WIA3001 Soft Computing

Tutorial 4: Introduction to Neural Networks and Genetic Algorithms

1. Unzip the file “tutorial4.zip”. There should be five files.
   1. create\_network.m: generates a simple multi-layer perceptron neural network.
   2. iris.csv: an example dataset. Each row is one data sample. Each column is one attribute, except for the last column which is the class label.
   3. run\_ann\_withGA.m:
   4. run\_ann\_withoutGA.m: generates a MLP neural network and immediately tests it with the dataset without training.
   5. test.m: for testing the neural network with a dataset.
2. You can run the program immediately by typing “**run\_ann\_withoutGA**” in the MATLAB command window. This will generate an MLP with randomized weights and tests it with the example dataset. An accuracy score will be given.

Since the accuracy depends on the MLP weights, and the weights are randomly generated, you can get different results if you run the program multiple times.

In “**run\_ann\_withoutGA.m**”, you can set the random number generator’s seed number at **Line 6**. The seed number ensures the RNG is not random and will produce the same results every time. Changing the seed number to other values will produce different random numbers.

**Exercise**: try different values for the seed number. What number will give you the best result?

1. The MLP is generated using the file “**create\_network.m**”. The parameters of the network are defined at **Lines 4-7**.

**input\_layer\_units** should be the same as the number of attributes from your dataset.

Likewise **output\_layer\_units** should be the same as the number of class labels from your dataset.

You can try to experiment with different number of **hidden layers**. For each hidden layer, set the number of hidden nodes in **hidden\_layer\_units**.

**Exercise**: Does accuracy increase when you increase the hidden layers? How many layers will give you the best accuracy? What about the number of hidden layer units?

If error occurs, it means that the settings you used is incompatible. Try to analyze the dimensions of the network weights. Each successive layer must be able to perform matrix multiplication.

1. The operations of the MLP is located in the file “**test.m**”.

In Line 25, the ReLU activation function is used for all hidden layers.

In Line 22, the sigmoid activation function is used for the output layer.

**Exercise**: you can try to replace them with other activation functions from “Lecture 2: Perceptrons”. Does changing the ReLU and sigmoid functions affect the accuracy of the model? Why?

1. Now we’ll use a genetic algorithms (GA) to look for the optimal weights for the MLP.

Run the program “**run\_ann\_withGA**” in the MATLAB command window.

It will run GA to optimize MLP weights and output the progress. When finished, the program will output three variables: the population of chromosomes, their fitness scores, and the progress. Type “**plot(progress)**” to see how the best and average fitness scores progress.

**Exercise**: try changing the seed number for the GA. Does the GA gives better or worse results?

**Exercise**: try changing the GA parameters like population size, the selection rate, and the mutation rate. Does the GA achieve better or worse results? Did the GA achieve the results in fewer generations or more generations?

1. The GA uses a stopping criterion at **Line 67**.

It stops when the average fitness score does not improve over a few generations (i.e. **Lines 51-57**).

**Exercise**: Try a different stopping criterion. Comment **Lines 70-72** and uncomment **Lines 75-82**. This new criterion slowly adjusts the selection rate and mutation rate whenever the fitness score stops improving. Run the GA again.

Does the GA achieve better or worse results? Did the GA achieve the results in fewer generations or more generations?

1. The program uses a default dataset “**iris.csv**”. You can try to look for other datasets, for example, from <https://archive.ics.uci.edu/ml/datasets.php>.

If you want to try the program for different datasets, make sure of the following:

1. The dataset is stored as a CSV file. You can see how “iris.csv” is arranged by opening it in MATLAB or Notepad.
2. Each row represents one data sample and each column represents one attribute.
3. The last column should be the class label. Labels should be continuous integers (i.e. [1,2,3], or [1,2,3,4,5,6,7]).
4. To load the dataset, change **Line 9** in “**run\_ann\_withGA**” and “**run\_ann\_withoutGA**”.
5. In addition, you need to modify the structure of the MLP in “**create\_network.m**”.

Change “input\_layer\_units” to match the number of attributes in the new dataset, and “output\_layer\_units” to match the number of unique class labels.