Project 2 - CNN Architecture Implementation

2021320117 Bae Minseong

In this project, we have to implement bottleneck building block (residual block) and layers for ResNet-50 model.

First, for implementing the residual block, we can use the pre-implemented functions conv1x1 and conv3x3. Like the description in the attached slide, we need to implement 3 consecutive convolution maps: 1x1, 3x3, and 1x1. The key point of implementation is, if the downsampling option of residual block is true, we need to halve the size of input image, so the stride of first 1x1 convolution map must be changed to 2. Then, the shape of original data x and relu(F(x)) can be same. Another key point is that we must consider the padding size for the consecutive convolution maps because we apply 3x3 convolution map between two 1x1 convolution maps, so we must give the padding size 1 to the 3x3 convolution map in the residual block to preserve the width and height of images.

Next, for the implementation of four layers in ResNet-50, we have to fill in the blanks in the skeleton code. First, there are ten classes in the CIFAR-10 dataset, so set the default value of num_classes to 10. In the first layer, we must apply 7x7 convolution net with 64 channels and through this convolutional layer, the input image size is halved. Therefore, we need to set the stride to 2 and padding to 3 to make the size of input image from 32x32 to 16x16 by the formula. (In the formula, we round down the fraction part for the size after convolutional layer.) In the similar manner, for the 3x3 max-pooling layer in the first layer, we have to set the stride to 2 and padding to 1 for the layer. In the second layer to fourth layer, we can use the ResidualBlock class that we already implemented. The number of channels of the input in the layer is determined by the number of channels of the previous layer's output. So, in the second layer, it should be 64 channels for in_channels in the first residual block. The parameters for other residual blocks can be easily determined by the description in the attached slide. It is important that the last residual block in the second and third layer, the downsampling option have to be true because we need to halve the size of images. For the average pooling layer, the output size of the fourth layer is 2x2, so we can determine the kernel size of average pooling layer to 2 to make the output 1x1. Last, for the fully connected layer, we must give the output of fourth layer to the input of fully connected layer so the number of input channels is 1024, and the number of output channels should be the number of classes.

When we train this ResNet-50 model with CIFAR-10 dataset with main.py (I just copied the code of main.py to the Jupyter Notebook and run with it.), we can get the test accuracy of 82.38% for the test dataset. Comparing to the original ResNet structure in the original paper, the structure of the neural network is reduced but it also shows us a great result for the image classification task. I think the most important thing to think when we implement these CNN architectures is considering that if the size of input and output images is correct as we intended because it has many variants like padding, stride, or size of filters.