Mathematics for AI and Machine Learning Coursework Report

Alexander Wilson; Student No. 1678600

9th December 2020

1 Introduction

In this report the performance of 3 models: Batch Gradient Descent, Stochastic Gradient Descent and Mini-Batch Gradient Descent will be analysed, discussed and evaluated using two evaluation metrics: Root Mean Squared Error (RMSE) and R-Squared. RMSE is a calculation of the standard deviation of the residuals (prediction error) of the model. A low RMSE suggests that the prediction error of the model is low. R-Squared is a calculation which represents what proportion of the variance of a dependent variable is able to be predicted by the independent variables. In this case the R-Squared explains what proportion of the variance of the probability of admission is explained by the independent variables. These two metrics will be used throughout the report for measuring the performance of all 3 models. Throughout the report each of the values presented are the average of 3 separate runs of the model at the respective hyper-parameter values in order for the values to not be impacted by slight variations in model performance which occurs.

2 Batch Gradient Descent

The Batch Gradient Descent model computes the gradient of the cost function, Sum Square error, with respect to the weights of the linear regression model. These calculations are performed over the whole training set, the entire batch. The weights of the linear regression model are then updated after every iteration, every computation of the gradients of the loss function with respect to the weights. The learning rate is a hyper-parameter which enables the model at each iteration to update at a slower or faster rate. A large learning rate will mean a larger 'step' which may be beneficial when attempting to avoid local minima, but may cause the model to diverge and move further away from the solution at each iteration. Similar issues occur if the learning rate is too small, the solution may not converge in an feasible amount of time.

The table below shows the results of the experiment looking at how changes to the number of iterations impacted the model performance of the Batch Gradient Descent model. The table shows that as the number of iterations is increased, both, the RMSE decreases and the R-squared increases. This show that the model performance is increasing. However at 100,000 iterations this performance progression stops and the RMSE stays the same. The R-Squared increases by a negligible amount, suggesting the model has fully converged. Therefore it would be expected that even at 10,000,000 iterations the performance of the model would not increase if the learning rate stayed at 0.1.

Batch Gradient Descent - Iteration Experiment		
No. Iterations	RMSE	R-Squared
10	0.1188	0.2128
100	0.0951	0.4985
1000	0.0682	0.7438
10000	0.0637	0.7757
100000	0.0563	0.8251
1000000	0.0563	0.8254

The table below shows the results of the experiment looking at how changes to the learning rate impacted the model performance of the Batch Gradient Descent model, all learning-rates were run 3 times for 100,000 iterations and then averaged. The results showed that at 100,000 iterations as the learning rate increased, model performance increased. However RMSE stopped falling at a learning rate of 0.1 and the difference in the R-squared value was negligible at a learning rate of 0.1 or higher. It can therefore be suggested that a RMSE of 0.0563 and an R-Squared value of 0.8254 represent the best performance possible for this Batch Gradient Descent model.

Batch Gradient Descent - Learning Rate Experiment		
Learning Rate	RMSE	R-Squared
0.0001	0.0909	0.5449
0.001	0.0713	0.7200
0.01	0.0647	0.7691
0.1	0.0563	0.8251
0.2	0.0563	0.8254
0.4	0.0563	0.8254

3 Stochastic Gradient Descent

The difference between Stochastic Gradient Descent and Batch Gradient Descent is that instead of computing the gradients for the whole 'batch' of training data a random instance of the training set is picked at every step and the gradients on that single instance are computed instead of a whole 'batch'. Due to the addition of randomness which is inherent the algorithm is less regular and instead of the value of the cost function gently decreasing it will jump around and only on average will it decrease. The final value may also not be optimal as the algorithm will continue to jump around until the number of epochs inputted has been reached instead of converging on the optimal value. In order to reduce the probability of the Stochastic Gradient Descent algorithm from leaving the optimum point once it has found its minimum is to gradually reduce the learning rate. Therefore as the number of epochs rises the impact that each update to the update equation has will decrease.

The table below shows the results of the experiment which assesses the impact of different epoch values, each result reported was run 3 times at its respective hyper-parameter level and then averaged. The learning rate was set with the values of t0 and t1 at 5 and 50 respectively. As the number of epochs increased the RMSE decreased and the R-squared value increased. This was shown for every epoch value. It would be expected that if a larger number of epochs was used then the model performance would be improved, however this was limited by the amount of computational power and time available. The values of RMSE and R-squared suggest a worse performance to the Batch Gradient Descent algorithm, however, this should not be considered explicitly true as the hyper-parameters used are looking to show a trend rather than being optimised for the highest model performance.

Stochastic Gradient Descent - Epoch Experiment		
No. Epochs	RMSE	R-Squared
10	0.0827	0.6165
100	0.0752	0.6864
1000	0.0732	0.7039
10000	0.0723	0.7118
100000	0.0698	0.7268

The table below shows the result for the experiment which investigates the impact of different learning rate on Stochastic Gradient Descent. Although the leaning rate is determined by the learning schedule, the initial learning rate can be modified using different values of t0 and t1. Each result was run 3 times at 10,000 epochs and then averaged. The results showed that; the RMSE decreased until the learning rate of 0.1 was used and then increased at 0.25, and the R-Squared was increasing until a learning rate of 0.1 was used and then fell at a learniong rate of 0.25. This suggests that the optimum starting learning rate is between 0.01 and 0.25, as the optimum rate could have been met any time between these values. When considering the optimum hyper-parameters for this model of Stochastic Gradient Descent a higher epoch value than was tested and an initial learning rate around the value of 0.1 is likely to provide optimum performance.

Stochastic Gradient Descent - Learning Rate Experiment		
Learning Rate	RMSE	R-Squared
0.0001	0.0854	0.5003
0.001	0.0753	0.6870
0.01	0.0705	0.7238
0.1	0.0688	0.7387
0.25	0.0778	0.6578

4 Mini-Batch Gradient Descent

Mini-Batch Gradient Descent differs from both Batch Gradient Descent and Stochastic Gradient Descent as instead of computing the gradients of the whole 'batch' as in Batch Gradient Descent or based of one random instance of the training set in Stochastic Gradient Descent, Mini-Batch Gradient Descent computes the gradients on small random sets of instances, mini-batches. It therefore does not have to compute the gradients of all of the instances of the training set, but is still less erratic than Stochastic Gradient Descent. Mini-batch Gradient Descent does however have another hyper-parameter to tune, the mini-batch size, in addition to the number of epochs and learning rate.

The table below gives the results to the experiment of the impact of different epoch values on model performance. The results were run 3 times at each value with a learning rate of 0.1 and a mini-batch size of 25 and then averaged. As the number of epochs was increased the RMSE decreased and the R-Squared increased until the value of 10,000 epochs was reached. After this point at 100,000 epochs the RMSE and R-Squared values stayed the same at 0.0730 and 0.7057 respectively. This suggests that this is the best performance possible given the other hyper-parameter values.

Mini-Batch Gradient Descent - Epoch Experiment		
No. Epochs	RMSE	R-Squared
10	0.1061	0.3853
100	0.0927	0.5244
1000	0.0754	0.6864
10000	0.0730	0.7057
100000	0.0730	0.7057

The table below gives the results to the experiment of the impact of different learning rates on model performance. The results were run 3 times at each value with a epoch value of 10,000 and a mini-batch size of 25 and then averaged. The results showed that as the learning rate was increased the model performance increased with the lowest values of RMSE and highest value of R-Squared both occurring at a learning rate of 0.2.

Mini-Batch Gradient Descent - Learning Rate Experiment		
Learning Rate	RMSE	R-Squared
0.0001	0.1101	0.3284
0.001	0.0967	0.4803
0.01	0.0872	0.5809
0.1	0.0730	0.7057
0.2	0.0729	0.7076

The table below gives the results to the experiment of the impact of different mini-batch sizes on model performance. The results were run 3 times at each value with a epoch value of 10,000 and a learning rate of 0.1 and then averaged. The results contained two peaks; one at a batch size of 25 with a RMSE of 0.0730 and an R-Squared of 0.7057 and another at a batch-size of 200, with a RMSE of 0.0587 and an R-Squared of 0.8127. The later achieving a significantly better performance.

Mini-Batch Gradient Descent - Batch Size Experiment		
Batch Size	RMSE	R-Squared
10	0.0812	0.5449
25	0.0730	0.7057
50	0.0806	0.6414
100	0.0808	0.6395
200	0.0587	0.8127
400	0.0606	0.7987

The optimum hyper-parameters are likely to produce a better performance than that of the highest which was reported for Mini-Batch Gradient Descent, however, the reported RMSE and R-Squared at 0.0587 and 0.8127 suggest a strong model which is able to predict the probability of success in a graduate application to university well.