Tecnológico de Monterrey

Computational Intelligence

José Carlos Ortiz Bayliss

Programming Assignment 04

01170065 - MIT
Xavier Fernando Cuauhtémoc Sánchez Díaz
xavier.sanchezdz@gmail.com

November 22, 2017

Problem 1

Kohonen

- A Kohonen network was trained and used to cluster the dataset provided.
- Script is attached.

Run details

`W = train_kohonen(data, 3, 10)` yielded

$$W = \begin{bmatrix} 0.79956 & 0.18422 \\ 0.19329 & 0.80481 \\ 0.54407 & 0.50552 \end{bmatrix}$$

Which looks something like the following:

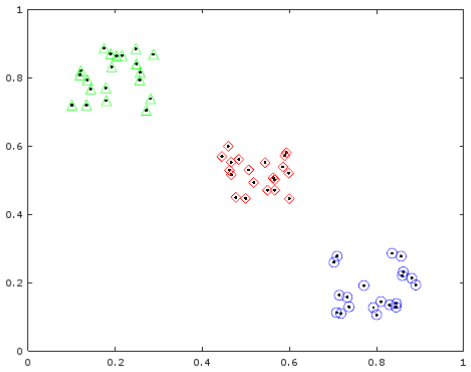

Figure 1: Clustered data using a SOM.

Problem 2

LVQ was quite difficult, as the number of neurons, the learning rate and decay parameters were tweaked until finding something.

Run details

The code `[w,acc] = lvq_loop(data1, 5000)` yielded the following:

$$acc = 1 \quad W = \begin{bmatrix} 0.462316 & 0.877874 \\ 0.164309 & 0.793170 \\ -0.104877 & 0.489462 \\ 0.465819 & -0.076281 \\ 0.330399 & 0.435579 \\ 0.155575 & 0.170246 \end{bmatrix}$$

By using 6 neurons, that is, overfitted to the maximum. However, 3 neurons were enough, with more training iterations.

`[w,acc] = lvq_loop(data1, 5000)`:

$$acc = 1 \quad w = \begin{bmatrix} -0.091944 & 0.733631 \\ 0.707465 & 0.011495 \\ 0.238132 & 0.330761 \end{bmatrix}$$

Problem 3

The same story goes for problem 3, but instead using the `Iris` dataset.

The following parameters were used:

```
>> [w,acc] = lvq_loop(data2, 1000)
lrate = 0.8;
decay = 0.05;
w =

   0.2229961   0.4159564   0.5371445   0.7452070
   0.6727481   0.9419399   0.4069340   0.0988960
   0.8154970   0.5843199   0.5376810   0.4165832
   0.7553312   0.3133215   0.7744005   0.4518696
   0.3642381   0.0749404   0.1702528   0.2977069
   0.6864519   0.5519802   0.7889211   0.6116886
   0.9809330   1.0312931   0.0545014   0.4077190
   0.1253019   0.3323812   0.0046333   0.6783671
   0.2664669   0.5714717   0.1522563   0.9543915
   0.7580878   0.9439515   0.7847167   0.9685561

acc =  1
```

It was also possible using even 3 neurons, given enough iterations.

```
  >> [w,acc] = lvq_loop(data2, 1000)
  w =

     0.78196   0.86169   0.20912   0.24113
     0.66412   0.39710   0.99623   0.84333
     0.87452   0.28492   0.94403   0.58132

  acc =  1
```