file: c:\math\mcad80\rwn_mcad\backprop_ex2.mcd   RWN   02/20/02
Example of backpropagation for 2-layer network

Assume a two layer double input triple output network with four tansig neurons in the first layer and three tansig neurons in the output layer to approximate [3p2cos(p1) p1*p2^3 p2/(p1+p2)]'; train on [p1=2.5 p2=1.1]'=a0

$$a0 := \begin{bmatrix} 2.5 \\ 1.1 \end{bmatrix} \qquad p := a0$$

Choose initial weights and biases;

For first layer

$$W1 := \begin{bmatrix} 0.1 & -0.1 \\ -0.2 & 0.2 \\ -0.4 & 0.5 \\ 0.5 & -0.4 \end{bmatrix} \quad b1 := \begin{bmatrix} 1.2 \\ -0.6 \\ 0.8 \\ -1.2 \end{bmatrix}$$

For0.1 second layer

$$W2 := \begin{bmatrix} 0.1 & -0.1 & 0.2 & -0.2 \\ -0.2 & 2.2 & 4.1 & -0.1 \\ -0.8 & 0.3 & -0.6 & 1.5 \end{bmatrix} \qquad b2 := \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

The training ouput is

$$t := \begin{bmatrix} 3 \cdot p_2 \cdot \cos(p_1) \\ p_1 \cdot (p_2)^3 \\ \dfrac{p_2}{(p_1 + p_2)} \end{bmatrix} \qquad t = \begin{bmatrix} -2.644 \\ 3.328 \\ 0.306 \end{bmatrix}$$

The network functions are

$$n1 := W1 \cdot a0 + b1 \qquad n1 = \begin{bmatrix} 1.34 \\ -0.88 \\ 0.35 \\ -0.39 \end{bmatrix} \qquad n1_1 = 1.34 \qquad n1_2 = -0.88$$

$$a1 := \tanh(n1)$$

$$a1 = \begin{bmatrix} 0.872 \\ -0.706 \\ 0.336 \\ -0.371 \end{bmatrix} \qquad a1_1 = 0.872 \qquad a1_2 = -0.706$$

$$n2 := W2 \cdot a1 + b2$$

$$n2 = \begin{bmatrix} 1.299 \\ 1.688 \\ -0.668 \end{bmatrix}$$

$$a2 := n2$$

We wish this to become t, that is, n2==>t is desired by training

Output difference, e, and error E $e := t - a2$ $e = \begin{bmatrix} -3.943 \\ 1.64 \\ 0.974 \end{bmatrix}$

$E := e^T \cdot e$ $E = (19.185)$

The function derivatives are found from y=tanh(x)=(2/(1+e^(-2x))-1) as dy/dx=(1-y)(1+y), while for the second layer it is the identity. Thus

$da1(y) := (1-y)\,(1+y)$ $da1(a1_1) = 0.24$

$da1(a1_2) = 0.501$

$$df1 := \begin{bmatrix} da1(a1_1) & 0 & 0 & 0 \\ 0 & da1(a1_2) & 0 & 0 \\ 0 & 0 & da1(a1_3) & 0 \\ 0 & 0 & 0 & da1(a1_4) \end{bmatrix} \qquad df1 = \begin{bmatrix} 0.24 & 0 & 0 & 0 \\ 0 & 0.501 & 0 & 0 \\ 0 & 0 & 0.887 & 0 \\ 0 & 0 & 0 & 0.862 \end{bmatrix}$$

$$df2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Start the backpropagation

$s2 := -1 \cdot df2 \cdot (t - a2)$ $s2 = \begin{bmatrix} 3.943 \\ -1.64 \\ -0.974 \end{bmatrix}$

$s1 := df1 \cdot W2^T \cdot s2$ $s1 = \begin{bmatrix} 0.361 \\ -2.151 \\ -4.745 \\ -1.798 \end{bmatrix}$

Weight update; using a learning rate α=0.3 overshoots so choose smaller

$\alpha := 0.01$

$W2new := W2 - \alpha \cdot s2 \cdot a1^T$ $W2new = \begin{bmatrix} 0.066 & -0.072 & 0.187 & -0.185 \\ -0.186 & 2.188 & 4.106 & -0.106 \\ -0.792 & 0.293 & -0.597 & 1.496 \end{bmatrix}$

$b2new := b2 - \alpha \cdot s2$ $b2new = \begin{bmatrix} 0.961 \\ 2.016 \\ 1.01 \end{bmatrix}$

$W1new := W1 - \alpha \cdot s1 \cdot a0^T$ $W1new = \begin{bmatrix} 0.091 & -0.104 \\ -0.146 & 0.224 \\ -0.281 & 0.552 \\ 0.545 & -0.38 \end{bmatrix}$

$b1new := b1 - \alpha \cdot s1$

$$b1new = \begin{bmatrix} 1.196 \\ -0.578 \\ 0.847 \\ -1.182 \end{bmatrix}$$

$n1new := W1new \cdot a0 + b1$

$a1new := \tanh(n1new)$

$$a1new = \begin{bmatrix} 0.865 \\ -0.617 \\ 0.607 \\ -0.25 \end{bmatrix}$$

$n2new := W2new \cdot a1new + b2new$

$a2new := n2new$ $\quad$ $enew := t - a2new$ $\quad$ $Enew := enew^T \cdot enew$

$$n1new = \begin{bmatrix} 1.313 \\ -0.72 \\ 0.704 \\ -0.256 \end{bmatrix} \quad a1new = \begin{bmatrix} 0.865 \\ -0.617 \\ 0.607 \\ -0.25 \end{bmatrix}$$

$$n2new = \begin{bmatrix} 1.222 \\ 3.024 \\ -0.593 \end{bmatrix} \quad a2new = \begin{bmatrix} 1.222 \\ 3.024 \\ -0.593 \end{bmatrix}$$

$$enew = \begin{bmatrix} -3.865 \\ 0.303 \\ 0.898 \end{bmatrix}$$

compare to previous

$$e = \begin{bmatrix} -3.943 \\ 1.64 \\ 0.974 \end{bmatrix}$$

$Enew = (15.84)$

$E = (19.185)$