Scott Bebington

U21546216

COS 314 Assignment 3 Report

Table of Contents

Back Propagation Neural Network

For this part of the project a Neural Network using back propagation was used to determine the weights associated with each input of mushroom data in the training data set.

Activation function

For this NN I use the sigmoid activation function as it is simple to implement and it returns values between 0 and 1 which is ideal for this type of classification algorithm.

Hidden layers

I decided to only use 2 hidden layers for this NN, this is to make the architecture of the program simpler as well as reducing the training time as the dataset is very large.

Learning rate

I experimented with the learning rate for the NN but in the end I decided to go with a low learning rate or 0.2, this would be to ensure that the data is trained more accurately and wouldn't run into a non-convergence loop.

Stopping Condition

I decided to run the program for 10 epochs, through trial and error I found that convergence for this algorithm was almost immediate, and the accuracy didn't change much throughout the experiments with different learning rates. For this reason, running the program for a large number of epochs wouldn't make sense.

Data preprocessing

For this model I found that the optimal values to get the best results was to train using the cap-diameter and stem-width as the values were diverse enough for the model to be able to train well enough.

Model

1. Read the train data to be processed.
2. For each epoch the following takes place.
    a. The data is fed forwards to obtain a classification using the sigmoid function.
    b. The data is then fed backwards to recalculate the weights associated with each input for the hidden layer and the output layer. After which the output layer error and hidden layer are calculated. Finally, the bias is recalculated.
3. The test data is read in and fed forward using the final weights that were calculated.
4. The metrics are calculated using the test data's results.

Table of results

```
Testing the neural network
Accuracy: 58.8723%
Specificity: 0
Sensitivity: 1
F-measure: 0.741127
```

Genetic Programming Classification Algorithm

For this section of the project a GP classification algorithm will be used to find the optimal tree to classify the mushroom data.

Tree depth

To limit the size of the trees I used a tree size of 6, this was to ensure that the size of the tree growth throughout the training process didn't get exponentially large as this can happen very quickly.

Mutation rate

As mentioned in tree depth, a smaller mutation rate of 0.3 is used. This is to make sure that the population DNA isn't completely replaced after a few generations as parents can make successful classifications. A smaller mutation rate also ensures that the tree size doesn't increase exponentially due to a terminal node to get a whole new tree.

Model

1. The training data is read in and stored.
2. A random initial population is made.
3. Using tournament style selection, the fittest parents are selected to create the new generation.
    a. To calculate the fitness the operations in the tree are carried out from bottom to top.
4. The test data is read in and used to test the final calculations.
5. The metrics for the test are calculated and outputted.

Table of results

```
Genetic Program
Accuracy: 58.8723%
Specificity: 0
Sensitivity: 1
F-measure: 0.588723
```