CSE 574 PROJECT 2 REPORT

Vishnu Varshath Harishankar
Person Number : 50291399
UBID : vharisha

Abstract

This project aims to detect the similarity of two images and detect if it was written by the same person. The data set has a set of 'and' images. Each image belonging to a writer. We make use of three algorithms namely Linear Regression, Logistic Regression and Neural Networks to compare and solve the problem.

1 Data Preprocessing and Feature Extraction

1. **GSC features**: Gradient Structural Concavity algorithm generates 512 sized feature vector for an input handwritten "AND"image. The entire data set consists of 71,531 same writer pairs and 762,557 different writer pairs(rows). As we did for the Human Observed data set we follow concatenation and subtraction approaches. GSC Dataset after concatentation of features contains a total of 1024 features GSC Dataset after feature subtraction contains a total of 512 features
2. **Human Observed features**: The entire dataset consists of 791 same writer pairs and 293,032 different writer pairs(rows). The data set is provided in such a way that the same pairs and different pairs files have the image Id's and corresponding target Value. The target Value is 1 if the images belong to the same writer and 0 if they belong to a different writer. There is also another file which has the list of features. We try to solve this problem using two approaches. First is to concatenate the features of two images giving 2n total features where n is the number of features for each image. Second approach is the subtraction technique the two features are subtracted. Therefore there will be n features resulting from the subtraction of feature of image 1 and image 2 Human Observed Dataset after feature subtraction contains a total of 9 features Human Observed Dataset after feature concatentation contains a total of 18 features

In order to prevent the dataset being skewed, an equal number of records is chosen from the same pairs and different pairs dataset. This means the probabiltity of either one happening is exactly 0.5 The training set constitutes 80 percent of the total data. The validation set constitutes 10 percent of the entire data The test set constitutes the rest 10 percent of the data

2 Implementation

The problem is implemented using the following algorithms

1. Linear regression
2. Logistic regression
3. Neural network

For each of the above methods the data set is divided into training, validation and testing. The model is tuned with hyper parameters that give the best results

3 Linear Regression Implementation

Learning a linear regression model means estimating the values of the coefficients popularly called as weights used in the representation with the data that we have available. This method is applied to all four data sets and compared with each other

3.1 SGD Solution

When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data.
The sum of the squared errors are calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible. We minimize the error function by following the gradients.The stochastic gradient descent algorithm first takes a random initial value $w_{(0)}$. Then it updates the value of w using

$$w_{(t+1)} = w_{(t)} + \Delta w_{(t)}$$

$$\Delta E = \Delta E_d + \lambda E_w$$

where λ is the regularization term and $E_w = w^T\ \Delta E_d$ is calculated as follows :

$$\Delta E_d = \frac{1}{2}(Expected\ Target - obtained\ Target)$$

where $obtained\ Target = w^T \phi(X)\ Expected\ Target = t$

$$w^T \phi(X) = w_1\phi(x) + w_2\phi(x)...w_n\phi(x)$$

Therefore,

$$\Delta E_d = \frac{1}{2}t - w_1\phi(x) + w_2\phi(x)...w_n\phi(x)$$

3.2 Human Observed Dataset with feature Concatenation

The Erms value for training , validation , testing data sets was kind of similar which can be inferred from the graph below.

The hyper-parameters are : learning Rate = 0.005; Regularization Term = 0.6

After the learning the model the erms for training , validation testing is 0.55231, 0.5618, 0.55892 respectively

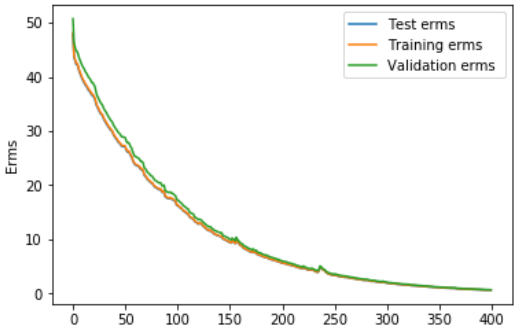

Figure 1: Accuracy

3.3 Human Observed Dataset with feature Subtraction

There was a negligible difference in performance between the concatenation and subtraction techniques.

For training this dataset the hyperparameters were set to :
learning Rate = 0.005
Regularization Term = 0.6

After the learning the model produced the below results :

Erms Training = 0.57156
Erms Validation = 0.57899
Erms Testing = 0.5801

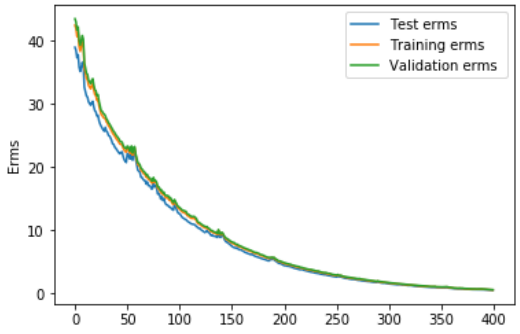

Figure 2: Accuracy

3.4 GSC Dataset with feature Concatenation

The Erms value for training , validation , testing data sets was kind of similar which can be inferred from the graph below. The hyper-parameters are :
learning Rate = 0.005
Regularization Term = 0.6

After the learning the model produced the below results :

Erms Training = 0.47910
Erms Validation = 0.482010
Erms Testing = 0.483911

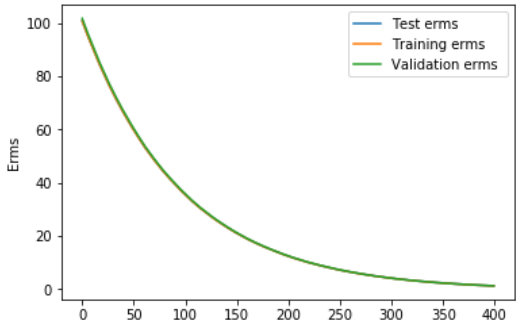

Figure 3: Accuracy

3.5 GSC Dataset with feature Subtraction

There was a negligible difference in performance between the concatenation and subtraction techniques.

learning Rate = 0.005
Regularization Term = 0.6

After the learning the model produced the below results :

Erms Training = 0.52119
Erms Validation = 0.51998
Erms Testing = 0.51565

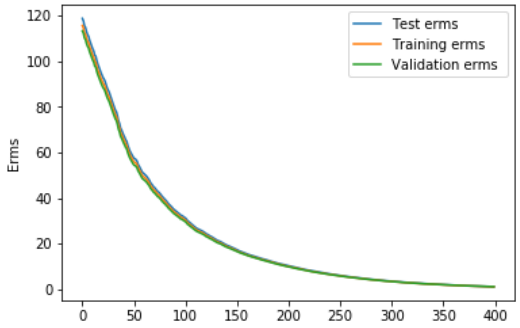

Figure 4: Accuracy

4 Logistic Regression Implementation

4.1 Human Observed data with feature concatenation

It can be seen from the graphs below that the model performed better by giving slightly higher accuracy than the subtraction technique

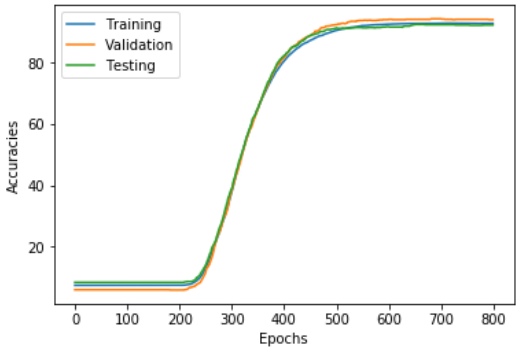

Figure 5: Accuracy

The learning rate was set as 0.001 and the number of epochs as 800.

Training Accuracy = 91.01437
Validation Accuracy = 92.4393
Testing Accuracy = 90.59259

4.2 Human Observed data with feature subtraction

When compared to the feature concatenation the results were lower

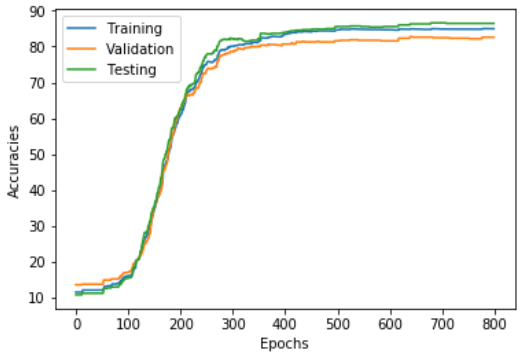

Figure 6: Accuracy

The learning rate was set as 0.005 and the number of epochs as 800.

Training Accuracy = 85.01473
Validation Accuracy = 82.76878
Testing Accuracy = 86.61765

4.3 GSC data with feature concatenation

The model performed slightly better when it was presented with the concatenated features.

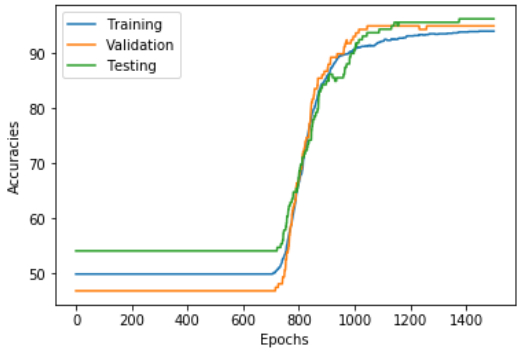

Figure 7: Accuracy

The learning rate was set as 0.001 and the number of epochs as 1500.

Training Accuracy = 93.99209
Validation Accuracy = 94.93671
Testing Accuracy = 96.22642

4.4 GSC data with feature subtraction

When compared to the feature concatenation technique the subtraction technique was not so efficient.

The learning rate was set as 0.001 and the number of epochs as 1500.

Training Accuracy = 86.16601
Validation Accuracy = 81.01266
Testing Accuracy = 83.018872

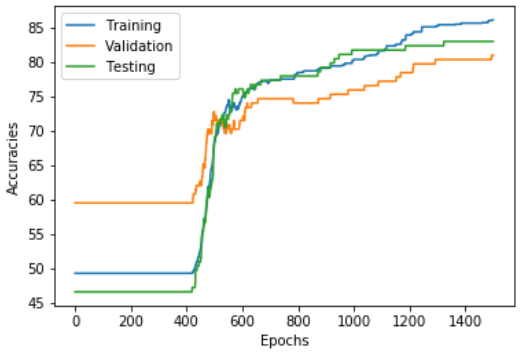

Figure 8: Accuracy

5 Neural Networks Implementation

The implementation was done using Keras. The model is based on neural networks and requires back propagation to update the weights which are first initialized randomly. We first calculate the loss function which is a mathematical difference between the predicted value of the model and the actual truth value. The loss for each input is calculated and an average is calculated. Then according to the learning rate, we can make the model learn rapidly (at the cost of not finding the optimal solution) or learn slower and find the optimal solution (at the cost of increase in time taken) to find the optimal solution. We must be careful about finding the right solution as there is a problem called over fitting where the model becomes too specific to the training data. To tackle such a problem, we made use of the dropout and early stopping features.

5.1 Human Observed data with feature concatenation

It can be seen from the graphs below that the model performed better by giving slightly higher accuracy than the subtraction technique

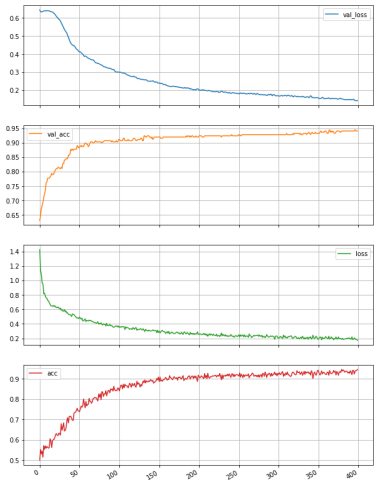

Figure 9: Loss and Accuracy

The following hyper parameters were used:

dropout = 0.4
first dense layer nodes = 50
second dense layer nodes = 2
Number of epochs = 400
Model Batch Size = 128
early patience = 100 epochs

Accuracy = 94.01437

5.2 Human Observed data with feature subtraction

When compared to the feature concatenation the results were lower

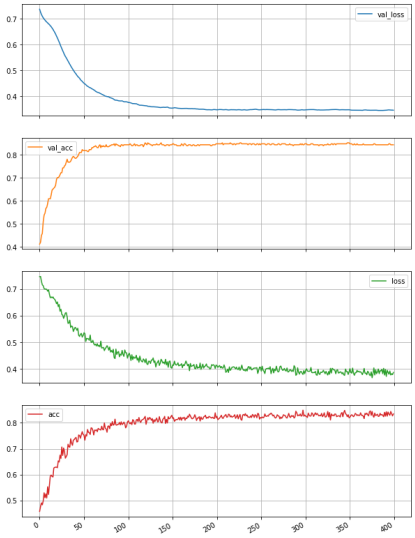

Figure 10: Accuracy

The following hyper parameters were used:

dropout = 0.5
first dense layer nodes = 50
second dense layer nodes = 2
Number of epochs = 400
Model Batch Size = 128
early patience = 100 epochs

Accuracy = 87.3451

5.3 GSC data with feature concatenation

The feature concatenation the results yielded better results than subtraction.

The following hyper parameters were used:

dropout = 0.5
first dense layer nodes = 100
second dense layer nodes = 2
Number of epochs = 800
Model Batch Size = 2048
early patience = 100 epochs

Accuracy = 98.677777

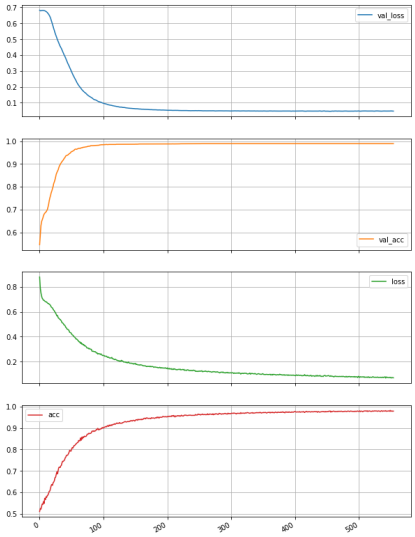

Figure 11: Accuracy

5.4 GSC data with feature subtraction

When compared to the feature concatenation the results were lower

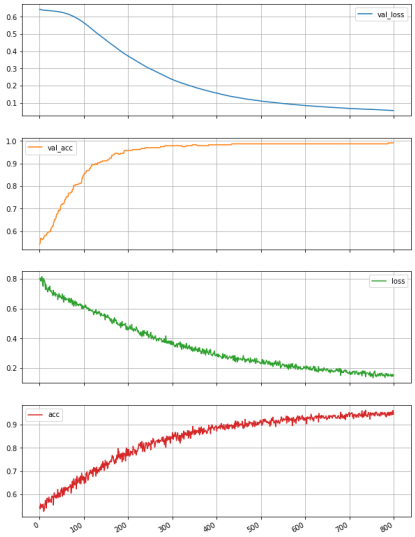

Figure 12: Accuracy

The following hyper parameters were used:

dropout = 0.5
first dense layer nodes = 100
second dense layer nodes = 2
Number of epochs = 800
Model Batch Size = 2048
early patience = 100 epochs

Accuracy = 94.333333

6 Findings

1. The results were better with the use of neural network. The second best was Logistic Regression. Linear Regression did not yield good results.
2. The accuracy while predicting from GSC dataset was better than that of Human observed data because of the large number of samples and the number of features.
3. The concatenation technique works better than the subtraction technique.