EECS 484 Computational Intelligence, Fall 2011
Problem Set 2: Self-Organizing Maps

Assigned: 9/13/11
Due: 9/20/11

Self-organizing maps (SOM) constitute an alternative way to perform clustering, yielding some additional valuable properties. In addition to enabling data compression through clustering (in this case, with clusters corresponding to grid points) and cluster centers (a prototype--a vector associated with each grid point), the self-organizing map also may recognize useful dimensional reduction and topological connectivity.

The accompanying zip file contains a set of training patterns and test patterns. These patterns correspond to binary 8x8 scenes. The scenes originate from coherent blobs, comprised of 2x2 and 3x3 squares of “on” pixels with the remaining pixels “off”. However, these individual pixels are scrambled between the original receptors and the terminals at the self-ordering map. Although the signals are scrambled in this transmission, all of them are scrambled according to a consistent remapping, e.g. as though respective axons had become tangled in the transmission from inputs to outputs.

A collection of clusters is arranged in a grid, which receives the disordered axon terminals. The dimensions of the grid of clusters may be the same as the input patterns (8x8) or may be smaller, larger, or non-square. The Matlab starter code posted on the web site performs the necessary operations for reading and unpacking the data. The data files are: scrambled_blobs.mat and scrambled_testpats.mat. The scrambled blobs should be used to train the grid of clusters, and the scrambled test patterns should be used to evaluate the success of the grid of clusters in performing decoding. The test patterns are the symbols I, H and X, scrambled per the same mapping as the scrambled training blobs.

The main Matlab file is som_clustering_main.m, which uses functions find_closest_cluster(), alphafnc(), vec2pat() and pat2vec(), and view_all_pattern_responses(), as well as the script eval_test_patterns.m. Some of these must be edited, as noted in the program comments.

The main routine loads the training images and converts these 8x8 images to corresponding 64x1 vectors (to be consistent with SOM training). The main routine also establishes a set of clusters arranged in a grid. E.g., try a cluster grid that is 6x6. Each cluster has an associated feature vector that matches the pattern vector dimensions (64x1). Every cluster feature vector should be initialized to random values and normalized to unit magnitude.

After initializing the cluster vectors, the main routine should cycle through *many* iterations involving choosing a training pattern at random and updating the cluster grid per the SOM training policy. Updating the cluster grid requires first finding the single cluster, designated by its grid coordinates ibest,jbest, that is most similar to the current pattern (as measured by the dot product between the training pattern and the cluster’s feature vector). Each selected training pattern (potentially) influences every cluster in the grid. However, clusters that are geometrically further from the most-similar cluster (as measured with respect to relative grid coordinates) should be influenced less by the training pattern than clusters that are closer to ibest,jbest. The influence of a pattern on a cluster is modulated by the function alphafnc(), which returns the coefficient "alpha", which should be used to scale the influence of a training pattern on a given cluster as a function of distance and time. The training pattern should alter the values of each cluster's feature vector to be more similar to the input feature vector, with the strength of the influence modulated by the parameter alpha. Note that every such modification should conclude with renormalizing the cluster feature vector to unity.

The grid of clusters may be used as follows. For a given stimulus vector (which does not have to be part of the training set), similarity to each cluster feature vector should be computed and this value assigned to the corresponding grid location i,j. The result may be visualized to interpret the entire cluster grid's response to the stimulation vector. For coherent patterns that have been scrambled per the same mapping as the training blobs, the hope is that the cluster response will unscramble the test pattern to recover its original image.

In the example code, after every 1000 training-pattern influences, the cluster grid is evaluated graphically for every training pattern and all three test patterns.

The test patterns, which are stored in file "scrambled_testpats.mat", correspond in the original receptor space to the symbols "X", "H" and "1". They have been scrambled with the same reordering as the training blobs. If your cluster training is successful, you should be able to recognize these symbols when they are used to stimulate the cluster grid. This evaluation is performed by the script "eval_test_patterns.m".

Your assignment is:

1) Edit the starter Matlab code to execute the SOM algorithm (include your code with your solution).
2) Experiment with different training parameters (e.g., number of iterations, radius of influence as a function of time, and value of alpha as a function of radius from the most similar cluster) to train a 6x6 cluster grid (optionally, additional different-sized grids) on the provided data.
3) Show your results with respect to the three test patterns.