Project 1: Logistic Regression

Sambo Dutta
University at Buffalo
UBIT Name: sambodut
UB PERSON NUMBER: 50318667
sambodut@buffalo.edu

Abstract

Regression is a statistical measurement to determine the relationship between one target variable and a series of changing variable.It is mostly used in the field of health science, finance and recommendation to predict a target value based on a series of given data.In this project we have implemented Logistic Regression,a type of regression which gives a binary result.A model was trained based on Breast Cancer Wisconsin Data Set which was later used in making prediction for patients to determine if they were diagnosed with breast cancer.

1 Introduction

Regression analysis is a statistical process to determine the relationship between a set of variables.Usually there is a dependent variable which is calculated based on a set of independent variables.Logistic Regression is a type of regression which uses a sigmoid function to get a categorical result.It is mostly used when we have to categorise data between two classes like in the case of email classification to check if a mail is a spam or not spam or in the field of health science to determine if a tumour is malignant or benign.In this project we have used a cancer dataset to develop a model in order to make a prediction if a patient is suffering from breast cancer.This health dataset was divided into three categories- training,testing,validation.The training part was used in mainly developing a model which would later make predictions for the testing data.This model was evaluated using evalutaion metrics like accuracy, precision and recall and a detailed analysis is given in the results section.

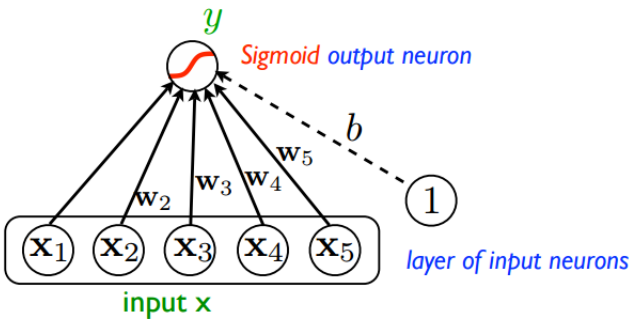

Figure 1: Logistic Regression Architecture

2 Algorithm

Step 1: The dataset was normalised after dropping the index column.
Step 2: After normalization,we split the dataset into three parts- training, testing and validation in the ratio 8:1:1.
Step 3:Randomly intialize the weight vector
Step 4:Start the training process:

For epochs in range(1 to 100(may vary)):

Repeat for every sample in the training data set:

Calculate the dot product between weights and the input sample and add bias, say $z = w^T X + b$.

Use a sigmoid activation function to classify the input sample.

Use Gradient descent method to adjust the weights accordingly.

For every epoch calculate the training and validation error.

Step 5. Use the adjusted weights to classify the testing data and measure the accuracy,precision and recall accordingly.

2.1 Sigmoid Activation Function

A sigmoid function is a S-curved function which is mainly used to map variables in the range 0 to 1. The following equation is mostly used as the sigmoid activation function:

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

On doing a dot product between the weight vector and the training input vector we might get a very large value, to avoid this we are using the sigmoid activation function to get the values in a smaller sub-range.

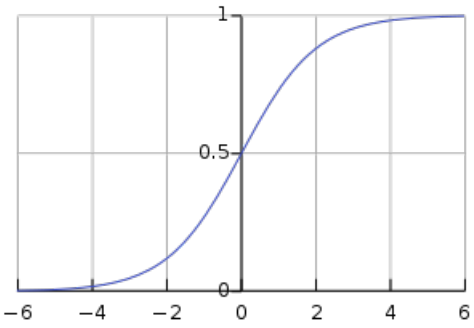

Figure 2: Plot of Sigmoid Activation Function

2.2 Tuning Parameters using Gradient Descent

Our goal is to minimize the loss function. y is the 1D vector containing class labels for all the training samples. X is the training data set of m * d size where d is the number of features. a is the 1D vector containing the predicted class labels for all the training samples. In this case we use cross entropy loss. The equation is given below:

$$L = \frac{-(ylog(a) + (1-y)log(1-a))}{m} \tag{2}$$

$$= -\frac{1}{m}(ylog(\sigma(w^T X)) + (1-y)log(1-\sigma(w^T X))) \tag{3}$$

$$= -\frac{1}{m}(ylog(\sigma(z)) + (1-y)log(1-\sigma(z))) \tag{4}$$

where m is the number of training samples, a is the predicted class label and y is the actual class label.

The weights and the bias are updated based on the gradient descent. The following formula is used to update the weights:

$$w_{new} = w_{old} - \eta \Delta w_{old} \tag{5}$$

where η is the learning rate.The learning rate value should not be very large because then the weight would not get its optimum value. Also if the learning rate is too small there is a high chance we might end up with a local minima.

$$\begin{aligned}
\Delta w_i = \frac{\partial L}{\partial w_i} &= -\frac{1}{m}\frac{\partial}{\partial w_i}(ylog(\sigma(z)) + (1-y)log(1-\sigma(z))) \\
&= -\frac{1}{m}(y\frac{1}{\sigma(z)}\frac{\partial}{\partial w_i}\sigma(z) + (1-y)\frac{1}{1-\sigma(z)}\frac{\partial}{\partial w_i}(1-\sigma(z))) \\
&= -y\frac{1}{\sigma(z)}\sigma(z)(1-\sigma(z)\frac{\partial}{\partial w_i}(z) + \\
&(1-y)\frac{1}{1-\sigma(z)}\sigma(z)(1-\sigma(z))\frac{\partial}{\partial w_i}(-(z)) \\
&= -\frac{1}{m}(y(1-\sigma(z))x_i + (1-y)\sigma(z)(-x_i)) \\
&= -\frac{1}{m}(y - y\sigma(z) - \sigma(z) + y\sigma(z))x_i \\
&= -\frac{1}{m}(y - \sigma(z))x_i
\end{aligned} \tag{6}$$

3 Experiments

3.1 Dataset

The dataset(Breast Cancer Wisconsin Data Set) used in this experiment has a dimension of 568*32.Each row has details of a particular patient where the column 'M' is the target variable which states either malign or benign condition.This target variable is based on the other features of the patient.We have divided the dataset into three parts.The training part consists of 80 percent of the data which was used to build the model while 10 percent was used for testing and validation each. For testing we have used the model derived from the training data and have made prediction whether the condition is benign or malignant.Later we have evaluated the predicted results with the actual results.

3.2 Normalization

Before the splitting the dataset into training,testing and validation,we are normalizing the dataset.The dataset has 30 unique features in which individual features may have in its own ranges.While some of these features may have a very low range, others which have a higher range might overpower its weightage and might become a more dominating feature just because of its higher range. So in order to avoid these kind of situations we are normalizing the data in the range 0 to 1 for each particular feature, so that each features have equal weightage.We have used the equation below to normalize each feature:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

3.3 Evaluation Metrics

In order to check the efficiency of the model we have used accuracy,precision and recall.

$$Accuracy = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \tag{8}$$

$$Precision = \frac{t_p}{t_p + f_p} \tag{9}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{10}$$

True positive(t_p) is the number of times the model could rightly predict class 1,in this case Malignancy while true negative(t_n) is the number of times it could correctly predict class 0, Benign status.False positive(f_p) and false negative(f_n) is the number of times the model was wrong in guessing the Malignancy and Benign status respectively.

4 Results

We have evaluated the results in two sections.In the first section we have varied the epoch.On increasing the epoch we see that all the three metrics accuracy, precision and recall increase slowly with each iteration.This is because in the initial stage the model in not fully developed, while with each epoch iteration we get a better trained model which gives a better classification.

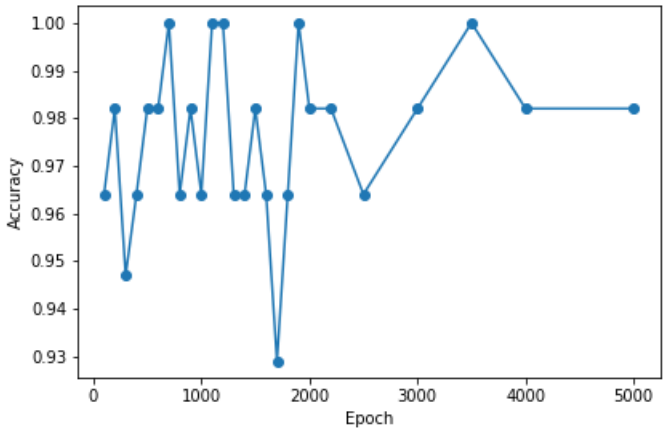

Figure 3: Accuracy vs Epoch

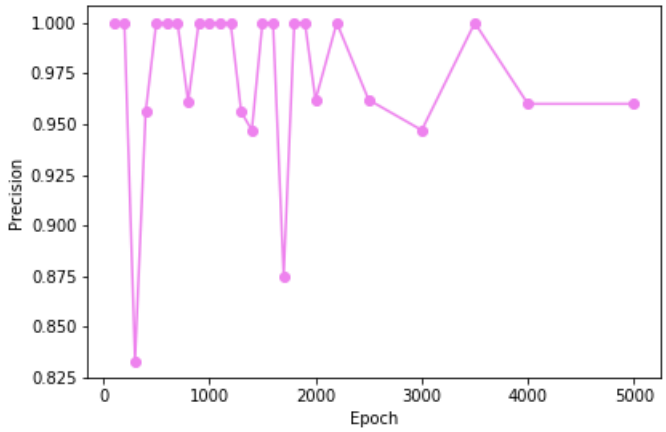

Figure 4: Precision vs Epoch

In the next section we have calculated the training and validation error using cross entropy.Graphs with epoch value ranging from 0 to 200 for 6 different learning rates is shown.We see that initially the the training error and validation error is comparatively high, however this error slowly minimizes on increasing the epoch iterations.This happens because in the initial stage the weights have been assigned randomly,however slowly in the next iterations the model adjusts the weight according to the features thus both the training and validation error decreases over time.
The Learning rate is varied from 0.001 to 3 to show how the nature of the plot changes as the learning

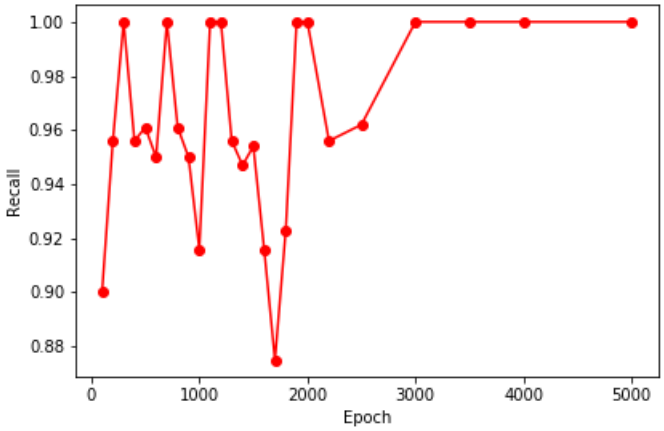

Figure 5: Recall vs Epoch

rate varies. WeUsing the graph plot we can determine the best learning rate for the model,in this case which is 0.01.

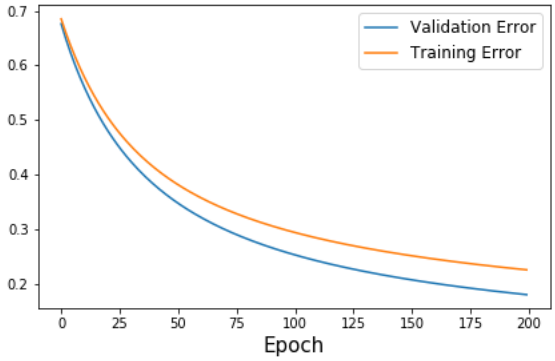

Figure 6: Training and Validation Error vs Epoch(Learning rate 0.001)

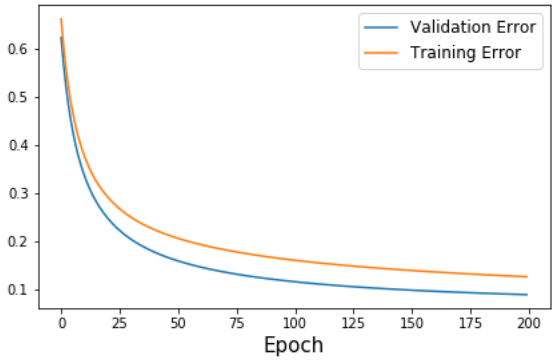

Figure 7: Training and Validation Error vs Epoch(Learning rate 0.005)

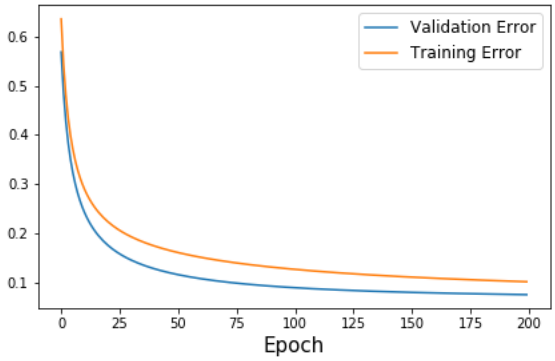

Figure 8: Training and Validation Error vs Epoch(Learning rate 0.01)

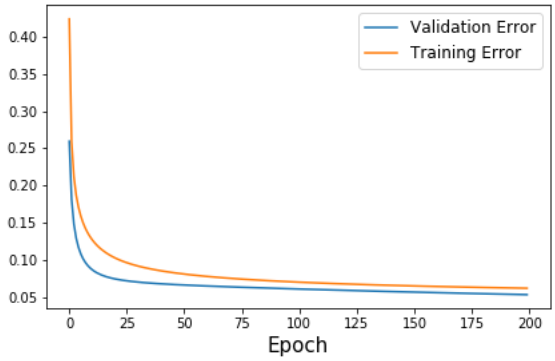

Figure 9: Training and Validation Error vs Epoch(Learning rate 0.1)

5 Conclusion

The experiment above demonstrated a detailed analysis of Logistic Regression on the WDBC dataset. A classification was needed to predict whether the patient would have Malignant or Benign status.Results show how the weights are tuned or aligned to develop a model which would later help in predicting the target variable.We also see how the training and validation loss reduces as the parameters approach their optimal values. Parameters were updated using gradient descent method and cross entropy error was used as the cost function. In best cases, the model gives an accuracy of 100%.

In this project, each feature was normalized so that no feature dominates the other, however in future we may employ techniques to ensure the importance of each feature. This will help us to get rid off any redundant feature and help in dimensionality reduction. Additionally, we can improve the performance of the model by employing pocket algorithm so as to ensure that we don't fit any outliers or noise while training the model.

6 References

1. Breast Cancer Wisconsin (Diagnostic) Data Set
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
2. Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop

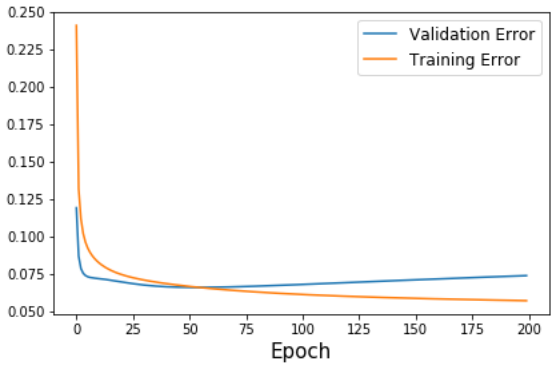

Figure 10: Training and Validation Error vs Epoch(Learning rate 1)

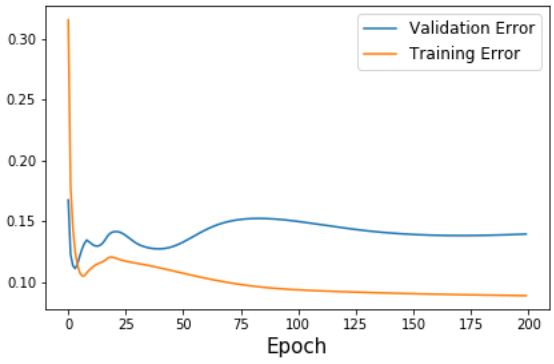

Figure 11: Training and Validation Error vs Epoch(Learning rate 3)