CSE5301 Neural Networks CSE5301

Practical/Assignment 3

3.1 Competitive Learning

1. Study the scripts `CmpInit.m, CmpLrn.m` from `http://www.csse.monash.edu.au/courseware/cse5301`

2. Modify the above scripts to increase dimensionality of the input space from two to three, and to implement the Frequency-Sensitive Competitive Learning (FSCL):

   - In the FSCL, for each neuron we create the count, c_i, of the weight updates performed for this neuron, that is, the number of times this neuron won. The weight update is inversely proportional to the count, c_i, that is, the learning gain is proportional to $1/c_i$.
   - Initialize the weights with a sample of m input vectors (as oppose to the random initialization of weights).

3. Implement the Vector Quantization Image Compression System using the Frequency-Sensitive Competitive Learning.

   - Test the VQ image compression system on a sample image available in MATLAB.
   - Discuss briefly the differences between image coding systems based on competitive learning (VQ) and the Generalised Hebbian learning.

3.2 Self-Organizing Feature Maps I

Implement the Kohonen's algorithm for self-organizing neural networks which performs feature mapping from a two-dimensional input space onto:

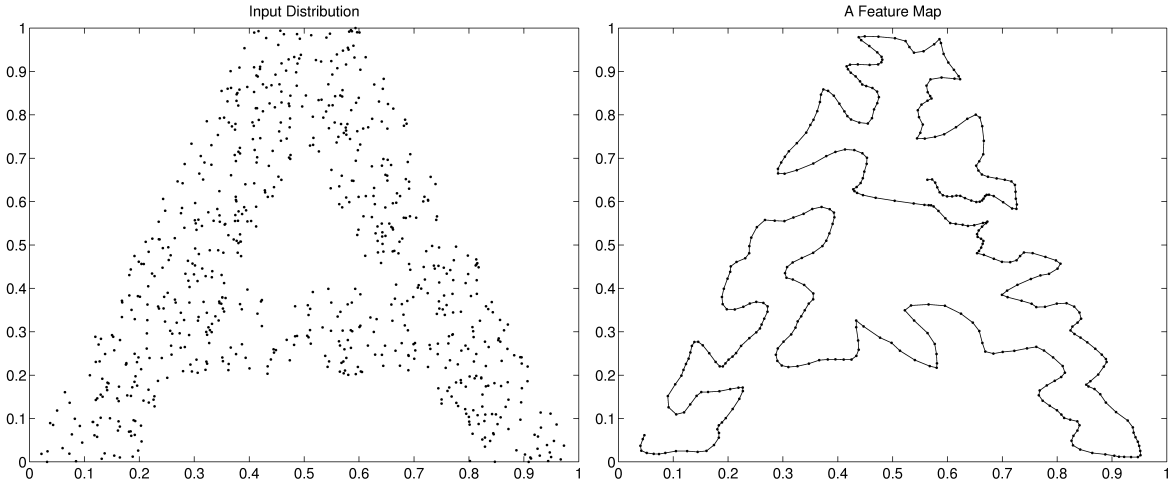
1. a one-dimensional feature space, and
2. a two-dimensional feature space.

The two-dimensional input space should have a shape of one of the following characters:

A, C, E, F, H, J, K, L, M, N, X, Y, Z

Use (studentID)mod13 as the pointer to select your character from the above list.

The input space and the resulting one-dimensional feature map might look as follows:



Refer to the script `sofm.m` for assistance, but

- plot the two-dimensional feature map using the "pcolor" MATLAB function,
- use the position matrix V, as in the lecture notes. (Note that `sofm.m` uses different method).

3.3 Self-Organizing Feature Maps II

Prepare a script to implement a data clustering algorithm using SOFMs, and study its behaviour.

- Generate randomly located 4 pairs of clusters of 2-D points, each cluster with a different variance and containing 8 points. Clusters forming a pair should be relatively close to each other.
- Use a 2×3 and a 3×3 SOFM to cluster the data, that is, to approximate distribution of data

Your submission should include:

- brief comments regarding the demo files,
- Relevant scripts and plots with suitable comments.