Homework 3, posted 1/24/2008

Due 1/31/2008, in hardcopy form (either neat handwriting or typeset), bring to class or drop in the pocket at my door.
Total possible score: 60 points (60 points = 100%)

Problem 1. (10 points)

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

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 (final) 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 complexity versus transfer function of an ANN. What is it?

Problem 3. (30 points total)

Perform the experiments described below, using the error correction Associative Memory code that I gave you a hardcopy of in class. I recommend using MATLAB. I have deposited an ascii file with the data (five characters E, H, T, O, M, in 12 x 12 pixel MATLAB format), in /home/erzsebet/ANNclass502/EHTOM.ascii for your perusal.

a) Store the five characters (E, H, T, O, M, in 12 x 12 pixel format). Consider to change the zeros to -1 in the encoding of the characters. Test the recall accuracy of the network on the training data.

b) Corrupt the data by flipping a prespecified percent of the pixels randomly. Test the previously created memory's recall success on the corrupted data. Use several levels of noise (for example, 10%, 25%, 50% or even more).

c) Create a new memory using the corrupted data for input and the original clean data for output, and test the recall success.