COMP24112 Lab Report

Licheng Chen

May 6, 2024

1 Linear Classification via Gradient Descent

1.1 Derivation of the training objection function

$$O = C\sum_{i=1}^{N} \max\left(0, 1 - y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right)\right) + \frac{1}{2}\mathbf{w}^T\mathbf{w}.$$

$$O = \sum_{i=1}^{N} C\max\left(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i + y_i w_0\right) + \frac{1}{2}\mathbf{w}^T\mathbf{w}.$$

$$O = \begin{cases} \sum_{i=1}^{N} C(1 - y_i\mathbf{w}^T\mathbf{x}_i + y_i w_0) + \frac{1}{2}\mathbf{w}^T\mathbf{w} & \text{if } \mathbf{w} < 0 \\ \frac{1}{2}\mathbf{w}^T\mathbf{w} & \text{otherwise} \end{cases}$$

$$\frac{dO}{d\mathbf{w}} = \begin{cases} \sum_{i=1}^{N}(-Cy_i\mathbf{x}_i) + \mathbf{w} & \text{if } \mathbf{w} < 0 \\ \mathbf{w} & \text{otherwise} \end{cases}$$

1.2 Model Training and Testing

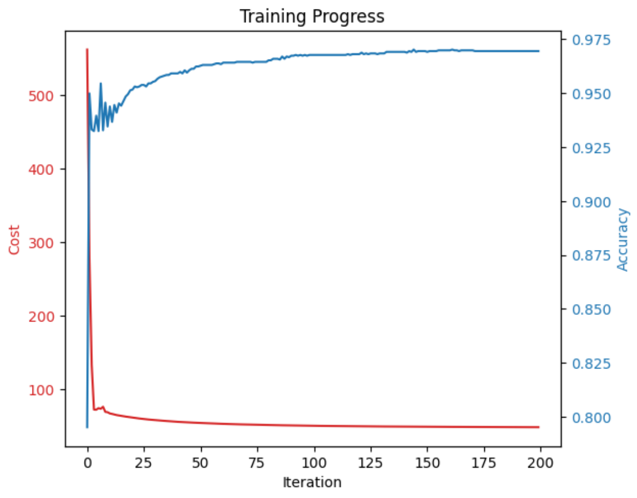
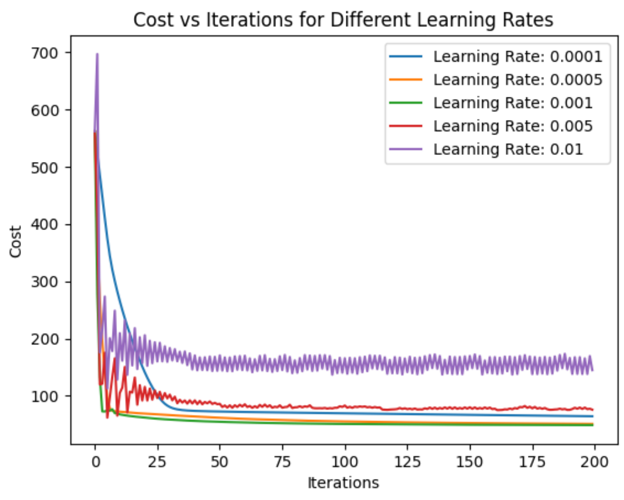


Figure 1: Graph of the cost and accuracy of training over iterations

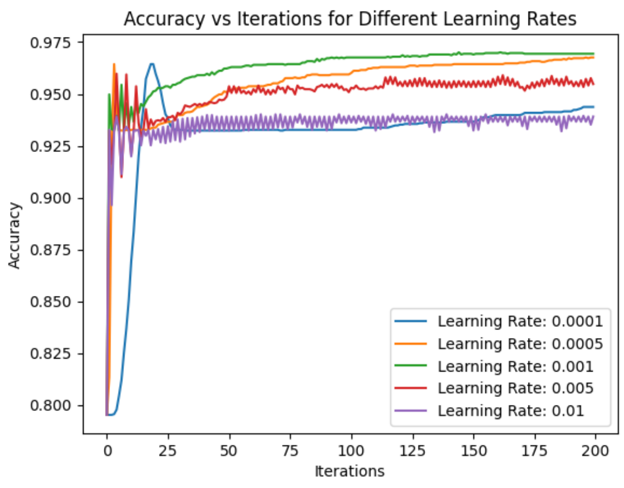
The model produced by the run shown above had an accuracy of 0.9657 and an F1 score of 0.7213 when run on the test data. While initially very high, the cost of each iteration reduced dramatically, quickly falling below 100 before stabilising after 10 iterations. In the first 25 iterations, the accuracy was rather spiky, suggesting that the model's parameters are oscillating around the optimal values before stabilising in the later iterations. This is probably caused by the fact that I chose to initialise my weights as zeros instead of randomising them, as this behaviour was not observed when I used randomised weights.

1.3 Learning Rate Analysis

The above graphs suggest that the most optimal learning rate seems to be 0.001. That value had the highest consistent accuracy when training, while 0.0001 initially had a higher accuracy, it dropped down to below that of 0.001 before it stabilised. The rates above 0.001 never stabilised at all. The costs are also higher the larger the learning rate is, except for 0.0001, which is higher than both 0.0005 and 0.001.



(a) Costs per iteration for various learning rates.



(b) Accuracy per iteration for various learning rates.

Figure 2: Comparison of costs and accuracy per iteration for various learning rates.

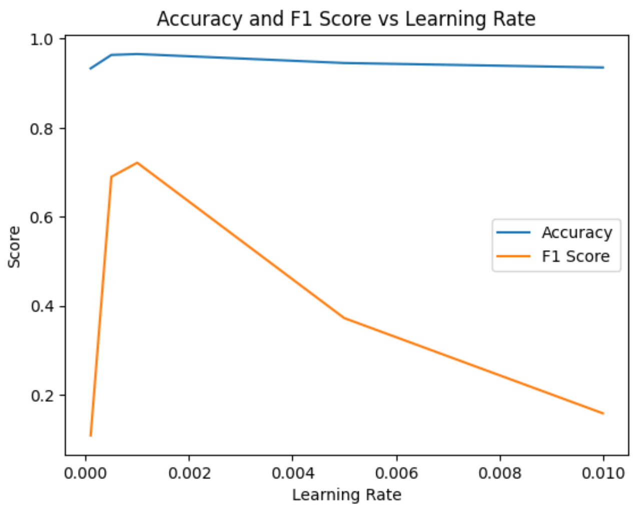


Figure 3: Graph of the accuracy and F1 score of of various learning rates.

The accuracy is consistently high for all learning rates, so they are all good at predicting true positives and negatives. However, accuracy cannot be solely relied on because it only deals with those values. We must also look at the F1 score, which gives us a more comprehensive view of the false positives and negatives. This graph supports the fact that the best learning rates are between 0.0005 and 0.001, as the F1 scores of the other learning rates are lower than those two.

2 Air Quality Analysis by Neural Network

2.1 Model Selection

The results reported that ReLU is the best activation function and the best hidden layer size is (100,100) with a cross-validation MSE of 0.1609. The standard deviation of cross-validation MSE is 0.1617, which suggests some variability in model performance across different folds of the cross-validation. The MSE on the test set is 0.2064, which is slightly higher than the cross-validation MSE. However, an R^2 score of 0.9071 suggests that the model explains approximately 90.71% of the variance in the CO(GT) values on the test set. This indicates that the model generalizes well to unseen data.

2.2 Training Algorithm Comparison: SGD and ADAM

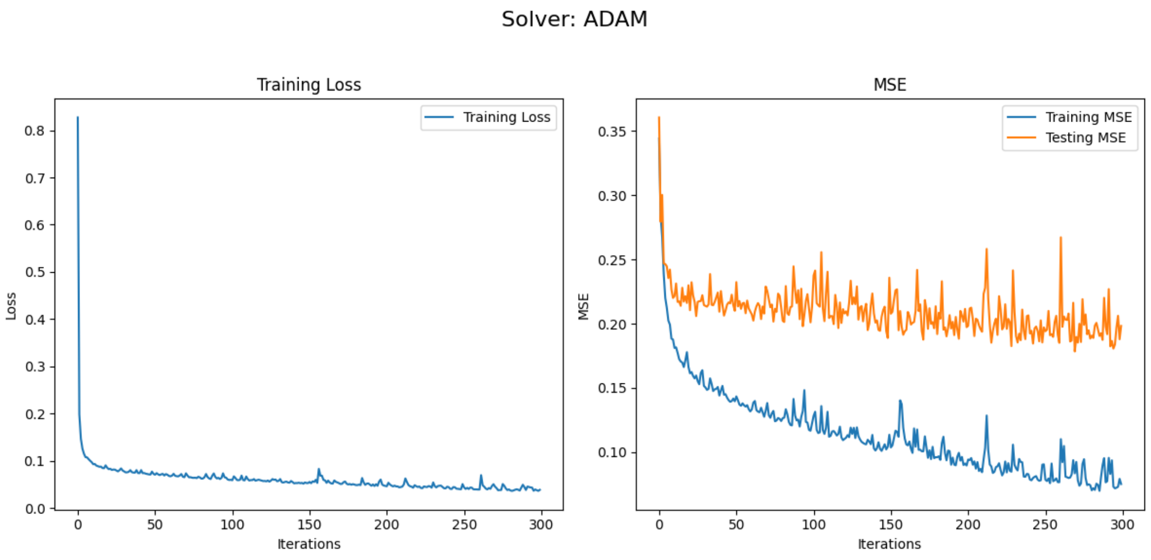


Figure 4: Loss and MSE of ADAM

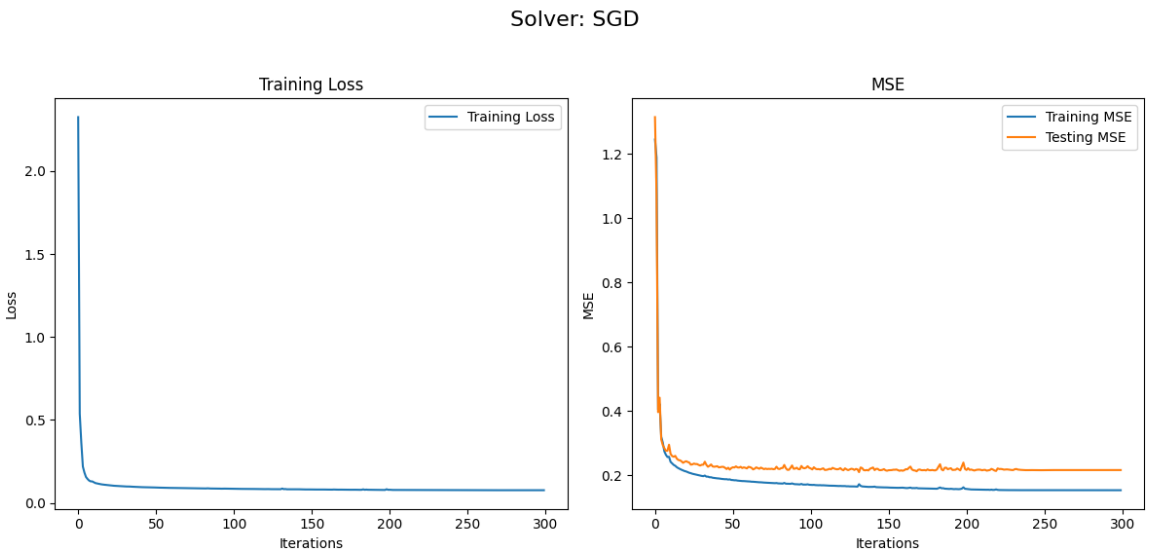


Figure 5: Loss and MSE of SGD

With a test MSE of 0.1983 and an R^2 score of 0.9108, ADAM appears to perform better than SGD. For SGD, the test MSE and R^2 score are 0.2157 and 0.9029, respectively, both of which are worse than those obtained with ADAM. However, the difference in performance seems marginal, as the gap between the two algorithms is only around 0.02 for MSE and 0.01 for the R^2 score.

3 Building A Robust MLP Regressor

I mostly reused the code from the previous section, but I used a pipeline instead and used different parameters.