National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework 3

Name: 吳欣鴻

Student ID: 112066523

Task A: Model Selection

1. *Model Choice*
   (1) ResNet-50
   (2) MobileNetV2

2. *The introduction*

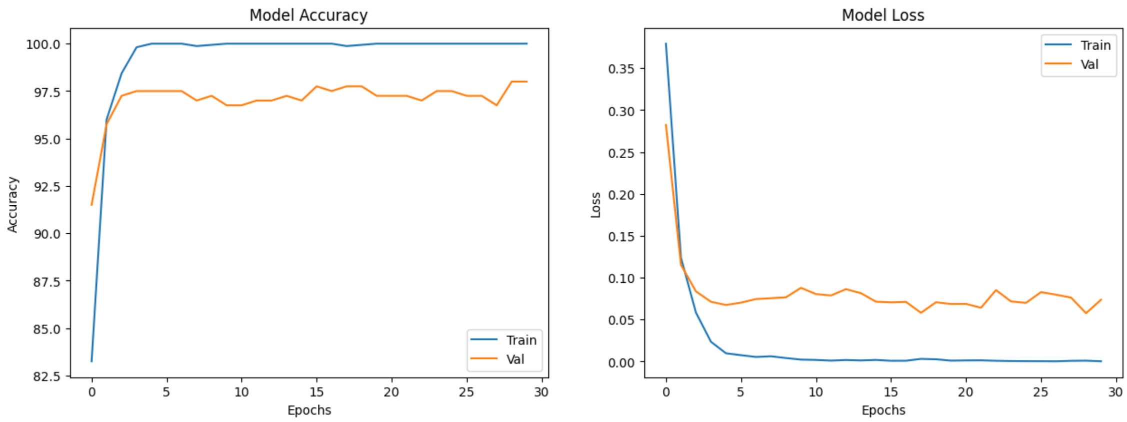
	ResNet-50	**MobileNetV2**
Complexity of Architecture	ResNet-50 is part of the Residual Network family, known for its deep architecture. ResNet-50 has 50 layers, making it a relatively deep model compared to earlier architectures.	MobileNetV2 is designed for mobile and embedded applications. It has a more lightweight architecture compared to models like ResNet-50. It's known for its efficiency.
Performance	ResNet-50 is known for its excellent performance on various computer vision tasks, including image classification.	While MobileNetV2 may not be as powerful as ResNet-50, it is still quite capable and performs well on tasks where computational resources are limited.
Computation Time	The computation time for fine-tuning ResNet-50 depends on factors like the hardware, the size of the dataset, and the specific task. It's a relatively large model, so training can be time-consuming, but transfer learning is faster as we are building upon pre-trained weights.	MobileNetV2 is faster to train and fine-tune compared to larger models like ResNet-50. Its smaller size and architectural design make it a good choice when you have limited computational resources.

Task B: Fine-tuning the ConvNet

1. The introduction

In this approach, we start with a pre-trained ConvNet and continue training it on your dataset. The weights of the network are updated in this process, allowing the model to adapt to the new task. We can choose to fine-tune the entire network or just a subset of layers

2. Performances & Methods



Test accuracy is 82.25%

Here's how we improve and correct code:

(1) Change the ResNet model:
In ResNet models, all convolutional layers apply the same convolutional window of 3 × 3, the number of filters increases following the depth of networks. We can change ResNet model from ResNet18 to ResNet50.

(2) Optimizer Learning Rate:

```
optimizer = optim.Adam(model.parameters(), lr=1e-5)
```

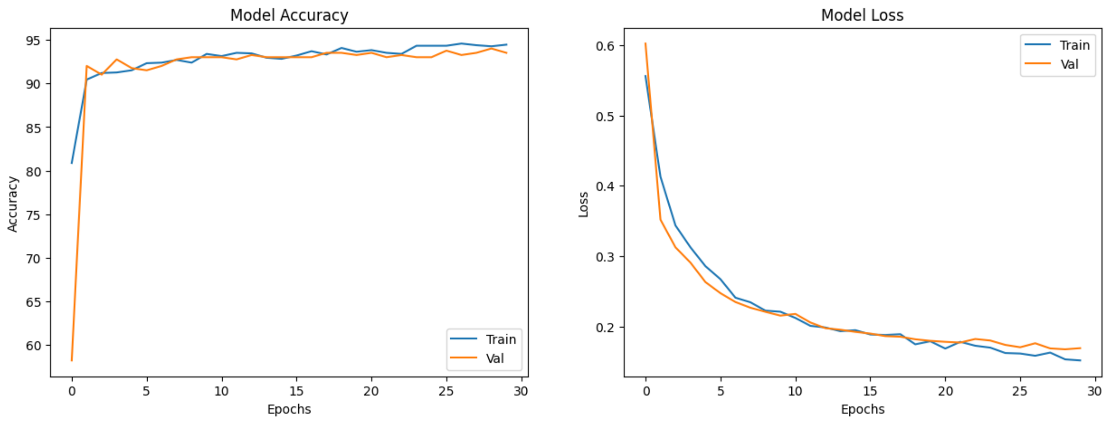
The learning rate of 1e-3 might be high for fine-tuning the pre-trained model. We can try a lower learning rate, such as 1e-5. This can help prevent overfitting when fine-tuning.

Task C: ConvNet as Fixed Feature Extractor

1. The introduction

In this approach, we use the pre-trained ConvNet as a feature extractor and keep its weights fixed. Remove the last fully connected layer and replace it with a new one that matches the number of output classes in your task.

2. *Performances & Methods*



Test accuracy is 85.0%

In short, compared with Fine-tuning the ConvNet, ConvNet as Fixed Feature Extractor is more able to achieve approximately the same in training and validation accuracy.

Optimizer Learning Rate:

```
optimizer = optim.Adam(model.parameters(), lr=1e-4)
```

Compared with Fine-tuning the ConvNet, the learning rate of ConvNet as Fixed Feature Extractor can be adjusted higher. Here I adjust it to 1e-5.

Task D: Comparison and Analysis

	Fine-tuning the ConvNet	**ConvNet as Fixed Feature Extractor**
Advantages	(1) The model can learn task-specific features, leading to potentially better performance. (2) Can work well with smaller datasets since the model is already pre-trained and only needs to learn the differences between the new task and the pre-training task.	(1) Computationally efficient, as you only need to train the new fully connected layer. (2) Reduces the risk of overfitting, as the pre-trained features remain fixed and only the final classification layer is updated.
Disadvantages	(1) Can be computationally expensive, as you need to update the weights of the network during training.	(1) The model cannot learn task-specific features since the weights of the ConvNet remain fixed.

	(2) There is a risk of overfitting if the new dataset is small and not diverse enough.	(2) May result in lower performance if the pre-trained features are not suitable for the new task.

Task E: Test Dataset Analysis

1. *Performance:*

   ConvNet as Fixed Feature Extractor finds it easier to fit the train and validation accuracy than Fine-tuning the ConvNet, and the final test accuracy is also relatively high.

   (Test accuracy is 85.0% v.s. 82.25%)

2. *Several reasons why the test accuracy may not be as high as expected include:*

   (1) Overfitting:

   The model might have overfit the training data, meaning it has learned to perform well on the training data but doesn't generalize well to unseen data.

   (2) Hyperparameter Tuning:

   The choice of hyperparameters (learning rate, batch size, number of epochs......) can significantly impact model performance. Suboptimal hyperparameters may result in poor test accuracy.

   (3) Model Architecture:

   The selected model architecture might not be well-suited for the task at hand. We might need a more complex or more suitable architecture.

   (4) Training Data Size:

   If the training dataset is too small, the model may not have seen enough examples to learn the underlying patterns effectively.