COMP 551 - Project 2 Report

Team 7: Xinchen Hou, Yixuan Li, Xuanzi Wang

Abstract

In this project, two datasets will be used to explore the optimization process of binary classification, using the technique of Logistic Regression.

For part 1, we investigated in optimizing the convergence of gradient-based methods by applying Logistic Regression and mini-batch stochastic gradient descent to the Diabetes dataset. Momentum is added in the later stage to compare its impact on convergence. We find that for simple Logistic Regression, the highest accuracy is reached when 1e6 max iteration and 1e-4 learning rate are chosen as parameters. Model with 50 batches and 30 epochs can achieve high accuracy while maintaining a reasonable running time. Adding momentum to the mini-batch method would have a positive effect on both the speed and accuracy.

In part 2, we used the Logistic Regression model to predict records in fake news dataset, and determine if the text is generated by a human or computer. Parameter tuning will be applied to two prediction models to achieve higher accuracy. In the end, 71.93% test accuracy is achieved with the Logistic Regression classifier with parameter tuning.

Introduction

For part 1, we first apply Logistic Regression to the Diabetes dataset. Highest accuracy of 72% is reached when we choose 1e5 max iteration and 0.0001 training rate. Mini-batch SGD is then added to test the performance of the model with batch size of 6, 12, 20, 30, 50 and 100, as well as a full-batch size for comparison. Since the norm of gradient does not converge to 0 always, we take the training and validation accuracy as the standard. The point that the accuracy curves become smooth represents the number of epochs that need to be taken for convergence. We get the result that when the number of batches is small, the test accuracy reaches its largest. Also, as the batch size increases, the time taken for convergence decreases, but the test accuracy decreases as well. Figure 2.1 represents the accuracy when batch size is 50, which becomes smooth after 30 iterations.

At last, we add momentum to the gradient descent method. By choosing 0.99, 0.6, and 0.2 as the momentum coefficients and applying them to the full batch, we find that both the convergence speed and the test accuracy decrease as the momentum coefficients decrease. After adding momentum to the mini-batch SGD with batch sizes of 6 and 100, we find that the time used to process one epoch decreases, and the test accuracy has also increased compared to the mini-batch method implemented without momentum.

The goal of Part 2 is to correctly determine whether the given text is generated by a computer. The fake news dataset is preprocessed and transformed before training. A base model with Logistic Regression is achieved with a test accuracy of 71.77% with inverse regularization strength C=100. In later stages, models with basic Logistic Regression and SGD classifiers are applied in Random Search with 5-fold cross-validation in order to find parameters with higher test accuracies.

For the SGD method, the best test accuracy of 71.90% is achieved with parameters of 1e-5 learning rate and log loss function with no loss penalty.
For the Logistic Regression method, 71.93% test accuracy is achieved with liblinear solver, l2 penalty, and 54.5 inverse regularization strength.

Dataset

In part 1, we use the Diabetes dataset composed of 9 columns: 8 columns of attributes that might potentially affect diabetes outcome, and 1 column representing the outcome. We want to predict the outcome is 0 or 1 based on the values of the previous 8 columns. The 'Outcome' column is used as the test column, and all other columns are used for training.

We tested the convergence speed and test accuracy of the models when they are implemented using full-batch, mini-batch with different batch sizes, different momentum coefficients to the full-batch, and adding momentum to mini-batch.

In part 2, we use the Fake News dataset composed of two columns. The text column stores the text we examine, and the Label column represents whether it is generated by a computer or a human. Data in the text column is preprocessed before applying it to the model. All the text data is transformed into lowercase, and any special characters or symbols are removed from the text. In the next step, the text data is standardized with CountVectorizer and TF-IDF Transformer to reduce any meaningless wordings.

For the modeling phase, three models are achieved to test out accuracy. The first one is the basic Logistic Regression model achieved by searching for the highest validation accuracy among different inverse regularization strength C. The second model is achieved with the Logistic Regression model with random searches in more parameters of loss penalty, solver, and C. And the third model is achieved with the SGD classifier with random searches in parameters in loss, loss penalty, and learning rate. And the test dataset will be used to compare the test accuracy among three models.

Results

Part 1

Figure 1.1

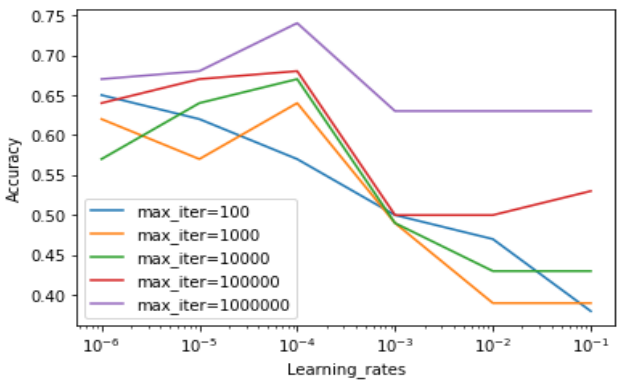
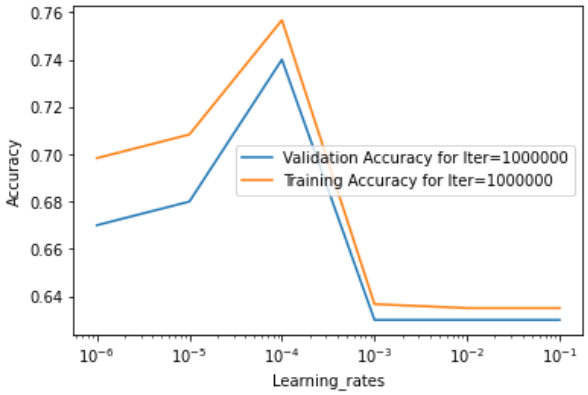


Figure 1.2



From figure 1.1, we see that with fixed learning rates, the accuracy increases on average as we increase the max_iter. When the max_iter is fixed, the overall accuracy increases as the learning _rates approaches 0.0004, and decreases afterwards. Based on the graph, the highest accuracy is reached when max_iter=1e6 and learning_rates=0.0004. As shown in figure 1.2 of

appendix, when the learning rate approaches 0, the distance between validation accuracy and training accuracy decreases, indicating that the solution is fully converged.

Figure 2.1 from appendix represents the validation set and the training set when batch sizes are 6, 20, 30, 50, and 100. When size is 50, the accuracy does not increase starting from 30 iterations over the whole dataset. We thus conclude that the model converges after 30 epochs.

Observing graphs from figure 2.2, which represents the time taken to process one epoch, and figure 2.3, which represents the time taken to converge using different batch sizes, we conclude that the greater the size, the less updates of w's are needed, and the model is faster to converge. Figure 2.4 plots the test accuracy against the batch sizes. We see that for the sizes that we tested, the largest accuracy is achieved when batch size is 12.

Figure 3.1 in the appendix represents the accuracy curves after adding momentum to the full batch. We could achieve a test accuracy of approximately 0.7. Decreasing momentum coefficient from 0.99 to 0.6 and 0.2 results in an decrease in test accuracy and increase in convergence speed. Figure 3.2 and 3.3 in the appendix represents adding momentum to the mini-batch method. They both converge quickly and increase their test accuracy compared to the results derived from the mini-batch method without momentum.

Part 2
Based on figure 4, we observed that as inverse regularization strength C increases, the validation and training accuracy increase as well. The distance between them increases as C increases, but the distance stabilizes after C=10.

However, after we tested the model after selecting more features from random search, the test accuracy did not increase significantly. Increasing max iteration for logistic regression did not have a great impact on the accuracy as well.

Discussion and Conclusion

For Part 1, when we simply apply the Logistic Regression model, we reach the highest test accuracy of 73.5% with 1e6 max_iter and 1e-4 learning rate. This model converged as we see in figure 1.2. But as the number of max_iter increases, the time taken increases as well. When we use 1e6, it does take a long time. For mini-batch SGD, we reach the highest test accuracy of 75% when the batch size is 12. However, the convergence speed is also high for batch size of 12 compared to batch size of 50. Adding momentum would make the convergence faster. However, in our case, the test accuracy was not significantly increased.

For Part 2, the testing accuracy (~72%) does not change much between the basic model and improved models with parameter tunings. Since a Random Search is applied for parameter tunings, there are possibilities that the search method did not find the optimal parameters with limited max search iterations. In order to increase the accuracy, one could increase the max iterations allowed for searching, as well as add more options for possible parameters. But the tradeoff would be a significantly longer searching phase.

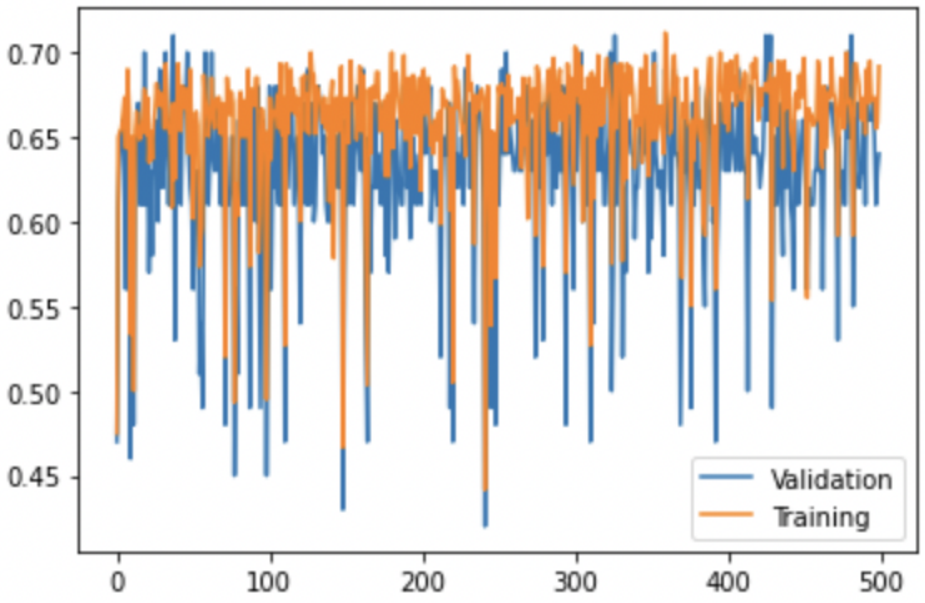
Statement of Contribution

Yixuan and Xuanzi completed part 1 of the project, and Xinchen completed part 2 of the project, and helped with generating basic models in part 1. Report is evenly split among three members.

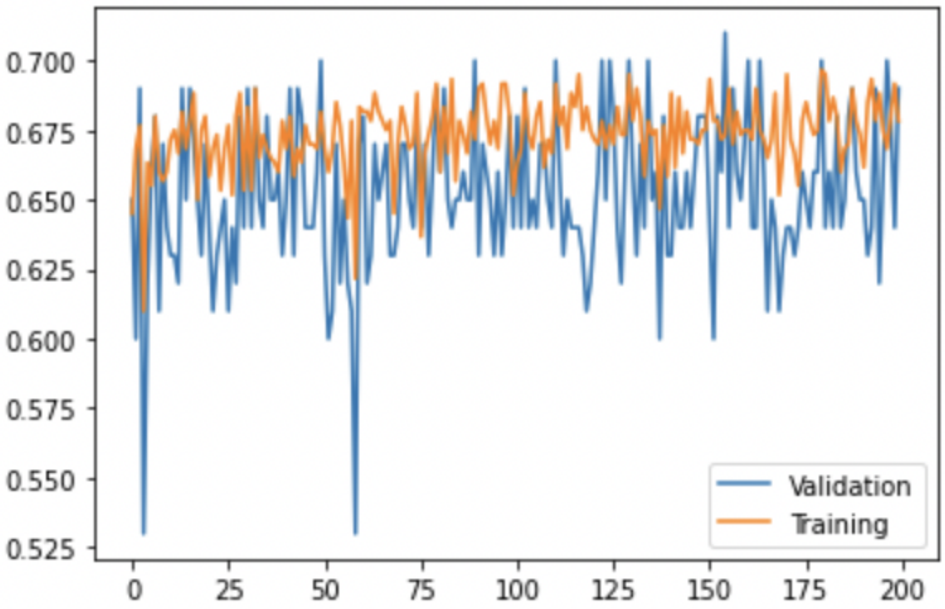
Appendix

Figure 2.1

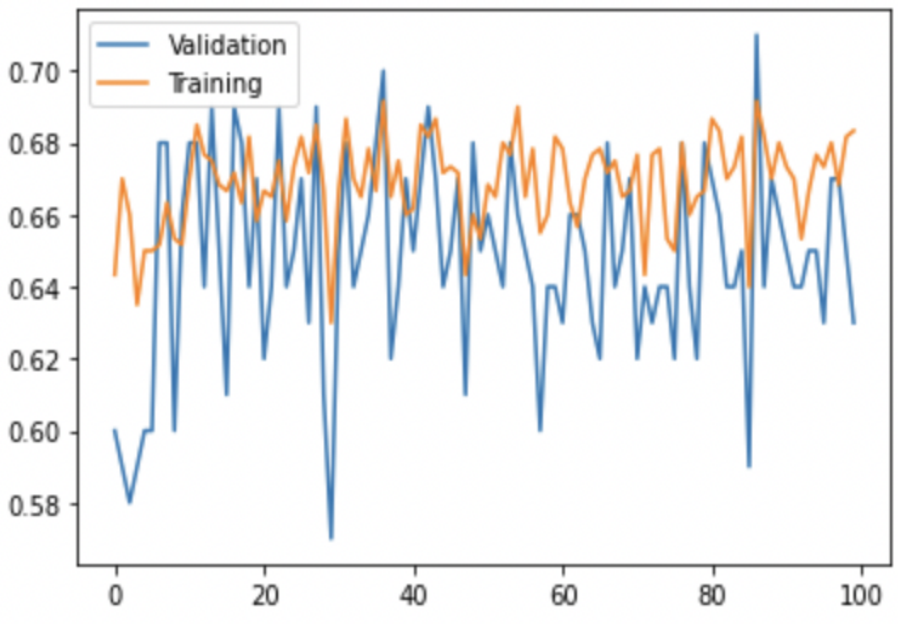
(Size=6)



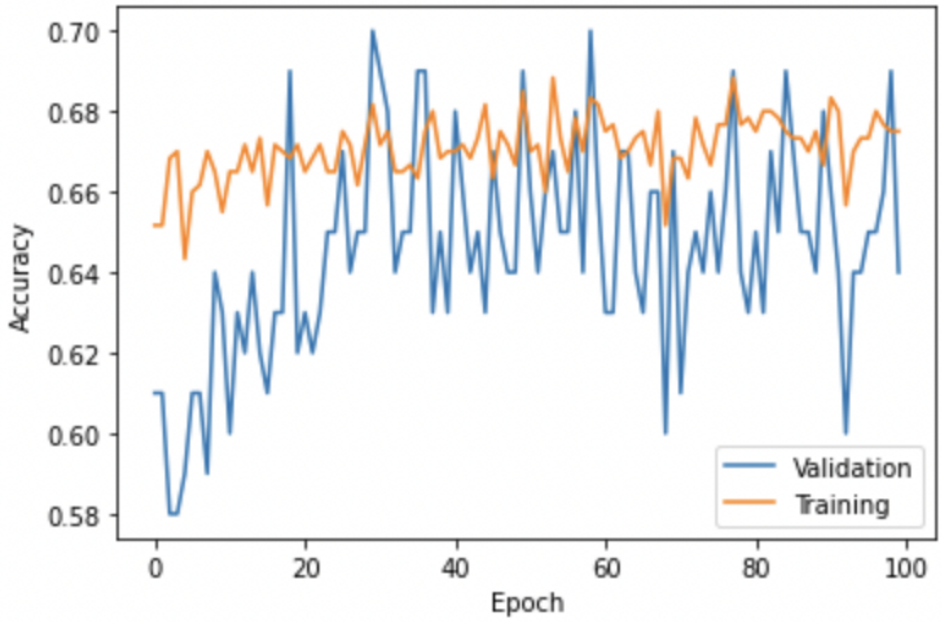
(Size=20)



(Size=30)



(Size=50)



(Size=100)

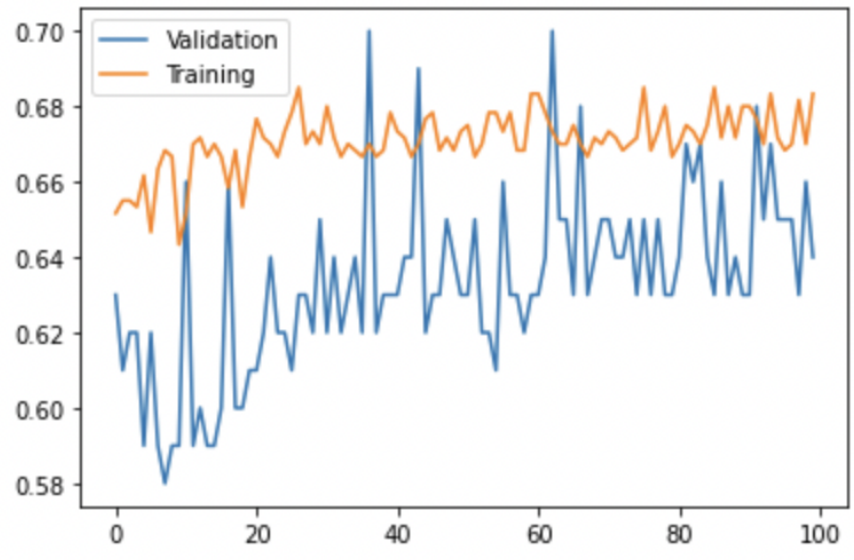


Figure 2.2

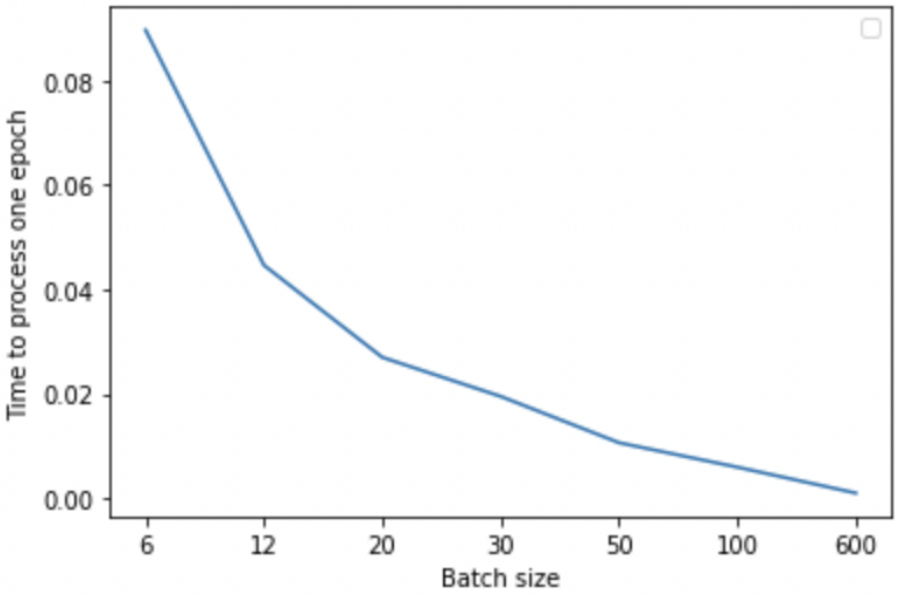


Figure 2.3

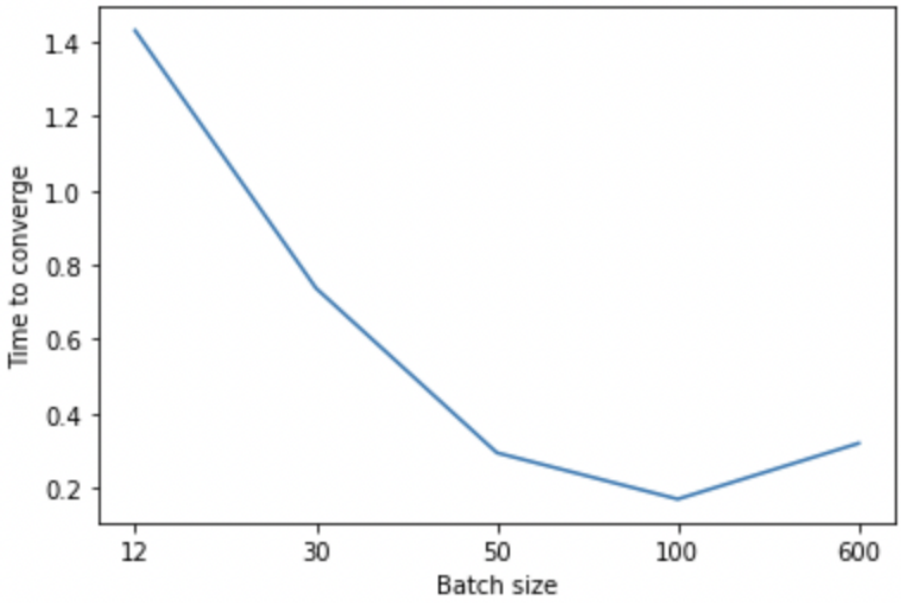


Figure 2.4

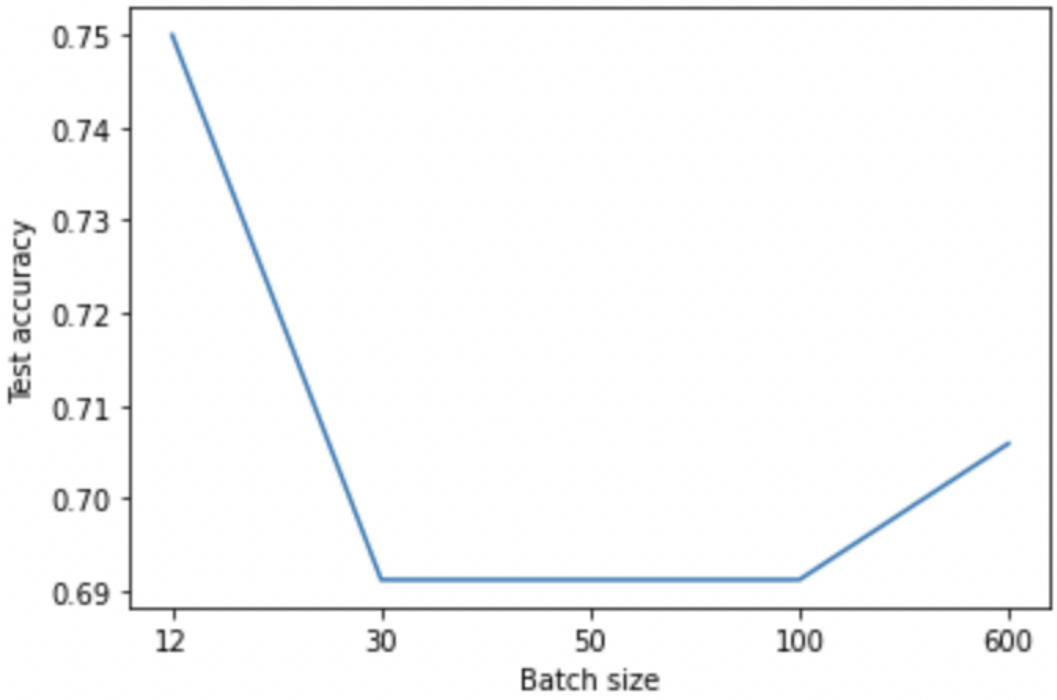


Figure 3.1

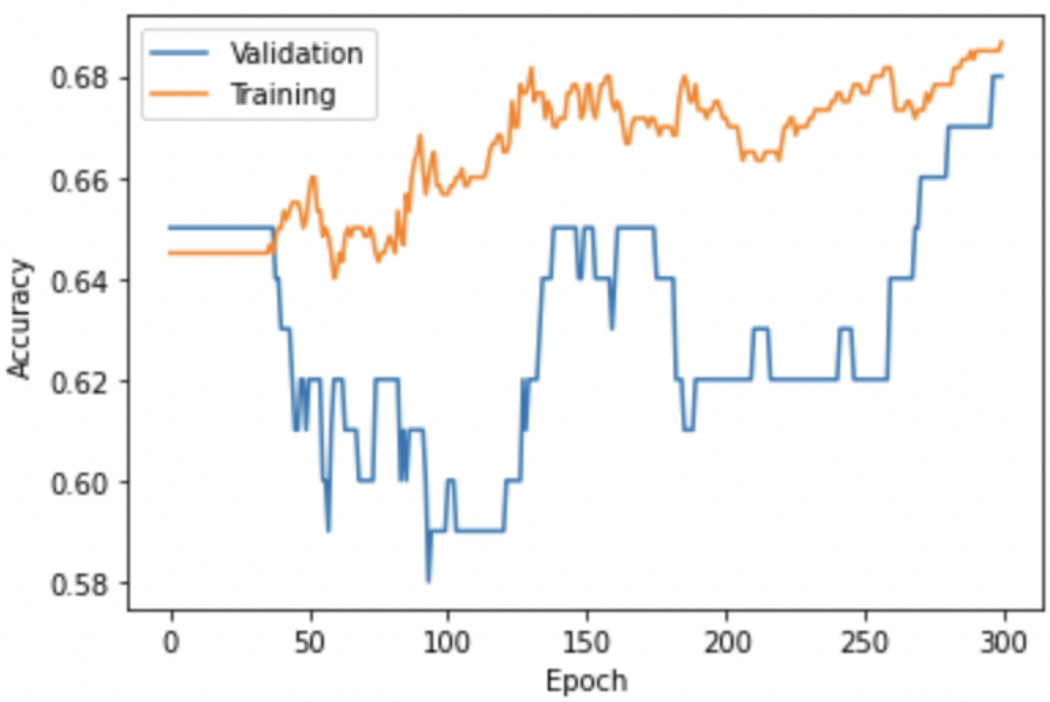


Figure 3.2 (size=6)

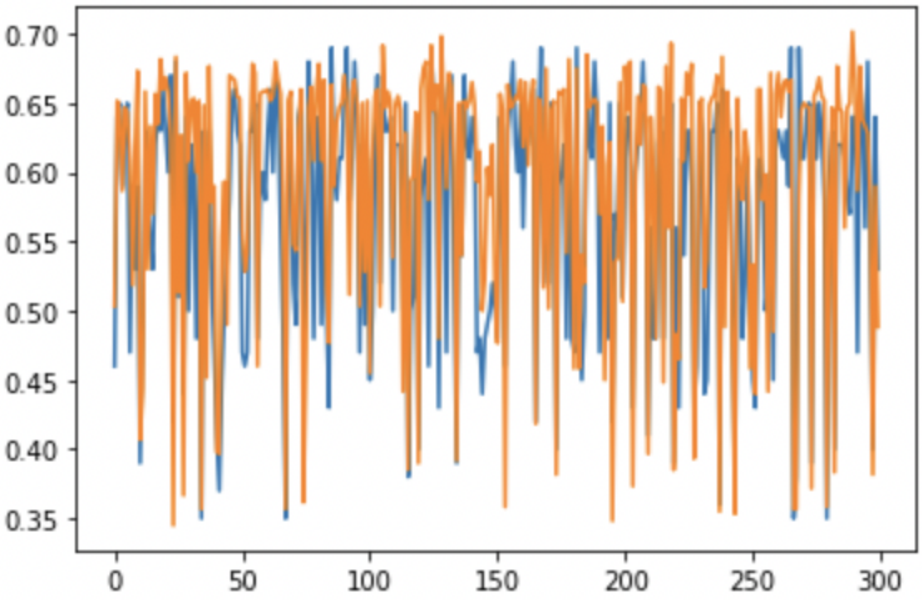


Figure 3.3 (size=100)

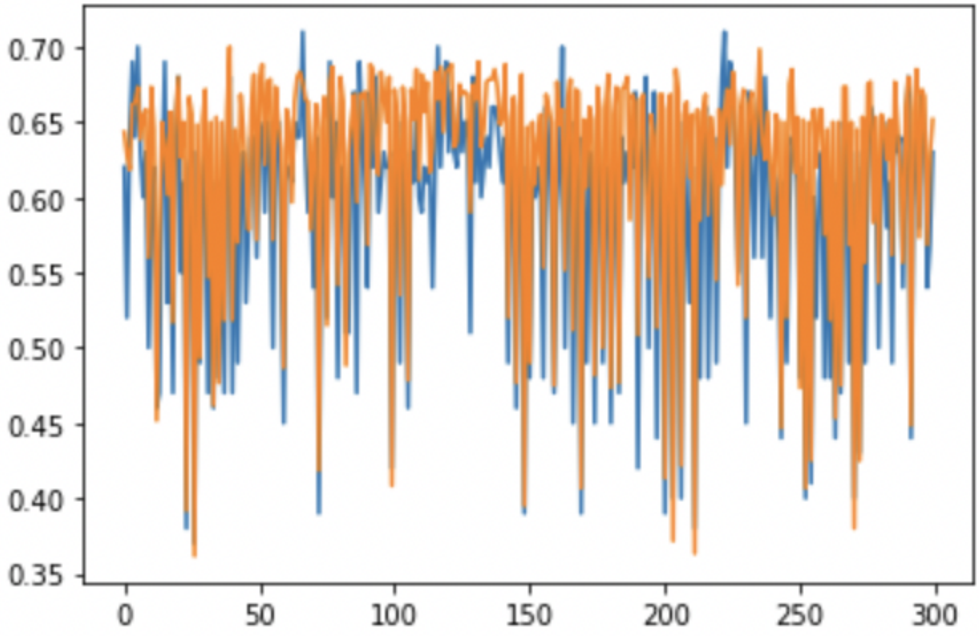


Figure 4