Lab1 - Deep Residual Learning

Wen-Jie Tseng (0556146)
Deep Learning and Practice - 2018 Spring Semester

1. INTRODUCTION

In this Lab, we implement *Residual Network (ResNet)* and train it on the CIFAR-10 dataset. As shown in Figure 1, training CIFAR-10 with plain convolution neural network (CNN) does not reduce train error and test error as the layer increased. When the depth of network increasing, accuracy gets saturated and then degrades rapidly, which is called the vanishing gradient problem. ResNet proposed the concept of residual block, a shortcut connection to fit the residual of input. In Figure 2, one has to learn $g(.)$ in $H(x) = g(x)$ in plain layer and learn $F(.)$ in $H(x) = F(x)+x$, which is easier to learn. Assume $x = 2.9$, after two conv layers, $H_1(x) = 3.0, F_1(x) = 0.1$. Then after another two conv layers, $H_2(x) = 3.1, F_2(x) = 0.2$. For plain layer, $delta = (3.1 - 3.0)/3.0 = 3.3\%$. However, the residual block obtained $delta = (0.2 - 0.1)/0.1 = 100\%$. Removing x brings efficiency in the ratio changing. We then start building ResNet 20, 56, and 110 layers compared to vanilla CNN 20, 56, and 110 layers.

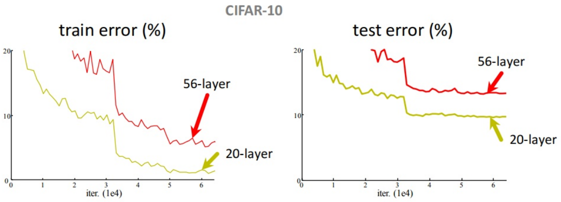

Figure 1. More layers but more error.

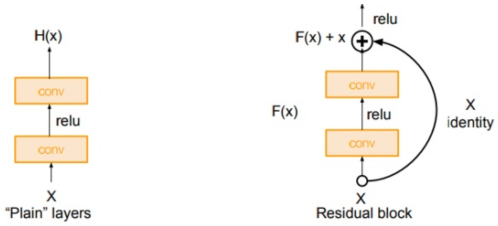

Figure 2. Residual block.

2. EXPERIMENT SETUP

2.1 The Detail of Model

First, 32x32 with inplane 16 input will be send to 1 layer conv layer. The number of residual blocks in the following three conv layer was determined by $totalDepth = 1(conv) + 6n + 1(linearlayer)$. Thus ResNet 20, 56, and 110 need 3, 9, and 18 number of residual blocks in their conv layer respectively. A detailed model is shown in Figure 3. For ResNet 56 and 110, just increase the residual blocks in three yellow layer. Note that when the total layer exceed 50, one has to change basic block into bottleneck block, to save the number of parameters. For vanilla CNN 20, 56, and 110, I removed the shortcuts in source code.

2.2 Hyper-Parameters

- Data preprocessing: Normalize each color channel (compute from entire CIFAR-10 training set) as follows, mean [R, G, B] = [0.4914, 0.4824, 0.4467] and standard deviation [R, G, B] = [0.2471, 0.2435, 0.2616].
- Method: SGD with momentum
- Mini-batch size: 128, leads to 391 iterations for each epoch in training.
- Total epochs: 164
- Momentum: 0.9
- Initial learning rate: 0.1, use optim.lr_scheduler.MultiStepLR(optimizer, milestones=[81, 122], gamma=0.1) to divide LR rate by 10 at 81 and 122.
- Weight decay = 0.0001

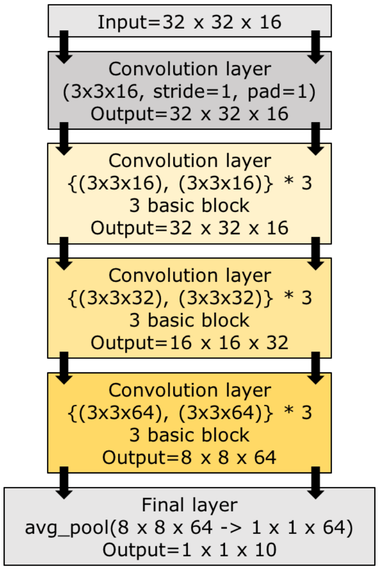

Figure 3. A Resnet 20 layer model.

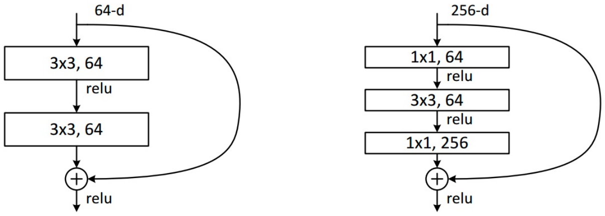

Figure 4. Basic block and bottleneck block

- Weight initialization: apply nn.init.kaiming_normal(m.weight) to initialize each convolution layer and linear layer.
- Loss function: cross-entropy

3. RESULT

The CIFAR-10 data set was trained and tested by 6 models with the Lab1 requirements. The final test error, curve of training loss, and curve of testing error rate of each model shows in Table 1, Figure 5, and Figure 6.

3.1 Final Test Error

As the layer increased, ResNet's final test error drops from 8.51, 6.20, to 5.68. However, vanilla CNN's final test error increased in 20 and 56 layers. The vanilla CNN 110 layers model did not learn from training, thus the testing error did not decrease.

Models	Layers	Final test error (%)
ResNet	20	8.51
	56	6.20
	110	5.68
Vanilla CNN	20	9.66
	56	15.05
	110	90.00

Table 1. Final test error rate

3.2 Training Loss Curve

As the layer increased, the training loss of ResNet shifting downward and the vanilla CNN shifting upward. A clear training loss drops at epoch 81 and 122 due to the update of learning rate.

3.3 Test Error Curve

The test error curve shows similar results as previous section. As the layer increased, the testing error rate of ResNet shifting downward and the vanilla CNN shifting upward. A clear training loss drops at epoch 81 and 122 due to the update of learning rate.

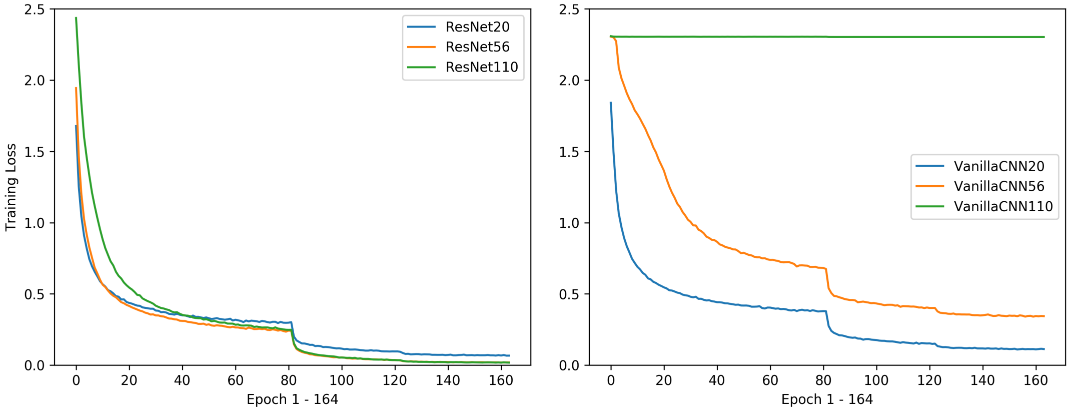

Figure 5. Training loss of ResNet 20, 56, 110 layers (Left) and vanilla CNN 20, 56, 110 layers (Right).

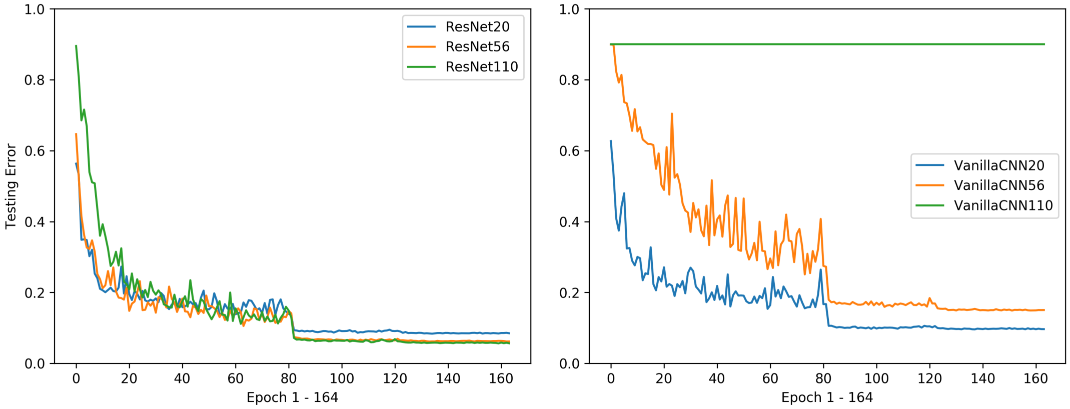

Figure 6. Testing error rate of ResNet 20, 56, 110 layers (Left) and vanilla CNN 20, 56, 110 layers (Right).

4. DISCUSSION

I think ResNet will be a nice option when we need to train data with high layers. To compare ResNet-20 and vanilla CNN-20, the final test error was in small difference. As the layer increasing, ResNet outperforms plain CNN.

It took me several times to make sure that inplanes numbers and layer dimensions were correct for ResNet-20. After ResNet-20 was confirmed, the other 5 models can be done easily, all one needs is time to complete training.

REFERENCES

1. Sample code provided in Lab1.
2. An intro to ResNet.
3. ResNet20 with Caffe2.