

COSE474 Deep Learning: Project #2: CNN Architecture Implementation

Project report REUTELSTERZ ELIAS

Description of the code:

First, I tested the code and the package itself with the given code of the vgg16 model. While doing this I stumbled over an error which I could solve by adding the "map_location=device" attribute while loading the checkpoint in the main.py file.

Afterwards I could answer the questions. I solved the first question with adding the layers conv1x1, conv3x3, conv1x1. One just had to add the downsampling in the first conv1x1 layer by adding stride=2 and the conv3x3 layers needed a padding of 1 to remain same size output.

I solved the second question by adding the number of classes = 10 because we got 10 different possible outcomes in the CIFAR10 dataset. When adding the first conv2d layer in the first layer, the in_channels had to fit the depth of the data, which was the number of colors = 3. Afterwards I implemented the layers just as it was described in the pdf-file. In the end it was important that the outcome of the fully connected layer has the same size as the num_classes and the average pooling has kernel_size = 2 because we want to reduce the dimension 2x2 to a scalar.

Results:

When running the vgg16 I get the following results:

Epoch [1/1], Step [100/500] Loss: 0.1835

Epoch [1/1], Step [200/500] Loss: 0.1866

Epoch [1/1], Step [300/500] Loss: 0.1806

Epoch [1/1], Step [400/500] Loss: 0.1829

Epoch [1/1], Step [500/500] Loss: 0.1887

Accuracy of the model on the test images: 85.84 %

Running the resnet50 model I get the following results:

Epoch [1/1], Step [100/500] Loss: 0.2811

Epoch [1/1], Step [200/500] Loss: 0.2807

Epoch [1/1], Step [300/500] Loss: 0.2891

Epoch [1/1], Step [400/500] Loss: 0.2945

Epoch [1/1], Step [500/500] Loss: 0.2986

Accuracy of the model on the test images: 82.61 %

Discussion:

I wondered about the reasoning behind reducing the downsampling in the conv1x1 layer instead of the conv3x3 layer. It's obviously less computing expensive to first reduce it because the conv3x3 layer has more parameters, but you have a loss of information if you just take every second pixel in the conv1x1 layer.

Additionally, one could discuss the difference between the two models. The vgg16 model has a better accuracy. One obvious difference is the data preprocessing. In the vgg16 model the data gets normalized and centralized, which can affect the whole calculation. Of course since the model infrastructure is a completely different one, we get different results out of both of the models.