# Handwritting Comparison

**Anunay Rao**
anunayra@buffalo.edu

## 1 Introduction

The project is to apply the different machine learning methods namely, Linear Regression, Logistic Regression and Neural Network on the dataset to develop a predictive model and compare the performance of each model. The task is to compare the handwriting and so the target value 0 denotes that the handwriting samples that we are comparing are from different writers whereas the target value 1 denotes that the handwriting samples are from the same writer.

## 2 Preprocessing of the Dataset

Initially we have been provided with two datasets namely, Human Observed Dataset and GSC Dataset.

### 2.1 Human Observed Dataset

This dataset consists 3 CSV files:

- **same_pair.csv (img_id_A, img_id_B, target):** where img_id_A and img_id_B are two id's of images from the same writer and so the target value corresponding to all the samples in this file is 1. This file contains 791 samples.
- **diffn_pairs.csv (img_id_A, img_id_B, target):** where img_id_A and img_id_B are two id's of images from the different writer and so the target value corresponding to all the samples in this file is 0. This file contains 293032 samples
- **HumanObserved-Features-Data.csv (img_id,f1,f2………,f9):** where img_id is id of a particular image and f1 to f9 are human observed features corresponding to that image.

Now since the same_pairs.csv has only 791 samples whereas diffn_pairs.csv has 293032 samples we need to make the probability of each class equal in order to apply any machine learning method. Further each image has 9 features so we will create 2 datasets one with concatenating the features thus getting 18 features for one image pair and other with subtracting the features (absolute value) thus getting 9 features for one image pair, so we need to follow the steps given below in order to get a proper dataset.

1. Copy the features corresponding to all the the image pairs in same_pairs.csv from HumanObserved-Features-Data.csv and and concatenate them to get 18 features and 1 as the target value in file named final_cat.csv.
2. Shuffle the rows in diffn_pairs.csv.
3. Copy the features corresponding to the first 791 image pairs in the diffn_pairs.csv from HumanObserved-Features-Data.csv and concatenate them to get 18 features and 1 as the target value and append them into final_cat.csv.

4. Shuffle the rows in the file final_cat.csv to get the final data.
5. Now to get the Dataset with 9 features i.e. feature subtraction subtract the col index[9] with col index[0], [1] with [10],……..[8] with [17]. This file will be final_sub.csv where col index[10] will be the target value.

## 2.1 GSC Dataset

This dataset also contains 3 CSV files:

- **same_pair.csv (img_id_A, img_id_B, target):** where img_id_A and img_id_B are two id's of images from the same writer and so the target value corresponding to all the samples in this file is 1. This file contains 71531 samples.
- **diffn_pairs.csv (img_id_A, img_id_B, target):** where img_id_A and img_id_B are two id's of images from the different writer and so the target value corresponding to all the samples in this file is 0. This file contains 762557 samples
- **GSC-Features.csv (img_id,f1,f2………,f9):** where img_id is id of a particular image and f1 to f512 are 512 features corresponding to that image which are extracted from the image with GSC algorithm.

Now since the same_pairs.csv has only 71531 samples whereas diffn_pairs.csv has 762557 samples we need to make the probability of each class equal in order to apply any machine learning method. Further each image has 512 features so we will create 2 datasets one with concatenating the features thus getting 1024 features for one image pair and other with subtracting the features (absolute value) thus getting 512 features for one image pair. These two datasets can be obtained in the similar fashion as described for Human Observed dataset.

For linear regression task, we need to compute the inverse of the matrix and and it is found that for GSC dataset it yields singular matrix which implies that some columns have all the entries as zero which needs to be processed so that we can compute the inverse. So, one way is to add some noise to the diagonal elements of the matrix or delete the columns with zero value. I have opted for the second method and found that there are 6 columns with zero value in feature concatenation namely column index [450], [452], [456], [457], [962], [964] and [968] in case of feature subtraction there are 3 columns namely column index [450], [452] and [456].

# 3 Performance Metric

**Linear Regression:** We will evaluate the solution obtained by using Root Mean Square (RMS) error defined as $E_{RMS} = \sqrt{2E(w^*)/N_V}$ where $w^*$ is the solution and $N_V$ is the size of dataset. Accuracy is not a good performance metric for this linear regression tasks.

**Logistic Regression and Neural Network:** We will evaluate the performance of these two models by accuracy which is defined as:

$$\boldsymbol{Accuracy = \frac{correct}{correct + wrong} \times 100}$$

As here accuracy makes more sense than Erms for calculating the performance.

# 4 Hyper-parameters values and Results:

## 4.1 Linear Regression

$\lambda$ for closed form: 0.03 (Regularization Term)
Training Percent = 80
Validation Percet = 10
Testing Percent = 10
$\lambda$ for gradient descent solution: 1.8 (Regularization Term)
learning rate : 0.01

### 4.1.1 Linear Regression on Human Observed Dataset with Feature Concatenation:

# of Gaussian Basis Function: 18

**Results:**

**Closed Form:**
E_rms Training = 0.4967303666643748
E_rms Validation = 0.4936395796664404
E_rms Testing = 0.49767401628116004

**Gradient Descent:**
E_rms Training = 0.49957
E_rms Validation = 0.49914
E_rms Testing = 0.49795

### 4.1.2 Linear Regression on Human Observed Dataset with Feature Subtraction:

# of Gaussian Basis Function: 9

**Results:**

**Closed Form:**
E_rms Training = 0.4991599889825706
E_rms Validation = 0.497113631312705
E_rms Testing = 0.4973578173205024

**Gradient Descent:**
E_rms Training = 0.50009
E_rms Validation = 0.49994
E_rms Testing = 0.49513

### 4.1.3 Linear Regression on GSC Dataset with Feature

**Concatenation:**

# of Gaussian Basis Function: 10

**Results:**

**Closed Form:**
E_rms Training = 0.5070384151105187
E_rms Validation = 0.5052882346175389
E_rms Testing = 0.506047865254507

**Gradient Descent:**
E_rms Training = 0.68516
E_rms Validation = 0.67824
E_rms Testing = 0.68111

### 4.1.4 Linear Regression on GSC Dataset with Feature Subtraction:

# of Gaussian Basis Function: 10

**Results:**

**Closed Form:**
E_rms Training = 0.7082241711685279
E_rms Validation = 0.7010008622415624
E_rms Testing = 0.704159509748694

**Gradient Descent:**
E_rms Training = 0.70822
E_rms Validation = 0.701
E_rms Testing = 0.70416

## 4.2 Logistic Regression:

Training Percent = 80
Validation Percet = 10
Testing Percent = 10
learning rate : 0.01

### 4.2.1 Logistic Regression on Human Observed Dataset with Feature Concatenation:

**Results:**

Training Accuracy = 56.962025316455694
Validation Accuracy = 55.69620253164557
Testing Accuracy = 45.859872611464965

### 4.2.2 Logistic Regression on Human Observed Dataset with

**Feature Subtraction:**

**Results:**

Training Accuracy   = 49.32806324110672
Validation Accuracy = 51.265822784810126
Testing Accuracy    = 54.140127388535035

### 4.2.3 Logistic Regression on GSC Dataset with Feature Concatenation:

**Results:**

Training Accuracy   = 55.83049366535605
Validation Accuracy = 54.65538934712708
Testing Accuracy    = 54.771059070255156

### 4.2.4 Logistic Regression on GSC Dataset with Feature Subtraction:

**Result:**

Training Accuracy   = 77.25993883792049
Validation Accuracy = 76.69509296798546
Testing Accuracy    = 76.77735057672143

## 4.3 Neural Network Implementaion:

Training Percent = 81
Validation Percet = 9
Testing Percent = 10
drop_out = 0.2
first_dense_layer_nodes  = 256
second_dense_layer_nodes = 1
Activation function first layer = ReLu
Activation function second layer = sigmoid
Optimizer = rmsprop
Loss = binary_crossentropy
model_batch_size = 128

### 4.3.1 Neural Network on Human Observed Dataset with Feature Concatenation:

num_epochs = 10000
early_patience = 100
input_sizze = 18

**Results:**

Errors: 66
Correct :91
Testing Accuracy: 57.961783439490446

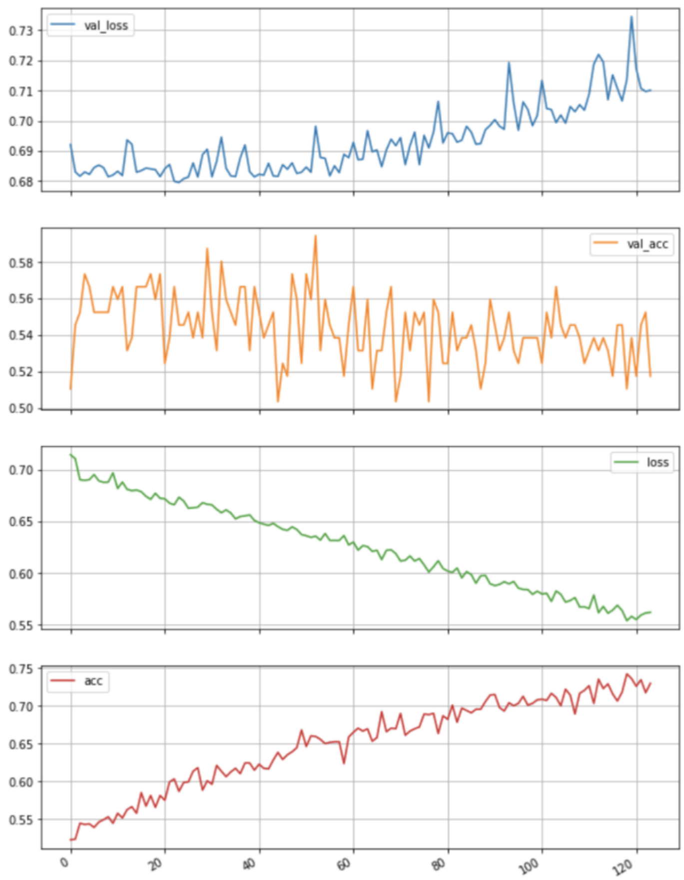


*Figure 1: Showing Validation loss, Validation Accuracy, Training Loss and Training Accuracy (top to bottom) against number of epochs*

### 4.3.2 Neural Network on Human Observed Dataset with Feature Subtraction:

num_epochs = 10000
early_patience = 100
input_sizze = 9

**Results:**

Errors: 19
Correct :138
Testing Accuracy: 87.89808917197452

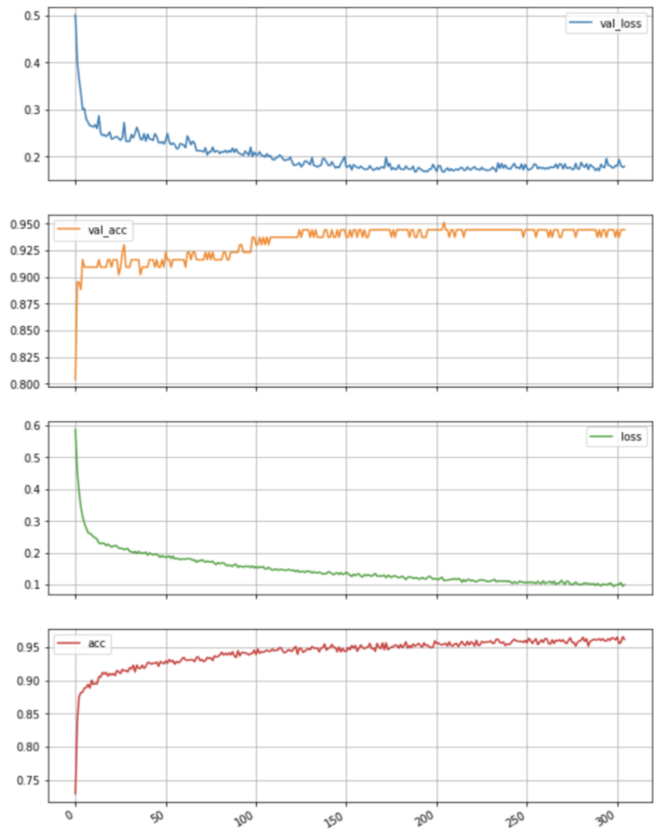


*Figure 2:Showing Validation loss, Validation Accuracy, Training Loss and Training Accuracy (top to bottom) against number of epochs*

### 4.3.3 Neural Network on GSC Dataset with Feature Concatenation:

num_epochs = 50
early_patience = 10
input_size = 1024

**Results:**

Errors: 950
Correct :13355
Testing Accuracy: 93.35896539671444

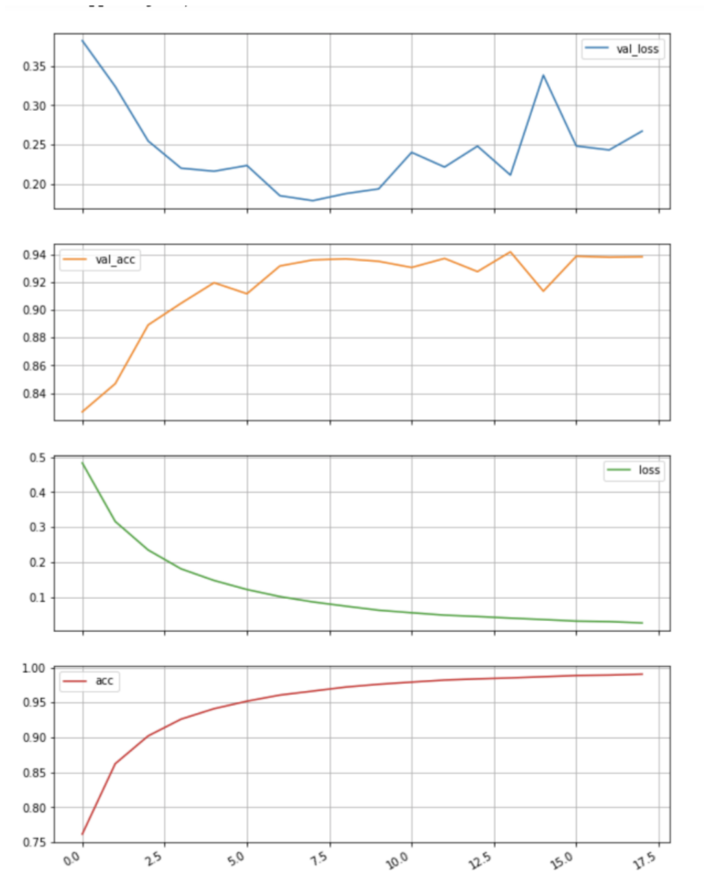


*Figure 3:Showing Validation loss, Validation Accuracy, Training Loss and Training Accuracy (top to bottom) against number of epochs*

#### 4.3.4 Neural Network on GSC Dataset with Feature Subtraction:

num_epochs = 50
early_patience = 10
input_size = 512

**Results:**

Errors: 2503
Correct :11802
Testing Accuracy: 82.50262146102762

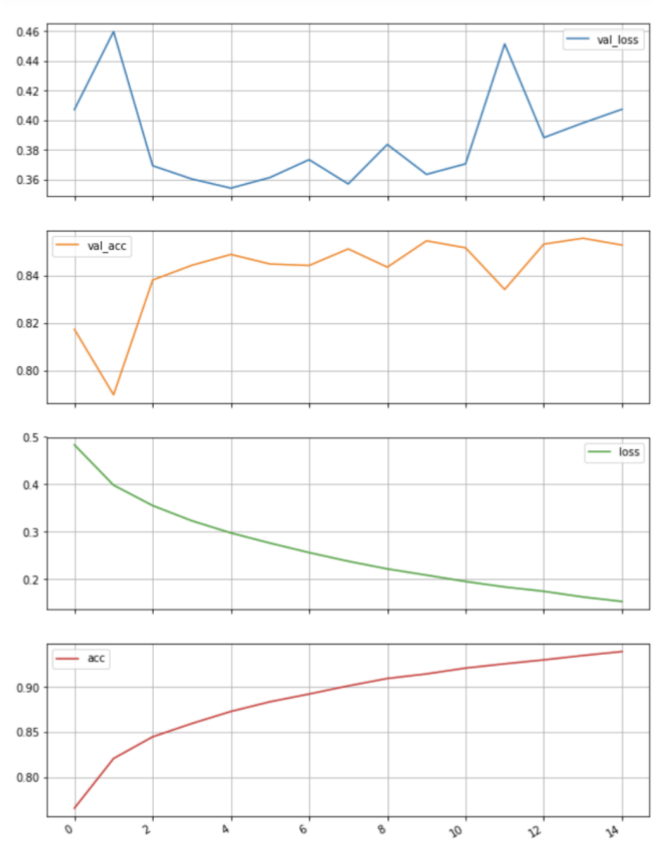


*Figure 4:Showing Validation loss, Validation Accuracy, Training Loss and Training Accuracy (top to bottom) against number of epochs*

## Conclusion:

| | **Human Observed Dataset** | **GSC Dataset** |
|---|---|---|
| **Logisitic Regression** (Concatenation) | 45.859 | 54.771 |
| **Logisitc Regression** (Subtraction) | 54.140 | 76.777 |
| **Neural Network** (Concatenation) | 57.961 | 93.358 |
| **Neural Network** (Subtraction) | 87.898 | 82.502 |

*Table 1: Testing Accuracy values for different models with different configuration*

As seen from the above data,

- Neural Network performed best on GSC Dataset with feature concatenation.
- Neural Network performed best on Human Observed data with feature subtraction.

- For Human Observed Dataset, Feature Subtraction setting is better than concatenation setting.

- For GSC Dataset, Logistic Regression performed better in subtraction setting as compared to concatenation setting. Whereas neural network performed better for Concatenation setting as compared to subtraction setting.

- When compared to Logistic regression, Neural network performed better on both the datasets in all the settings whether it is concatenation or subtraction.

## References

[1] Towards Data Science. (2018). Machine Learning – Towards Data Science. [online] Available at: https://towardsdatascience.com/machine-learning/home

[2] Brownlee, J. (2018). Machine Learning Mastery. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/