Machine Learning- Project 1 Report

2-Class classification using Machine Learning

Gulfam Hussain
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
gulfamhu@buffalo.edu

Abstract

This project report consists of a two-class problem classification using Machine Learning. The dataset contains the features of fine needle aspirate (FNA) of a breast mass. This report presents the various aspects of FNA cells classification into 2 different class called as Benign (class 0) or Malignant (class 1) using the logistic regression. The dataset taken to perform this classification is Wisconsin Diagnostics Breast Cancer (wdbc.dataset). Using the logistic regression approach, we will be performing the logistic regression operations on the data divided into the training data. Upon, various iterative of learning rates and epochs, the weight factor will be determined from the training data and will be used to test the testing data in order to classify the problem as significant as it could be.

Introduction:

Machine learning is one of the fields which deals with training the system to produce the outcome and let the machine comprehends itself for the final outcome for such problems in future. There are various aspects of training our systems/machines using machine learning approaches which help us identity the outcome probability based on the certain factors which affects the outcome without much intervention of human involvement. These machine algorithms are categorized into different category and serves the best purpose based on the available datasets and mainly classified into supervised and unsupervised machine learning approaches. In the supervised machine learning algorithm, the input data and the output data are provided to the systems in order to provide a learning basis of future data processing. The systems get trained on the training dataset and thus, capable of using generative data model which could be helpful in identifying the hidden details from the available data set.

The unsupervised learning deals with the algorithm which trains itself using only the given data without help of any output dataset. In this project, we are using an approach called Logistics regression which falls under supervised learning category. The logistic regression after working on the training and validation data sets, provides the required classification of a problem.

2-Class classification using Machine Learning

Dataset:

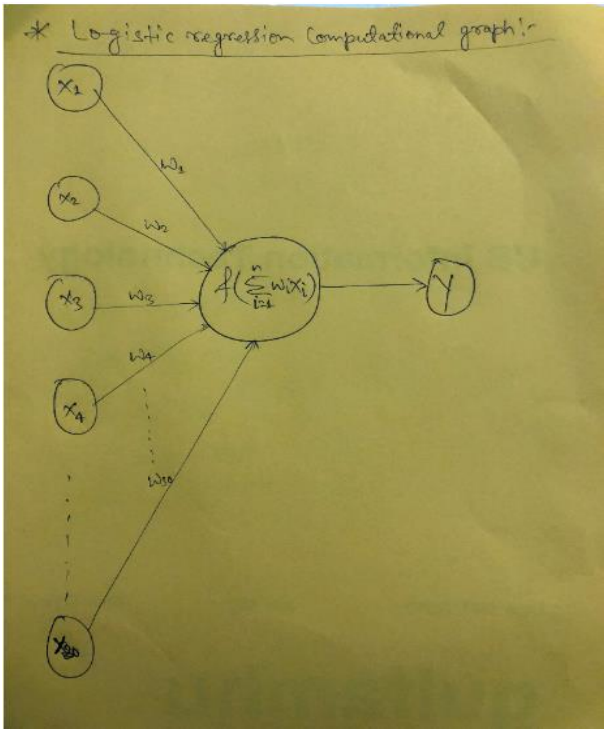
This project and its validation report are performed on one of the datasets called Wisconsin Diagnostics Breast cancer (wdbc) dataset. This dataset consists of 569 instances with 32 attributes (ID, diagnosis, B/M), 30 real value input features. These features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The computed features include different characteristics of the cell nuclei present in the image. There are total 30 features in the dataset computed for each image.

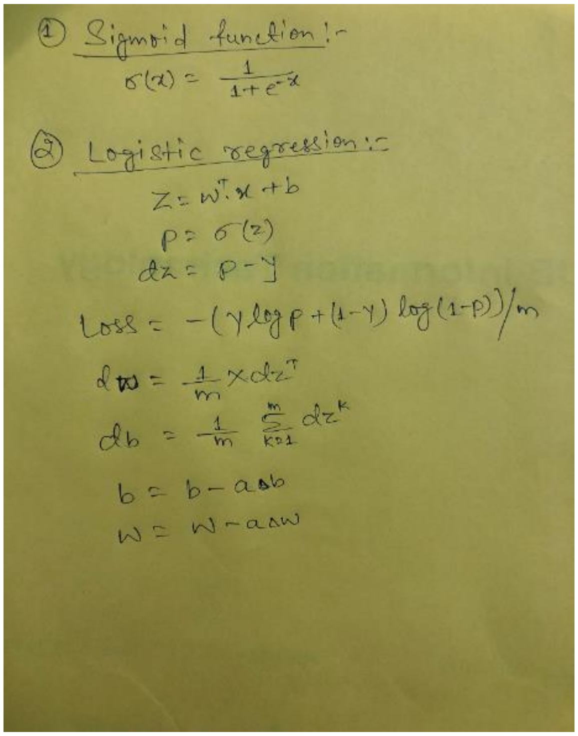
Preprocessing:

There are various preprocessing steps involved before performing the logistic regression on the given dataset. The steps are as below:

1) Read the data file through Python library
2) Process the data file by dropping the column id from the given dataset file and map label column to 0 & 1.
3) Perform partition on the data frame and split them into training, validation and testing data set.
4) Normalize the dataset to keep all the values within the same range in order to perform logistic regression.
5) Initialize the weights and biases and learning rate.
6) Perform logistic regression over different range of epochs on training and validation data set.

Architecture:


* Logistic regression Computational graph:-
x_1
x_2
x_3
x_4
x_{30}
w_1
w_2
w_3
w_4
w_{30}
$f(\sum_{i=1}^{n} w_i x_i)$
Y



① Sigmoid function:-

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

② Logistic regression:-

$$Z = w^T \cdot x + b$$

$$p = \sigma(z)$$

$$dz = p - y$$

$$Loss = -(y \log p + (1-y)\log(1-p))/m$$

$$dw = \frac{1}{m} x\, dz^T$$

$$db = \frac{1}{m}\sum_{k=1}^{m} dz^k$$

$$b = b - a\Delta b$$

$$w = w - a\Delta w$$

Machine Learning- Project 1 Report

2-Class classification using Machine Learning

<u>Results</u>:

After training the train data and validation data over multiple combinations of learning rate and epochs, I chose the below values which gave me the 92% accuracy.

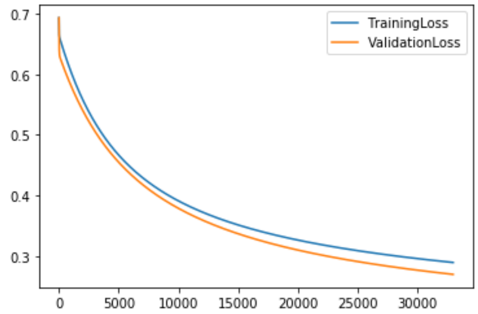
Learningrate= 0.13
Epoch = 15000

<u>Loss graphs based on epochs:</u>

The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters.

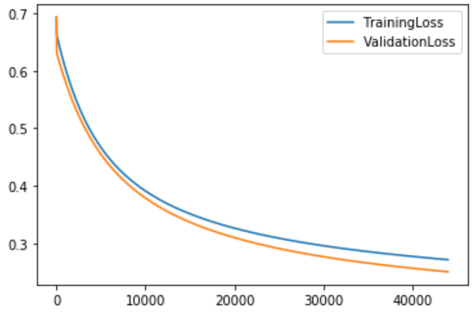
learningrate = 0.13
epoch = 8000

```
Out[16]: <matplotlib.legend.Legend at 0x1d3c5e0a400>
```



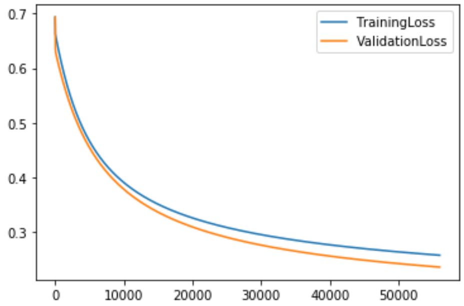
learningrate = 0.13
epoch = 11000

```
Out[18]: <matplotlib.legend.Legend at 0x1d3c5efdb00>
```



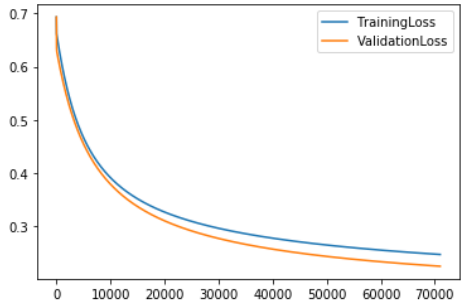
learningrate = 0.13
epoch = 12000

```
Out[20]: <matplotlib.legend.Legend at 0x1d3c6fce400>
```



learningrate= 0.13
epoch = 15000

```
Out[22]: <matplotlib.legend.Legend at 0x1d3c70f0240>
```

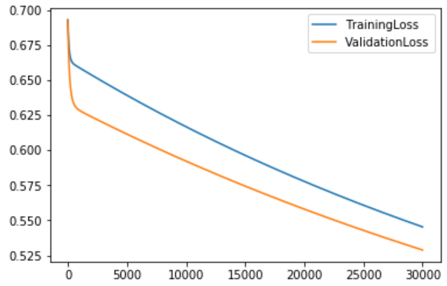


__Loss graphs based on learning rate:__

The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Choosing the learning rate is challenging as a value too small may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process.

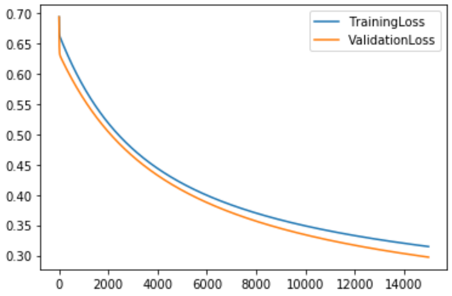
learningrate = 0.01
epoch = 15000

```
Out[16]: <matplotlib.legend.Legend at 0x22f561f0278>
```



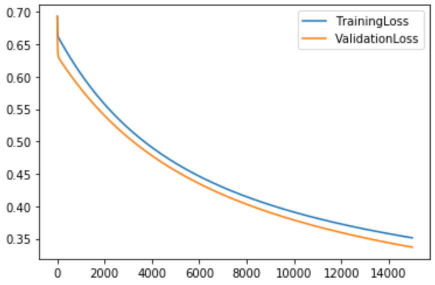
learningrate= 0.2
epoch = 15000

```
Out[20]: <matplotlib.legend.Legend at 0x22f560f33c8>
```



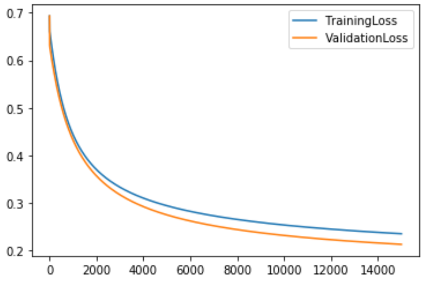
learningrate = 0.13
epoch = 15000

```
Out[23]: <matplotlib.legend.Legend at 0x22f57286278>
```



learningrate= 0.8
epoch = 15000

```
Out[26]: <matplotlib.legend.Legend at 0x22f572fab70>
```



Conclusion:

This particular project based on logistic regression provided an opportunity to learn more on machine learning approaches and their implementation in real world problems. The model was successfully built for the given dataset that has an accuracy of 92% after training the data over 15000 epochs with learning rate as 0.13.

References:

[1] supervised learning:
https://searchenterpriseai.techtarget.com/definition/supervised-learning
[2] Machine learning:
https://en.wikipedia.org/wiki/Machine_learning#Approaches
[3] Understand the Impact of Learning Rate on Neural Network Performance
https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/
[4] Class slides and project description