Machine Learning Homework 2 Report

Kasturi Chandra Shekhar
UTA-ID:1001825454

The University of Texas at Arlington
Computer and Information Sciences

CSE 6363 Machine Learning
Prof.Won Hwa Kim

Submission-date: 11/05/2020(late submission)
Using One of my late submission days.

0.1 Problem 1 Logistic Regression

The Logistic regression is a statistical model conducted when the dependent variable is dichotomous i.e; binary-(win/loss),(0/1),(pass/fail).Like all other regression analyses,logistic regression is a predictive analysis.In a binary logistic-regression model it classifies between two levels whereas in multinomial logistic regression it categorizes the data into multiple levels.

$$\mu_1 = [1, 0], \ \mu_2 = [0, 1], \ \Sigma_1 = \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix}, \ \Sigma_2 = \begin{bmatrix} 1 & -0.5 \\ 0.5 & 1 \end{bmatrix}$$

For this problem the training a $random.multivariate_normal(mean, sigma, samplesize)$ dataset of 1000 samples have been obtained with the mean and sigma values stated above and assigned class-labels 0 and 1. In the similar fashion the testing dataset is obtained by sampling 500 samples from each class-label.Logistic Regression is performed by calculating the sigmoid values of the net value $sigmoid(z)$ and computing the difference between the sigmoid scores and train-labels then the gradient is minimized to find the minimum-points by adjusting the weights and training the model. The overall loss is calculated by cross-entropy loss function.

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

The number of iterations is set to 10000 but the regression stops if the loss function converges or if the norm of the gradient becomes small.The initial dataset and the regressed data is as shown below.

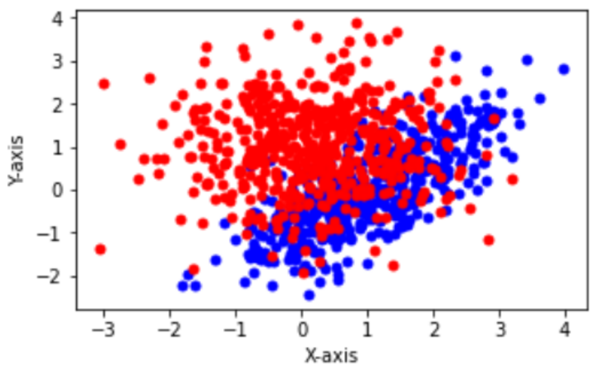

Figure 1: Initial dataset

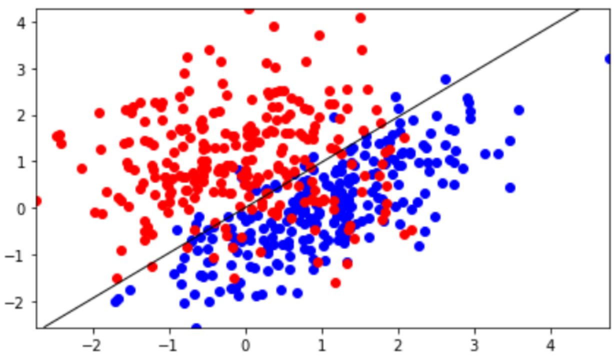

Figure 2: Regressed Data

The cross-entropy losses,accuracy for training and testing data and the number of iterations for different values of learning rate is obtained as follows:

```
cross-entropy loss = 0.49481710020951786 learning-rate= 0.0001 Number of iterations= 10000
Train Accuracy: 0.804%
Test Accuracy: 0.836%
cross-entropy loss = 0.43947519011859604 learning-rate= 0.001 Number of iterations= 10000
Train Accuracy: 0.808%
Test Accuracy: 0.836%
the loss converges at iteration= 3484 with loss value= 0.4393679880608745 when learning rate= 0.01
Train Accuracy: 0.806%
Test Accuracy: 0.836%
the loss converges at iteration= 2568 with loss value= 0.439367988060787 when learning rate= 0.1
Train Accuracy: 0.806%
Test Accuracy: 0.836%
the gradient vanishes at iteration= 269 with norm value= 9.826443995479118e-09 when learning rate= 1
Train Accuracy: 0.806%
Test Accuracy: 0.836%
```

The ROC Curve is plotted by calculating the TruePositive Rate(tpr) vs False Positive Rate(fpr) showing the performance of the binary classification model at all classification thresholds. The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.The AUC value provides an aggregate measure of performance across all possible classification thresholds.

A graph is plotted for parameter learning rate against the number of iterations it takes for each learning rate to regress the model logistically.The number of iterations decreases with the increase in the learning rate.If the learning rate is too small, the algorithm becomes slow because many iterations are needed to converge at the (local) minima.On the other hand, if learning rate is too large, you may overshoot the minima and risk diverging away from it.

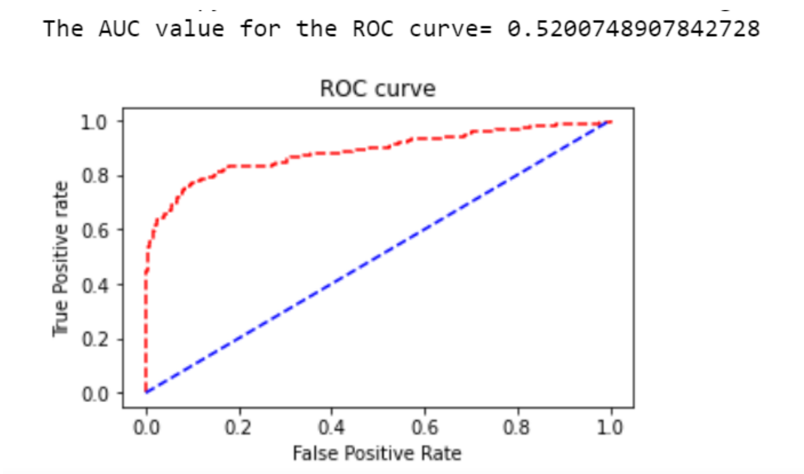

Figure 3: ROC Curve and AUC value

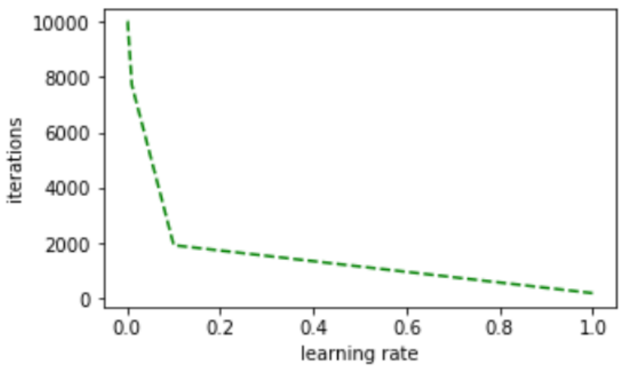

Figure 4: Iterations vs Learning-rate

0.2 Problem 2 Multi-class Logistic Regression

Multi-class logistic regression generalizes the logistic regression to multiclass problems with more number of outcomes Given a set of data it predicts the probabilities of the possible values of categorical data variable independent of the the other variables.Multinomial logistic regression is a particular solution to classification problems that use a linear combination of the observed features and some problem-specific parameters to estimate the probability of each particular value of the dependent variable.In this problem we use MNIST data set with class labels ranging from [0-4] thereby reducing the dataset from setting the maximum number of iterations to 10000 and using a softmax function as the objective function to regress the data for classification and cross-entropy as the loss function.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

The softmax function is used to predict our labels as probabilities which are strictly positive.The net value $z = Wx + p$ is passed to the softmax function (which is a more genralized activation function for multiclass classification) and the respective model parameters are trained to predict the data.The cross-entropy loss function calculates the probability of each element to be predicted correct.The loss is minimized using gradient descent to converge to a minimum value by updating the weights of the model with the given learning rate.

The accuracy determines how may values in the dataset are predicted correct by evaluating it with the test data.The value precision determines that how many instances are relevant among the predicted values whereas the value recall depicts that how many relevant instances are actually retrieved .Both precision and recall are calculated by using the TPR(True Positive rate),FPR(False positive rate),FPR(False positive rate) and FNR(False Negative rate) values.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

```
accuracy of softmax-regression:  0.9735357073360577
precision of softmax regression 0.9733759992249394
recall of softmax regression 0.9733855446123956
```

Figure 5: Accuracy Precision and Recall for softmax-regression

0.3 Problem 3 Artificial Neural Network

Artificial Neural Networks are a series of algorithms that mimic the operations of a human brain to recognize relationships between vast amounts of data.The basic concept of neural networks is that it can adopt to changing inputs so that it would avoid redesigning the output criteria.A neuron in the neural network can be seen as a mathematical function that collects and classifies the data according to the architecture by learning using the internal-weighting system to produce the output.

In this problem we use the same MNIST dataset by using a subset of the entire data using only classes from [0-4] similar to problem 2. The training and testing data are reshaped to be stacked on the perceptron.The first layer in the perceptron is the Flatten layer which converts the data pooled feature map to a single column which is then passed to a dense layer with the respective activation functions adding to the neural network.The loss function used is cross-entropy to evaluate the performance of the neural network.

By comparing the values of accuracies obtained for the perceptron networks using sigmoid and relu as activation function we get the following outputs:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	980
1	0.97	0.99	0.98	1135
2	0.95	0.91	0.93	1032
3	0.95	0.95	0.95	1010
4	0.95	0.98	0.96	982
accuracy			0.96	5139
macro avg	0.96	0.96	0.96	5139
weighted avg	0.96	0.96	0.96	5139

Figure 6: ANN with sigmoid

	precision	recall	f1-score	support
0	0.99	0.99	0.99	980
1	0.99	0.99	0.99	1135
2	0.96	0.94	0.95	1032
3	0.97	0.97	0.97	1010
4	0.97	0.98	0.98	982
accuracy			0.98	5139
macro avg	0.98	0.98	0.98	5139
weighted avg	0.98	0.98	0.98	5139

Figure 7: ANN with relu

Both the activation functions are monotonic but the the relu function is less computationally expensive when compared to the sigmoid which is considered good for training a neural network. With the sigmoid function the problem arises at the end where the predicted values so no respond to the changes in the inputs because the gradient vanishes or becomes too small which inhibits the network to learn further. With the relu function the sparsity of the the activation is vast meaning the activation of every datapoint is considered for determining the output there by increasing the accuracy from 0.96 of sigmoid to 0.98 relu.

The Accuracy for neural networks is always higher when compared to the regression because the regression tries to fit a straight line(the output being categorical data for logistic-regression) even for non-linear data which might not be possible for sparsely scattered data whereas the neural network tries to fit circles, ellipses to the same data thereby increasing the probability of classification.NN could outperform the regression model since it using a sophisticated architecture with designed activation function, such as ReLu, tanh, sigmoid, and even regression model by which the accuracy increases as it tries to compute multiple regression functions to get the output.Neural networks deal with the non-linearities in the data so if we have non-linear dependencies the NN model works better than the regression model.