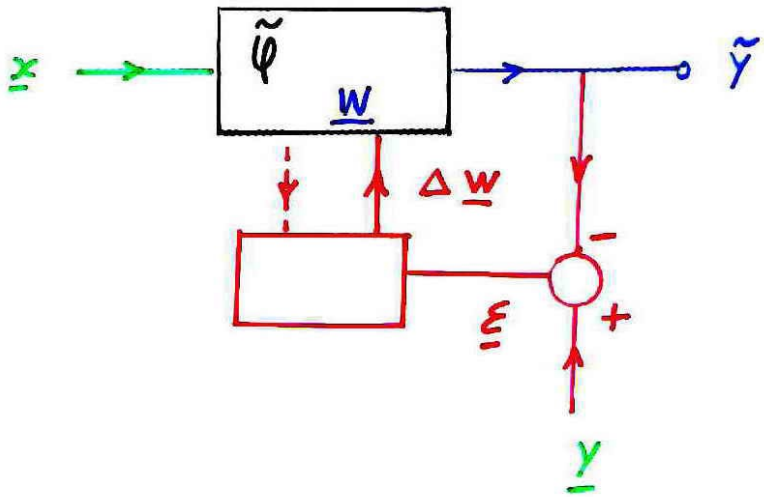
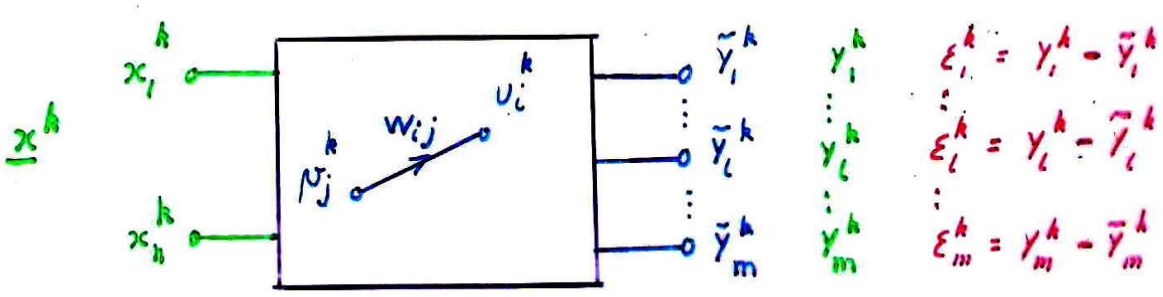
Treinamento Backpropagation de uma Rede Neural Feedforward



Erro na saída **Função objetivo à minimizar**

A saída pode ser multidimensional:



Erro quadrático na saída para o k-ésimo par entrada-saída k

$$\left(\varepsilon^k\right)^2 = \left|\underline{y}^k - \underline{\tilde{y}}^k\right|^2 = \sum_{l=1}^{L}\left(y_l^k - \tilde{y}_l^k\right)^2 = \sum_{l=1}^{L}\left(\varepsilon_l^k\right)^2$$

onde $\varepsilon_l^k = y_l^k - \tilde{y}_l^k$ e $\tilde{y}_l^k = \tilde{y}_l^k(\underline{w})$

Erro quadrático **médio** para todos os P pares entrada-saída

$(\underline{x}^k, \underline{y}^k) \qquad k = 1, \ldots P$

Erro médio quadrático:

$$F(\underline{w}) = \underset{k}{E} (\varepsilon^k)^2 = \frac{1}{P} \sum_{k=1}^{P} (\varepsilon^k)^2$$

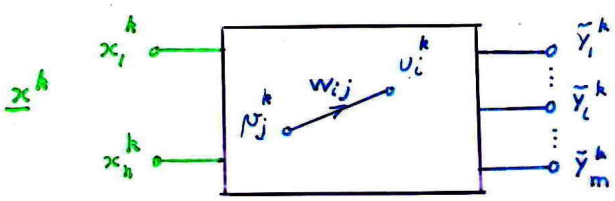
Função objetivo à minimizar

Acréscimo a aplicar em cada sinapse:

$$F_0 = E(\varepsilon^{2p}) = F_0(\underline{w}) \qquad \Delta w_{ij} = -\alpha \frac{\partial F_0(w_{ij})}{\partial w_{ij}}$$

Como calcular

$$\frac{\partial F_0(w_{ij})}{\partial w_{ij}} \quad ???$$



$$1 - \quad F_0(w_{ij}) = \underset{k}{E}\left[\left(\varepsilon^k\right)^2 \right] = \frac{1}{P}\sum_{k=1}^{P}(\varepsilon^k)^2$$

$$\frac{\partial}{\partial w_i} \underset{\forall k}{E} (\varepsilon^k)^2 = \frac{\partial}{\partial w_i} \frac{1}{P} \sum_{k=1}^{P} (\varepsilon^k)^2 = \frac{1}{P} \sum_{k=1}^{P} \frac{\partial}{\partial w_i} (\varepsilon^k)^2 = \underset{\forall k}{E} \frac{\partial}{\partial w_i} (\varepsilon^k)^2$$

$$\nabla \underset{\forall k}{E} (\varepsilon^k)^2 = \underset{\forall k}{E} \nabla (\varepsilon^k)^2$$

Propriedade importante:

o gradiente do valor esperado do erro quadrático é igual ao valor esperado do gradiente do erro quadrático

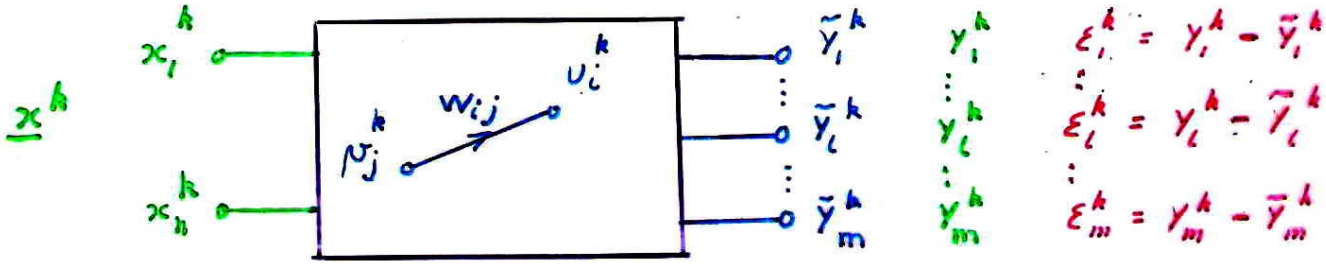
cada par entrada-saída é tratado isoladamente.

Consequência:

Calcularemos separadamente para cada par $\frac{\partial}{\partial w_i} (\varepsilon^k)^2 \quad \forall\ k$

e tomaremos a média $\frac{\partial}{\partial w_i} \underset{k}{E} (\varepsilon^k)^2 = \underset{k}{E} \frac{\partial}{\partial w_i} (\varepsilon^k)^2$

2 – Cálculo de $\frac{\partial\left(\varepsilon^k\right)^2}{\partial w_{ij}}$ para cada par entrada – saída k



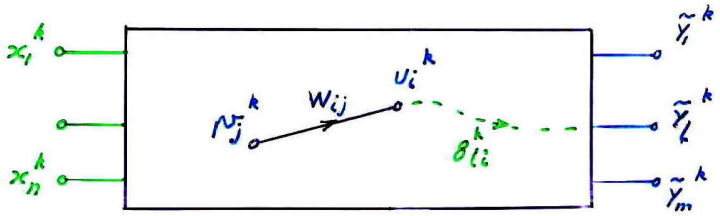
$$(\varepsilon)^2 = \sum_{l=1}^{m} \left(\varepsilon_l\right)^2 = \sum_{l=1}^{m} \left(y_l - \tilde{y}_l\right)^2$$

$$\frac{\partial(\varepsilon)^2}{\partial w_{ij}} \quad ???$$

$$\frac{\partial(\varepsilon)^2}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_{l=1}^{m} \varepsilon_l^{\,2} = \sum_{l=1}^{m} \frac{\partial \varepsilon_l^{\,2}}{\partial w_{ij}} = 2\sum_{l=1}^{m} \varepsilon_l \frac{\partial \varepsilon_l}{\partial w_{ij}} =$$

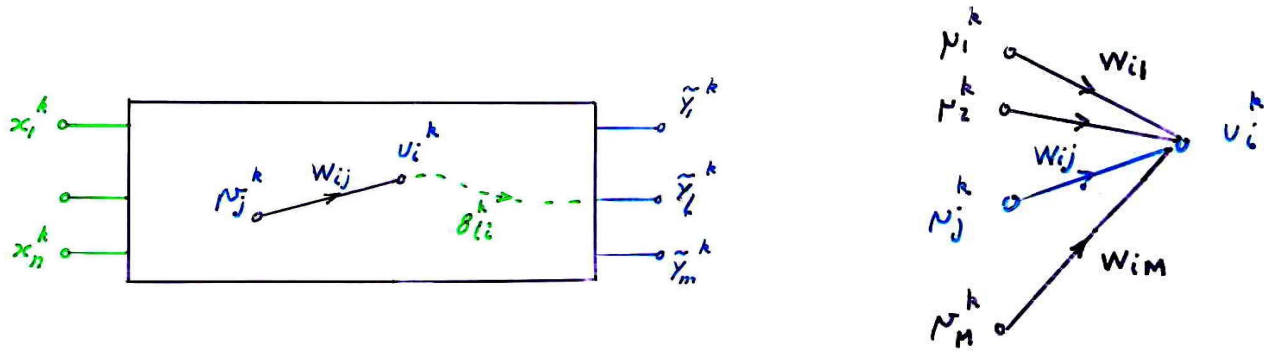
$$= 2\sum_{l=1}^{m} \varepsilon_l \frac{\partial\left(y_l - \tilde{y}_l\right)}{\partial w_{ij}} = -2\sum_{l=1}^{m} \varepsilon_l \frac{\partial \tilde{y}_l}{\partial w_{ij}} = -2\sum_{l=1}^{m} \varepsilon_l \frac{\partial \tilde{y}_l}{\partial u_i} \frac{\partial u_i}{\partial w_{ij}}$$

$$\frac{\partial \tilde{y}_l}{\partial u_i} = g_{li}$$



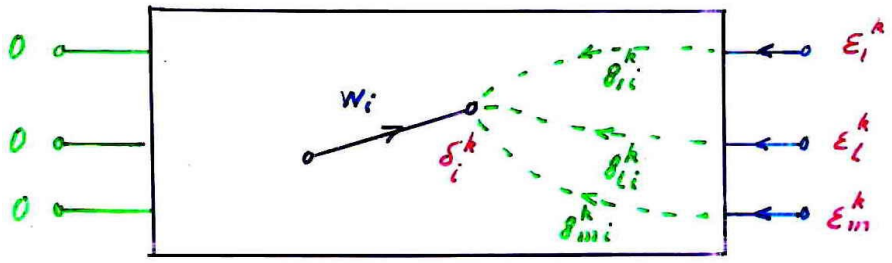
$$\frac{\partial(\varepsilon)^2}{\partial w_{ij}} = -2\sum_{l=1}^{m} \varepsilon_l \frac{\partial \tilde{y}_l}{\partial u_i}\frac{\partial u_i}{\partial w_{ij}} = -2\sum_{l=1}^{m} \varepsilon_l \; g_{li} \; v_j$$

$$\frac{\partial \tilde{y}_l}{\partial u_i} = g_{li} \qquad u_i = \sum_{m=1}^{M} w_{mj} v_j \qquad \frac{\partial u_i}{\partial w_{ij}} = v_j$$



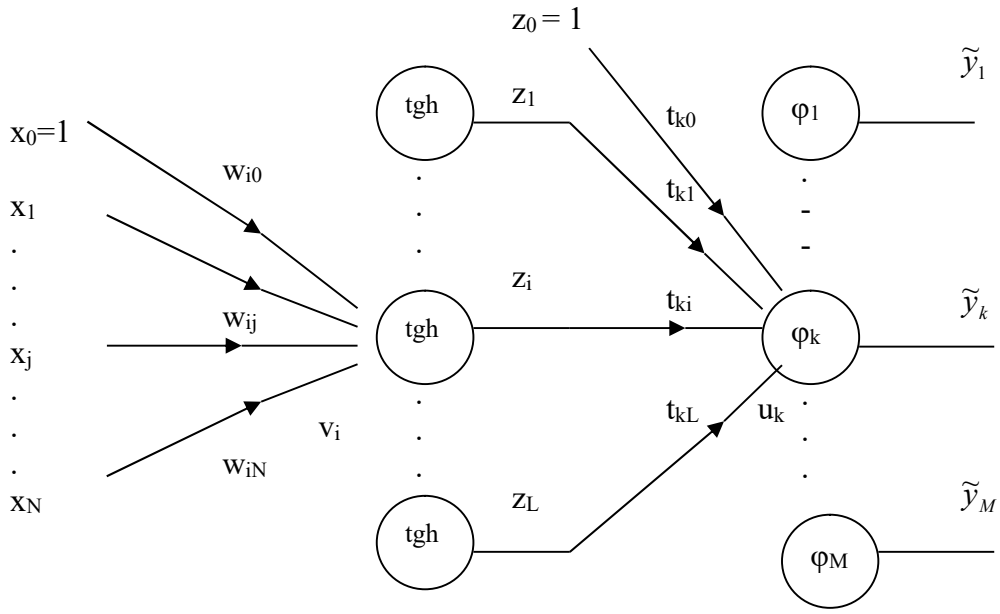
$$\frac{\partial(\varepsilon)^2}{\partial w_{ij}} = -2\sum_{l=1}^{m} \varepsilon_l \; g_{li} \; v_j = -2v_j \sum_{l=1}^{m} \varepsilon_l g_{li} = -2 \; v_j \; \delta_i$$

$$\delta_i = \sum_{l=1}^{m} \varepsilon_l \frac{\partial \tilde{y}_l}{\partial u_i} = \sum_{l=1}^{m} \varepsilon_l \; g_{li}$$



δ_i é o erro retropropagado da saída até a extremidade da sinapse

Cálculo de $g_{li} = \frac{\partial \tilde{y}_k}{\partial u_i}$



Cálculo de $g_{li} = \frac{\partial \tilde{y}_k}{\partial u_i}$ Camada de saída

$$\tilde{y}_k = \varphi_k(u_k) = \begin{cases} u_k & \text{neurônio linear} \\ tgh\,(u_k) & \text{neurônio tgh} \end{cases}$$

$$g_{ki} = \frac{\partial \tilde{y}_k}{\partial u_i} = \begin{cases} 0 \;\; se \;\; i \neq k \\ se \;\; i = k \end{cases} \quad g_{kk} = \frac{\partial \tilde{y}_k}{\partial u_k} = \begin{cases} 1 & \text{neurônio linear} \\ 1 - \tilde{y}_k^2 & \text{neurônio tgh} \end{cases}$$

Cálculo de $g_{li} = \frac{\partial \tilde{y}_k}{\partial u_i}$ Camada intermediária

$$\tilde{y}_k = \phi_k(u_k) = \begin{cases} u_k & \text{neurônio de saída linear} \\ tgh\,(u_k) & \text{neurônio de saída tgh} \end{cases}$$

$$u_k = \sum_{i=0}^{L} t_{ki} z_i \qquad z_i = tgh\,(v_i) \qquad z_0 = 1$$

$$g_{ki} = \frac{\partial \tilde{y}_k}{\partial v_i} = \frac{\partial \tilde{y}_k}{\partial u_k}\frac{\partial u_k}{\partial z_i}\frac{\partial z_i}{\partial v_i} = \begin{cases} 0 \;\; se \;\; k \neq i \\ se \;\; k = i \;\; \begin{cases} t_{ki}(1-z_i^2) & \text{neurônio de saída linear} \\ (1-\tilde{y}_i^2)t_{ki}(1-z_i^2) & \text{neurônio de saída tgh} \end{cases} \end{cases}$$

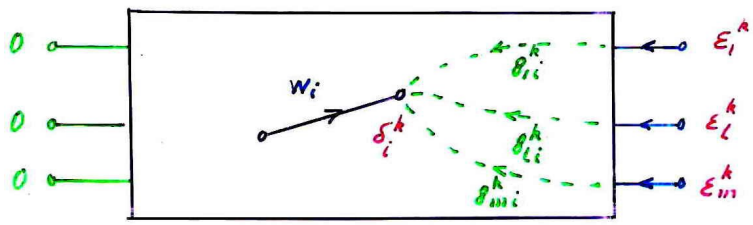
ou

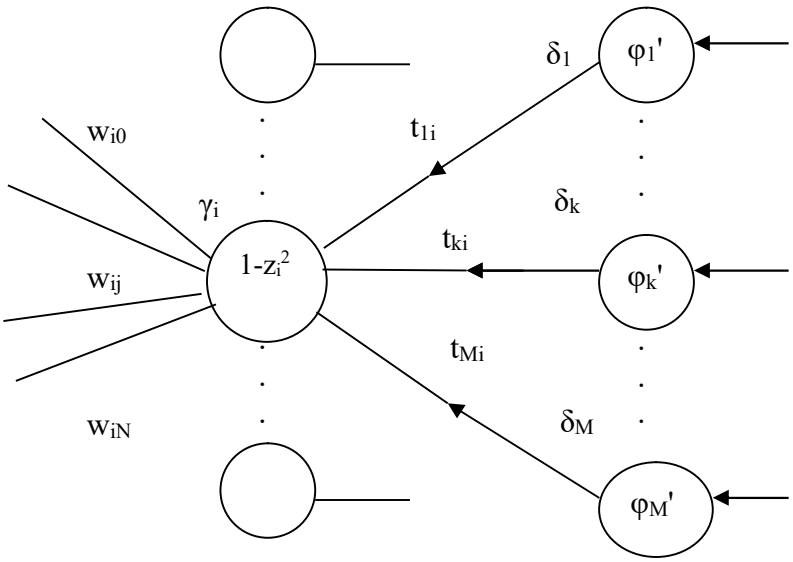
$$\frac{\partial \tilde{y}_k}{\partial v_i} = G_S G_I$$ ganho neurônio saída x ganho intermediário

Rede Associada

$$\delta_k = G_{Sk}\varepsilon_k$$

$$\gamma_i = \sum_{\forall k} G_{Ik} G_{Sk} \varepsilon_k = \sum_{\forall k} G_{Ik}\delta_k$$





3 - como:

$$\Delta w_{ij} = -\alpha \frac{\partial F_0(w_{ij})}{\partial w_{ij}}$$

$$\frac{\partial F_0(w_{ij})}{\partial w_{ij}} = \frac{1}{P}\sum_{p=1}^{P} \frac{\partial}{\partial w_{ij}}(\varepsilon^{2^p})$$

$$\frac{\partial(\varepsilon)^2}{\partial w_{ij}} = -2\, v_j\, \delta_i$$ para um par entrada -saída

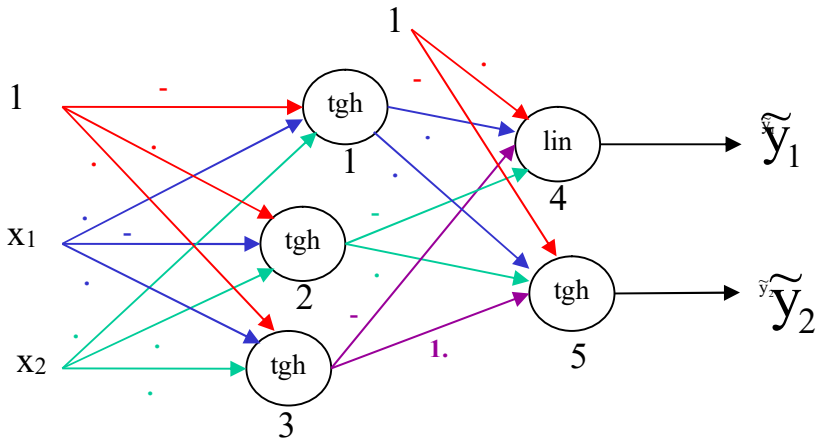
Para os P pares entrada saída

$$\Delta w_{ij} = -\alpha \underset{\forall p}{E} \frac{\partial(\varepsilon)^2}{\partial w_{ij}}\bigg|_p$$

$$\Delta w_{ij} = 2\alpha \frac{1}{P}\sum_{p=1}^{P} v_j\, \delta_i\big|_p$$

Error Backpropagation - Princípio do Algoritmo:

1 - Rede Original:



Propagar o sinal na rede original e conservar o valor do sinal v_j na entrada de cada sinapse w_{ij}. Calcular o erro em cada saída.

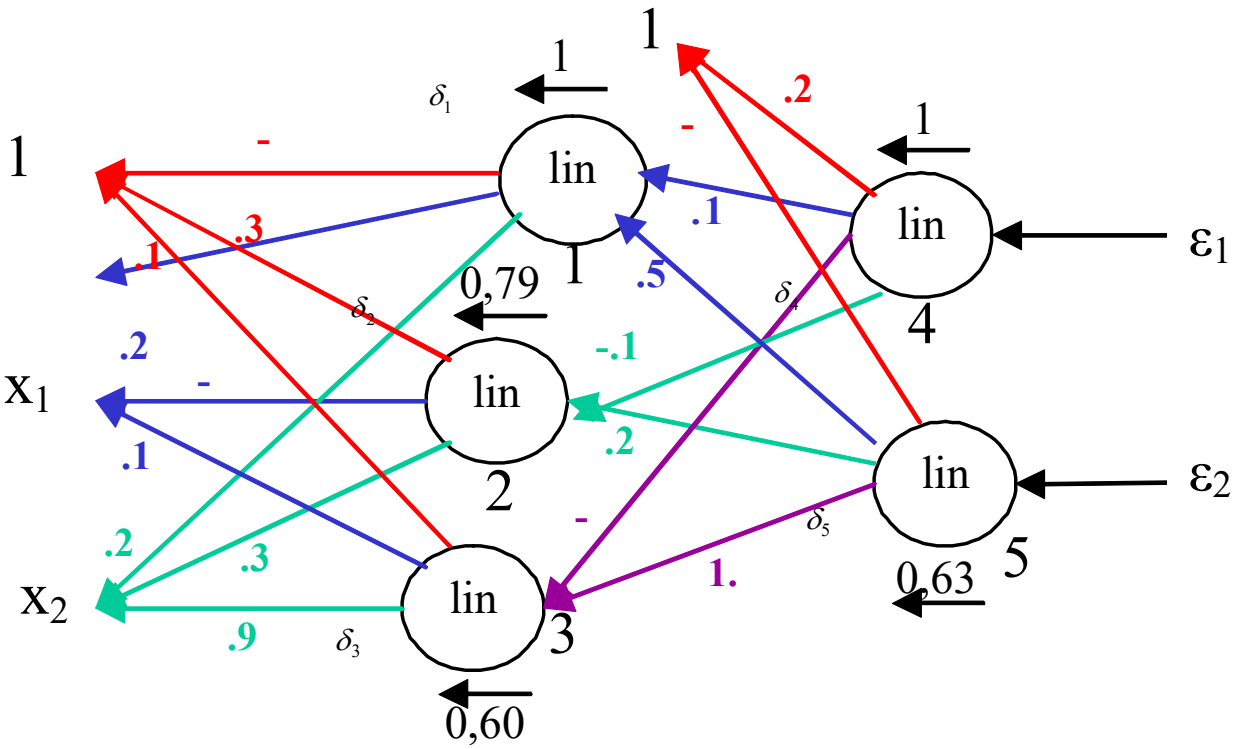
2 – Criar uma "Rede Associada" a partir da Rede Original

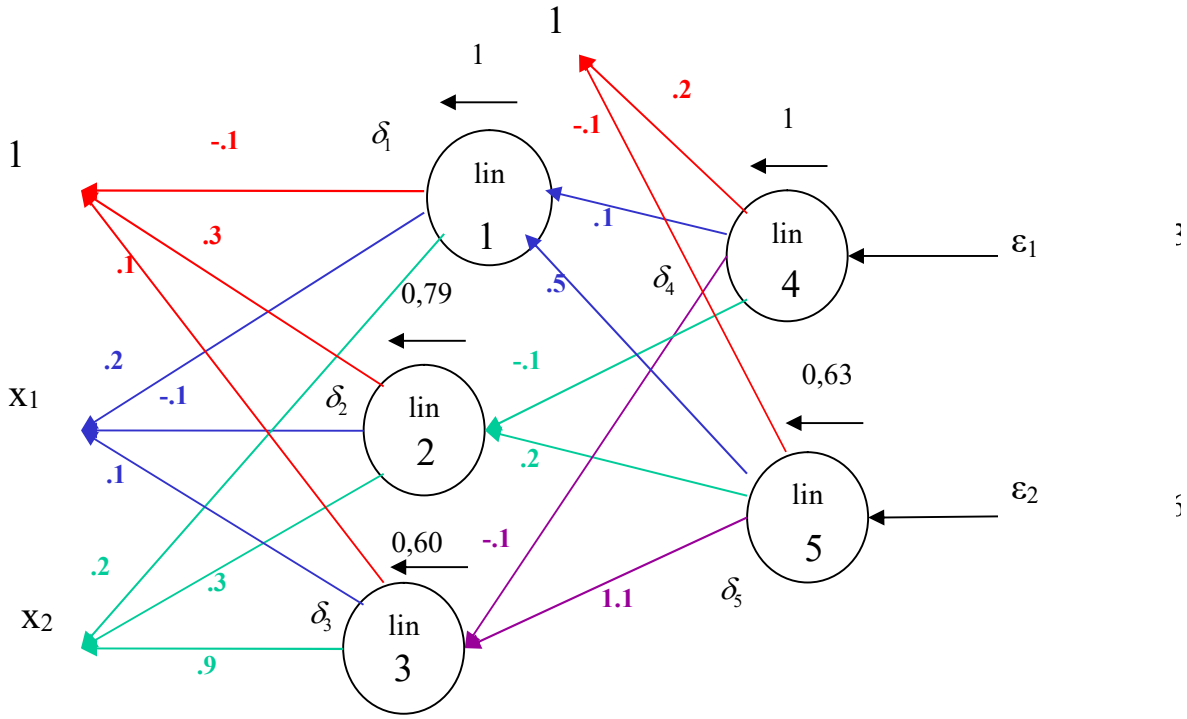
mesma arquitetura, mas

2.1 - "Linearizar" (os neurônios) da rede original

Rede Original	→	**Rede Associada**
Neurônio não linear $v_0 = tgh(u_0)$	→	**Neurônio linear** $v = (1-v_0^2)\, u$
Neurônio linear $v_0 = u_0$	→	**Neurônio linear** $v = u$

2.2- Inverter todos os sentidos de transmissão (dos neurônios e sinapses)





4 – O acréscimo para cada sinapse wij para o par p entrada-saída em operação é dado por:

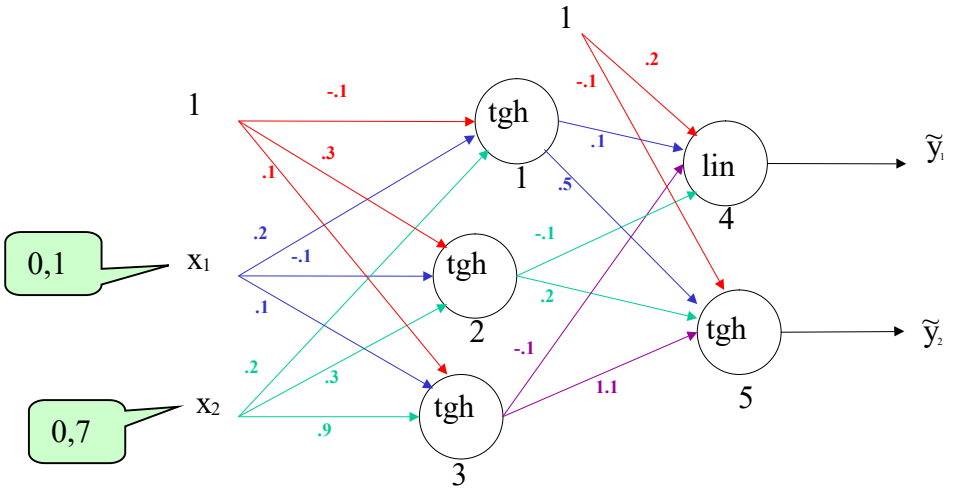
$$\Delta_p w_{ij} = 2\,\alpha\, v_j\, \delta_i$$

5 – O acréscimo a ser aplicado na sinapse w_{ij} é o valor esperado dos acréscimos calculados para todos os pares entrada-saída.

$$\Delta w_{ij} = \underset{p}{E}\left(\Delta_p w_{ij}\right) = 2\,\alpha\,\underset{p}{E}\left(v_j \delta_i\Big|_p\right)$$

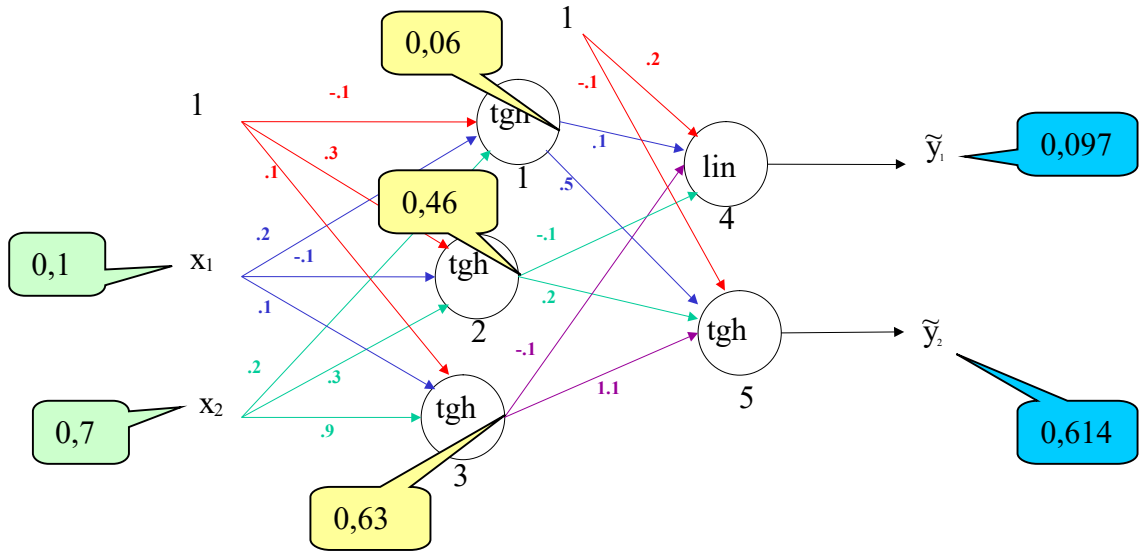
Processo em Batelada – Batch

Exemplo anterior (regra delta):



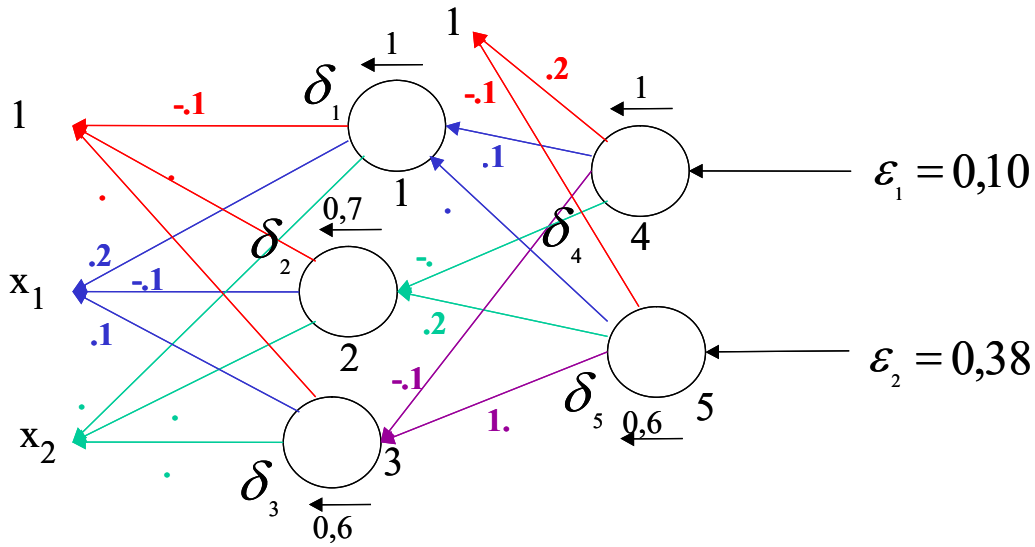
$$\underline{\mathbf{x}} = \begin{bmatrix} 0,1 \\ 0,7 \end{bmatrix} \qquad \tilde{\mathbf{y}} = ? \qquad \mathbf{y} = \begin{bmatrix} 0,2 \\ 0,1 \end{bmatrix}$$

Signal Feedforward



$$\underline{\mathbf{x}} = \begin{bmatrix} 0,1 \\ 0,7 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 0,2 \\ 1 \end{bmatrix} \qquad \tilde{\mathbf{y}} = \begin{bmatrix} 0,097 \\ 0,614 \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} 0,103 \\ 0,386 \end{bmatrix}$$

Rede associada e retropropagação do erro:



$$\delta_5 = (0,386)(0,63) = 0,24 \qquad \delta_4 = (0,103)(1) = 0,103$$

$$\delta_3 = [(0,103)(-0,1) + (0,24)(1,1)](0,60) = 0,152$$

$$\delta_2 = [(0,103)(-0,1) + (0,24)(0,2)](0,79) = 0,030$$

$$\delta_1 = [(0,103)(0,1) + (0,24)(0,5)](1) = 0,130$$

Treinamento Regra Delta

Atualização dos valores das sinapses:

$$\Delta w_{ij} = 2\alpha v_j \delta_i$$

$$\Delta b_4 = (2)(0,1)(1)(0,103) = 0,021$$

$$b_4 \rightarrow 0,2 + 0,021 = 0,221$$

$$\Delta b_1 = (2)(0,1)(1)(0,130) = 0,026$$

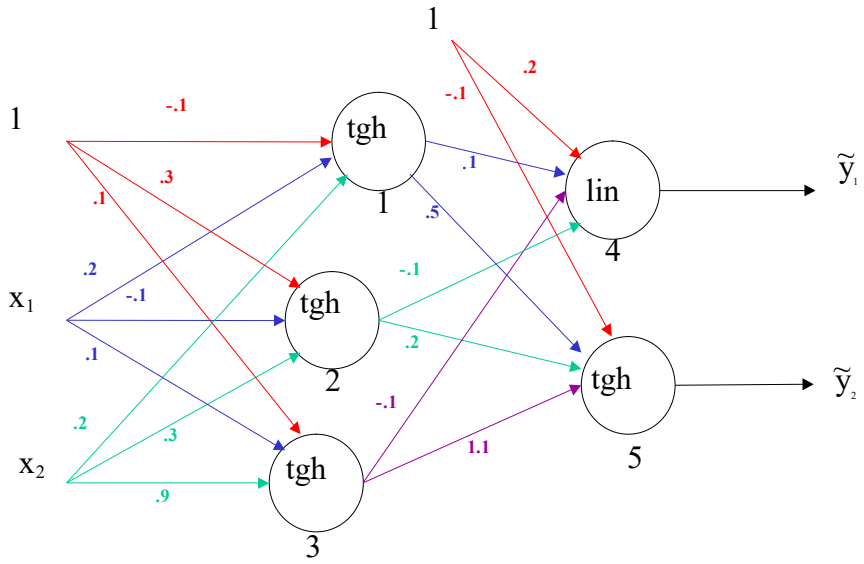
$$b_1 \rightarrow -0,1 + 0,026 = -0,074$$

$$\Delta w_{41} = (2)(0,1)(0,06)(0,103) = 0,001$$

$$w_{41} \rightarrow 0,1 + 0,001 = 0,101$$

$$\Delta w_{3b} = (2)(0,1)(0,7)(0,152) = 0,021$$

$$w_{3b} \rightarrow 0,9 + 0,021 = 0,921$$



Algorítmos BP

para cálculo do acréscimo na sinapse w_{ij} devido ao par p, $\Delta_p w_{ij}$ para dois casos específicos:

Redes com uma única camada

Definição das variáveis:

Rede Original

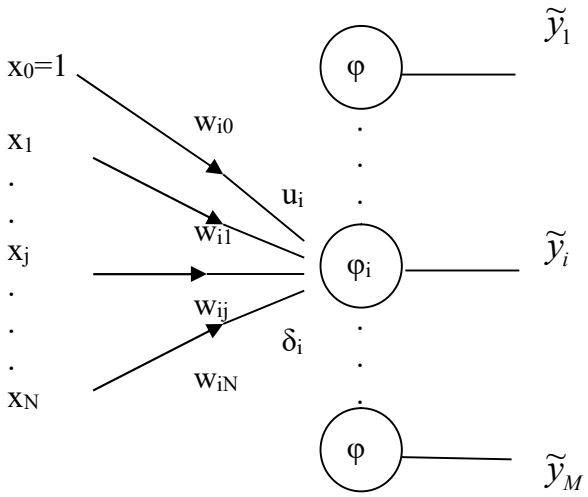
N entradas: $x_1, x_2, ..., x_N$ e x_0 (bias) = 1

w_{ij} - sinapse conectando a entrada j ao neurônio i

M neurônios com função de ativação φ(.) linear ou tgh

u_i - excitação interna do neurônio i

$\tilde{y}_i$ - saída do neurônio i



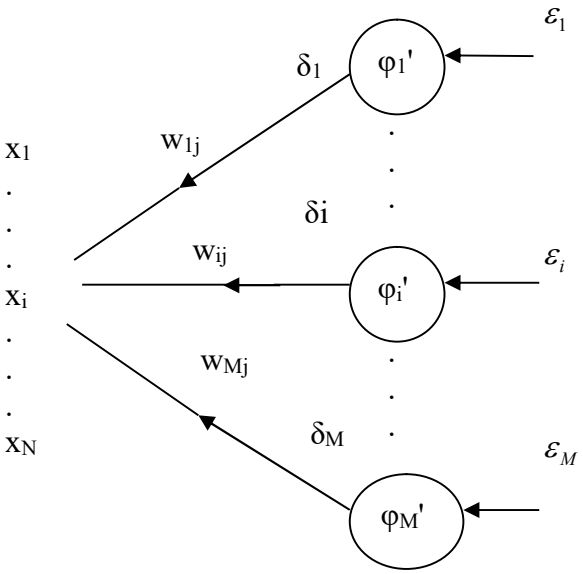
Definição das variáveis:

Rede Associada

ε_i - erro na saída do neurônio i

δ_i - erro retropropagado até a entrada do neurônio i

φ_i' - valor numérico da derivada de φ_i no ponto de operação



Algorítmo BP acréscimo Δw_{ij}^{p} - Rede 1 camada

Para o par entrada-saída p $(\underline{x}, \underline{y})$

Signal feedforward: Propague o sinal para a frente, calcule a saída e o erro em cada saída:

$$u_i = \sum_{j=0}^{N} w_{ij} x_j$$

$$\tilde{y}_i = \varphi_i(u_i) = \begin{cases} u_i & \textit{neurônio linear} \\ tgh\,(u_i) & \textit{neurônio tgh} \end{cases}$$

$$\varepsilon_i = y_i - \tilde{y}_i$$

Calcule o erro retropropagado até a entrada de cada neurônio:

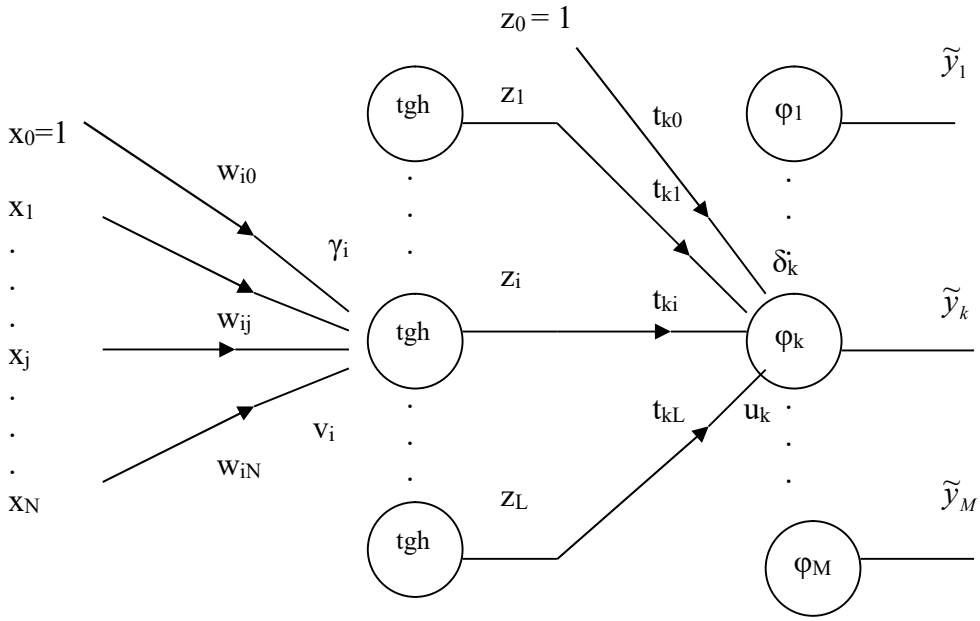
$$\delta_i = \begin{cases} \varepsilon_i & \textit{neurônio linear} \\ \left(1 - \tilde{y}_i^2\right)\varepsilon_i & \textit{neurônio tgh} \end{cases}$$

Calcule o acéscimo em cada sinapse devido ao par p:

$$\Delta w_{ij}^{p} = 2\alpha\, x_j \delta_i$$

fim do algorítmo

Redes com duas camadas - Rede Original



Definição das variáveis:

N entradas: $x_1, x_2, ..., x_N$ e x_0 (bias) = 1

L neurônios na primeira camada com função de ativação tgh

M neurônios na camada de saída com função de ativação φ(.) linear ou tgh

w_{ij} - sinapse conectando a entrada j ao neurônio i da camada intermediária

t_{ki} - sinapse conectando a saída z_i do neurônio i da camada intermediária com a entrada do neurônio k da camada de saída.

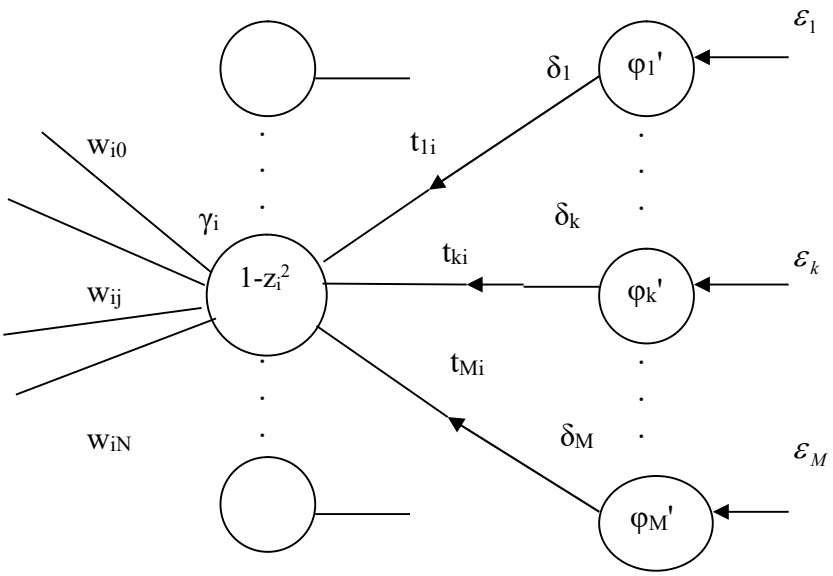
v_i - excitação interna do neurônio i da primeira camada

z_i - saída do neurônio i da camada intermediária. z_0 (bias) = 1

u_k - excitação interna do neurônio k da camada intermediária

$\tilde{y}_k$ - saída do neurônio k da camada de saída

Rede Associada



Definição das variáveis:

ε_k - erro na saída do neurônio da camada de saída

δ_k - erro retropropagado até a entrada do neurônio k da camada de saída

φ_i' - valor numérico da derivada de φ_i no ponto de operação

Algorítmo BP acréscimos Δw_{ij}^p e Δt_{ki}^p - Rede com 2 camadas

Para o par entrada-saída p $(\underline{x}, \underline{y})$

Signal feedforward: Propague o sinal para a frente, calcule a saída e o erro em cada saída:

Para todos os neurônios da primeira camada i = 1,...,L

$$v_i = \sum_{j=0}^{N} w_{ij} x_j \qquad x_0 = 1$$

$$z_i = tgh\,(v_i) \qquad z_0 = 1$$

Para todos os neurônios da segunda camada k = 1,..., M

$$u_k = \sum_{i=0}^{L} t_{ki} z_i \qquad z_0 = 1$$

$$\tilde{y}_k = \varphi_k(u_k) = \begin{cases} u_k & \text{neurônio linear} \\ tgh\,(u_k) & \text{neurônio tgh} \end{cases}$$

$$\varepsilon_k = y_k - \tilde{y}_k$$

Retropropagação do erro

Para todo neurônio da segunda camada, k = 1,...,M:

$$\delta_k = \begin{cases} \varepsilon_k & \textit{neurônio linear} \\ \left(1-\tilde{y}_k^2\right)\varepsilon_k & \textit{neurônio tgh} \end{cases}$$

Para todo neurônio da primeira camada, i = 1,...,L

$$\gamma_i = \left(1-z_i^2\right)\sum_{k=1}^{M} t_{ki}\delta_k$$

Acréscimo nas sinapses da primeira camada devido ao par p:

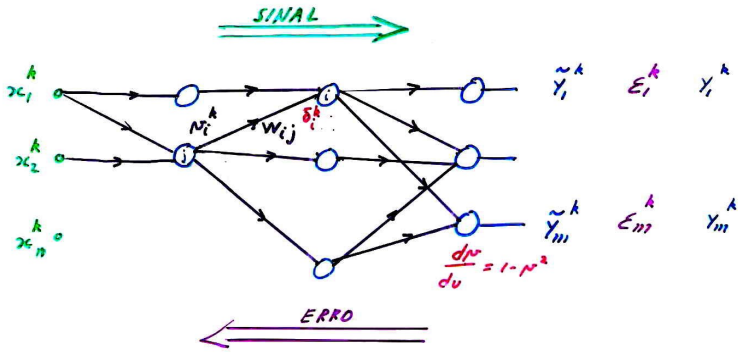
$$\Delta w_{ij}^p = 2\,\alpha\, x_j \gamma_i \qquad \forall\, j = 0,1,....,N \quad e \quad i = 1,2,...,L$$

Acréscimo nas sinapses da segunda camada devido ao par p:

$$\Delta t_{ki}^p = 2\alpha\, z_i\, \delta_k \qquad \forall\, i = 0,1,...,L \quad e \quad k = 1,2,...,M$$

fim do algorítmo

Error Backpropagation - Resumo:



$$\frac{\partial \varepsilon^{2\,k}}{\partial w_{ij}} = -2\,\mu_j^k\,\delta_i^k$$

$$\nabla F_0 = \frac{\partial F_0}{\partial w_{ij}} = -2\,E(\mu_j\,\delta_i)$$

$$\Delta w_{ij} = 2\,\alpha\,E(\mu_j\,\delta_i)$$