VS265 - HOMEWORK 4 - HOPFIELD NETWORKS

DUE THURSDAY, OCTOBER 9TH

(1) For this problem you will need to download the scripts hopnet.m, genpat.m, and the data patterns.mat from the class webpage
   (a) Load the patterns in patterns.mat. Each is a 100x1 vector that you will need to reshape into a 10x10 array in order to display as an image. Store these three patterns in the weight matrix T using the outer product rule.
   (b) Using hopnet.m as a starting point, simulate the Hopfield network dynamics. First, try initializing the network to each of the stored patterns above to verify they are stable basins of attraction. Then try initializing to a corrupted version of each pattern by flipping a certain fraction of its bits (you can use the script corrupt.m to do this). What is the maximum Hamming distance away (number of flipped bits) you can start from which still leads to the same basin of attraction?
   (c) You can generate new patterns of your own using the script genpat.m (type help genpat for instructions). Try storing these additional patterns in the network. How many additional patterns can you store while keeping all basins stable? What is the effect on the Hamming radius around each basin as you store more patterns?

(2) One of the problems with storing visual patterns in the network is that they are typically correlated with one another. Try storing a number of random patterns in the network to explore its true capacity using the outer product rule. You will no longer be able to use your visual system though to verify the stability of memories since they are now random patterns, so you will need to write a script to automate this process (you should also remove the display components to speed things up).
   (a) How many patterns can you now store in the network while keeping all (desired) basins of attraction stable?
   (b) When you are near the capacity limit, try exploring the energy landscape by initializing to random patterns. What fraction of the basins encountered correspond to stored patterns?