ROCHESTER INSTITUTE OF TECHNOLOGY

CSCI-630 Project 2: Metal Part Sort

Thang Nguyen - Ishan Gulhane

November 20, 2015

1 Algorithms

Briefly discuss the similarities and differences between the two learning algorithms. Which type did you expect to perform better in the experiment, and why?

Both neural network and decision tree can be employed on classification task and provide robust model to classify data but they have some different:
- While **neural network may give us better classification resul**t by using strong activation functions and multiple nodes/hidden layers, it is not easy to visualize the network, while in decision tree algorithm the tree can be represented visually.
- It takes a long time(large epochs) to train neural network while the decision tree train very fast, thus with large data sets decision tree may be a better method for classification tasks

2 Data

Figure 2.1 and 2.2 show plots for the training and test data sets.
a) Training set

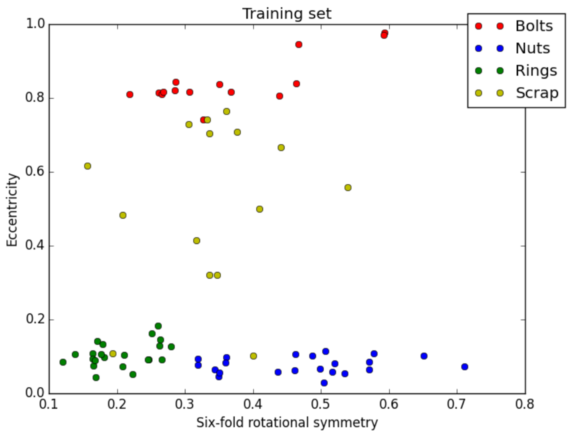

Figure 2.1: *Training set*

The training samples are distributed in separated region

b) Test set

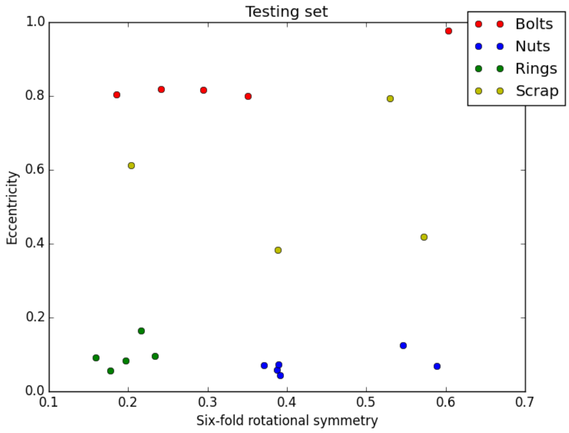

Figure 2.2: *Testing set*

It can be seen from figure 2.2 that in the testing set, all testing samples are linearly separated, so it is easy to reach correct classification.

3 Results

3.1 MLP

i) Figure 3.1 to 3.5 show the test samples and classification regions produced by different numbers of training epochs 0,10,100,1000 and 10000

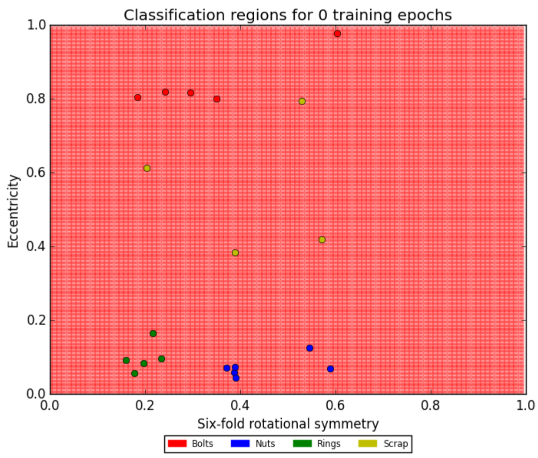

Figure 3.1: *Epochs = 0*

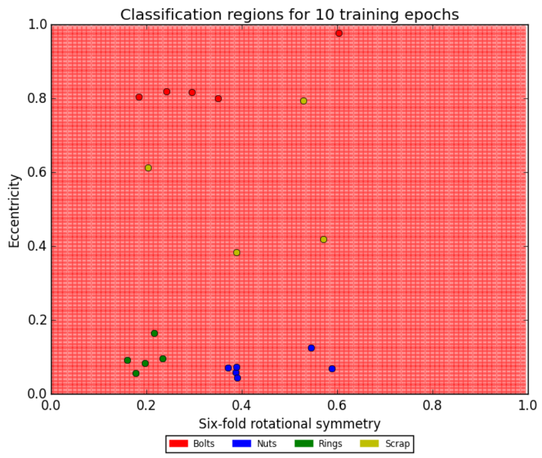

Figure 3.2: *Epochs = 10*

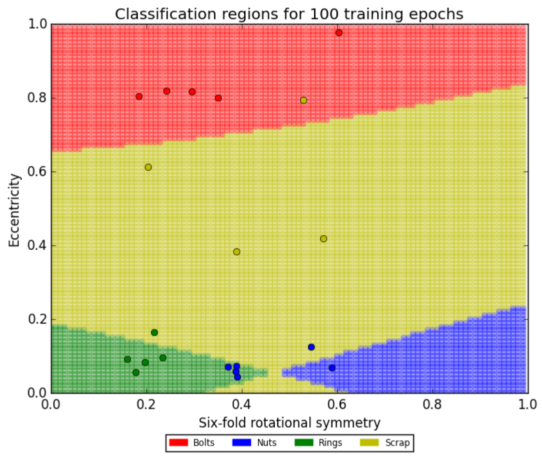

Figure 3.3: *Epochs = 100*

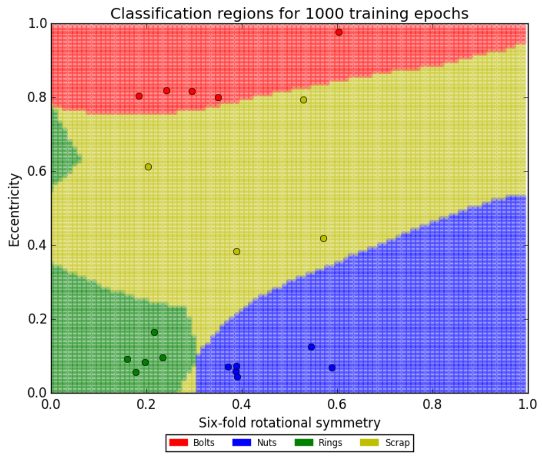

Figure 3.4: *Epochs = 1000*

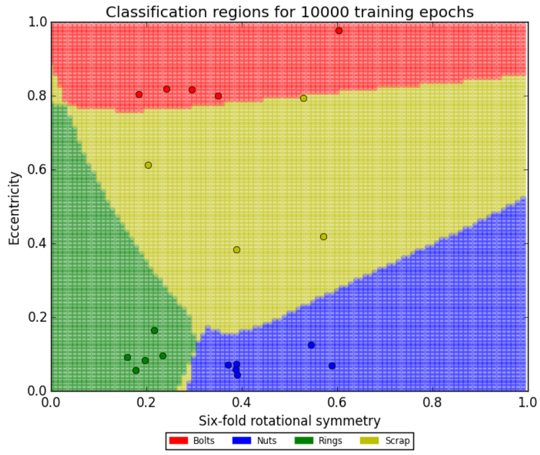

Figure 3.5: *Epochs = 10000*

ii) Learning curve image(SSE vs Epoch) for the trained MLP

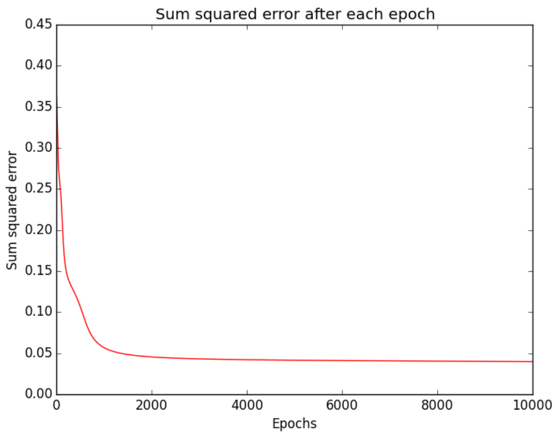

Figure 3.6: *Learning curve*

iii) Table 3.1 showing the recognition rate and profit for each number of saved epochs for the MLP. In epoch 0 and 10, because there is not enough data to calculate correct output thus the model recognize all of samples as bolts to maximize profit

Epochs	Recognition rate	Profit(cents)
0	25%	-5
10	25%	-5
100	50%	65
1000	100%	203
10000	100%	203

Table 3.1: *Recognition rate and profit for each number of saved epochs for the MLP*

3.2 Decision Trees

i) Figure 3.7 and 3.8 showing the test samples and classification regions produced by each of the two decision trees.

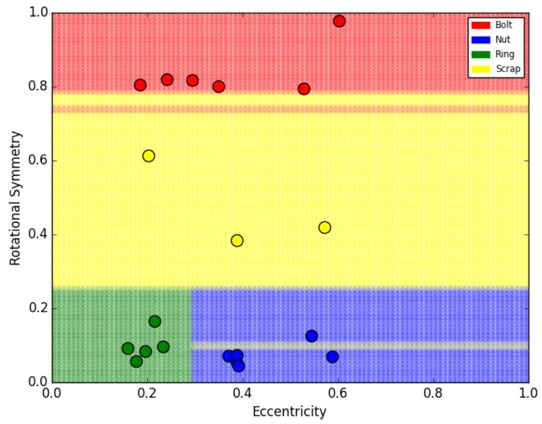

Figure 3.7: *Test samples and classification regions produced by normal tree*

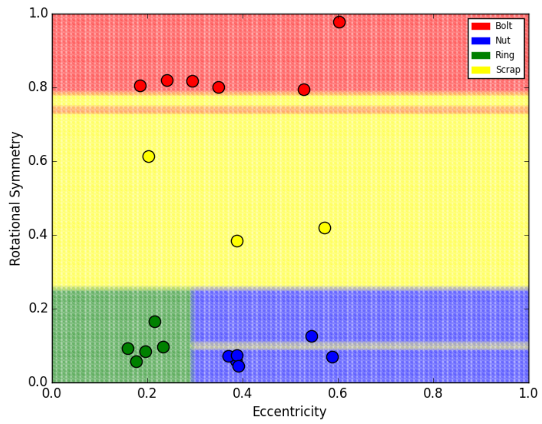

Figure 3.8: *Test samples and classification regions produced by pruned tree*

ii) Figure 3.9 and 3.10 show how feature space is split by each decision tree.

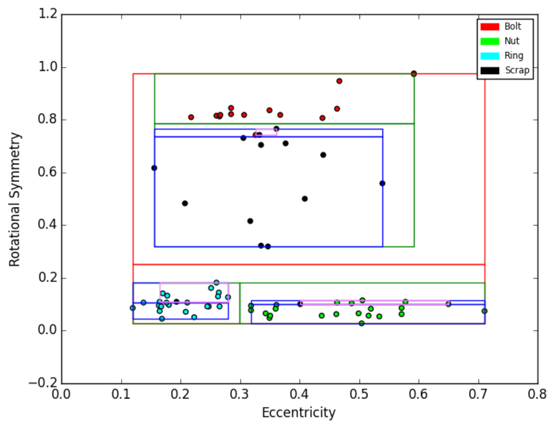

Figure 3.9: *Feature space splitted in normal tree*

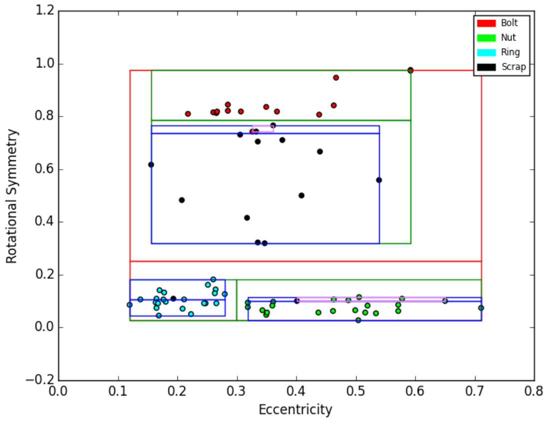

Figure 3.10: *Feature space splitted in pruning tree*

iii)Table3.2 The recognition rate and profit obtained by each decision tree, along with the tree metrics produced by trainDT.py.

Tree	Normal tree	Prunned tree
Recognition rate	95%	95%
Profits(cents)	199	199
No of nodes	19	17
No of internal nodes	9	8
No of leaf nodes	10	9
Max depth	4	4
Min depth	2	2
Average depth	3.5	3.333

Table 3.2: *Recognition rate and profit for each number of saved epochs for the MLP*

4 Discussion

a) Which versions of the classifiers performed best in terms of 1) accuracy and 2) profit? Did this meet your expectations?
In both term of accuracy and profit, MLP seems perform better, the result comeback as expectation

b) How do the hypotheses (i.e. class boundaries) and performance metrics differ between the different version of the MLP and decision trees, and why?
- Multi-layer Perceptron: As can be seen from figure 3.1 and 3.2 for epoch = 0 and epoch = 10, respectively, MLP do not achieve enough weight accuracy, there is not much different in 4 nodes of output, thus all test sampels are classified as scrap with no boundaries between classes. As epochs increase, the perceptron network achieve better weight for all layers, thus improve accuracy of the output. With epochs = 100, the network reach 50% recognition rate and reach 100% accuracy at 1000 epochs. Because the testing samples are small and separated with not much noise thus with small epochs the network can achieve perfect results. In 0 and 10 epoch
- Decision trees: As can be seen from table 3.2, only one node is pruned, thus there is not much difference between regions splitting of normal tree and pruned tree and both tree can achieve 95% recognition rate, same performance.

5 Bonus