University at Buffalo

CSE 574: Introduction to Machine Learning
(Fall 2018)

Classification - Handwriting Recognition

Author: Ameen M Khan
Person Number: 50288968

Problem Statement Implement machine learning methods for the task of classification of recognizing gray-scale handwritten digit image and identify it as a digit among 0-9 range. Train a Logistic Regression classifier, Deep Neural Network, Random Forest and a Support Vector Machine classifier on the MNIST digit images.
Combine the results of the individual classifiers to make a final decision (ensemble classifier).

1 Methodology

In this project, softmax logistic regression (mini-batch gradient descent), deep neural network, random forest and support vector machine methodologies were applied to classify 28×28 gray-scale handwritten digit image. The models are trained using MNIST dataset and tested on both the MNIST and USPS datasets. In addition to testing the classifiers on the two datasets, the No Free Lunch Theorem is verified.
All classifiers are evaluated using classification accuracy,

$$Accuracy = \frac{N_{correct}}{N}$$

1.1 Preparing the data

MNIST dataset is used for training the different models. The training, validation, and testing data are extracted directly from publicly available dataset. USPS dataset, which is used to test the pre-trained model, are extracted from images, and resized to 28*28 matrix.

1.2 Mini-batch Gradient Descent logistic regression

Since the problem statement is a multi class - classification problem, logistic regression with softmax activation is a promising approach to train the model. The targets are converted to 1X10 sized hot vector, each bit corresponding to different classes. A bias is also added in the training data. The softmax function is calculate using the equations

$$p(C_k) \mid \phi) = y_k(\phi) = \frac{exp(a_k)}{\sum exp(a_j))}$$

where y = softmax(a) and $a_k = w_k^T \phi$. The mini-batch gradient descent 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). The error function here is calculated by taking negative logarithm of the cross entropy error function and solving for its gradient, which gives :

$$\nabla_{w_j} E(w_1, ...w_{10}) = \sum (y_n - t_n)\phi_n$$

.

1.3 Deep Neural Network

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 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, compares to the other optimizers.

1.4 Random Forest classifier

A random forest fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. 'Information gain' criteria is used to measure the quality of a split for the decision trees and optimal number of trees for the forest are explored when tuning.

1.5 Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods favourable for our classification problem. Different kernel methods, i.e. linear and RBF are explored.

1.6 Ensemble Classifier

The results of the individual classifiers are combined using a classifier combination method, majority voting. Majority voting rule is a decision rule that selects outputs which have a majority, that is, more than half the votes. For our case, since there maybe instances when that can not be possible, or there may be a tie. So we using highest possible voted output as the resultant output. Random subset of training data are selected from the original dataset for training of each classifier before combining the outcomes of all the models.

1.7 Confusion Matrix

A confusion matrix (also known as error matrix) is a table that is used to describe the performance of a classifier on a set of test data for which the true values are known. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. Diagonal values indicates the classes that are correctly classified. Accuracy can be calculated using the confusion matrix, Acc= (True Positives + True Negatives)/total predictions.

2 Experiments and Observations

2.1 Logistic Regression

The hyperparameters λ(Regularization Coefficients), η(Learning Rate) are tuned for the model using mini batch GD (batching at 200 datapoints per epoch) approach for the different datasets. The regularization coefficient is kept kept constant at 0.01 and optimal value for learning rate is searched for.

The confusion matrix for the testing of the MNIST and USPS data is plotted and accuracy for both is calculated using it.

Supports **No Free Lunch Theorem**.

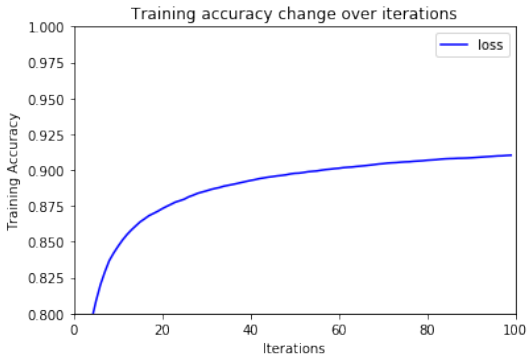


Figure 1: Training accuracy

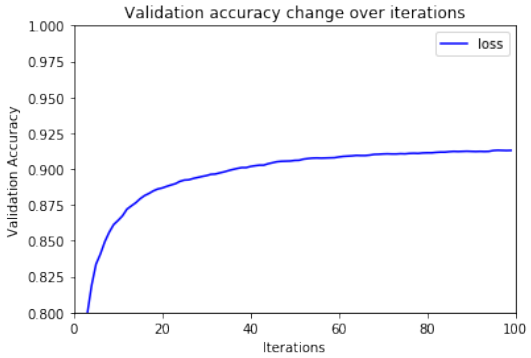
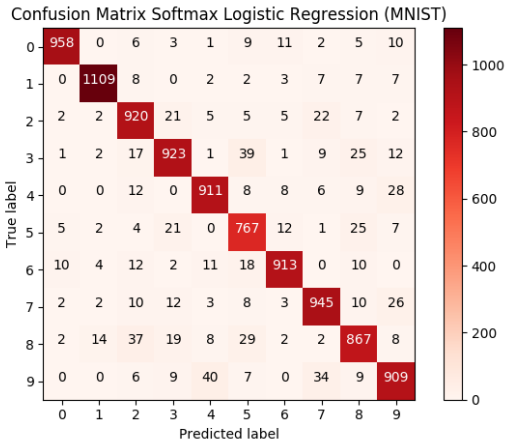
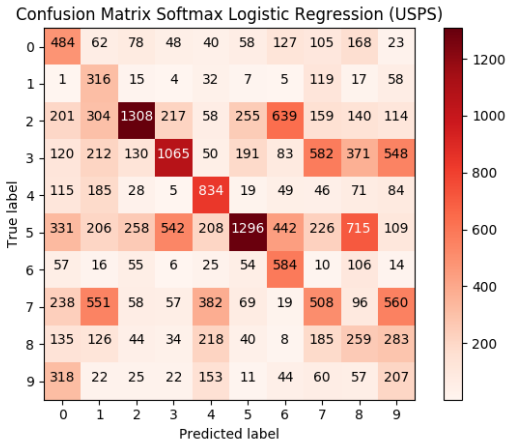


Figure 2: Validation accuracy



(a) MNIST Accuracy : 92.22



(b) USPS Accuracy : 34.30

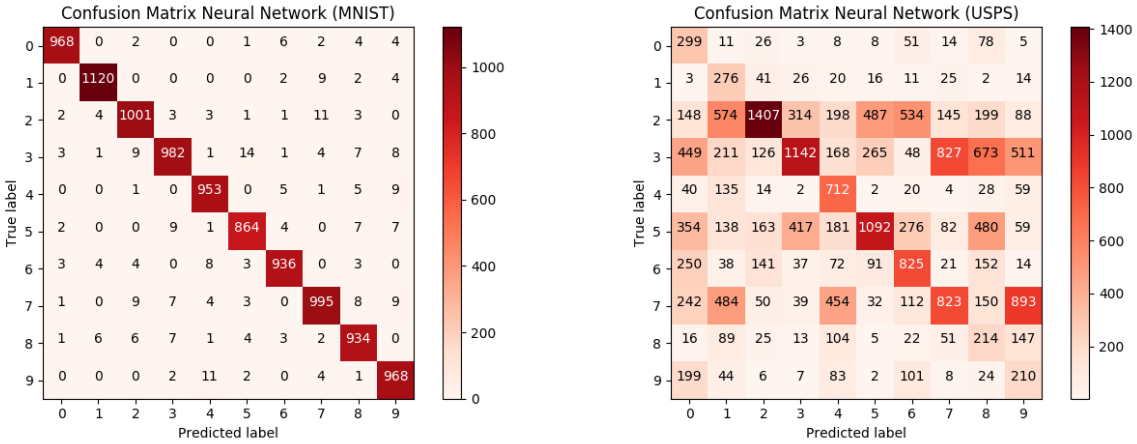
Figure 3: Confusion Matrix for Logistic Regression for the 2 datasets

2.2 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. Not much variation is observed in accuracy in

all available optimizers, but Adam did provide highest accuracy from 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.

The confusion matrix for the testing of the MNIST and USPS data is plotted and accuracy for both is calculated using it.



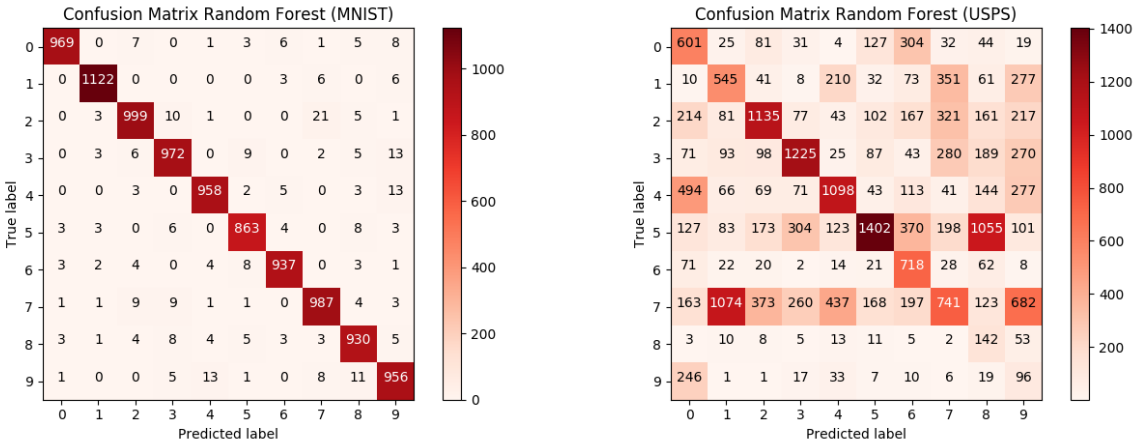
(a) MNIST Accuracy : 97.7 (b) USPS Accuracy : 35.01

Figure 4: Confusion Matrix for Deep Neural Network for the 2 datasets

Supports **No Free Lunch Theorem**.

2.3 Random Forest

The default settings for random forest are used. Grid search for number of trees yields best accuracy at 100 trees.



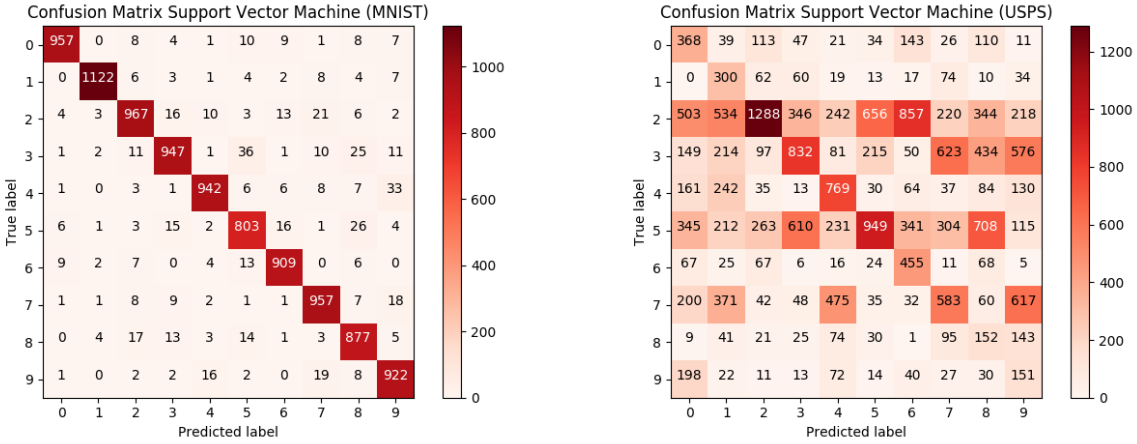
(a) MNIST Accuracy : 96.93 (b) USPS Accuracy : 38.52

Figure 5: Confusion Matrix for Random Forest for the 2 datasets

Supports **No Free Lunch Theorem**.

2.4 Support Vector Machines

The default settings for support vector machines are used. Linear and RBF kernels are compared, latter took a lot more time to converge, specially when using gamma=1, therefore linear kernel is used.



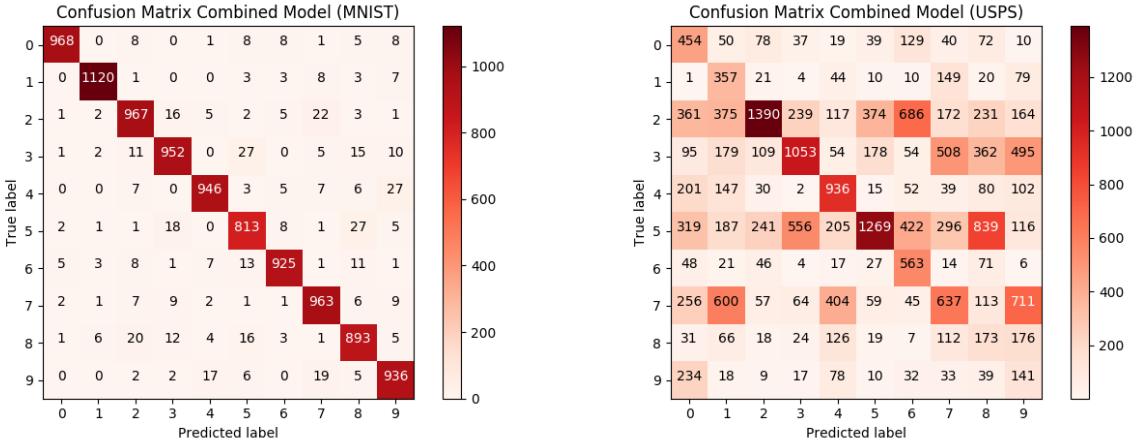
(a) MNIST Accuracy : 94.03 (b) USPS Accuracy : 29.24

Figure 6: Confusion Matrix for Support Vector Machine for the 2 datasets

Supports **No Free Lunch Theorem**

2.5 Ensemble Model (Bagging)

For training the models, the sub-sample dataset for each individual model are selected at random, and the predicted values are combined using majority voting.



(a) MNIST Accuracy : 95.05 (b) USPS Accuracy : 36.27

Figure 7: Confusion Matrix for Ensemble Model for the 2 datasets

Supports **No Free Lunch Theorem**

3 Conclusion

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

- **Mini-batch Logistic Regression**: The regularization factor, kept constant at 0.01, with mini-batch GD approach yields hist accuracy with 0.01 learning rate.
- **Deep Neural Network** : Optimal Setting of the architecture is single hidden layer dense neural network (64 nodes in the layer), with Adam optimizer and categorical crossentropy loss function, softmax activation in hidden layer and sigmoid in the output layer.
- **Random Forest**: The model, implemented using sklearn library, yields best possible accuracy for both the datasets with 'n estimator' at 100.
- **Support Vector Machine**: The model is trained with the linear kernel for faster convergence.

As can be observed from the confusion matrix for all the models, for MNIST dataset, the models were best able to identify digit 1, as compared to the other digits, whereas for the USPS dataset, the digit 2 was best classified, closely followed by digit 5, in contrast to the other digits All the models support the **No Free Lunch Theorem**. From the individual models, Deep Neural Network yields best possible accuracies for both the MNIST and the USPS testing data. 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. Random Forest implementation from scikit-learn.
3. Support Vector Machine Implementation from scikit-learn.