Logistic Regression and Neural Network Implementation

Akash Sharma
as475@buffalo.edu
Department of Computer Science
University at Buffalo
Buffalo, NY 14214

10 October 2021

Abstract

Diabetes is very common disease nowadays. It is a additional inventor to other disease. To diagnose it, patient needs to visit the hospital. This Project aim to predict whether the female patient have the diabetes by observing the other features like blood sugar, pregnancies etc. This project will predict diabetes of the female patient with the help of implementation of logistic Regression and neural network.

1 Implementation of Logistic regression

1.1 What is Logistic Regression?

Logistic Regression is method which is predict data by analysing prior available data. It is the algorithm that classify the new data based on historical data. On the contrary to the name it is a classification model rather than regression model.

2 Logistic Regression Overview

2.1 Formulas for Logistic Regression

$$Z = W^T * Dia\_x\_train + B$$

$$A = 1/(1 + e^{-Z}))$$

$$Cost(w, b) = \sum_{n=1}^{m}$$

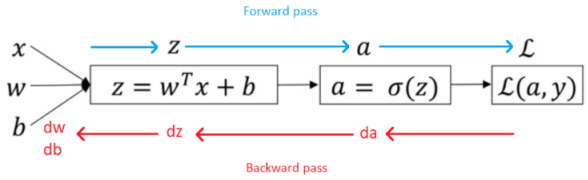

Figure 1: Logistic regression input to loss path

$$y_n log(predicted_n) + (1 - y_n) log(1 - predicted_n) \frac{-1}{m}$$

$$dZ = A - Y$$

$$= X * dZ^T \frac{1}{m}$$

$$W = W - (learningrate) * dW$$

$$B = W - (learningrate) * dB$$

3 Data Set and its features

Pima Indians Diabetes Database dataset will be used for training, and testing. The dataset contains medical data of female patients above the age of 21 and 768 instances with the diagnostic measurements of 8 features. The 8 features are as follows:
1 Glucose (Blood Glucose level)
2 Pregnancies (The number of pregnancies the patient has had)
3 Blood Pressure(mm Hg)
4 Skin Thickness(Triceps skin fold thickness (mm))
5 Insulin level
6 BMI (Body Mass Index : weight in kg/(height in m)2)
7 Diabetes Pedigree Function
8 Age (In years)

4 Preprocessing of Data

Data should be refined or transformed before feeding to the algorithm. So It is necessary to convert the raw data into clean and transformed according to the requirement of algorithm.
The data set contain 8 features in column and 768 rows containing values of each female patient. Below steps are used to preprocessing of data.
1. Data is divided in to features and outcomes as x and y respectively
2. Row containing column name dropped.
3. Standard scaler function is used to scale or normalise the data.
4. Data is divided into train, test and validation, 60%, 20% and 20% respectively.
5. Data sets shape changed, so that it can feed to the algorithm.
6. Sigmoid and Costfunction defined to calculate the cost of data.
7. Iteration set to 90000 and learning rate set .00015.

4.1 Implementation of Logistic Regression

Logistic regression uses equation same as linear regression but give output in 0 or 1. Input features x combined using weights to predicts the outcome value y.

It uses sigmoid function to get the output in 0 or 1.

$$sigmoidfunction = 1/(1 + e^{-Z}))$$

where is base to natural log and z is linear expression.

$$Z = W^T * Dia_xtrain + b$$

Each column in input data have coefficient and should be calculated from the training data.

4.1.1 Gradient Descent

Gradient Descent function is used to minimize the error of the function.It is the way to optimize the algorithm so that the output should be more accurate.

$$Cost(w, b) = (\sum_{n=1}^{m}(y_n log(predicted_n) + (1 - y_n)log(1 - predicted_n))\frac{-1}{m}$$

4.1.2 Estimating the cost

Gradient descent use to estimate the cost of the function. Cost is nothing but the error observed between actual data and predicted data.

There is 2 parameters to get cost.
1. Iteration- Number of times to run training the data.
2. Learning rate- to limit the coefficient of function.

Loop is required to train the data. In the project 90000 iteration is used and .00015 learning rate to get cost. Each time coefficients will be updated based on the error of the model.

4.2 Calculating the accuracy of data

To obtain the accuracy data should be calculated as training data, test data and validation data. Training data is used to train the algorithm, test data is used for testing the algorithm and validation data is used to validate the algorithm.

All training, testing and validation data should be iterated to get the predict the the accuracy of the algorithm.

5 Result for Logistic Regression

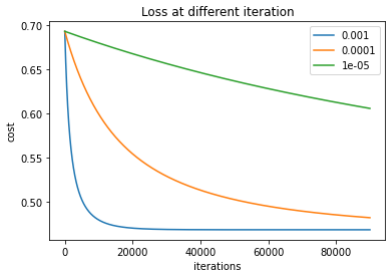

Figure 2: Loss at different iteration

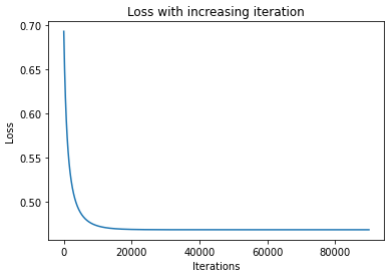

Figure 3: Loss at 90k iteration with .001 learning rate

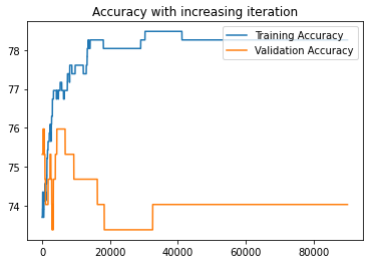

Figure 4: Accuracy with increasing iteration

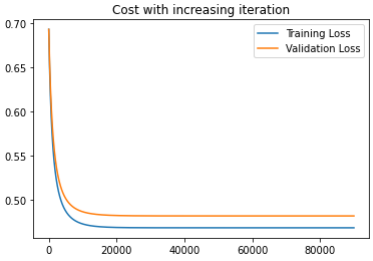

Figure 5: Cost with increasing iteration

6 Part 1-Accuracy

Accuracy given below for train, test and validation when iteration 90000 and learning rate is .00015.

S.No	Data	Accuracy
1.	Training Data	78.61%
2.	Testing Data	81.82%
3.	Validation Data	75.32%

Table 1: Accuracy with different sets of data

7 Implementation of Neural Network

7.1 What is Neural Network?

Neural Network is a algorithm which is designed as the human brain works. It consists of neurons which is a mathematical function that collects and classify the data to a specific architecture. It can help to predict the data in various field like in hospital to predict various disease, in stock market etc.

8 Hidden Layers

A hidden layer presents in neural network found between input layers and output layers, in which neurons take a set of weighted inputs and produce an output through an activation function.

9 L1 and L2 Regularization

Regularization is a technique which is use to resolve the problem of overfitting.

What is Overfitting?
Overfitting is phenomenon which occurs when machine learns the details of training data basically it memorize the input and output which negatively impact the performance of the new data.

L1 Regularization
L1 regularization basically add the penalty to error function. Penalty is the sum of absolute values of weights.

$$\sum_{n=1}^{m} y_n log(predicted_n) + (1 - y_n) log(1 - predicted_n) \frac{-1}{m} + \lambda \sum_{n=1}^{m} |w|$$

where w is weights and lamda is tunnning parameter.

L2 Regularization
L2 regularization also add the penalty to the error function but penalty is sum of square of absolute values of weights.

$$\sum_{n=1}^{m} y_n log(predicted_n) + (1 - y_n) log(1 - predicted_n) \frac{-1}{m} + \lambda \sum_{n=1}^{m} |w|^2$$

10 Dropout

Dropout is also one of the remarkably effective method to reduce the overfitting problem in machine. As the name suggest in dropout, model randomly drop the neurons to improve the generalization so that model doesn't learn the details as that extend which negatively impact the accuracy of the model.

11 Part 2-Result

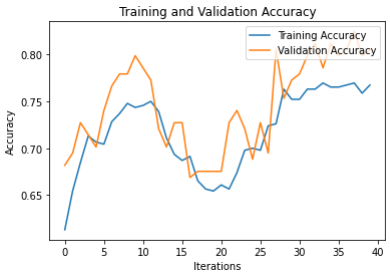

Figure 6: Training vs Validation Accuracy

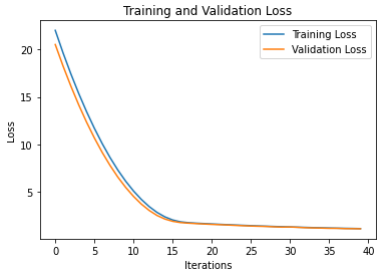

Figure 7: Training vs Validation Loss

11.1 Accuracy without L1 and L2 Regularization

S.No	Data	Accuracy
1.	Training Data	78.70%
2.	Testing Data	81.82%
3.	Validation Data	72.32%

Table 2: Accuracy without L1 and L2 Regularization

11.2 Accuracy with L1 and L2 Regularization

After applying the L1 and L2 regularization in which L1=.01 for 1st hidden layer
L2=.01 for 2nd hidden layer
L1=.01 and L2 =.001 for 3rd layer

S.No	Data	Accuracy
1.	Training Data	77.60%
2.	Testing Data	82.82%
3.	Validation Data	76.00%

Table 3: Accuracy with L1 and L2 Regularization

12 Part 3-Comparison between L1, L2 regularization and Dropout

12.1 Accuracy without Regularization

L1 and L2 both regularization use to resolve the problem of overfitting, the main difference in both regularization is that L1 tries to find median while tries to find the mean of data. L1 is more expensive in computation with respect to L2 regularization. L2 is more accurate than L1 regularization.

S.No	Data	Accuracy
1.	Training Data	78.70%
2.	Testing Data	81.82%
3.	Validation Data	72.32%

Table 4: Accuracy without L1 and L2 Regularization

12.2 Accuracy with L1 and L2 Regularization

After applying the L1 and L2 regularization in which L1=.01 for 1st hidden layer
L2=.01 for 2nd hidden layer
L1=.01 and L2 =.001 for 3rd layer

S.No	Data	Accuracy
1.	Training Data	77.60%
2.	Testing Data	82.82%
3.	Validation Data	76.00%

Table 5: Accuracy with L1 and L2 Regularization

In Above table validation data accuracy increases which is nearly equal to training data accuracy which means overfitting problem decreases.

12.3 Accuracy with L1 and L2 Regularization with Dropout

S.No	Data	Accuracy
1.	Training Data	77.17%
2.	Testing Data	80.52%
3.	Validation Data	75.50%

Table 6: Accuracy with L1 and L2 Regularization and Dropout=0.5

In Above accuracy of testing decreases, this is because of due to drop overfitting problem resolve and all accuracy is somewhat nearer to eachother which show model is predicting accurately.

12.4 Accuracy with only Dropout

S.No	Data	Accuracy
1.	Training Data	79.78%
2.	Testing Data	79.22%
3.	Validation Data	75.30%

Table 7: Accuracy with Dropout=0.5

In Above accuracy table, accuracy of all data become very similar with respect to only L1 and L1 regularization which we see in Table 5. So its clearly visible that with dropout accuracy of model increases with respect to L1 and L2 Regularization.

13 References

1. https://en.wikipedia.org/wiki/Logistic_regression
2. https://www.coursera.org/learn/neural-networks-deep-learnin/home/welcome
3. https://careerfoundry.com/en/blog/data-analytics/what-is-logistic-regression/#what-is-logistic-regression
4. https://en.wikipedia.org/wiki/Neural_/network
5. https://www.investopedia.com/terms/neuralnetwork
6. https://afteracademy.com/blog/what-is-regularization-in-machine-learning
7. https://medium.com/analytics-vidhya/l1-vs-l2-regularization-which-is-better-d01068e665