Maya Hussein

CSCE-4604

September 2022

Lab Assignment 1

1 Introduction

This lab assignment tackles the understanding and application of linear & logistic regression. Below are the detailed description of each regression type.

2 Part I: Linear Regression

In the first lab, the question asked if we could predict the sale price over a set of house features. Expectantly, linear regression would be the most fit model for this problem, as it tackles the behavior of the output over a given set of features.

The model was coded through a single dense layer with one node.

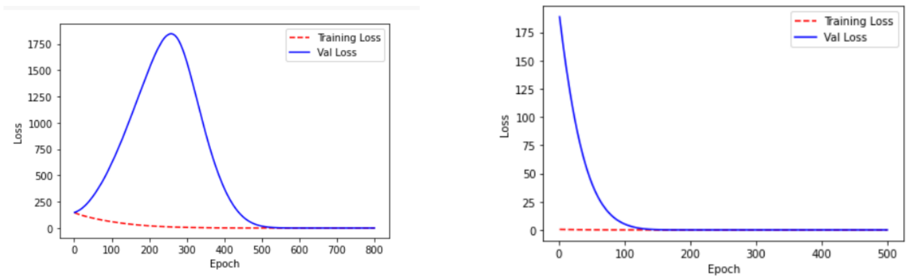
Use this note to define a **sequential model of one dense layer with one unit using Tensorflow.Keras**.

```
[56] #regularizers = l2(0.01)

     model = Sequential()
     model.add(Dense(1, kernel_regularizer =l2(0.01)))

     # TODO: Define the Model using Tensorflow.Keras
```

Also, note that regularization was added to reduce the over-fitting on the linear model. In particular, Ridge Regression was utilized as multi-collinearity occurs and the variance is large; hence applying L2 has lower error that L1.



(a) With Specified Rate (b) Unspecified Rate

Figure 1: Linear Models with Regularization

Above are two linear models with L2 regularization.

In the first graph, the learning rate was specified, forcing the data to enhance at a certain rate. However, the rate of the second model was dynamic. Notice how the loss in the second graph declines at a faster pace than the specified rate. The model quickly converges to zero because the rate is being adjusted as the number of Epochs increase. Moreover, the first model has great loss from 200-300 Epochs, where at the same number of Epochs, the second graph is already converging to zero.

**Note that in graph (a), the number of Epochs is increased to show eventual convergence of the model.

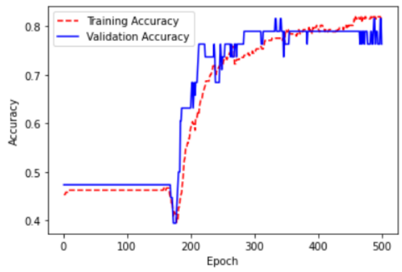
Hence, even though both models eventually converge to a loss equivalent to zero, the second model's rate is much faster; eventually achieving the same result faster.

3 Part II-Logistic Regression

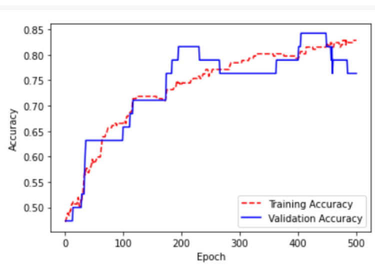
The second assigned lab consisted of a scientific classification of the likelihood of being diagnosed with a heart disease. Since this is binary classification; its only best to use logistic regression.

3.1 Scaling Techniques

Mean & Standardized Scaling - Min & Max



(a) Min & Max Rate



(b) Standardized

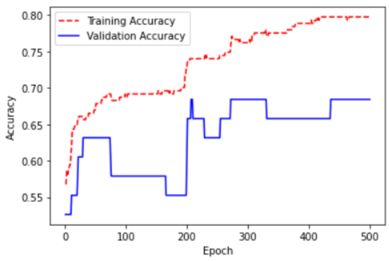
Figure 2: Logistic Models with Scaling Techniques

After using both scalars, standard scaling has a smoother probability rates and better dynamics in terms of area under the curve. Standardized Scaling is also known as feature scaling, which is known to be better for logistic regression since standardizes the independent features present in the data in a fixed range.

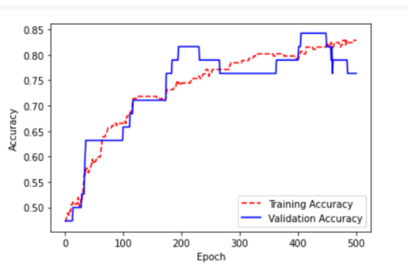
3.2 Artificial Features

X - X^2 - X^3

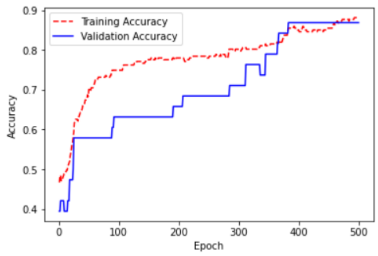
As shown when comparing all three graphs, it is noticed that the graph with no artificial features has an incredibly low validation accuracy. Hence, adding artificial features adds more non-linearity into the graph giving it higher performance. Moreover, it became apparent that after increasing the artificial features more than x^2, the accuracy remains constant (87% to 84%). As a result, there is no need to add more dimensions.



(a) X



(b) X^2



(c) X^3

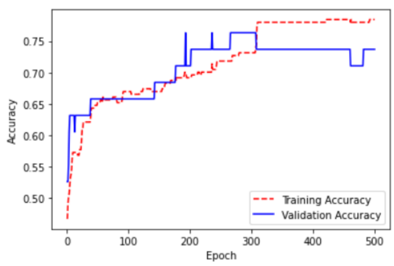
Figure 3: Logistic Models with Artficial Features

3.3 Activation Functions

Sigmoid Function - Softmax Formulation

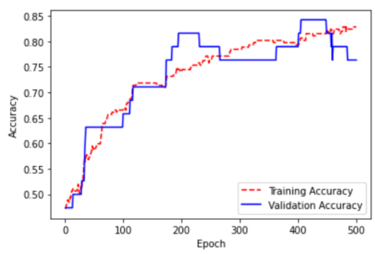
When it come to which activation function is more suitable, the sigmoid function had more pros than the softmax formulation. It is clear from the Accuracy and Loss results that even though their accuracies are quite similar, the sigmoid had a higher percentage. Moreover, the softmax formulation has a significant loss of 3.01 which is about 3 times more than the sigmoid function. In addition, the softmax formulation had potentially much lower validation accuracy.

Also, other activation functions were inefficient before even testing as they do produce outcomes that are beyond the [0,1] interval.




```
Test fraction correct (NN-Loss) = 3.01
Test fraction correct (NN-Accuracy) = 0.76
```

(a) Softmax Formulation




```
Test fraction correct (NN-Loss) = 0.95
Test fraction correct (NN-Accuracy) = 0.87
```

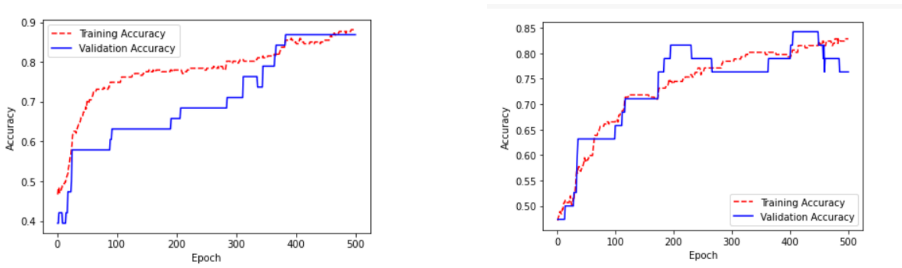
(b) Sigmoid Function

Figure 4: Logistic Models with Different Activations

3.4 Regularization Techniques

L1 - L2

It is apparent that the model associated with L2 regularization has lower losses between the validation and training. Not only that but also the overall accuracy of the Ridge Regularization is about 20% higher from 200 - 400 Epochs.



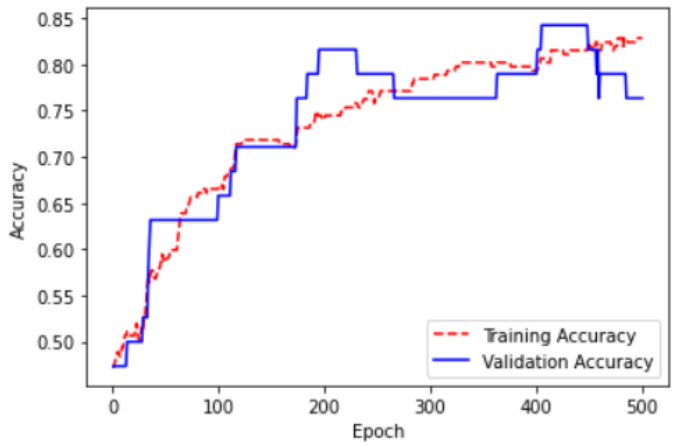
(a) Lasso Regression L1 (b) Ridge Regression - L2

Figure 5: Logistic Models with Different Regularizations

3.5 Loss Functions

Binary Cross Entropy

As for the optimal loss, Binary Cross Entropy was the most fit to the problem as logistic regression losses can either be determined using Binary Cross Entropy or using Neg. Log Likelihood. However, after extensive research on Keras and Tensor Flow, it was identified that Neg. Log Likelihood Loss Function was not available. Also, the Mean Square Error was not a suitable match



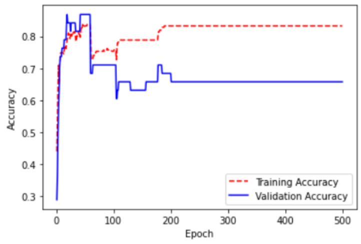
as the loss function is in the form of a cost and not the probability/likelihood of occurrence, which defies the primary goal of the model.

For more info regarding the Neg. Likelihood Loss Function: https://keras.io/api/losses/

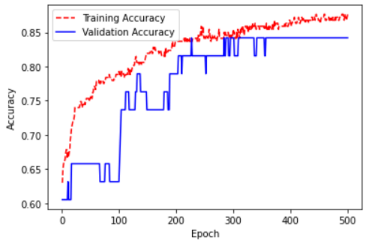
3.6 Optimization Functions

Adam - RMS Propagation - Gradient Descent with Momentum

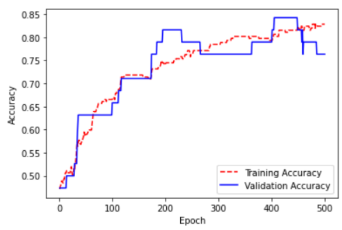
As presented in the graphs, Adam optimizer is the most fit for this model as it the most optimizer that reduces the losses. It also is the most fit when changing the attributes including the weights and the learning rates. Graph (a) gradient descent has the greatest loss over longer Epochs, while RMS seems to be more lenient in optimization.



(a) Gradient Descent with Momentum



(b) RMS Propagation Rate



(c) Adam

Figure 6: Optimization Techniques