National Tsing Hua University

Fall 2023 11210IPT 553000

Deep Learning in Biomedical Optical Imaging

Homework 2

CHUN-HAO LI[1]

[1]*Ultrafast Photonics Lab, Institute of Photonics Technologies, Tsing Hua University.*

Student ID:112066513

1. Task A: Performance between BCE loss and CE loss

1.1 Analysis of BCELoss and CELoss

Since BCELoss contains a sigmoid function in the process, we can see the result tend to be oscillate in a much smaller extent than CELoss, due to CELoss has more than one class and use softmax function to make sure the outputs are between 0 and 1, then use sum to make sure the total summation is 1.

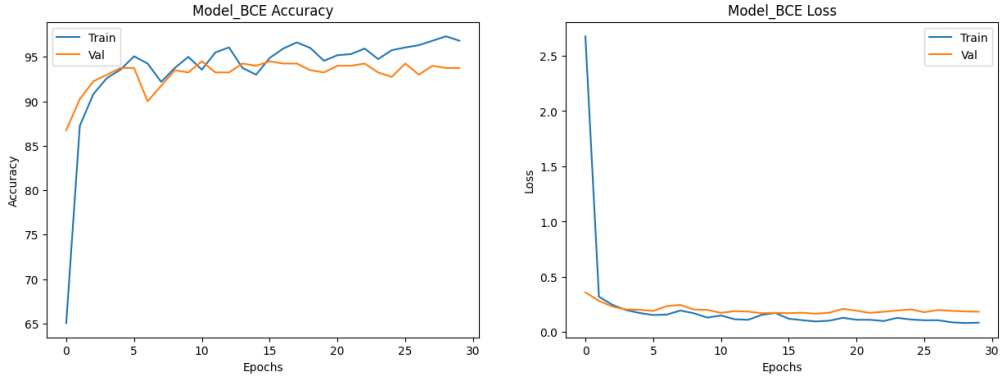

Fig. 1. Result from training with BCELoss

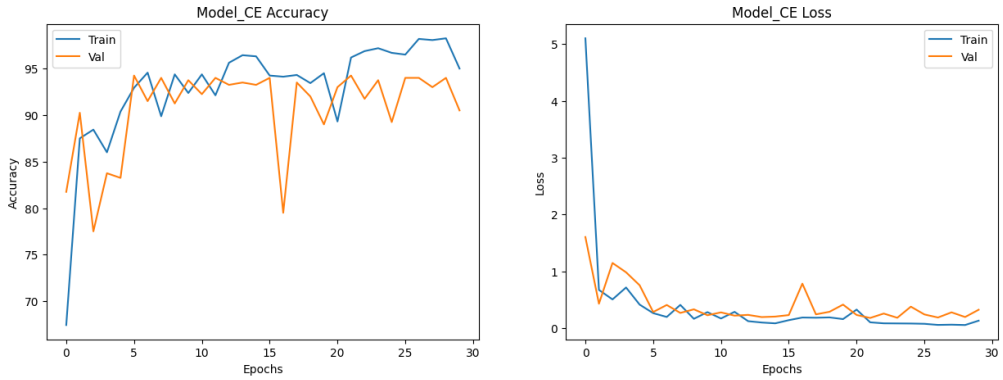

Fig. 2. Result from training with CELoss

2. *Task B: Performance between Different Hyperparameters*

For the hyperparameters I choose are 1. Batch Size and 2. Learning Rate Scheduler. So, the following I will divide the report into 2 parts.

2.1 Analysis with Batch Size

In this part, I use the following as the initial hyperparameters for the training program: epoch equals 30, BCEWighLogistsLoss as loss function, Adam as the optimizer, CosineAnnelingLR as learning rate scheduler, and these hyperparameters won't change during the first part. I also use two hidden layers with the input and output are (256*256*1, 64), (64, 64) for both training and testing part.

The results of Part 2.1 are shown as the Fig.3.4.5. With dividing batch size by 16, 64, 256, we can find that the result of training accuracy becomes much smooth when the batch size gets bigger. I find it is because when batch size is small, although the training validation will be much accurate during the training process, but the training loss can't go less than 0.08. As the batch size gets bigger, training validation will become less accurate, but the training loss can get smaller, 0.0422 with batch size equals 256.

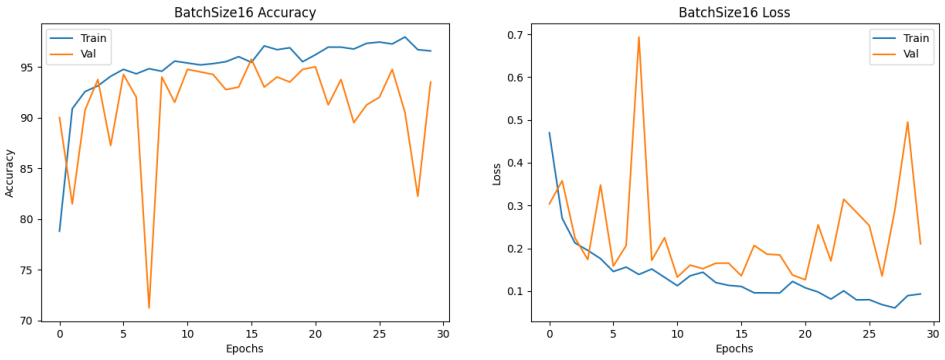

Fig. 3. Results from using batch size = 16

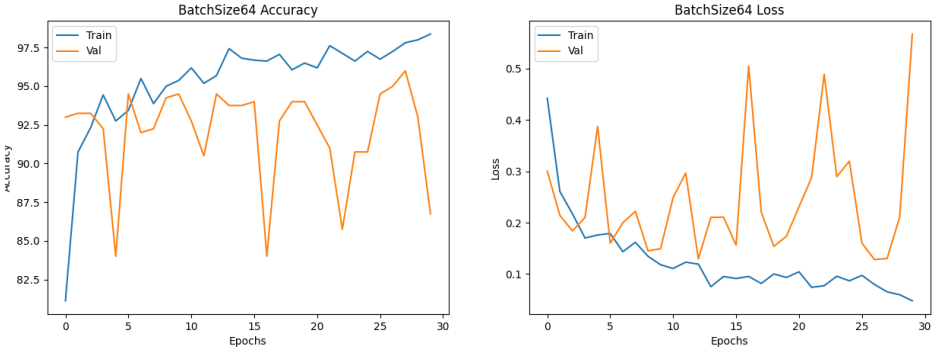

Fig. 4. Results from using batch size = 64

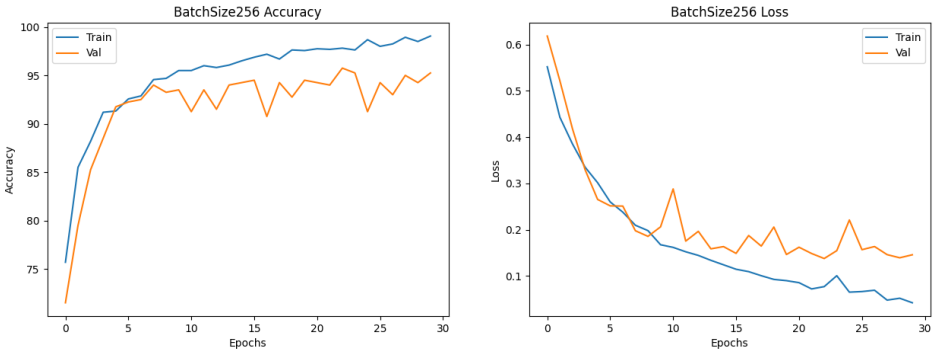

Fig. 5. Results from using batch size = 256

Next, is the test results from each batch size: (1) Batch size=16: test accuracy 69.5% (2) Batch size=64: test accuracy 70.75% (3) Batch size=100: test accuracy 73%. Just as the training results show the loss can reach smaller as batch size gets bigger, test accuracy

becomes larger when batch size gets bigger. And we can see there is a difference between training accuracy and test accuracy, this is mostly due to the testing dataset comes from different distribution compare to training dataset.

2.2 Analysis with different Learning Rate Scheduler

In this part, I use the following as the initial hyperparameters for the training program: batch size equals 30, epoch equals 30, BCEWighLogistsLoss as loss function, Adam as the optimizer, and these hyperparameters won't change during the second part. I also use two hidden layers with the input and output are (256*256*1, 64), (64, 64) for both training and testing part. And I choose CosineAnnelingLR, OneCycleLR, ExponentialLR as testing learning rate scheduler function. The results are shown in the following figures.

From the three Learning Rate Scheduler, we can see the training accuracy of CosineAnnealingLR and ExponentialLR tends to goes up as the epochs process, and training accuracy of OneCycleLR reaches maximum at about 20% of the process, then slowly decreases to a value smaller than the begining. The validation accuracy of ExponentialLR is the steadiest of the three, can mostly stays at around 90%-95% correct, as well as the validation loss is the steadiest, either. Although OneCycleLR has the most chaotic graph, we can see the training and validation loss is the steadiest of the three (mostly below 1), but when I look into the result data, OneCycleLR has the worst training result, can only reach 0.26 at train loss, 90.12% at train accuracy, 0.19 at validation loss, 93.75% at validation accuracy.

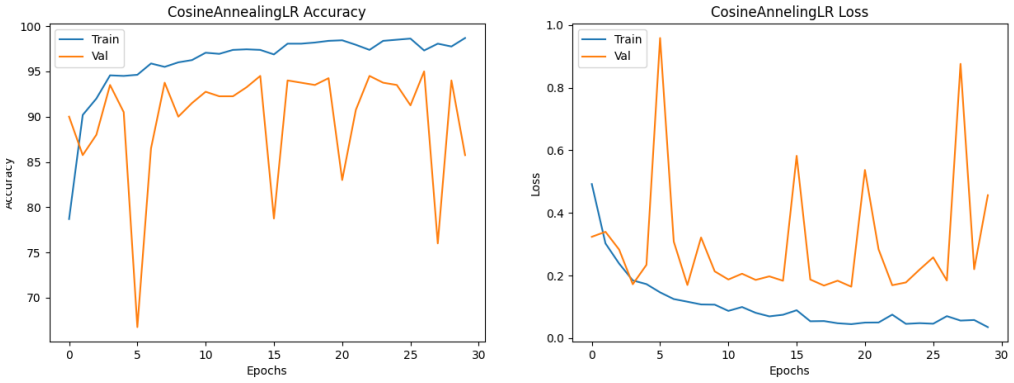

Fig. 6. Results from using CosineAnnealingLR

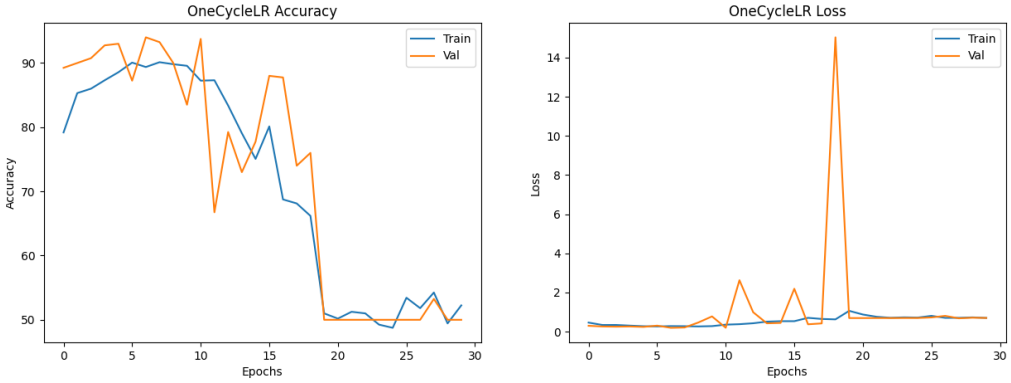

Fig. 7. Results from using OneCycleLR

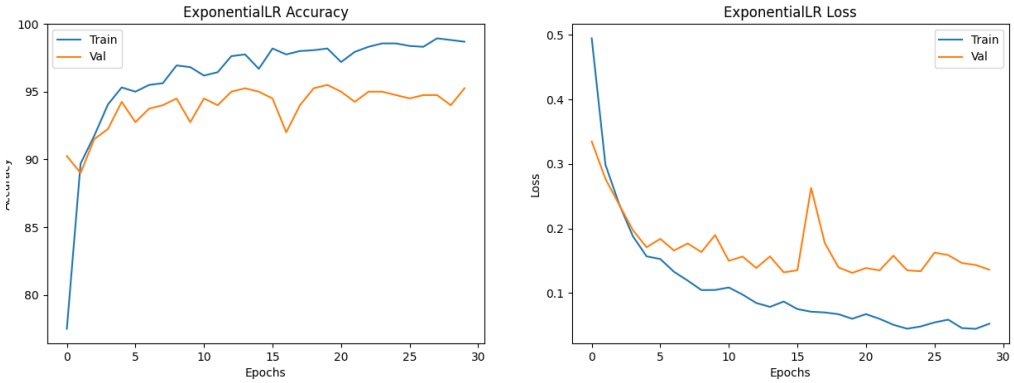

Fig. 8. Results from using ExponentialLR

Next, is the test results from each batch size: (1) CosineAnnealingLR: test accuracy 78.75% (2) OneCycleLR: test accuracy 80.5% (3) ExponentialLR: test accuracy 72.5%. Just as the training results show the validation becomes smaller as batch size gets bigger, test accuracy becomes larger when batch size gets bigger. So, we can see is OneCycleLR has the most accurate test accuracy of the three methods. And also, the same as Part 2.1, we can see there is a difference between training accuracy and test accuracy, it is mainly because the data source comes from different distribution.