Assignment 3: Out date: 12.03.14 In date: 23.04.14

Artificial Neural Networks, Back-Propogation Learning**, Genetic Algorithms, Evolving the weights of an Artificial Neural Network for Character Recognition Problem**

In this assignment, you are asked to do character recognition using a feedforward artificial neural network. Character recognition is not a linear problem for that reason, a multi layer feedforward network needs to be used, and it needs to be trained to learn a set of characters.

Learning Method:

a) Either implement the backpropogation learning algorithm to train the neural network for the character recognition problem.
b) Or apply the genetic algorithm to train the weights for the neural network.

1) Training the Network with the back-propagation Learning Algorithm

The topology of the network:

The only constrained related to the topology is that it needs to be a multi layer feed forward network. That means, the network will be composed of a number of nodes in the input layer, a number of nodes in the hidden layer, and a number of nodes in the output layer. The number of hidden layers, the number of nodes in the hidden layer, as well as the learning rate should be parametric in your implementation.

a) **Input data**

A character set given on fronter is for training the network. After you unzip the character set, you will find 20 separate files for each character. You might use the first 10 for each character as the training set, and the second 10 as the test set. Each character has a size of 30 x 60 pixels. The resolution of the character set is enough to learn the 26 characters (A-Z).

b) **Hidden layer**

There does not exist a good rule for selecting the number of hidden units for performing backpropogation algorithm successfully. For that reason, we ask you to experiment with different number of hidden layers and hidden units in a layer to find out what gives the most satisfactory result.

c) Output data

Since we are particularly interested in the categorization of the characters, we use one and only one output unit which responds to a certain character. The output unit with the highest activity will be accepted as the character which matches the input data best.

d) The Program

Use a file to keep information about the network such as the number of characters or patterns to learn, the number of hidden units to use, the learning rate and what error limit. Store the eventual

set of weights that gives the best recognition rate, and demonstrate your program with those weights uploaded.

The weights are assigned randomly to the neural network. Indicate the weight range.

Define a certain number of epochs where at each epoch, all the individuals in the training set will be used for updating the weights of the neural network.

Things to do:

a) Draw the topology that you trained for learning the given character set (Set the learning rate).
b) Test the program for various numbers of hidden layers, and explain if the program learns using one hidden layer, or whether there is any advantage observed with using more than one hidden layer.
c) Test your program for the given training character set. Report the success.
d) Test your program for the given test character set. Report the success.

2) Training the Network weights with a genetic algorithm approach

If you are not using back-propogation algorithm in order to learn the weight set, then you are asked to use the genetic algorithm approach to train the weights of the neural network.

Weights: The weights are assigned randomly to the neural network. Indicate the weight range.

Encoding: Real values in chromosomes represent weights in the neural network.

Initial Population: Set a certain number of individuals in the initial population which will be kept constant over generations.

Crossover: Use single point crossover.

Mutation: Randomly decide if an individual in a generation will be mutated or not. Define your mutation operator.

Fitness Function: Define a fitness function and calculate the fitness of each individual in each generation. An example fitness function for an individual might be the number of characters which an individual classifies correctly in a given generation.

Selection Methodologies: Use any of the selection methodologies roulette-wheel selection, rank selection in order to select the individuals for the next generation to combine. In each generation, the number of individuals should be kept constant.

Things to do:

a) Draw the neural network topology that you trained for learning the given character set

b) Test the program for various numbers of generations, and find the average number of generations which the program learns using the given character set.
c) Test your program for the given training character set. Report the success by plotting the fitness landscape.
d) Test your program for the given test character set. Report the success by plotting the fitness landscape.

The delivery:

The code for either the backpropogation learning or the genetic algorithm should be submitted to the fronter for your group on **22th April, by midnight**. Also, you will be delivering a report to fronter for the parts in the assignment text where you are asked to give/explain in report, e.g. plots, fitness landscape, topology etc.

On 23rd April, each group will make a demonstration of 30 minutes of their implementation at a predefined time.