COMP 551 P2: Implementing Multi-Class Logistic Regression and Gradient Descent

Zirui He 260766420, Cong Zhu 260826128, Yumika Shiba 260863694. Group 53

1. Abstract

Softmax regression is a generalized form of logistic regression suitable for multi-class classification. In this project, we implemented the softmax regression model from scratch, along with an optimizer using mini-batch gradient descent with momentum. The effect of different hyperparameters of the optimization process on model performance was analyzed and the training and validation accuracy in each iteration were tracked. The model was also compared against another classification algorithm, K-Nearest Neighbour (KNN). We found that KNN had a more stable performance when the two models were trained on a dataset with a larger number of features than the other dataset.

2. Introduction

Logistic regression is one of the most widely used binary models for the analysis of categorical data [1]. An extension of logistic regression is softmax regression, which is a generalized form of logistic regression suitable for multi-class classification [2]. In this project, we built a softmax regression model and an optimizer using mini-batch gradient descent with momentum.

Two publicly available datasets were used, and 5-fold cross validation was implemented and used to estimate the model performance, in terms of accuracy, with varying hyperparameter values. More specifically, the effect of varying values of batch sizes, learning rates, and momentum was studied by analyzing validation and training accuracy. Upon training and validating the model, we compared its performance with another classification algorithm, KNN. With a dataset with more features, KNN had a more stable performance than softmax. For the analysis of the effect of hyperparameter values, we observed different behaviours between the two datasets used, which can be partly due to the small sample size of one of the datasets.

3. Datasets

The digit dataset [3] was obtained from Scikit-Learn and heart-h (hugarian) dataset [4] from OpenML. The digit dataset (will be referred to as dataset 1) contained 1797 instances, 64 features, and 10 classes (a set of 10 integer digits in the range 0-9). A histogram of class label distribution (Figure 0.1) revealed that the labels are almost uniformly distributed. The heart-h dataset (will be referred to as dataset 2) was a dataset containing heart disease diagnosis. It contained 294 instances, 64 features, and 5 classes. A histogram of the class distribution can be found in the appendix (Figure 0.2).
There was no missing data in either of the datasets.

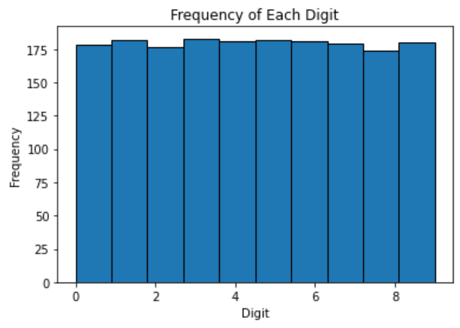


Figure 0.1: Histogram of class labels for the digit dataset

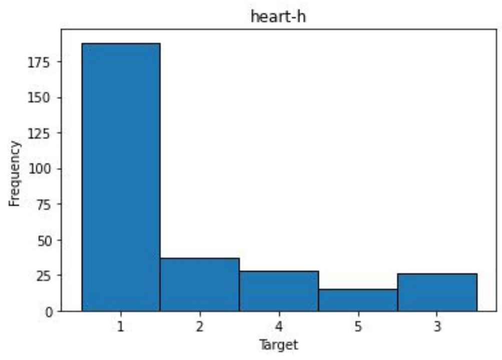


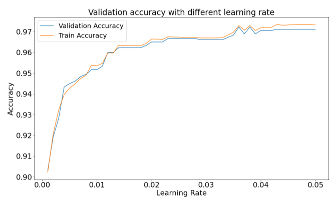
Figure 0.2: Histogram of class labels for the heart disease dataset

4. Results

4.1 Analysis of softmax regression performance with respect to different hyperparameter values

The effect of batch size, learning rate, and momentum, respectively, were studied. When the effect of a hyperparameter was analyzed, the values of other hyperparameters were fixed at their values which maximized validation accuracy. The training and validation curve for different optimization hyperparameters can be found below on the left side of the aligned graphs.

Graphs for digit dataset:



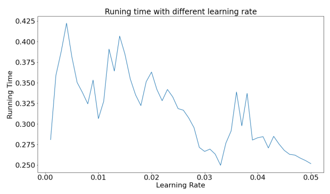
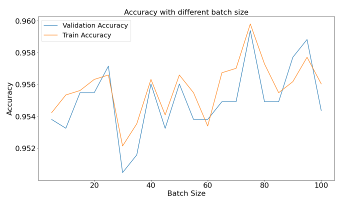


Figure 1.1 (left): learning rate vs validation accuracy
Figure 1.2 (right): learning rate vs running time



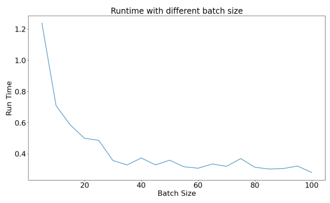
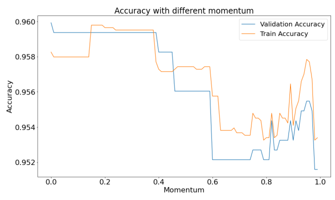


Figure 2.1 (left): batch size vs validation accuracy
Figure 2.2 (right): batch size vs running time



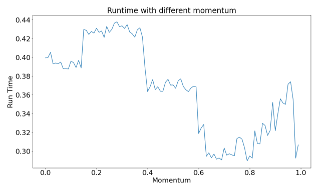
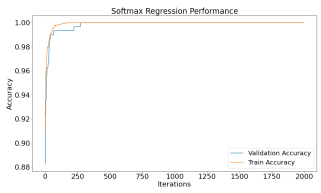


Figure 3.1 (left): momentum vs validation accuracy
Figure 3.2 (right): momentum vs running time

4.3 Termination Conditions

The optimization was terminated if the validation error has not been decreasing in the past 10 iterations. The graph below shows the training and validation accuracy for 2000 iterations (without the termination condition).



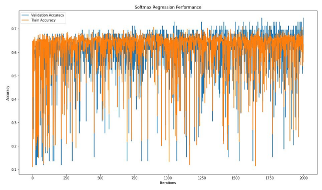
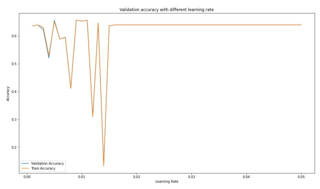


Figure 4.1 (left): iterations vs accuracy (dataset 1)
Figure 4.2 (right): iterations vs accuracy (dataset 2)

Graphs for heart-h dataset:



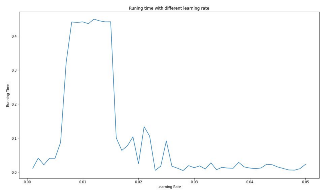
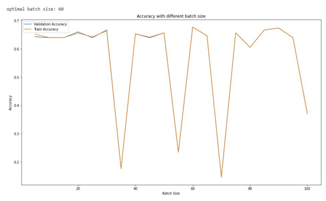


Figure 5.1 (left): learning rate vs validation accuracy
Figure 5.2 (right): learning rate vs running time



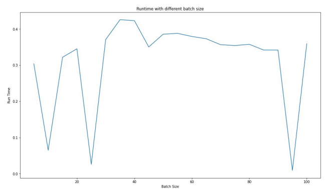
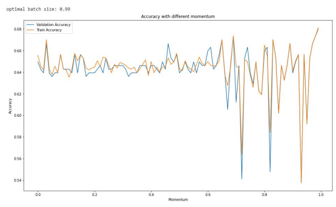


Figure 6.1 (left): batch size vs validation accuracy
Figure 6.2 (right): batch size vs running time



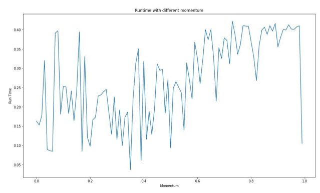
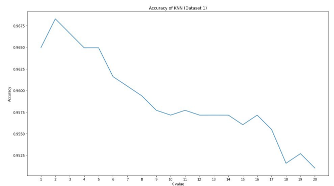


Figure 7.1 (left): momentum vs validation accuracy
Figure 7.2 (right): momentum vs running time

4.4 Comparison of the accuracy of KNN and softmax regression



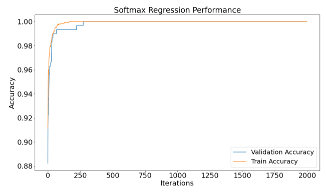
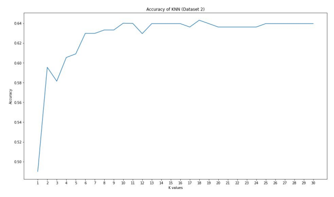


Figure 8.1 (left): KNN Accuracy (dataset 1)
Figure 8.2 (right): Softmax Accuracy (dataset 1)



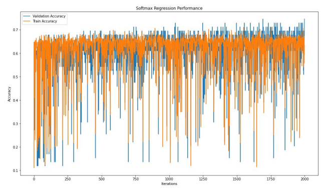


Figure 9.1 (left): KNN Accuracy (dataset 2)
Figure 9.2 (right): Softmax Accuracy (dataset 2)

A KNN-classifier was built to compare its performance with that of the softmax regression model. After training and validating the model with different hyperparameter values, K in this case, we found that the KNN model is at its best performance when K = 2 with an accuracy of 96.83%.

5. Discussion and conclusion

5.1 Key take-aways

During this project, we mainly explored the multi-class logistic regression by building it from scratch in Python. After training and validating the model under different scenarios, we had some findings that may be worth discussing:

- When we train the models with the digits dataset, the performance of multiclass logistic regression and KNN classifier are quite similar. However, when we train with the heart-h dataset, KNN performs more stable compared to the softmax regression model. Comparing the difference of two datasets, we find that the digits dataset has a larger number of features compared to the heart-h dataset so it leads to our assumption that when dataset has a limited number of features, KNN classifier is more stable than the multiclass logistic regression in terms of predicting the target classes.

5.2 Different hyperparameter values vs softmax regression performance

The performance and running time of the model with varying hyperparameter values were studied. As a reference, the best set of hyperparameter values for the digit dataset were: batch size = 75, learning rate =0.038, and momentum = 0.9.
After deciding on the optimal hyperparameters for the model, we used gradient descent to iterate 2000 times and obtained Figure 4. We observed that the accuracy convergent speed was very fast; both the validation and train accuracy almost reached 1.0 within 50 iterations. Therefore, we decided to conclude that this set of parameters are well-chosen so that gradient descent method does not need too many iterations to achieve a high prediction accuracy.

5.3 Training and Validation Curve for Different Optimization Hyperparameters

Learning Rate

Accuracy: For the digit dataset, we observed a general trend where, as the learning rate increased, the accuracy increased. (Figure 1.1). The maximum accuracy was achieved when the learning rate was 0.038. For the heart dataset, the validation accuracy fluctuated a lot until around learning rate of 0.014 where it plateaued.
Running time: (Figures 1.2 and 5.2), for both datasets, we observed a sharp increase in running time at a low learning rate (<0.01) and then a decrease. However, in the digit dataset, there were more fluctuations in running time and this trend observed can be an overgeneralization.

Batch Size

The validation and train accuracy fluctuate a lot regardless of the batch size for both datasets (Figures 2.1, 6.1). The highest accuracy was observed at batch sizes 75 and 60 for the digit dataset and the heart dataset, respectively.
Running time: For the digit dataset, as batch size increased, the running time decreased (Figure 2.2), but this was not the case for the heart dataset. In theory, as the batch size increases, the running time should increase since there are more batches to go through, and we suspect an error in the implementation which we could not identify.

Momentum

Accuracy: For the digit dataset, as the momentum increased, the accuracy first went down and then wet up a little, with a lot of fluctuations (Figure 3.1). The highest accuracy was found at momentum = 0.99 for the heart dataset.

Running time: Figure 3.2 shows that, for the digit dataset, the shortest running time was around 0.7. The running time generally decreased up until around momentum=0.7, which may be explained by the fact that momentum adds a fraction of the previous weight, increasing the step toward the minimum. For the heart dataset, the minimum running time was found around 0.36.

5.4 Comparison of the accuracy of KNN and softmax regression

A KNN-classifier was built to compare its performance with that of the softmax regression model. After training and validating the model with different hyperparameter values, K in this case, we found that the KNN model was at its best performance when K = 2 with an accuracy of 96.83%.

When we trained the models with digits dataset, the performance of multiclass logistic regression and KNN classifier were quite similar. They both predicted the targets with accuracy of around 95%. However, when we ran both models against the heart-h dataset, the performance of the softmax regression model fluctuated in the range between 50% to 70%, given different values of batch size, momentum and learning rate as inputs. However, the performance of the KNN classifier remained stable at around 65% after K increased to 5 and above. Comparing the difference of two datasets, we found that the digits dataset has a larger number of features compared to the heart-h dataset so it leads to our assumption that when dataset has a limited number of features, KNN classifier is more stable than the multiclass logistic regression in terms of predicting the target classes.

Limitations

Overall, we observed some unexpected trends, including increasing batch size not resulting in an increased running time. It is possible that something was not right in the implementation of the model, and or especially for dataset 2, the sample size was too small (n=294). Therefore, one future topic of exploration would be using a dataset with a large sample size.

6. Statement of contributions

The work was evenly distributed to three of the members. Zirui: Softmax regression and the Hyperparameter analysis. Cong: Dataset import, 5-fold cross validation and KNN method. Yumika: mini-batch gradient descent and 5-fold cross validation.

7. Reference

[1]Scikit Learn, “Digits Dataset” [Online].
Available:
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

[2]OpenML, “Heart Disease Databases” [Online].
Available: https://www.openml.org/d/1565

[3] Wilson J.R., Lorenz K.A. (2015) Short History of the Logistic Regression Model. In: Modeling Binary Correlated Responses using SAS, SPSS and R. ICSA Book Series in Statistics, vol 9. Springer, Cham.
https://doi-org.proxy3.library.mcgill.ca/10.1007/978-3-319-23805-0_2

[4] UFLDL Tutorial, “Softmax Regression”
Available:
http://deeplearning.stanford.edu/tutorial/supervised/SoftmaxRegression/