Department of Computer Science

Artificial Intelligence
COS314

Project 3: Neural Networks & Meta-Heuristics

Due: Monday 29 May, 13:00

For this project you will use a meta-heuristic to train an LVQ-I neural network to classify a dataset.
What follows are the specifications of this project:

- You may implement the project in any language, provided that it will compile and run on Linux. Make sure that it compiles and runs on the computers in the Informatorium, in which case it will do so on my computer.
- Make sure that you have a makefile, or a build script, and provide clear instructions in a pdf file as to how your program should be compiled and run. Name the pdf file `????????.pdf`, where the question marks are replaced with the digits of your student number.
- Submit an archive, which should expand to a folder (the name of the folder is your student number), and your pdf file must be in the root of that folder.
- Your program has to make use of command line arguments. The first argument is the name of the data file. The second argument is the number of clusters to be used. The third argument is the output file. The fourth argument is the test file. The fifth algorithm specifies the algorithm to use. If the fifth argument is given as 0, then use the competitive learning approach given on the slides. If the fifth argument is given as 1, use your choice of meta-heuristic. The last argument is the number of iterations. Note that if you implement only the competitive learning approach as on the slides, you will earn a maximum of 50% for the project.
- Your program will receive a data file as input. Each row in the data file represents one data pattern to be clustered. The number of columns are the number of input variables, one per column. Columns are tab delimited. We will only use input variables that are numeric. You may test on any data set. Again, you can use data sets available on the UCI Machine Learning Repository, but consider only those sets where all the input variables are numeric.
- You may use your choice of a genetic algorithm or a particle swarm optimization algorithm to train the neural network. Describe the algorithm that you have used in detail, and provide all detail for your work to be repoducible. The detail should include the values that you have assigned to control parameters.
- Your program will provide the following output, printed both to the screen and to the output file provided as the third argument:

– The centroid vectors, in the following format:

```
Centroid 1: value1 value2 value3 ... valueN
Centroid 2: value1 value2 value3 ... valueN
...
Centroid K: value1 value2 value3 ... valueN
```

where N is the total number of input variables and K is the total number of clusters.

– The quantization error and the average inter-cluster distances, in the following format:

```
Quantization error: value
Average inter-cluster distance: value
```

The quantization error is simply the average over all the inter-cluster distances.

– For each centroid, the average intra-cluster distance, as follows:

```
Intra-distance cluster 1: value
Intra-distance cluster 2: value
...
Intra-distance cluster K: value
```

– For each pattern in the test file, provide the cluster centroid to which the pattern belongs, as follows:

```
Pattern 1: clusterNumber, distance1, distance2, ..., distanceK
Pattern 2: clusterNumber, distance1, distance2, ..., distanceK
...
Pattern P: clusterNumber, distance1, distance2, ..., distanceK
```

where P is the total number of patterns in the cluster, and $distance_k$ is the Euclidean distance of the pattern to each of the cluster centroids.

If you have done all of the above, and it works correctly, you will earn a maximum of 80%. To earn another 20%, you need to implement an approach to dynamically grow the number of outputs. That is, to also determine the optimal number of clusters. If you do implement this dynamic growing network, then use -1 as the second argument on the command line. Describe in detail how you have implemented this growing mechanism. You need to figure out yourself how you should do this.