COSE474 Deep Learning ASSIGNMENT2 REPORT

MuYeong Jeong
Korea University
2019320112
jmy3033@naver.com

Abstract

In This assignment is about implementing a ResNet model to classify CIFAR-10 data. The task is to take image data of size 32x32x3 as input and classify it into 10 different classes.

ResNet uses residual blocks. A residual block consists of multiple convolutional layers and a shortcut connection. The shortcut connection directly connects the input to the output, helping to solve the gradient vanishing problem

1. Implementation

I will explain the implementation process step by step.

1.1. Residual block

$$\left\lfloor \frac{N+2*P-F}{S} \right\rfloor + 1 = O \qquad (1)$$

In the implementation of the residual block, a bottleneck building block structure was used, consisting of a 1×1 convolution layer followed by a 3×3 convolution layer and another 1×1 convolution layer. Two types of blocks were considered: one that requires down sampling and one that does not.

Down sampling is necessary when the output feature map needs to have half the width and height of the input. This was achieved by setting the stride to 2 and the padding to 0 in the first 1×1 convolution layer of the block. In cases where down sampling was not required, the first 1×1 convolution layer was configured with a stride of 1 and padding of 0, thereby preserving the original input image size. In this layer, the input and output channel arguments were set to in_channels and middle_channels, respectively.

The following 3×3 convolution layer was also designed to maintain the spatial dimensions of the image. This was achieved by setting the stride to 1 and padding to 1. Both the input and output channel arguments for this layer were set to middle_channels.

Finally, the last 1×1 convolution layer was configured with a stride of 1 and padding of 0. In this layer, the input channel was set to middle_channels, and the output channel was set to out_channels.

1.2. Layers

Layer number	Network	Output Image size
Layer 1	7x7 conv, channel = 64, stride = 2 3x3 max pool, stride = 2	8 x 8
Layer 2	[1x1 conv, channel = 64, 3x3 conv, channel = 64, 1x1 conv, channel = 256] x 2 [1x1 conv, channel = 64, stride = 2 3x3 conv, channel = 64, 1x1 conv, channel = 256] x 1	4 x 4
Layer 3	[1x1 conv, channel = 128, 3x3 conv, channel = 128, 1x1 conv, channel = 512] x 3 [1x1 conv, channel = 128, stride = 2 3x3 conv, channel = 128, 1x1 conv, channel = 512] x 1	2 x 2
Layer 4	[1x1 conv, channel = 256, 3x3 conv, channel = 256, 1x1 conv, channel = 1024] x 6	2 x 2
	AvgPool	1 x 1
	Fully connected layer	?

Figure 1: Layer Configuration of ResNet.

The overall network architecture was constructed following the structure presented in Figure 1. In the final fully connected layer, the input size was set to 1024, and the output size was set to 10, corresponding to the number of target. This configuration allows for the application of a softmax function at the output layer.

2. Result

During the training process, the loss values showed a slight upward trend, increasing from 0.2740 at step 100 to 0.3012 at step 500 within a single epoch. This mild increase can be attributed to several factors, such as fluctuations in mini-batch composition, or the limited training duration. Despite the one epoch, the model achieved an accuracy of 82.37% on the test dataset. This result indicates that the model is capable of learning meaningful representations from the data even with minimal training. However, the upward trend in loss suggests that further training with multiple epochs, hyper parameter tuning, or additional regularization techniques may be necessary to achieve more stable and improved performance.