Let us suppose that for the input $\begin{bmatrix}1\\0\end{bmatrix}$ we expect the network to output $\begin{bmatrix}1\\0\end{bmatrix}$.

Initial weights:

$$W = \begin{bmatrix} 0.1 & -0.2 & 0.3 \\ -0.4 & 0.5 & -0.6 \end{bmatrix}, \qquad V = \begin{bmatrix} 0.15 & -0.25 & 0.35 \\ -0.45 & 0.55 & -0.65 \end{bmatrix}$$

Input (the last coordinate is always -1 and it encodes the bias):

$$X^{(1)} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

The first layer (W) gives

$$net_1 = W \cdot X^{(1)} = \begin{bmatrix} -0.2 \\ 0.2 \end{bmatrix}$$

after applying the function $f(x) = (1+e^{-x})^{-1}$ element-wise

$$Y^{(1)} = \begin{bmatrix} f(-0.2) \\ f(0.2) \end{bmatrix} = \begin{bmatrix} 0.450166 \\ 0.549834 \end{bmatrix},$$

and after appending -1 we obtain the input for the second layer:

$$X^{(2)} = \begin{bmatrix} 0.450166 \\ 0.549834 \\ -1 \end{bmatrix}$$

The second layer (V) gives

$$net_2 = V \cdot X^{(2)} = \begin{bmatrix} -0.4199336 \\ 0.749834 \end{bmatrix}$$

after applying the function $f(x) = (1+e^{-x})^{-1}$ element-wise we obtain the output of the network:

$$Y^{(2)} = \begin{bmatrix} f(-0.4199336) \\ f(0.749834) \end{bmatrix} = \begin{bmatrix} 0.39653264 \\ 0.67914253 \end{bmatrix}.$$

We expected to obtain $\begin{bmatrix}1\\0\end{bmatrix}$, so the error is $b^{(2)} = \begin{bmatrix}1\\0\end{bmatrix} - Y^{(2)} = \begin{bmatrix} 0.60346736 \\ -0.67914253 \end{bmatrix}$ From there we obtain *the delta signal* by multiplying elements by the value of f' in corresponding points net_2; it holds $f'(x) = f(x)(1-f(x))$, therefore

$$\delta^{(2)} = \begin{bmatrix} 0.60346736 \cdot f'(-0.4199336) \\ -0.67914253 \cdot f'(0.749834) \end{bmatrix} = \begin{bmatrix} 0.14440642 \\ -0.147990559 \end{bmatrix}$$

We adjust the weights V, taking the *learning rate* equal to $c = 0.1$,

$$\tilde{V} = V + c\,\delta^{(2)} \cdot (X^{(2)})^T = V + c \begin{bmatrix} 0.06500686 & 0.07939956 & -0.1444 \\ -0.06662 & -0.08137 & 0.14799 \end{bmatrix} = \begin{bmatrix} 0.156500686 & -0.24206 & 0.33556 \\ -0.456662 & 0.541863 & -0.6352 \end{bmatrix}$$

We calculate the error for the first layer as follows

$$b^{(1)} = V^T \cdot \delta^{(2)} = \begin{bmatrix} 0.08825672 \\ -0.11749641 \\ 0.14673611 \end{bmatrix}.$$

The last coordinate (0.14673611) is redundant (it corresponds to the constant input -1, which encodes the bias) – we omit it, and we multiply the remaining ones by the values of f' at the points net_1, to obtain the delta signal for the first layer,

$$\delta^{(1)} = \begin{bmatrix} 0.08825672 \cdot f'(-0.2) \\ -0.11749641 \cdot f'(0.2) \end{bmatrix} = \begin{bmatrix} 0.021845 \\ -0.0290823 \end{bmatrix}.$$

We adjust the weights W, taking again the *learning rate* equal to $c = 0.1$,

$$\tilde{W} = W + c\,\delta^{(1)} \cdot (X^{(1)})^T = \begin{bmatrix} 0.1021845 & -0.2 & 0.2978155 \\ -0.40290823 & 0.5 & -0.59709177 \end{bmatrix}.$$

We obtain a network with modified weights $\tilde{W}$ i $\tilde{V}$, and repeat...

Notes:

(1) The above equalities are not exact, some rounding errors are possible. For the function $f(x) = (1+e^{-x})^{-1}$ it holds $f'(x) = f(x)(1-f(x))$ (as one may easily verify). If we took $\tilde{f}(x) = f(\lambda x)$, then $\tilde{f}'(x) = \lambda \tilde{f}(x)(1-\tilde{f}(x))$. Nevertheless the factor λ may be omitted in the formulae by incorporating it into the learning rate c.

prepared by B.D.