Harshvardhan
Student Id: 609162
COSC 522: Machine Learning

Task 1

I trained BPNN model on the training dataset and applied it on the test dataset. I got following accuracy convergence plot. Following hyperparameters were used: 300 epochs, 10 mini batch size and a learning rate of 1. The neural network has three hidden layers with ten nodes each.

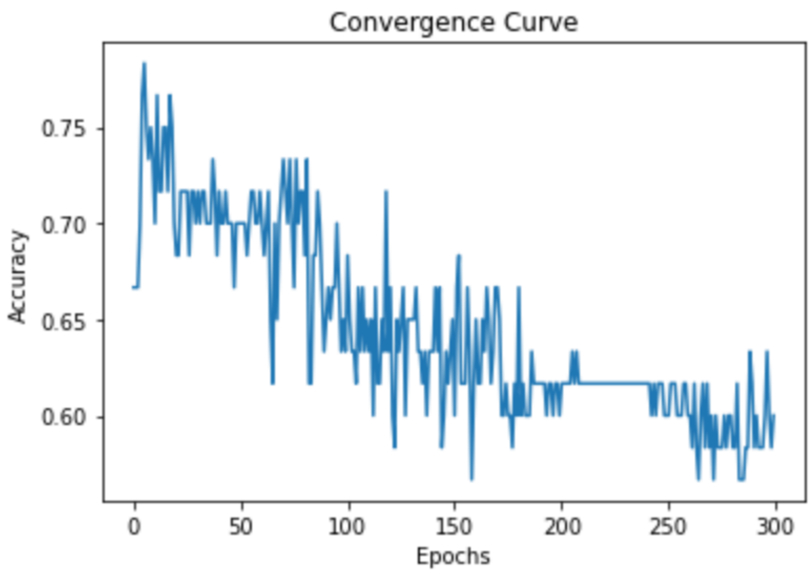

Figure 1 Accuracy convergence curve for BPNN on Pima dataset.

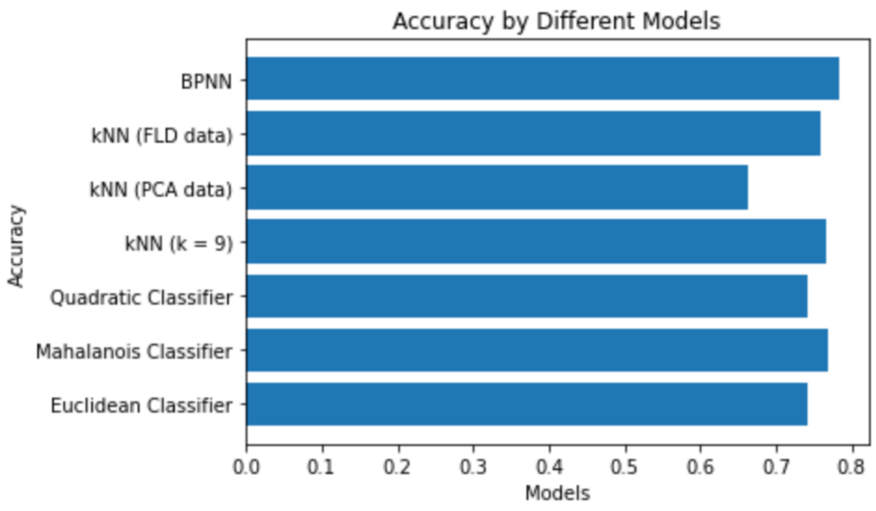

Figure 2 Bar plot of accuracies by various methods. BPNN and Mahalanobis classifiers perform the best.

Method	Accuracy
Euclidean Classifier	0.741
Mahalanobis Classifier	0.768
Quadratic Classifier	0.741
kNN (k = 9)	0.765
kNN (PCA data)	0.663
kNN (FLD data)	0.759
BPNN	0.783

Table 1 Accuracy of various methods used throughout the course on Pima dataset.

As observed, BPNN performs best with 78% accuracy. Mahalanobis classifier and kNN (k = 9) closely follow with 77% accuracy each. Based on all these accuracies, I would likely use Mahalanobis classifier or kNN classifier, depending on how confident I am about the parameters I use in Mahalanobis classifier. It also demonstrates that it is not necessary that a complex neural network would perform better than other statistical methods.

Task 3

The codes for CARLO are sourced from their GitHub repository. Some of these will be modified for this problem set's requirement.
Repository: https://github.com/Stanford-ILIAD/CARLO
Parts of this code base was given to us by Prof Hairong Qi. Some of them were written by Ximu Zhang.

Human Control

The error in human control is plotted below. I have excluded the errors which were greater than 6 as around 500th iteration, the error at a point shoots up.

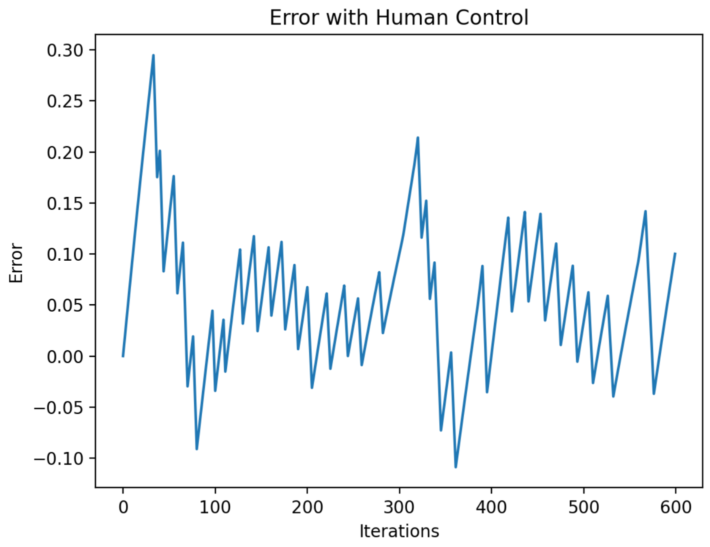

Figure 3 Error rate with time/iterations for human control.

Human Control to False

The error with the base policy (where human control is set to false) is plotted below. I have excluded the errors which were greater than 6 as around 500th iteration, the error at a point shoots up.

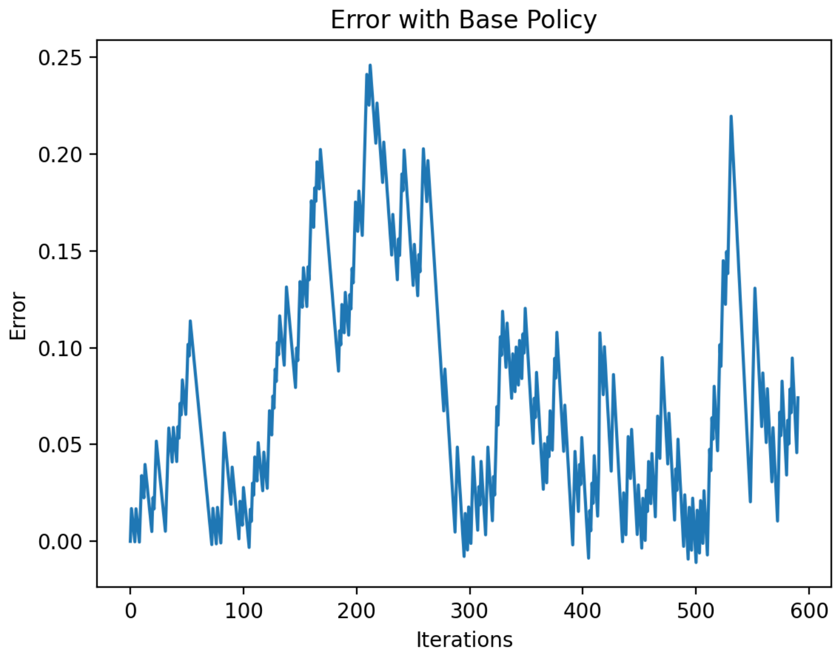

Figure 4 Error vs iterations/time for base control (human_control = False).

PID Control

Error plot when PID control is used is plotted below.

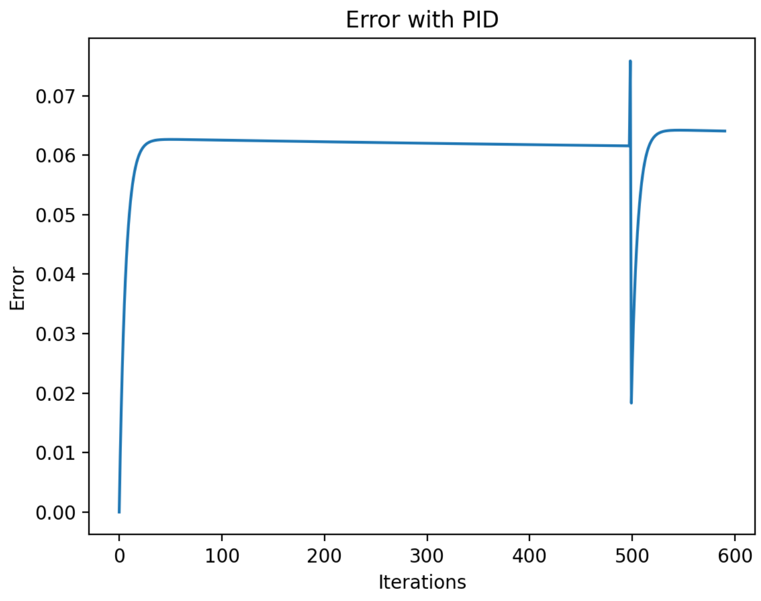

Figure 5 Error vs time/iteration for PID control.

Q-Learning (Reinforcement Learning)

After applying reinforcement learning, this is the error plot. I have excluded the errors which were greater than 6 as around 500th iteration, the error at a point shoots up.

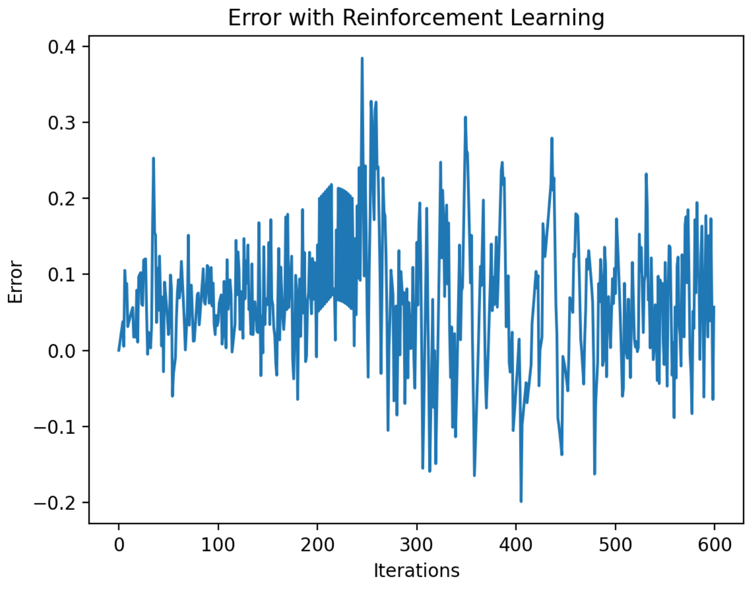

Figure 6 Error vs time/iteration for Q Learning or Reinforcement Learning.

Discussion

The error plot with human controls looks most like random walk centered around zero — which should be the ideal case because I can most likely control the car much better than any algorithm in this simple case.

The error with base policy shows that the algorithm is likely biased at certain positions. Position (iteration) 150 to 250 show high biased errors. In PID, the error shows that the algorithm is biased after an iteration and doesn't learn anything new anymore. The reinforcement learning error plots show that it is best likely to be close to the true absolute ideal path. It frequently changes its positions and thus has most jiggleness in the plot.

Based on these errors, I would likely only use human control. If I had to rely on an algorithm, I would follow reinforcement learning because this gives the least error and is likely unbiased (as noted from visual inspection of the plots).

Reference

Cao, Z., Biyik, E., Wang, W., Raventos, A., Gaidon, A., Rosman, G., & Sadigh, D. (2020). Reinforcement Learning based Control of Imitative Policies for Near-Accident Driving. In Proceedings of Robotics: Science and Systems (RSS).