**EPL442 Machine Learning**

**Assignment 3**

Nicolas Zachariou 916782

17/11/2020

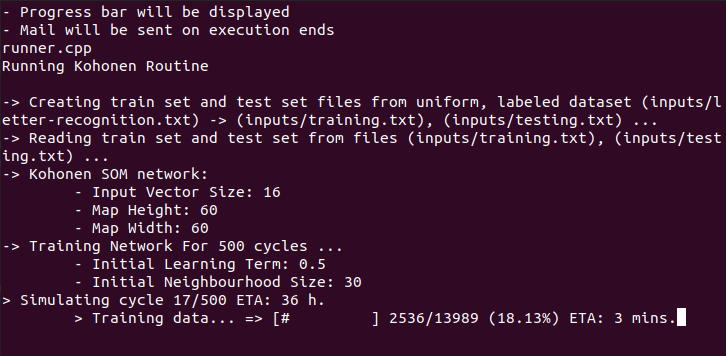
**Compile & run Instructions:**

* To compile: **./compile.sh**
* For execution tips: **./a.out -help**
* Program parameters are read from the “**parameters.txt**” file
* Kohonen Example run: **./a.out kohonen -c -progress**
  + **-c** : will split single **data** set file into specified **input** and **output** files with given **train/test ratio** from the **parameters.txt** file
  + **-progress** : show a dynamic progress bar and display estimated remaining execution-time.
* Parameters description:

|  |  |
| --- | --- |
| numInputNeurons | Input Vector Size |
| numOutputNeurons | Number of Target Outputs |
| mapHeight | The height of the neuron map |
| mapWidth | The width of the neuron map |
| neighbourhoodWidth | Initial Neighbourhood Width |
| learningRate | Initial Gain Term |
| datasetFile | File containing Data |
| dataSetMinValue | The minimum value of the integer dataset (0). Used in normalization |
| dataSetMaxValue | The maximum value of the integer dataset (15). Used in normalization |
| trainFile | The name of the train set. (if specified in the program arguments, this will be created from the dataset) |
| testFi;e | The name of the test set. (if specified in the program arguments, this will be created from the dataset) |

**Development:**

At first the code was not optimized, and compiler optimizations were not used. Training and testing a single network with sizes of 60x60 for 500 iterations took almost 2 days (38h).

****

Afterwards I noticed that the execution time of the same program written in java was much lower, therefore I begun optimizing code and introduced compiler optimizations to shrink execution time down to 40 minutes for a grid of 60x60 and 400 iterations.

**Results:**