National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework4

陳芷韻 *Student ID:112066514*

1. Model Selection

1.1 Model Choice

First, I search the common used transform learning model for small training dataset, and find the common used are MobileNet, squeezenet1_0, convnext_small…and others. I chose these three models to do transform learning in both "Fine-turning" and "Fixed Feature Extractor", however, in the first task, fine-turning, first two have good performance, while in the second task of fixed feature, the training and validation accuracy are zeros all the time, and convnext_small take much training time. Thus, I try the other common model with fully connect layer, **ResNet and GoogleNet**, whose accuracy that after training are nonzero, but the performance not as good as expected.

1.2 Explanation

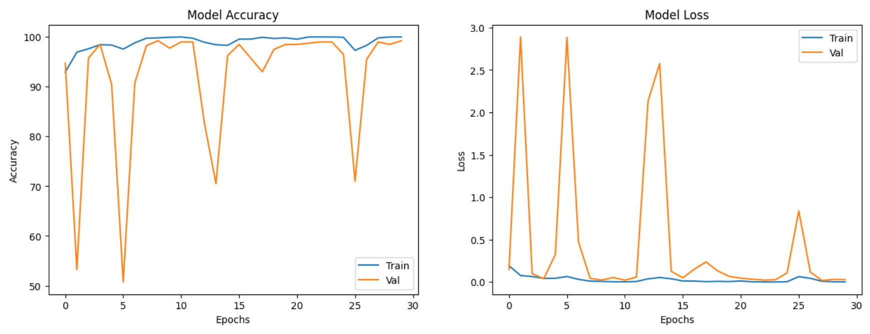
Regarding the number of in_features, ResNet has 512, while GoogleNet has a higher number, which is 1024. As for the number of output features, ResNet has 2, which exactly meets our requirements. However, GoogleNet has 1000 output features, so we need to convert these features into binary features separately. As for the training time, in our dataset, GoogleNet takes slightly more time compared to ResNet.

2. Fine-tuning the ConvNet

The following are the hyperparameters that we fixed constant in our comparison experiments.
Table . Fixed parameters in the comparison.

Parameters	Model	Loss	Epoch	Batch size	Learning rate scheduler
	CNN	BCE	30	32	CosineAnnealingLR

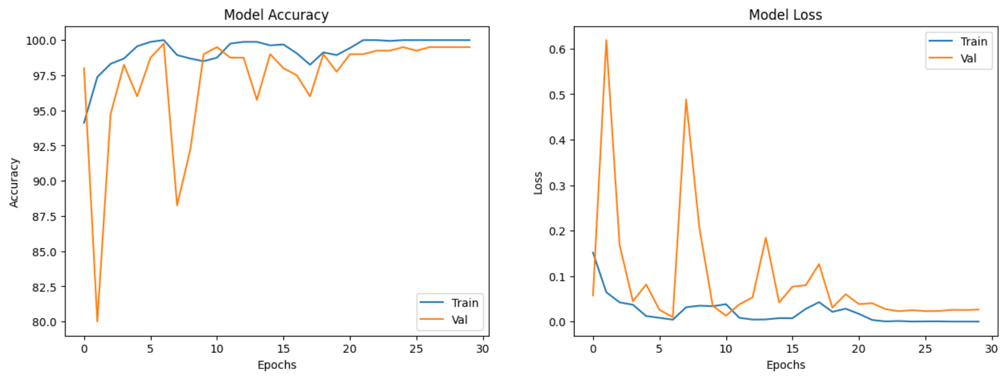
Figure 1 displays the results of fine-tuning ResNet, and (Figure 2) shows the results of GoogleNet.
Performance of ResNet:



(a) Model accuracy evolution (b) Model loss evolution
Figure 1. Training and validation evolutions with ResNet model.

From Figure 1, we observe that the training accuracy of ResNet exhibits a very smooth behavior, converging rapidly. However, the validation accuracy shows significant fluctuations. And the training results shown in Table 1 shows the good performance.

Performance of GoogleNet:



a) Model accuracy evolution (b) Model loss evolution

Figure 2. Training and validation evolutions with ANN model, increase the epoch number.

In Figure 2, compared to ResNet, GoogleNet shows some slight oscillations in training accuracy and takes a bit longer to converge. However, its final performance is on par with ResNet.

Comparing the performance of both models, we can observe that GoogleNet required a longer training time, over one minute. Therefore, for fine-tuning purposes, ResNet may be a preferable choice for this model. The test accuracy will be discussed in final section.

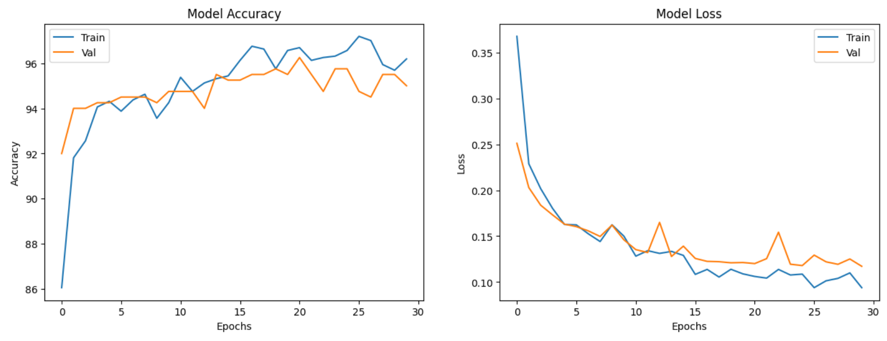
Table 1. Train, validation and test results of ResNet and GoogleNet.

Model	Train Loss	Train Accuracy	Best Val Loss	Best Val Accuracy	Test Accuracy	Training time
ResNet	0.0011	100.00%	0.0178	99.25%	69.25%	03:08
GoogleNet	0.0001	100.00%	0.0100	99.75%	80.75%	04:11

3. ConvNet as Fixed Feature Extractor

Figure 3 and Figure 4 respectively depict the performance of ResNet and GoogleNet when used as a fixed feature extractor with ConvNet.

Performance of ResNet:

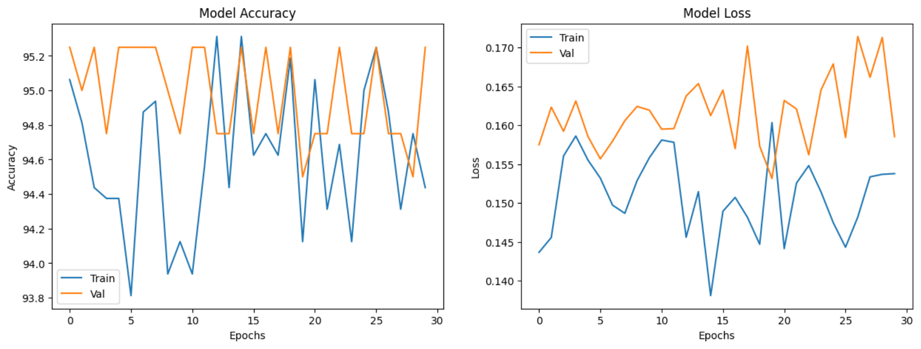


a) Model accuracy evolution (b) Model loss evolution

Figure 3. Fixed feature extractor training and validation evolutions with **ResNet** model.

From Figure 3, it can be observed that both training and validation accuracy show a slower increase compared to the previous two experiments. The loss also decreases gradually with smaller fluctuations. However, from the numerical performance in Table 2, we can observe that the accuracy decreases to approximately 96%. Fortunately, it seems that overfitting did not occur.

Performance of GoogleNet:



a) Model accuracy evolution (b) Model loss evolution

Figure 4. Fixed feature extractor training and validation evolutions with **GoogleNet** model.

From Figure 4, it can be observed that when using GoogleNet, the training process exhibits severe fluctuations, and there doesn't appear to be an upward trend. Increasing the training time may not necessarily improve accuracy. The numerical performance, as shown in Table 2, is slightly below that of ResNet, but both models fail to achieve 100% accuracy.

It's worth mentioning that, in the fixed feature extraction scenario, the training time required is shorter.

Table 6. Train, validation and test results of ResNet and GoogleNet.

Model	Train Loss	Train Accuracy	Best Val Loss	Best Val Accuracy	Test Accuracy	Training time
ResNet	0.0939	96.19%	0.1172	96.25%	84.25%	01:20
GoogleNet	0.1538	94.44%	0.1532	95.25%	84.75%	01:39

4. Comparison and Analysis

Next, we compare the two transfer learning methods mentioned above: 1) fine-tuned and 2) fixed feature. Clearly, when the models are fine-tuned, it takes about two to three times more training time compared to fixed feature. This is possibly because fixed feature doesn't require searching for suitable weights from scratch, making the training process faster. However, in this training, fine-tuned performance outperforms that of fixed feature.

5. Test Dataset Analysis

However, there is still a performance gap of approximately 15% compared to the training and validation accuracy from the above figure, indicating that our test dataset and training dataset might have different distributions. The exact extent of this difference is uncertain. It's possible that switching to another model could lead to a slight improvement.

Additionally, in this experiment, when using ResNet and fine-tuning, the test accuracy is only approximately 70%. We are uncertain about the reasons for this problem.

6. Problems as using Alex model to conduct the fine-turning transform learning

Initially, I chose the AlexNet model for fine-tuning experiments, and the performance at epoch 30 seemed to indicate that it had not converged yet (Figure 5). Therefore, I expected to see better results by increasing the number of epochs to 120. However, the accuracy started to decline dramatically around epoch 36 (Figure 6). Interestingly, this phenomenon did not occur when using other models. This issue remains unresolved.

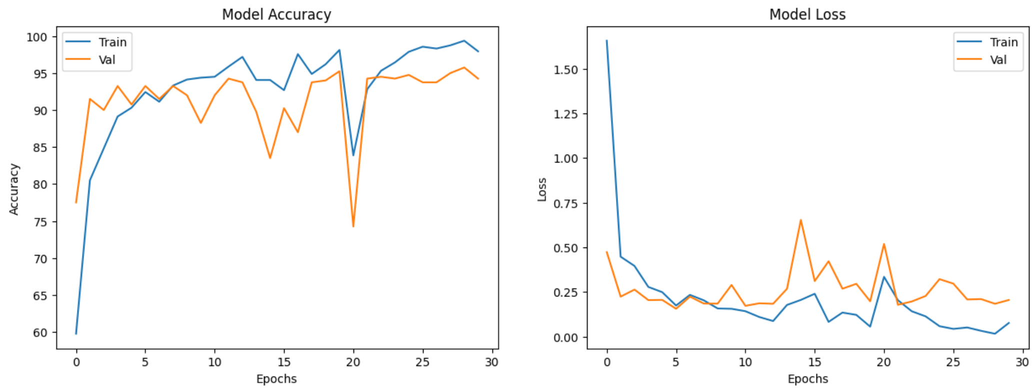


Figure 5. AlexNet model with fine-turning method, epoch = 30.

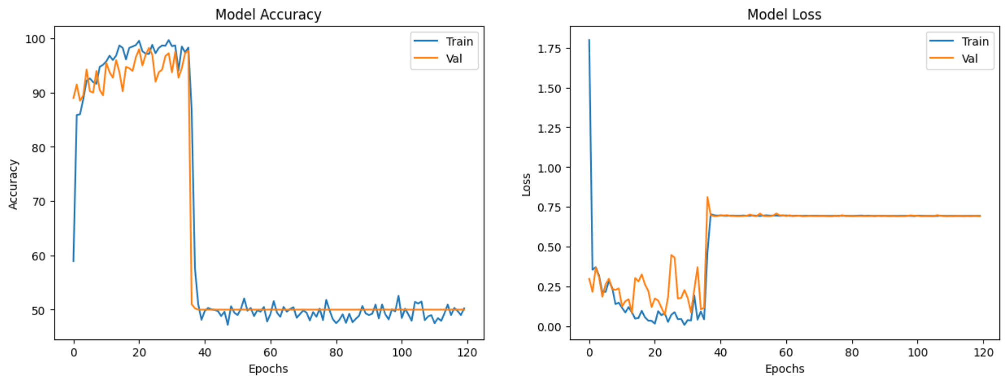


Figure 6. AlexNet model with fine-turning method, epoch = 120.