CPE 721 2012 1ª série - correção

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.

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

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

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

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

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

+ + × / exp   + ×

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

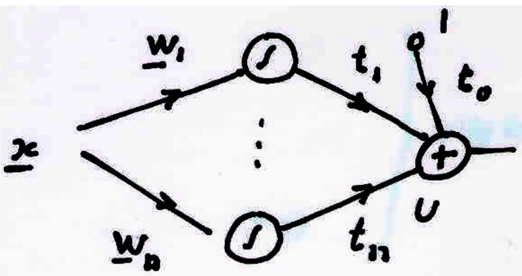
+ × / exp   × × +

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 tgh(.), são equivalentes em capacidade de mapeamento entrada-saída (a menos de uma constante e um fator de escala).



tgh

$$u = t_0 + \sum t_i \, \mathrm{tgh}\, \underline{w}_i^t \underline{x}$$

lg

$$u = t_0 + \sum t_i \, \mathrm{lg}\, \underline{w}_i^t \underline{x}$$

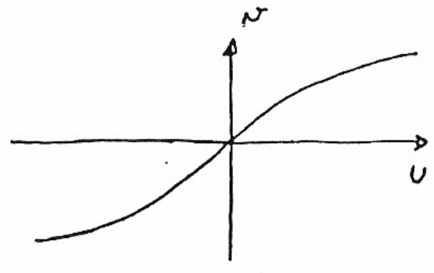
$$= t_0 + \sum \left[\frac{t_i}{2} \, \mathrm{tgh}\, \underline{w}_i^t \underline{x} + \frac{t_i}{2} \right]$$

$$= a_0 + \sum a_i \, \mathrm{tgh}\, \underline{w}_i^t \underline{x}$$

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

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

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



2.1 - Desenvolva um processo de aprendizado BP para uma rede com duas camadas, usando neuronios tipo log na camada intermediária e neuronios 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 algorítmo que determine de forma explícita os acréscimos a serem aplicados nas sinapses da primeira e da segunda camada, w_{ij} e t_{li}.

$$F = E\, e$$

Para cada par

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

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

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

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

só altera g_{li}

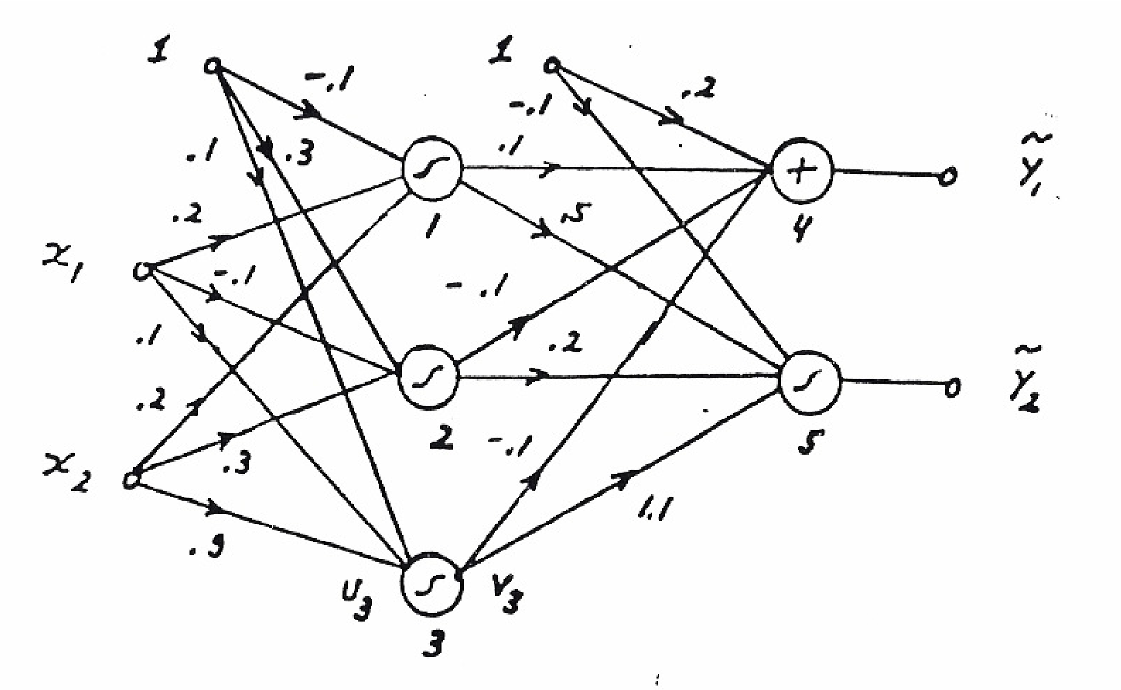
Rede associada

$$\mu = \pm \ln(1 \pm v) \longrightarrow \frac{1}{1 \pm v} = \frac{1}{1+|v|}$$

$$\mu = v \longrightarrow 1$$

$$\mu = tgh(v) \longrightarrow 1 - \mu^2$$

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 {x ; y}, onde x = $[0{,}1 \; ; \; 0{,}7]^t$ e y = $[0{,}2 \; ; \; 1{,}0]^t$. Quais os novos valores das sinapses após o passo de treinamento ?



Signal feedforward

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

$v_1 = + \ln(1+.06) = .058 \qquad g_1 = \frac{1}{1+.06} = .94$

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

$v_2 = + \ln(1.43) = .358 \qquad g_2 = \frac{1}{1.43} = .70$

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

$v_3 = + \ln(1.74) = .554 \qquad g_3 = \frac{1}{1.74} = .575$

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

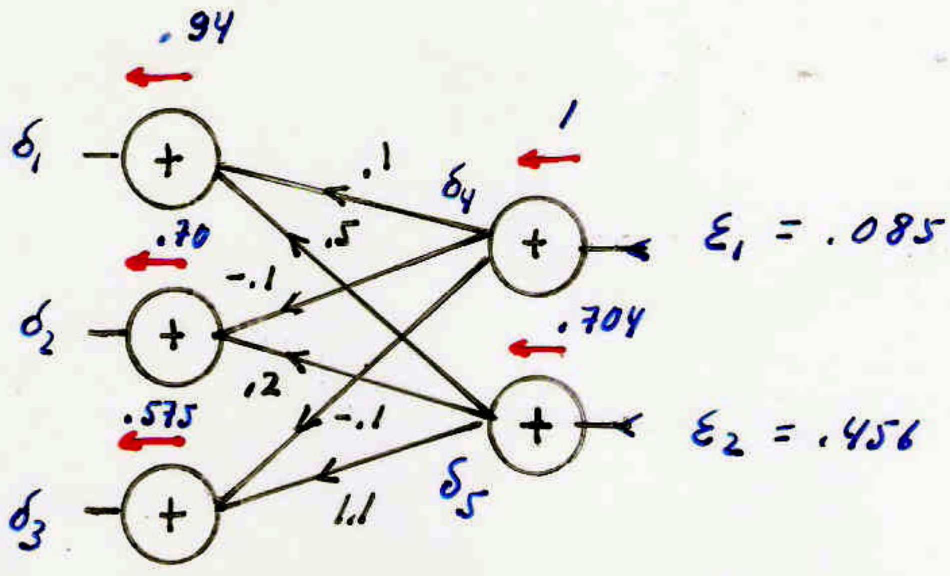
$v_4 = u_4 = .115 \qquad g_4 = 1$

$u_5 = -.1 + (.058)(.5) + (.358)(.2) + (.554)(1.1) = .61$

$v_5 = \text{tgh } .61 = .544 \qquad g_5 = 1 - .544^2 = .704$

$\varepsilon_1 = .2 - .115 = .085 \qquad \varepsilon_2 = 1 - .544 = .456$

Rede associada



Retropropagação do erro

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

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

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

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

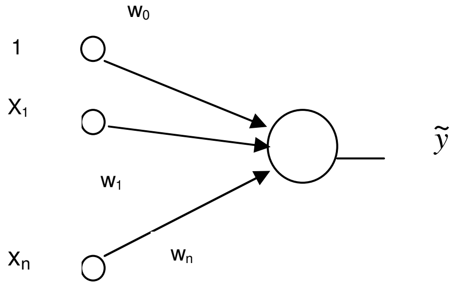
$$\delta_3 = [.085(-.1) + (.321)(1.1)]\,.575 = .157$$

Correção das sinapses

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

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 $\tilde{y} = u$,

treinada usando backpropagation regra delta com α = 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) + Δ $\underline{w}$(i)

3a - Deseja-se retroceder este passo. É possível calcular $\underline{w}$(i-1) conhecendo-se **apenas** 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 $\quad \varepsilon = Y_i - \tilde{Y}_i = Y_i - \underline{x}_i^t \, \underline{w}_{i-1}$

Correção sinapse

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

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

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

Novo valor da sinapse

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

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

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

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

$\therefore$

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

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

Caso $\tilde{y} = tgh\, u$

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

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

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

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

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

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

Soluções:

1- Linearização (Taylor)

$$\underbrace{\underline{x}^t\,\underline{w}_{i-1}}_{v+\Delta v} = \underbrace{\underline{x}^t\,\underline{w}_i}_{v} + \underbrace{\underline{x}^t(-\Delta\underline{w})}_{\Delta v}$$

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

2 - Otimização (gradiente descendente)

4 - Considere uma rede feedforward multicamadas com neurônios do tipo v = tgh u. Estabeleça um algoritmo de treinamento que minimize o erro de saída F_0 abaixo:

$$F_0 = \mathrm{E}\{ \sum_{i=1}^{m} [\ 3\left(\frac{y_i - \tilde{y}_i}{y_i}\right)^2 + 5(y_i - \tilde{y}_i)^2\ ]\ \}$$

Apresente um algorítmo explícito para os acréscimos nas sinapses do tipo:

$$\Delta w_{ij} = f[\alpha, \varepsilon_l, g_{li}] \quad onde\ \ \varepsilon_l = y_l - \tilde{y}_l \quad e \quad g_{li} = \frac{\partial \tilde{y}_l}{\partial u_i}$$

Observe que na dedução do método backpropagation usamos a regra de cadeia $\frac{\partial F}{\partial w_{ij}} = \sum_k \frac{\partial F}{\partial \tilde{y}_k}\frac{\partial \tilde{y}_k}{\partial w_{ij}}$ onde os primeiros termos, $\frac{\partial F}{\partial \tilde{y}_k}$, dependem apenas da função objetivo a ser minimizada, e os segundos termos, $\frac{\partial \tilde{y}_k}{\partial w_{ij}}$, dependem apenas da rede a ser utilizada.

$$F = E \sum_l e_l$$

$$\frac{\partial}{\partial w_{ij}} \sum_l e_l = \sum_l \frac{\partial}{\partial w_{ij}} \left\{ \frac{3}{Y_l^2} \varepsilon_l^2 + 5\, \varepsilon_l^2 \right\}$$

$$\varepsilon_l = Y_l - \tilde{Y}_l$$

$$= \sum_l \frac{\partial e_l}{\partial \varepsilon_l} \frac{\partial \varepsilon_l}{\partial \tilde{Y}_l} \frac{\partial \tilde{Y}_l}{\partial w_{ij}}$$

$$= \sum_l \left\{ \frac{6}{Y_l^2} \varepsilon_l + 10\, \varepsilon_l \right\} (-1) \frac{\partial \tilde{Y}_l}{\partial w_{ij}}$$

$$= -2 \sum_l \left\{ \frac{3}{Y_l^2} + 5 \right\} \varepsilon_l \underbrace{\frac{\partial \tilde{Y}}{\partial v_i}}_{g_{li}} \underbrace{\frac{\partial v_i}{\partial w_{ij}}}_{\mu_j}$$

$$2 \mu_j \sum_l \left\{ \frac{3}{Y_l^2} + 5 \right\} \varepsilon_l\, g_{li}$$

$$- \alpha \frac{\partial e}{\partial w_{ij}} =$$

$$= 2 \alpha \mu_j \underbrace{\sum_l \underbrace{\left\{ \frac{3}{Y_l^2} + 5 \right\}}_{\varepsilon'_l} \varepsilon_l\, g_{li}}_{\delta_i}$$