

1 – Escreva a relação entre as funções de ativação logística $L(u)$ e tangente hiperbólica $tgh(u)$, e também entre suas derivadas $dL(u)/du$ e $d\,tgh(u)/du$.

$$L(u) = \frac{1}{1 + e^{-2u}}$$

$$tgh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} = \frac{1 - e^{-2u}}{1 + e^{-2u}}$$

$$v = \frac{1}{1 + e^{-2u}} \quad \frac{dv}{du} = 2v(1 - v)$$

$$v = tgh(u) = \frac{1 - e^{-2u}}{1 + e^{-2u}} \quad \frac{dv}{du} = 1 - v^2$$

$$tgh\,v = 2\,lg\,v - 1 \quad lg\,v = \frac{1}{2} tgh\,v + \frac{1}{2}$$

1.1 - Mostre que o custo computacional de treinar ou operar as duas redes é similar.

$$v = tgh\,v = \frac{1 - e^{-2v}}{1 + e^{-2v}} \quad \frac{dv}{du} = 1 - v^2$$

++ x / exp + x

$$v = lg\,v = \frac{1}{1 + e^{-2v}} \quad \frac{dv}{du} = 2v(1 - v)$$

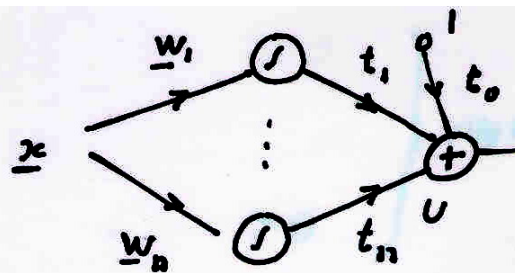
+ x / exp x x +

Carga computacional equivalente

1.2 - Considerando que a excitação interna do neurônio é uma soma ponderada das entradas adicionada a uma polarização,

$$u_i = \sum_{j=1}^n w_{ij} x_j + w_{i0} ,$$

mostre que redes feedforward (mesmo multicamadas) com neurônios com os dois tipos função de ativação, $L(u)$ ou $\tanh(\cdot)$, são equivalentes em capacidade de mapeamento entrada-saída (a menos de uma constante e um fator de escala).



$\tanh$

$$u = t_0 + \sum t_i \tanh w_i^t x$$

$\log$

$$u = t_0 + \sum t_i \log w_i^t x$$

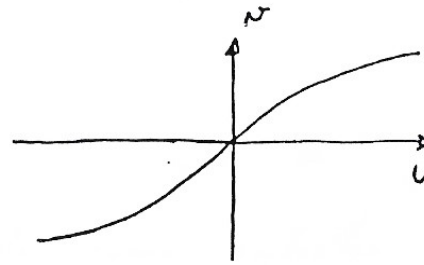
$$= t_0 + \sum \left[\frac{t_i}{2} \tanh w_i^t x + \frac{t_i}{2} \right]$$

$$= a_0 + \sum a_i \tanh w_i^t x$$

2 - Um neurônio tipo log(.) tem função de excitação u e de ativação v dadas por

$$u = \sum_{i=0}^n w_i x_i \quad x_0 = 1$$

$$v = \begin{cases} \ln(1+u) & u \geq 0 \\ -\ln(1-u) & u < 0 \end{cases}$$



2.1 - Desenvolva um processo de aprendizado BP para uma rede com duas camadas, usando neurônios tipo log na camada intermediária e neurônios lineares e/ou tipo tgh na camada de saída, como na rede abaixo. A função objetivo a ser minimizada é o e.m.q. na saída. Apresente um algoritmo que determine de forma explícita os acréscimos a serem aplicados nas sinapses da primeira e da segunda camada, w_{ij} e t_{ij} .

$$F = E e$$

Para cada $p \in P$

$$e = \frac{1}{2} \sum_l \|y_l - \tilde{y}_l\|^2 = \frac{1}{2} \sum_l (y_l - \tilde{y}_l)^2 = \sum_l \varepsilon_l^2$$

$$\frac{\partial \varepsilon_l^2}{\partial w_{ij}} = 2 \varepsilon_l \frac{\partial \varepsilon_l}{\partial w_{ij}} = -2 \varepsilon_l \frac{\partial \tilde{y}_l}{\partial w_{ij}}$$

$$= -2 \varepsilon_l \frac{\partial \tilde{y}_l}{\partial v_i} \frac{\partial v_i}{\partial w_{ij}} = -2 \rho_j \varepsilon_l g_{li}$$

$$\frac{\partial e}{\partial w_{ij}} = -2 \rho_j \sum_l \varepsilon_l g_{li} = -2 \rho_j \delta_i$$

↑ só altera g_{li}

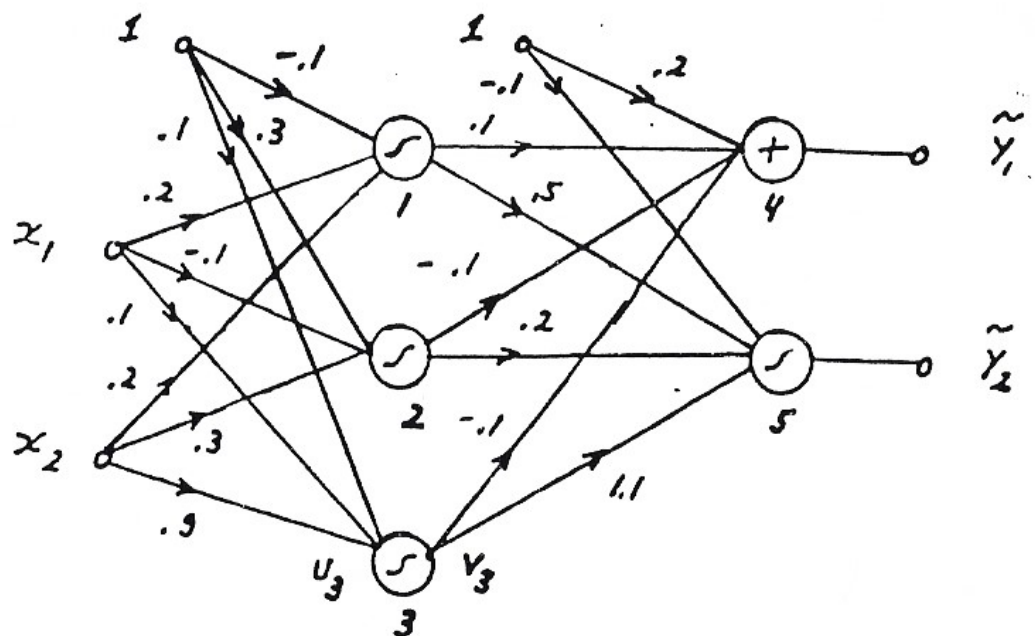
Rede associada

$$p = \pm \ln(1 \pm u) \rightarrow \frac{1}{1 \pm u} = \frac{1}{1 + |u|}$$

$$p = u \rightarrow 1$$

$$p = \tanh(u) \rightarrow 1 - p^2$$

2.2 - Na rede abaixo os neurônios 1, 2, 3 são tipo log, o neurônio 4 é linear ($v = u$) e o 5 é tipo tgh. O treinamento é tipo regra delta sem momento com $\alpha = 0,1$. É apresentado o par entrada-saída $\{\underline{x}; \underline{y}\}$, onde $\underline{x} = [0,1; 0,7]^T$ e $\underline{y} = [0,2; 1,0]^T$. Quais os novos valores das sinapses após o passo de treinamento?



Signal feed forward

$$u_1 = -.1 + (.1)(.2) + (.7)(.2) = .06$$

$$p_1 = + \ln(1+.06) = .058 \quad g_1 = \frac{1}{1+.06} = .94$$

$$u_2 = .3 + (.1)(-.1) + (.7)(.2) = .43$$

$$p_2 = + \ln(1.43) = .358 \quad g_2 = \frac{1}{1.43} = .70$$

$$u_3 = .1 + (.1)(.1) + (.7)(.9) = .74$$

$$p_3 = + \ln(1.74) = .554 \quad g_3 = \frac{1}{1.74} = .575$$

$$u_4 = .2 + (.058)(.1) + (.358)(-.1) + (.554)(-.1) = .115$$

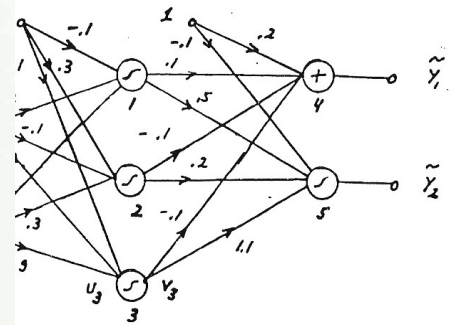
$$p_4 = u_4 = .115 \quad g_4 = 1$$

$$u_5 = -.1 + (.058)(.5) + (.358)(.2) + (.554)(.1) = .61$$

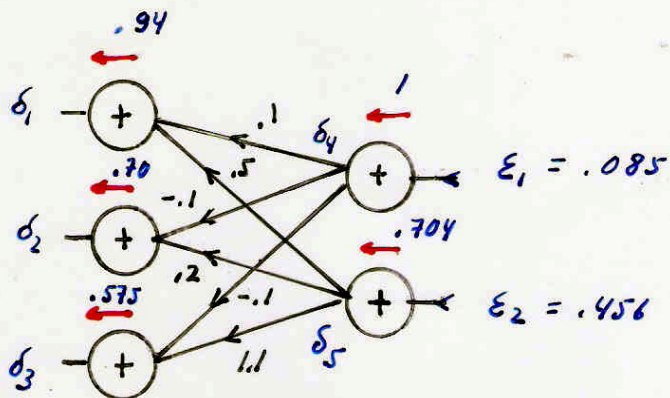
$$p_5 = \tanh .61 = .544 \quad g_5 = 1 - .544^2 = .704$$

$$E_1 = .2 - .115 = .085$$

$$E_2 = 1 - .544 = .456$$



Rede associada



Retropropagação do erro

$$\delta_4 = .085 (1) = .085$$

$$\delta_5 = .456 (.704) = .321$$

$$\delta_1 = [.085 (.1) + (.321) (.5)] .94 = .158$$

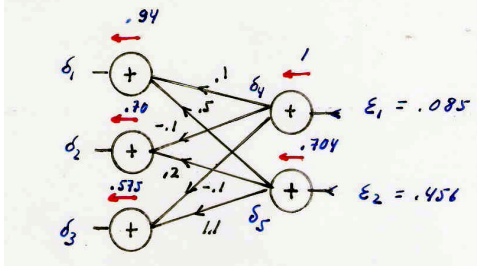
$$\delta_2 = [.085 (-.1) + (.321) (.2)] .70 = -.015$$

$$\delta_3 = [.085 (-.1) + (.321) (.11)] .575 = .157$$

Correção das sinapses

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

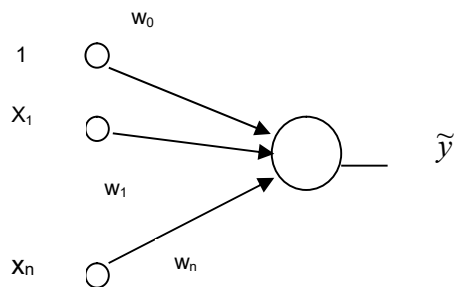
Rede associada



3 - Considere uma rede neural que utiliza um único neurônio linear, como na figura abaixo,

$$u = \sum_{i=0}^n w_i x_i = \underline{w}^t \underline{x}$$

$$e \quad \tilde{y} = u,$$



treinada usando backpropagation regra delta com $\alpha = 0,2$ e sem momento. No início do i -ésimo passo de treinamento o vetor sinapse é $\underline{w}(i-1)$. É então apresentado o par entrada-saída $\{\underline{x}(i), y(i)\}$ e o novo $\underline{w}$ é calculado,

$$\underline{w}(i) = \underline{w}(i-1) + \Delta \underline{w}(i)$$

3a - Deseja-se retroceder este passo. É possível calcular $\underline{w}(i-1)$ conhecendo-se apenas $\underline{w}(i)$ e $\{x(i), y(i)\}$? Se sim, apresente uma fórmula explícita $\underline{w}(i-1) = f\{\underline{w}(i), \underline{x}(i), y(i)\}$.

Signal feedforward

$$\underline{x}_i \longrightarrow \tilde{y}_i = \underline{x}_i^t \underline{w}_{i-1}$$

Erro $\quad \quad \quad \varepsilon = y_i - \tilde{y}_i = y_i - \underline{x}_i^t \underline{w}_{i-1}$

Correção sinapso

$$\Delta w_k = 2\alpha x_k \varepsilon$$

$$\Delta \underline{w} = 2\alpha \underline{x}_i \varepsilon$$

$$= 2\alpha \underline{x}_i [y_i - \underline{x}_i^t \underline{w}_{i-1}]$$

Novo valor da sinapse

$$\underline{w}_i = \underline{w}_{i-1} + \Delta \underline{w}$$

$$= \underline{w}_{i-1} + 2\alpha \underline{x}_i [y_i - \underline{x}_i^t \underline{w}_{i-1}]$$

$$\underline{w}_{i-1} - 2\alpha \underline{x}_i \underline{x}_i^t \underline{w}_{i-1} = \underline{w}_i - 2\alpha y_i \underline{x}_i$$

$$[\underline{I} - 2\alpha \underline{x}_i \underline{x}_i^t] \underline{w}_{i-1} = \underline{w}_i - 2\alpha y_i \underline{x}_i$$

$\therefore$

$$\underline{w}_{i-1} = [\underline{I} - 2\alpha \underline{x}_i \underline{x}_i^t]^{-1} [\underline{w}_i - 2\alpha y_i \underline{x}_i]$$

3b - Repita agora para o caso em que o neurônio é do tipo $\tilde{y} = \tanh(u)$, apresentando a equação a ser resolvida para determinar $\underline{w}(i-1)$. Obs: use formulação matricial.

Caso $\tilde{y} = \tanh u$

$$\tilde{y}_i = \tanh \underline{x}_i^t \underline{w}_{i-1}$$

$$\varepsilon = y_i - \tanh \underline{x}_i^t \underline{w}_{i-1}$$

$$\Delta \underline{w} = 2 \alpha \underline{x}_i \varepsilon (1 - \tilde{y}_i^2)$$

$$\underline{w}_i = \underline{w}_{i-1} + \Delta \underline{w}$$

$$= \underline{w}_{i-1} + 2 \alpha \underline{x}_i \left[y_i - \tanh \underline{x}_i^t \underline{w}_{i-1} \right]$$

$$\left[1 - \tanh^2 \underline{x}_i^t \underline{w}_{i-1} \right]$$

eq. transcendente em $\underline{w}_{i-1}$

Soluções:

1 - Linearização (Taylor)

$$\begin{array}{ccccc} \underline{x}^t \underline{w}_{i-1} & = & \underline{x}^t \underline{w}_i & + & \underline{x}^t (-\Delta \underline{w}) \\ \uparrow & & \uparrow & & \uparrow \\ v + \Delta v & & v & & \Delta v \end{array}$$

$$\tanh(v + \Delta v) = \tanh v + (1 - \tanh^2 v) \Delta v$$

2 - Otimização (gradiente descendente)