Name: Saurabh Kumar Chauhan

UBName: Chauhan9

UBId: 50290975

Project 2: Similarity Prediction of handwritten samples

Introduction:

In this project we are going to predict the similarity between two handwritten samples, whether both are similar or not, for this we are going to use 3 techniques we have learn so far.

Materials:

1. Provided an 2 data set(feature information) in the form of CSV file
   a. HumanObserved-Features-Data.csv
      It has 9 feature values set for every image observed by human.
   b. GSC-Features.csv
      It has 512 feature values set for every image observed by GSC.
2. Provide collection of set of images with the information of similar sample images and different sample images, which basically gives us the target values of 1 and 0 based on the similarity between the set of pair of sample images.

Approach:

We are going to try to solve this problem as regression problem as well as classification problem. So we are going to implement it using 3 techniques as

a. Linear regression (Regression
b. Logistic regression (Classification)
c. Neural network (Classification)

But before we apply any of these techniques we are going to prepare the provided data into four different sets by manipulating the feature values of different pair and similar pair, to get four set of data instead of just two and will perform all 3 techniques on these 4 feature sets.

Data Preparation:

We are going to prepare 4 data sets using different and same pair sets, using subtraction and concatenation on feature values of each type of data.

- prepare the basic feature set
  1. Human Observed Dataset with feature Subtraction
     a. Load up the feature values for images from csv file (HumanObserved-Features-Data.csv)
     b. Load up the same pair set of images and same number of records from different pair set randomly.
     c. Jumble the resultant set so records don't get align by target values
     d. Load up features of each images and subtract the feature value of one image from another image's feature value and take absolute value so we don't have negative values.
     e. The resultant set will have 9 features.
     f. Resultant feature set is one of the dataset we will perform all 3 techniques.

  2. Human Observed Dataset with feature concatenation
     a. After step c from above data preparation, instead of subtracting feature values, concatenate all the feature values to create features size of 18.
     b. The resultant set will have 18 features.
     c. Resultant feature set is one of the dataset we will perform all 3 techniques.

  3. GSC Dataset with feature subtraction

a. Load up the feature values for images from csv file (GSC-Features-Data.csv)
b. Load up the same pair set of images and same number of records from different pair set randomly.
c. Jumble the resultant set so records don't get align by target values
d. Load up features of each images and subtract the feature value of one image from another image's feature value and take absolute value so we don't have negative values.
e. The resultant set will have 512 features.
f. Resultant feature set is one of the dataset we will perform all 3 techniques.

4. GSC Dataset with feature concatenation
   a. After step c from above data preparation, instead of subtracting feature values, concatenate all the feature values to create features size of 1024.
   b. The resultant set will have 1024 features.
   c. Resultant feature set is one of the dataset we will perform all 3 techniques.

- Polish the data so we don't have issue with the implementation.
  1. For all 4 data set, check variance of all the feature values, and remove the column with 0 variance.
- Push all four feature set into a list for we can work with every dataset for all 3 techniques.
- Now we are ready for implementation.

Implementation:

1. Linear Regression

We are going to reuse the code from project 1.2 to perform the linear regression. We will push all 4 sets of data one by one into our linear regression implementation. We will at the result one by one for close form and gradient descent.

a. Close form (ERMS values for each set training, validation and testing)

- Human Observed Dataset with feature Subtraction

```
E_rms Training   = 0.4984457302074536
E_rms Validation = 0.49767630523249085
E_rms Testing    = 0.499420523380488
```

- Human Observed Dataset with feature concatenation

```
E_rms Training   = 0.4959352087292521
E_rms Validation = 0.5038773841539148
E_rms Testing    = 0.49617536753661834
```

- GSC Dataset with feature subtraction

```
E_rms Training   = 0.4983889858658336
E_rms Validation = 0.4998810090733542
E_rms Testing    = 0.5014478404743594
```

- GSC Dataset with feature concatenation

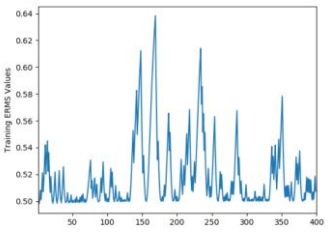
```
E_rms Training   = 0.4978638202677722
E_rms Validation = 0.5045611597847816
E_rms Testing    = 0.49820091409436307
```

b. Gradient Descent (ERMS values for each training, validation and testing set)

- Human Observed Dataset with feature Subtraction

```
E_rms Training   = 0.49859
E_rms Validation = 0.4978
E_rms Testing    = 0.49948
```

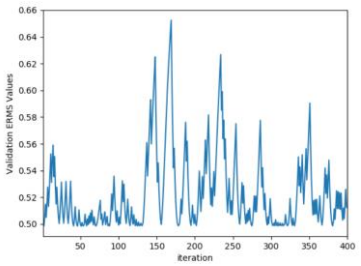
  - Graph between ERMS and iterations



- Human Observed Dataset with feature concatenation

```
E_rms Training   = 0.49641
E_rms Validation = 0.49895
E_rms Testing    = 0.49673
```

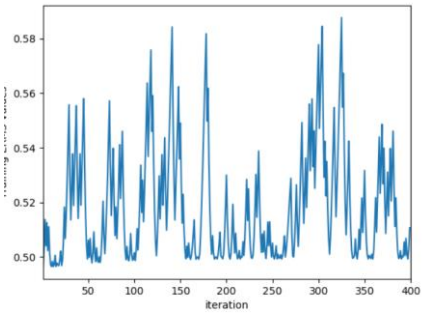
  - Graph between ERMS and iterations



- GSC Dataset with feature subtraction

```
E_rms Training   = 0.49853
E_rms Validation = 0.49996
E_rms Testing    = 0.5002
```

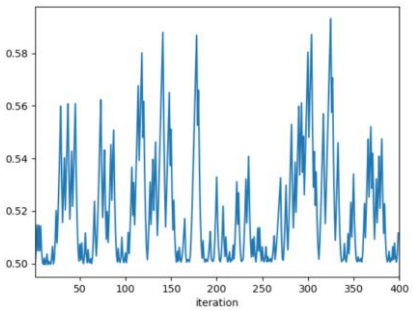
  - Graph between ERMS and iterations



- GSC Dataset with feature concatenation

```
E_rms Training   = 0.49799
E_rms Validation = 0.50018
E_rms Testing    = 0.49854
```

  - Graph between ERMS and iterations



Observations:

1. We can observe the ERMS values for all the dataset are very close to each other.
2. Linear regression work similarly in case of all the data set.

2. Logistic Regression:

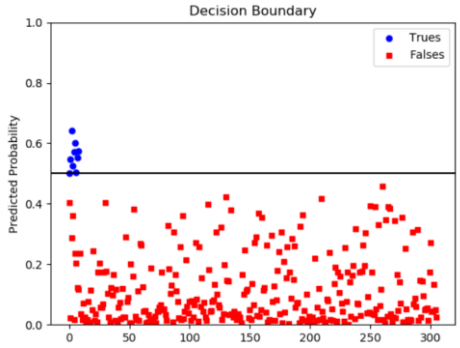
We will be performing another technique which is logistic regression it's a classification technique, which uses sigmoid to predict the probabilities of the given set and then based on that probability we decide if it belong to class 0 or class 1 (any label you can select to define class).

1. We have used sigmoid to make sure that prediction are based on probabilities and they always range between 0 and 1.
2. Then based on the values higher than 0.5 we will classify it as class 1, other class 0
3. Here we have decided our boundary at 0.5

4. Based on the predicted class we can compare the accuracy of our model by comparing the predicted target value to actual target values.
5. Accuracy for each data set.
    - Human Observed Dataset with feature Subtraction

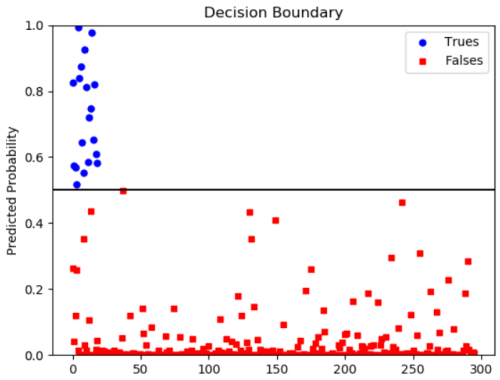
      ```
      accuracy = 50.48%
      ```

        - Scatter graph based on probability predicted



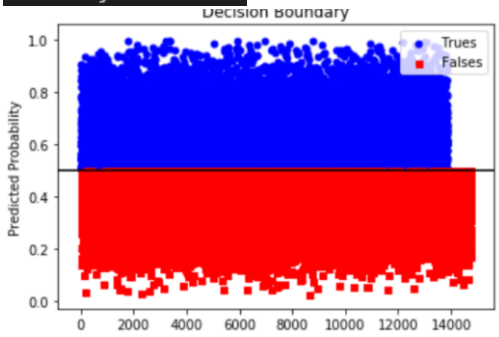
    - Human Observed Dataset with feature concatenation

      ```
      accuracy = 51.43%
      ```



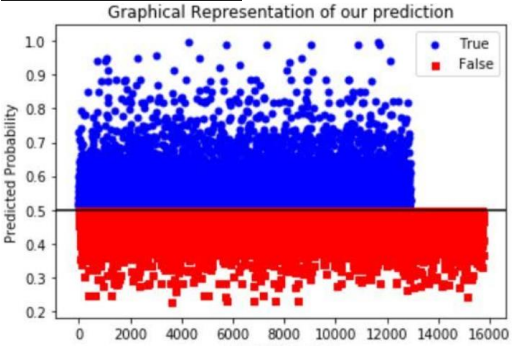
    - GSC Dataset with feature subtraction

      ```
      accuracy = 73.83%
      ```



    - GSC Dataset with feature concatenation

      ```
      accuracy = 76.34%
      ```



Observations:

1. During logistic regression we have notice the prediction tends to be more accurate if we have more data as we can see the prediction in Human observe set has lower accuracy.
2. Another observation with less data is that, model tend to predict one type of class more often or it tends to classify object to a single class if we don't have large data.
3. That's why for GSC we had better accuracy and scattered graph is also filled both type of classes.

3. Neural network

We are going to use the code base as we had in project 1.1 and going modify it to meet the requirement of current data set.

It's also a classification approach where we will decide whether an object a similar to different to another one.

1. I have used different optimizers and activation function to train the model like relu, tanh etc and gradient descent, adam etc.
2. We have classified our output as two classes.
3. Input layer for the neural network I have feed as feature values, which will different for every set based on the feature values.
6. Accuracy for each data set.

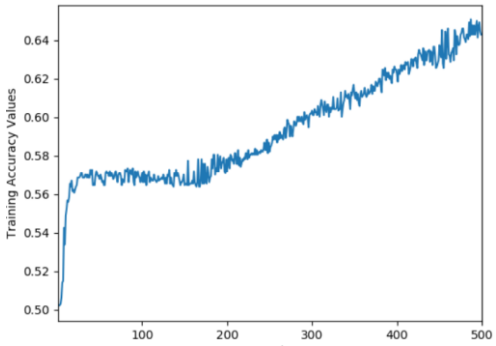
   Hyperparamters :

   a. GradientDescentOptimizer
   b. Relu activation function
   c. 500 Epochs
   d. Learning rate 0.01

- Human Observed Dataset with feature Subtraction

```
Testing Accuracy: 58.943434
```

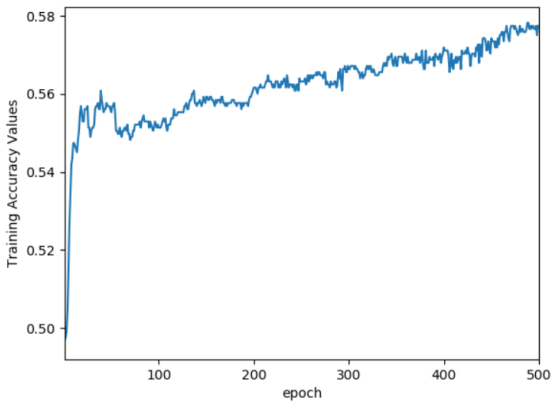
  - graph between training Accuracy and Epochs



- Human Observed Dataset with feature concatenation

```
Testing Accuracy: 54.3456334
```

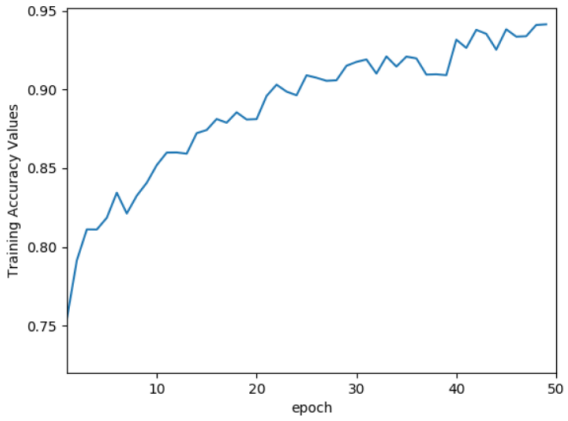
  - graph between training Accuracy and Epochs



- GSC Dataset with feature subtraction

```
Testing Accuracy: 61.965343
```

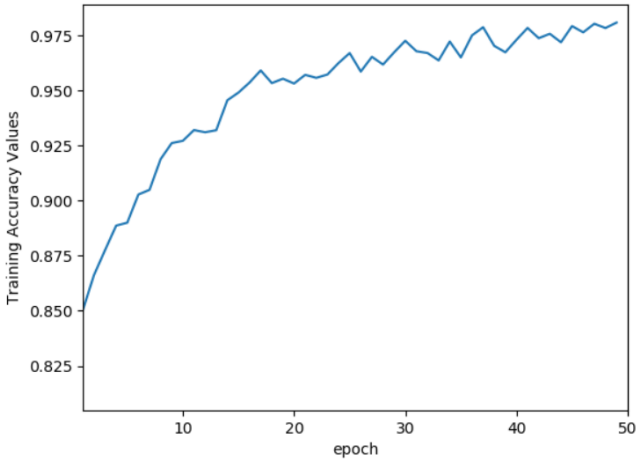
  - graph between training Accuracy and Epochs



- GSC Dataset with feature concatenation

```
Testing Accuracy: 65.324545
```

  - graph between training Accuracy and Epochs



Observations:

1. I have observed the testing accuracy not increasing after try different combination of hyper parameters.
2. For GSC the training accuracy is very high as we train our model but testing accuracy does not match up with it, the data for GSC may be getting over fitted, tried to change it by tuning hyper parameters, this is the best accuracy I got.
3. Another observation model training is too unpredictable as with the same hyper parameter setting getting different accuracy by great factor
4. For human observe dataset we don't have enough data to to train the model correctly so training accuracy itself is not increasing