COMP 551 Fall 2020

Mini Project 2: Multi-Class Logistic Regression and Gradient Descent

Written Report

Group 6

Haowei Qiu (260762269) Liang Zhao (260781081) Ningchen Ma (260784506)

1 Abstract

In this project, we implemented a SoftMax Regression Model and a Gradient Descent Momentum Optimizer. The time-consumed and the average accuracy with 5-fold cross-validation is then be compared on both testing and training dataset to investigate the effects of using different hyper-parameters in the optimization procedure. Moreover, the multi-class logistic regression and KNN classifiers are compared in three different datasets with respect to their accuracy.

2 Introduction

In this project, we implemented a model with multi-class logistic regression(SoftMax) and mini-batch optimization using gradient descent with momentum. Three datasets are used: waveform-5000, optdigits and digits loaded from Scikit Learn. The model in our project is trained and tested with 5-fold validation and is used to estimate the performance on the resultant accuracy and the training time. We used the grid-search test to select the best hyper-parameters combination of batch size, learning rate and momentum. The termination condition is determined if the validation error has not been decreasing in the past T iterations and our optimizer will return the model with the best validation accuracy. Lastly, KNN is applied to the three datasets to compare with the Multi-class Logistic Regression in terms of accuracy. In our test with three datasets, KNN has higher accuracy than our implemented Multi-class Logistic Regression.

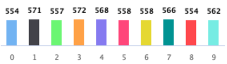
3 Datasets

3.1 optdigits

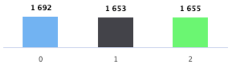
This is a dataset of Optical Recognition of Handwritten Digits which contains 64 features, 10 classes, 5620 instances.

3.2 waveform-5000

This dataset contains 3 classes of waves, 40 attributes describing the waveform and 5000 instances.



optdigit



waveform-5000

3.3 Digits Dataset from Scikit-Learn

The Digits Dataset is made up of 1797 8x8 images by collecting 250 hand-written samples from 44 writers. We turned the data into a matrix of samples and features. This dataset is displayed by the projection in 2 first principal axis in Figure 1.1.

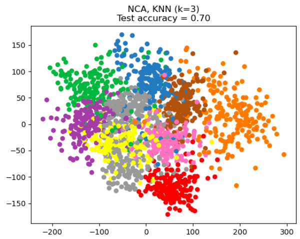


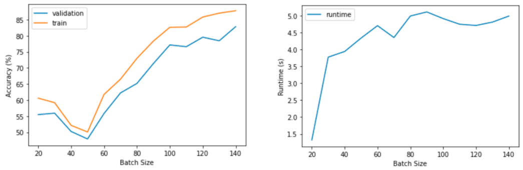
Figure 1.1: Scikit-learn digits

4 Results

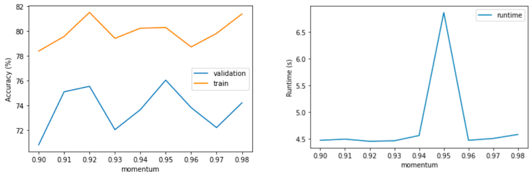
1. A discussion of how the multi-class logistic regression performance depends on the parameters of optimization

To analyze the performance of multi-class logistic regression, we plot the run time with respecting to three hyper-parameters: batch size, momentum and learning rate by keeping two of the parameters the same and varying the other parameter. Here we applied to digit and optdigit datasets as examples and the results are shown in graphs below.

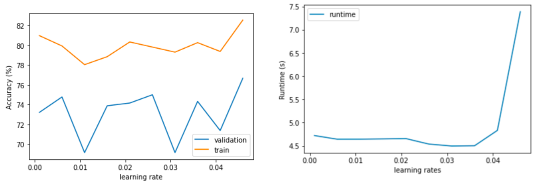
1.1.Results of accuracy and run time for different optimization hyper-parameters on Digits Dataset from Scikit-Learn



Runtime and accuracy for different batch size on Scikit-Learn dataset

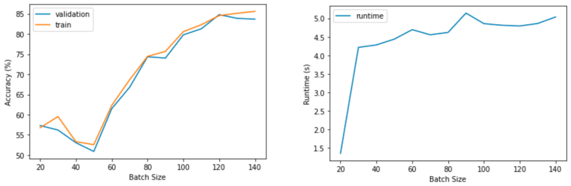


Runtime and accuracy for different momentum on Scikit-Learn dataset

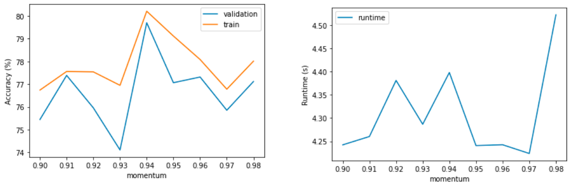


Runtime and accuracy for different learning rate on Scikit-Learn dataset

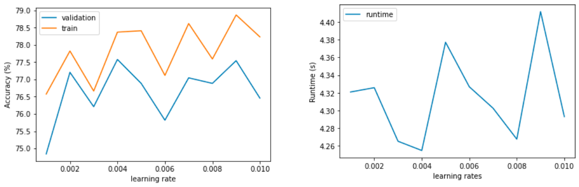
1.2.Results of accuracy and run time for different optimization hyper-parameters on dataset optdigits



Runtime and accuracy for different batch size on dataset optdigits



Runtime and accuracy for different momentum on dataset optdigits



Runtime and accuracy for different learning rate on dataset optdigits

By using grid search, we found that the best combination of hyper-parameters for dataset digits is batch size = 90, learning rate = 0.001, momentum = 0.93 and the accuracy with this combination is 0.803; For dataset optdigit, the best combination is batch size = 90, learning rate = 0.003, momentum = 0.93 and the accuracy with this combination is 0.867.

2. A discussion of the training and validation curve for different optimization hyper-parameters

In general training curves had higher accuracy compared with validation curves and the training curves had the similar shape as the validation curves for all different optimization hyper-parameters.
As our graphs in 1.1 shown, we found that while batch size increases, the accuracy would also increase for both training validation curve. However, we didn't see an obvious

relation between learning rate, momentum and accuracy, we will discuss this in details at conclusion section.

3. Comparison of accuracy of KNN and logistic regression.

We used KNN to compare with our implemented model. We applied both methods to our datasets. The best performances of these two methods are selected for comparison.

	waveform	optdigits	digits
KNN	0.794	0.985	0.973
LR	0.780	0.867	0.803

Table 1. Accuracy of KNN and Logistic Regression

5 Discussion and Conclusion

Overall, our model performed worse than the Scikit-learn implementations of KNN in terms of accuracy for three datasets, but we notice that for waveform dataset, our model is performing just as good as KNN, the reason might be that waveform has lots of outliers and our model is better at dealing with outliers than KNN. For effects of hyper-parameters, Batch size has a positive relation with accuracy in our model while we didn't see clear relations between momentum, learning rate and accuracy.

Batch size: For both scikit-learn digit and optdigit datasets, as batch size increases, the run time increases, this is because we use more data points to optimize (during the calculation of the gradient vector) our parameter in each iteration. The accuracy also increases, since the batch with greater size is more likely to have gradient vector that points along with the global gradient vector, therefore more likely to give us the global optimal w.

Learning rate: In general, the leraning rate(LR) has a significant effect on gradient descent. Too small LR would take too much time to converge, whereas too large LR will possibly lead to great oscillation or even overshoot. For our datasets, there isn't clear relationships between the LR and the accuracy and the run-time. The model performance with respect to LR mainly depends on the dataset we chose.

Momentum: For both scikit-learn and optdigit, there is no obvious trend between the accuracy and the momentum. The use of momentum is a measure to suppress the oscillatory behaviour of the gradient descent. It contributes little to the accuracy improvement. The run-time showed a random behavior as well.

For further improvement, if we have enough resources and time, we would use several more classifiers including Gaussian Naive Bayes for comparison instead of just using KNN and our model. We would also use more datasets, wider range of hyper-parameters(batch size, momentum and learning rate) and smaller step size of grid search to get more values on parameters for more comprehensive comparisons.

6 Statement of Contributions

Liang Zhao (ID: 260781081) contributed to dataset and Comparison against another classifier, Ningchen Ma (ID: 260784506) contributed to SoftMax Regression and Termination Conditions, and Haowei Qiu (ID: 260762269) contributed to Hyper-parameters of the optimization procedure. All 3 group members participated in the mini project report writing.