Project 2

Intro to Machine Learning

Illa Nuka Saranya
50248926

01-Nov-18
Saranya@buffalo.edu

Contents

1. PROBLEM DEFINITION

The goal of this project is to use machine learning to solve a problem that finds similarity between the handwritten samples of the known and the questioned writer by using linear and logistic regression.

DATASET:

Each instance in the CEDAR "AND" training data consists of set of input features for each handwritten "AND" sample. The features are obtained from two different sources:

1. Human Observed features: Features entered by human document examiners manually

2. GSC features: Features extracted using Gradient Structural Concavity (GSC) algorithm

TASKS:

We preprocess the given above dataset and perform each of the following tasks:

1. Linear Regression on each of the 4 datasets:

(a) Human Observed Dataset with feature concatentation

(b) Human Observed Dataset with feature subtraction

(c) GSC Dataset with feature concatentation

(d) GSC Dataset with feature subtraction

2. Logistic Regression on each of the 4 datasets:

(a) Human Observed Dataset with feature concatentation

(b) Human Observed Dataset with feature subtraction

(c) GSC Dataset with feature concatentation

(d) GSC Dataset with feature subtraction

2. IMPLEMENTATION

2.2 Preprocessing of the Datasets:

2.2.1 Preprocessed Datasets (Given files)

2.2.1.1 Human observed dataset:

- All of the hand-written samples are examined by the document examiner and 9 human engineered features are extracted. HumanObserved-Features-Data.csv contains this data.
- The above file is preprocessed a bit to produce two more files namely same_pairs and diffn_pairs .

- Same_pairs.csv consists of 791 pairs of data samples written by same person which on comparison gives the target value of 1 whereas diffn_pairs.csv consists of 293033 pairs of data samples written by different persons that give target value as 0 when compared.

2.2.1.2 GSC dataset:

- Gradient Structural Concavity algorithm generates 512 sized feature vector for an input handwritten "AND"image. The dataset is named as "GSC-Features-Data".
- The above file is preprocessed a bit to produce two more files namely same_pairs and diffn_pairs .
- Same_pairs.csv consists of 71531 pairs of data samples written by same person which on comparison gives the target value of 1 whereas diffn_pairs.csv consists of 7,62,557 pairs of data samples written by different persons that give target value as 0 when compared.

2.2.2 Preparing the Dataset

2.2.2.1 Human observed dataset:

Two datasets are prepared namely shuffled_concatenated_pairs.csv and shuffled_subtracted_pairs.csv.

- hod_shuffled_concatenated_pairs.csv consists of thoroughly shuffled 791 same paired and 791 different paired data samples along with 18 features for each pair that are obtained by concatenating 9 features from each person.
- hod_shuffled_subtracted_pairs.csv consists of thoroughly shuffled 791 same paired and 791 different paired data samples along with 9 absolute feature values for each pair that are obtained by subtracting (9) features of one person from another in comparison.

2.2.2.2 GSC dataset:

Two datasets are prepared namely gsc_shuffled_concatenated_pairs.csv and gsc_shuffled_subtracted_pairs.csv.

- gsc_shuffled_concatenated_pairs.csv consists of thoroughly shuffled 1000 same paired and 1000 different paired data samples along with 1024 features for each pair that are obtained by concatenating 512 features from each person.
- gsc_shuffled_subtracted_pairs.csv consists of thoroughly shuffled 1000 same paired and 1000 different paired data samples along with 512 absolute feature values for each pair that are obtained by subtracting (512) features of one person from another in comparison.

2.2.3 Fetching the Dataset

2.2.3.1 Human Observed dataset

- From the hod_shuffled_concatenated_pairs.csv and hod_shuffled_subtracted_pairs.csv files,create two vectors namely hodRawConcatenatedTarget and hodRawSubtractedTarget respectively using hodGetTarget method that reads the csv file line by line to generate a list containing only the target values.
    - The length of these lists is 1582 which is (791*2)

- Convert the shuffled_concatenated_pairs.csv and shuffled_subtracted_pairs.csv files into a two data matrices namely hodRawConcatenatedData and hodRawSubtractedData respectively that contains all the columns other than the target values.
    - The shape of these matrices is (18, 1582) and (9, 1582)

2.2.3.2 GSC dataset

- From the gsc_shuffled_concatenated_pairs.csv and gsc_shuffled_subtracted_pairs.csv files,create two vectors namely gscRawConcatenatedTarget and gscRawSubtractedTarget respectively using gscGetTarget method that reads the csv file line by line to generate a list containing only the target values.
    - The length of these lists is 2000 which is (1000*2)
- Convert the gscshuffled_concatenated_pairs.csv and gscshuffled_subtracted_pairs.csv files into a two data matrices namely gscRawConcatenatedData and gscRawSubtractedData respectively that contains all the columns other than the target values.
    - The shape of these matrices is (1024, 2000) and (512, 2000)

2.2.4 Splitting the Dataset

Splitting of the dataset can be done in many ways but here we do the split of training, validation and testing data in continuous way using the percentage values to find the lengths.

2.2.4.1 Human Observed dataset

Both of the shuffled human observed concatenated and subtracted dataset consists of 1582 samples (rows).Intially we, split the dataset in the following percentages

Training Percent=80;

Validation Percent=10;

Test Percent=10;

2.2.4.1.1 *Training Dataset*:

The sample of data used to fit the model from which it sees and learns.

We use 80% of this dataset as Training dataset

Concatenation

The shape of hodConcatenatedTrainingTarget is (1266,)
The shape of hodConcatenatedTrainingDataMatrix is (18, 1266)

Subtraction

The shape of hodSubtractedTrainingTarget is (1266,)
The shape of hodSubtractedTrainingDataMatrix is (9, 1266)

2.2.4.1.2 Validation Dataset:

The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper-parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

We use 10% of our given dataset as validation

Concatenation

The shape of hodConcatenatedValidationTarget is (158,)
The shape of hodConcatenatedValidationDataMatrix is (18, 158)

Subtraction

The shape of hodSubtractedValidationTarget is (158,)
The shape of hodSubtractedValidationDataMatrix is (9, 158)

2.2.4.1.3 Test Dataset:

The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

We use 10% of our given dataset as testing data

Concatenation

The shape of hodConcatenatedTestingTarget is (157,)
The shape of hodConcatenatedTestingDataMatrix is (18, 157)

Subtraction

The shape of hodSubtractedTestingTarget is (157,)
The shape of hodSubtractedTestingDataMatrix is (9, 157)

2.2.4.2 GSC Dataset

Both of the shuffled human observed concatenated and subtracted dataset consists of 2000 samples (rows).Intially we, split the dataset in the following percentages

Training Percent=80;

Validation Percent=10;

Test Percent=10;

2.2.4.2.1 *Training Dataset*:

The sample of data used to fit the model from which it sees and learns.

We use 80% of this dataset as Training dataset

Concatenation

The shape of gscConcatenatedTrainingTarget is (1600,)
The shape of gscConcatenatedTrainingDataMatrix is (1024, 1600)

Subtraction

2.2.4.2.2 Validation Dataset:

The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper-parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

We use 10% of our given dataset as validation

Concatenation

The shape of gscConcatenatedValidationTarget is (200,)
The shape of gscConcatenatedValidationDataMatrix is (1024, 200)

Subtraction

2.2.4.2.3 Test Dataset:

The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

We use 10% of our given dataset as testing.

Concatenation

The shape of gscConcatenatedTestingTarget is (200,)
The shape of gscConcatenatedTestingDataMatrix is (1024, 200)

Subtraction

The next step is to train the linear regression model using the training dataset to find the optimal weights that gives the best fit of solution.

We find this by using Stochastic Gradient Descent Solution

The stochastic gradient descent algorithm first takes a random initial value of weights and keeps on updating the values of it using the following formula

$w^{(\tau+1)} = w^{(\tau)} + \Delta w^{(\tau)}$

$w^{(\tau)}$ -- current weights

$w^{(\tau+1)}$ -- updated weights

$\Delta w^{(\tau)}$ -- $-\eta^{(\tau)}\nabla E$ where η – learning rate, ∇E - gradient error

2.3 Stochastic Gradient Descent Solution for Linear Regression

For linear Regression using Radial Gaussian functions,we have

$\nabla E = \nabla E_D + \lambda \nabla E_W = -(t_n - w^{(\tau)}\phi\ (x_n))\phi\ (x_n)$

$\nabla E_W = w^{(\tau)}$

Where *Φ - is the design matrix and $\Phi_j(x)$ - Basis functions that converts the input vector x into a scalar value. In this project, you are required to use the Gaussian radial basis function*

$$\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$$

- From the above equation, we need to get the phi matrix on training dataset called TRAINING_PHI to find the weights.
- Design matrix consists of scalar values produced by using radial Gaussian functions. The number of Gaussian functions we use is equal to the number of clusters obtained by using k means clustering of dataset.
- Determine the mean and the variance of the clustered datasets to compute the basis function values.

The formula can be modified to

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right)$$

- We find the variance of all the Training Dataset values on every feature.
- That is we have in total of 18 variances which will be the diagonal values of BigSigma. Calculate the inverse of BigSigma as the modified formula suggests.
- By having the mean and inverse BigSigma values, it is easy to compute scalar values of the design matrix using radial Gaussian functions.
- The shape of the design matrix is (1266, 10).

SGD Solution:

- In the process of finding optimal weights using gradient descent solution, set the initial weights randomly.
- Here we set the weights normally distributed random values with certain standard deviation by using random_normal function of tensorflow.
- **Error regularization factor** (lambda) can be treated as a hyper-parameter which is chosen in a way as to reduce the Erms values without losing the generalization of the data.
- **Learning rate decides** how big each update statement would be and choosing it too big could lead to divergence and choosing it too small could lead to intolerably slow convergence.

Update the weights in iterations to find next set of weights using the current set of weights in a way as to minimize the E_{RMS}.

- We choose the number of iterations using the concept of early stopping to avoid overfitting. **Early stopping** is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as **gradient descent**. It works on the assumption that the performance on data can be improved only till certain number of iterations but after that there will be no significant improvement as such.

For every iteration, calculate the Training, validation and testing output by using the design matrix and updated weights.

Also, compute the E_{RMS} values for each of the data split and append these values to the list to find the minimum value.

- The regression line predicts the average y value associated with a given x value. It gives the measure of the spread of the y values around that average. To do this, we use the root-mean-square error (Root mean squared error).
- Calculate the root mean squared errors between actual target values of Validation stored in ValidationTarget with the computed output values (using validation Design matrix and the weight matrix obtained using closed form). Root Mean Square (RMS) error, defined as E_{RMS} $=\sqrt{(\mathbf{2E}(\mathbf{w^*})/\mathbf{N_v}}$ where w^* is the solution and N_V is the size of the test dataset

The possible hyper-parameters that can be changed here are:

- The number of basis functions (No of clusters used in K-Mean clustering) which will change the MuMatrix that has means of different clusters. $(\boldsymbol{\mu})$
- The data split on training, validation and testing on the actual raw dataset.
- Scalar values by which co-variance matrix is multiplied.
- Error Regularization factor
- Learning Rate
- No of iterations(early stopping)

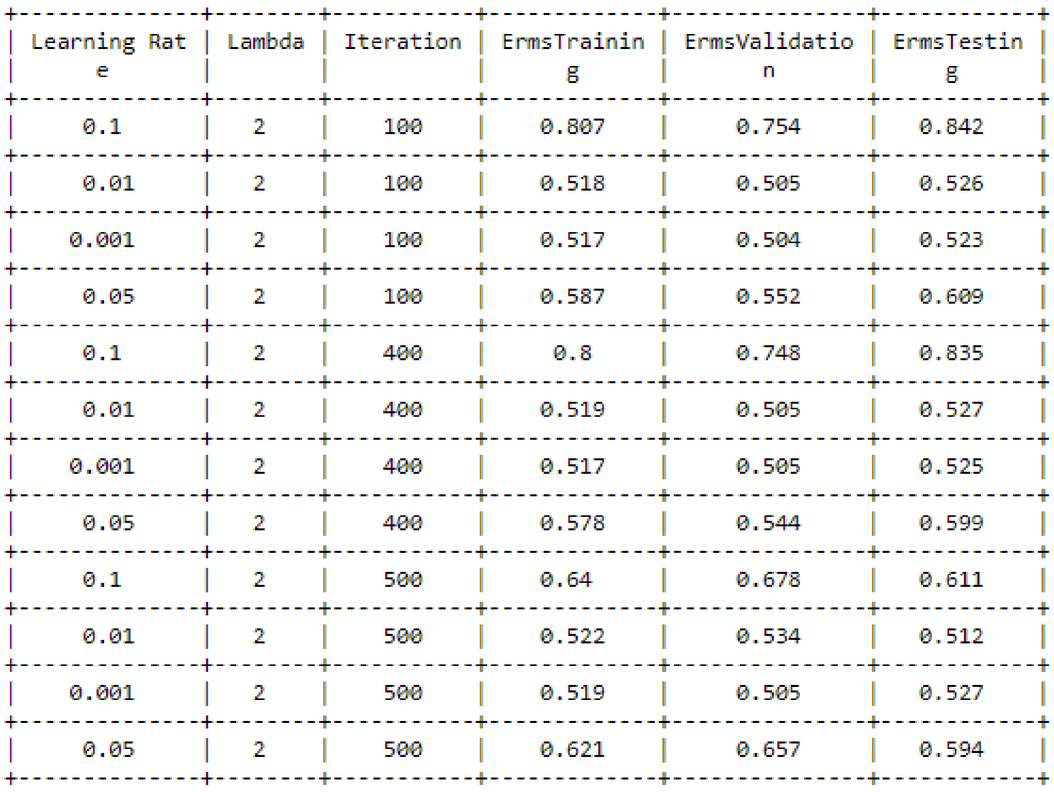
We repeat the same procedure for all the four datasets

2.3.1 Human Observed Dataset with feature concatentation

Tables and graphs

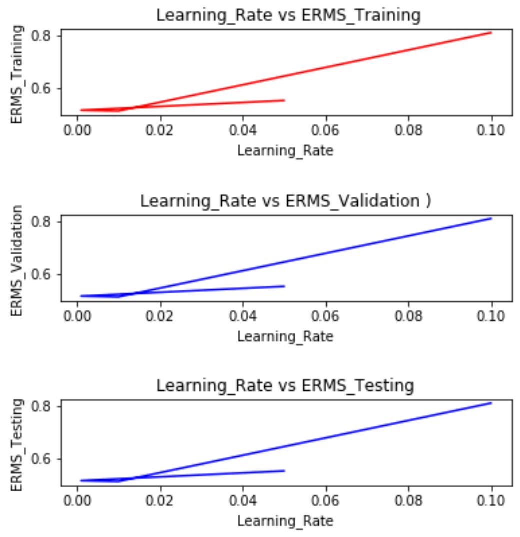
Same_pairs=different_pairs=791,Number of basis functions (M) = 10,Training,Validation,Testing percentages on data =80,10,10 respectively

For lambda = 2 ,Erms value table is as below

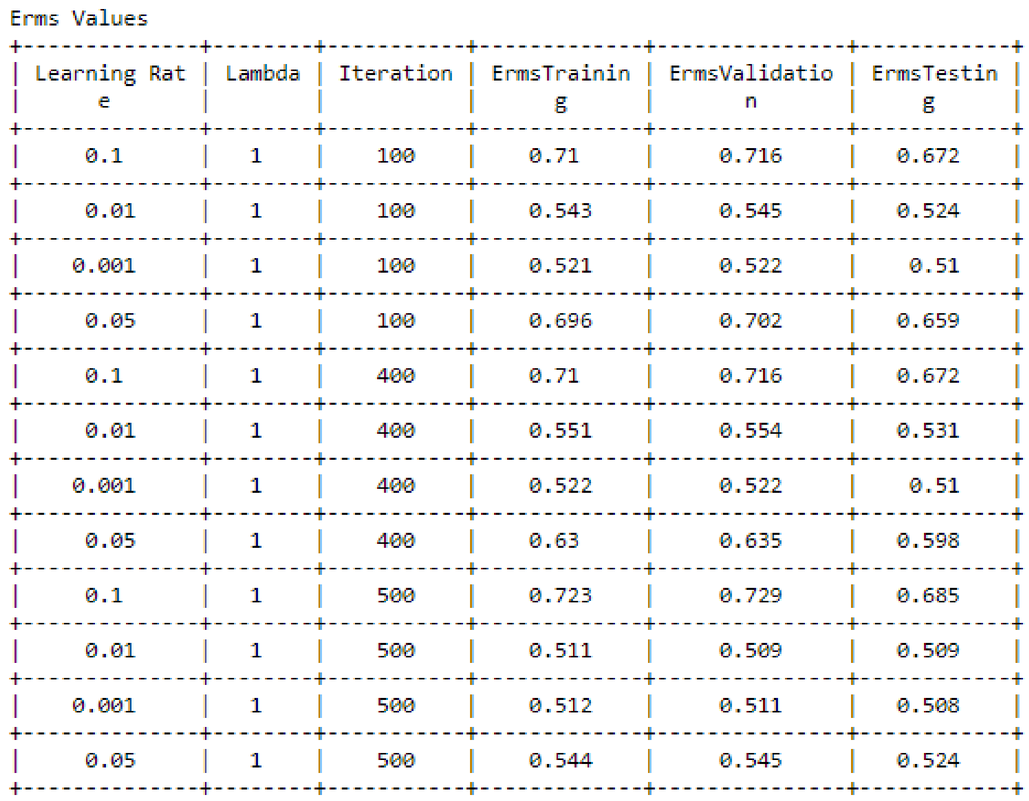


```
+--------------+--------+-----------+-------------+--------------+------------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e       |        |           |      g      |       n      |     g      |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.1      |   2    |    100    |    0.807    |    0.754     |   0.842    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.01     |   2    |    100    |    0.518    |    0.505     |   0.526    |
+--------------+--------+-----------+-------------+--------------+------------+
|    0.001     |   2    |    100    |    0.517    |    0.504     |   0.523    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.05     |   2    |    100    |    0.587    |    0.552     |   0.609    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.1      |   2    |    400    |     0.8     |    0.748     |   0.835    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.01     |   2    |    400    |    0.519    |    0.505     |   0.527    |
+--------------+--------+-----------+-------------+--------------+------------+
|    0.001     |   2    |    400    |    0.517    |    0.505     |   0.525    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.05     |   2    |    400    |    0.578    |    0.544     |   0.599    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.1      |   2    |    500    |    0.64     |    0.678     |   0.611    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.01     |   2    |    500    |    0.522    |    0.534     |   0.512    |
+--------------+--------+-----------+-------------+--------------+------------+
|    0.001     |   2    |    500    |    0.519    |    0.505     |   0.527    |
+--------------+--------+-----------+-------------+--------------+------------+
|     0.05     |   2    |    500    |    0.621    |    0.657     |   0.594    |
+--------------+--------+-----------+-------------+--------------+------------+
```

Number of iterations=500

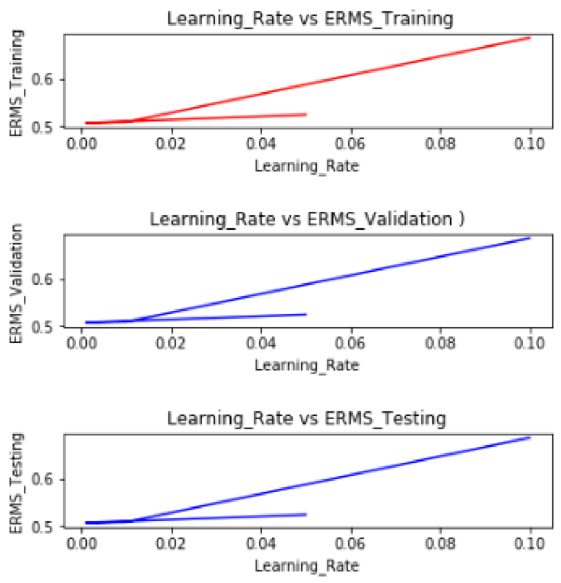


For lambda = 1

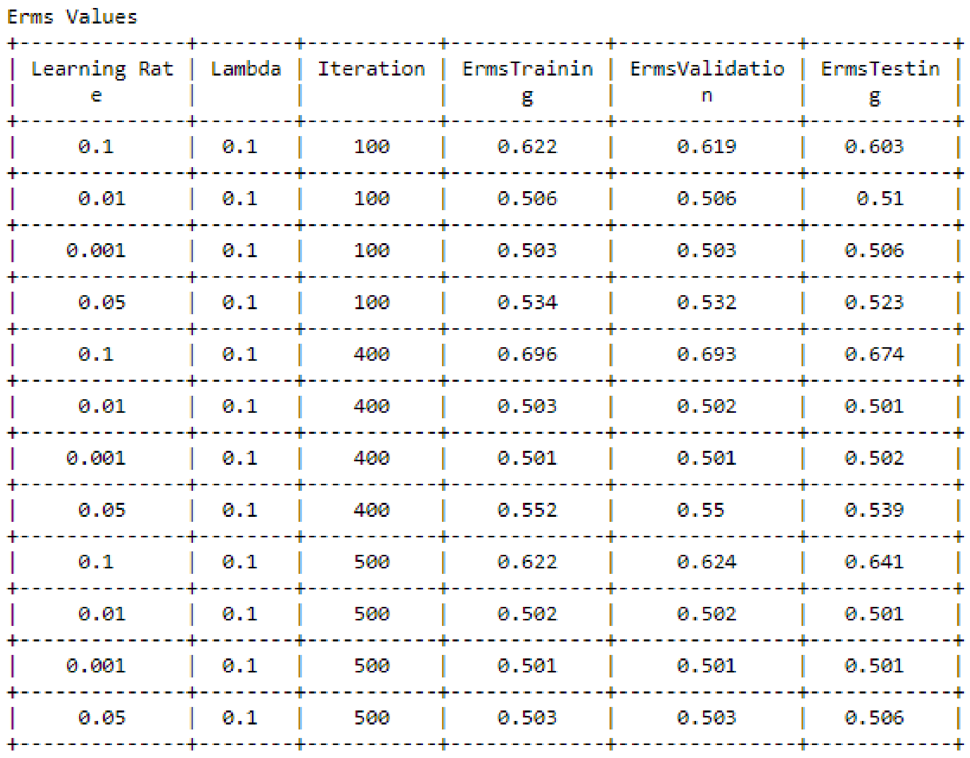


```
Erms Values
+--------------+--------+-----------+-------------+---------------+------------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e       |        |           |      g      |       n       |     g      |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.1      |   1    |    100    |    0.71     |     0.716     |   0.672    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.01     |   1    |    100    |    0.543    |     0.545     |   0.524    |
+--------------+--------+-----------+-------------+---------------+------------+
|    0.001     |   1    |    100    |    0.521    |     0.522     |    0.51    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.05     |   1    |    100    |    0.696    |     0.702     |   0.659    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.1      |   1    |    400    |    0.71     |     0.716     |   0.672    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.01     |   1    |    400    |    0.551    |     0.554     |   0.531    |
+--------------+--------+-----------+-------------+---------------+------------+
|    0.001     |   1    |    400    |    0.522    |     0.522     |    0.51    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.05     |   1    |    400    |    0.63     |     0.635     |   0.598    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.1      |   1    |    500    |    0.723    |     0.729     |   0.685    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.01     |   1    |    500    |    0.511    |     0.509     |   0.509    |
+--------------+--------+-----------+-------------+---------------+------------+
|    0.001     |   1    |    500    |    0.512    |     0.511     |   0.508    |
+--------------+--------+-----------+-------------+---------------+------------+
|     0.05     |   1    |    500    |    0.544    |     0.545     |   0.524    |
+--------------+--------+-----------+-------------+---------------+------------+
```

Number of iterations=500

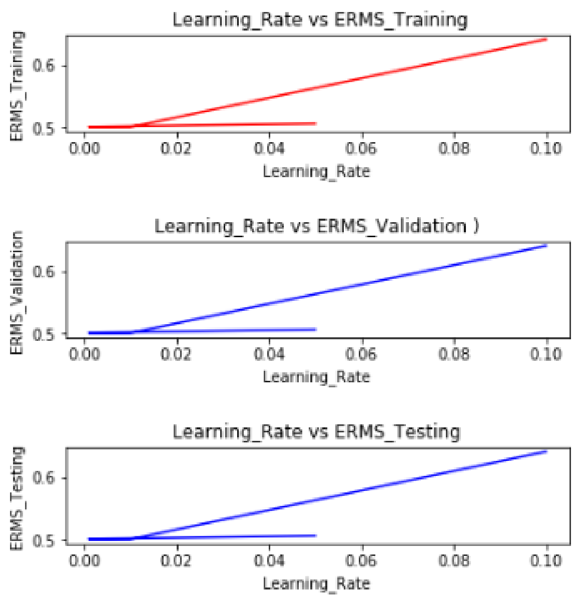


For lambda = 0.1

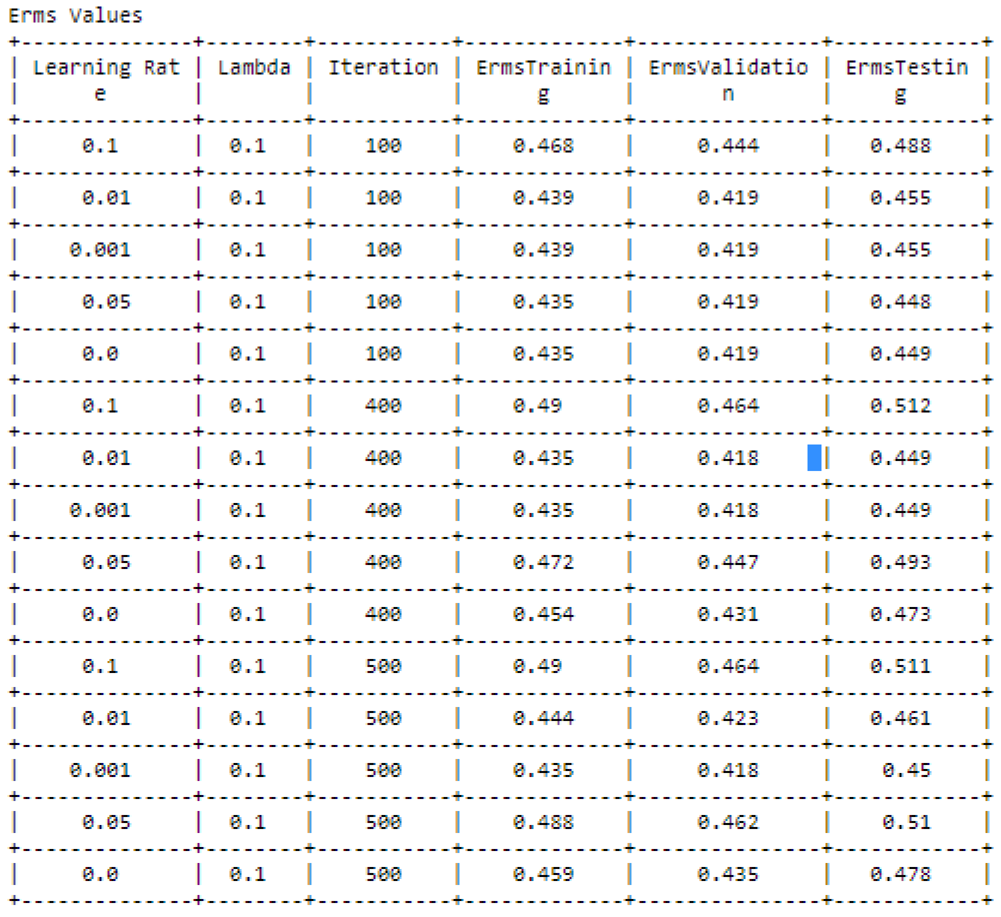


```
Erms Values
+--------------+--------+-----------+------------+--------------+-----------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e       |        |           |      g      |       n       |     g      |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.1      |  0.1   |    100    |   0.622    |    0.619     |   0.603   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.01     |  0.1   |    100    |   0.506    |    0.506     |    0.51   |
+--------------+--------+-----------+------------+--------------+-----------+
|    0.001     |  0.1   |    100    |   0.503    |    0.503     |   0.506   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.05     |  0.1   |    100    |   0.534    |    0.532     |   0.523   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.1      |  0.1   |    400    |   0.696    |    0.693     |   0.674   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.01     |  0.1   |    400    |   0.503    |    0.502     |   0.501   |
+--------------+--------+-----------+------------+--------------+-----------+
|    0.001     |  0.1   |    400    |   0.501    |    0.501     |   0.502   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.05     |  0.1   |    400    |   0.552    |     0.55     |   0.539   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.1      |  0.1   |    500    |   0.622    |    0.624     |   0.641   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.01     |  0.1   |    500    |   0.502    |    0.502     |   0.501   |
+--------------+--------+-----------+------------+--------------+-----------+
|    0.001     |  0.1   |    500    |   0.501    |    0.501     |   0.501   |
+--------------+--------+-----------+------------+--------------+-----------+
|     0.05     |  0.1   |    500    |   0.503    |    0.503     |   0.506   |
+--------------+--------+-----------+------------+--------------+-----------+
```

No of iterations=500

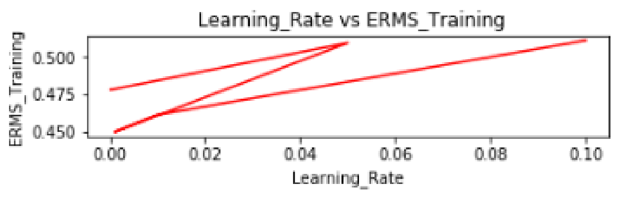


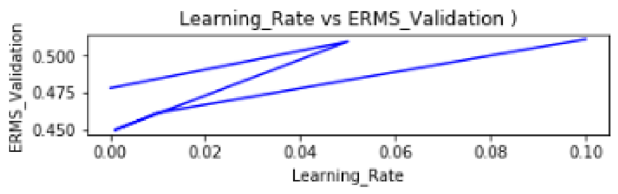
For same_pairs=791 and different_pairs=3000,lamda=0.1 and for the following learning rates

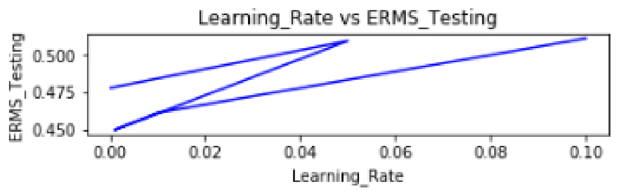


```
Erms Values
+--------------+--------+-----------+-------------+---------------+-----------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e       |        |           |      g      |       n       |     g     |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.1      |  0.1   |    100    |    0.468    |     0.444     |   0.488   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.01     |  0.1   |    100    |    0.439    |     0.419     |   0.455   |
+--------------+--------+-----------+-------------+---------------+-----------+
|    0.001     |  0.1   |    100    |    0.439    |     0.419     |   0.455   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.05     |  0.1   |    100    |    0.435    |     0.419     |   0.448   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.0      |  0.1   |    100    |    0.435    |     0.419     |   0.449   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.1      |  0.1   |    400    |    0.49     |     0.464     |   0.512   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.01     |  0.1   |    400    |    0.435    |     0.418     |   0.449   |
+--------------+--------+-----------+-------------+---------------+-----------+
|    0.001     |  0.1   |    400    |    0.435    |     0.418     |   0.449   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.05     |  0.1   |    400    |    0.472    |     0.447     |   0.493   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.0      |  0.1   |    400    |    0.454    |     0.431     |   0.473   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.1      |  0.1   |    500    |    0.49     |     0.464     |   0.511   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.01     |  0.1   |    500    |    0.444    |     0.423     |   0.461   |
+--------------+--------+-----------+-------------+---------------+-----------+
|    0.001     |  0.1   |    500    |    0.435    |     0.418     |    0.45   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.05     |  0.1   |    500    |    0.488    |     0.462     |    0.51   |
+--------------+--------+-----------+-------------+---------------+-----------+
|     0.0      |  0.1   |    500    |    0.459    |     0.435     |   0.478   |
+--------------+--------+-----------+-------------+---------------+-----------+
```

Iteration number =500





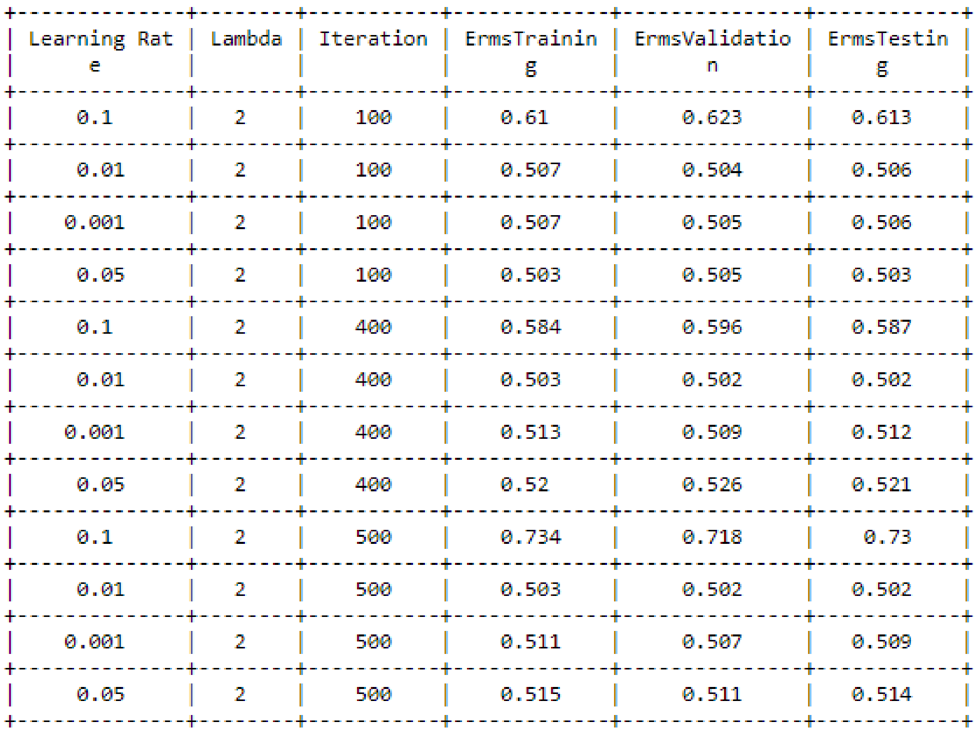


2.3.2 Human Observed Dataset with feature subtraction

Tables and graphs

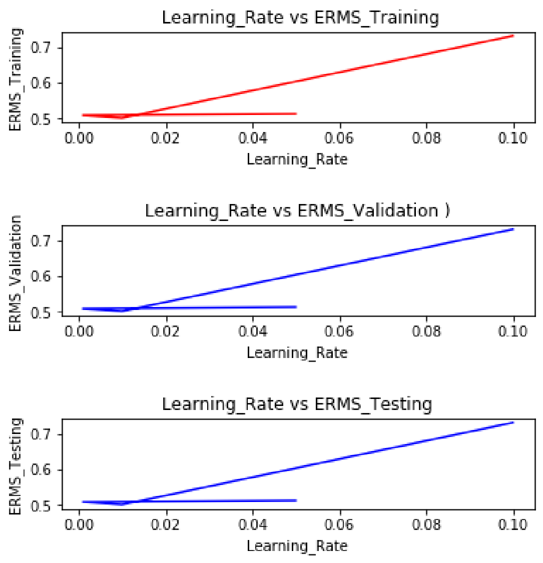
Number of basis functions(M) = 10,Training,Validation,Testing percentages on data =80,10,10 respectively

For lambda = 2



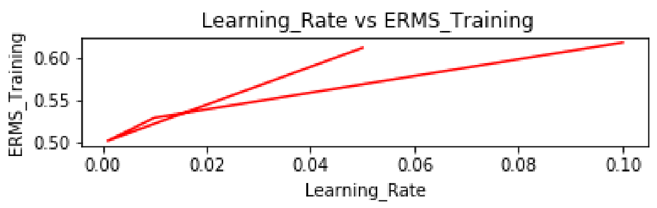
```
+--------------+--------+-----------+------------+-------------+-----------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e       |        |           |      g     |       n     |     g     |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.1      |   2    |    100    |    0.61    |    0.623    |   0.613   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.01     |   2    |    100    |   0.507    |    0.504    |   0.506   |
+--------------+--------+-----------+------------+-------------+-----------+
|    0.001     |   2    |    100    |   0.507    |    0.505    |   0.506   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.05     |   2    |    100    |   0.503    |    0.505    |   0.503   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.1      |   2    |    400    |   0.584    |    0.596    |   0.587   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.01     |   2    |    400    |   0.503    |    0.502    |   0.502   |
+--------------+--------+-----------+------------+-------------+-----------+
|    0.001     |   2    |    400    |   0.513    |    0.509    |   0.512   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.05     |   2    |    400    |    0.52    |    0.526    |   0.521   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.1      |   2    |    500    |   0.734    |    0.718    |    0.73   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.01     |   2    |    500    |   0.503    |    0.502    |   0.502   |
+--------------+--------+-----------+------------+-------------+-----------+
|    0.001     |   2    |    500    |   0.511    |    0.507    |   0.509   |
+--------------+--------+-----------+------------+-------------+-----------+
|     0.05     |   2    |    500    |   0.515    |    0.511    |   0.514   |
+--------------+--------+-----------+------------+-------------+-----------+
```

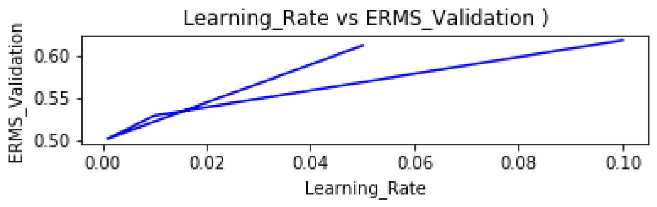
No of iterations=500

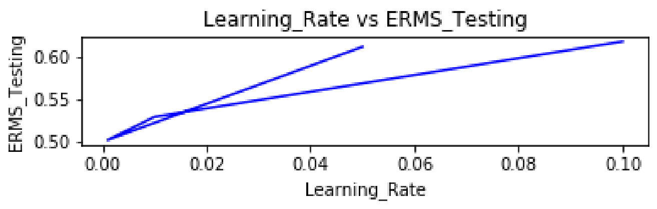


For lambda = 1

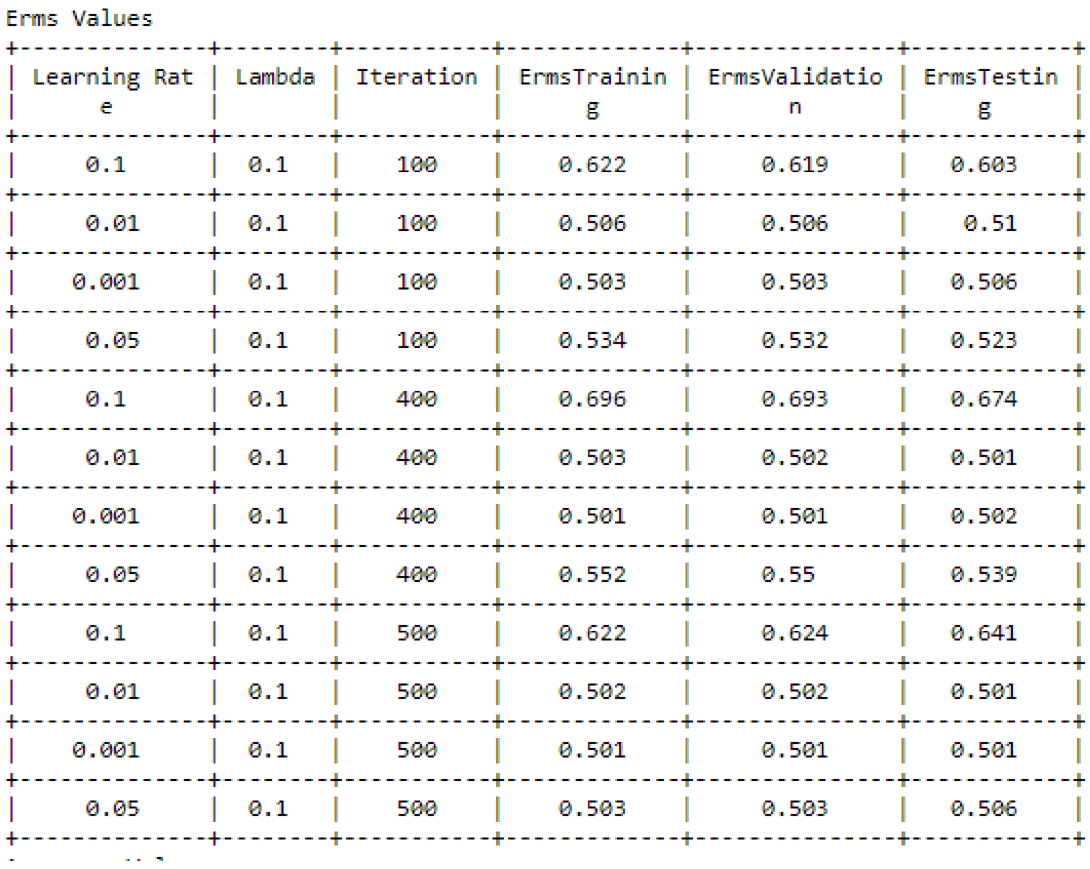
```
Erms Values
+--------------+-------+-----------+------------+---------------+------------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e       |       |           |      g     |       n       |      g     |
+--------------+-------+-----------+------------+---------------+------------+
|     0.1      |   1   |    100    |   0.636    |      0.65     |   0.608    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.01     |   1   |    100    |   0.503    |     0.505     |   0.501    |
+--------------+-------+-----------+------------+---------------+------------+
|    0.001     |   1   |    100    |   0.503    |     0.502     |   0.506    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.05     |   1   |    100    |   0.584    |     0.596     |   0.562    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.1      |   1   |    400    |   0.706    |     0.689     |   0.736    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.01     |   1   |    400    |   0.523    |     0.516     |   0.535    |
+--------------+-------+-----------+------------+---------------+------------+
|    0.001     |   1   |    400    |   0.504    |     0.502     |   0.508    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.05     |   1   |    400    |   0.635    |      0.62     |   0.661    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.1      |   1   |    500    |   0.647    |     0.662     |   0.619    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.01     |   1   |    500    |   0.547    |     0.556     |   0.529    |
+--------------+-------+-----------+------------+---------------+------------+
|    0.001     |   1   |    500    |   0.502    |     0.503     |   0.502    |
+--------------+-------+-----------+------------+---------------+------------+
|     0.05     |   1   |    500    |    0.64    |     0.655     |   0.612    |
+--------------+-------+-----------+------------+---------------+------------+
```





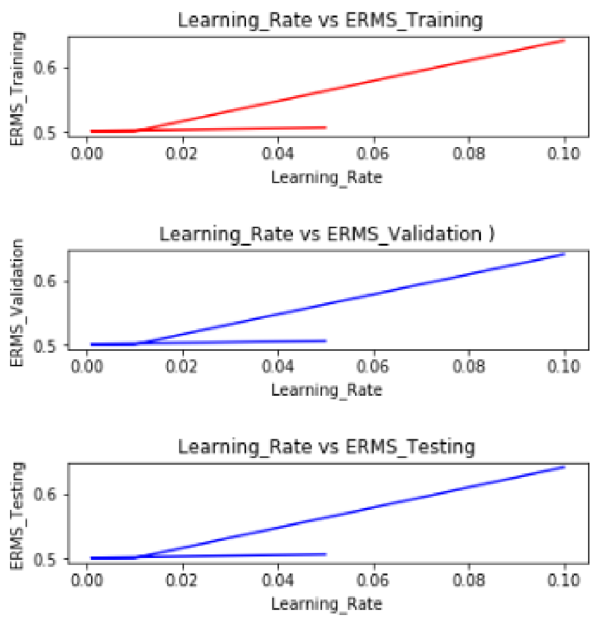


Lamda=0.1

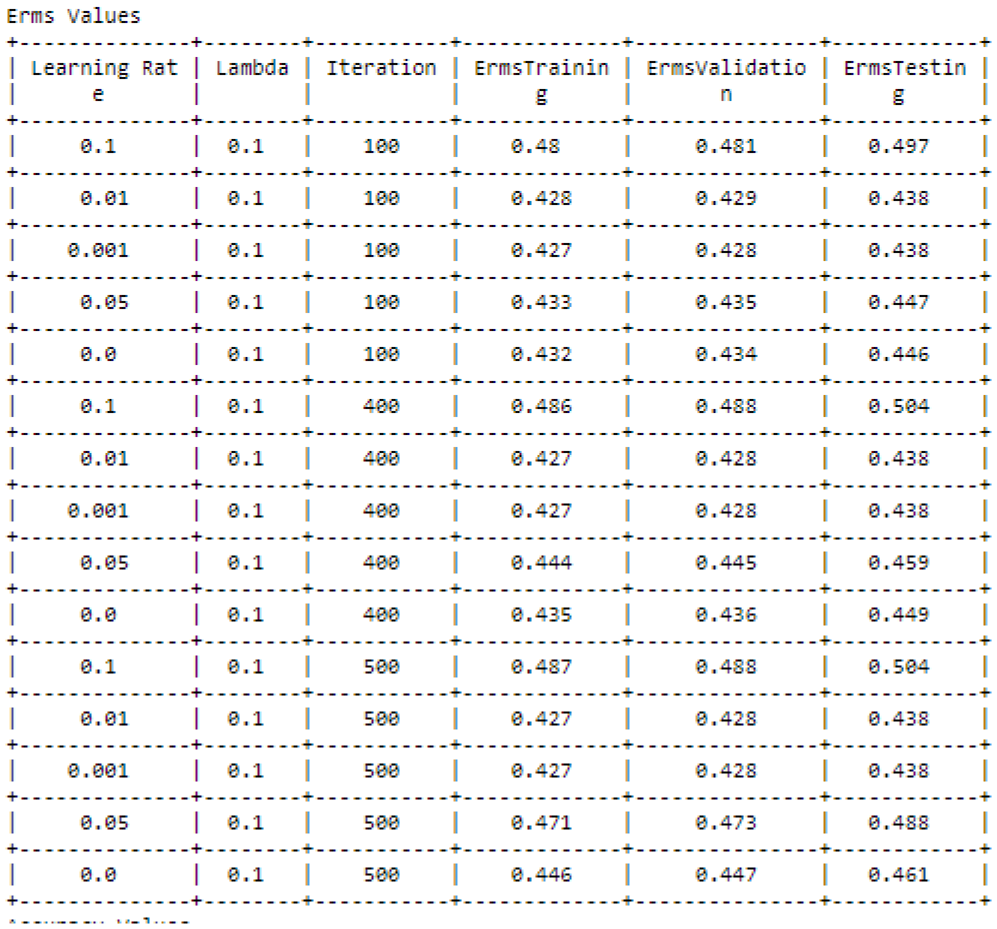


```
Erms Values
+---------------+---------+-----------+-------------+----------------+-------------+
| Learning Rat  | Lambda  | Iteration | ErmsTrainin | ErmsValidatio  | ErmsTestin  |
|      e        |         |           |      g      |       n        |      g      |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.1      |   0.1   |    100    |    0.622    |     0.619      |    0.603    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.01     |   0.1   |    100    |    0.506    |     0.506      |     0.51    |
+---------------+---------+-----------+-------------+----------------+-------------+
|     0.001     |   0.1   |    100    |    0.503    |     0.503      |    0.506    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.05     |   0.1   |    100    |    0.534    |     0.532      |    0.523    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.1      |   0.1   |    400    |    0.696    |     0.693      |    0.674    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.01     |   0.1   |    400    |    0.503    |     0.502      |    0.501    |
+---------------+---------+-----------+-------------+----------------+-------------+
|     0.001     |   0.1   |    400    |    0.501    |     0.501      |    0.502    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.05     |   0.1   |    400    |    0.552    |      0.55      |    0.539    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.1      |   0.1   |    500    |    0.622    |     0.624      |    0.641    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.01     |   0.1   |    500    |    0.502    |     0.502      |    0.501    |
+---------------+---------+-----------+-------------+----------------+-------------+
|     0.001     |   0.1   |    500    |    0.501    |     0.501      |    0.501    |
+---------------+---------+-----------+-------------+----------------+-------------+
|      0.05     |   0.1   |    500    |    0.503    |     0.503      |    0.506    |
+---------------+---------+-----------+-------------+----------------+-------------+
```

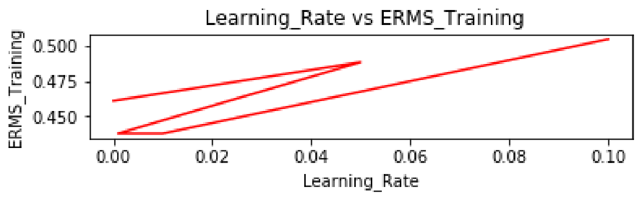
No of iterations=500

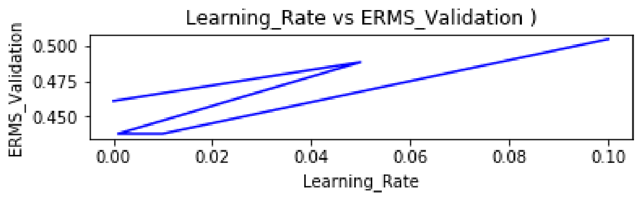


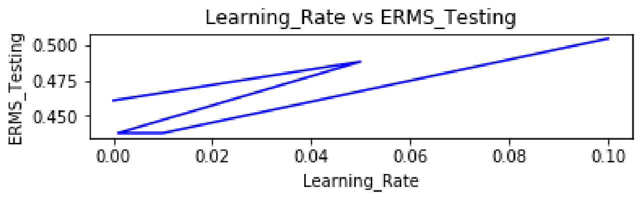
For same_pairs=791 and different_pairs=3000,lamda=0.1 and for the following learning rates



```
Erms Values
+-------------+-------+----------+-----------+-------------+-----------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e      |       |          |     g     |      n      |     g     |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.1     |  0.1  |   100    |    0.48   |    0.481    |   0.497   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.01    |  0.1  |   100    |   0.428   |    0.429    |   0.438   |
+-------------+-------+----------+-----------+-------------+-----------+
|    0.001    |  0.1  |   100    |   0.427   |    0.428    |   0.438   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.05    |  0.1  |   100    |   0.433   |    0.435    |   0.447   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.0     |  0.1  |   100    |   0.432   |    0.434    |   0.446   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.1     |  0.1  |   400    |   0.486   |    0.488    |   0.504   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.01    |  0.1  |   400    |   0.427   |    0.428    |   0.438   |
+-------------+-------+----------+-----------+-------------+-----------+
|    0.001    |  0.1  |   400    |   0.427   |    0.428    |   0.438   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.05    |  0.1  |   400    |   0.444   |    0.445    |   0.459   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.0     |  0.1  |   400    |   0.435   |    0.436    |   0.449   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.1     |  0.1  |   500    |   0.487   |    0.488    |   0.504   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.01    |  0.1  |   500    |   0.427   |    0.428    |   0.438   |
+-------------+-------+----------+-----------+-------------+-----------+
|    0.001    |  0.1  |   500    |   0.427   |    0.428    |   0.438   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.05    |  0.1  |   500    |   0.471   |    0.473    |   0.488   |
+-------------+-------+----------+-----------+-------------+-----------+
|     0.0     |  0.1  |   500    |   0.446   |    0.447    |   0.461   |
+-------------+-------+----------+-----------+-------------+-----------+
```





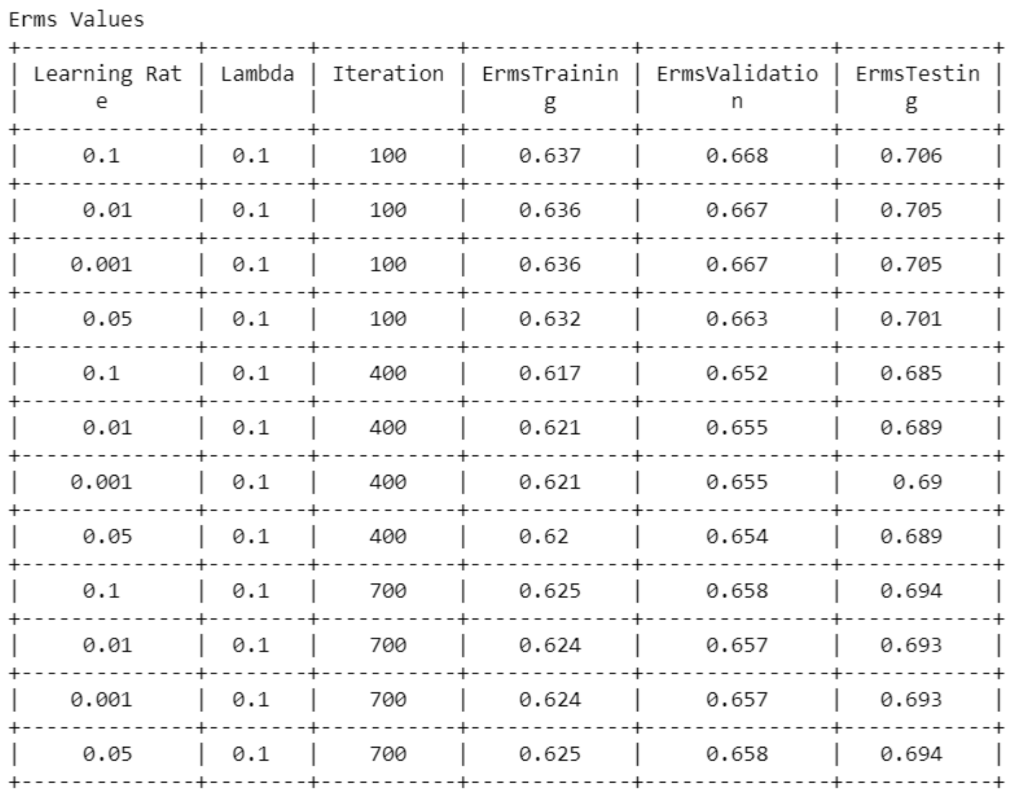


2.3.3 GSC Dataset with feature concatentation

Tables and graphs

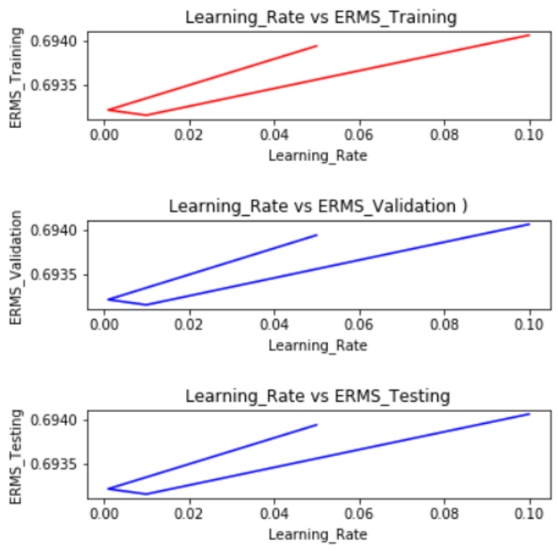
Number of basis functions(M) = 10,Training,Validation,Testing percentages on data =80,10,10 respectively

For lambda = 0.1,same_pairs=500,different_pairs=500



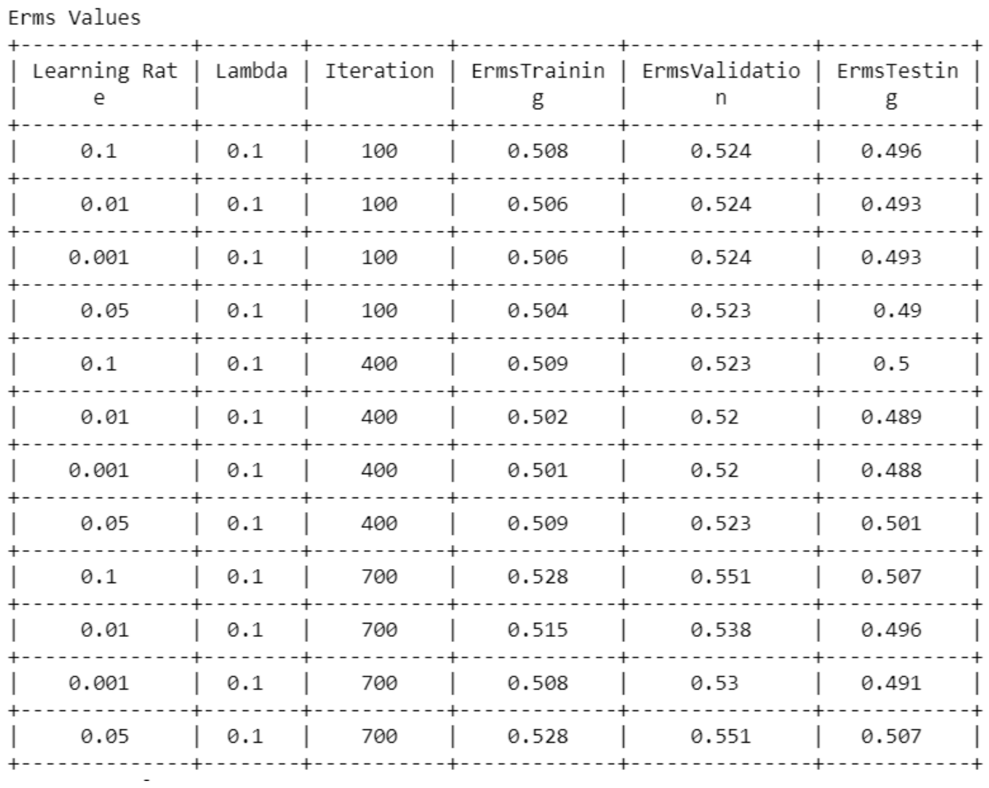
```
Erms Values
+-------------+--------+-----------+-------------+--------------+-----------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin |
|      e      |        |           |      g      |       n      |     g     |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.1     |  0.1   |    100    |    0.637    |    0.668     |   0.706   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.01    |  0.1   |    100    |    0.636    |    0.667     |   0.705   |
+-------------+--------+-----------+-------------+--------------+-----------+
|    0.001    |  0.1   |    100    |    0.636    |    0.667     |   0.705   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.05    |  0.1   |    100    |    0.632    |    0.663     |   0.701   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.1     |  0.1   |    400    |    0.617    |    0.652     |   0.685   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.01    |  0.1   |    400    |    0.621    |    0.655     |   0.689   |
+-------------+--------+-----------+-------------+--------------+-----------+
|    0.001    |  0.1   |    400    |    0.621    |    0.655     |    0.69   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.05    |  0.1   |    400    |     0.62    |    0.654     |   0.689   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.1     |  0.1   |    700    |    0.625    |    0.658     |   0.694   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.01    |  0.1   |    700    |    0.624    |    0.657     |   0.693   |
+-------------+--------+-----------+-------------+--------------+-----------+
|    0.001    |  0.1   |    700    |    0.624    |    0.657     |   0.693   |
+-------------+--------+-----------+-------------+--------------+-----------+
|     0.05    |  0.1   |    700    |    0.625    |    0.658     |   0.694   |
+-------------+--------+-----------+-------------+--------------+-----------+
```

Number of Iterations=700



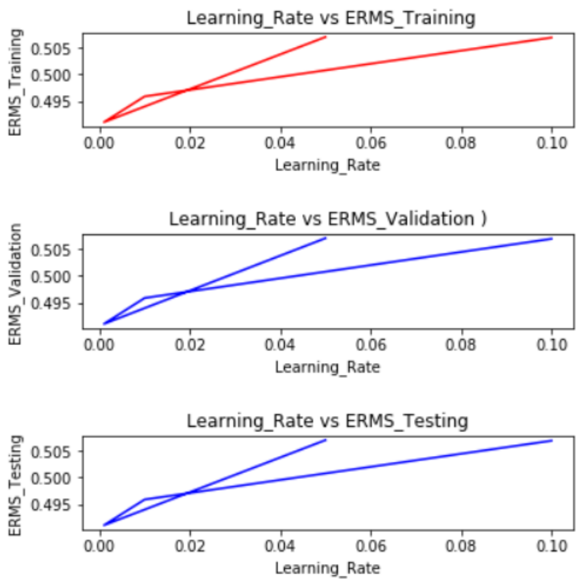
2.3.4 GSC Dataset with feature subtraction

Tables and graphs



```
Erms Values
+--------------+--------+-----------+-------------+---------------+-------------+
| Learning Rat | Lambda | Iteration | ErmsTrainin | ErmsValidatio | ErmsTestin  |
|      e       |        |           |      g      |       n       |      g      |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.1      |  0.1   |    100    |    0.508    |     0.524     |    0.496    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.01     |  0.1   |    100    |    0.506    |     0.524     |    0.493    |
+--------------+--------+-----------+-------------+---------------+-------------+
|    0.001     |  0.1   |    100    |    0.506    |     0.524     |    0.493    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.05     |  0.1   |    100    |    0.504    |     0.523     |     0.49    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.1      |  0.1   |    400    |    0.509    |     0.523     |     0.5     |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.01     |  0.1   |    400    |    0.502    |      0.52     |    0.489    |
+--------------+--------+-----------+-------------+---------------+-------------+
|    0.001     |  0.1   |    400    |    0.501    |      0.52     |    0.488    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.05     |  0.1   |    400    |    0.509    |     0.523     |    0.501    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.1      |  0.1   |    700    |    0.528    |     0.551     |    0.507    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.01     |  0.1   |    700    |    0.515    |     0.538     |    0.496    |
+--------------+--------+-----------+-------------+---------------+-------------+
|    0.001     |  0.1   |    700    |    0.508    |      0.53     |    0.491    |
+--------------+--------+-----------+-------------+---------------+-------------+
|     0.05     |  0.1   |    700    |    0.528    |     0.551     |    0.507    |
+--------------+--------+-----------+-------------+---------------+-------------+
```

Number of iterations=700



2.4 Stochastic Gradient Descent Solution for Logistic Regression

To generate probabilities, logistic regression uses a function that gives outputs between 0 and 1 for all values of X. We use sigmoid function here

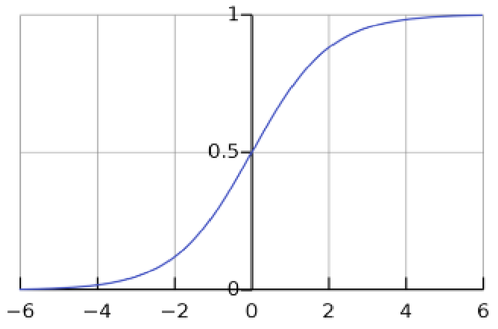


Fig. 2 — Logistic function

$$h_\theta(x) = g(\theta^T x)$$

$$z = \theta^T x$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

Loss Function :

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1-y)^T \log(1-h)\right)$$

Gradient Descent: Goal is to minimize the loss function. Finding partial derivate of loss function with respect to each weight , we have

$$\frac{\delta J(\theta)}{\delta \theta_j} = \frac{1}{m} X^T (g(X\theta) - y)$$

$\nabla E = X^T(g(X(\theta)\text{-}y))$

For every iteration, calculate the Training, validation and testing output by using the design matrix and updated weights.

Also, compute the E_{RMS} values for each of the data split and append these values to the list to find the minimum value.

- The regression line predicts the average y value associated with a given x value. It gives the measure of the spread of the y values around that average. To do this, we use the root-mean-square error (Root mean squared error).

- Calculate the root mean squared errors between actual target values of Validation stored in ValidationTarget with the computed output values (using validation Design matrix and the weight matrix obtained using closed form). Root Mean Square (RMS) error, defined as E_{RMS} =$\sqrt{(}\mathbf{2E}(\mathbf{w^*})/\mathbf{N_v}$ where w^* is the solution and N_V is the size of the test dataset

The possible hyper-parameters that can be changed here are:

- The data split on training, validation and testing on the actual raw dataset.
- Learning Rate
- No of iterations(early stopping)

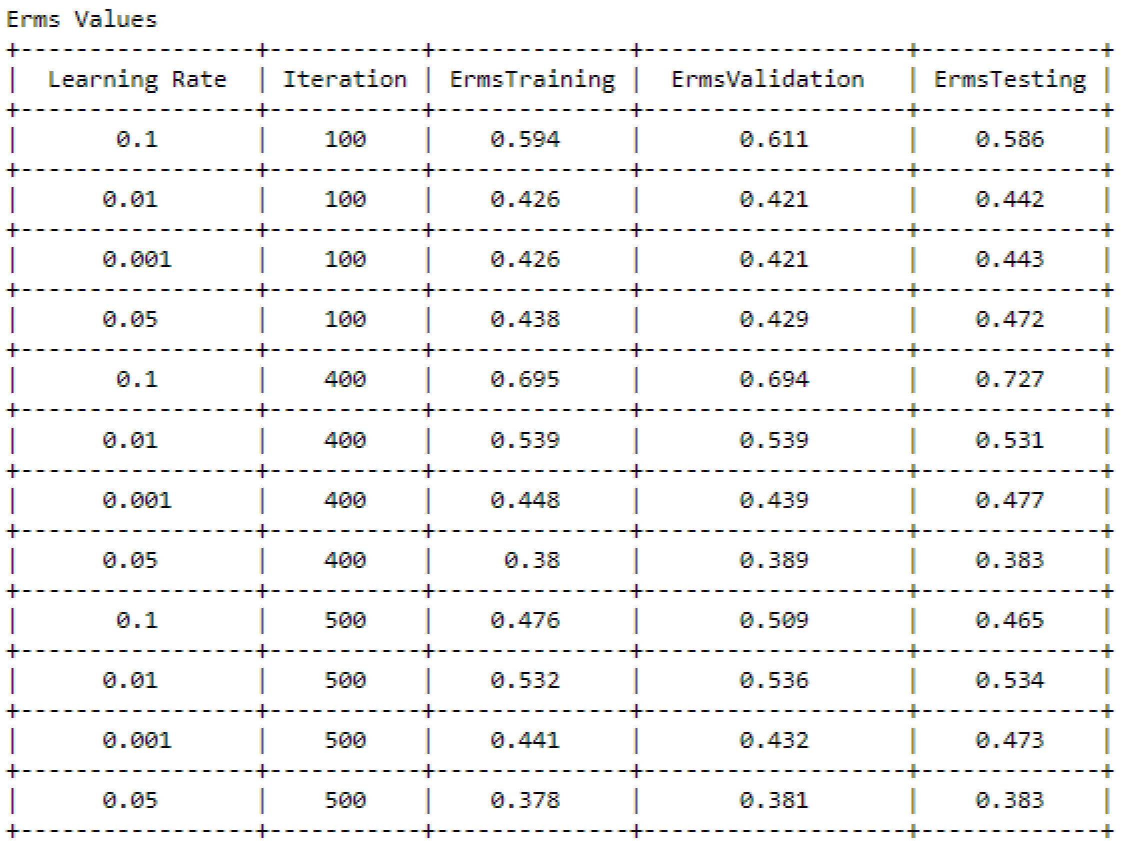
We repeat the same procedure for all the four datasets

2.4.1 Human Observed Dataset with feature concatentation

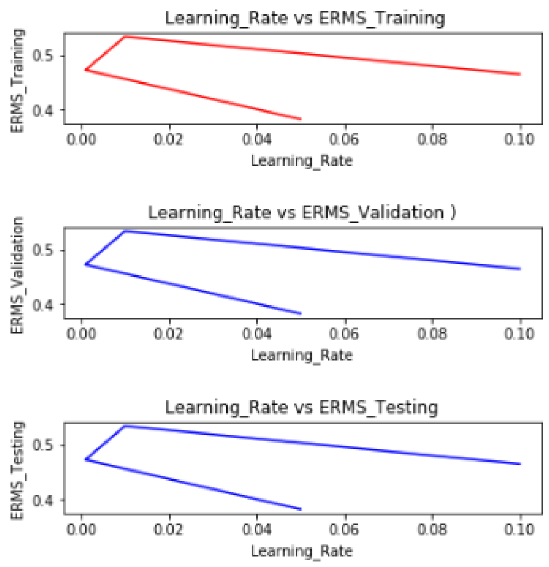
Learning Rates=[0.1,0.01,0.001,0.05,0.0001]

Tables and graphs

Number of basis functions(M) = 18,Training,Validation,Testing percentages on data =80,10,10 respectively

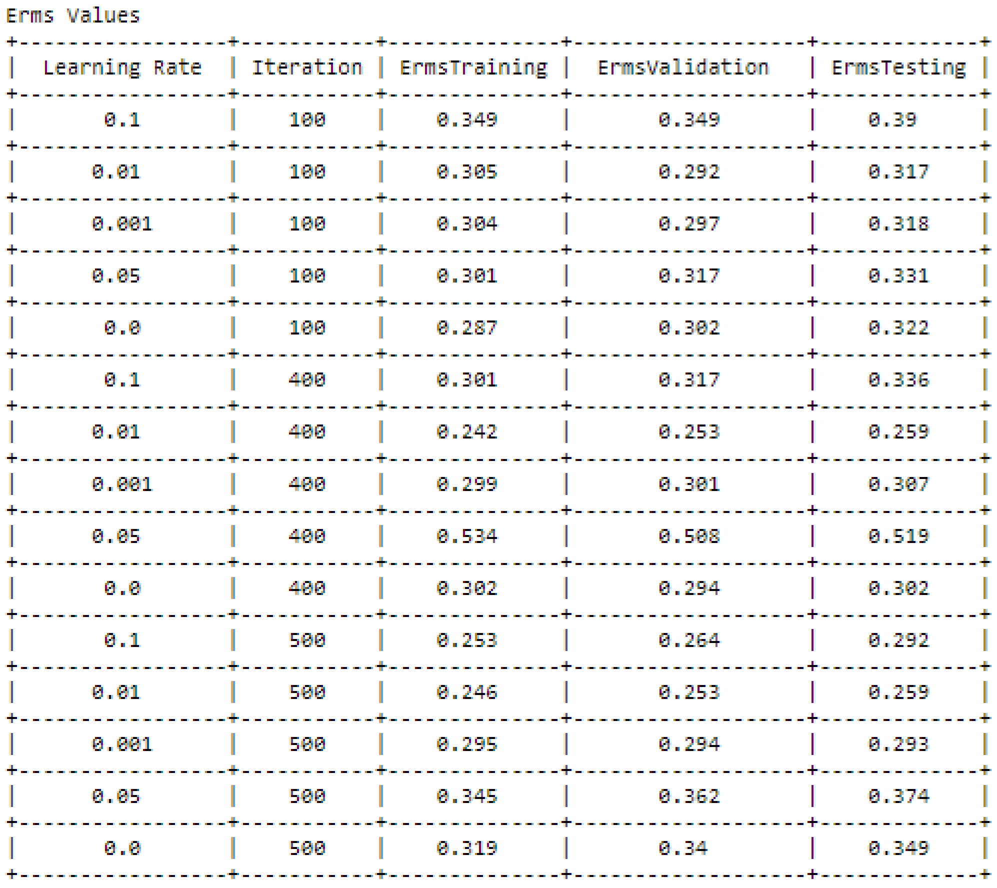


```
Erms Values
+-----------------+-----------+--------------+-------------------+-------------+
|  Learning Rate  | Iteration | ErmsTraining |  ErmsValidation   | ErmsTesting |
+-----------------+-----------+--------------+-------------------+-------------+
|       0.1       |    100    |    0.594     |       0.611       |    0.586    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.01       |    100    |    0.426     |       0.421       |    0.442    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.001      |    100    |    0.426     |       0.421       |    0.443    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.05       |    100    |    0.438     |       0.429       |    0.472    |
+-----------------+-----------+--------------+-------------------+-------------+
|       0.1       |    400    |    0.695     |       0.694       |    0.727    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.01       |    400    |    0.539     |       0.539       |    0.531    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.001      |    400    |    0.448     |       0.439       |    0.477    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.05       |    400    |     0.38     |       0.389       |    0.383    |
+-----------------+-----------+--------------+-------------------+-------------+
|       0.1       |    500    |    0.476     |       0.509       |    0.465    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.01       |    500    |    0.532     |       0.536       |    0.534    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.001      |    500    |    0.441     |       0.432       |    0.473    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.05       |    500    |    0.378     |       0.381       |    0.383    |
+-----------------+-----------+--------------+-------------------+-------------+
```



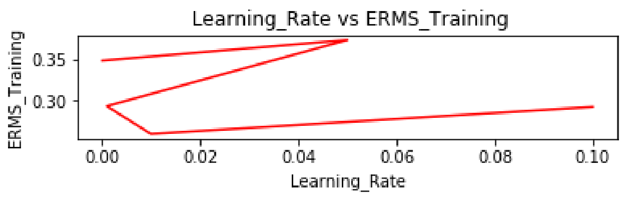
Number of iterations=500,

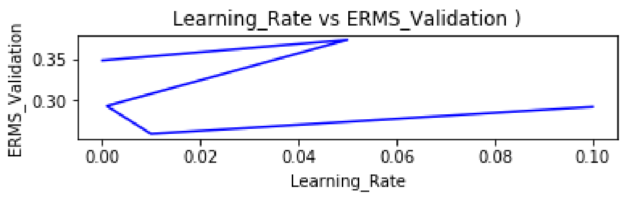
Same_pairs=791,different pairs=2500,

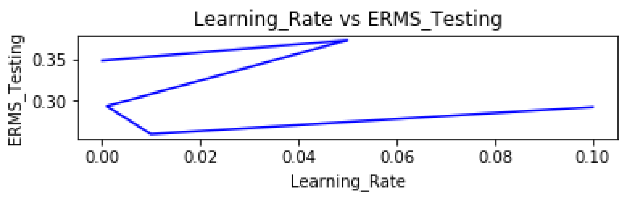


```
Erms Values
+----------------+-----------+--------------+--------------------+------------+
| Learning Rate  | Iteration | ErmsTraining |   ErmsValidation   | ErmsTesting |
+----------------+-----------+--------------+--------------------+------------+
|      0.1       |    100    |    0.349     |       0.349        |    0.39    |
+----------------+-----------+--------------+--------------------+------------+
|     0.01       |    100    |    0.305     |       0.292        |   0.317    |
+----------------+-----------+--------------+--------------------+------------+
|     0.001      |    100    |    0.304     |       0.297        |   0.318    |
+----------------+-----------+--------------+--------------------+------------+
|     0.05       |    100    |    0.301     |       0.317        |   0.331    |
+----------------+-----------+--------------+--------------------+------------+
|      0.0       |    100    |    0.287     |       0.302        |   0.322    |
+----------------+-----------+--------------+--------------------+------------+
|      0.1       |    400    |    0.301     |       0.317        |   0.336    |
+----------------+-----------+--------------+--------------------+------------+
|     0.01       |    400    |    0.242     |       0.253        |   0.259    |
+----------------+-----------+--------------+--------------------+------------+
|     0.001      |    400    |    0.299     |       0.301        |   0.307    |
+----------------+-----------+--------------+--------------------+------------+
|     0.05       |    400    |    0.534     |       0.508        |   0.519    |
+----------------+-----------+--------------+--------------------+------------+
|      0.0       |    400    |    0.302     |       0.294        |   0.302    |
+----------------+-----------+--------------+--------------------+------------+
|      0.1       |    500    |    0.253     |       0.264        |   0.292    |
+----------------+-----------+--------------+--------------------+------------+
|     0.01       |    500    |    0.246     |       0.253        |   0.259    |
+----------------+-----------+--------------+--------------------+------------+
|     0.001      |    500    |    0.295     |       0.294        |   0.293    |
+----------------+-----------+--------------+--------------------+------------+
|     0.05       |    500    |    0.345     |       0.362        |   0.374    |
+----------------+-----------+--------------+--------------------+------------+
|      0.0       |    500    |    0.319     |        0.34        |   0.349    |
+----------------+-----------+--------------+--------------------+------------+
```

Number of iteration=500



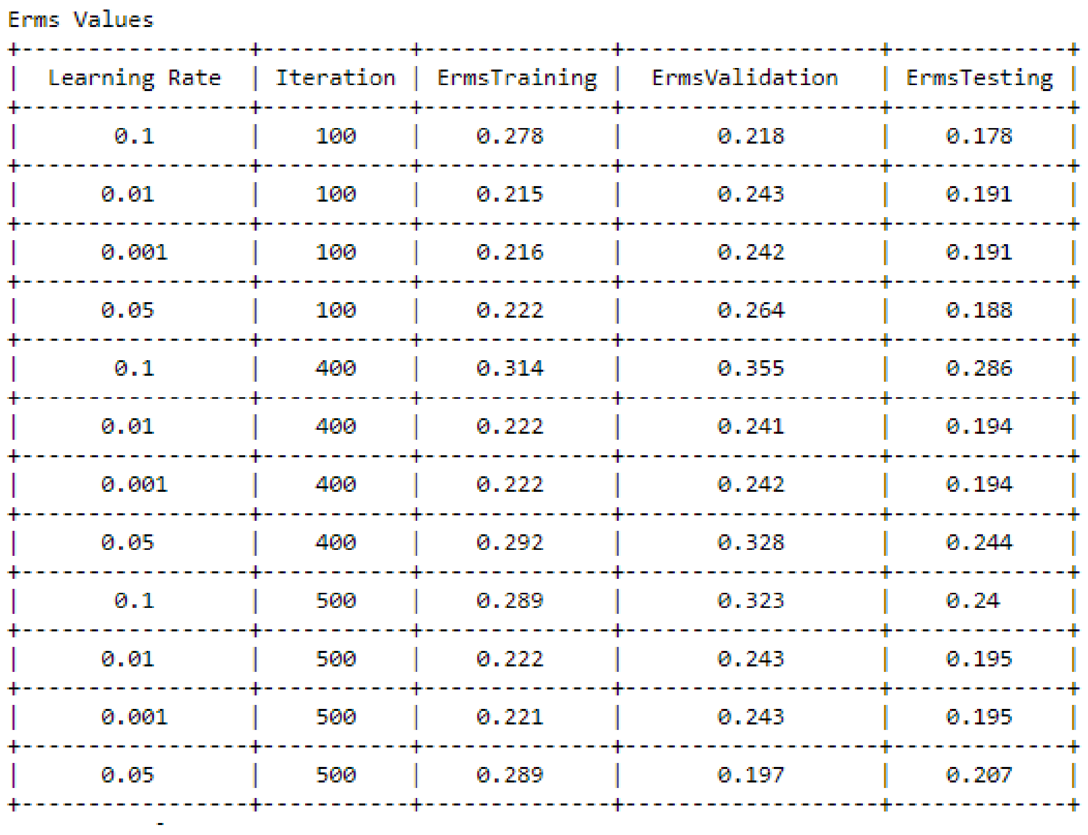




2.4.2 Human Observed Dataset with feature subtraction

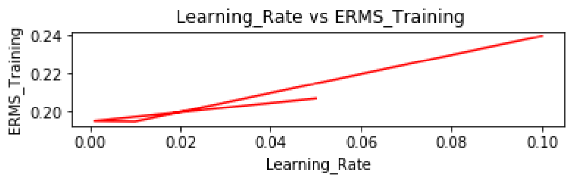
Tables and graphs

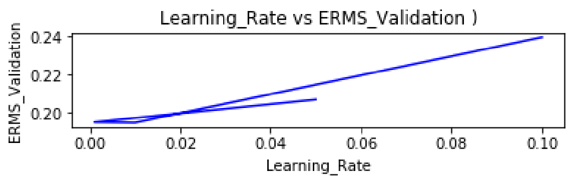
Number of basis functions(M) = 9,Training,Validation,Testing percentages on data =80,10,10 respectively

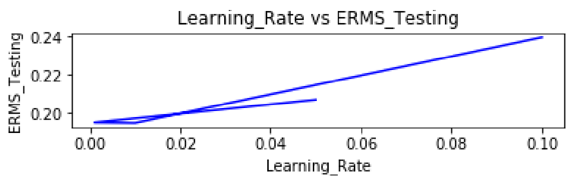


```
Erms Values
+-----------------+-----------+--------------+-------------------+-------------+
|  Learning Rate  | Iteration | ErmsTraining |  ErmsValidation   | ErmsTesting |
+-----------------+-----------+--------------+-------------------+-------------+
|       0.1       |    100    |    0.278     |       0.218       |    0.178    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.01       |    100    |    0.215     |       0.243       |    0.191    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.001      |    100    |    0.216     |       0.242       |    0.191    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.05       |    100    |    0.222     |       0.264       |    0.188    |
+-----------------+-----------+--------------+-------------------+-------------+
|       0.1       |    400    |    0.314     |       0.355       |    0.286    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.01       |    400    |    0.222     |       0.241       |    0.194    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.001      |    400    |    0.222     |       0.242       |    0.194    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.05       |    400    |    0.292     |       0.328       |    0.244    |
+-----------------+-----------+--------------+-------------------+-------------+
|       0.1       |    500    |    0.289     |       0.323       |    0.24     |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.01       |    500    |    0.222     |       0.243       |    0.195    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.001      |    500    |    0.221     |       0.243       |    0.195    |
+-----------------+-----------+--------------+-------------------+-------------+
|      0.05       |    500    |    0.289     |       0.197       |    0.207    |
+-----------------+-----------+--------------+-------------------+-------------+
```

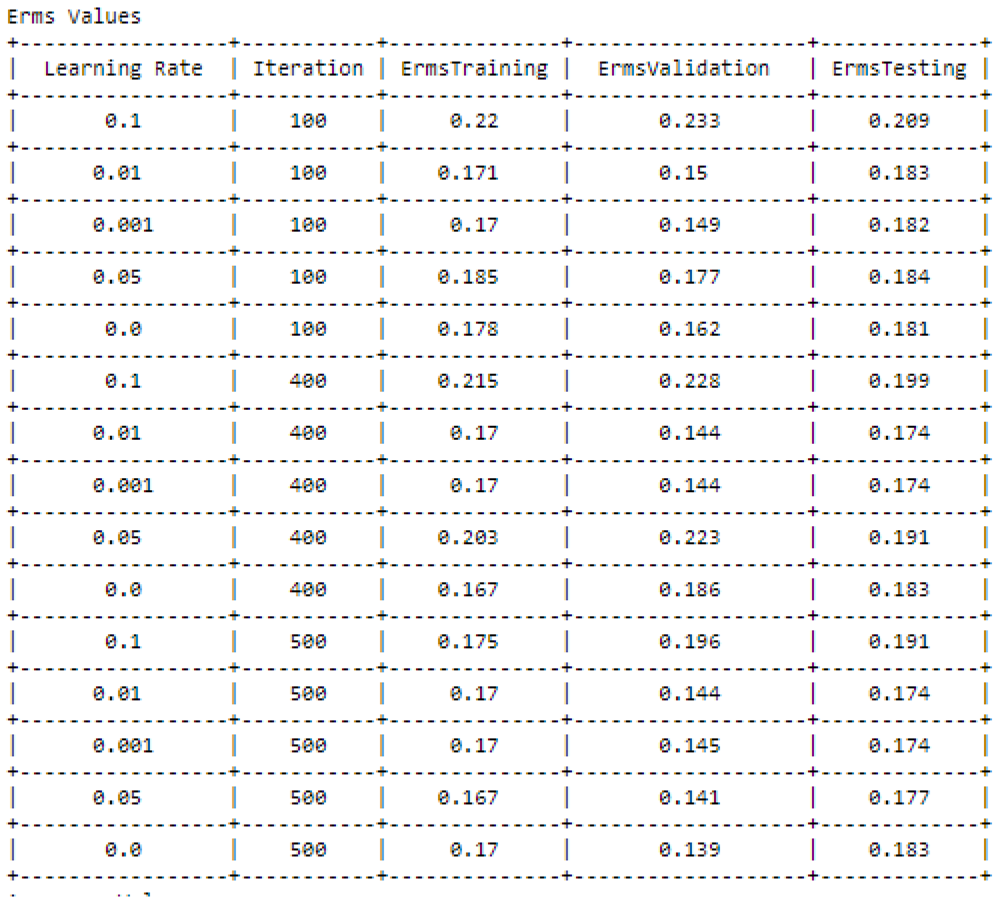
Iterations=500





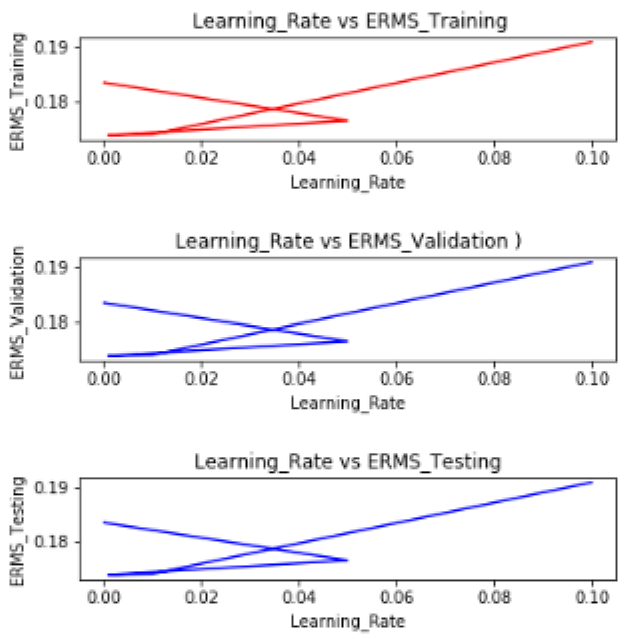


Same_pairs=791, different pairs=2500 and following learning rates



```
Erms Values
+-----------------+-----------+--------------+--------------------+-------------+
|  Learning Rate  | Iteration | ErmsTraining |   ErmsValidation   | ErmsTesting |
+-----------------+-----------+--------------+--------------------+-------------+
|       0.1       |    100    |     0.22     |       0.233        |    0.209    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.01       |    100    |    0.171     |        0.15        |    0.183    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.001      |    100    |     0.17     |       0.149        |    0.182    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.05       |    100    |    0.185     |       0.177        |    0.184    |
+-----------------+-----------+--------------+--------------------+-------------+
|       0.0       |    100    |    0.178     |       0.162        |    0.181    |
+-----------------+-----------+--------------+--------------------+-------------+
|       0.1       |    400    |    0.215     |       0.228        |    0.199    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.01       |    400    |     0.17     |       0.144        |    0.174    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.001      |    400    |     0.17     |       0.144        |    0.174    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.05       |    400    |    0.203     |       0.223        |    0.191    |
+-----------------+-----------+--------------+--------------------+-------------+
|       0.0       |    400    |    0.167     |       0.186        |    0.183    |
+-----------------+-----------+--------------+--------------------+-------------+
|       0.1       |    500    |    0.175     |       0.196        |    0.191    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.01       |    500    |     0.17     |       0.144        |    0.174    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.001      |    500    |     0.17     |       0.145        |    0.174    |
+-----------------+-----------+--------------+--------------------+-------------+
|      0.05       |    500    |    0.167     |       0.141        |    0.177    |
+-----------------+-----------+--------------+--------------------+-------------+
|       0.0       |    500    |     0.17     |       0.139        |    0.183    |
+-----------------+-----------+--------------+--------------------+-------------+
```

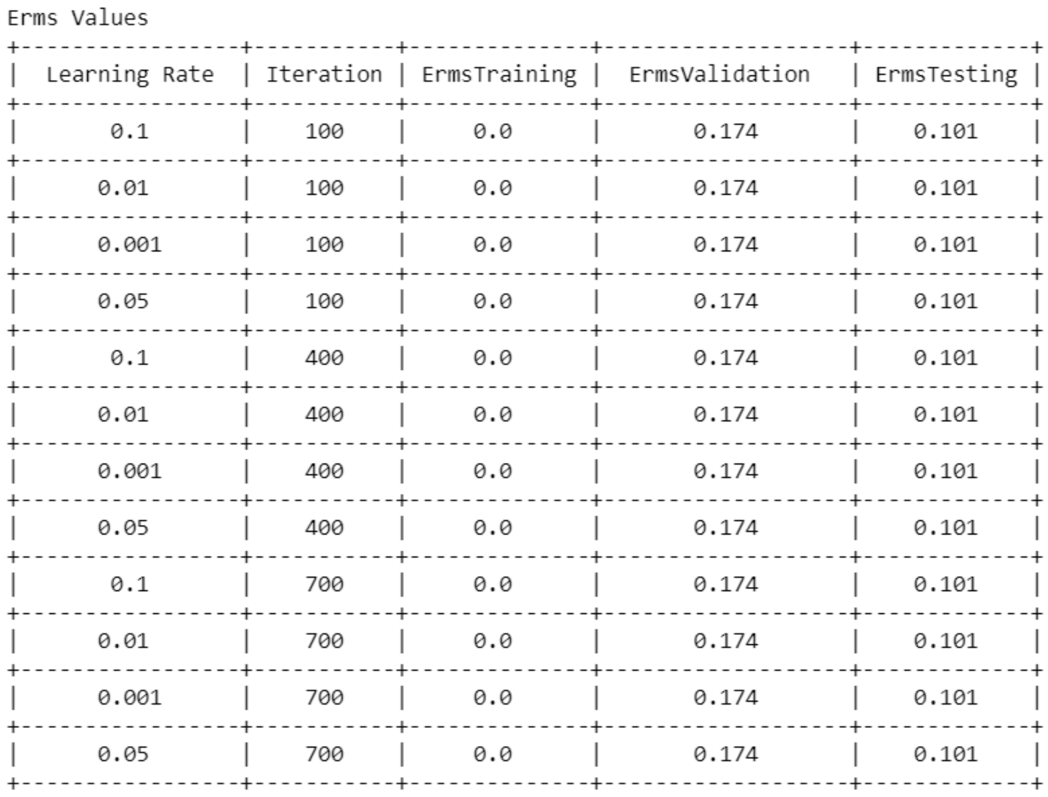
Number of iteration=500



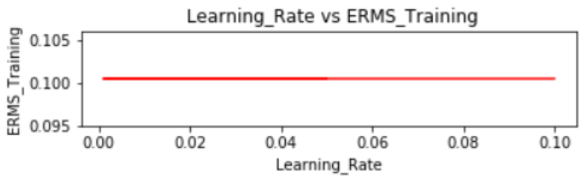
2.4.3 GSC Dataset with feature concatentation

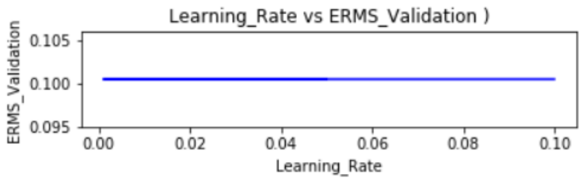
Tables and graphs

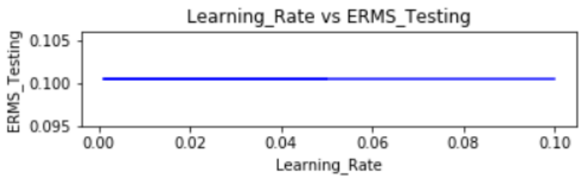
No of basis functions=1024, learning rate=[0.1,0.01,0.001,0.05]



```
Erms Values
+-----------------+-----------+-------------+-------------------+-------------+
|  Learning Rate  | Iteration | ErmsTraining |  ErmsValidation   | ErmsTesting |
+-----------------+-----------+-------------+-------------------+-------------+
|       0.1       |    100    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.01       |    100    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.001      |    100    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.05       |    100    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|       0.1       |    400    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.01       |    400    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.001      |    400    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.05       |    400    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|       0.1       |    700    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.01       |    700    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.001      |    700    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
|      0.05       |    700    |     0.0     |       0.174       |    0.101    |
+-----------------+-----------+-------------+-------------------+-------------+
```







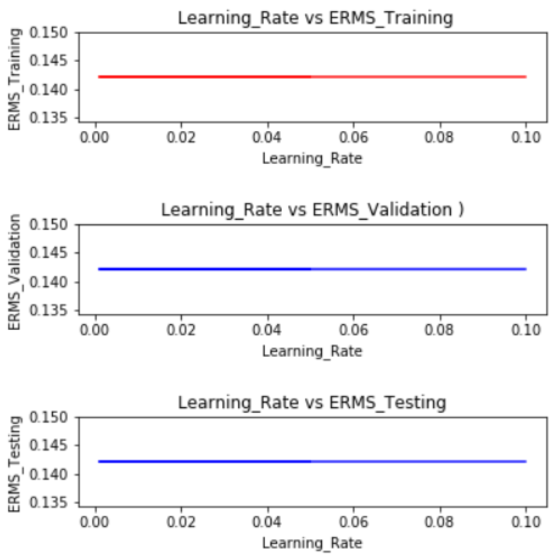
2.4.4 GSC Dataset with feature subtraction

Tables and graphs

No of basis functions=512, learning rate=[0.1,0.01,0.001,0.05]

```
Erms Values
+-----------------+------------+---------------+--------------------+-------------+
|  Learning Rate  | Iteration  | ErmsTraining  |  ErmsValidation    | ErmsTesting |
+-----------------+------------+---------------+--------------------+-------------+
|      0.1        |    100     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.01       |    100     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.001      |    100     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.05       |    100     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.1        |    400     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.01       |    400     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.001      |    400     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.05       |    400     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.1        |    700     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.01       |    700     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.001      |    700     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
|      0.05       |    700     |     0.0       |      0.101         |    0.142    |
+-----------------+------------+---------------+--------------------+-------------+
```

Number of iterations=700



3. Neural Network Implementation

THE LEARNING SYSTEM

- Here we use Keras which is a high-level neural networks API because of its user-friendliness, modularity, easy extensibility and work with python.
- The given problem is considered as categorical classification problem because we focus on the assignment of categorical target labels to the samples.
- We first have to build a model in neural network to perform the task of testing in software2.0. As the core abstraction of a model is a layer and there is a stack of layers that has input and output, we use **sequential** model to organize them.
- Convert the input to vector of activations by encoding them to binary representation. This is given to the first dense layer of the neural network (hidden) in the form of a csv file to activate the nodes of it.
- Do one-hot encoding on label data to convert it into binary class matrices of 4 categories.
- Create a simple model for which different layers are added. We first add a first **dense** layer called the hidden layer with certain number of nodes and also declare the shape of the given input vector.
- **Number of nodes** in a layer is arbitrarily chosen. The accuracy of the model increases with the increase in the number of nodes in the layer and tends to decrease after some time due to overfitting and high variance.
- A 'layer' does not have an activation. Each individual neuron has an activation. The **state** of a neuron is it's bias + all incoming connections (weight * activation from source neuron). The **activation** is the state of a neuron passed through an activation function.
- We then do the **activation** to provide linearity to the model and choose it in a way that it is helpful in backpropagation to adjust the weights.
- We do **regularization** by adding a **dropout** layer to randomly drop the neurons so as to reduce interdependent learning amongst them. The fraction of neurons that are to be dropped is given as a parameter of Dropout.
- Here we use the loss function **Softmax** as the activation function on the output layer which returns the probabilities of each class where target class has the highest probability.
- After defining the model, we **compile** the model using tensorflow in the backend because of its efficient numerical libraries.
    - Compiling the model: Use **categorical_crossentropy** as loss function to evaluate a set of weights as we have a *Multi-class classification problem.*
    - **Optimizer** is used to search through different weights for the network and optimize the loss function.
    - **Metrics** are collected and reported at the end of every epoch.
- After compiling, we have to **fit** the model to execute this on training data. We set the following parameters along with the processed data,
    - **Epoch**- Number of passes over the entire dataset.
    - **Batch Size** - Number of instances that are evaluated before a weight update in the network is performed.

- **Callbacks**
    - *Tensorboard* - Visualization tool provided by tensorflow where callback writes a log for visualize dynamic graphs of training and test metrics, as well as activation histograms.
    - *Earlystopping* - Stop training when a monitored quantity with respect to a mode has stopped improving working with a patience of certain number of epochs, for which there is no improvement of the system.
- **Validation-split**: The fraction of the training data held back on which we evaluate the loss and any model metrics at the end of each epoch without training it.

- Use Matplotlib, plotting library for the Python programming language, to take the model and produce histograms of the training data over different parameters.

Activation functions:

- **Sigmoid** is a non linear activation function that is capable of giving non binary activations with a smooth gradient unlike step function.But there is vanishing gradient problem that can be seen in the graphs produced below.

f(x)=1/(1+e^-x)

- **Relu** is Rectified Linear Unit - non-linear nature.
    - Not bound as it ranges from 0 to inf. Compared to tanh / sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLU can be implemented by simply thresholding a matrix of activations at zero.
    - Non-saturation of its gradient, which greatly accelerates the convergence of stochastic gradient descent compared to the sigmoid / tanh functions is the biggest advantage of relu. Using relu can kill the some weights for input less than 0 .

f(x)=max(0,x)

3.2 Human Observed Dataset

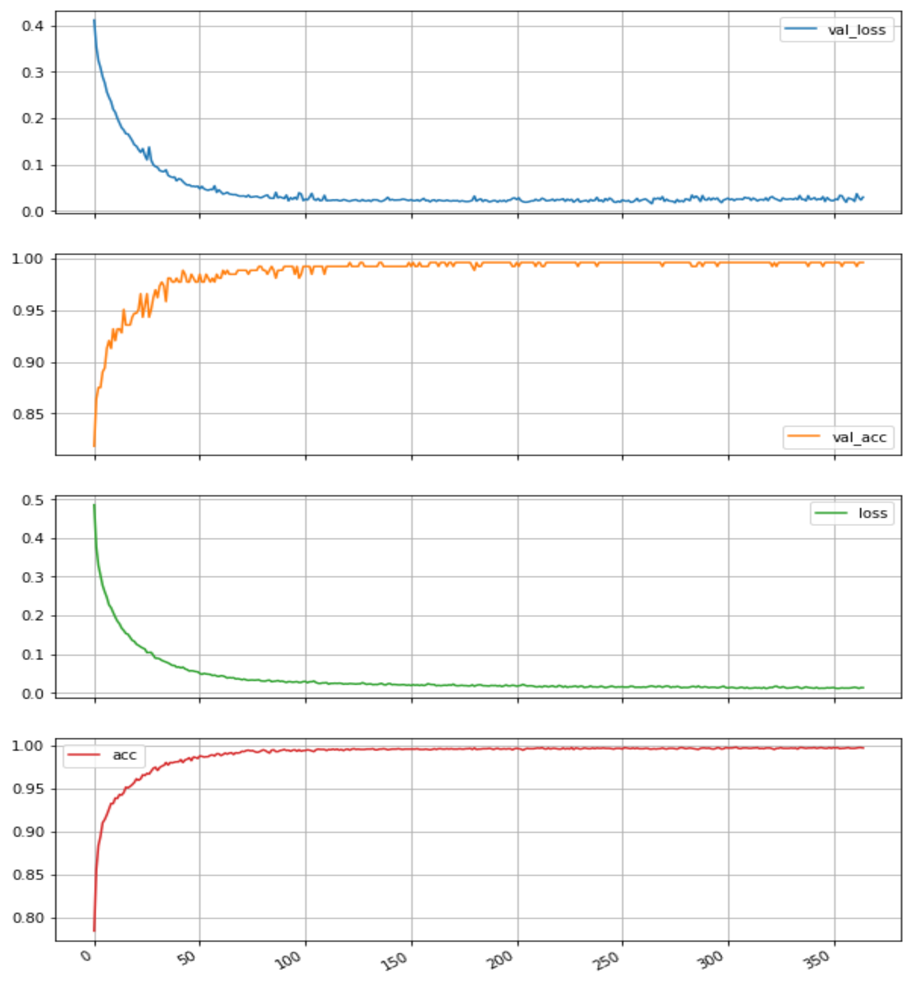
validation_data_split = 0.1,

num_epochs = 10000,

model_batch_size = 128,tb_batch_size = 32,early_patience =100

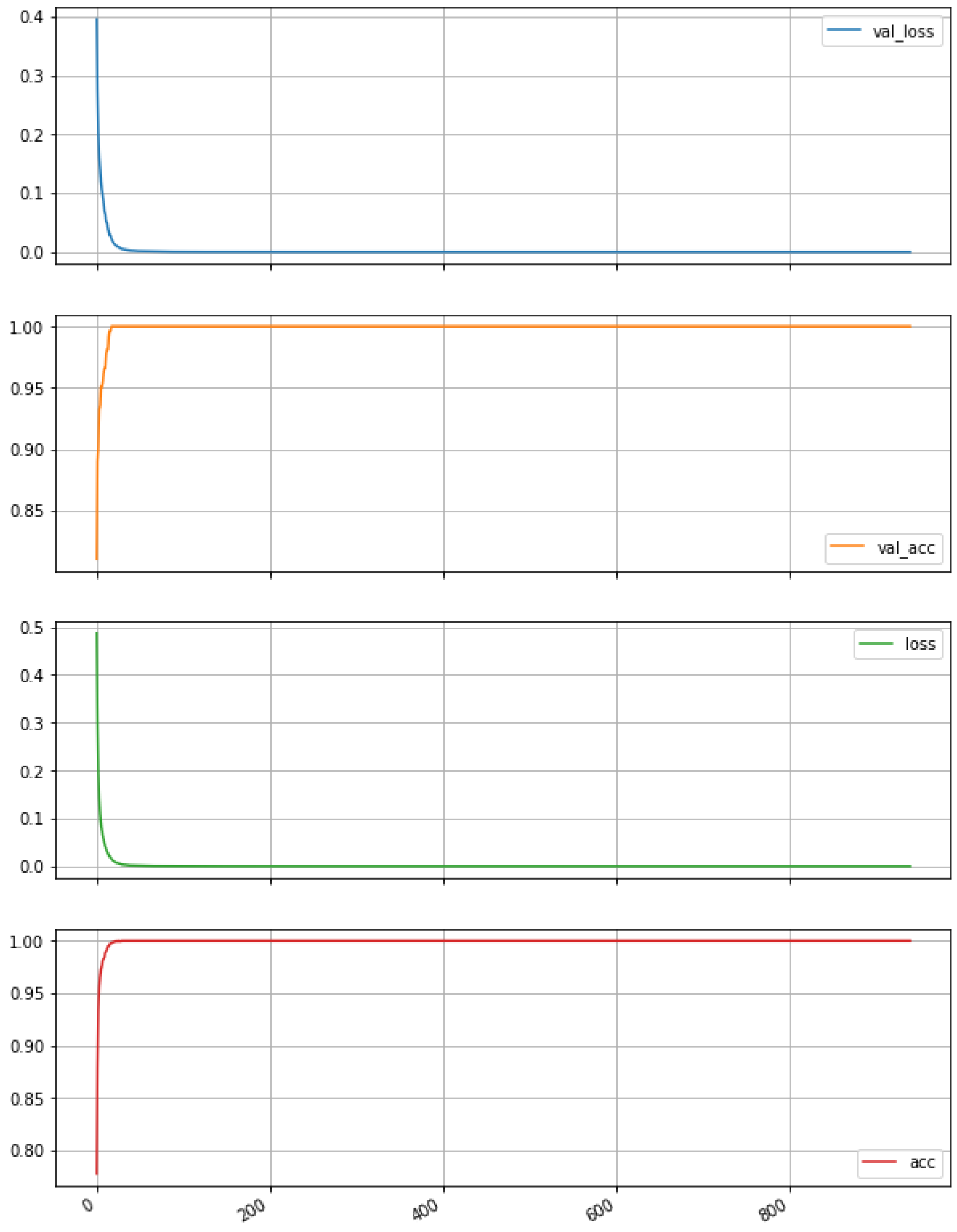
Concatenated Values

The Validation Accuracy on the human observed dataset is: 0.9787234042553191
The Validation Loss on the human observed dataset is: 0.07999092399632406
The Test Accuracy on the human observed dataset is: 0.9939024390243902
The Test Loss on the human observed dataset is: 0.03190572576869915



Subtracted Values

The Validation Accuracy on the human observed dataset is: 1.0
The Validation Loss on the human observed dataset is: 1.0647064534713340e-07
The Test Accuracy on the human observed dataset is: 1.0
The Test Loss on the human observed dataset is: 1.0486090738273871e-07



3.3 GSC Dataset

4. Conclusion

The following observations are made:

Dataset Size: As the size of the dataset on which the model training is done is increased, the accuracy values are also increased. But this leads to over-fitting of data as the model is over trained and cannot work properly on general data and loses generalization.

Regularization factor: As the value of the lambda increases, the accuracy of the model increases to some extent but as the value is increased further, the accuracy values keeps on decreasing this is because of overfitting of data.

Learning rate: The more the learning rate of the model, the faster it converges.This may lead to underfitting of data as it shoots to the convergent value quickly. Low learning rates delays the convergence of data and may lead to over-fitting sometimes.

Early stopping: Updating the weights in the model by calculating error at every data point may lead to over-fitting of data. The might be no updating of weights after certain iterations due to stagnation. Hence there we can stop doing number of updates after certain level. If the number of iterations is less, it may lead to under-fitting where as more number of iterations for the updating weights lead to over-fitting of data.

Number of basis functions: As the number of basis functions increases, the accuracy values of the model increases to some extent and then keep on decreasing as the number of basis functions increases. This is due to over-fitting of data for more basis functions and under-fitting of them for less basis functions.

<u>NN observations:</u>

- As the number of nodes increases, the value of the accuracy increases till some time where it achieves maximum value and then decreases afterwards.
- Dropout values range from 0-1 .Small dropout values lead to over-fitting.Large dropout values lead to under-fitting
- Accuracy increases with the increase in the number of epochs on the dataset and then decreases after certain point of time due to overfitting.