EEE 443/543
Mini Project 1, Part 2
Ömer Morgül

The following function $h(x)$ is to be approximated by using a multilayer neural network.

$$h(x) = \sin(x^2) \qquad -4 \leq x \leq 4$$

i : Use a multilayer structure which has two inputs (one representing x, and the other representing the treshold input, i.e. -1), one output layer, and one hidden layer, as shown below. Number of neurons in the hidden layer should be selected as $n = 10,\ 15,\ 20$. As activation function, choose bipolar sigmoidal function ($f(v) = (1 - e^{-v})/(1 + e^{-v})$). Choose the training set $x(i)$ as

$$x(i) = -4 + \frac{(i-1)}{N} \quad , \quad i = 1, 2, \ldots, 8N + 1$$

where $N = 10,\ 15,\ 20$. Find the desired outputs as

$$d(i) = h(x(i))$$

.

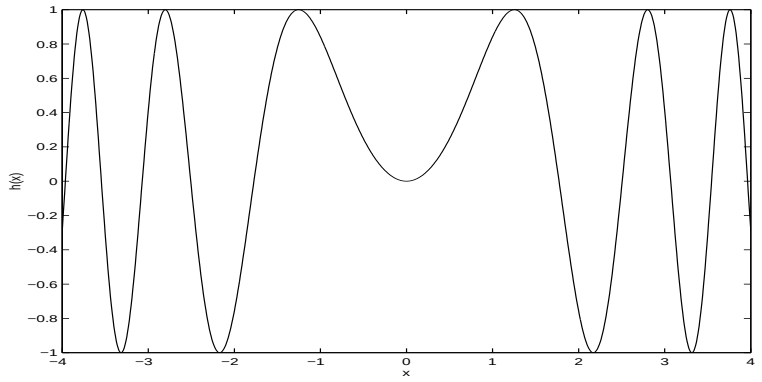
Choose the learning coefficient as $\eta = 0.1,\ 0.5,\ 0.9$ and by using the back propagation algorithm, find the weights so that the average error E_{ave} over the training set satisfies $E_{ave} \leq 10^{-3}$. Record the average error achieved after the convergence as well as the number of epochs that took place until convergence.(Write your own back propagation code in an acceptable program, for such programs consult with Arda, or look at the course Web page, but do not use directly a neural network software. In case you cannot achieve this error, try to make it as low as possible. Also note that with the prescribed selection of $n,\ N,\ \eta$, you should run the algorithm at least $3 \times 3 \times 3 = 27$ times.)

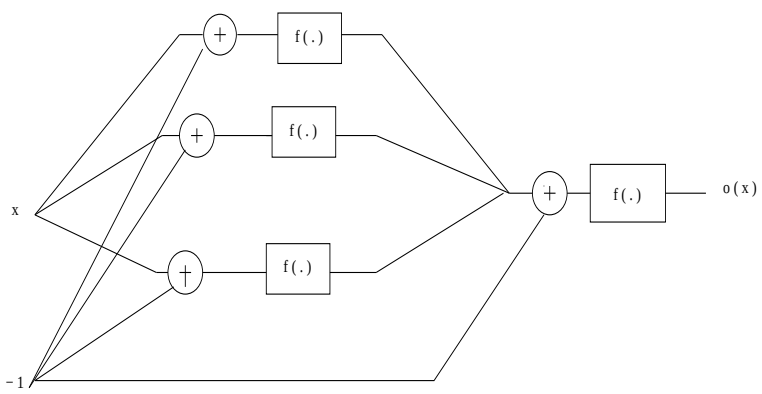
ii : After the training of the neural network, now choose the following test points :

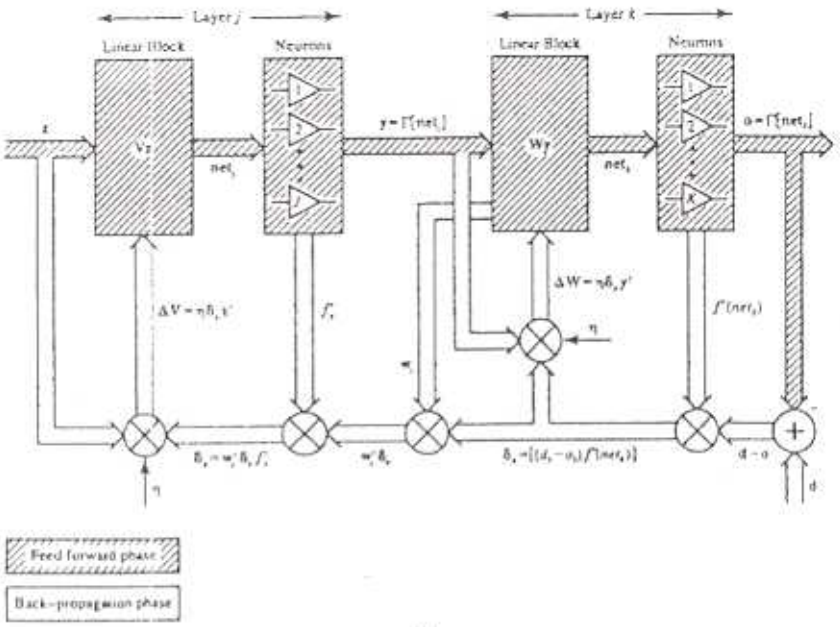
$$x_{test}(i) = \frac{x(i) + x(i+1)}{2} \quad , \quad i = 1, 2, \ldots, 8N$$

where $x(i)$ are found above, and find the average error the neural network you obtained above produces over the test set.

iii: Write your results in a report form, indicating the work you have done.

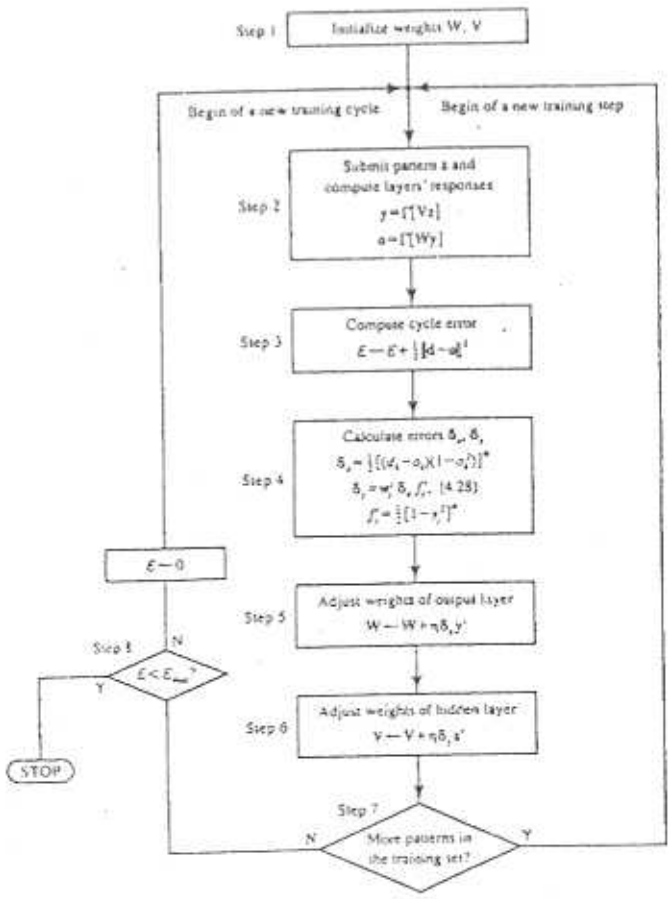






(b)

Figure 4.8b Error back-propagation training (EBPT algorithm) (continued): (b) block diagram illustrating forward and backward signal flow.



* If $f(net)$ given by (2.4a) is used in Step 2, then in Step 4 use

$$\delta_k = \tfrac{1}{2}(d_k - o_k)(1 - o_k^2), \quad f_j' = \tfrac{1}{2}(1 - y_j^2)$$

(a)

Figure 4.8a Error back-propagation training (EBPT algorithm): (a) algorithm flowchart.