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

JUNQIANG-TANG

[1] *Institute of Photonics Technologies, National Tsing Hua University, Hsinchu 30013, Taiwan*

Student ID: 110066544

1. Abstract

In this project, the result by using Binary Cross-Entropy (BCE) loss for task A is the accuracy of 72.75%, and the one by using Cross-Entropy (CE) loss is the accuracy of 69.50%. Then, I will talk about the performance between different hyperparameters. The hyperparameters I chose to change are batch size and learning rate. The performance is getting better when I increase the batch size and learning rate, respectively.

2. Introduction

In the beginning, we load and process the chest X-ray dataset. There are the files of abnormal and abnormal scans in the dataset. The purpose of this binary classification is to identify the pictures of pneumonia and normal one. If a pneumonia picture is scanned out, which will be labeled as 1. In the other hand, if the picture is scanned as normal lungs, it will be labeled as 0. Inside the code, it has already designed the neural network by using sequential method. The activation layer used inside is ReLU. Then, we are going to training the neural network we made on the chest X-ray dataset with both Binary Cross-Entropy and Cross-Entropy loss.

The purpose of this project is to compare and analyze the model's performance between Binary Cross-Entropy loss and Cross-Entropy loss. Moreover, with the same deep learning architecture and hyperparameters. Furthermore, by changing the hyperparameters, try to observe and analyze those different results.

3. Results

3.1 Performance between BCE loss and CE loss

In the training phase, it shows an almost consistent decrease in BCE loss, which means that the model is learning to minimize the binary classification loss. And a keeping increasing training accuracy means that the model is learning from the training data. In the same way, so does CE loss. As for the testing phase, I got the test accuracy of 72.75% (Fig. 1) and 69.50% (Fig. 2) for BCE loss and CE loss, respectively.

From the results, it looks like that BCE loss is still more appreciate for binary classification, because it directly models the probability which belongs to one class. As for CE loss, it is designed to deal with multi-class cases, so it is more suitable for modeling multi-classification. I think that is a reason why the test accuracy from CE loss is a little worse than that with BCE loss, because the results in our project are binary (pneumonia vs. normal). Despite this, both loss functions still could be used effectively in this project. We can see that the test accuracy difference between BCE loss and CE loss is not a massive number and both versions can make reasonable predictions in the end.

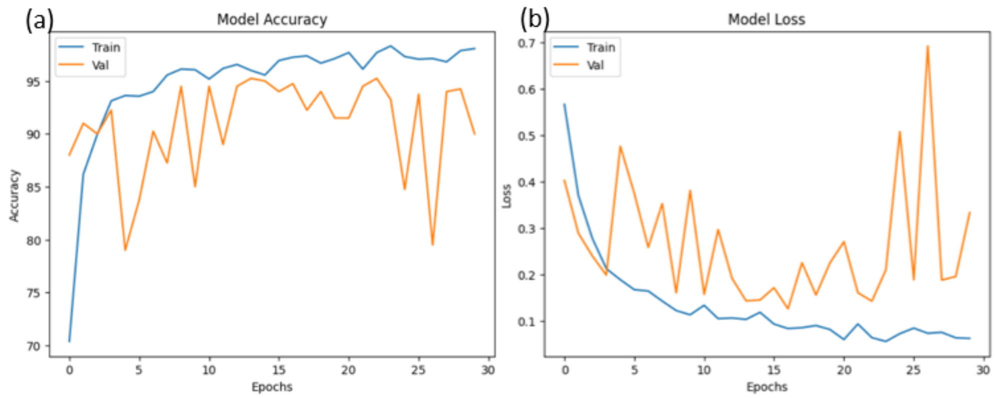

Fig. 1 (a) The plot of training and validation accuracy with BCE loss. (b) The plot of training and validation loss with BCE loss. Blue line is training curve and orange line is validation curve.

From the results, it looks like that BCE loss is still more appreciate for binary classification, because it directly models the probability which belongs to one class. As for CE loss, it is designed to deal with multi-class cases, so it is more suitable for modeling multi-classification. I think that is a reason why the test accuracy from CE loss is a little worse than that with BCE loss, because the results in our project are binary (pneumonia vs. normal). Despite this, both loss functions still could be used effectively in this project. We can see that the test accuracy difference between BCE loss and CE loss is not a massive number and both versions can make reasonable predictions in the end.

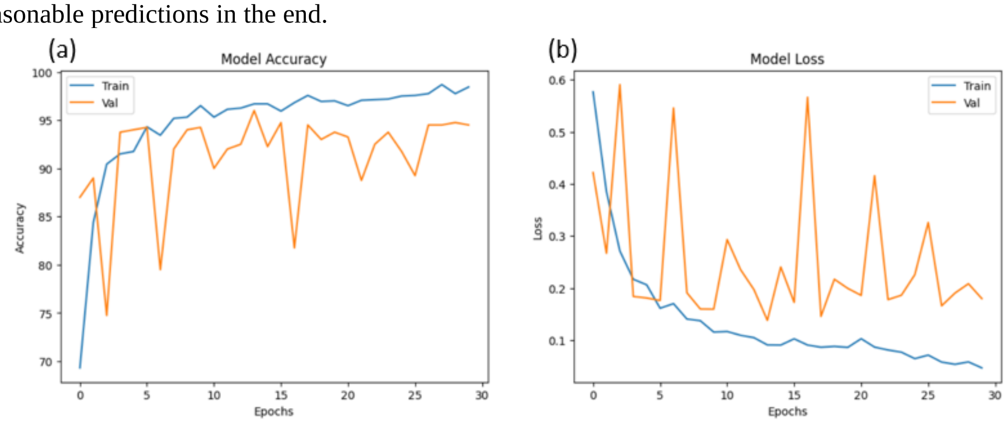

Fig. 2 (a) The plot of training and validation accuracy with CE loss. (b) The plot of training and validation loss with CE loss. Blue line is training curve and orange line is validation curve.

In summary, although the test accuracy generated by CE loss (69.50%) is slightly lower than that by BCE loss (72.75%), the results still represent that both loss functions can be used effectively for binary classification.

3.2 Performance difference between different hyperparameters

The hyperparameters I chose to change are batch size and learning rate. First, I change the batch size to 8, 32, and 128 to see the difference, the results I got are 73.75%, 75.25%, and 79.25%, respectively (Fig. 3). Then, I tuned the learning rate to 0.1, 0.0001, and 0.000001, the test accuracies are 79.50%, 75.25%, and 69.50%, respectively (Fig. 4).

After increasing the batch size from 8 to 128, I found that both training and validation curve are getting smoother, and the training time decreased a lot. In the beginning (batch size = 8), the training process cost me almost one minute to finish it. But with the batch size of 128, it only needs roughly 20 seconds to achieve it. In fact, the change of batch size does affect the convergence speed. In my case, a larger batch size makes the convergence faster as shown in Fig. 3, because many samples were being processing in each iteration, and the model parameters are updated less frequently. So, we can see a more stable gradient coming out and a faster training time. And in this model, the performance with larger batch size is also better than that with smaller batch size. I think it is because there are more samples in each batch, so the model could collect various set of examples during each update, which leads to an improvement.

Then, I keep the batch size to 128, and decrease the learning from 0.1 to 0.000001. The result shows that a larger learning rate has a better performance, and the training curve is smoother when the learning is increased. Usually, a large learning rate can make the model easy to overfitting because it quickly adapts to the training data and ignores some patterns. So, I'm not sure whether the overfitting appears when I tune the learning rate to 0.1. Maybe the best learning rate for these three experiments is 0.001.

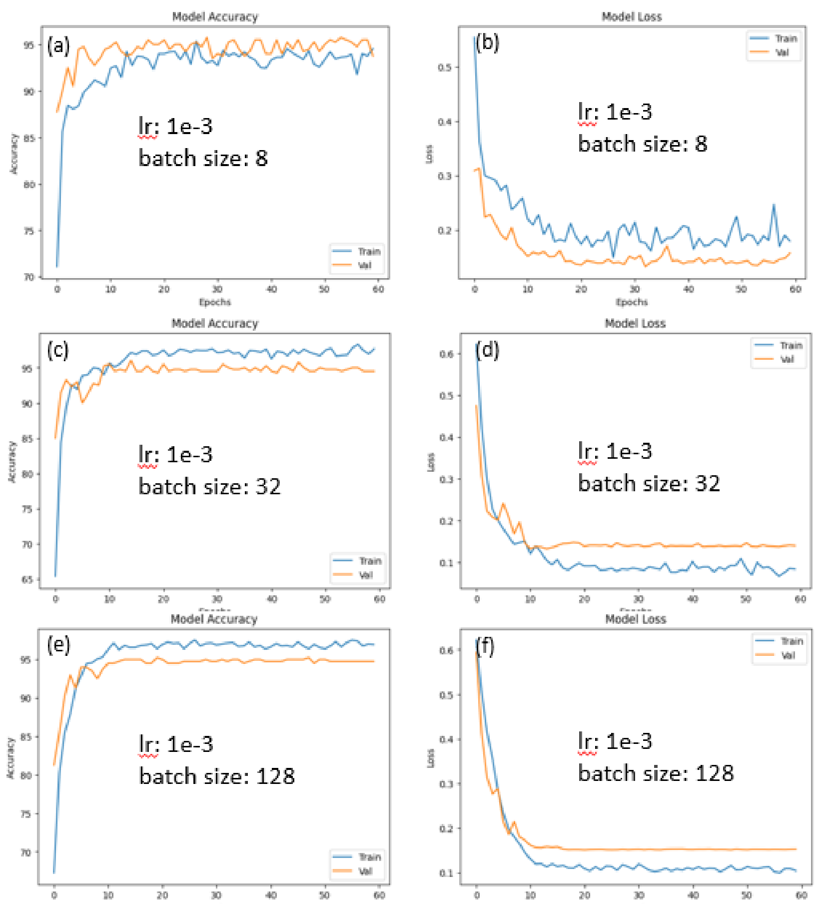

Fig. 3 The plots of training and validation accuracy, and training and validation loss with BCE loss. Blue line is training curve and orange line is validation curve. (a), (b) Batch size is 8 with test accuracy of 73.75%. (c), (d) Batch size is 32 with test accuracy of 75.25%. (e), (f) Batch size is 128 with test accuracy of 79.25%.

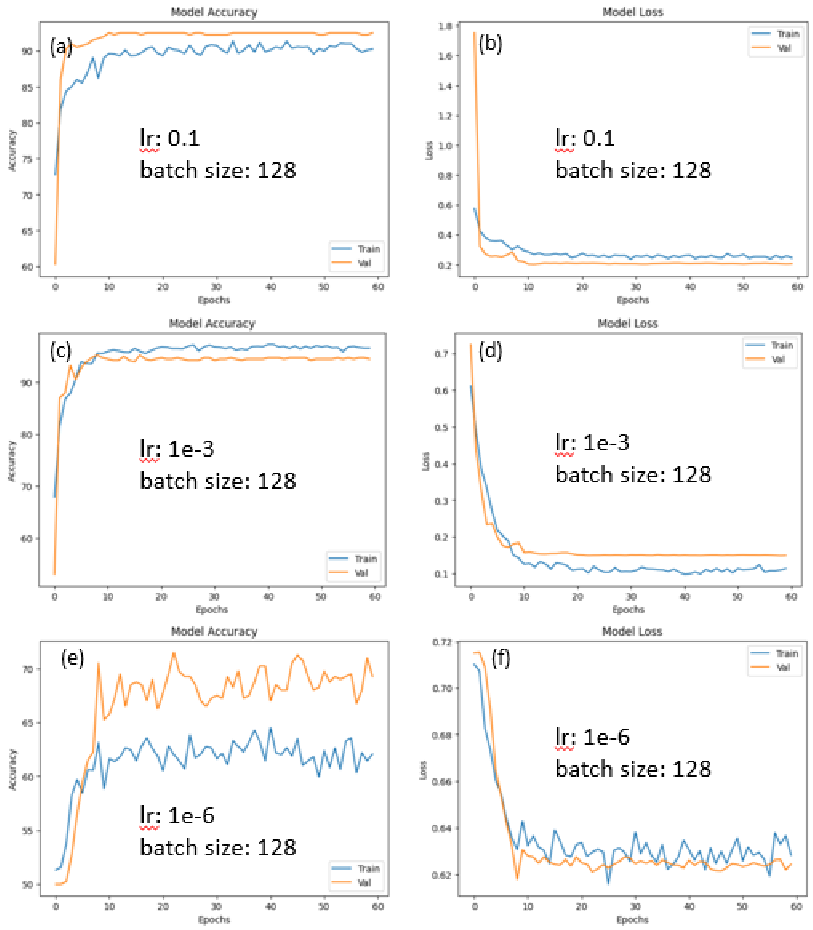

Fig. 4 The plots of training and validation accuracy, and training and validation loss with BCE loss. Blue line is training curve and orange line is validation curve. (a), (b) Learning rate is 0.1 with test accuracy of 79.50%. (c), (d) Learning rate is 0.001 with test accuracy of 75.25%. (e), (f) Learning rate is 69.50 with test accuracy of 69.50%.