National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework 4

YOUHAN-WEN[1]

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

Student ID:110066554

1. Task A: Model Selection

1.1 GoogLeNet

GoogLeNet was first published by Google, which presents fewer parameters but is deeper and more accurate than AlexNet in the year 2014. At the time we couldn't figure out the right timing to use the Max-pooling and the Convolution, so GoogLeNet instead chose the right one but using all of it, GoogLeNet took different sizes of the kernel of the Convolution and Max-pooling concatenate into output, this so calls Inception model, and GoogLeNet is built by several Inception models.

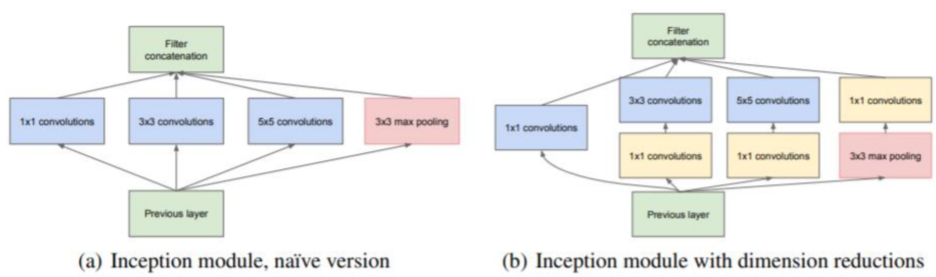

Fig. 1. (a)Original inception module. (b)To downsize the training parameters we stack a 1x1 convolution layer, which can decrease the output value.

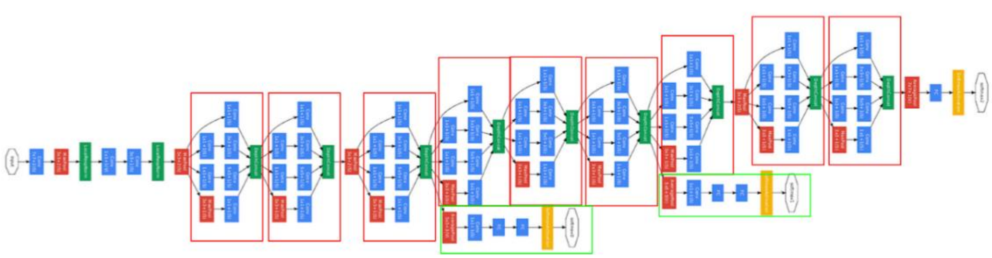

Fig. 2. GoogLeNet diagram with nine inception modules includes, the process direction from left to right, the red frame is the inception module and the green frame is the auxiliary classifiers.

GoogLeNet is a deep-learning neural network, that will face the gradient to zero problem, an auxiliary classifier is to solve this problem which enhances the stability and speeds up the convergence. The ways auxiliary classifier does this to calculate the loss of different layers of the inception model, multiply with the weight 0.3, and add the inception model loss and the real loss together to get the final total loss.

1.2 ResNet

ResNet was first proposed by Microsoft in 2015, the same year the competition ResNet used 152 layers of the neural network, with even deeper compared with GoogLeNet, and also has an accuracy of up to 96.4% which defeats human accuracy of 1.5% higher.

How ResNet has this astonishing performance, the main contribution is to propose Residual Learning, which starts the new era of the deep learning neural network and makes it easier to train. In the past, with more layers increased the performance became better, but what if we stack even more layers, the performance now will be disappointing, in fig 3 we can see the performance of 56 layers is worse than 20 layers, it isn't the problem of overfitting, but the degradation of the neural network.

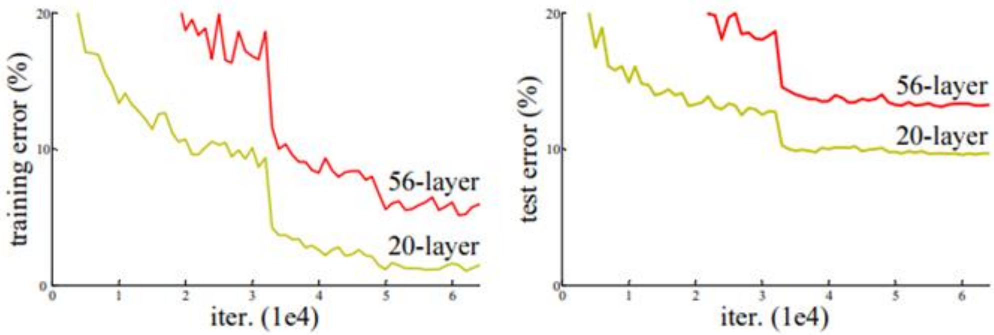

Fig. 3. This figure shows the training error and the test error with more than six iterations, we can see that with more layers stacks the performance doesn't become better but decreases.

What degradation is, that when the layer's values become larger gradient will vanish in the backpropagation process, which leads to the gradient value not being updated with the correct parameters so that the more layers neural network module has the worse train performance.

The characteristic of ResNet is using the shortcut connection structure, from fig 4 we have two branches, one is the transfer value X with cross-layers, and another one is F(x). These two branches are added together, and input to the active function, which so call residual learning that can solve the model degradation problem.

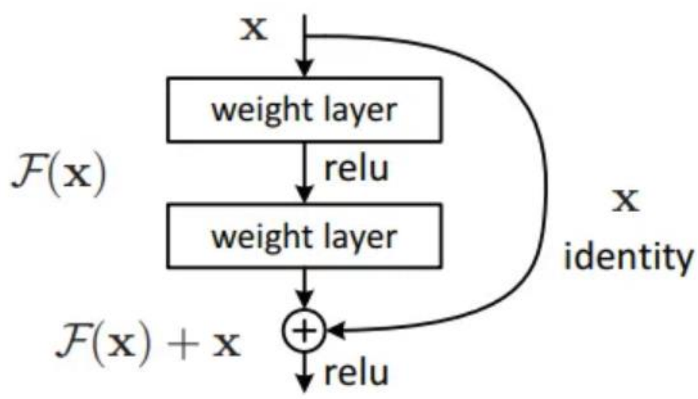

Fig. 4 Residual learning block.

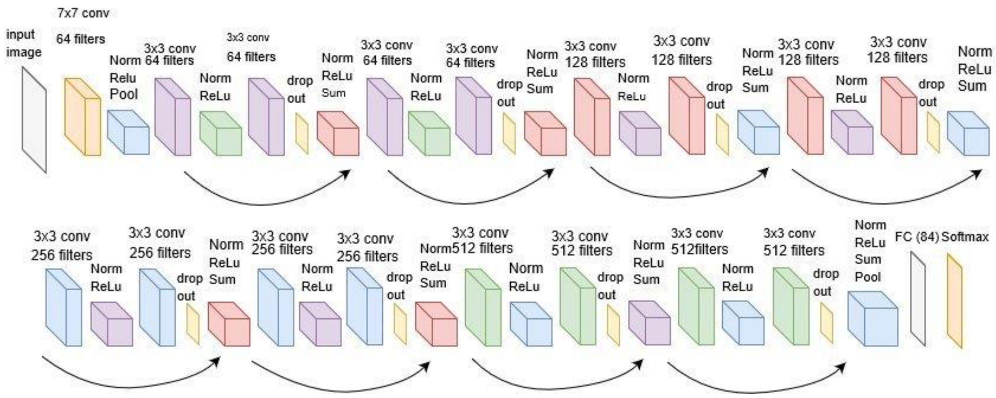

Fig. 5. ResNet-18 diagram, in the first layer of convolution, is a 7X7 matrix connected with a 3X3 max-pooling, stack with multi residual block, and finally use Global Average Pooling and pass it into the fully connected layer for classification. The solid line means the shortcut connection.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 6. Different layers of ResNet network architecture, we can see when the layer is over 50 layers we will input a 1X1 convolution to downsize the calculation.

2. Task B: Fine-tuning the ConvNet

2.1 GoogLeNet

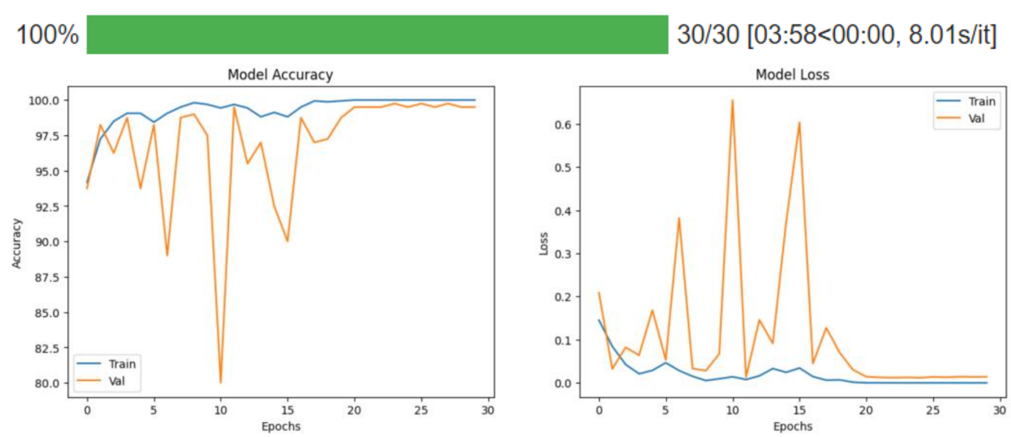

Fig. 7. GoogLeNet model accuracy and loss with 30 epochs.

This transfer learning took me almost four minutes, and from Fig. 7 it doesn’t overfit the training accuracy reaches 100%, the best validation accuracy is 99.75%, and the test accuracy is 80.75%, which benefits from GoogLeNet high accuracy with fewer parameters so it can achieve high accuracy and less time of training,

2.2 ResNet-50

This module has 48 layers of convolution layers and 2 layers of fully connected layers, the training accuracy reaches 100%, the best validation accuracy is 99%, and the test accuracy is 81.75%, with the higher accuracy it took 10 minutes to train.

ResNet-50's key advantage lies in its ability to train very deep networks effectively, leading to high accuracy and generalization in a wide range of computer vision tasks.

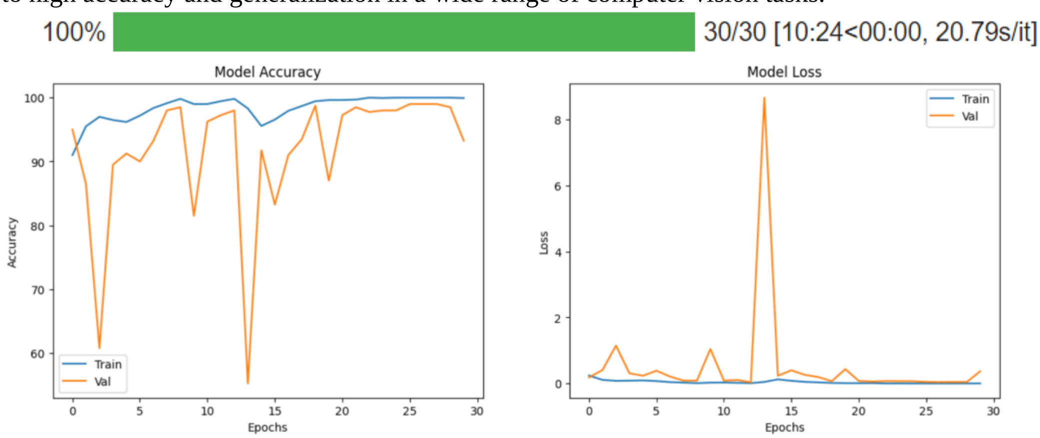

Fig. 8. ResNet-50 model accuracy and loss with 30 epochs.

3. Task C: ConvNet as Fixed Feature Extractor

Freezing the parameters in a neural network means that you prevent those parameters from being updated during the training process. In other words, the weights and biases of the frozen layers remain fixed, and only the unfrozen layers (typically the top layers) are updated during training. This can be particularly useful when you have limited data for your target task or when you want to speed up training and save computational resources.

3.1 GoogLeNet

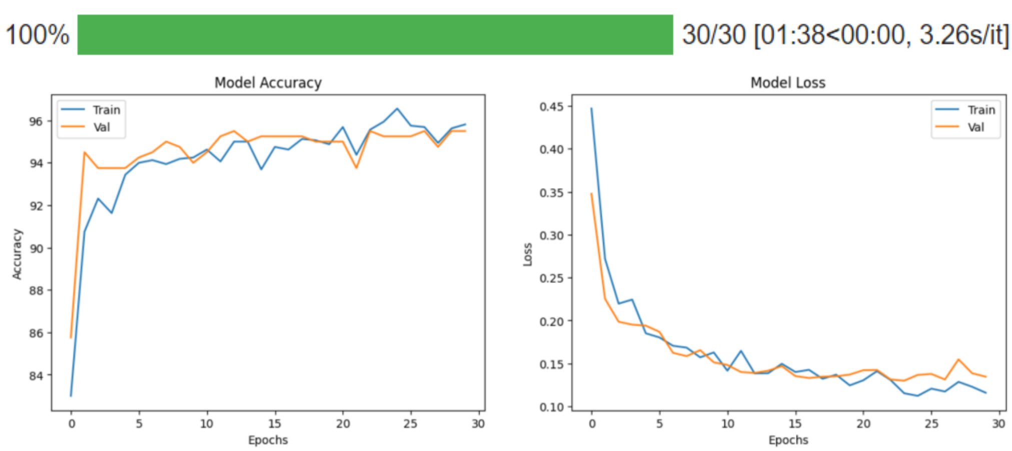

Fig. 9. GoogLeNet model accuracy and loss with 30 epochs with fixed parameters.

From Fig.9 we can see the training time of the module reduced to one and a half minutes, and even higher test accuracy to 89.25%, the training accuracy reaches 95.81%, and the best validation accuracy is 95.5%.

I think this effort leads to pre-trained model having already learned useful features from a large dataset. By freezing the lower layers that contain these pre-trained features, the model retains the knowledge of basic patterns, textures, and shapes. This knowledge can be beneficial for model specific tasks because the lower layers capture general, reusable information.

3.2 ResNet-50

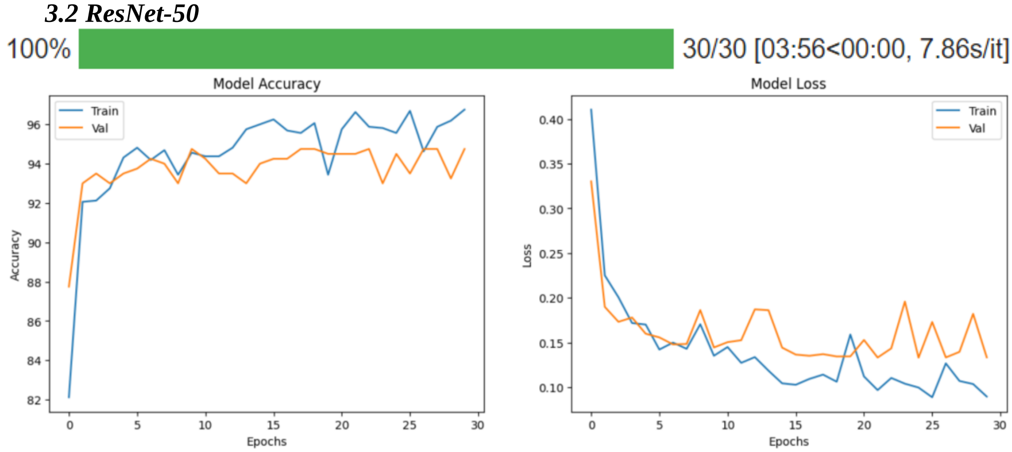

Fig. 10. ResNet-50 model accuracy and loss with 30 epochs with fixed parameters.

From Fig.10 we can see the training time of the module reduced reduce two and a half times, and test accuracy to 84.75%, the training accuracy reaches 96.75%, and the best validation accuracy is 94.75%.

I use the chatgpt tryto understand why the performance isn’t that good compared with GoogLeNet,that the pre-training task and your target task are substantially different, and the features learned by the pre-trained ResNet may not be directly applicable. In this case, the pre-trained model's knowledge might not provide a significant advantage, and fine-tuning only the top layers may not be sufficient to adapt the model to the new task.

4. Task D: Comparison and Analysis

- Quadrant 1 — large data size but has a small data similarity. In this case, better if we develop the model from scratch
- Quadrant 2 — large data size but has a high data similarity. In this case, we should consider training some layers in the feature extraction layer while the others are frozen. The number of layers is debatable, it depends on the needs
- Quadrant 3 — small data size but has a small data similarity. Similar to the quadrant 2 scenario
- Quadrant 4 — small data size but has a high data similarity. In this case, we can implement the fixed feature extractor method for transfer learning

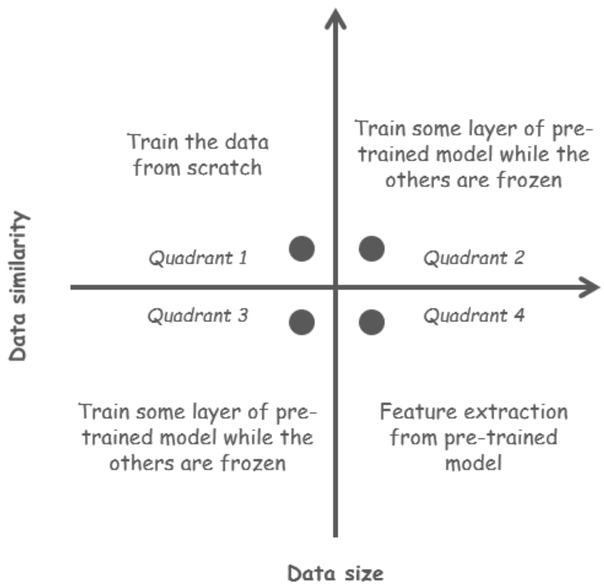

Fig. 11. Transfer learning quadrant

Table 1. Optical Constants of Thin Films of Materials

Pros of Fixed Parameters	**Cons of Fixed Parameters**
Faster Training	**Limited Adaptability**
Pre-trained Knowledge	**Dissimilarity Between Tasks**
Reduction in Overfitting	**Data Mismatch**
Efficient Use of Resources	**Inadequate Data**
Stability in Features	**Underfitting**

5. Task E: Test Dataset Analysis

From the model of all reports, I found out that our training model will have high training and validation accuracy, but the test accuracy couldn't reach as high as train accuracy, there are a few reasons I guess. First, when the validation and the test data swap the result is still the same, so I think that this validation data is quite different from the test data, so the accuracy can't perform well. Second, perhaps the dataset wasn't large enough, we have 1600 images for training and 400 images for validation so a large dataset is needed.