Deep Learning Practice Lab 3

0856056 Yu-Wun Tseng

April 2020

1 Introduction

This goal of this lab is to analysis diabetic retinopathy. We need to implement our own Dataloader and use the pre-trained models, and finally print the confusion matrix. The pre-trained models are ResNet18 and ResNet50, in addition, we need to compare the performance of the models with or without pre-training.

2 Experiment Setups

The lab is done by using Pytorch and sklearn. I'll explain the setup of models and Dataloader and the detail of confusion matrix in this section.

2.1 ResNet

In theory, with the deepening of the neuron network, the model will get better performance. However, the fact is that the accuracy gets saturated and degrades rapidly. This is the problem of gradient vanishing, which is due to the zero gradients. Therefore, the network can't converge to the optimized parameters. ResNet [1] solves the problem by adding a shortcut connection to add the input to the output, as Figure 1 shows. If the gradient of this layer is zero when backpropagating, the model can pass the gradient of the previous layer to avoid passing zero gradient.

I use the models provided by torchvision library and load the ResNet18 model and ResNet50 model. Each model has two types, including with and without pre-training. The epochs of ResNet18 and ResNet50 are 12 and 5, and the batch size are 8 and 4. The learning rate of both models is 1e-3 and the loss function is cross entropy. The optimizer of both models is SGD, but the weight decay of ResNet18 is 1e-3 and the other model is 5e-4.

2.2 Dataloader

I use the Dalaloader and transform package provided by Pytorch. I read data using PIL package instead of skimage in order to applying transform functions more easily. I overwrite the ___getitem___ methods to return a processed image tensor. For ResNet18, I load image and rotate it by 30 degrees to reduce overfitting through data augmentation, and shuffle it when using Dataloader. Then convert the numpy array into tensor. For ResNet50, the rotation is skipped.

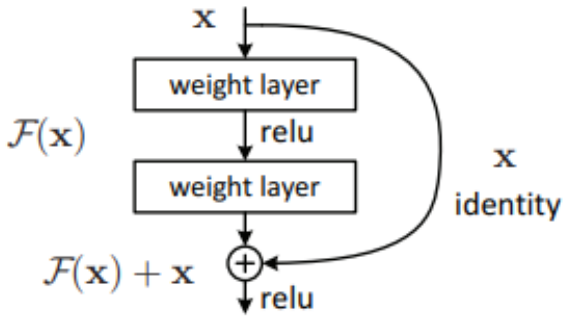
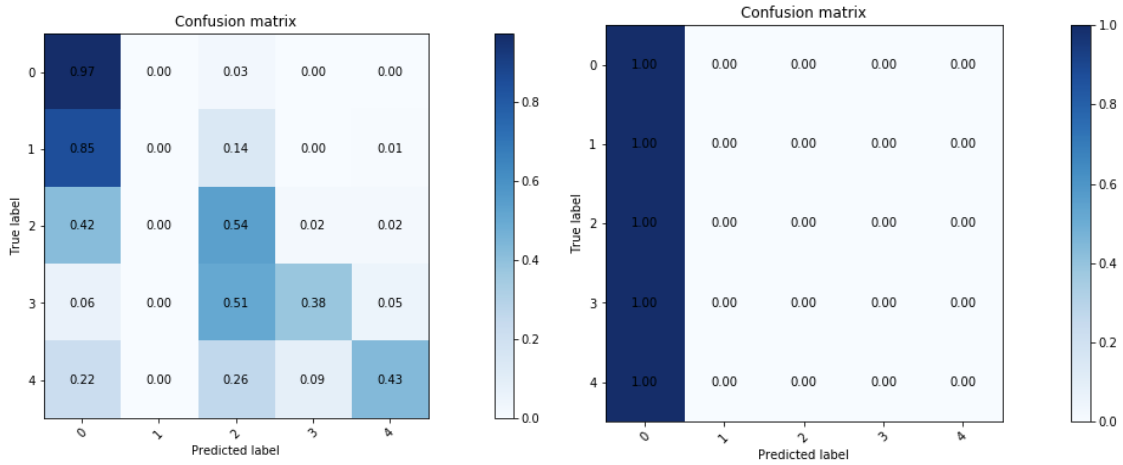


Figure 1: Residual Block



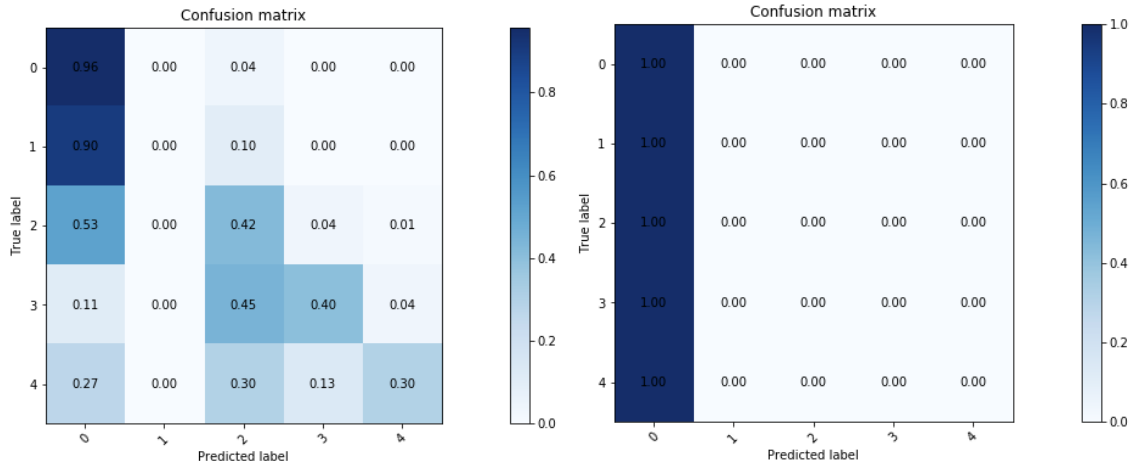
Accuracy of resnet18_ft_shuffle: 0.8138078291814946

(a) With pre-training

Accuracy of resnet18: 0.7335231316725979

(b) Without pre-training

Figure 2: ResNet18 confusion matrix



Accuracy of resnet50_ft: 0.7814946619217081

(a) With pre-training

Accuracy of resnet50: 0.7335231316725979

(b) Without pre-training

Figure 3: ResNet50 confusion matrix

2.3 Evaluation

The normalized confusion matrix of each models is shown on Figure 2 and 3. As shown in the figures, the prediction of the models without pre-training is poor, all data are classified as class 0. The dataset is biased, as a result, the accuracy is not very poor. The diagonal is the recall, and the other values are type two error. It's the proportion predicted to be false when the truth is true, so called false negative. The matrix of ResNet18 with pre-training is similar to ResNet50 with pre-training, the recall of class 0 is close to 100%, while the recall of class 1 is close to 0%. The recalls of other three classes are not high, most of them do not exceed 50%.

```
Epoch:  0, train accuracy: 0.7799565820847717, test accuracy: 0.7755160142348755 - time: 11.941605087121328
Epoch:  1, train accuracy: 0.7957934446065696, test accuracy: 0.7877580071174377 - time: 12.318785818417867
Epoch:  2, train accuracy: 0.7983202249190363, test accuracy: 0.7920284697508897 - time: 12.027101476987204
Epoch:  3, train accuracy: 0.8133029645183103, test accuracy: 0.8012811387900356 - time: 12.014646983146667
Epoch:  4, train accuracy: 0.8159009217409873, test accuracy: 0.8076868327402136 - time: 12.018304550647736
Epoch:  5, train accuracy: 0.8298160076871063, test accuracy: 0.8103914590747331 - time: 12.031717197100322
Epoch:  6, train accuracy: 0.824335385600911, test accuracy: 0.8095373665480428 - time: 12.02246514558792
Epoch:  7, train accuracy: 0.8339798569344105, test accuracy: 0.8132384341637011 - time: 12.055637578169504
Epoch:  8, train accuracy: 0.8279654080216378, test accuracy: 0.8130960854092527 - time: 12.324069507916768
Epoch:  9, train accuracy: 0.8414890209616, test accuracy: 0.802846975088968 - time: 12.034594190120696
Epoch: 10, train accuracy: 0.8453681625680629, test accuracy: 0.8132384341637011 - time: 12.028079156080882
Epoch: 11, train accuracy: 0.8424499092494395, test accuracy: 0.8125266903914591 - time: 12.016655627886454
The highest accuracy of resnet18_ft model is 0.8132384341637011
```

Figure 4: ResNet18 training result

3 Results

3.1 The highest testing accuracy

The model with the highest testing accuracy is ResNet18 with pre-training, which the accuracy is 0.813. The training result is on Figure 4.

3.2 Comparison figures

The accuracy trend comparison of each model is on Figure 5 and Figure 6. The accuracy trend of the models without pre-training has no increasing. The accuracy trend of ResNet18 with pre-training has raised stably, while the ResNet50 with pre-training is overfitting.

4 Discussion

First, I ran the pre-trained ResNet18 with the hyperparameters TA provided, the accuracy achieved only 0.79. But my classmate used the same hyperparameters and achieved an accuracy of over 0.8, additionally, the running time of the model is about half of mine. It's really frustrating. Anyway, I found that the model was overfitting, thus, I increasing the weight decay from 5e-4 to 1e-3. The result was a little better. Then I added rotation into data processing procedure, the accuracy achieved 0.8. After I added shuffle to Dataloader, the accuracy achieved 0.813. In addition, I found that more epochs didn't improve the performance of the models. The pre-trained models get good performance after a few epochs. However, no matter how many the epochs there are, the accuracy of the models without pre-training will no longer improve.

References

[1] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385, 2015.

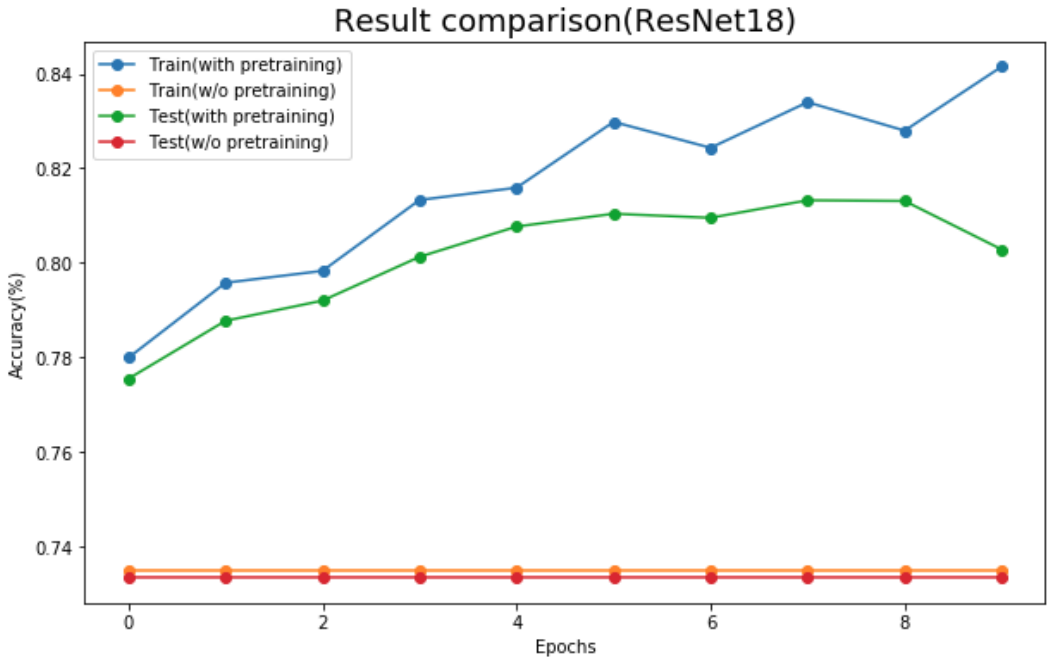


Figure 5: ResNet18 accuracy trend

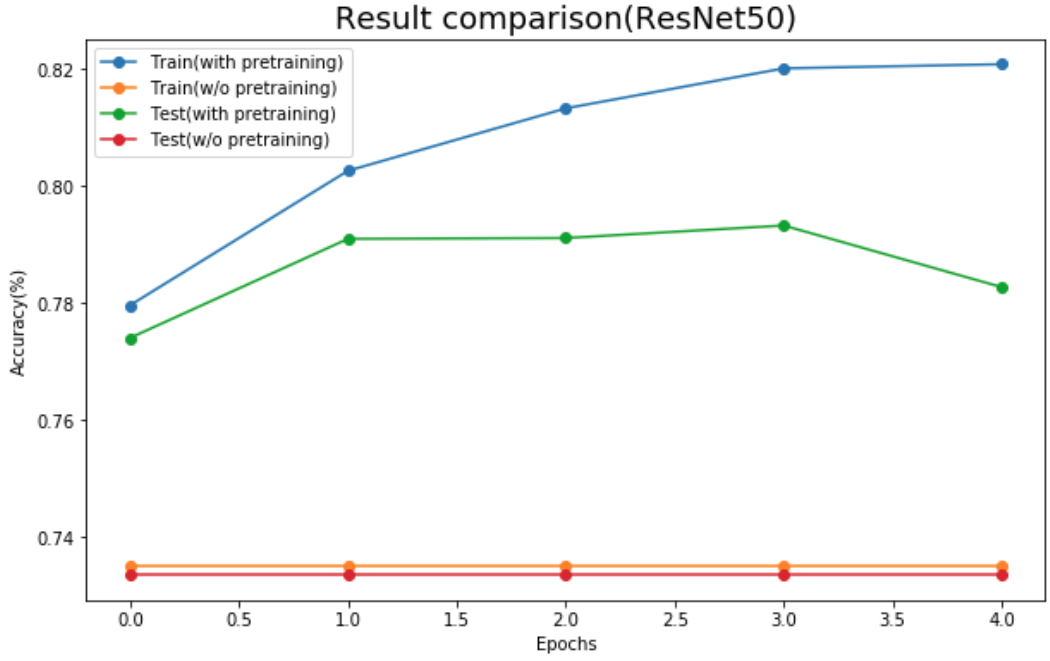


Figure 6: ResNet50 accuracy trend