CSE 574 – Introduction to Machine Learning

Project 2.0 – Handwriting Recognition

Asish Kakumanu
UB Person No :50288695
asishkak@buffalo.edu

Abstract

The Abstract of the project is to develop machine learning models to solve a case involving handwritten documents. Our task here is to predict whether the written documents belong to a same writer or a different writer. To predict accordingly, we develop regression, classification models, neural network and calculate the errors accordingly and use them to improve the model further.

1 Problem Description

The problem of comparing AND handwritten images is to be solved using three different approaches.

1. **Linear Regression** which gives continuous values
2. **Logistic Regression** provides discrete values
3. **Neural Networks** which solves the problem by mapping inputs to output values.

Input: The input is the set of features from two different images.

Output: Either 0 or 1, 0 if the two images are from a different writer and 1 if both the images are from the same writer.

Process: We compare the features from both the images and predict whether they are from same writer or different writer.

2 Overview of Dataset

The datasets provided are of two types, based on the feature extraction process.

- Human Observed Dataset
- Gradient Structural Concavity (GSC) Dataset

2.1 Human Observed Dataset

Each image ids has 9 features. According to the image below. Each image has a name, for instance 1121a_num1 which is in the format **XXXXy_numZ** where **XXXX** is the **writer number**, **y** being the **page number** from where the sample is from and **Z** being the **sample number**.

Figure 2: Human Observed Dataset Example

img_id_A	img_id_B	f_{A1}	f_{A2}	f_{A3}	f_{A4}	f_{A5}	f_{A6}	f_{A7}	f_{A8}	f_{A9}	f_{B1}	f_{B2}	f_{B3}	f_{B4}	f_{B5}	f_{B6}	f_{B7}	f_{B8}	f_{B9}	t
1121a_num1	1121b_num2	2	1	1	3	2	2	0	1	2	2	1	1	0	2	2	0	3	2	1
1121a_num1	1386b_num1	2	1	1	3	2	2	0	1	2	3	1	1	0	2	2	0	1	2	0

2.2 GSC Observed Dataset

Each image id, 512 features and the format of the image name is the same as above.

img_id_A	img_id_B	f_{A1}	f_{A2}	f_{A3}	f_{A4}	f_{A5}	f_{A6}	. . .	f_{A512}	f_{B1}	f_{B2}	f_{B3}	f_{B4}	f_{B5}	f_{B6}	. . .	f_{B512}	t
1121a_num1	1121b_num2	0	1	1	0	1	0	. . .	0	0	1	1	0	0	1	. . .	1	1
1121a_num1	1386b_num1	0	1	1	0	1	0	. . .	0	1	1	1	0	1	0	. . .	0	0

3 Processing the Datasets

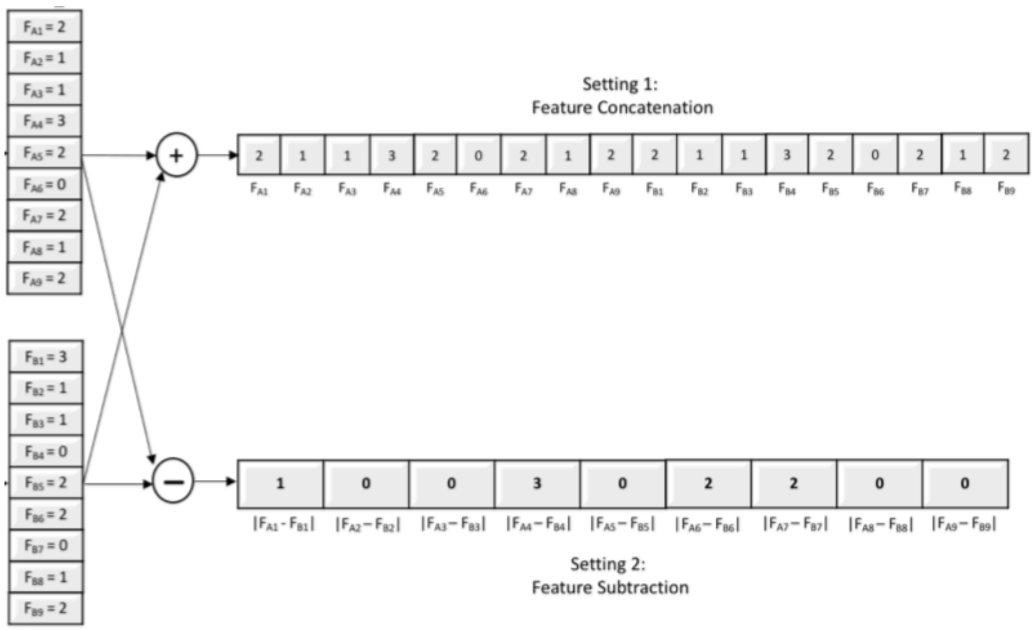
3.1 Human Observed Dataset

3.1.2 Concatenation of Human Observed Dataset

In this step, we prepare the dataset from human observed Dataset. We get the features for both the image ids in the same pairs and concatenate the features and make a dataset with **18** features and also add the corresponding target which is 1 for all the same pairs.

3.1.2 Subtraction of Human Observed Dataset

In this step, we prepare the dataset from human observed Dataset. We get the features for both the image ids in the same pairs and subtract the features and make a dataset with **9** features and also add the corresponding target which is 0 for all the different pairs.



3.2 GSC Observed Dataset

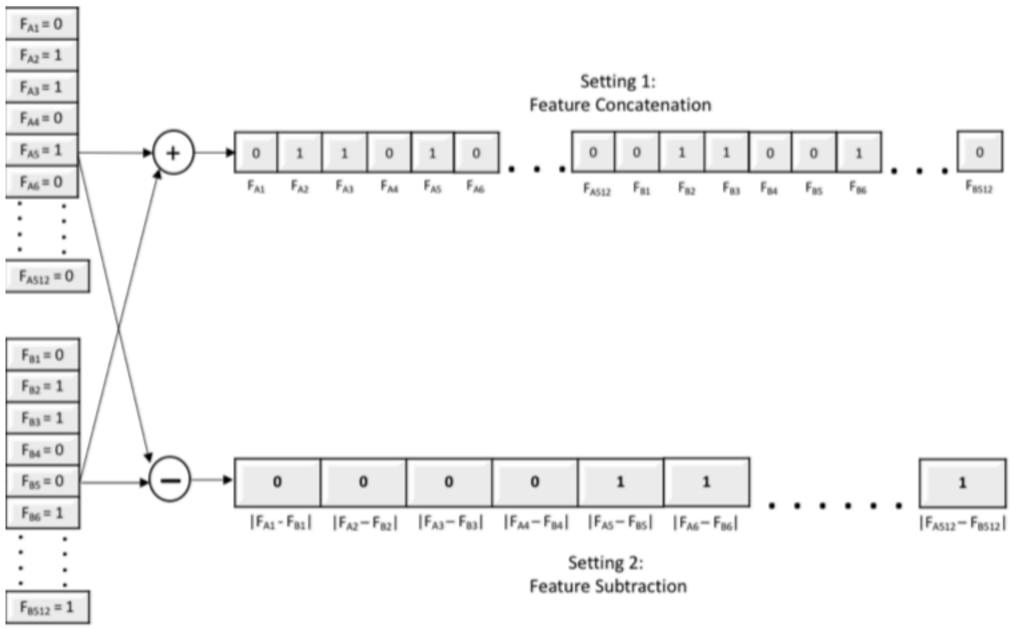
3.2.1 Concatenation of GSC Observed Dataset

In this step, we prepare the dataset from GSC observed Dataset. We get the features for both the image ids in the same pairs (same_pairs.csv) and concatenate the features and make a dataset with a total of 1024 features, 512 features of each image and also add the corresponding target which is 1 for all the same pairs. We use concat, merge methods from panda's library to do the concatenation of GSC Observed Data.

3.2.2 Subtraction of GSC Observed Dataset

In this step, we prepare the dataset from GSC observed Dataset. We get the features for both the image ids in the different pairs and subtract the features and make a dataset with **absolute** values of **512** features and also add the corresponding target which is 0 for all the different pairs. We use *sub (subtract)*, *merge* methods from panda's library to do the subtract of GSC Observed Data

Both Subtraction and Concatenation of the Datasets are clearly demonstrated in the picture below.



3.3 Combining Datasets

We finally make 4 Datasets, One dataset for concatenation of human observed features (Same pairs and Different pairs), One for subtraction of human observed features (Same and Different pairs). Similarly, 2 datasets for GSC observed features, one with concatenation of features consisting of both same and different pairs and the other with Subtraction of features consists of Same and Different Pairs.

3.4 Splitting Datasets

Finally, the datasets are now split into three different parts for training, testing and validation of the model.

Training Dataset is 80% of the datasets.
Testing Dataset is 10% of the datasets.
Validation Dataset is 10% of the datasets.

4 Linear Model

Our Linear Model function $y(x, w)$ has the form:

$$y(x, w) = w^T \phi(x)$$

W is the weight Vector learnt from training samples
ϕ is vector of M basis functions.

We consider $\phi_{0(x)} = 1$ to become a bias in the system. This parameter counterweighs for the difference in the avg. values of target vector in the training data with avg. of the basis function values.

The Closed form solution for linear regression is carried out using Gaussian basis Function which is given by:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right)$$

Where μ_j is the center of basis function and $\sum_j -$ decides the spread of the basis function.

4.1 Stochastic Gradient Descent Solution

We use gradient descent method to compute the weights to minimize the lowest mean error. Here, η is carefully chosen such that we don't observe any variation in convergence. Instead we can reduce the risk in the local minima by selecting larger learning rate in the beginning and reducing proportional to the time. Now, weight is calculated using the equation below:

$$w^{\tau+1} = w^{\tau} + \eta\left(t_n - w^{(\tau)\left({}^{T}\phi_n\right)}\right)\phi_n$$

4.2 Hyper-Parameters

Hyper-parameters are the one which have more effect on the model. By modifying these values, the performance of the model is improved. Hyper-parameters used here are number of basis functions(M), learning rate(η), regularization factor(λ), epochs.

Epochs is a hyper-parameter also known as number of iterations. After certain number of epochs, if the value of weights to be updated do not show much variation then we stop iterating through the given data. This point is known as early stopping rate.

Basis functions(M) are determined with the help of k-means clustering. By using k-means the given data is first divided into clusters and then the cluster centers are fitted with basis functions. this number depends on size of data, there are no fixed numbers for finding the correct value, it is adjusted by finding the error value and changing the numbers.

Regularization helps to solve the problem of overfitting; it just removes unwanted data by using a factor called regularization factor(λ). When we have a large dataset, all the data may not be needed for the model. In such cases this factor is included in the function and is used to update the parameters. Increasing the value to a certain extent improves the performance. High increase may decrease the accuracy of the model. Update weights with various value and see which predicts the best.

Learning rate(η) is a parameter that controls how much weights we are adjusting with the help of gradient. Very high and very low learning rates leads to the problem of overfitting or underfitting. The correct value can be determined by computing the error for each value of η.

4.3 Evaluation

With regularized weight obtained from the above equation we can calculate the sum of squared errors, defined as

$$E_{rms} = \sqrt{\frac{2E(w*)}{N_V}}$$

5 Logistic Regression

It is a **classification** algorithm that gives output as discrete set of classes. It transforms the output using sigmoid function to return a value that can be mapped to two or more classes. In our problem logistic regression outputs, a value of 0 if image pairs are from different writers and outputs a value of 1 if they belong to same writer. We use sigmoid function and limit the values between 0 and 1.

Sigmoid Equation

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$Cross\ Entropy = -(ylog(y') + (1 - y)\log(1 - y'))$$

5.1 Hyper-Parameters

Hyper-parameters are the one which have more effect on the model. By modifying these values, the performance of the model is improved. Hyper-parameters used here are number of basis functions(M), learning rate(η), regularization factor(λ), epochs.

Epochs is a hyper-parameter also known as number of iterations. After certain number of epochs, if the value of weights to be updated do not show much variation then we stop iterating through the given data. This point is known as early stopping rate.

Learning rate(η) is a parameter that controls how much weights we are adjusting with the help of gradient. Very high and very low learning rates leads to the problem of overfitting or underfitting. The correct value can be determined by computing the error for each value of η.

6 Neural Networks

Neural Networks are class of machine learning algorithms that use multiple hidden layers and activation functions for classification of data. It takes an input passes it through hidden neurons in the multiple layers and produces an output that represents the total input of all neurons. For Human Concatenation we take input size as 18. For Human subtraction, we take input size as 9. Whereas, 512 for GSC subtracted dataset and 1024 for GSC concatenation dataset.

Activation functions: Now after computing the function, we apply activation functions which introduces some non-linear properties to our network which computes and learns any function. If we do not use these functions the output would be a simple linear function which are limited and less complex and have low performance rate.

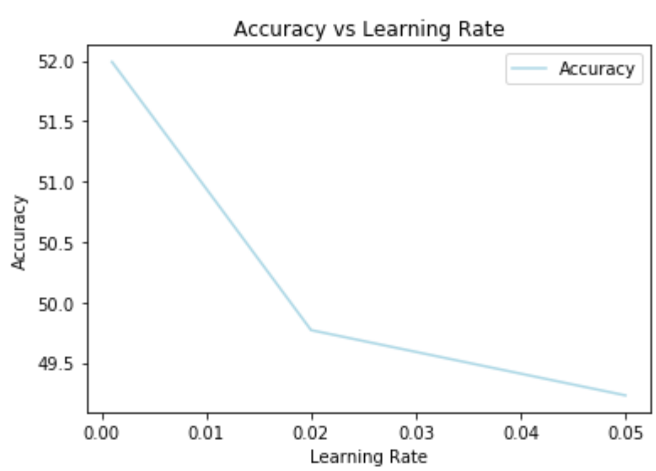
7 Experiments

7.1 Linear Regression

7.1.1 Human Observed Dataset: Concatenation

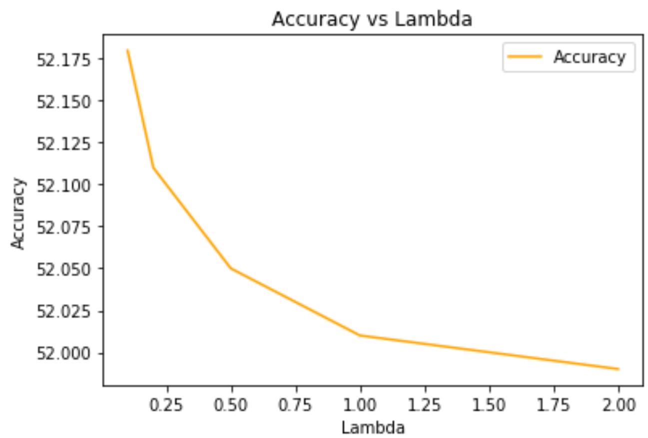
While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

M (Basis Functions)	Learning Rate	Lambda	Train Acc	Val acc	Test Acc
10	0.05	2	51.25	42.01	49.23
10	0.02	2	51.34	42.05	49.77
10	0.01	2	51.5625	42.13	50.94
10	0.001	2	52.83	44.178	51.99



While changing Lambda, the observations are as follows.

M (Basis Functions)	Learning Rate	Lambda	Train Acc	Val acc	Test Acc
10	0.001	0.1	53.17	44.57	52.18
10	0.001	0.2	53.16	44.41	52.11
10	0.001	0.5	53.11	44.29	52.05
10	0.001	1	52.91	44.22	52.01
10	0.001	2	52.83	44.178	51.99



7.1.2 GSC Observed Dataset: Concatenation

While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

M (Basis Functions)	Learning Rate	Test Acc
10	0.05	58.55
10	0.02	58.64
10	0.01	58.8625
10	0.001	60.13

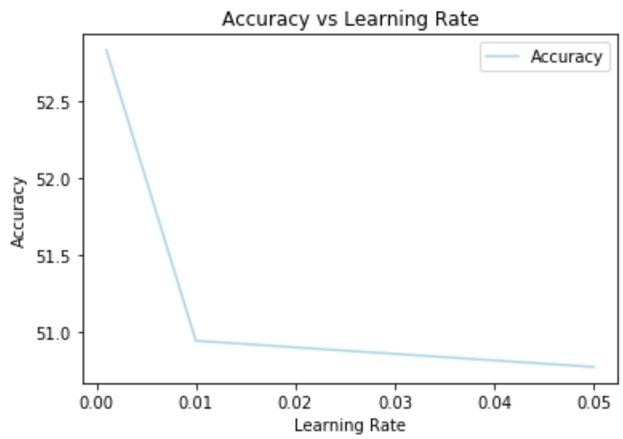
While changing Lambda, the observations are as follows.

M (Basis Functions)	Learning Rate	Lambda	Accuracy
10	0.001	0.1	57.67
10	0.001	0.2	57.51
10	0.001	0.5	57.39
10	0.001	1	57.32
10	0.001	2	57.278

7.1.3 Human Observed Dataset: Subtraction

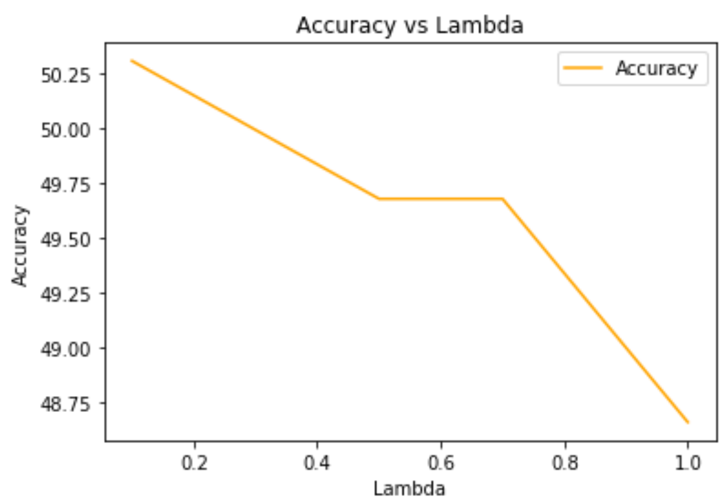
While Changing the Learning rate for each experiment, we've noted down the performance and accuracy of the model.

M (Basis Functions)	Learning Rate	Lambda	Train Acc	Val acc	Test Acc
10	0.05	2	51.14	60.97	50.77
10	0.01	2	51.56	61	50.94
10	0.001	2	51.64	61	52.83



While changing Lambda, the observations are as follows.

M (Basis Functions)	Learning Rate	Lambda	Train Acc	Val acc	Test Acc
10	**0.001**	**0.1**	50.15	61	**50.31**
10	**0.001**	0.5	48.59	58.49	49.68
10	**0.001**	0.7	48.59	58.49	49.68
10	**0.001**	1	48.22	59.41	48.66



7.1.2 GSC Observed Dataset: Subtraction

While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

M (Basis Functions)	Learning Rate	Lambda	Train Acc	Val acc	Test Acc
10	0.05	2	63.51	72.78	63.87
10	0.01	2	63.93	72.81	64.04
10	0.001	2	64.01	72.81	65.93

While changing Lambda, the observations are as follows.

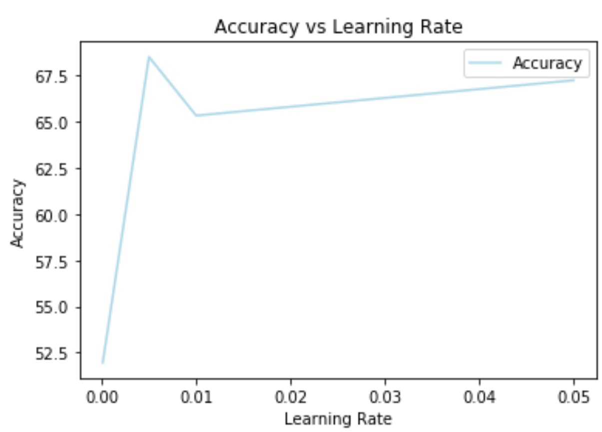
M (Basis Functions)	Learning Rate	Lambda	Train Acc	Val acc	Test Acc
10	**0.001**	**0.1**	59.06	72.37	**62.41**
10	**0.001**	0.5	57.5	69.86	61.78
10	**0.001**	0.7	57.5	69.86	61.78
10	**0.001**	1	57.13	70.78	60.76

7.2 Logistic Regression

7.2.1 Human Observed Dataset: Concatenation

While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

Learning rate	**Iterations**	**Val Acc**	**Test Acc**
0.0001	10000	53.58	51.95
0.005	10000	71.3	68.51
0.01	10000	71.94	65.33
0.05	10000	72.57	67.24



7.2.2 GSC Observed Dataset: Concatenation

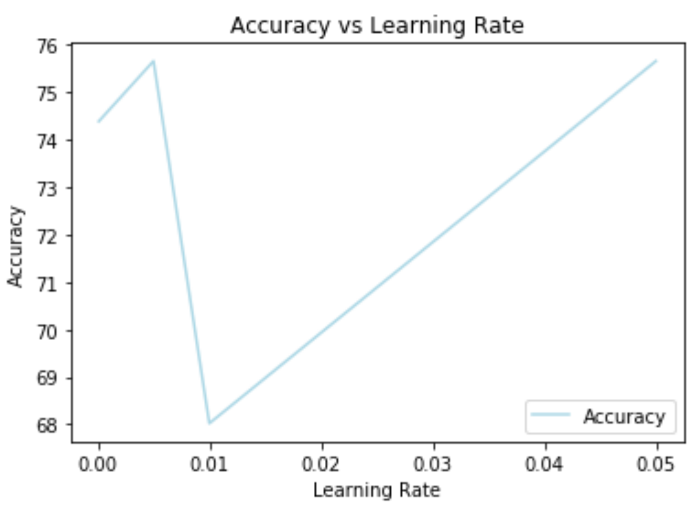
While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

Learning Rate	Validation
0.0001	67.7
0.005	85.42
0.01	86.06
0.05	86.69

7.2.3 Human Observed Dataset: Subtraction

While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

Learning rate	Validation Accuracy	Testing Accuracy
0.0001	67.16	74.39
0.005	91.56	75.66
0.01	84.6	68.02
0.05	90.92	75.66



7.2.4 GSC Observed Dataset: Subtraction

While Changing the **Learning rate** for each experiment, we've noted down the performance and accuracy of the model.

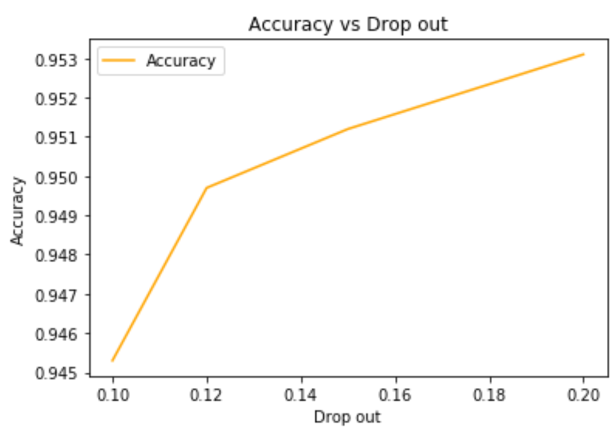
Learning rate	Training Accuracy	Validation Accuracy	Testing Accuracy
0.0001	79.63	80.37	86.62
0.005	87.45	91.77	87.89
0.01	81.13	84.81	80.25
0.05	87.52	91.13	87.89

7.3 Neural Network

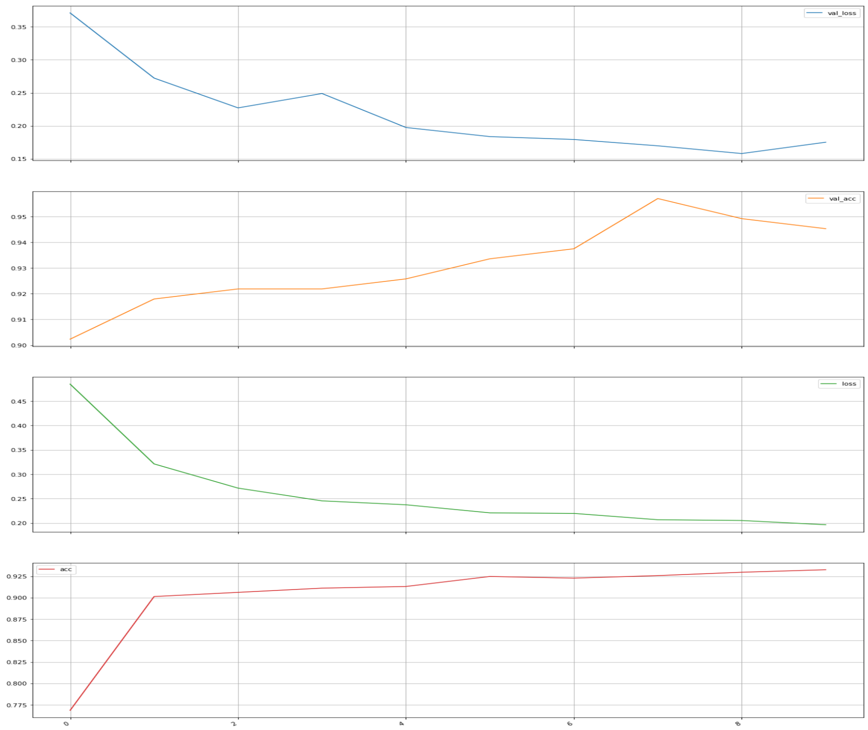
7.3.1 Human Observed Dataset: Subtraction

While Changing the Drop-out for each experiment, we've noted down the performance and accuracy of the model.

Drop out	epochs	Loss	acc	val acc
0.2	10	0.207	0.9268	0.9531
0.1	10	0.279	0.9163	0.9453



When Dropout is **1**

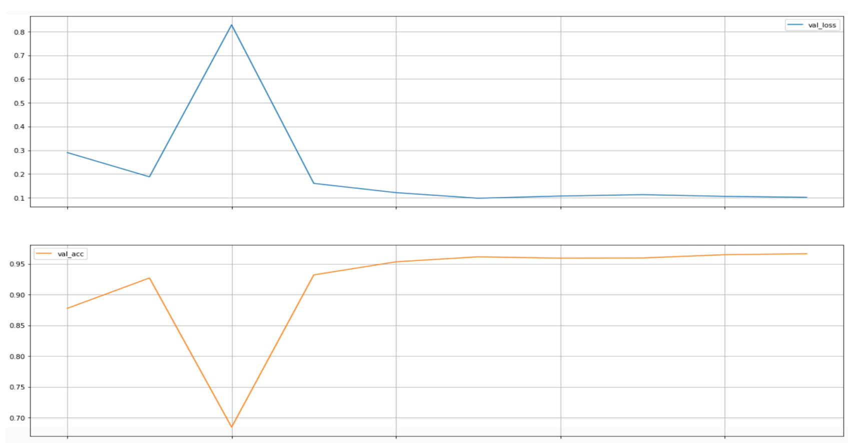


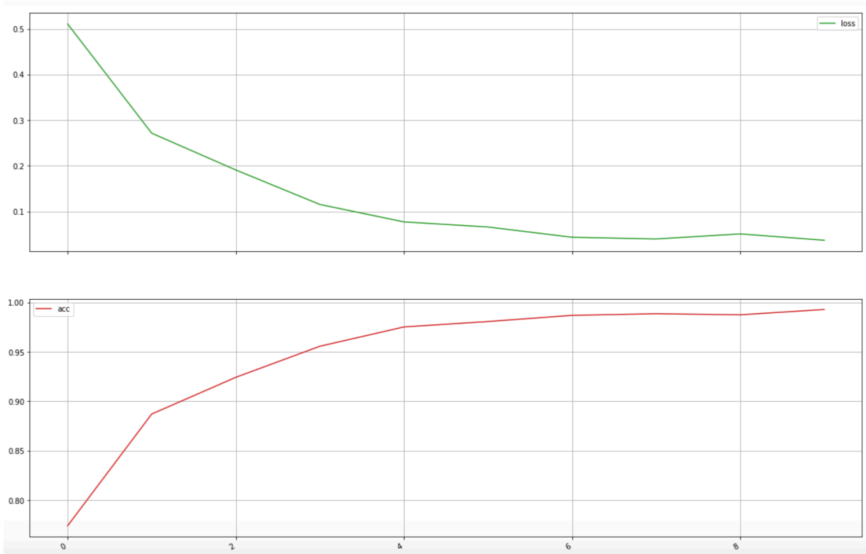
Drop out	epochs	Loss	acc	val acc	1 layer
0.2	10	0.207	0.9268	0.9531	2048
0.2	10	0.271	0.9376	0.9531	4096

The above graph explains that the **accuracy** increases as no of **epochs** increases until an extent.

7.3.2 GSC Observed Dataset: Subtraction

The below graph explains that the **accuracy** increases as no of **epochs** increases until an extent and **Loss** decreases as the no. of **epochs** increases

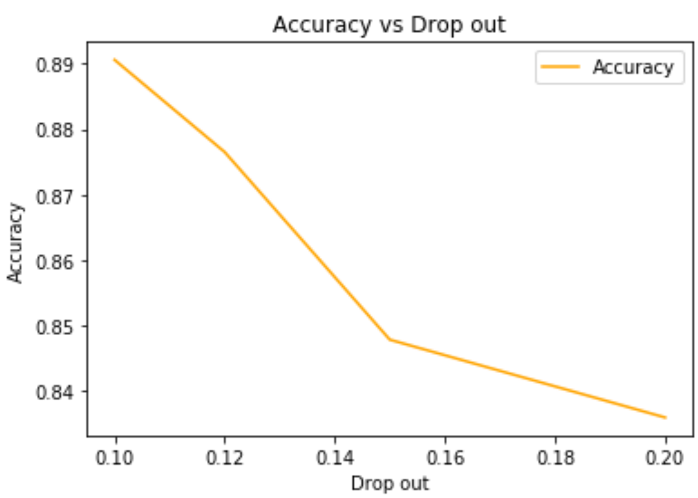




7.3.3 Human Observed Dataset: Concatenation

While Changing the **Dropout** for each experiment, we've noted down the performance and accuracy of the model.

Drop out	epochs	Loss	acc	val acc
0.2	10	0.207	0.9268	0.8359
0.1	10	0.187	0.9573	0.8906

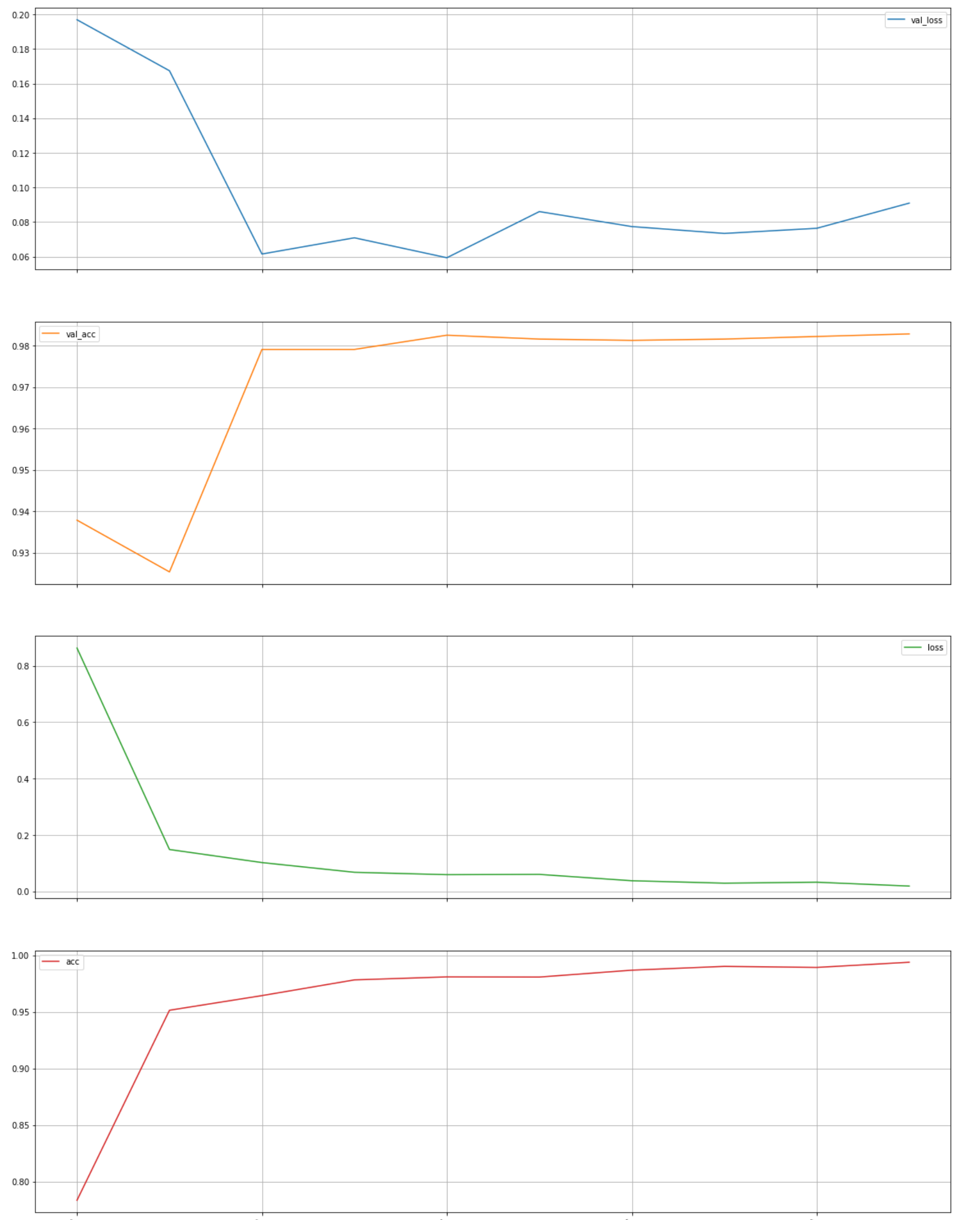


Observations while changing the no of layers

Drop out	epochs	Loss	acc	val acc	1 layer
0.2	10	0.207	0.9268	0.8359	2048
0.2	10	0.186	0.956	0.8633	4096

7.3.4 GSC Observed Dataset: Concatenation

The below graph explains that the **accuracy** increases as no of **epochs** increases until an extent and **Loss** decreases as the no. of **epochs** increases



8 Conclusion

Hence, according to the observations. We conclude that using **neural network** gives us the most accuracy of above 90% all the time.