COMP5212 Programming Assignment 1 Report

Weiqi Wang
20563565
wwangbw@connect.ust.hk

1. Experiment Setup

We trained both Linear Logistic Regression model [3] and Linear Support Vector Machine [2] (without regularization term) with two types of optimizers: Stochastic Gradient Descent (SGD) [1] and SGD with momentum [5]. All models are trained on a single CPU with a batch size of 64 and 20 epochs in default.

To investigate the effect of different learning rate, we also experimented with eight learning rates in decreasing order: 5×10^{-2}, 1×10^{-2}, 5×10^{-3},1×10^{-3},5×10^{-4},1×10^{-4},5×10^{-5},1×10^{-5}. In total, $8 \times 2 \times 2 = 32$ models were trained and evaluated for further analysis.

When adding momentum to SGD optimizer, we set the momentum to 0.9, as it's the default value for momentum in many popular deep learning packages, such as PyTorch [4].

2. Loss Analysis

We plot the training loss in every epoch to reflect the convergence process. Specifically, the loss is calculated by the equation below.

$$L = \sum_{b=1}^{B} \sum_{d=1}^{D_b} \frac{loss(label_{b,d}, f_b(data_{b,d}))}{D_b}$$

B is the total number of batches, f_b is the model (Logistic regression or Linear SVM) after updated by b-th batch and D_b is the number of data points in b-th batch. An epoch is defined as one iteration of all dataset. For better visual analysis, we plot each model and each corresponding optimizer separately. To compare the effect of different learning rate, we also plot the curve according to the record of four learning rates: 5×10^{-2}, 5×10^{-3}, 5×10^{-4}, and 5×10^{-5}. The curve is presented in figure 1.

From the epoch training loss curve, we can observe that all training process are converging smoothly. Specifically, all figures followed a pattern of red-green-orange-blue from top to bottom. This indicates that the convergence speed of the model is related to the learning rate. The lower the learning rate, the more number of epochs required for the convergence of the training loss. Most of the models successfully converged with a loss that is near to 0. However, for Logistic Regression with simple SGD and a low learning rate (the red line in the left upper plot), the loss is still high. A potential reason is that the model is not fully converged yet, which needs more epoch to train the model better. Alternatively, we can either increase the learning rate or add momentum so that the model will also converge faster.

By comparing between two optimizers, we can observe that adding momentum to SGD can significantly decrease the loss in every epoch. For example, at the first epoch, Logistic Regression with simple SGD has a training loss of 0.55 while SGD-Momentum has only 0.3. This is understandable because momentum is effectively accelerating the parameter adjusting process in every iteration, thus even one epoch can lead to larger decrease in training loss.

3. Test Set Accuracy

We report the image classification accuracy on test set in table 1. From the statistics, we can find that almost all models perform perfectly on test set, where all models achieved at least 99% accuracy. Specifically, we observe that Support Vector Machine's performance is slightly better as some of them even achieved 100% accuracy.

	Logistic Regression		Support Vector Machine	
Learning Rate	SGD	SGD-M	SGD	SGD-M
5×10^{-2}	99.95%	99.95%	99.95%	100.0%
1×10^{-2}	99.91%	99.95%	100.0%	100.0%
5×10^{-3}	99.91%	99.95%	100.0%	99.95%
1×10^{-3}	99.86%	99.91%	99.95%	99.95%
5×10^{-4}	99.81%	99.91%	99.95%	99.95%
1×10^{-4}	99.81%	99.86%	99.81%	99.95%
5×10^{-5}	99.81%	99.81%	99.72%	99.95%
1×10^{-5}	99.53%	99.81%	99.62%	99.81%

Table 1. Test Accuracy by both models trained with two optimizers on different learning rates.

Among logistic regression, we can see that as the learning rate decreases, the test accuracy is slightly decreasing. This is possi-

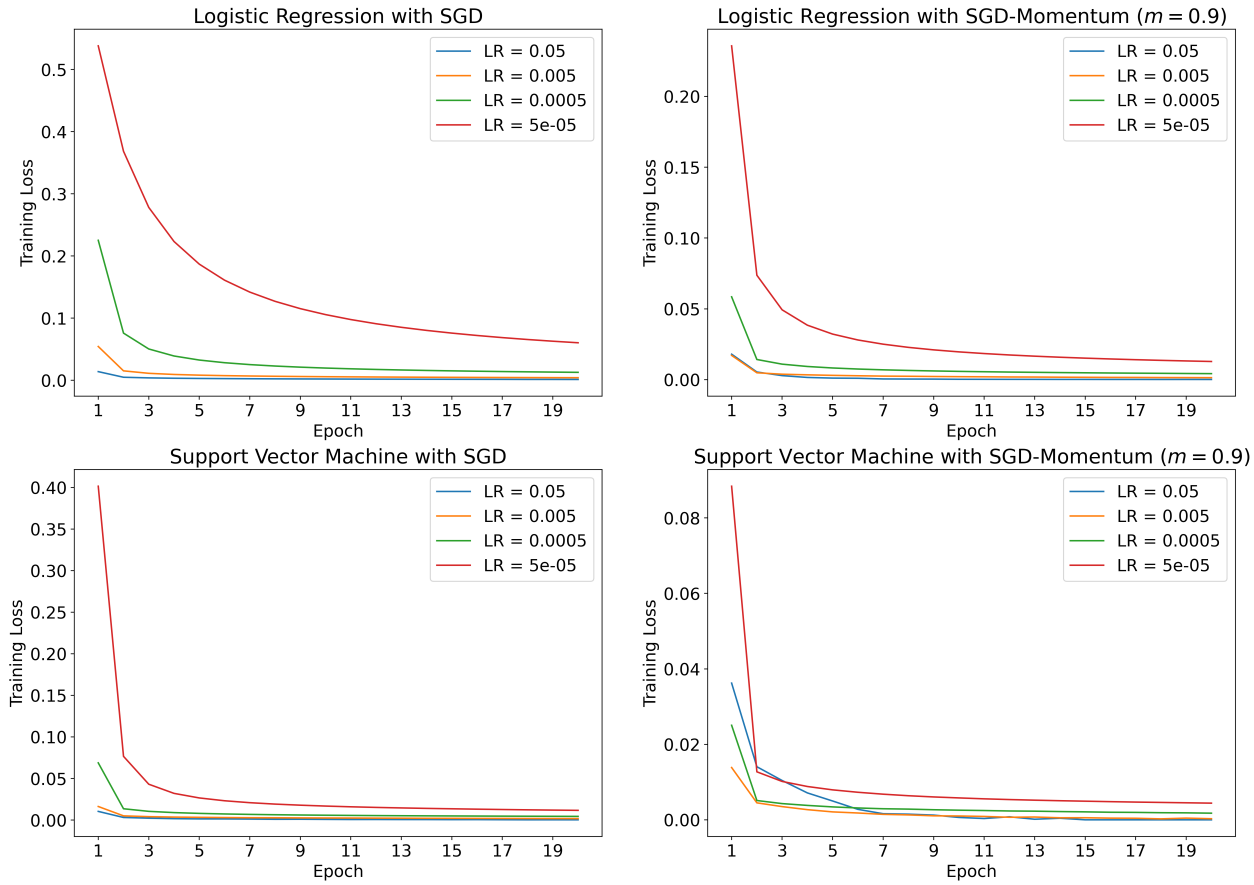

Figure 1. Training loss curve for both models with SGD and SGD-Momentum optimizers in three learning rates: 5×10^{-2}, 5×10^{-3}, 5×10^{-4}, and 5×10^{-5}. The "LR" in legend refers to learning rate.

bly because that the models are not fully trained due to limited epochs. Training them with a larger number of epochs can lead to better consistency. At the same time, comparing SGD with adding momentum to SGD, we observe that the test accuracy slightly increases a bit. This suggests that the model indeed converges faster with the help of momentum.

By comparing the performance of linear support vector machine, we can find that SVM is much more stabilized. This indicates that classifying digits 0 and 1 can be solved by transforming the image input into a numerical value and locate a boundary to classify them to two classes. Similar conclusions can be drawn by comparing two optimizers, adding a momentum can lead to higher test accuracy when the learning rate is low (e.g., 1×10^{-5}).

4. Conclusion

In conclusion, we found that both Logistic Regression and SVM models can conquer the task of digit classification on 0 and 1 with a very high test accuracy. Specifically, SGD with momentum can lead to faster convergence compared with SGD, which is particularly effective at small number of epochs and low learning rate. Also, the lower the learning rate, the more epochs it takes to fully train the model.

References

[1] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012. 1

[2] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 1

[3] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013. 1

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[5] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 1