MNIST Image Classification

Dhayanidhi Gunasekaran
University at Buffalo
dhayanid@buffalo.edu

Abstract

To implement machine learning for the task of classification.

1 Project Statement

To implement the MNIST image classification problem using various classification models such as Logistic regression, Neural Network, SVM and Random Forest classifier. Also, to implement the ensemble of all the classifiers and to combine the results of individual classifiers. Finally, to test the trained model with USPS Dataset.

2 Solution

2.1 Data Preparation

The given set of MNIST images are processed to training, validation and testing dataset. Each image is of shape 28 * 28 which is then processed to form a single dataset containing 784 features. The training dataset contains 50,000 images while the validation dataset and testing dataset contains 10,000 images.

The model is trained using training dataset while the hyperparameter tuning is done with validation dataset. Finally, the model accuracy is calculated using testing dataset.

2.2 Logistic Regression

The problem of MNIST image classification is solved by Multinomial Logistic regression in which the logistic regression model classifies the input images in to more than three categories. Since there are ten different numerical digits this is 10 class Logistic regression where the output ranges from 0 – 9.

Logistic regression is similar to linear regression in which the model is trained and the output is generated with the only difference that the output is probabilistic to classify the input to output categories.

The following is the representation of Logistic regression using input features, Weight and output. The input feature phi can be denoted as

$$\phi=[\phi_1,.., \phi_M]^T$$

The activation can be found by using the following formula

$$a_k = \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{\phi} + b_k,$$

where w is the weight and b is the bias.

The obtained activation is then converted to the range of 0 -1 using Softmax activation function.

Y = Softmax(a)

where the Softmax function is represented as

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

2.2.1 One hot vector representation

One hot encoding transforms categorical features to a format that works better with classification and regression algorithms. In the given problem, there are 10 output categories for the integer values from 0-9.

For example, integer 3 can be represented as follows.

3 = [0 0 0 1 0 0 0 0 0 0]

2.2.2 Gradient Descent solution

The loss can be computed using the following formula,

$$E(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{10}) = -\ln p(T \mid \boldsymbol{w}_1, .., \boldsymbol{w}_{10}) = -\sum_{n=1}^{N} \sum_{k=1}^{10} t_{nk} \ln y_{nk}$$

Since we use one hot encoding to represent the target, the values which representing the particular digit gets contributed the loss value while the remaining values in the matrix produces zero value.

The gradient of the error function can be found as

$$\nabla_{\mathrm{w}_j} E(\mathrm{w}_1, \ldots, \mathrm{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \phi(\mathrm{x}_n)$$

The computed gradient is subtracted to the weight and the process is repeated until the model converges.

$$\mathrm{w}^{\tau+1} = \mathrm{w}^{\tau} - \eta \nabla E_n$$

2.2.3 Evaluations

The loss graph is plotted as follows.

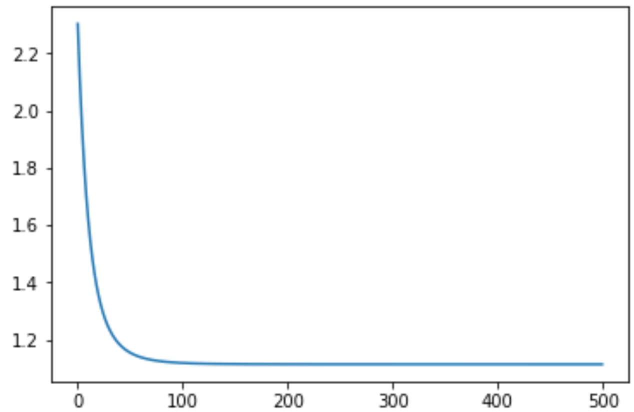

Figure 1: Loss graph

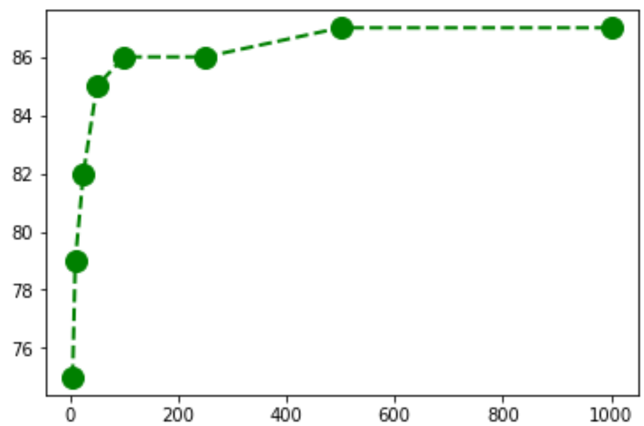

Figure 2: Iterations vs Accuracy

The hyperparameters used for training the model is listed below.

Hyperparameter	**Value**
Learning Rate	0.1
No. of iterations	500
Regularization factor	0.1
Accuracy MNIST	87%
Accuracy USPS	33%

Table 1: Hyperparameters

2.2.3.1 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Confusion matrix for MNIST Logistic regression

```
[[ 949    0    2    2    0    2   16    1    8    0]
 [    0 1100    4    3    1    1    4    0   22    0]
 [  15   25  845   26   18    0   28   22   45    8]
 [    5    4   21  891    1   24    8   17   25   14]
```

```
 [   3   10    5    0  861    0   16    2   10   75]
 [  28   17    3   91   21  628   31   11   45   17]
 [  20    5   13    2   13   17  883    0    5    0]
 [   4   42   24    1   13    0    4  887    9   44]
 [  11   28   13   39   12   18   17   15  801   20]
 [  15   14   10   13   51    9    2   27    9  859]]
```

Confusion matrix for USPS Logistic regression

```
[[ 711    5  397   48  330   39   73   37   91  269]
 [ 288  307  155  267  286   33   45  285  320   14]
 [ 288   44 1116  123   75   39  108  100   89   17]
 [ 168    4  140 1158   46  173   48   76  119   68]
 [ 127  105   37   46 1098   86   25  120  247  109]
 [ 251   26  220  244   57  834  149   83   96   40]
 [ 539   17  366   97  119   92  648   26   65   31]
 [ 218  262  342  373   72   68   44  283  301   37]
 [ 286   46  180  210  184  386  140   39  442   87]
 [ 109  238  170  409  195   55   18  361  319  126]]
```

2.3 Neural Network

Neural network is the network of artificial neurons or nodes connected together mathematically to solve machine learning problems. There are number of architectures in neural networks such as perceptron, Convolutional neural network, Recurrent neural network, Long short term memory and so on.

Dense neural network is regular neural network in which each neuron receives input from all the neurons in previous layer thus called as dense neural network. The layer has a weight, bias b and activation of previous layer.

In our implementation of neural network, we have used two activation functions such as Sigmoid and Softmax Activation function. It is a two hidden layer neural network of classes 64 and 10 respectively.

2.3.1 Evaluations

The evaluation metrics for the neural network by tuning the hyperparameters are as follows.

Hyperparameter	**Value**
Number of epochs	10
Batch size	128
No. of layers	2
Accuracy MNIST	92%
Accuracy USPS	30%

Table 2: Hyperparameters for Neural network

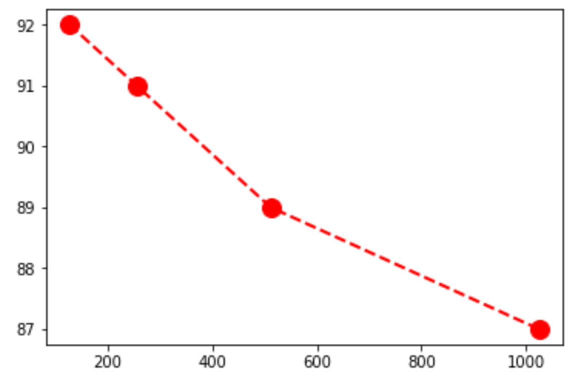

Figure 3: Batch size vs Accuracy

Confusion matrix for MNIST

```
[[ 951    0    3    2    2    6   11    1    4    0]
 [   0 1103    3    4    0    0    4    1   18    2]
 [   9    1  932   17   11    6   10   15   29    2]
 [   3    0   23  922    0   26    1   11   16    8]
 [   1    4    4    0  895    1   11    1    3   62]
 [  11    2    4   47    7  776   11    5   20    9]
 [  16    3    7    1   12   13  900    1    5    0]
 [   3   10   22    7    6    0    0  943    5   32]
 [  12    3    6   22    8   23   10   12  871    7]
 [  14    0    2   12   22   10    1   19    8  921]]
```

Confusion Matrix for USPS

```
[[ 765    0  523   44   81  199    7   45   30  306]
 [ 187   66 1040  295    4  138    1  250   11    8]
 [ 131    0 1633   59    9  136    0   14   12    5]
 [ 130    0  392 1091    0  349    1   27    5    5]
 [ 158   23  318   82  554  317    1  227  145  175]
 [ 169    4  384  121    4 1267    3   30   12    6]
 [ 377    0 1140   23   36  303   91   12    5   13]
 [ 230   23 1017  374    4  109    0  212   13   18]
 [ 292    4  492  168   24  857   10   28   89   36]
 [  61   53  427  474   27  225    0  429  173  131]]
```

2.4 Support Vector Machines

Support vector machines are the supervised learning model with associated learning algorithms that analyze data used for classification and regression. The idea behind SVM is constructing the hyperplane which is used to classify the data. There are many hyperplanes that might classify the data. The best hyperplane is the one that represents the largest separation or margin between the classes. So the hyperplane is chosen based on the distance from the nearest data point on either side is maximum. There are types of SVM, of which Radial Basis Function (rbf) is the one that is implemented in this classification.

2.4.1 Evaluations

The evaluation metrics for the Support vector machine by tuning the hyperparameters are as follows.

Hyperparameter	**Value**
Kernel	Radial Basis Function(rbf)
C range	2
Gamma range	0.05
Accuracy MNIST	98%
Accuracy USPS	26%

Table 3: Hyperparameters for SVM

Confusion matrix for MNIST

```
[[ 982    0    5    0    0    0    1    0    1    2]
 [   0 1056    1    2    0    0    2    1    2    0]
 [   1    0  980    0    0    1    0    3    5    0]
 [   0    0    3 1007    0    6    0    1   11    2]
 [   0    5    0    0  969    0    0    1    2    6]
 [   2    0    3   10    2  887    4    1    5    1]
 [   2    0    0    0    1    1  963    0    0    0]
 [   0    6    5    0    1    0    0 1071    0    7]
 [   1    0    4    4    0    3    1    0  995    1]
 [   2    3    2    7    8    3    0    5    6  925]]
```

Confusion matrix for USPS

```
[[ 226    0 1564    2   26   35    2    0   79   66]
 [  78  257  713  172  262   77   12  337   88    4]
 [   8    0 1944    6    2   20    1    6   11    1]
 [   4    0 1193  725    0   41    0    0   37    0]
 [   6    0 1045   18  522   96    0   56  252    5]
 [  15    0 1305   16    1  626    0    0   37    0]
 [  78    0 1534    2   10   61  290    0   22    3]
 [  17    6 1435  129    6  134    0  220   52    1]
 [   7    0 1387   14    4  221    0    0  367    0]
 [   1    0 1508   79   26   29    0   39  267   51]]
```

2.5 Random Forest Classification

Random forest is the supervised learning algorithm where it builds an ensemble of decision trees most of the time trained by bagging method. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.
Decision tree is a model in which an item is observed and the target value about an item is predicted. In Classification model, the target variables get a discrete set of values.

2.5.1 Evaluations

The evaluation metrics for the Random forest classification by tuning the hyperparameters are as follows.

Hyperparameter	Value
No of estimators	10
Accuracy MNIST	94%
Accuracy USPS	31%

Table 4: Hyperparameters for Random Forest classifier

Confusion Matrix for MNIST dataset

```
[[ 978    0    1    2    1    0    2    0    5    2]
 [   0 1049    7    4    0    1    1    1    0    1]
 [   4    1  954    4    3    1    2   12    6    3]
 [   3    1   11  971    1   17    0    3   18    5]
 [   2    3    4    3  931    3    6    2    6   23]
 [   9    0    5   44    3  828   11    1    9    5]
 [   3    2    2    0    5    6  945    0    3    1]
 [   2    9   13    3    6    1    1 1045    1    9]
 [   3    8   12   21    3   18    4    6  922   12]
 [   5    0    2   13   21   12    4   12    7  885]]
```

Confusion Matrix for USPS dataset

```
[[ 730   36  282   80  322  172   91  123   22  142]
 [ 111  582  208  145  140   60   64  652   26   12]
 [ 195  146 1038  133   62  147   60  163   22   33]
 [ 109   70  278  973   67  315   20   98   19   51]
 [  77  207  163   88  793  156   53  311   59   93]
 [ 271   74  221  233   59  927   50  122   21   22]
 [ 431  103  355  115  131  235  506   83   17   24]
 [  70  372  526  192   56  205   39  487   14   39]
 [ 181  120  317  315   96  616   78   96  136   45]
 [  87  299  369  305  215  135   39  388   55  108]]
```

3 Questions to be answered

3.1 No Free lunch Theorem

No free lunch theorem states that there is no optimization technique which is the best for the generic and all special cases. We have trained our model using MNIST dataset and got the accuracy of nearly 95 percent which shows that the model trained is a best technique for the MNIST dataset. However, the same model does poor classification with USPS dataset which is also the same image classification problem which gives the accuracy of 30 percent.

This proves the No Free lunch theorem by which the logistic regression is best technique which works well for all the generic cases.

3.2 Confusion Matrix and best classifier

The confusion matrix for the all the classifier is mentioned in the previous section. Based on the metrics provided, Support vector Machine classifies the images with highest accuracy of 98%.

3.3 Ensemble Techniques

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. In this implementation we combine all the 4 classifiers such as Logistic, Neural Network, SVM and Random Forest together to produce a single classification model.

Majority voting

It is type of voting classification, in which the results of the individual data of all the classifiers are considered and the one with more than 50% matching is considered to be the result. For example is a particular image is to be classified into particular category say 5, then more than half, which is minimum 2 classifier should predict that particular image as 5.

The combined accuracy of the ensemble model is 96% which is less than the SVM's Accuracy of 98%.

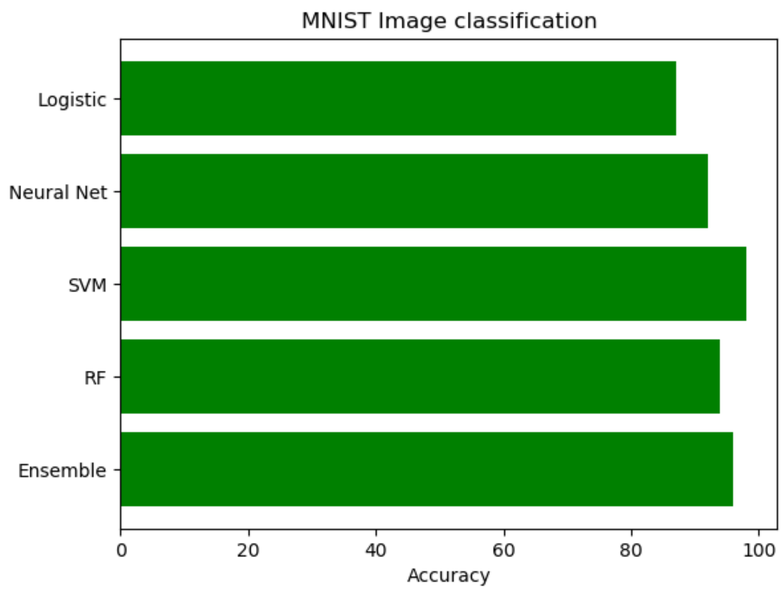

Figure 4: Classifiers vs Accuracy

References

[1] https://matplotlib.org/api/api_overview.html

[2] https://keras.io

[3] https://scikit-learn.org/stable/modules/ensemble.html