3pts: show all of your work to receive full credit

Homework #3 (due 10/4) by 1:35pm

1. (1.5pts) Build a heteroassociative neural net using the Hebb rule to set the weights via outer products which will read in the small set (3) of training(S):target(T) paired images that I provided to you ("SandT_patterns_HW3_P1.mat") in 9x7x3 matrices on Canvas. Use bipolar images and activation functions for the network.
    a. Train the network on these images. After you train the network, fix the weights, and run the net against each training image and see what target is generated. Print out the training – target pair you obtain as a result.
    b. Next add two more input:output pairs of your own design (same dimensions as the training:target pairs I provided) and train the network from scratch on all 5 images. After it is trained, run the network with each input image and print out the input:output pairs you obtain.
    c. Pick any 3 of the 5 input images and make 12 randomly selected pixels in each image be either 'missing' or 'incorrect' to create noisy images. Run the noisy images through the net and print out the input:output pairs.
2. (1.5 pt) Build an autoassociative NN using the Hebb rule to set the weights via outer products that will read in the set of S (training) images that I provided to you in 9x7x10 ("S_patterns_HW3_P2.mat") matrices. Use bipolar images and activation functions for the network.
    a. How many images can be successfully stored by the NN? Does the limit depend upon the images you pick to store and why?
    b. How much noise can your NN handle in the input images for which it still produces correct results? Does the network's level of noise tolerance depend upon the number of vectors stored?
    c. Pick a situation from the above where the network doesn't exhibit perfect recall. Permit the network to iterate when it doesn't produce a correct result. Does the performance improve?

**You must submit your results and working code to receive credit. If the code has errors or can't be run to verify the functionality, you will lose a significant majority of credit for the problem.