COMP5212 Machine Learning
Programming Homework 1

LO, Li-yu
20997405
e-mail: lloac@connect.hkust.hk

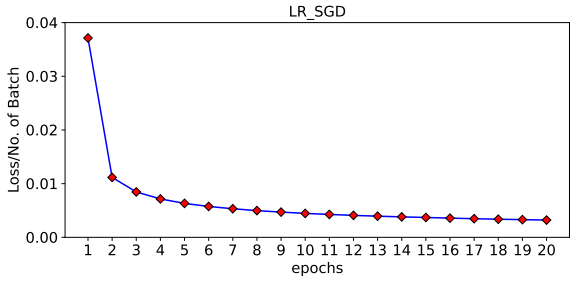
October 16, 2023

Abstract

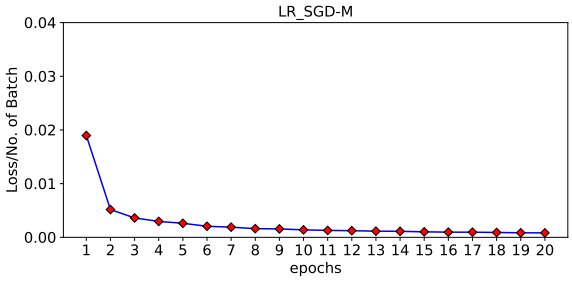
In this assignment, a binary classification model based on (1) **Linear Logistic Regression (LLR)** and (2) **Linear Support Vector Machine (LSVM)** is trained. In particular, we perform the the classification on the **0** and **1** subclass of the MNIST digit dataset, whilst adopting PyTorch library as our main programming framework. Below then reports the training results.

General Training Results

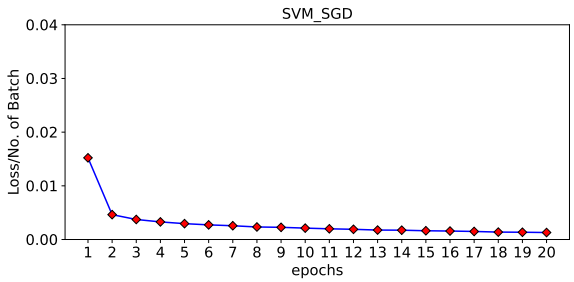
Below first present the training results of each model by plotting the *batch-averaged loss versus epochs*, in which LLR and LSVM utilize both Stochastic Gradient Descent (SGD) and Stochastic Gradient Descent Momentum (SGD-M) optimizer. Also note that in the below cases, we set *learning rate* $\alpha = 0.01$ and *epoch* $= 20$. For SGD-M, we set momentum beta coefficient with $\beta = 0.9$.



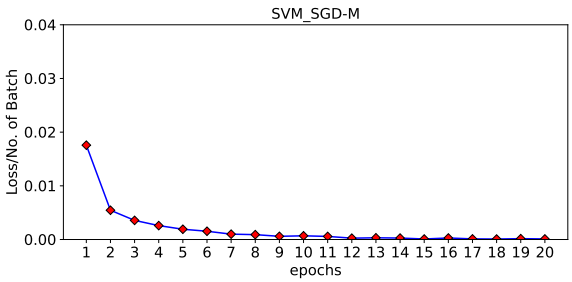
(a) Linear Logistic Regression w/ Stochastic Gradient Descent



(b) Linear Logistic Regression w/ Stochastic Gradient Descent Momentum



(c) Linear Support Vector Machine w/ Stochastic Gradient Descent

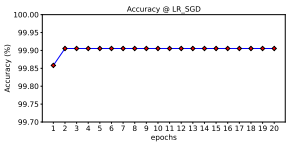


(d) Linear Support Vector Machine w/ Stochastic Gradient Descent Momentum

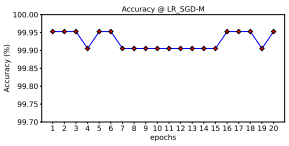
Figure 1: Training results of different cases.

Testing Accuracy

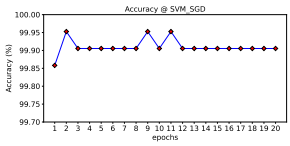
We also report the *accuracy versus epoch* retrieved from the testing dataset, where plots of the 4 cases are shown. It can be observed that for all 4 cases, all models swiftly approach to accuracy of over 99 %, which show extremely high performance in the binary classification task.



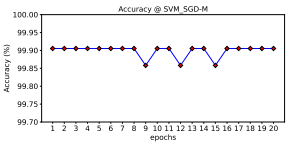
(a) Linear Logistic Regression w/ Stochastic Gradient Descent



(b) Linear Logistic Regression w/ Stochastic Gradient Descent Momentum



(c) Linear Support Vector Machine w/ Stochastic Gradient Descent



(d) Linear Support Vector Machine w/ Stochastic Gradient Descent Momentum

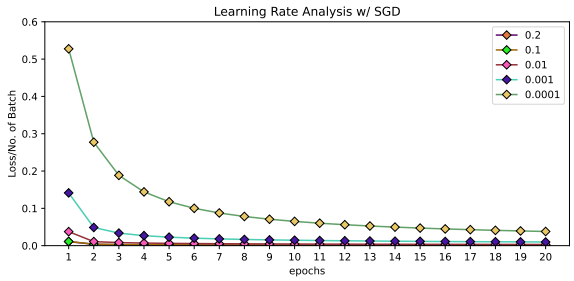
Figure 2: Model accuracy versus epochs.

On SGD v. SGD-M

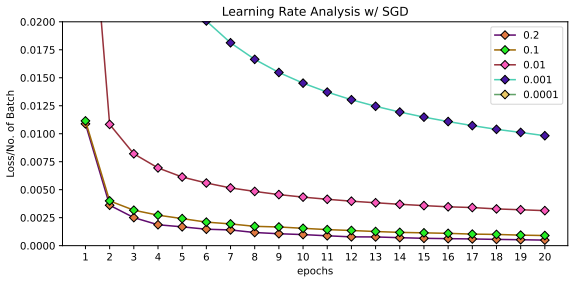
By referring to figure 1a and 1b, or figure 1c and 1d, it can be easily seen that with momentum, in which previous gradients information are taken into consideration, the training process could converge faster. Although in the case of support vector machine, i.e., figure 1c and 1d, SGD with momentum converge slightly slower at first, yet, after 5 epochs, it could be observed that it further suppress the loss, leading to a more optimal minimum region.

On Learning Rate α Settings

The last section presents the analysis on hyperparameter settings of learning rate α. To fixate same comparison condition, we use Linear Logistic Regression with Stochastic Gradient Descent for analysis. In addition, for α, we set $\alpha = [0.2, 0.1, 0.01, 0.001, 0.0001]$. Below plots the final comparative results. From above, it can be easily concluded that with larger learning



(a) Linear Logistic Regression w/ Different α



(b) Zoomed-in Linear Logistic Regression w/ Different α

Figure 3: Learning rate analysis.

rate, the model could converge faster, as we are moving our model in the downward slope with larger step. Nevertheless, it is also discovered that, when setting α with a larger scale, say $\alpha = 0.9$, the learning process could result in disastrous failure, since such settings could lead to huge divergence. For instance, weights could be updated by large values with large α, and hence leading to an exploding gradients; then, the training will end up diverging at an exponentially increasing rate.