# CSE574 Introduction to Machine Learning

# Project 2

# Handwriting Comparison in Forensics By Using Linear Regression, Logistic Regression and Neural Network

Project Report

Submitted By

Karan Manchandia

**Abstract**

This project aims to apply the concept of machine learning to solve the handwriting comparison task in forensics. Basically, the task of this project is to find the similarity between handwriting samples of known and questioned writers by using linear regression, logistic regression and neural network. There are two data sets given, One is Human Observed Features Data Set that contains nine features for each image instance and the another is GSC features data set that contains 512 features for each image instance. We have to compare two images by subtracting as well as concatenating the features. If the target output is one then the prediction is that the images are similar (i.e. the images come from the same writer). If the target output is zero then the prediction is that the images are dissimilar (i.e. the images come from a different writer).

**Introduction**

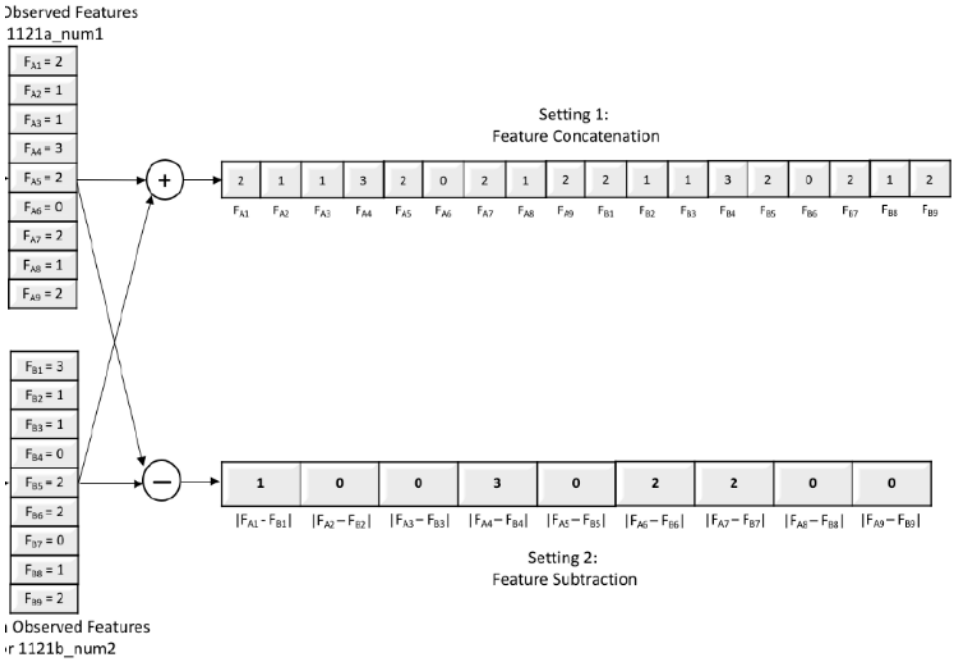
There are three ways we solve this problem:

1. By using Linear Regression Solution
2. By using Logistic Regression Solution
3. By using Neural Network Solution

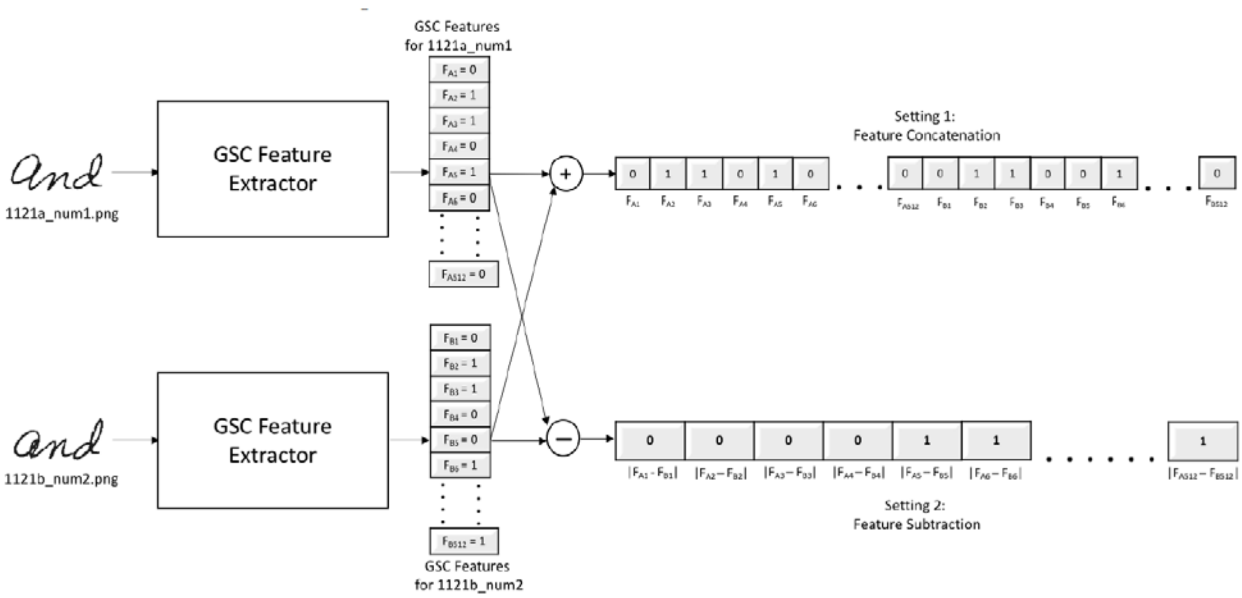
**Steps Involved in Project Implementation:**

1. **Processing of Data:**

   Process the three given csv files to create two files with image pairs, concatenated and subtracted features and target values for human observed data set. Repeat the same steps for GSC data set.




Feature Concatenationa and Subtraction for Human Observed Data Set




Feature Concatenationa and Subtraction for GSC Data Set

2. **Data Partioning:**

   Partition data into training testing and validation data sets.

3. **Train and Test using Linear Regression Model:**

   Use Gradient Decent for linear regression to train and test the model using a group of hyperparameters on each of the four input data sets.

4. **Train and Test using Logistic Regression Model:**

Use Gradient Decent for logistic regression to train and test the model using a group of hyperparameters on each of the four input data sets.

5. **Train and Test by using Neural Network Model:**

   Use a neural network model to train and test the model using a group of hyperparameters on each of the four input data sets.

**Linear Regression using Gradient Decent Solution:**

Linear Regression is an iterative method for finding weights to solve a particular problem (Forensic writing analysis in this case). The stochastic gradient decent solution first takes a random initial value of weights and then iteratively update the weights. This model goes in the opposite direction of the gradient of error, that is why it is called gradient decent. This model updates the weights using the equation:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)}$$

Where ,

Because of the linearity of differentiation we have:

$$\nabla E = \nabla E_D + \lambda \nabla E_W$$

Here,

$$\nabla E_D = -(t_n - \mathbf{w}^{(\tau)\top}\phi(\mathbf{x}_n))\phi(\mathbf{x}_n)$$
$$\nabla E_W = \mathbf{w}^{(\tau)}$$

Finally, we calculate the Erms value, which is given by the formula, as shown . The root mena square error (Erms) is used to measure the differences between predicted values by a model and the sample values.

$$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^*)/N_{\mathrm{V}}}$$

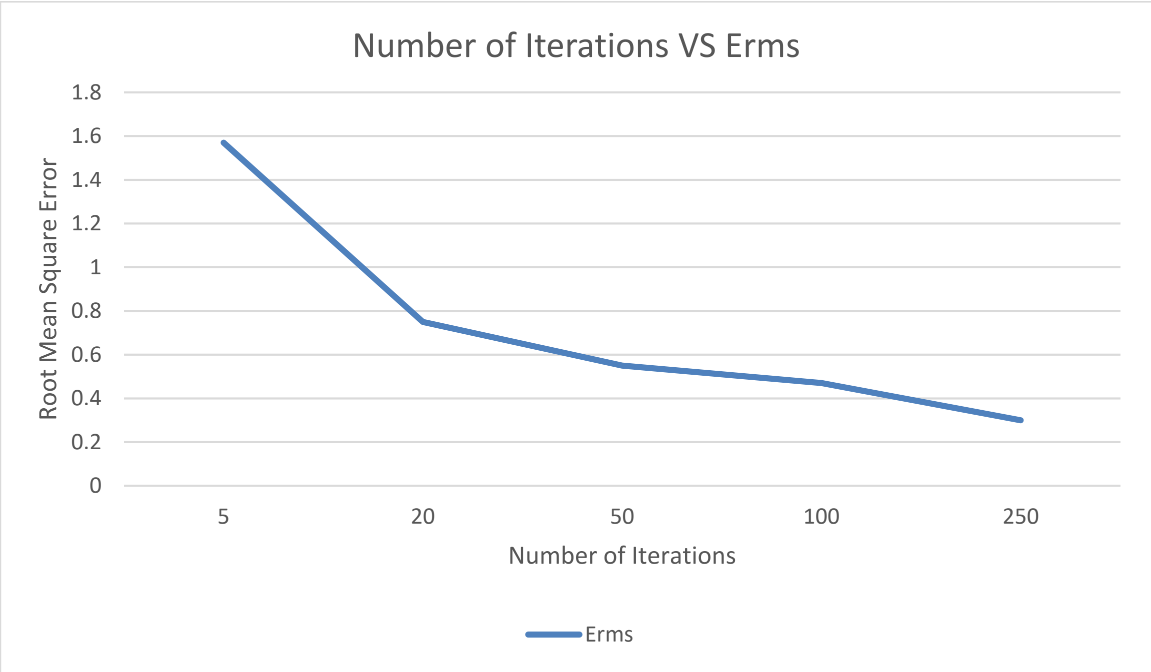
**What is the difference between Stochastic Gradient Decent and simple Gradient Decent?**

The simple gradient decent solution might find a local minimum rather then a global minimum, while the stochastic gradient decent is more efficient for finding the global minimum.

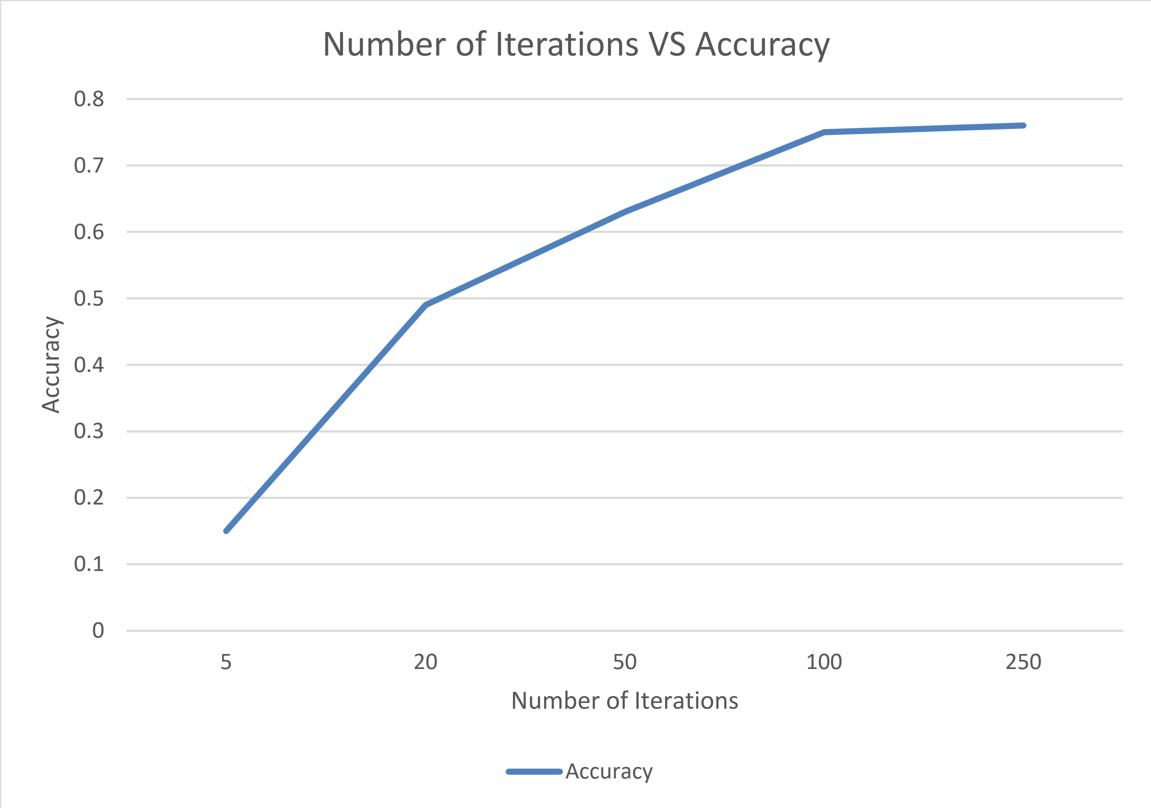
**For learning rate = 0.01, varying no. of iterations and checking accuracy and Erms for Human data set/ concat**

As the name suggests, Learning Rate defines, how quickly a model updates its parameters.

| Learning Rate | No. of Iterations | Accuracy | Erms |
|---|---|---|---|
| 0.01 | 5 | 0.15 | 1.57 |
| 0.01 | 20 | 0.49 | 0.75 |
| 0.01 | 50 | 0.63 | 0.55 |
| 0.01 | 100 | 0.75 | 0.47 |
| 0.01 | 250 | 0.74 | 0.30 |




It can be clearly seen in the graph above that as we increase the number of iterations Erms decrease.

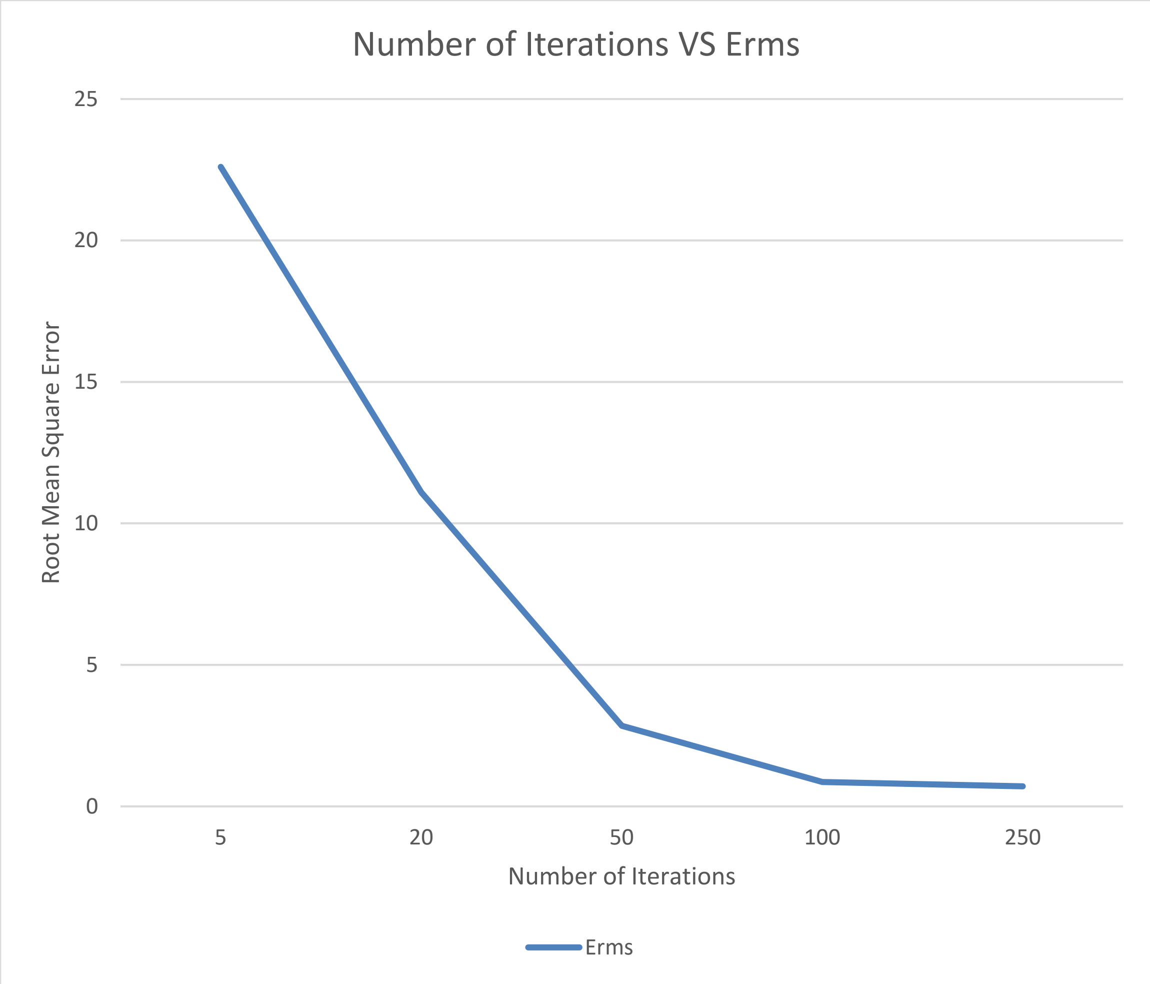



It can be clearly seen from the graph that accuracy increases as the number of iterations increases.

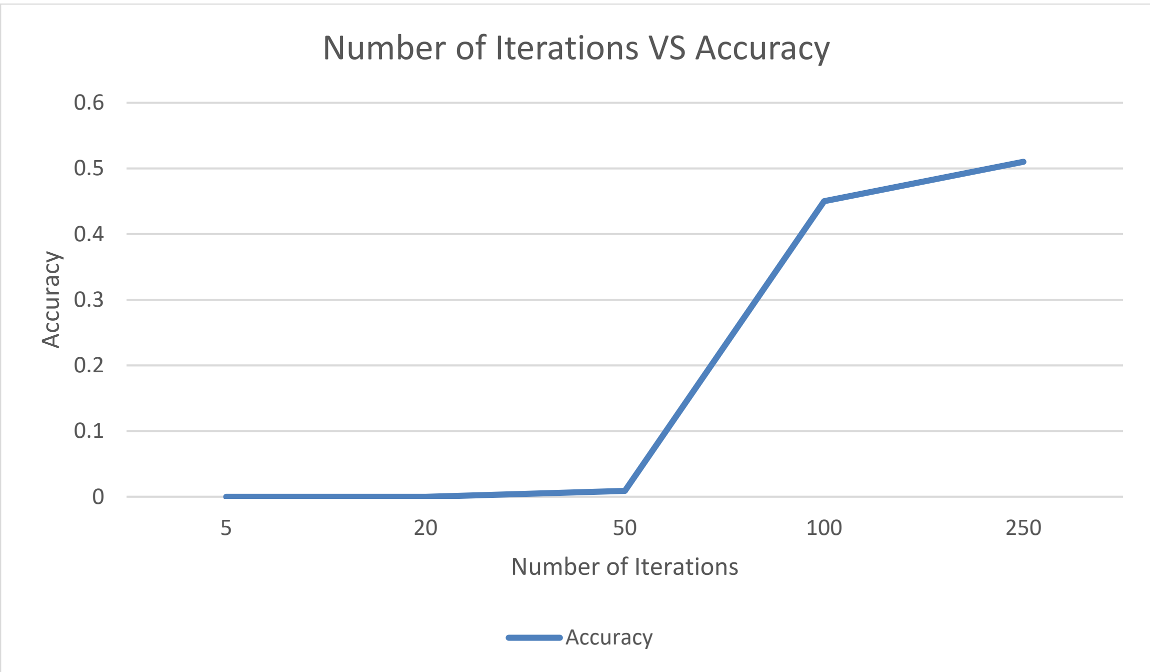
**For learning rate = 0.001, varying no. of iterations and checking accuracy and Erms for Human data set/ concat**

As the name suggests, Learning Rate defines, how quickly a model updates its parameters.

| Learning Rate | No. of Iterations | Accuracy | Erms |
|---|---|---|---|
| 0.001 | 5 | 0.0 | 22.6 |
| 0.001 | 20 | 0.0 | 11.1 |
| 0.001 | 50 | 0.0089 | 2.85 |
| 0.001 | 100 | 0.45 | 0.86 |
| 0.001 | 250 | 0.51 | 0.71 |



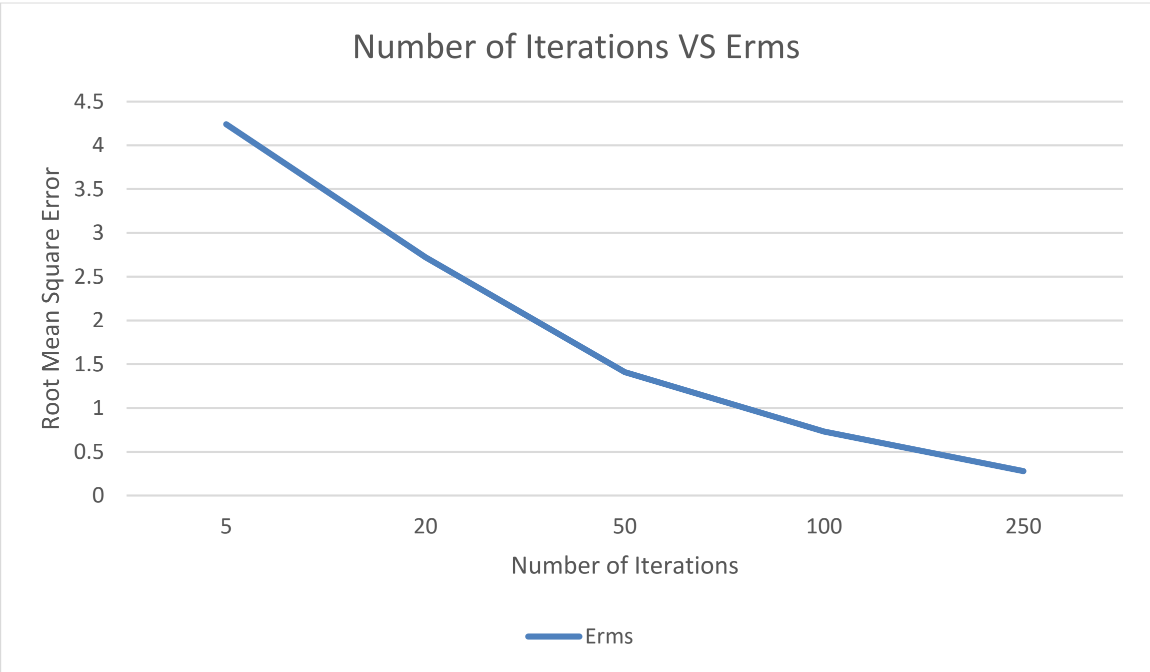

It can be clearly seen in the graph above that as we increase the number of iterations Erms decrease.



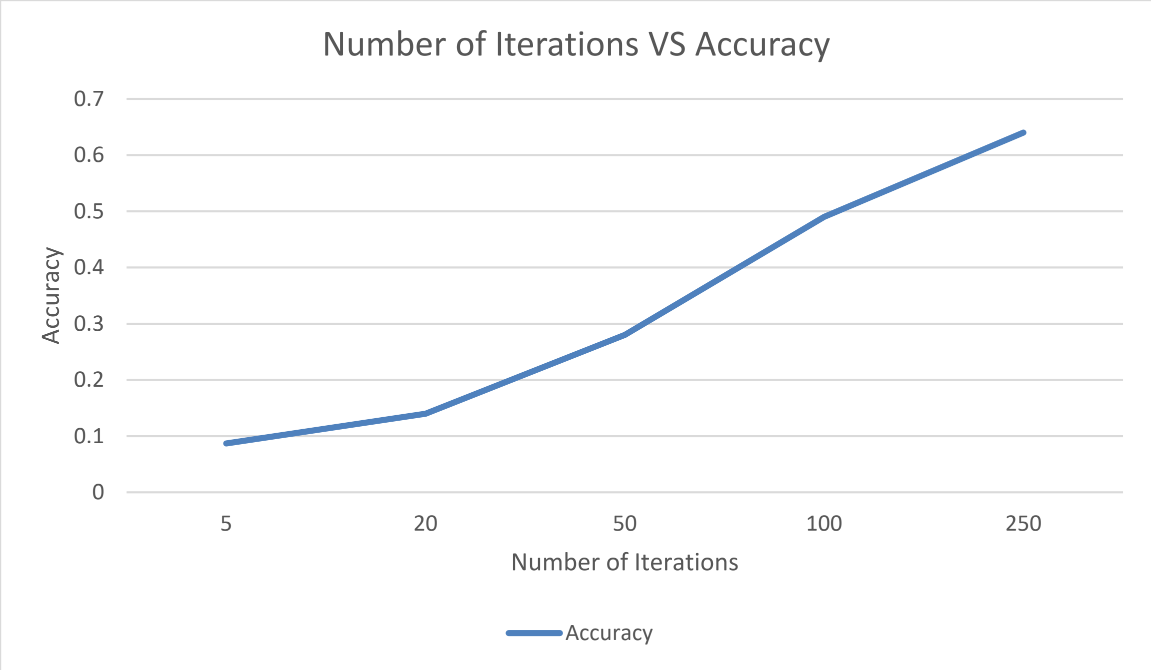
It can be clearly seen from the graph that accuracy increases as the number of iterations increases.

**For learning rate = 0.01, varying no. of iterations and checking accuracy and Erms for Human data set/ subtract**

| Learning Rate | No. of Iterations | Accuracy | Erms |
|---|---|---|---|
| 0.01 | 5 | 0.087 | 4.24 |
| 0.01 | 20 | 0.14 | 2.72 |
| 0.01 | 50 | 0.28 | 1.41 |
| 0.01 | 100 | 0.49 | 0.73 |
| 0.01 | 250 | 0.64 | 0.28 |




It can be clearly seen in the graph above that as we increase the number of iterations Erms decrease.




It can be clearly seen from the graph that accuracy increases as number of iterations increases.

**For learning rate = 0.001, varying no. of iterations and checking accuracy and Erms for Human data set/ subtract we get the data as shown in the table below**

| Learning Rate | No. of Iterations | Accuracy | Erms |
|---|---|---|---|
| 0.001 | 5 | 0.78 | 4.91 |
| 0.001 | 20 | 0.78 | 4.68 |
| 0.001 | 50 | 0.086 | 4.25 |
| 0.001 | 100 | 0.109 | 3.64 |
| 0.001 | 250 | 0.166 | 2.41 |

**Similarly, tuning the hyper parameters for GSC data set:**

For Gsc data set we can see that Accuracy and Erms vary in the similar way as it is when we are using Human Observed Data Set. The only difference that accuracy is a little higher in each case.

**Logistic Regression Solution:**

Logistic Regression is defined as a classification algorithm that can be used to map each observation to a discrete set of classes. Logistic Regression uses a logistic sigmoid function to return a value which can be mapped to various classes.

**Why we use Sigmoid Function in logistic regression?**

The sigmoid function is used to map a particular real valued output to a output between zero and one i.e. in order to map a predicted value to probability we use the sigmoid function. We use a sigmoid function to map predictions to probabilities in machine learning.

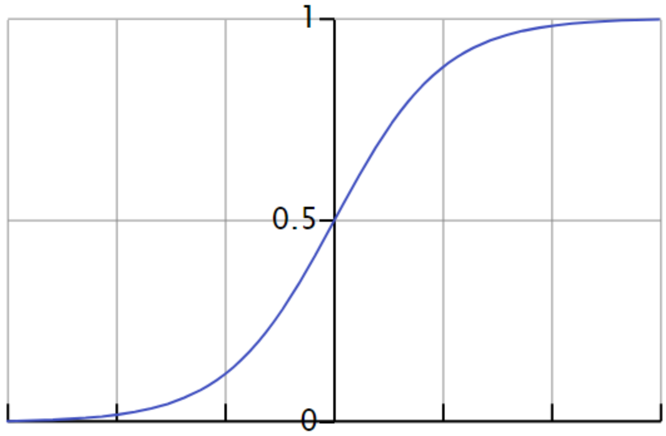
The formula for sigmoid function is given by:

Here, f(t) = output between 0 and 1.

T = input the the function

e = base of the natural log

A sigmoid function is shown below:




In logistic regression, we can not use the same cost function as it was used in case of linear regression. So, we use logarithmic cost function as shown below:

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

The above equation can be rewritten as:

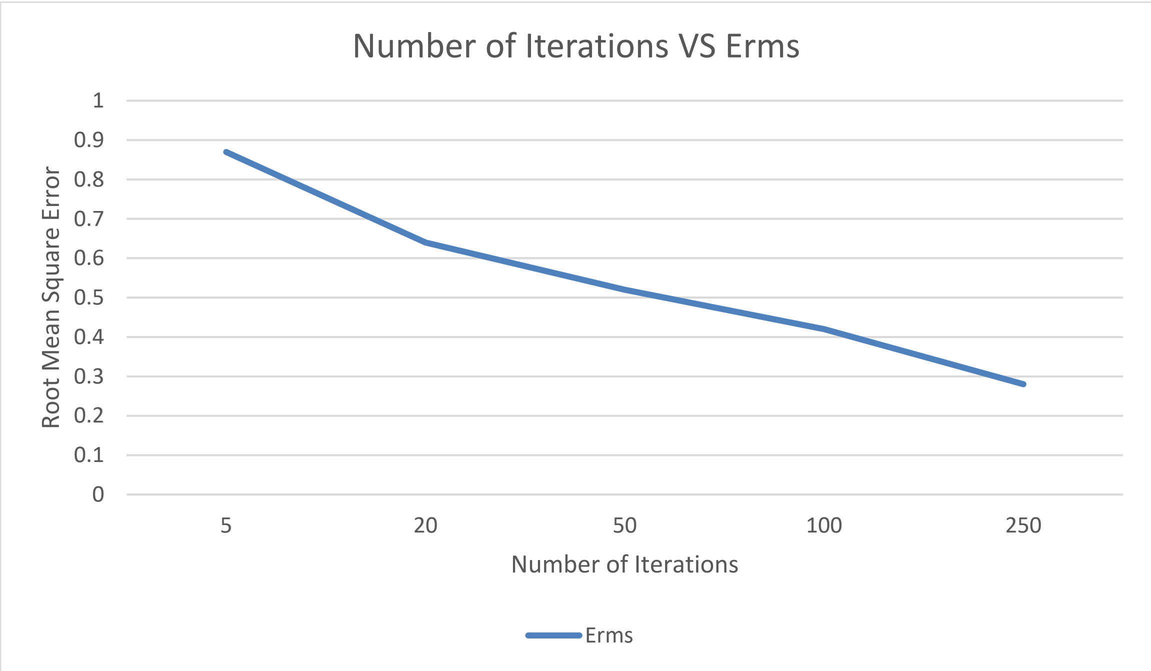
$$-\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Finally, we use the equation of the type as shown below for calculating gradients:

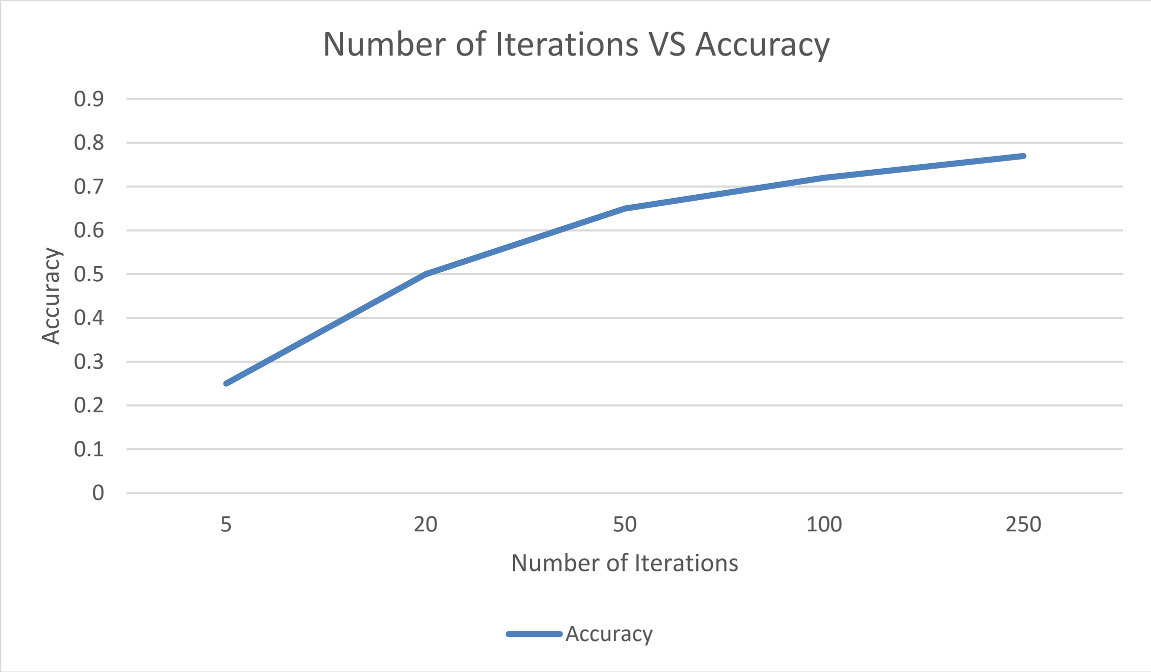
$$\frac{\partial J}{\partial \theta_n} = \frac{1}{m} \cdot x_i \cdot [\sum_{i=1}^{m} h_i - y_i]$$

**For learning rate = 0.01, varying no. of iterations and checking accuracy and Erms for Human data set/ concat for logistic regression**

| Learning Rate | No. of Iterations | Accuracy | Erms |
|---|---|---|---|
| 0.01 | 5 | 0.25 | 0.87 |
| 0.01 | 20 | 0.50 | 0.64 |
| 0.01 | 50 | 0.65 | 0.52 |
| 0.01 | 100 | 0.72 | 0.42 |
| 0.01 | 250 | 0.77 | 0.28 |



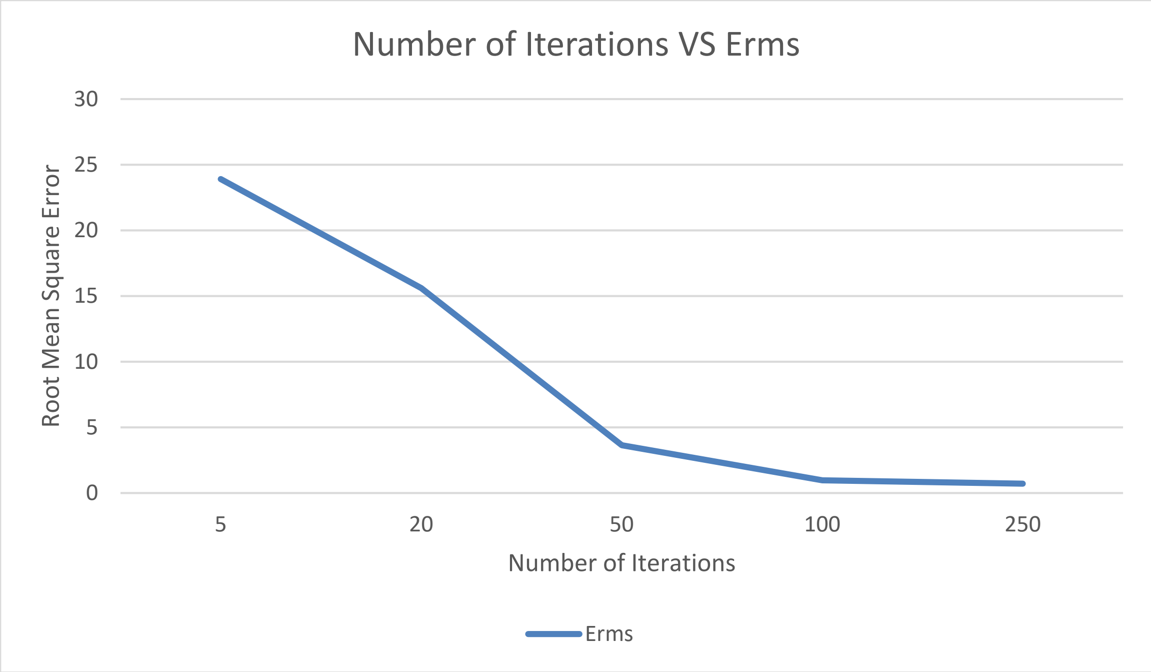

It can be clearly seen in the graph above that as we increase the number of iterations Erms decrease.



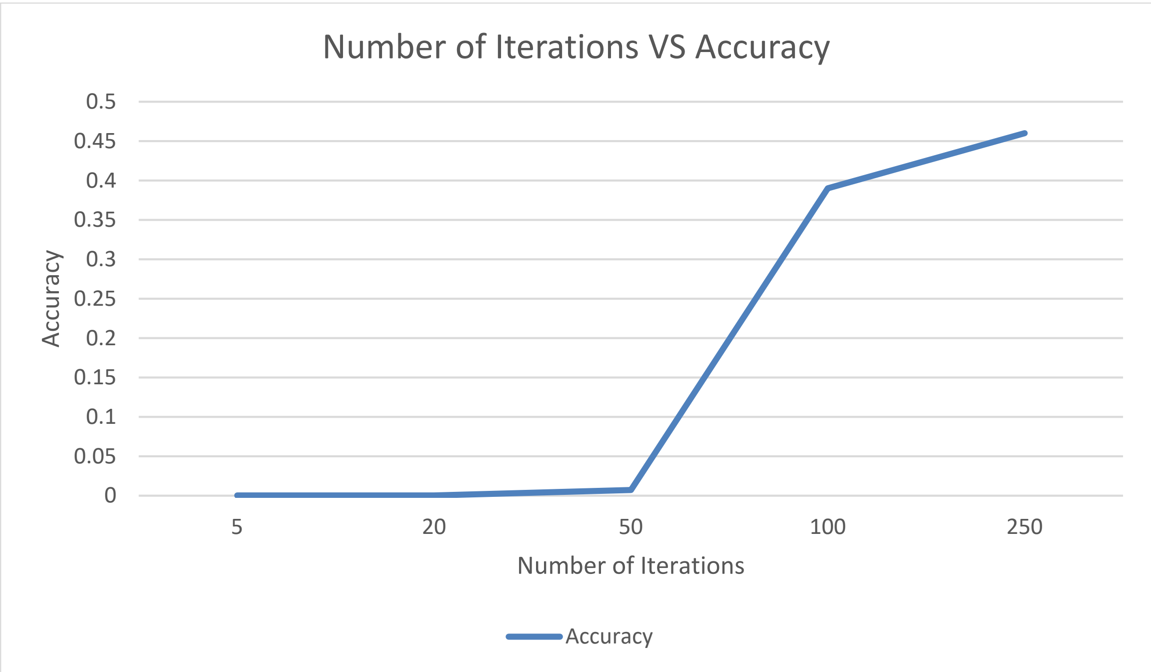

It can be clearly seen from the graph that accuracy increases as number of iterations increases.

**Similarly taking learning Rate = 0.001 and changing Number of Iterations we get accuracy and Erms values as shown in the figure below:**

| Learning Rate | No. of Iterations | Accuracy | Erms |
|---|---|---|---|
| 0.001 | 5 | 0.00032 | 23.9 |
| 0.001 | 20 | 0.00029 | 15.6 |
| 0.001 | 50 | 0.0072 | 3.64 |
| 0.001 | 100 | 0.39 | 0.97 |
| 0.001 | 250 | 0.46 | 0.72 |




It can be clearly seen in the graph above that as we increase number of iterations Erms decrease.




It can be clearly seen from the graph that accuracy increases as number of iterations increases.

**For learning rate = 0.01, varying no. of iterations and checking accuracy and Erms for Human data set/ subtract for logistic regression:**

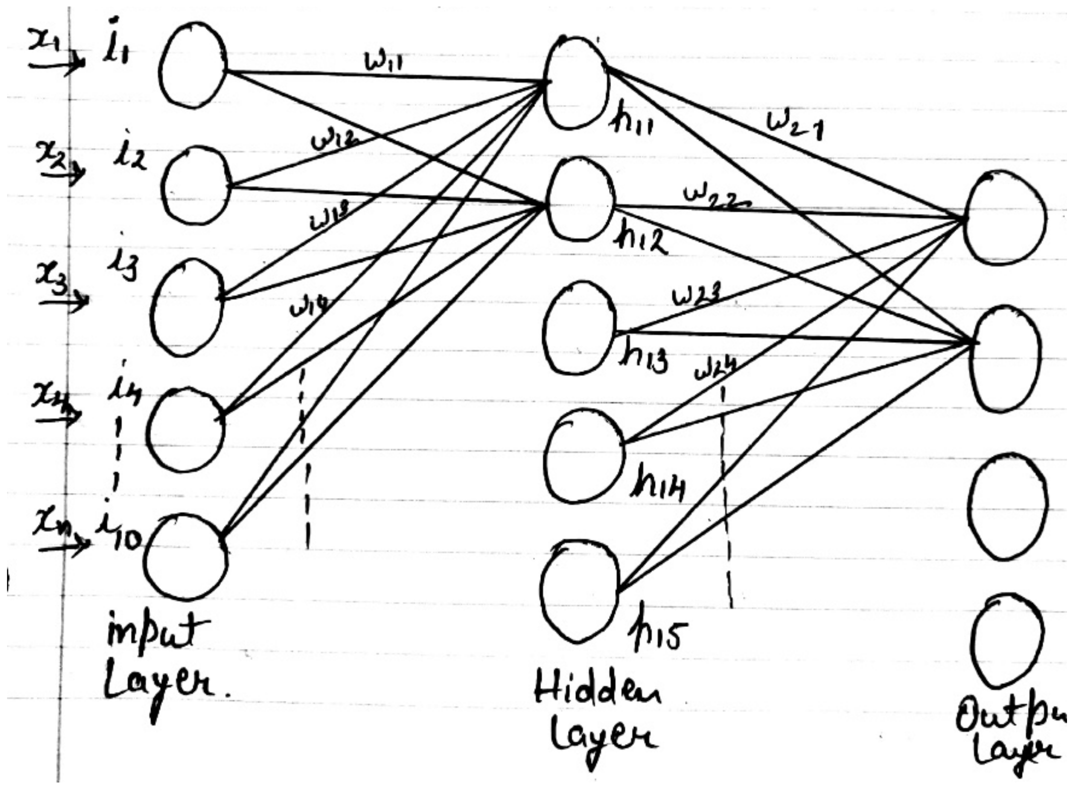
In a similar way as done before, we vary number of iterations, by keeping learning rate = 0.01 we get the maximum accuracy that was slightly lower then what we got for concatenation. The maximum accuracy values and the hyperparameters are mentioned in the conclusion. In the same way, we vary hyperparameters by keeping learning rate = 0.001.

**Similarly, tuning the hyper parameters for GSC data set for logistic regresion:**

For Gsc data set we can see that Accuracy and Erms vary in the similar way as it is when we are using Human Observed Data Set. The only difference that accuracy is a little higher in each case for the GSC data set.

**Neural Network Solution:**

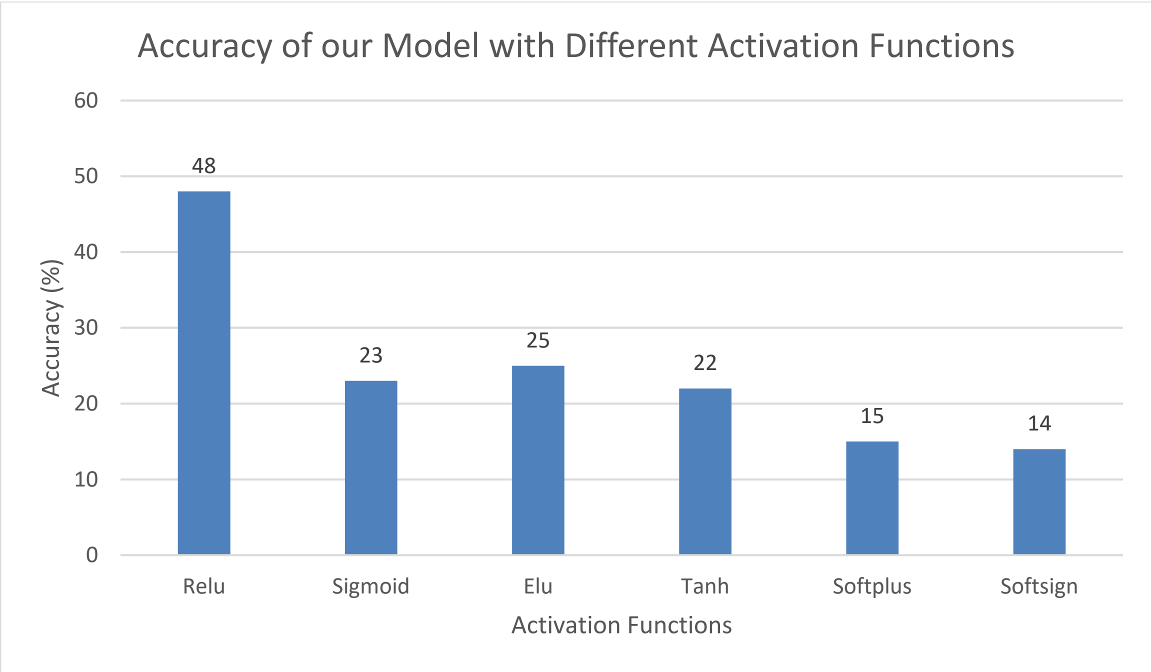
Neural Network is defined as a specific set of algorithm that is inspired by biological neural networks. Neural network is general function algorithm and hence can be applied to any problem that can be solved using machine learning. The figure below shows the block diagram of a two layered Neural Network Model.




1. **The relationship between Activation Function and accuracy:**

The functions that are used to introduce non linearity into the model are called activation functions. The following table and graph below show the accuracy of the Neural Network for different activation functions.

| Activation Function | Accuracy (%) |
|---|---|
| Relu | 48 |
| Sigmoid | 23 |
| Elu | 25 |
| Tanh | 22 |
| Softplus | 15 |
| Softsign | 14 |
| Relu | 48 |




The highest accuracy obtained for the Handwriting comparison problem is obtained with Rectified Linear Unit Activation Function.

**Why Relu?**

The following might be the reasons, why relu activation function gives the best accuracy for our forensic handwriting comparison problem:

- The gradient is 1 for all possible positive values of input. This has made the algorithm called work much faster.
- Relu gives zero output for negative values and positive output for all the positive inputs.
- Learning rate is fast when using Relu as compared to the sigmoid function.
- Relu gives the highest possible accuracy for our problem for neural Network Solution.

**Conclusion:**

**Selection of the most optimal Hyperparameters for linear regression**

As presented in the above paragraphs and the graphs, the following are the most optimal values for the hyperparameters for the Human-Observed data set. Almost a little higher accuracy is obtained for GSC data set.

For Linear Regression Solution:

Concatenation of Features:

No. of Iterations = 250

Learning Rate = 0.01

Accuracy = 0.76

Subtraction of Features:

Concatenation of Features:

No. of Iterations = 250

Learning Rate = 0.01

Accuracy = 0.64

As presented in the above paragraphs and the graphs, the following are the most optimal values for the hyperparameters for the Human-Observed data set. Almost a little higher accuracy is obtained for GSC data set.

For Logistic Regression Solution:

Concatenation of Features:

No. of Iterations = 250

Learning Rate = 0.01

Accuracy = 0.74

Subtraction of Features:

Concatenation of Features:

No. of Iterations = 250

Learning Rate = 0.01

Accuracy = 0.67

The following points can be concluded from our results:

- Among all the three solutions i.e. Linear Regression, Logistic regression and Neural Network Logistic regression gives the best accuracy for Human- observed as well as the GSC data set.

- So, among these three solutions, Logistic Regression is the best fit for both the data sets.
- As there are more number of features in the GSC data set, it gives better results. This has been verified from our results.
- Concatenating the features is a better practice as compared to subtraction. This is because all the features of both the images are preserved in case of concatenation, which is not the case with subtraction. This has been verified from the results as concatenation gives more accuracy.

**References**


- www.fon.hum.uva.nl/praat/manual/epoch.html
- https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a
- https://machinelearningmastery.com/logistic-regression-for-machine-learning
- https://stats.stackexchange.com/questions/176209/combining-several-features-into-one-single-feature