.8022}^{y_{pj}=f[v_2^{h1}(x)]} \cdot (1-0.8022) \cdot 1 = \mathbf{-0,2097}$$

$$\mathrm{w}_{12}^{h1} = 0.8 + 0.5 \cdot (-0.1357) \cdot 0.9 \cdot 0.8022 \cdot (1 - 0.8022) \cdot 1 = \mathbf{0,7903}$$

$$\mathrm{w}_{22}^{h1} = 0.8 + 0.5 \cdot (-0.1357) \cdot 0.9 \cdot 0.8022 \cdot (1 - 0.8022) \cdot 1 = \mathbf{0,7903}$$

Exemplo - XOR

Taxa aprendizado	0,5					
	t	0	1	2	3	4
Entrada	x_1	1	0	0	1	
	x_2	1	0	1	0	
Saída desejada	y	0	0	1	1	
Pesos	$w_{\theta 1}^{h1}$	-0,6	-0,5934			
	w_{11}^{h1}	0,4	0,4066			
	w_{21}^{h1}	0,5	0,5066			
	$w_{\theta 2}^{h1}$	-0,2	-0,2097			
	w_{12}^{h1}	0,8	0,7903			
	w_{22}^{h1}	0,8	0,7903			
	$w_{\theta 1}^{out}$	-0,3	-0.3679			
	w_{11}^{out}	-0,4	-0.4390			
	w_{21}^{out}	0,9	0,8456			
Camada h_1	$v_1^{h1}(x)$	0,3				
	$v_2^{h1}(x)$	1,4				
	$f[v_1^{h1}(x)]$	0,5744				
	$f[v_2^{h1}(x)]$	0,8022				
Camada *out* (saída)	v_1^{out}	0,1922				
	$y' = f[v_1^{out}]$	0,5479				

Backpropagation

- Completa-se uma **época** ao se atualizarem todos os exemplos de treinamento uma vez.
  - $(0,0) \rightarrow 0$
  - $(0,1) \rightarrow 1$
  - $(1,0) \rightarrow 1$
  - $(1,1) \rightarrow 0$

Exemplo - XOR

Taxa aprendizado	0,5					
	t	0	1	2	3	4
Entrada	x_1	1	0	0	1	
	x_2	1	0	1	0	
Saída desejada	y	0	0	1	1	
Pesos	$w_{\theta 1}^{h1}$	-0,6	-0,5934	-0,5876	-0,5951	-0,6018
	w_{11}^{h1}	0,4	0,4066	0,4066	0,4066	0,4000
	w_{21}^{h1}	0,5	0,5066	0,5066	0,4991	0,4991
	$w_{\theta 2}^{h1}$	-0,2	-0,2097	-0,2217	-0,2092	-0,1968
	w_{12}^{h1}	0,8	0,7903	0,7903	0,7903	0,8027
	w_{22}^{h1}	0,8	0,7903	0,7903	0,8028	0,8028
	$w_{\theta 1}^{out}$	-0,3	-0,3679	-0,4255	-0,3594	-0,2969
	w_{11}^{out}	-0,4	-0,4390	-0,4595	-0,4278	-0,3995
	w_{21}^{out}	0,9	0,8456	0,8197	0,8619	0,9020
Camada h_1	$v_1^{h1}(x)$	0,3	-0,5934	-0,0809	-0,1885	
	$v_2^{h1}(x)$	1,4	-0,2097	0,5686	0,5811	
	$f[v_1^{h1}(x)]$	0,5744	0,3559	0,4798	0,4530	
	$f[v_2^{h1}(x)]$	0,8022	0,4478	0,6384	0,6413	
Camada *out* (saída)	v_1^{out}	0,1922	-0,1455	-0,1226	-0,0005	
	$y' = f[v_1^{out}]$	0,5479	0,4637	0,4694	0,4999	