COS 314: Assignment III

Iwan de Jong

u22498037
University of Pretoria

Contents

1 Assignment-specific Parameters

A seed value of 1 has been used.

2 Pre-Processing

For pre-processing, the data was normalised using the Min-Max method. Both the testing and training data were normalised using the same parameters. The values represented in the data were scaled to a range of 0 to 1.

3 ANN Technical Specification

3.1 ANN Structure

The ANN consists of 3 layers: an input layer, a hidden layer, and an output layer.

The input layer consisted of 8 nodes, the hidden layer consisted of 5 nodes, and the output layer consisted of 1 node. The output layer uses a sigmoid activation function, while the hidden layer uses a sigmoid activation function. It only has one output node, which is used to determine the output of the ANN (which is binary (either 0 or 1)).

It was then feed-forwarded through the network, and the error was back-propagated to adjust the weights.

When training, I also made the mushroom input random, so that it wouldn't converge to a local minimum.

In this ANN, 0 represented a non-edible (toxic) mushroom, while 1 represented an edible mushroom.

3.2 Activation Function

The activation function used is the sigmoid function. The sigmoid function is defined as:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

Likewise, the derivative of the sigmoid function is defined as:

$$f'(x) = f(x) * (1 - f(x)) \tag{2}$$

I used the sigmoid function as it is a common activation function for binary classification problems, since it maps the output to a range of 0 to 1.

3.3 Learning Rate

The learning rate is set to 0.23.

For this model, the parameter greatly affects the convergence of the model. I chose a learning rate of 0.23 as it provided the best results. I found that lower learning rates resulted in weaker results, while higher learning rates resulted in the model not converging.

3.4 Stopping Condition

The stopping condition is set to go 1000 epochs, regardless of improvement. I found that the model converged within 1000 epochs, and that the error rate did not improve significantly after this point.

3.5 Additional Statistics

```
Statistics
Precision: 0.742678
Recall: 1.000000
Accuracy: 0.796020
F1 Score: 0.852341

Output Statistics:
Mean: 0.464287
Variance: 0.000900
Standard Deviation: 0.029995
Range: 0.116922
```

3.6 Critical Analysis

The ANN model had a great accuracy and good recall, but the precision was not good. As seen with the output, the variance is extremely low, which is not good, since the value of the output is binary, there should be a higher variance. The range is also very low, which is not good, as the output should be more varied. This will make it easy for the model to predict the output.

Consequently, when the seed/initial weights are changed, the model will not perform as well, as the model is not generalised to perform well on unseen data.

Overall, it does have a good accuracy, and the mean is close to 0.5, which is good, as the output is binary.

3.7 Graphical Representation

The graphical representations obtained from ‘output.csv‘ are generated after each epoch (generated by ANN.cpp).

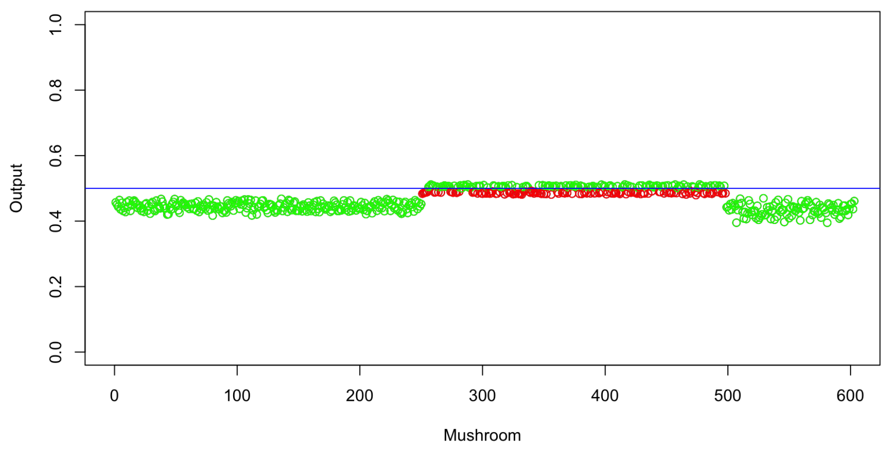

Figure 1: Scatterplot of the expected data versus the ANN output. The red dots represents errors.

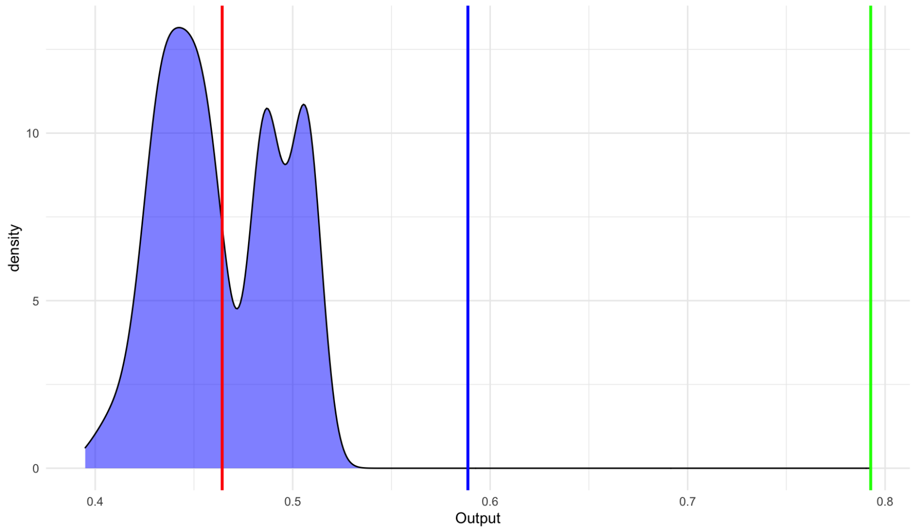

Figure 2: Density Plot of the ANN output. The red line gives the average output, the blue line gives the expected output, and the green line gives the actual output.

The graphical representations obtained from ‘error.csv‘ are generated after each epoch (generated by ANN.cpp).

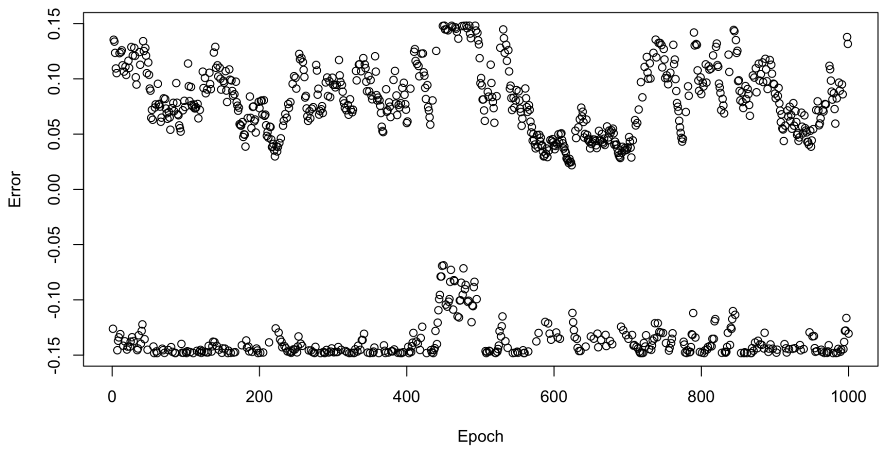

Figure 3: Error per Epoch Graph

4 Genetic Programming Technical Specification

4.1 GP Structure

- Population size: 100
- Number of generations: 50
- Crossover rate: 0.7
- Mutation rate: 0.08
- Max Depth: 4

4.2 Graphical Representation

The graphical representations obtained from ‘output.csv‘ are generated after each epoch (generated by GP.cpp).

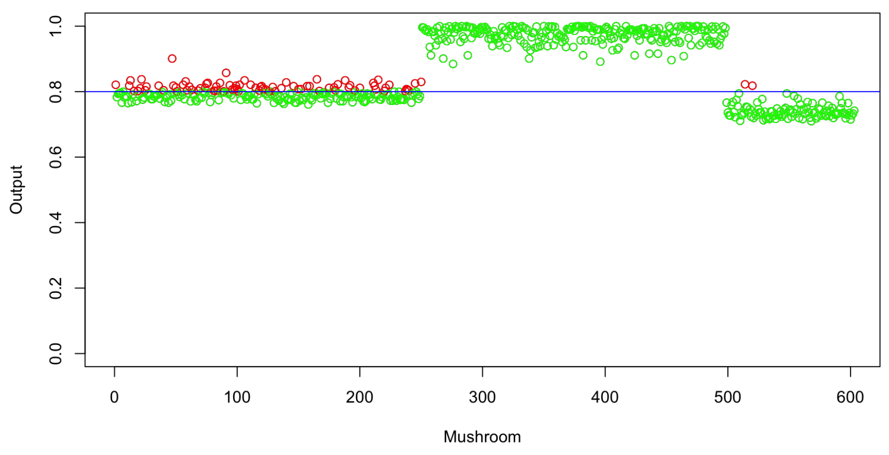

Figure 4: Scatterplot of the expected data versus the GP output. The red dots represents errors.

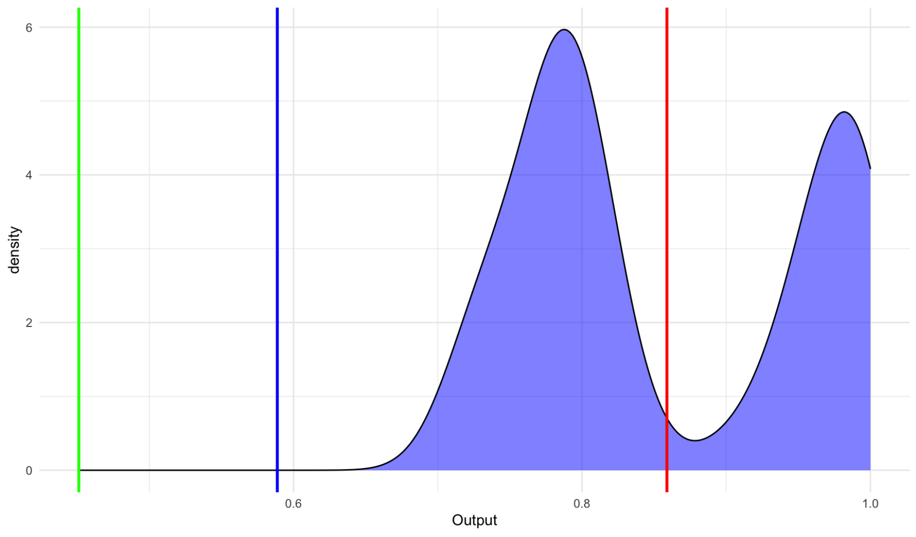

Figure 5: Density Plot of the GP output. The red line gives the average output, the blue line gives the expected output, and the green line gives the actual output.

The graphical representations obtained from ‘error.csv‘ are generated after each epoch (generated by GP.cpp).

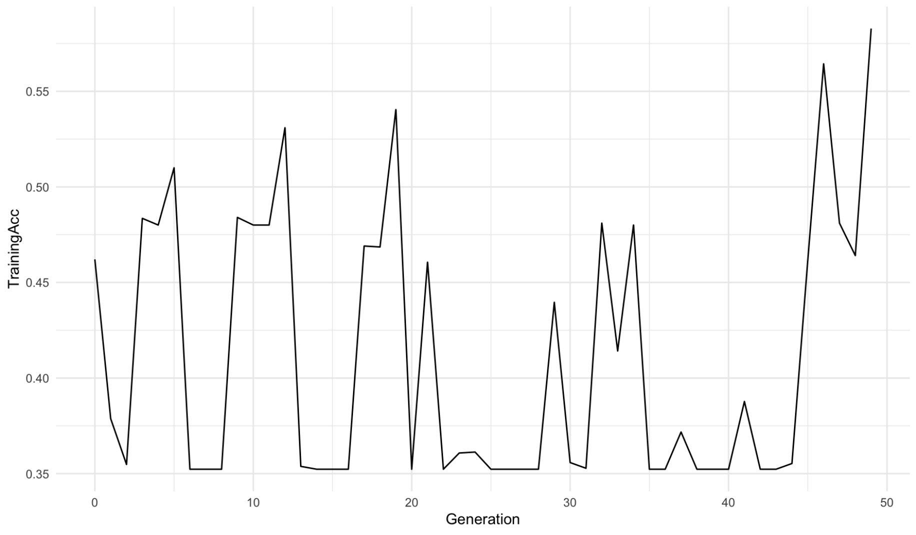

Figure 6: Error per Epoch Graph

4.3 Critical Analysis

The GP model provided much safer output in the sense that good mushrooms can be seen as good, and bad mushrooms can be seen as bad. The accuracy is quite high, and performs well with the test data. The model is also generalised, as it performs well with unseen data.