CSE 574: Introduction to Machine Learning

Amlan Gupta
#50288686
amlangup@buffalo.edu

Abstract

The project requires to apply machine learning to solve the handwriting comparison task in forensics using CEDAR dataset. We use linear regression, logistic regression and neural network where we map a set of input features x to a real-valued scalar target y(x,w). Using the output of three model we compare how appropriate they are to be used as a solution to this kind. The general objective of this project is to identify if a pair of handwriting sample is written by same writer or not.

1 What is CEDAR letter dataset?

The CEDAR letter dataset consisting of lines of text, handwritten on a writing tablet by approximately 200 writers, and stored in on-line format. The total number of words contained in the database is 105,573. The database contains both cursive and printed writing, as well as some writing which is a mixture of cursive and printed. Because it contains entire lines of text, instead of just individual words, the database will be useful for studying word separation and recognizing words in context, as well as for general on-line word recognition.

2 Data Preparation

Our dataset uses "AND" images samples extracted from CEDAR Letter dataset. Image snippets of the word "AND" were extracted from each of the manuscript using transcript-mapping function of CEDAR-FOX. Based on feature extraction process, two datasets are there to train our models.

2.1 Human Observed Dataset

The Human Observed dataset shows only the cursive samples in the data set. A human handwriting expert analyzed individual samples and noted 9 different features for each sample.

The entire dataset consists of 791 same writer pairs and 293,032 different writer pairs.

For training out models we take 791 same writer pairs and randomly picking 791 different data pair to avoid over-fitting. So number of total pairs we will be working on is 1582 for Human Observed dataset.

We have to train our models using two settings.

2.1.1 Feature Concatenation

As each image sample has 9 features and we are creating pairs, we are concatenating the features side by side. So for 1582 entries, we will have 1582 x 18 feature matrix to work on.

2.1.2 Feature Subtraction

As each image sample has 9 features and we are creating pairs, we are subtracting second image's features from first image's features then noting the absolute value to keep the difference between each features. So for 1582 entries, we will have 1582 x 9 feature matrix to work on.

2.2 Gradient Structural Concavity Dataset

Gradient Structural Concavity algorithm generates 512 sized feature vector for an input handwritten "AND"image. The entire dataset consists of 71,531 same writer pairs and 762,557 different writer pairs.

For training out models we take 71,531 same writer pairs and randomly picking 71,531 different data pair to avoid over-fitting. So number of total pairs we will be working on is 1,43,062 for GSC dataset.

We have to train our models using two settings.

2.2.1 Feature Concatenation

As each image sample has 512 features and we are creating pairs, we are concatenating the features side by side. So for 1,43,062 entries, we will have 143062 x 1024 feature matrix to work on. After deleting features with no variation the final feature matrix is of 143062 x 1017

2.2.2 Feature Subtraction

As each image sample has 512 features and we are creating pairs, we are subtracting second image's features from first image's features then noting the absolute value to keep the difference between each features. So for 1,43,062 entries, we will have 143062 x 512 feature matrix to work on. After deleting features with no variation the final feature matrix is of 143062 x 509

3 Results

Out of the total datapoint 10% has been used for testing, 10% has been used for validation and the remaining 80% has been used to train the models.

Table 1: Training Dataset Accuracy

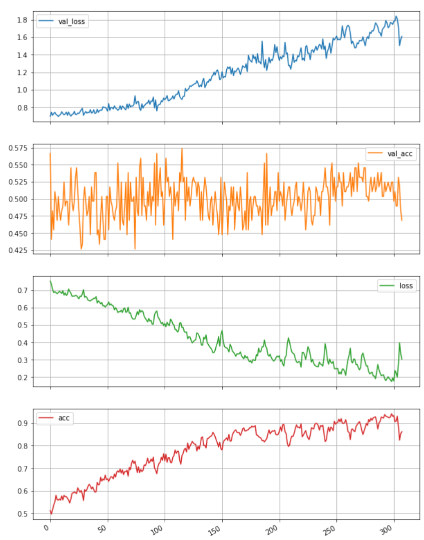
Data	Linear Regression	Logistics Regression	Neural Network
HO Data - Concatenation	54.89731	57.26698	93.05
HO Data - Subtraction	52.05371	50.7109	83.5
GSC Data - Concatenation	52.05242	50.55292	99.95
GSC Data - Subtractiion	50.16339	49.96418	74.3

Table 2: Validation Dataset Accuracy

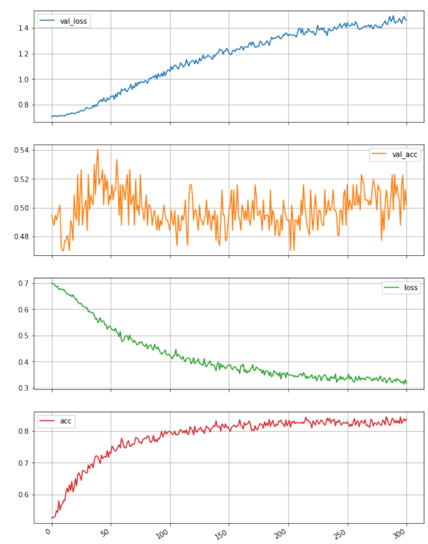
Data	Linear Regression	Logistics Regression	Neural Network
HO Data - Concatenation	64.55696	57.59494	57.75
HO Data - Subtraction	58.22785	52.53165	54.1
GSC Data - Concatenation	51.86635	52.53165	78.2
GSC Data - Subtractiion	50.74095	50.51727	52.04

Table 3: Testing Dataset Accuracy

Data	Linear Regression	Logistics Regression	Neural Network
HO Data - Concatenation	57.96178	66.50955	48.73
HO Data - Subtraction	59.23567	55.40764	49.36
GSC Data - Concatenation	51.96784	57.40764	77.6
GSC Data - Subtractiion	51.0381	54.77281	50.8

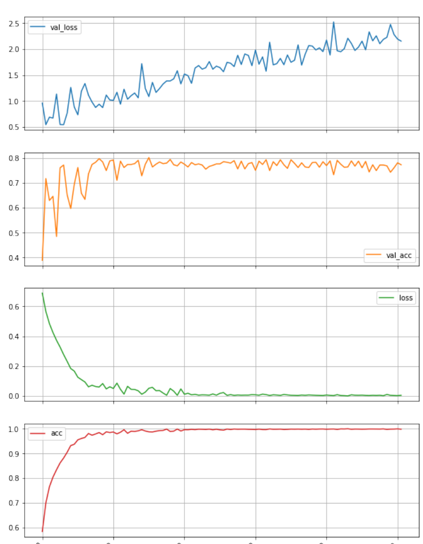


(a) Feature Concatenation

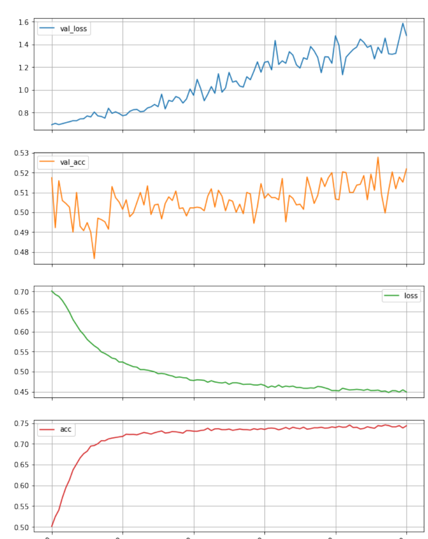


(b) Feature Subtraction

Figure 1: Neural Network performance for Human Observed dataset



(a) Feature Concatenation



(b) Feature Subtraction

Figure 2: Neural Network performance for GSC dataset

4 Hyper-parameter Tuning

a Linear Regression:

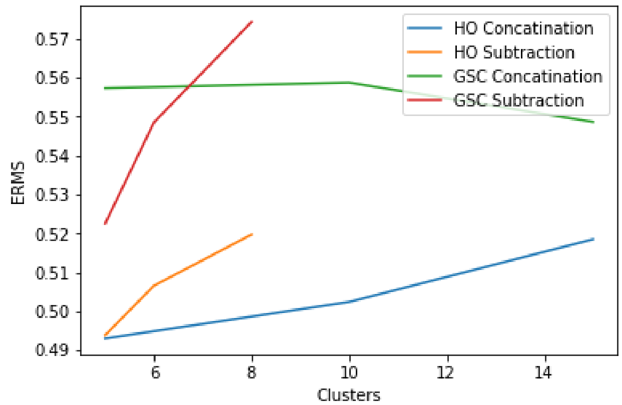


Figure 3: Change of EMS due to cluster change

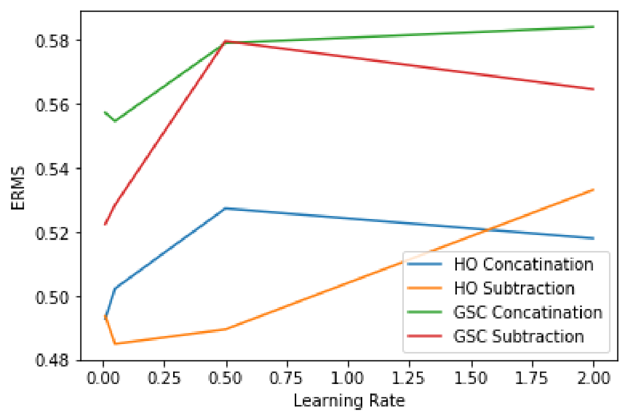


Figure 4: Change of ERMS due to learning rate change

b Logistic Regression:

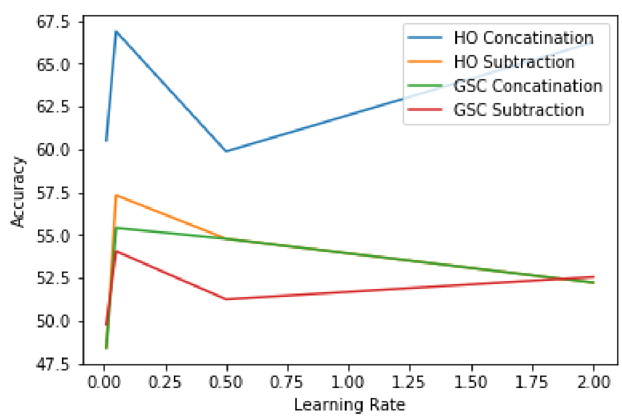


Figure 5: Change of Accuracy due to Learning rate change

5 Conclusion

From the tables we can conclude tat neural network is generating better accuracy compared to other models. Logistic regression is also working better than Linear Regression for this use case.

GSC dataset will perform better than Human Observed dataset as there are more features available for a sample. Due to limited environment capability, it was not possible to train the model with full available dataset. A sample subset was used to generate the tables. Processing the full feature matrix should consistently give better results for GSC dataset.

References

[1] andrew Ng. Machine learning. Coursera, 2012.

[2] Prof. K. Ferens. *Vectorized Implementation of Logistic Regression*. U of Manitoba, 2017.