National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework 2

陳芷韻

Student ID:112066514

1. Abstract

In the report, the performance of training results with two loss function, BCEWithLogitsLoss (BCE) and Cross Entropy Loss (CE) will be shown respectively in section 2. In the section 3, we will change the hyperparameters learning rate scheduler and batch size and compare the performance. After comparison, we can't see any obvious improvement. However, we found that no matter how good the training and validation results are, the accuracy of the test stays around 74%. This may indicate that the distribution of the training dataset and the test dataset are different.

2. Performance between BCE loss and BC loss

2.1 Fixed Parameters in the Test

The following table shows the fixed parameters in the comparison of BCE and CE loss function.

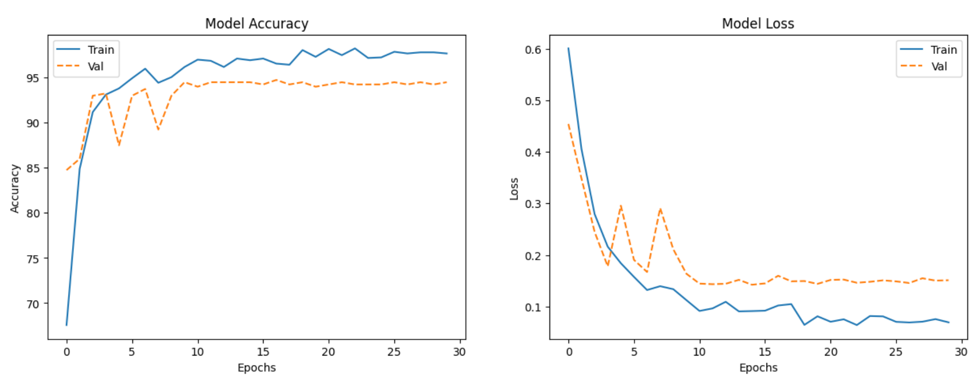
Table 1. Fixed parameters in the comparison.

Parameters	Epoch	Batch size	Learning rate scheduler
	30	32	StepLR

2.2 Visualization of model performance

(Figure 1) and (Figure 2) shows the accuracy and loss evolution during the training process of BCE and CE loss respectively.

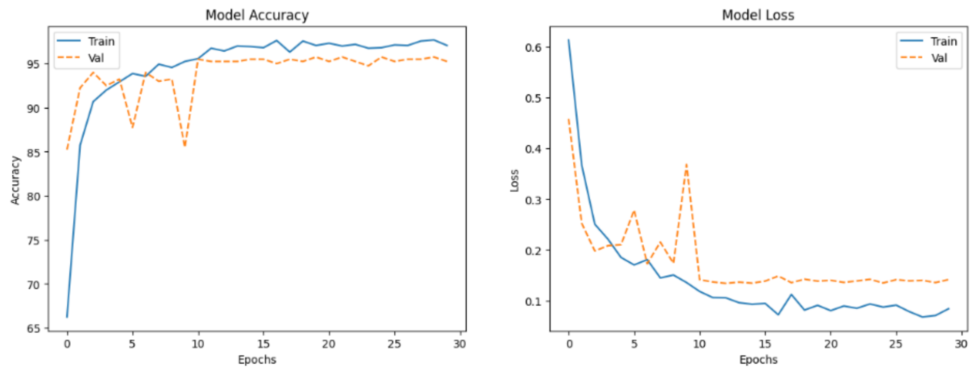
BCE loss:



(a) Model accuracy evolution (b) Model loss evolution

Figure 1. Training evolutions with the utilization of BCE loss.

CE loss:



(a) Model accuracy evolution (b) Model loss evolution

Figure 2. Training evolutions with the utilization of CE loss.

From the accuracy in **(Figure 1-(a))** and **(Figure 2-(a))**, compare the difference between train and validation results, we find the CE loss structure may have the better performance than that in BCE loss, or if the difference is large enough, it may overfitting in this model. The following table 2 shows the comparison result, actually, we could not see the obvious improvement by the test accuracy. However, such low test accuracy may indicate the different distributions of the training data and test data.

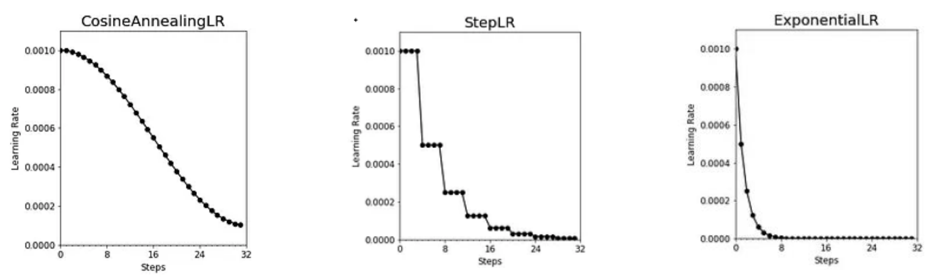
Table 2. Train, validation and test results.

	Train Loss	Train Accuracy	Val Loss	Val Accuracy	Test Accuracy
BCE	0.0777	97.44%	0.1416	95.25%	74.75%
CE	0.0840	97.06%	0.1416	95.25%	76.75%

3. Performance between Different Hyperparameters

3.1 Hyperparameter 1: Learning rate scheduler

First, we change the hyperparameter, three kinds of learning rate scheduler, CosineAnnelingLR, StepLR, and ExponentialLR, as **(Figure 3)** shows. The common feature among these three schedulers is the utilization of a decreasing learning rate, typically with a smaller learning rate being employed during the later stages of training for finer convergence.

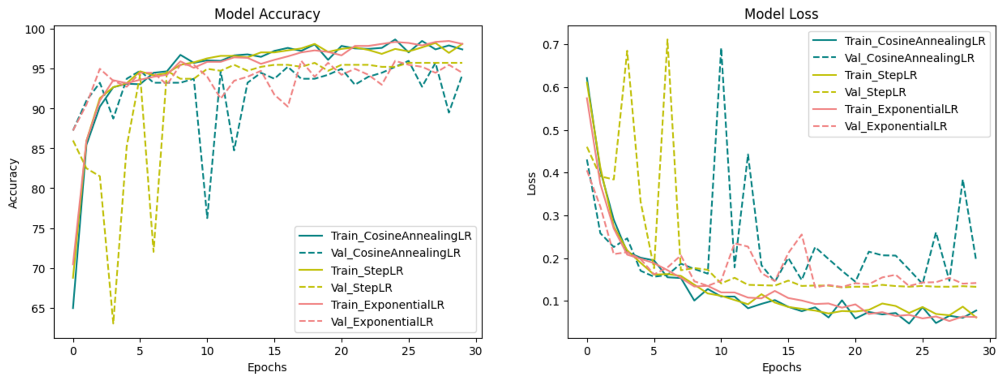


(a) CosineAnnelingLR (b) StepLR (c) ExponentialLR

Figure 3. Learning rate scheduler [1]

In the turning process, I use for loop to change the target parameter, and found each time we adjust a parameter, the model must be re-run, otherwise, the model will retain the previously trained weights. After putting the model and some parameter have to be re-run in the loop, it success, and the results as **(Figure 4)** shows.

Visualization of model performance:



(a) Model accuracy evolution (b) Model loss evolution

Figure 4. Training and validation evolutions with different learning rate scheduler.

Table 3. Train, validation and test results.

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
CosineAnnelingLR	0.0777	97.44%	0.1928	94.25%
StepLR	0.0613	98.12%	0.1331	95.75%
ExponentialLR	0.0624	98.12%	0.1420	94.50%

(Table 3) and **(Figure 4)** show that the training and validation results are similar across the three different learning rate schedulers. However, in the case of CosineAnnealingLR, there are more noticeable fluctuations in performance during the later stages.

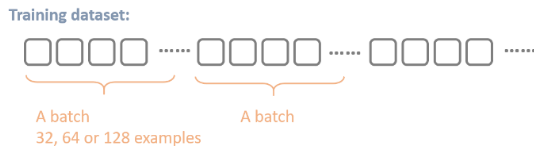
Table 4. Corresponding ***test accuracy*** in the models with different learning rate schedular:

Learning rate	CosineAnnelingLR	StepLR	ExponentialLR
Test accuracy	74.5%	71.5%	73.75%

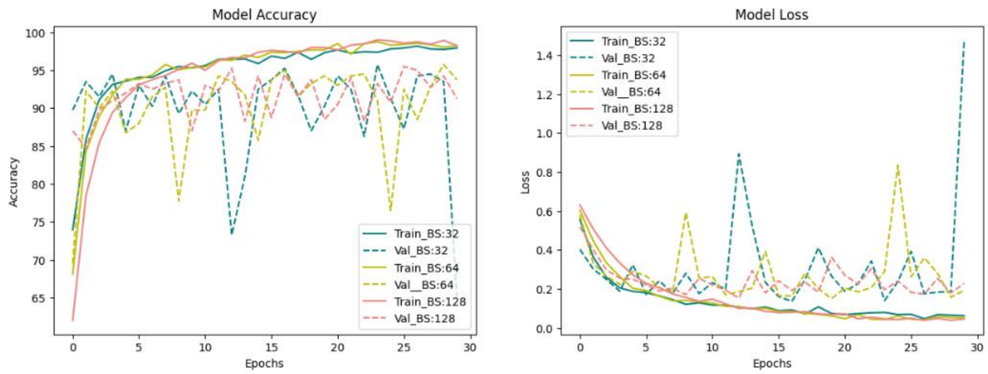
From the above (Table 4), we find the test accuracy always preform around 75 % which cannot be improve, even if both of training accuracy and validation accuracy are high. Since the dataset distribution of the training data and test data are different. For example, the dataset may obtain different machines. Thus, during the deep learning process, it is better to confirm the distribution of training dataset and test dataset are same.

3.4 Hyperparameter 2: Batch size

Second, we change the batch size for the load of training data, the diagram of batch size as following figure shows. (Figure 5) and (Table 5) shows the training and validation result, as the batch size are 32, 64, 128 respectively.



Visualizing model performance:



(a) Model accuracy evolution (b) Model loss evolution

Figure 5. Training and validation evolutions with different learning rate scheduler.

From (Figure 5-(a)), we can see the larger batch size with higher accuracy, however, case 32 in this training experiment with large fluctuations. The train and validation result of three cases is shown in (Table 5). From the table, we can see the similar accuracy as batch size are 64 and 128, and the lower accuracy as batch size is 32. Similarly, the test result in (Table 6) also show the un-improved accuracy and different distribution between training data and test data.

Table 5. Train, validation and test results with different batch size.

Batch size	Train Loss	Train Accuracy	Val Loss	Val Accuracy
32	0.0626	97.94%	1.4776	64.75%
64	0.0524	98.25%	0.1918	93.75%
128	0.0456	98.25%	0.2277	91.25%

Table 6. Corresponding test accuracy in the models with different batch size.

Batch Size	32	64	128
Test accuracy	73.75%	74.25%	75.75%

4. References

[1] Figure3: https://towardsdatascience.com/a-visual-guide-to-learning-rate-schedulers-in-pytorch-24bbb262c863#eec7 (2023/10/16)