Homework 3

Total possible score: 60 points (60 points = 100%)

Problem 1. (10 points)

A fully connected feed forward artificial neural network has 3 input PEs, two hidden layers with 4 PEs in the first hidden layer, 2 PEs in the second hidden layer, and a single output PE. Draw an architectural scheme for this network and annotate it fully, using the notation we introduced in class. That is, label every structural element: PEs, weights, with appropriate indexing with respect to layers. (As you remember, fully connected feed forward means that there are no feed back connections in the network and each PE in one layer is connected to all PEs in the next layer.)

Problem 2. (20 points total)

2/a) (15 points) Show that if a fully connected multilayer feed forward network has a linear transfer function for all PEs, then the mapping

$$F : \mathbf{x} \longrightarrow \mathbf{y} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ denote the input vector and the output vector at the top layer, respectively, can be written as

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2}$$

where $\mathbf{W}$ is composed appropriately from all weights in the network.

2/b) (5 points) (2/a) proves an important property about the structural complexity versus transfer function of this ANN. What is it?

Problem 3. (30 points total)

Perform the experiments described below, using the error correction Associative Memory code errcorr.m I provided at http://terra.ece.rice.edu/ANNclass502/. In the same directory you find a data file EHTOM.ascii, with five characters E, H, T, O, M, in 12 x 12 pixel MATLAB matrix format.

a) Store (learn) the five characters (E, H, T, O, M). Consider changing the zeros to -1 in the encoding of the characters. Test the recall accuracy of the network on the training data. The recall accuracy can be measured in a number of different ways. With visual data (such as characters) one always wants to show the images of the input and desired output characters, and the actual recalled characters, side by side. In addition, one might choose to show the difference images (be sure to provide a scale for the difference values). A numerical summary - in addition to the images - can be the % of the pixels that were recalled incorrectly, or the average absolute difference (over all pixels of a character).

b) Corrupt the data by flipping a pre-specified percentage of the pixels randomly, in each pattern. (E.g., change -1 to 1 or 1 to -1 in 29 randomly chosen pixels, which is about 20% of one character image.) Test the recall success of the memory created in part a) (i.e., do not train a new memory!), on the corrupted data. Use several levels of noise (corruption), for example, 10%, 25%, 50%, for this experiment, or even more.

c) Create a new memory using corrupted data (one of the corrupted sets of patterns) for training input and the original clean data for desired output, and test the recall success as in parts a) and b).

For each case of a) - c), report your results as follows:

- List the pertinent network and learning parameters (in parts a) and c)). State what error measure you use to assess the quality of the recall.

- Show the images of the input and desired output training patterns that you used for creating the memory; the input(s) for recall(s); the recalled patterns (characters); the error image and a numerical error measure of the recall success. Use uniform scale for images of the same kind (input and recalled images; error images for different cases) and show the scale.

An example of a good presentation and organization of results is in HW03-P3-SamplePresentation.docx, in Resources / Sample Solutions in Owl Space, along with suggested MATLAB functions and code snippet to create summary graphics, scales, etc.