University at Buffalo

CSE 574: Introduction to Machine Learning
(Fall 2018)

Learning to Rank using Linear Regression

Author: Ameen M Khan
Person Number: 50288968

Problem Statement Develop predictive models for detection of crime dealing with evidence provided by handwritten documents. The project requires you to apply machine learning to solve the handwriting comparison task in forensics, where we map a set of input features x to a real-valued scalar target y(x,w) by using, Linear Regression, Logistic Regression and Neural Network(optional).

1 Methodology

In this project, linear regression, logistic regression and sequential neural network methodologies were applied to solve the handwriting comparison task in forensics. In both Linear and Logistic regression, stochastic gradient descent iterative optimization approach is used, which gradually tweaks the model parameters to minimize the cost function over the training set, eventually converging to the same set of parameters as the first method. For evaluating Linear Regression, E_{RMS} is used, and Logistic and Neural Network implementations are evaluated by directly assessing the prediction error.

1.1 Preparing the data

The pre-processing step for the Human Observed Dataset and GSC Dataset are similar. The Human observed dataset and GSC Dataset have 18 and 1024 features for a pair of handwritten sample respectively (9 and 512 for each dataset respectively). The entire dataset consists of both same writer pairs and different writer pairs, and the *final* data set consists of combination of the two set of samples (same number of samples from both). The final dataset have 4 variations by concatenating and subtracting the two datasets.

Datasets	Concat Features	Subtract Features	Samples
Human Observed Dataset	18	9	1791 (1000+791)
GSC Dataset	1024	512	2000 (1000+1000)

The data partitioning, like in previous project was done by splitting the preprocessed data into 80:10:10 for training, validating, testing purposes.

1.2 Linear Regression with Radial Basis Functions

For basis function in linear regression function ($y(w, x) = w^T * \phi(x)$), Gaussian Radial Basis functions were used. The SGD approach updates the weights (initialized randomly) using

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

where Δw^{τ} is the weight update which is the dot product of hyperparameter η and ∇E(gradient of the error).

1.3 Logistic Regression

Since the problem statement is a two class - classification problem, logistic regression is a promising approach to train the model. This approach is similar to the linear regression, besides the difference in the genesis function and error/cost function. The SGD approach, like in linear regression, updates the weights (initialized randomly) using

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

where Δw^{τ} is the weight update which is the dot product of hyperparameter η and ∇E(gradient of the error). The error function here is calculated by taking negative logarithm of the cross entropy error function and solving for its gradient, which gives :

$$\nabla E(w) = \sum (y_n - t_n)\phi_n$$

. The gradient of the error are are activated by the sigmoid function, to give the prediction probability as either 1 or 0, ie, the image pair either belongs to the same person or to different individuals

1.4 Neural Network (Sequential)

The network is made up of an input layer (encoded input data), one or more hidden layer (Dense), and one output layer consisting of 2 units (each corresponding to an output class). Hyperparameter tuning involved making network architecture decisions like selection of number of hidden layers, number of hidden nodes in each layer(256-1024). For our problem statement, which is a common classification task, I chose to stick to the general guidelines and used *categorical crossentropy* loss but preferred Adam optimizer because of faster convergence and better accuracy outputs at the default learning rates compare to the other ones, with default learning rate, as suggested by the official documentation.

2 Experiments and Observations

2.1 Linear Regression

The hyperparameters M, λ(Regularization Coefficients), η(Learning Rate) are tuned for the model using SGD approach for the different datasets. Following are the plots for accuracy for the said hyperparameters. When searching for optimal Learning Rate, M was kept at 10 and regularization coefficient at 0.03, and when searching for M, Learning Rate and 0.1 and same regularization coefficient as before.

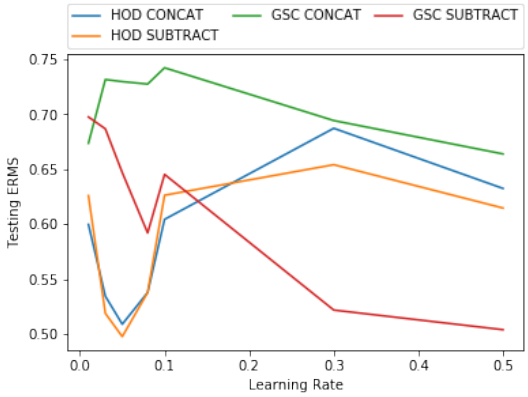


Figure 1: Testing Erms vs Learning Rate

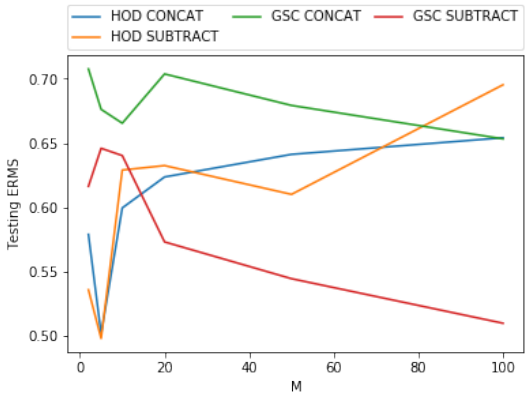


Figure 2: Testing Erms vs M

2.2 Logistic Regression

The hyperparameters λ(Regularization Coefficients), η(Learning Rate) are tuned for the model using SGD approach for the different datasets. The regularization coefficient is kept kept constant at 2 and optimal value for the learning rate for the 4 datasets is searched for, as displayed in the following plot:

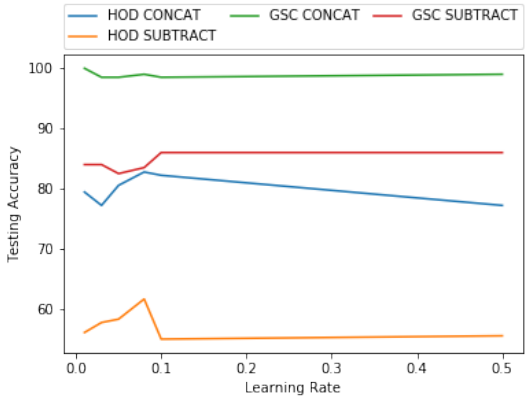


Figure 3: Accuracy vs Learning Rate

2.3 Neural Network

The hyperparameters, Hidden Layer and nodes within, activation methods for the layers, optimizer method, loss function and epochs are explored. As suggested by guidelines, the loss function used is categorical crossentropy. SGD, Adam, Adagrad, Adadelta optimizer are explored for all four datasets. Not much variation is observed in accuracy in said optimizer, but Adam did provide highest accuracy from them all, with the default learning rate value. Epoch value is set high, with early stopping capability provided by Keras-Tensorflow, which makes varying it inconsequential. When the number of hidden layers were increase, the accuracy dropped significantly for all datasets, so only single-layer sequential neural network is used for testing the hyperparameters.

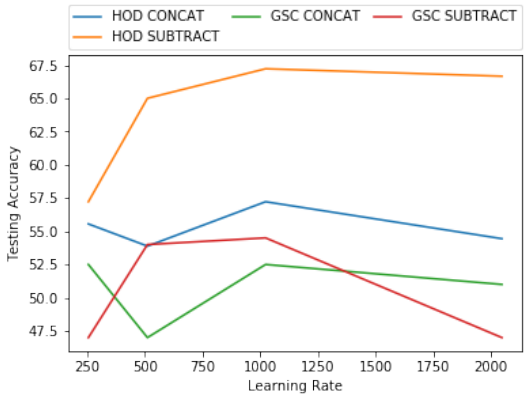
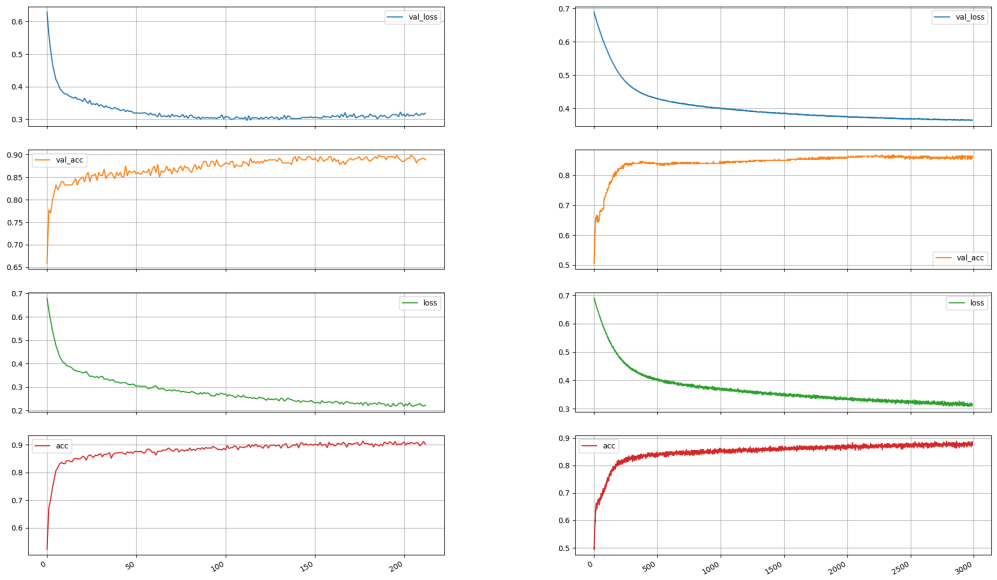


Figure 4: Accuracy vs Hidden nodes in first layer

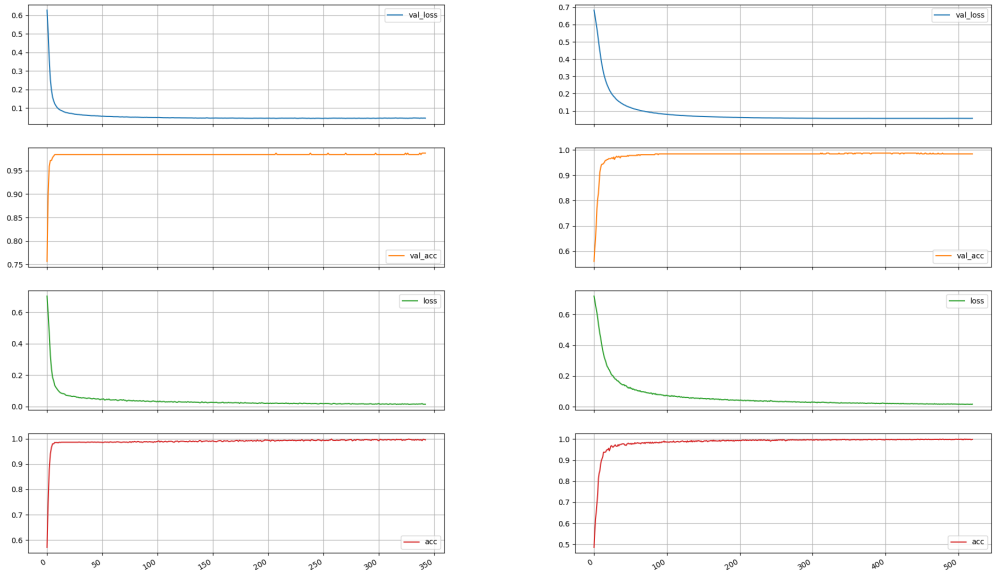
The above plot displays accuracy when changing the number of neural nodes in the single hidden layer.



(a) Human Observed Dataset, Concatenated Features

(b) Human Observed Dataset, Subtracted Features

Figure 5: Graphs for neural network accuracies for Human Observed Dataset



(a) GSC Dataset, Concataneted Features (b) GSC Dataset, Subtracted Features

Figure 6: Graphs for neural network accuracies for different GSC Dataset

3 Conclusion

The final testing accuracy of the different models for best possible hyperparameters are :

- **Linear Regression** : Hyperparameters (M, Learning Rates) are tuned for all datasets. The best results were for the GSC Dataset with concatenated feature values. The best values were achieved for the GSC Dataset with M = 10 and learning rate at 0.1, accuracy for which is 71
- **Logistic Regression** : Learning Rates are tuned for logistic regression for all sets of data. Similar to in case of Linear Regression, the best accuracies were observed for GSC Concatenated data, where accuracy at it's best at 99 percent.
- **Neural Network** : Optimal Setting of the architecture is single hidden layer dense neural network (1000 nodes in the layer), with Adam optimizer and categorical crossentropy loss function, relu activation in hidden layer and sigmoid in the output layer.

Also, as can be observed from the plots in the previous section, the GSC dataset, with concatenation of features of the pair of images, generally give higher accuracy for regression techniques, but in case of Neural Network, Human Observed dataset with substracted feature faired well when tuning different hyperparameter.

Implementation details and explanation of the above machine learning methodologies can be found within the code itself.

4 References

1. Hands-On Machine Learning with Scikit-Learn and TensorFlow, OReilly Publication
2. https://github.com/donnemartin/data-science-ipython-notebooks