PROJECT REPORT

CSE574
INTRO TO MACHINE LEARNING

PROJECT 2

BY
YASH AHUJA
50245092
yashahuj@buffalo.edu

1. Introduction

This is a machine learning class project. The goal of this project is to perform Linear Regression, Logistic Regression and Neural Networks on a dataset from which the data has been extracted and interpreted from a set of handwritten images.

2. Problem and model description

- *Definition:* Our objective is to compare some handwritten "and" words using machine learning algorithms. The algorithms I used are:
    - Linear Regression using Stochastic Gradient Descent
    - Logistic Regression using Gradient Descent
    - Neural Networks
- *Data sets:* There were two types of data sets provided: *Human Observed* and *GSC data sets*. The two methods used to tackle the partitioning of data used were: *by concatenation* and *by subtraction*. We eventually took 80% of the total data as training sets, 10% as the validation sets and 10% as the test.
- *Hyperparameters:* For linear and logistic regression, the hyperparameters varied were:
    - Regularization term (Lambda)
    - Learning Rate (eta)

  For neural networks, the hyperparameters tuned were:
    - Activation Function
    - Optimizer

For Linear Regression, the ERMS and Accuracy is calculated. For logistic and Neural networks, just the accuracy is calculated.

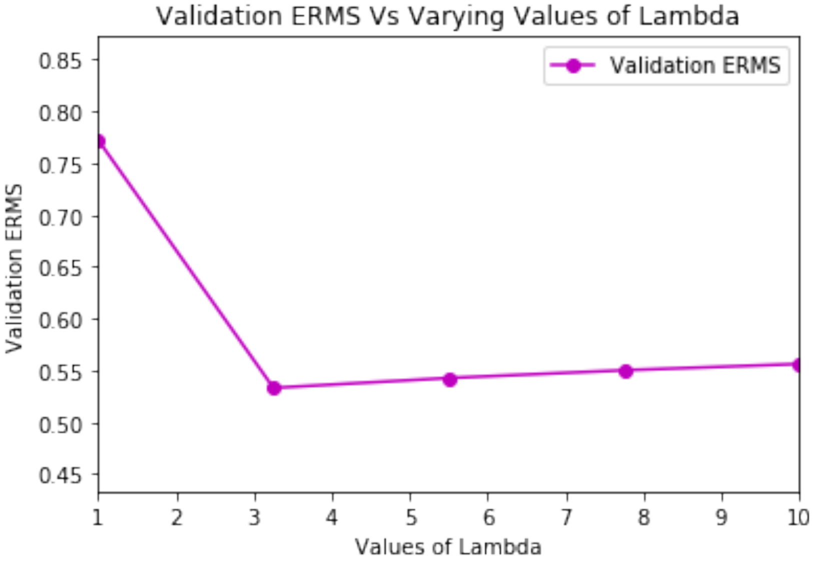
3. Results

PLOTS

LINEAR REGRESSION

(i) Plots and Discussion:

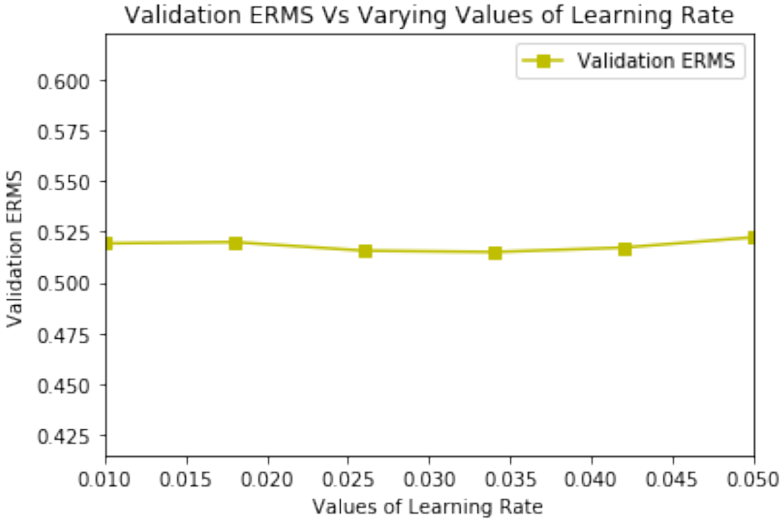
(a1) Human Observed Dataset with feature Concatenation (Varying Lambda i.e., regularization term)



Comments: The hyperparameter values, lambda, are varied in this graph. As observed, the least validation ERMS is obtained at the Lambda between 3 and 4. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **75.0%**. The ERMS Value is **41.68%**.

Note: The regularization term is varied from 1 to 10.

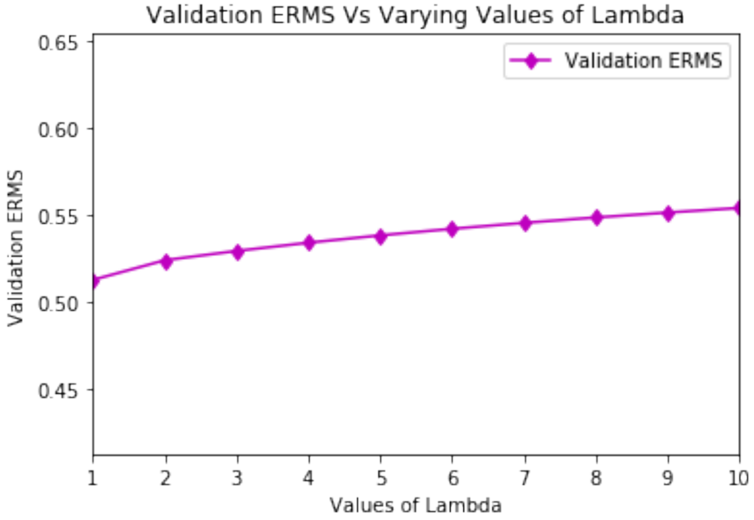
(a2) Human Observed Dataset with feature Concatenation (Varying Learning Rate)



Comments: The hyperparameter values, learning rate, are varied in this graph. As observed, the validation ERMS is almost constant while the learning rate is varied. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **75.0%**. The ERMS Value is **41.68%**.

Note: The Learning rate value is varied from 0.01 to 0.05.

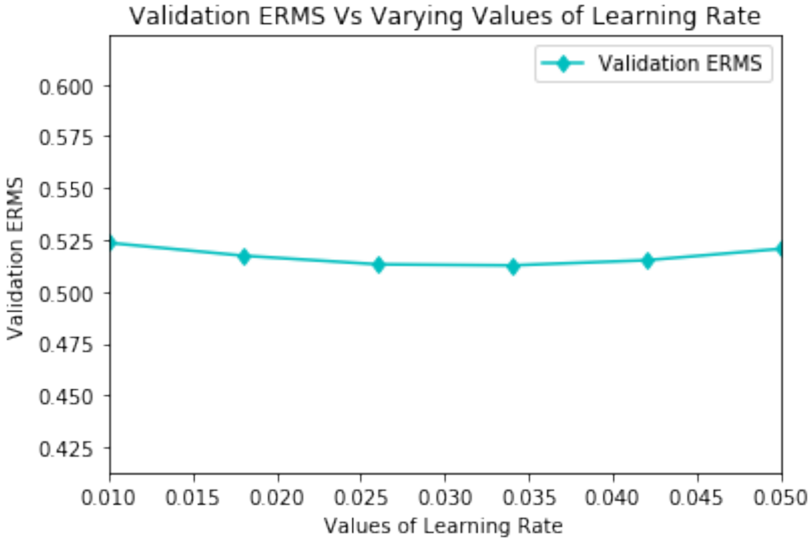
(b1) Human Observed Dataset with feature Subtraction (Varying Lambda i.e., regularization term)



Comments: The hyperparameter values, lambda, are varied in this graph. As observed, the least validation ERMS is obtained at Lambda = 1. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **73.08%**. The ERMS Value is **46.14%**.

Note: The regularization term is varied from 1 to 10.

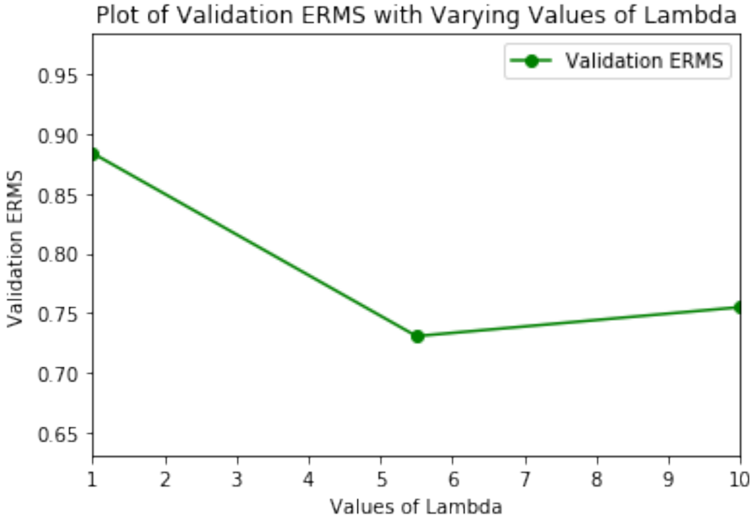
(b2) Human Observed Dataset with feature Subtraction (Varying Learning Rate)



Comments: The hyperparameter values, learning rate, are varied in this graph. As observed, the validation ERMS is almost constant while the learning rate is varied. But an observable low for ERMS is around learning rate = 0.035. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **73.08%**. The ERMS Value is **46.14%**.

Note: The Learning rate value is varied from 0.01 to 0.05.

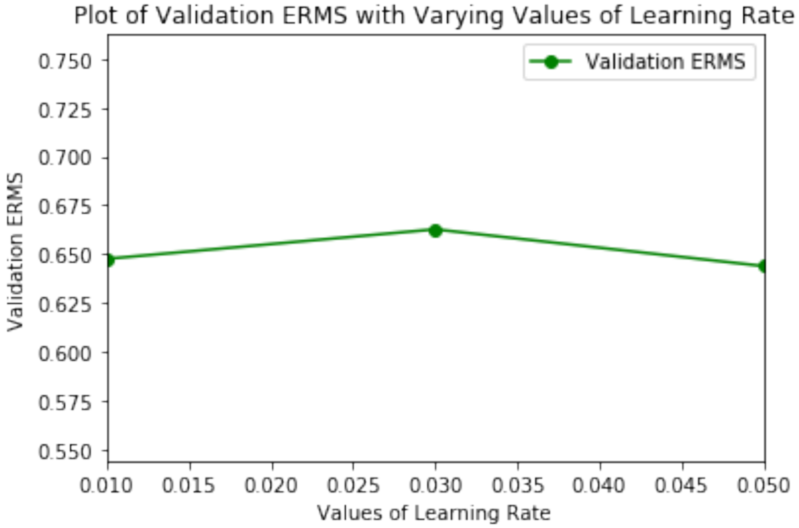
(c1) GSC Observed Dataset with feature Concatenation (Varying Lambda i.e., regularization term)



Comments: The hyperparameter values, lambda, are varied in this graph. As observed, the least validation ERMS is obtained at Lambda 5.5. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **53.34%**. The ERMS Value is **66.08%**.

Note: The regularization term is varied from 1 to 10.

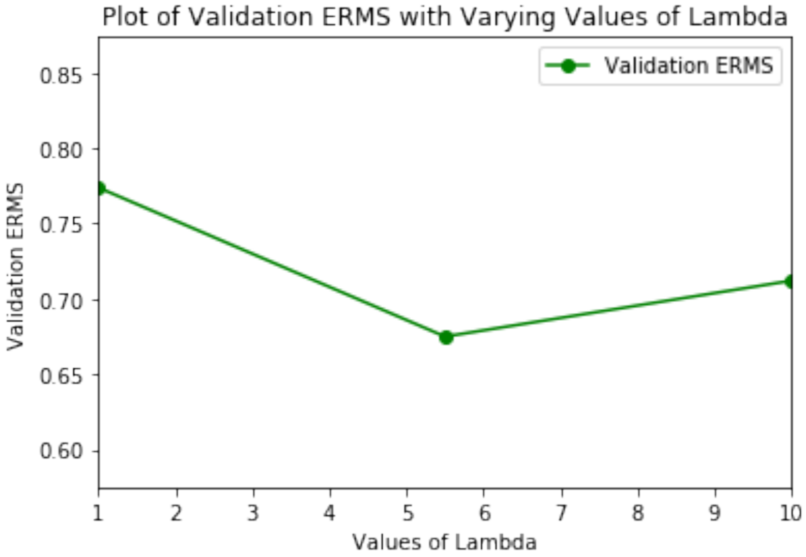
(c2) GSC Observed Dataset with feature Concatenation (Varying Learning Rate)



Comments: The hyperparameter values, learning rate, are varied in this graph. As observed, the least validation ERMS is around learning rate = 0.05. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **63.29%**. The ERMS Value is **51.22%**.

Note: The Learning rate value is varied from 0.01 to 0.05.

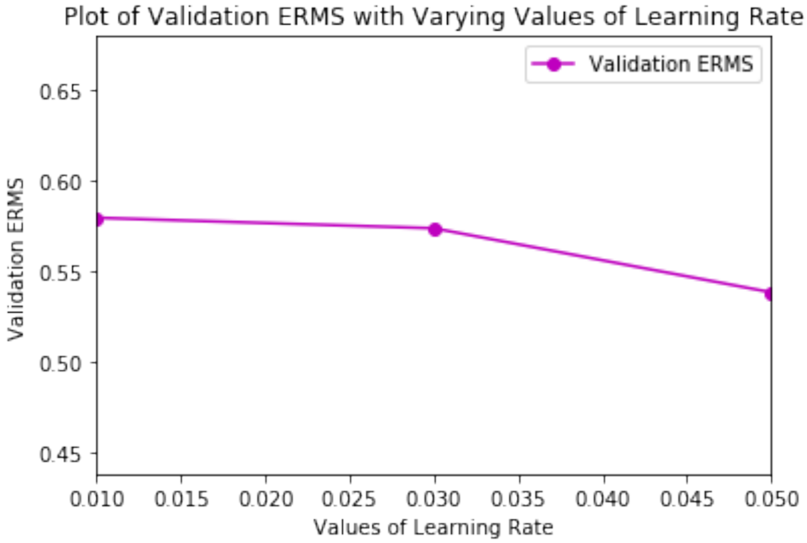
(d1) GSC Observed Dataset with feature Subtraction (Varying Lambda i.e., regularization term)



Comments: The hyperparameter values, lambda, are varied in this graph. As observed, the least validation ERMS is obtained at Lambda 5.5. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **57.35%**. The ERMS Value is **56.27%**.

Note: The regularization term is varied from 1 to 10.

(d2) GSC Observed Dataset with feature Subtraction (Varying Learning Rate)



Comments: The hyperparameter values, learning rate, are varied in this graph. As observed, the least validation ERMS is around learning rate = 0.05. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **68.62%**. The ERMS Value is **45.94%**.

Note: The Learning rate value is varied from 0.01 to 0.05.

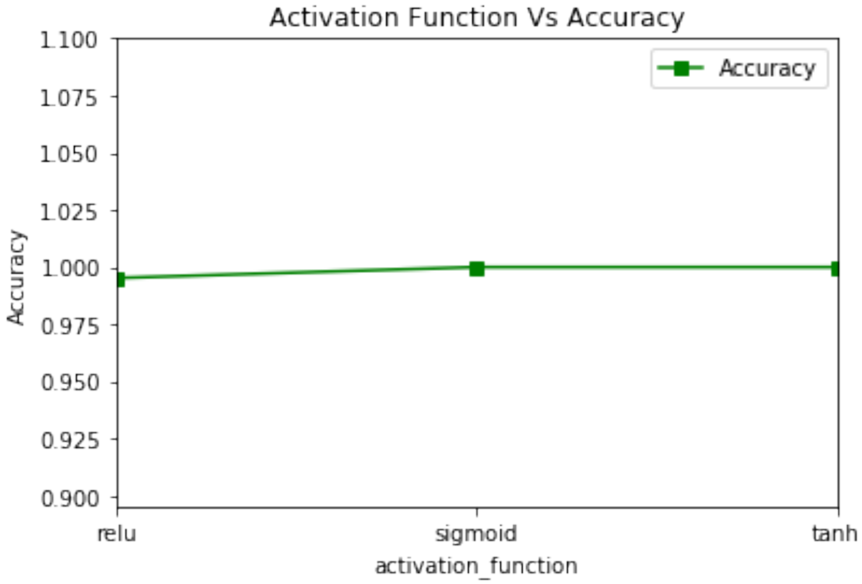
(ii) Conclusion-Linear Regression:

Taking *learning rate* into consideration, changing it does not really change the ERMS or accuracy too much. Using a small learning rate did the model good, as it did not overshoot the optimal value and did not give us a worse accuracy than it already is. On the other hand, the *regularization term* varied depending on the dataset used. The regularization term was high for *concatenation (relatively higher error)* and small for *subtraction*.

NEURAL NETWORKS

(i) Plots and Discussion:

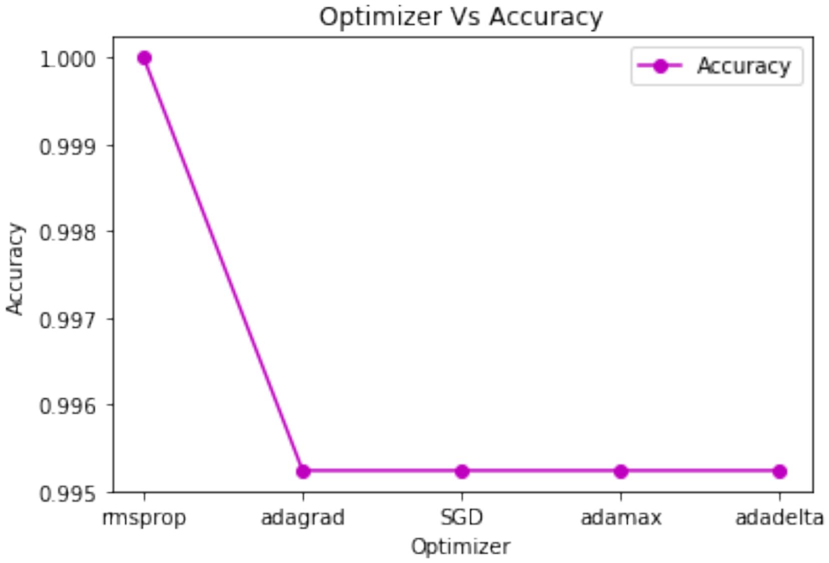
(a1) Human Observed Dataset with feature Concatenation (Varying Activation Function)



Comments: The hyperparameter variables, the activation functions, are varied in this graph. As observed, the accuracy is almost constant for different activation functions. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **100%.**

Note: The activation functions taken into consideration are relu, sigmoid and tanh.

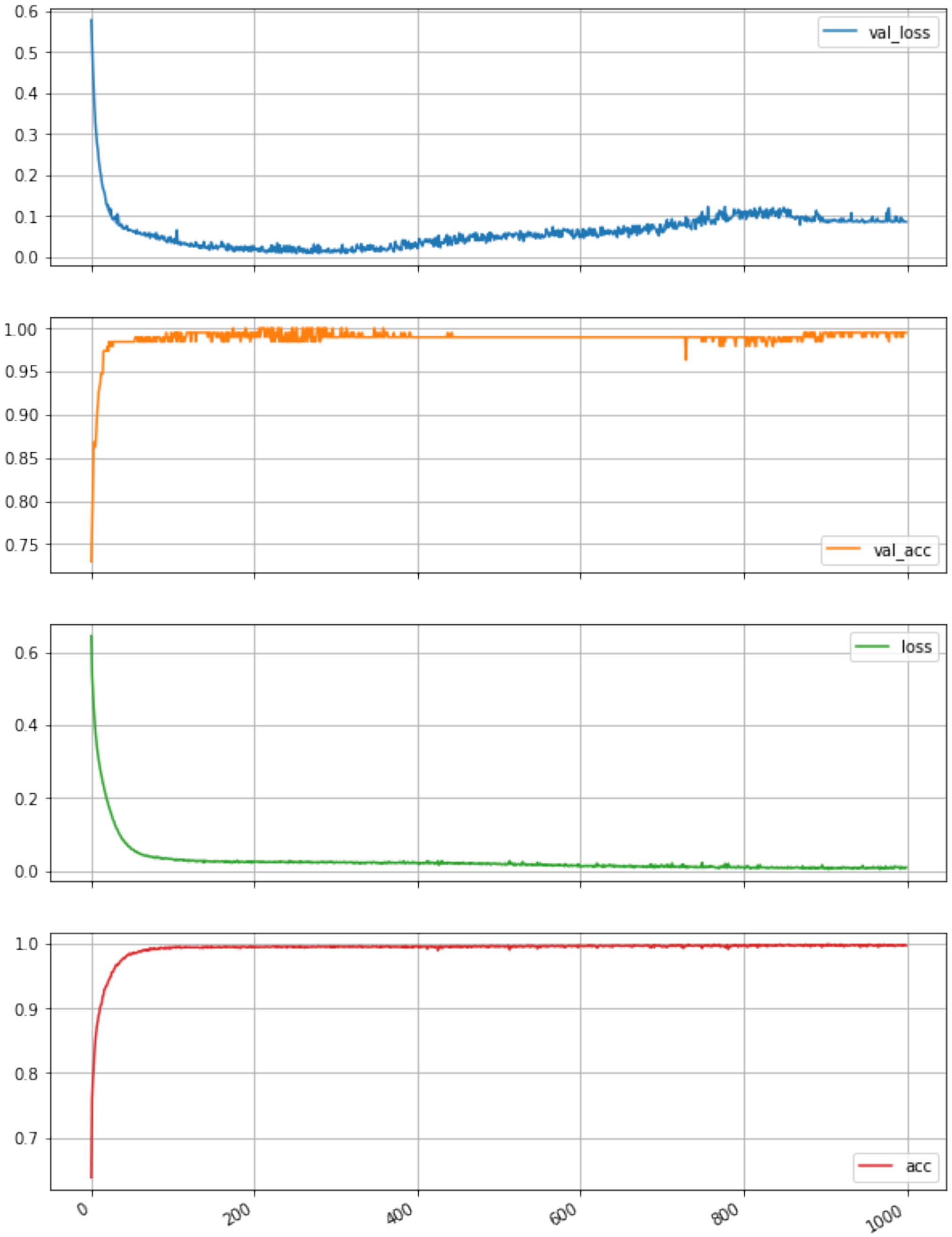
(a2) Human Observed Dataset with feature Concatenation (Varying the Optimizer)



Comments: The hyperparameter variables, optimizers, are varied in this graph. As observed, the accuracyis almost constant for all the optimizers except for ‘rmsprop’. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **100%**.

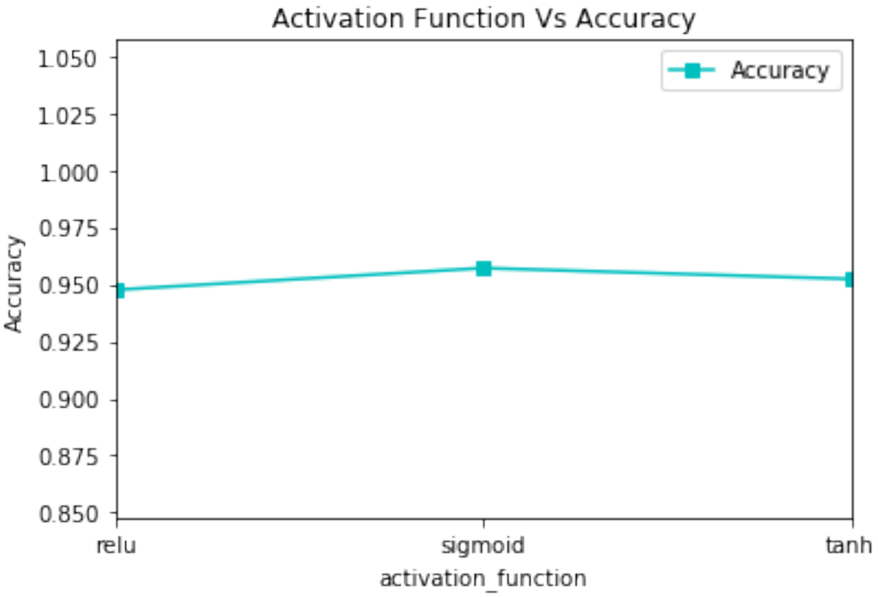
Note: The optimizers taken into consideration are rmsprop, adagrad, SGD, adamax, and adadelta.

The matplotlib graph for human concatenation is:



Observation: The trends seen in these graphs are not varied too much. It shows that the model runs smoothly, there is no noise and that the accuracy is perfect.

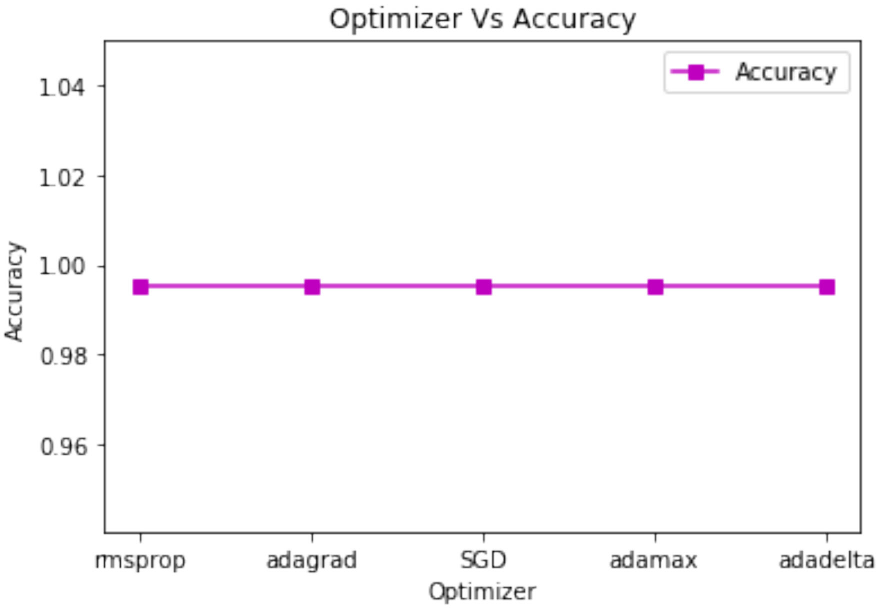
(b1) Human Observed Dataset with feature Subtraction (Varying Activation Function)



Comments: The hyperparameter variables, the activation functions, are varied in this graph. As observed, the accuracy is almost constant for different activation functions. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **95%.**

Note: The activation functions taken into consideration are relu, sigmoid and tanh.

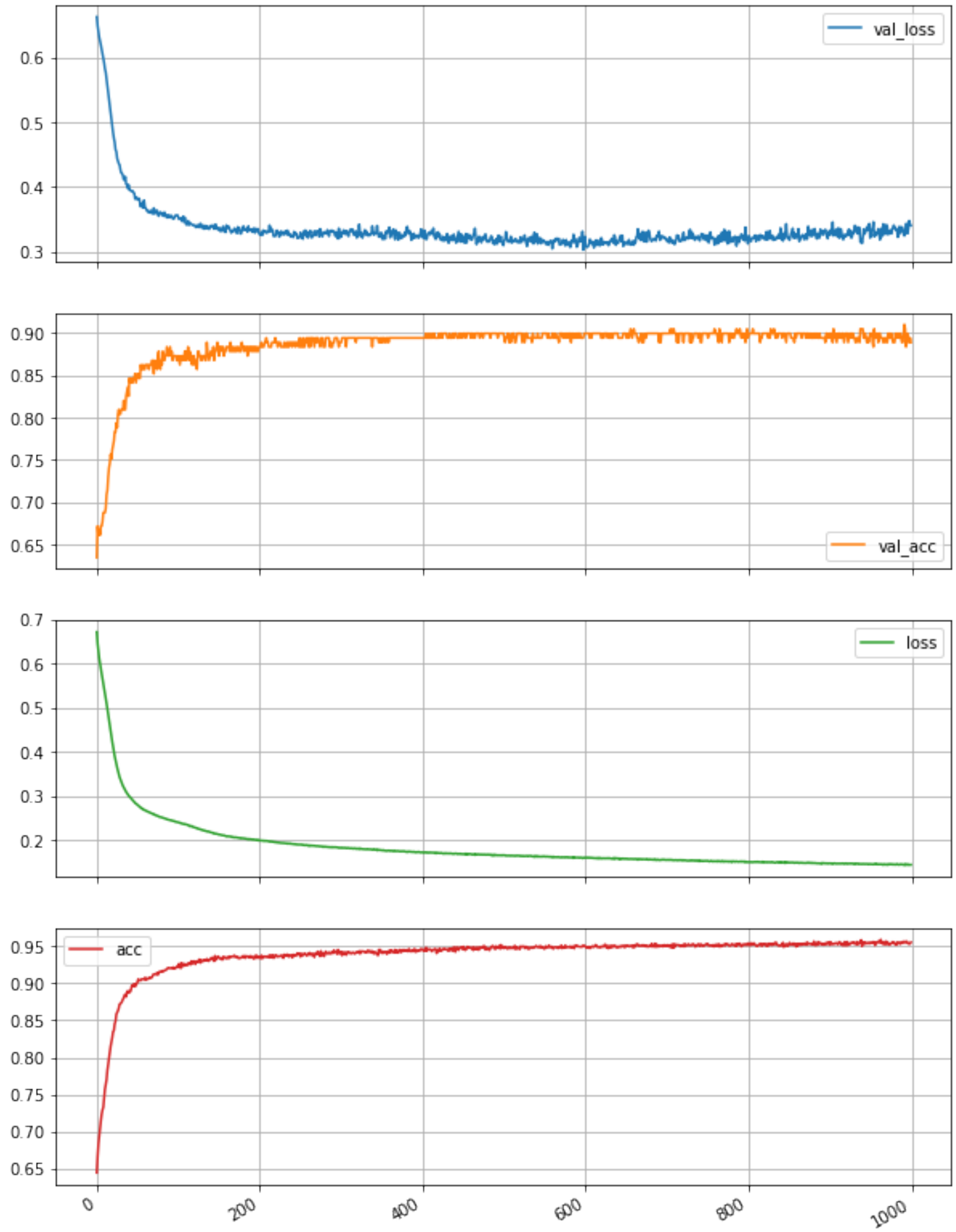
(b2) Human Observed Dataset with feature Subtraction (Varying the Optimizer)



Comments: The hyperparameter variables, optimizers, are varied in this graph. As observed, the accuracy is completely constant. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **95%**.

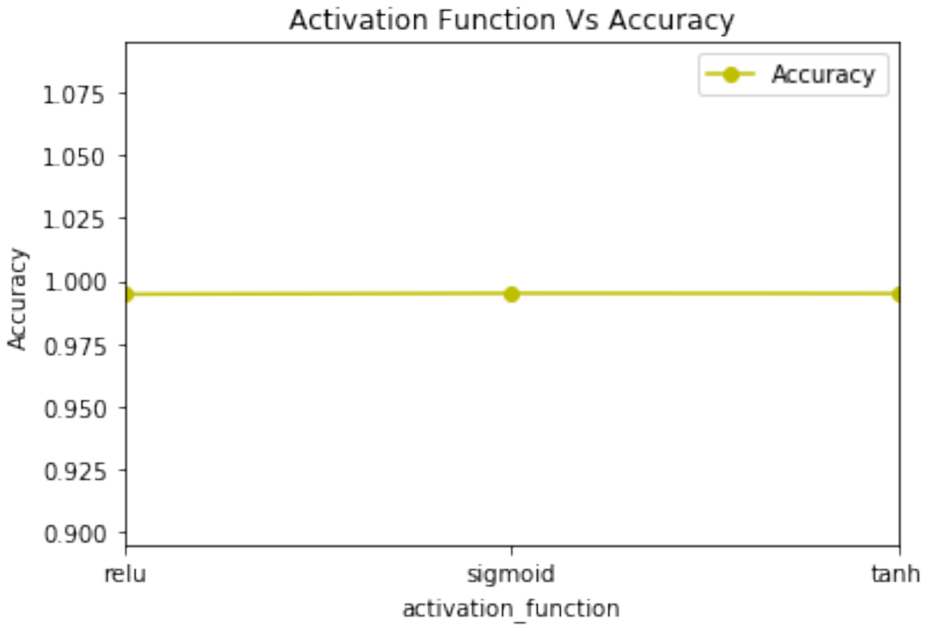
Note: The optimizers taken into consideration are rmsprop, adagrad, SGD, adamax, and adadelta.

The matplotlib graph for human subtraction is:



Observation: Again, the trends seen in these graphs are not varied too much. It shows that the model runs smoothly, there is no noise and that the accuracy is perfect.

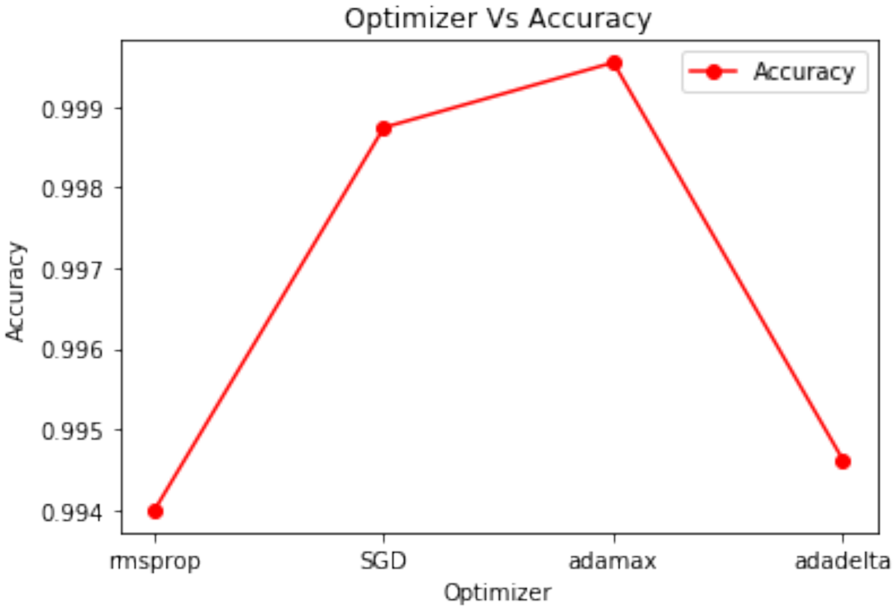
(c1) GSC Observed Dataset with feature Concatenation (Varying Activation Function)



Comments: The hyperparameter variables, the activation functions, are varied in this graph. As observed, the accuracy is constant for different activation functions. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **99%.**

Note: The activation functions taken into consideration are relu, sigmoid and tanh.

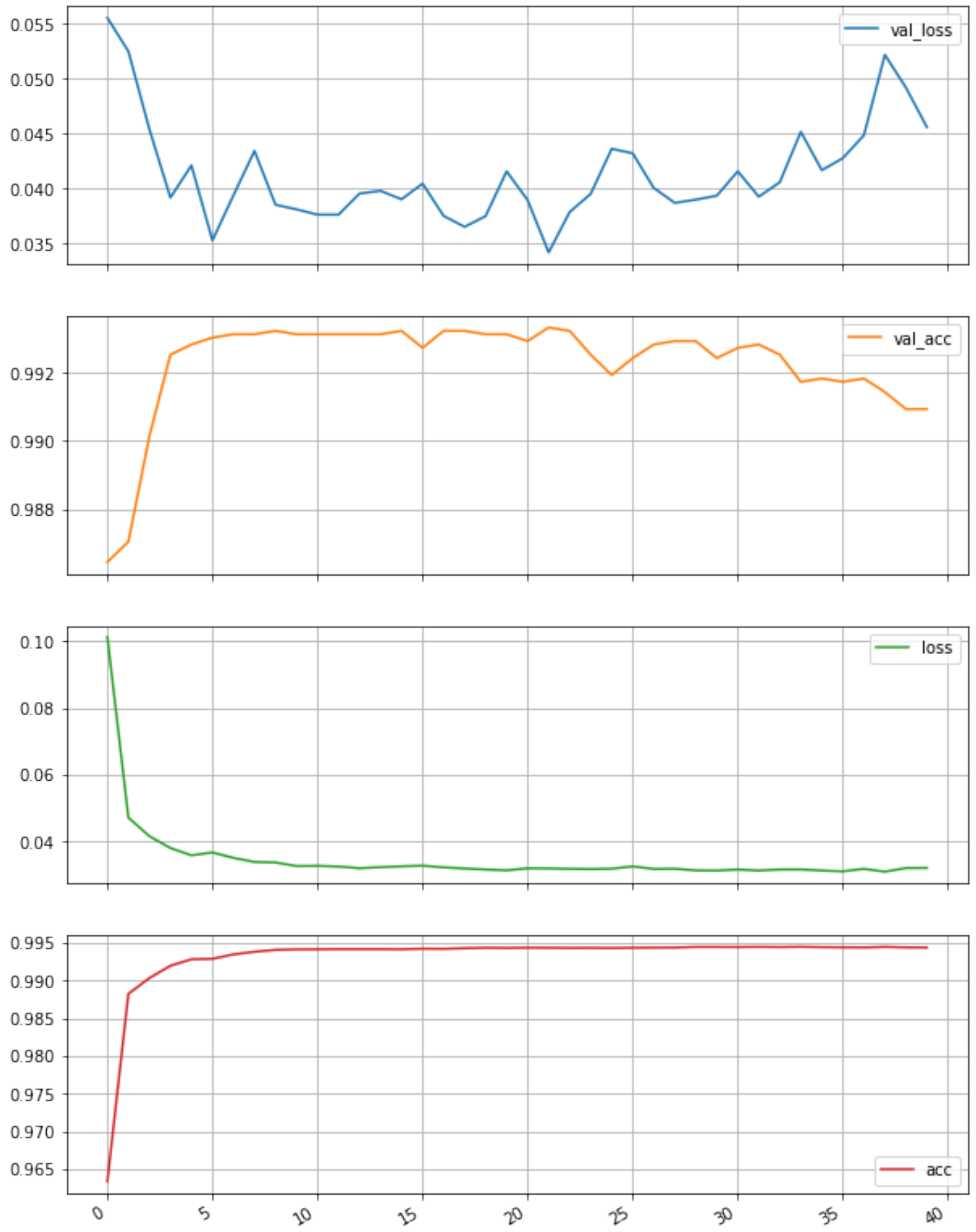
(c2) GSC Observed Dataset with feature Concatenation (Varying the Optimizer)



Comments: The hyperparameter variables, optimizers, are varied in this graph. As observed, the accuracyis almost constant for all the optimizers except for ‘adamax’. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **99%**.

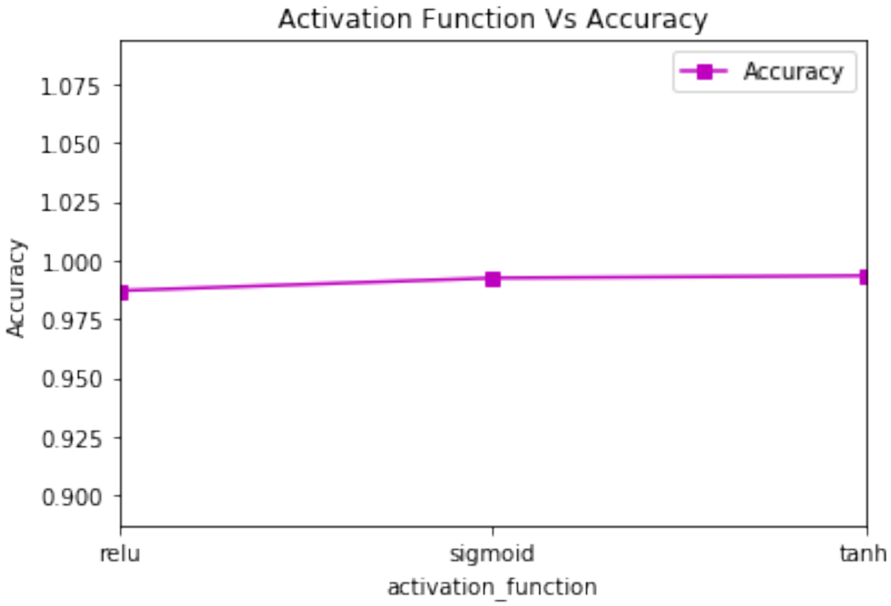
Note: The optimizers taken into consideration are rmsprop, SGD, adamax, and adadelta.

The matplotlib graph for GSC concatenation is:



Observation: The trends seen in these graphs are not varied too much. It shows that the model runs smoothly, there is a bit of noise in the validation accuracy and validation loss, but the resulting testing accuracy is smooth. It is a good fit.

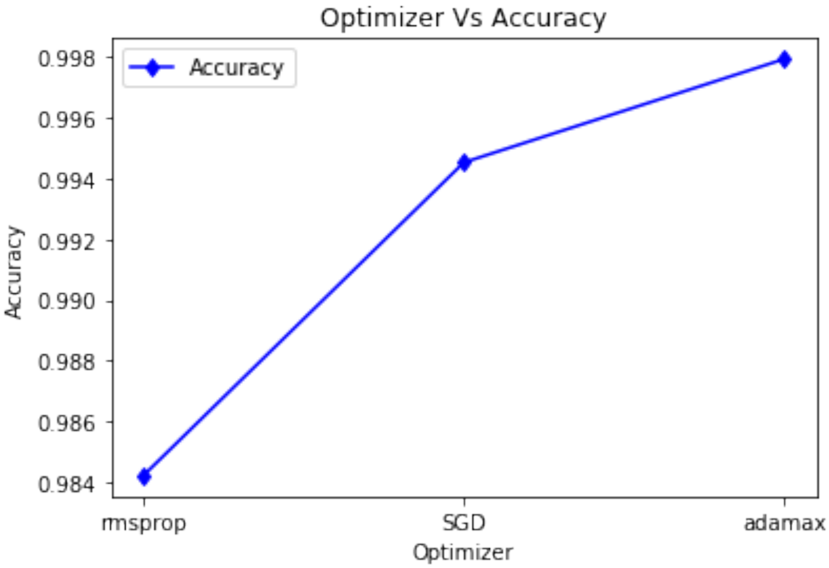
(d1) GSC Observed Dataset with feature Subtraction (Varying Activation Function)



Comments: The hyperparameter variables, the activation functions, are varied in this graph. As observed, the accuracy is almost constant for different activation functions. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **99%.**

Note: The activation functions taken into consideration are relu, sigmoid and tanh.

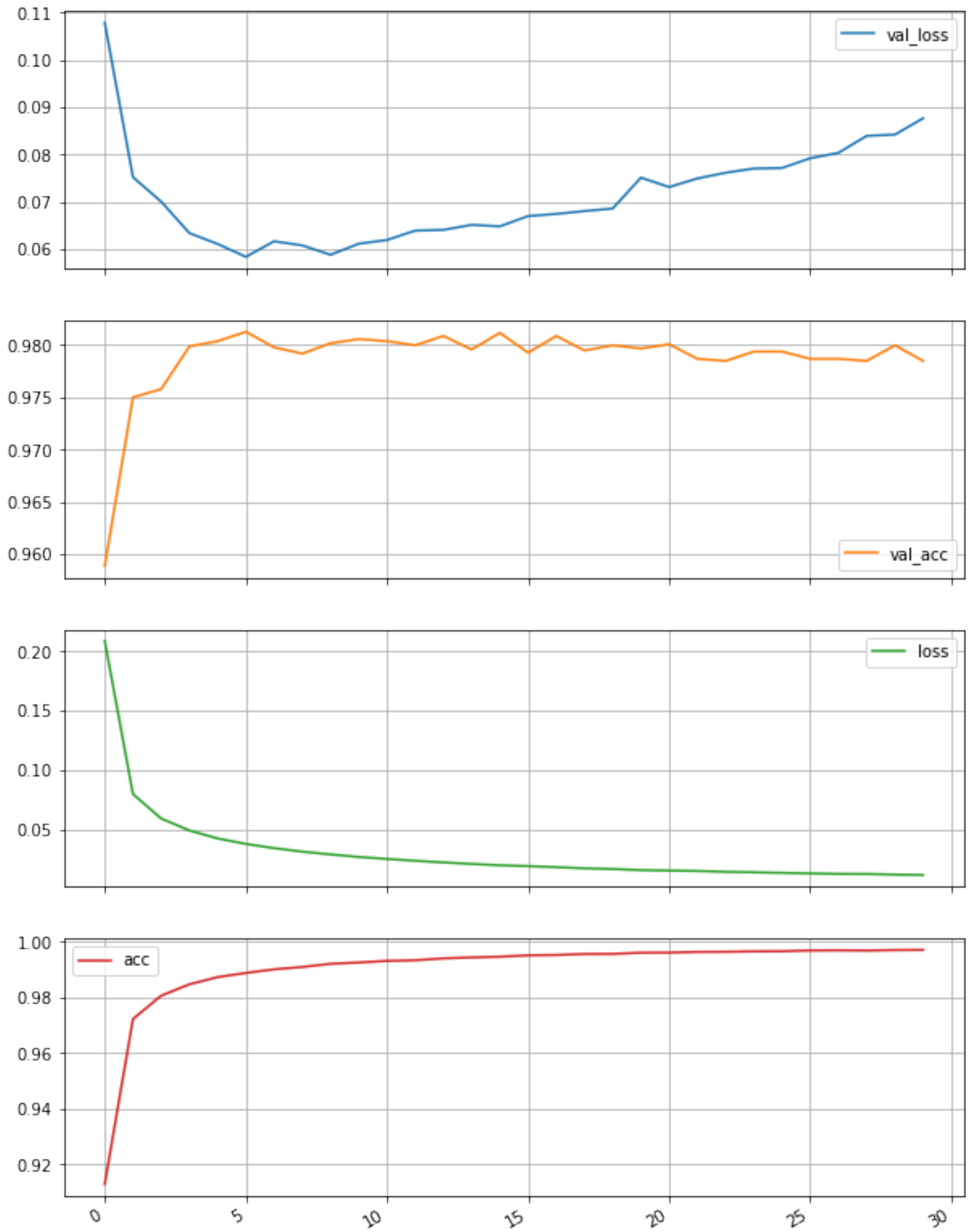
(d2) GSC Observed Dataset with feature Subtraction (Varying the Optimizer)



Comments: The hyperparameter variables, optimizers, are varied in this graph. As observed, the accuracy is completely constant. Hence, the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **100%**.

Note: The optimizers taken into consideration are rmsprop, SGD, and adamax.

The matplotlib graph for GSC subtraction is:



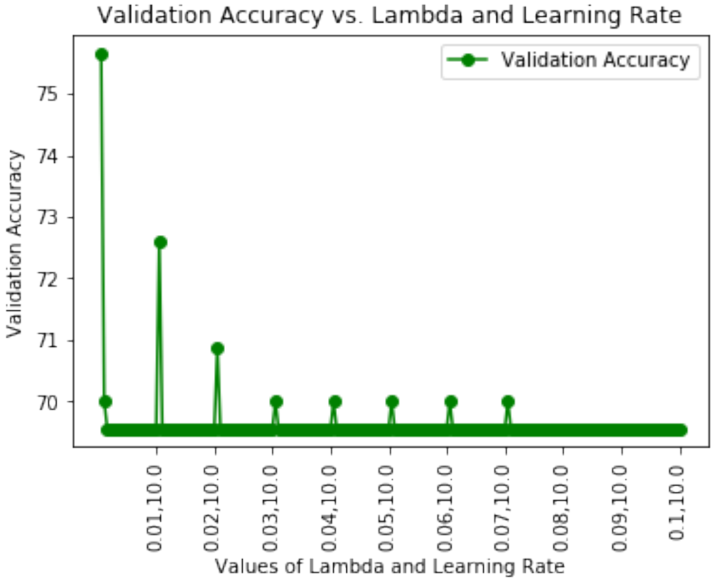
Observation: The trends seen in these graphs are not varied too much. It shows that the model runs smoothly, there is not too much noise in the validation accuracy and validation loss as well, but the resulting testing accuracy is smooth.

It is a good fit.

LOGISTIC REGRESSION

(i) Plots and Discussion:

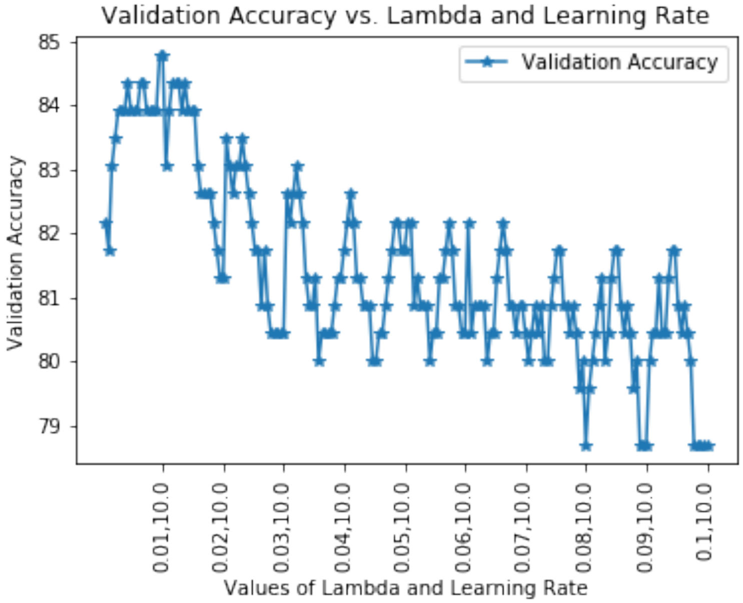
(a) Human Observed Dataset with feature Concatenation



Comments: The learning rate and Lambda values are varied in this graph. As observed, the highest validation accuracy is obtained at the learning rate = 0.05 and Lambda = 10. Hence the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **69.86%**.

Note: There are actually 100 points on the X-axis. There are 10 Lambda points in between each learning rate.

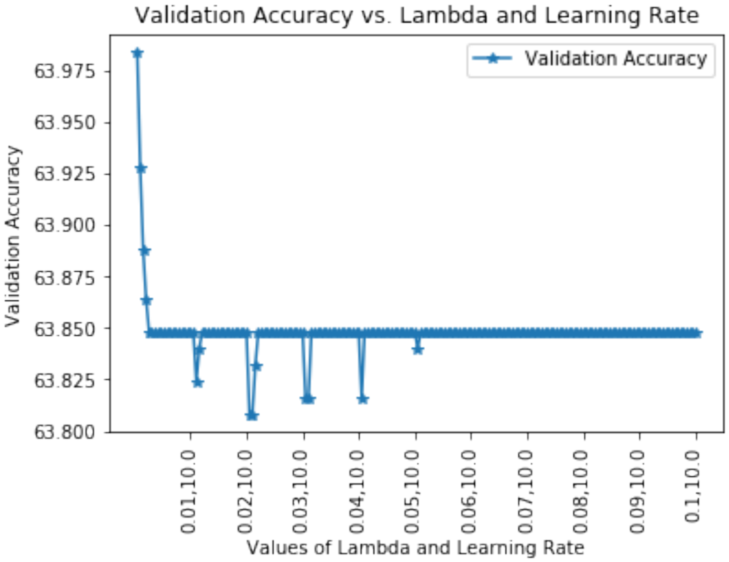
(b) Human Observed Dataset with feature subtraction



Comments: The learning rate and Lambda values are varied in this graph. As observed, the highest validation accuracy is obtained at the learning rate = 0.1 and Lambda = 10. Hence the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **82.09%**.

Note: There are actually 100 points on the X-axis. There are 10 Lambda points in between each learning rate.

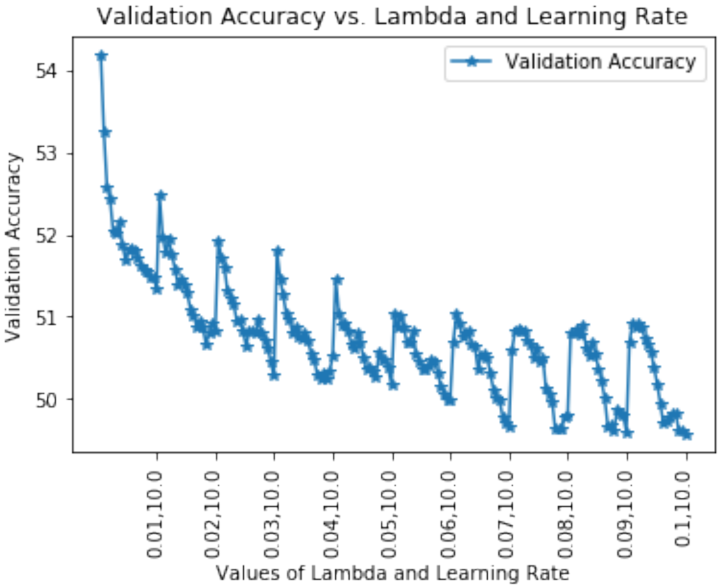
(c) GSC Dataset with feature concatenation



Comments: The learning rate and Lambda values are varied in this graph. As observed, the highest validation accuracy is obtained at the learning rate = 0.03 and Lambda = 1.5. Hence the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **63.84%**.

Note: There are actually 100 points on the X-axis. There are 10 Lambda points in between each learning rate.

(d) GSC Dataset with feature subtraction



Comments: The learning rate and Lambda values are varied in this graph. As observed, the highest validation accuracy is obtained at the learning rate = 0.1 and Lambda = 10. Hence the testing accuracy is obtained for those values of the hyperparameters. The resulting testing accuracy is found to be **58.18%**.

Note: There are actually 100 points on the X-axis. There are 10 Lambda points in between each learning rate.

4. Conclusion

After tuning the hyperparameters, and applying the best ones to the testing dataset, it is found that Neural Networks is giving the best accuracy. Linear regression gave the least best accuracy followed by Logisitc (gave better results, and the best ones by NN.