# Project 2.0

**UBIT : mihirraj**

**#No : 50290849**

**Objective:**

The objective of this task is to find similarity between handwritten samples of the known and question writers using: Linear Regression, Logistic Regression, Neural Networks.
We have two datasets:
1. Human Observed features: Features entered by human document examiners manually
2. GSC features: Features extracted using Gradient Structural Concavity (GSC) algorithm.
Every dataset has a specific number of features and a target value {0,1}.
Our goal is to predict the correct target value given a set of features.

**Approach:**

1. Data Preparation:
   We have been given 3 CSV files for each dataset and we have to prepare the dataset.
   And for each dataset we have to prepare two types of datasets
      a. Feature Concatenation
      b. Feature Subtraction
2. Implementation:
   We have three different approaches to solve this problem.
   1. Linear Regression
   2. Logistic Regression
   3. Neural Networks

**Dataset Preparation:**

1. Human Observed Dataset:
   We have 3 CSV files namely – same_pairs, diffn_pairs, HumanObserved-Features-Data.
   These files contain a same pairs of writers, different pairs of writers, Features of every single writer.

   a. Feature Concatenation:
      For every writer in the “same_pairs.csv” we fetch the features. Similarly, for every writer in “diffn_pairs.csv” we fetch the features. Then, we append these two together which gives us our final dataset with Feature Concatenation.
      This dataset has 21 columns (2 img ids, 18 features, 1 target).
      The number of rows in “same_pairs.csv” is 791 and hence for “diffn_pairs.csv” we take the same number of rows.
      Therefore, our final dataset looks like – (1582 X 21)

b. Feature Subtraction
   For every writer in the “same_pairs.csv” we fetch the features. Similarly, for every writer in “diffn_pairs.csv” we fetch the features as in (a). But here, we subtract the features of each writer resulting in only 9 columns giving us Feature Subtraction
   This dataset has 12 columns (2 img ids, 9 features, 1 target).
   The number of rows in “same_pairs.csv” is 791 and hence for “diffn_pairs.csv” we take the same number of rows.
   Therefore, our final dataset looks like – (1582 X 12)

2. GSC Dataset:
   For GSC Dataset also we have 3 csv files same as earlier – Same pairs, Different pairs, GSC-Dataset-Features. Using these 3 files we create our final dataset. Since the number of writers in this dataset is approximately close to 71K for Same Pairs and 700K for Different pairs, we only choose 2500 writers from both the files for the ease of implementation.

   a. Feature Concatenation:
      For every writer in the “same_pairs.csv” we fetch the features. Similarly, for every writer in “diffn_pairs.csv” we fetch the features. Then, we append these two together which gives us our final dataset with Feature Concatenation.
      This dataset has 1027 columns (2 img ids, 1024 features, 1 target).
      We take 2500 rows from both same and different pair files.
      Therefore, our final dataset looks like – (5000 X 1027)

   b. Feature Subtraction:
      For every writer in the “same_pairs.csv” we fetch the features. Similarly, for every writer in “diffn_pairs.csv” we fetch the features as in (a). But here, we subtract the features of each writer resulting in only 512 columns giving us Feature Subtraction
      This dataset has 515 columns (2 img ids, 512 features, 1 target).
      We take 2500 rows from both same and different pair files.
      Therefore, our final dataset looks like – (5000 X 515).

**Data Preprocessing:**
The dataset may contain features with variance = 0, we eliminate such features by dropping them. The size of all the datasets might reduce as per it. We also add small noise to avoid this problem.

**Implementation:**

1. **Linear Regression**:

We have used Stochastic Gradient Descent to implement Linear Regression as seen in Project1.1. To implement that we are finding "PHI MATRIX" for a fixed value of M (Basis functions). First, we need to decide on the number of radial basis functions. For this we are using the "Elbow-Method"

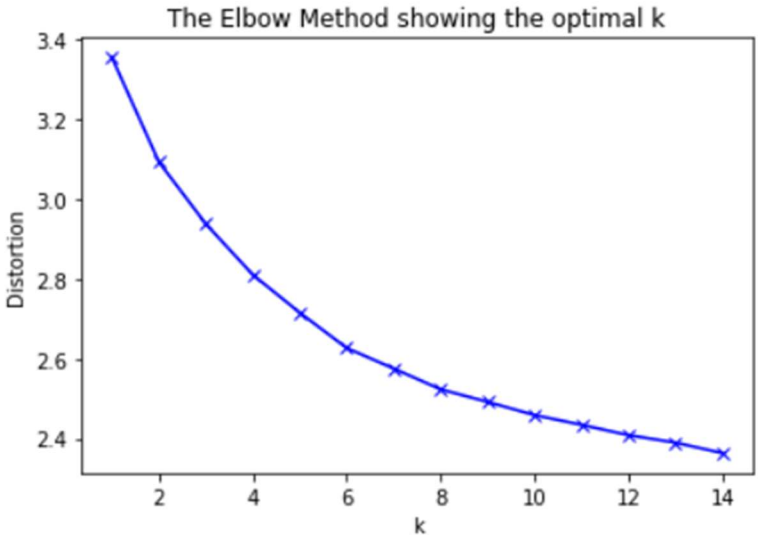


Fig: Elbow Method

We have chosen M = 6 after looking at the graph above. Using this, we find the "PHI MATRIX" and Weights using Closed Form Equation. Using this we then solve Linear Regression using Gradient Descent.

Linear Regression is implemented for all 4 datasets and the following results can be seen:

a. Human Observed Dataset with Feature Concatenation:
   M = 6
   Regularization constant = 0.5
   Learning Rate = 0.2

   Output:

```
Human Observed Dataset with Concatenation selected
(1582, 18)
----------Gradient Descent Solution--------------------
E_rms for Human Observed Dataset with feature concatenation
E_rms Training   = 0.35639
E_rms Validation = 0.36046
E_rms Testing    = 0.35066
```

b. Human Observed Dataset with Feature Subtraction:
   M = 6
   Regularization constant = 0.5

Learning Rate = 0.2

Output:

```
Human Observed Dataset with Subtraction selected
----------Gradient Descent Solution--------------------
E_rms for Human Observed Dataset with feature subtraction
E_rms Training   = 0.45775
E_rms Validation = 0.42991
E_rms Testing    = 0.4342
```

c. GSC Dataset with Feature Concatenation:

M = 6

Regularization constant = 0.5

Learning Rate = 0.2

Output:

```
GSC Dataset with feature concatenation
(940, 5000)
----------Gradient Descent Solution--------------------
E_rms for GSC Dataset with feature concatenation
E_rms Training   = 0.38227
E_rms Validation = 0.40908
E_rms Testing    = 0.38213
```

d. GSC Dataset with Feature Subtraction:

M = 6

Regularization constant = 0.5

Learning Rate = 0.2

Output:

```
GSC Dataset with feature subtraction
(474, 5000)
----------Gradient Descent Solution--------------------
E_rms for Human Observed Dataset with feature subtraction
E_rms Training   = 0.41219
E_rms Validation = 0.42542
E_rms Testing    = 0.38851
```

2. Logistic Regression:

We implement logistic regression using the following equations:

$Z = W^T * X$

Y = sigmoid(Z) (Sigmoid activation function)
E = (Y-T) * X (Gradient of loss function)
W = W – (d * E)

Where,
W – initial weights
Y – Predicted output
T – Target
E – Error
d – Learning Rate

Results:

a. Human Observed Dataset with feature concatenation:

```
Human Observed Dataset with Concatenation selected

Accuracy measures:
Training data accuracy 0.9549763033175356
Validation data accuracy 0.9367088607594937
Test data accuracy 0.9554140127388535

Sk learn accuracy is calculated only to cross check the model
performance

Accuracy from sk-learn on training: 0.9605055292259084
Accuracy from sk-learn on validation: 0.9430379746835443
Accuracy from sk-learn on testing: 0.9490445859872612
```

We can see that the model performs well for given dataset.

b. Human Observed Dataset with feature concatenation:

.

```
Human Observed Dataset with Subtraction selected

Accuracy measures:
Training data accuracy 0.6216429699842022
Validation data accuracy 0.620253164556962
Test data accuracy 0.6751592356687898

Sk learn accuracy is calculated only to cross check the model
performance

Accuracy from sk-learn on training: 0.8199052132701422
Accuracy from sk-learn on validation: 0.7784810126582279
Accuracy from sk-learn on testing: 0.8535031847133758
```

We can see that the model performs considerable for given dataset.

c. GSC Dataset with feature concatenation:

```
GSC Dataset with Feature concatenation selected

Accuracy measures:
Training data accuracy 1.0
Validation data accuracy 0.9799599198396793
Test data accuracy 0.9839679358717435

Sk learn accuracy is calculated only to cross check the model
performance
Accuracy from sk-learn on training: 1.0
Accuracy from sk-learn on validation: 0.9719438877755511
Accuracy from sk-learn on testing: 0.9559118236472945
```

We can see that the model performs really good for the given dataset.

d. GSC Dataset with feature subtraction:

```
GSC Dataset with Feature Subtraction selected

Accuracy measures:
Training data accuracy 0.54425
Validation data accuracy 0.4649298597194389
Test data accuracy 0.5050100200400801

Sk learn accuracy is calculated only to cross check the model
performance
Accuracy from sk-learn on training: 0.63675
Accuracy from sk-learn on validation: 0.5270541082164328
Accuracy from sk-learn on testing: 0.4789579158316633
```

We can see that the model performs really good for the given dataset.

3. Neural Networks:

Now, we implement Neural Network on the same dataset created earlier using Keras library function.
Configuration used for Neural Network:
Input_size = Number of features (9-18-512-104) as per the dataset.
The network has 1 input layer, 2 hidden layers and 1 output layer.

The nodal configuration for the layers is:
first_dense_layer_nodes = 256
hidden_dense_layer = 512
second_dense_layer_nodes = 1

Activation functions used:
“Relu” for hidden layers and “Sigmoid” for Output layer.

Loss function: Binary_crossentropy as the problem is to classify dataset into two categories {0,1} which is a binary classification problem.
Optimizer used is “Adam”.

Results for all the datasets:

a. Human Observed Dataset with Feature Concatenation:

```
Human Observed Dataset with Concatenation selected
(1582, 18)
```

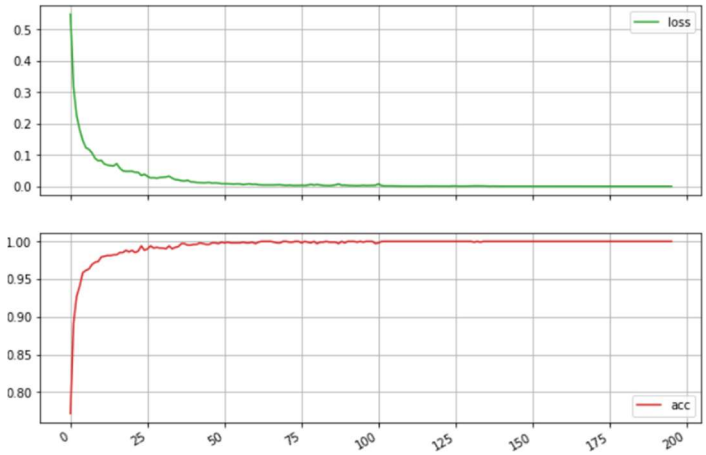


Fig: Loss & accuracy

Testing accuracy:

```
Errors: 0  Correct :157
Testing Accuracy: 100.0
```

b. Human Observed Dataset with Feature Subtraction:

```
Human Observed Dataset with Subtraction selected
(1582, 9)
```

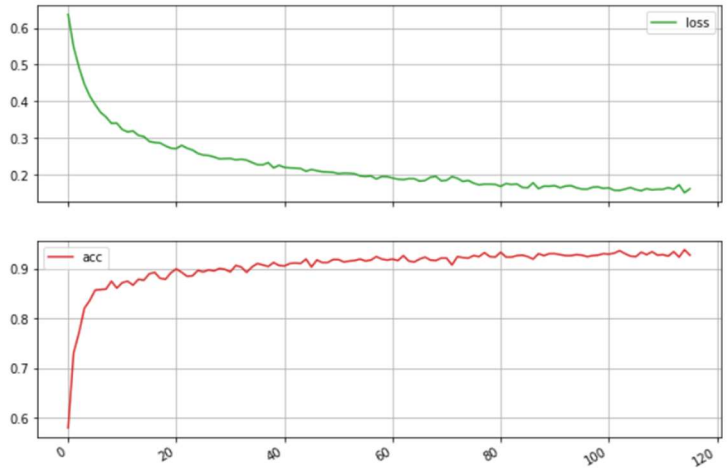


Fig: Loss & accuracy

Testing accuracy:

```
Errors: 0  Correct :157
Testing Accuracy: 100.0
```

c. GSC Dataset with Feature Concatenation:

```
GSC Dataset with feature concatenation
(940, 5000)
```

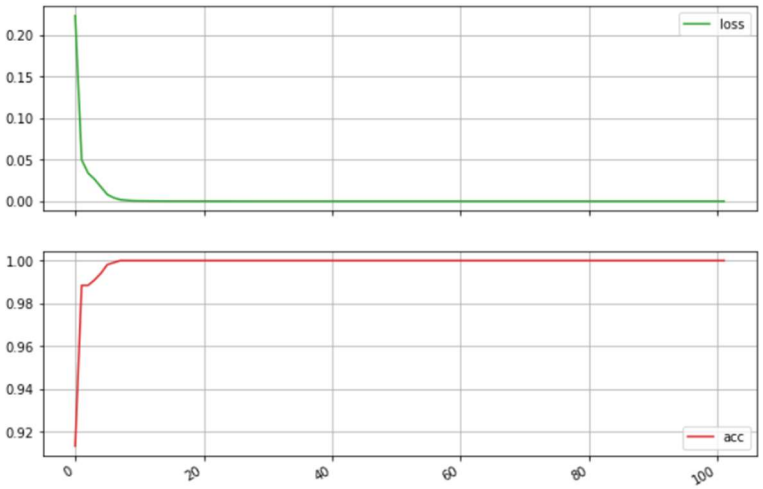


Fig: Loss & accuracy

Testing accuracy:

```
Errors: 0  Correct :499
Testing Accuracy: 100.0
```

d. GSC Dataset Feature Subtraction:

```
GSC Dataset with feature concatenation
(474, 5000)
```

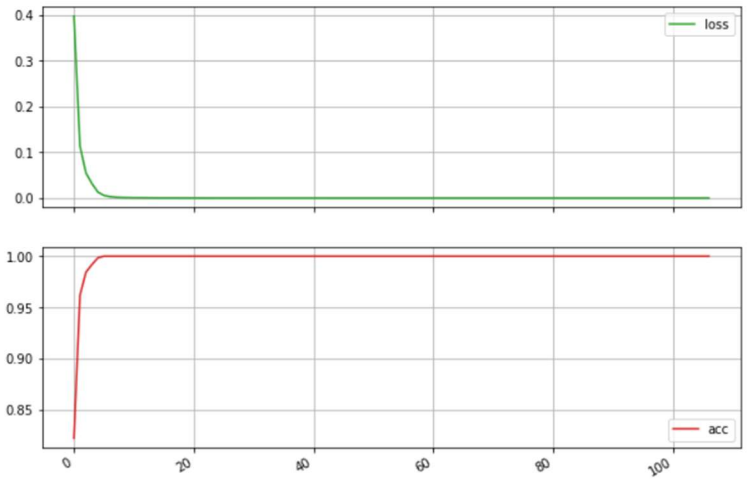


Fig: Loss & accuracy

Testing accuracy:

```
Errors: 0  Correct :499
Testing Accuracy: 100.0
```

**Conclusion:**

We have implemented the given problem using different algorithms. On comparing the results, we can conclude with the following observations:

- The accuracy or performance for every model depends on the dataset used. That is, the problem of Seen-Unseen writer can affect the model performances.
- Linear Regression has the lowest performance as it doesn't predict discrete classes but gives continuous values between 0 to 1.
- Logistic Regression performs well as the sigmoid function can classify the dataset into two linearly separable classes.
- Neural Network has the best performance as compared to the other two algorithms.