CPE 721 – 2016 - RNs Feedforward
1ª Série de Exercícios - Treinamento BP

Obs: O objetivo da série de exercícios é a fixação do aprendizado. A série pode ser feita em grupo, mas é importante que cada um tente achar as soluções individualmente antes do trabalho em grupo.

1 – A função logística L(u) também é utilizada como função de ativação em redes neurais feedforward. 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 em função de u e de L(u) e tgh (u), respectivamente.

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

1.1 - Mostre que o custo computacional de treinar ou operar as duas redes é similar, comparando o número e tipo de operações necessárias para calculá-las.

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

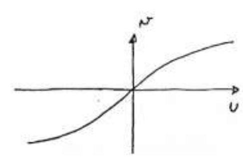
$$u_i = \sum_{j=1}^{n} w_{ij} x_j + w_{i0} \,,$$

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). Sugestão: mostre que as séries que representam as saídas são similares.

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

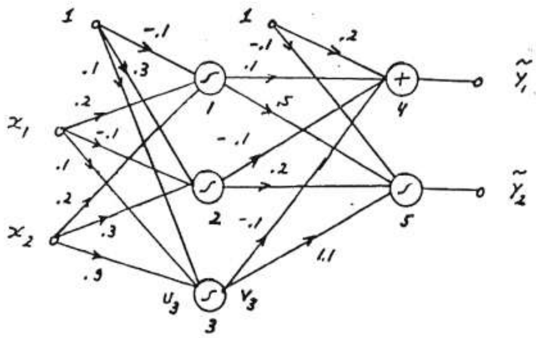
$$\mathrm{u} = \sum_{0}^{\mathrm{n}} \mathrm{w_i x_i} \qquad x_0 = 1$$

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



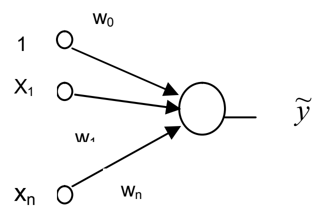
2.1 - Desenvolva um processo de aprendizado BP para uma rede com duas camadas, usando neurônios tipo log na camada intermediária e neurônios lineares e/ou tipo tgh na camada de saída, como na rede abaixo. A função objetivo a ser minimizada é o e.m.q. na saída. Apresente um algorítmo que determine de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

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 ?



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

$$u = \sum_{i=0}^{n} w_i x_i = \underline{w}^t \underline{x} \quad \text{e} \quad \tilde{y} = u \,,$$



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

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

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

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 <u>w</u>(i-1).

Sugestão: use formulação matricial.

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:

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

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

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

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.