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

SHIU-FENG CHENG[1]

[1] *Brain Research Center, National Tsing Hua University, Hsinchu 30013, Taiwan.*

Student ID:111080527

1. Introduction

Existing pre-trained models are convenient tools to use. In this report I applied a modified Resnet34 convolution neural network for image classification of 6 class of biological samples.

2. Result

2.1 The structure of Resnet34

The models I adopted is Resnet34 which is based on the Deep Residual Learning for Image Recognition paper [1]. The Resnet models are composed of 4 convolution layers which are called "basic blocks". A basic block of Resnet is built by two sets of 3*3 convolutions followed by 2D batch normalization and ReLU activation function additionally there is a shortcut that forwards (additive) the basic block input to the front of the last ReLU activation function (Fig.1a). The advantage of adding a shortcut to introduce residual term is to reserves the information of early layers. The difference between Resnet models (such as Resnet18, Resnet34, Resnet50) is the number of basic blocks in each layer, the model I used which is Resnet34 has respectively 3, 4, 6, and 3 basic blocks for layer 1, 2, 3, 4. The number 34 indicates the number of learnable layers which are the convolution layers and the output Linear layer.

The default setting of Resnet34 have output layer which gives 2 output values for bilinear tasks; however, our dataset has 6 classes of sample thus the dimension of the layer is adjusted to 6, in order to match our requirement. An increase in performance is expected while having more inputs to the output layer. I also did an experiment on the model to test how different adaptive average pooling output dimensions (1*1, 2*2, 4*4, 8*8) affects the network performance. The loss function I applied is cross entropy loss and the optimizer is Adam optimizer. The learning rate while training varies for every epoch which follows the setting of CosineAnnealingLR. CosineAnnealingLR varies the learning rate follows a cosine function. The amplitude of the cosine function in my work is set to 10^-3 and the whole training process go through half a cycle of cosine (the learn rate continuously approaches 0 while training).

2.2 image dataset

The sample includes 6 classes of sample which are tumor, stroma, complex, lympho, debris, and mucosa with Hematoxylin and Eosin (H&E) staining treatment. The format is 150*150 pixels RGB image and was precropped into 128*128 for the convenience of dimension calculation.

2.3 Performance of Resnet34

The performance of Resnet34 is shown in Fig.2 the pool n in the legend represents the output of adaptive average pooling gives n*n outputs for every input kernel. The training process for our dataset (6 classes, 425 images for each class) took 791.806, 791.584, 791.479, 787.027 secs for pool 1, 2, 4, 8 groups. Although the difference is minor, it seems that the averaging pooling a wider area (smaller pooling output) computing takes more time compared to linear computation and weight update with more input dimensions.

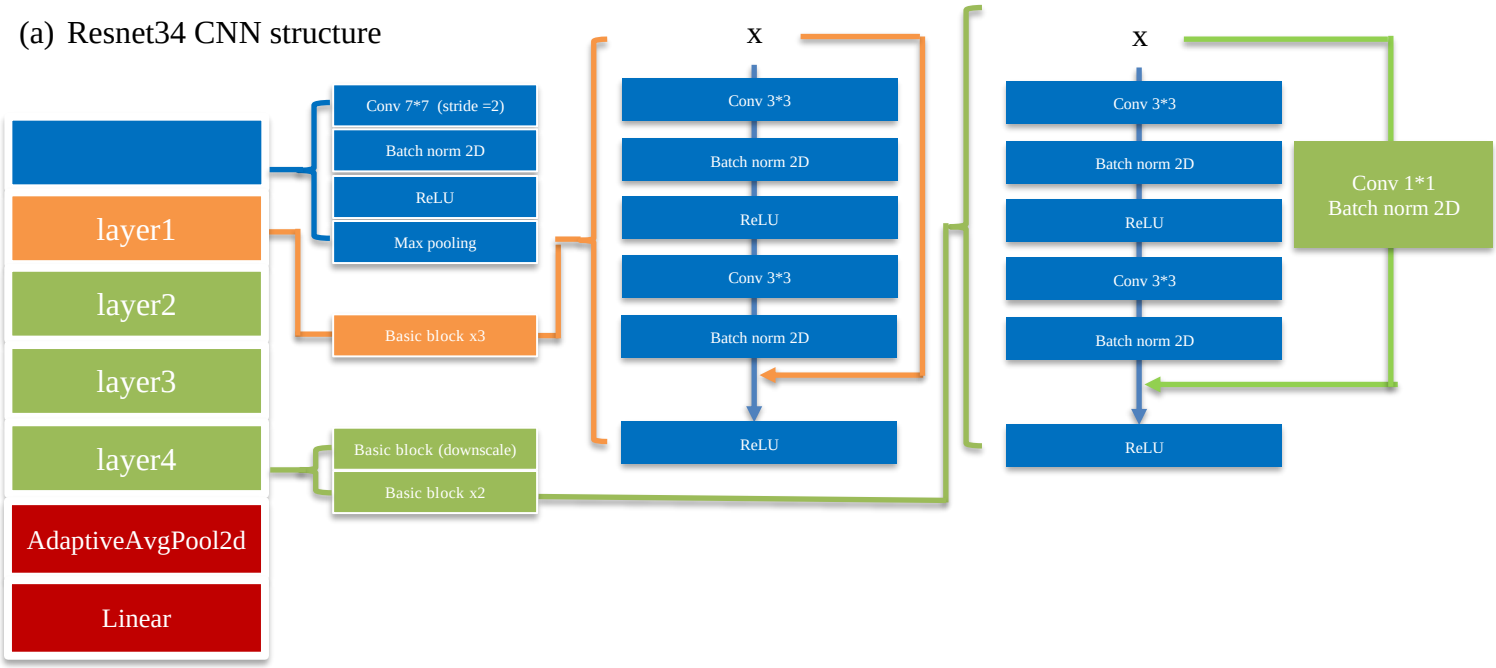

Fig. 1. The Resnet34 model structure. Resnet models shares same basic block structures but having different numbers of building block for each layer.

From the result shown in Fig.2 We can find that the Resnet34 model gives low total loss and near 100% training accuracies for each group after around 50 epochs. However, the performance on the validation is not that ideal, the accuracies caps in the range of 85%~90%. While more pooling output gives lower accuracies. Similarly, the validation loss is much higher (~0.6) than training loss (~0.002) and more pooling output tends to give higher loss.

Since more inputs to output layer lowers the performance, it is valid to guess the model is overfitted. The actual input dimension of the output linear layer is 512*n*n which n is the pooling output dimension. Thus n= 1, 2, 4, 8 respectively gives 512, 2048, 8192, 32768 inputs to the output layers which indicates the output layer has 518, 2054, 8198, 32774 variables to fit the training dataset (to be notice that there are also more variables in convolution layers). Since we only get 2550 training images also no image augmentation is applied, the number of variables for n>1 groups are sufficient to fit a perfect curve for each training image which indicates the model is overfitted. This problem might be fixed by either adding image augmentation, or simply adding more training datasets.

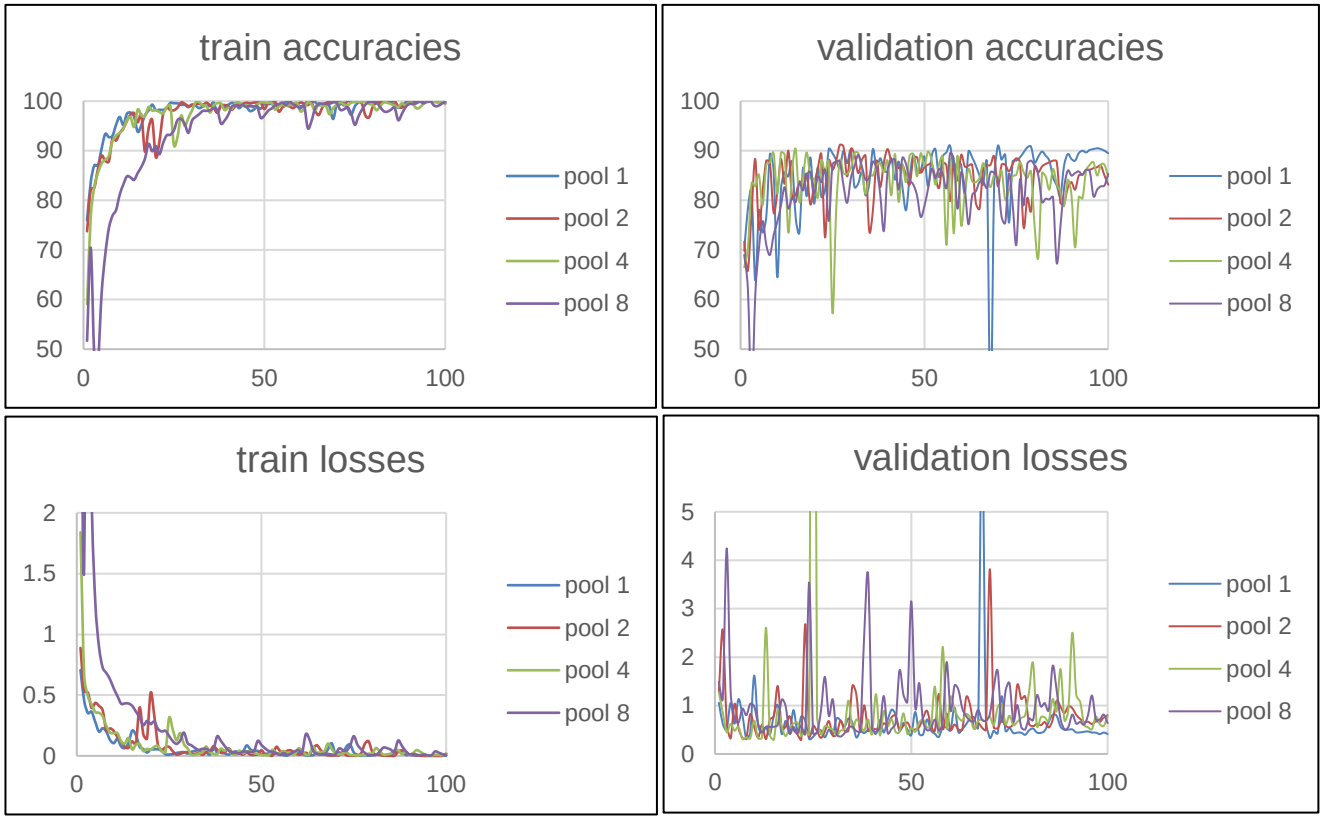

Fig. 2. The performance of the Resnet34 model with different adaptive pooling output size. pool n in the legend indicates the output dimension of adaptive pooling is n*n.

3. Material and Method

3.1 Resnet34 model code

The network is modified from Resnet34 to match the demanding output dimension.
The code can be found at:
https://github.com/Frank497/NTHU_2023_DLBOI_HW/blob/main/report/report_resnet34_with_dropout.ipynb

4. References

1. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.