CIFAR-10 Data Image Classification

Yutae Lee

Texas A&M University
CSCE636
11-22-2022

Abstract

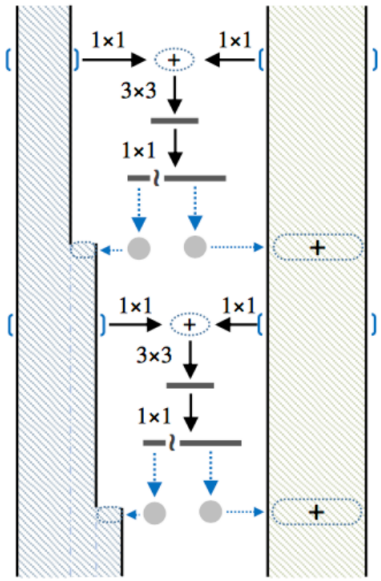
As the technology of Deep Learning advances, different types of model appears. This project aims to classify the CIFAR-10 image data with 10 classes using the Dual Path Network system which combines the idea of ResNet with DenseNet.

1 Model Proposal

One of the most common network that is widely used is ResNet which is developed by stacking of Residual Blocks. These Residual Blocks are mainly created by multiple skip connections, where it connects activation to the farther layer by skipping layers. ResNet is beneficial in that if a layer is not performing well it allows the model skip. However, there are still limits in that if the layer is too deep ResNet finds it hard to detect which layer may harm its architecture. To complement this weakness, DenseNet has been introduced. In DenseNet, rather than skipping the layers, it projects to fully connect the layers, in other words each layers receives extra inputs from all layers that came before it and transmits its feature map to all layers that came after it. However DenseNet also has problems in high memory along with computational complexity. This is main because of how the architecture is set up data is being replicated multiple times.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Figure 1: Image of Dual Path Network



This project utilizes Dual Path Network system in order to embed the idea of ResNet while having a bit of architect of DenseNet along with it.

Dual Path Network is formed from stacking BottleNeck structured blocks. The BottleNeck that structures this block starts with 1×1 convolutional layer followed by 3×3 convolutional layer ending with 1×1 convolution. Then the output is splitted into two where one goes into ResNet and other going into DenseNet. The architect can be seen easily by the figure 1 above.

2 Implementation

In this section, I am going to explain how the code implemented the Dual Path Network.

As a data preprocessing procedures, I have used two codes DataLoader.py and ImageUtils.py In the code DataLoader.py, I was able to succefully load the cifar-10-batches-py file using pickle.

Mainly the code was inspired from the default code that was given in the https://www.cs.toronto.edu/ kriz/cifar.html website

There I first made an empty np.array for x and y, and I appended each data_batch_'s by reshaping X.

Through ImageUtils.py, I have done some preprocessing procedure such as parsign[,3072] into [depth,height,width]. Also during preprocessing, I have added the original image into the padded image of pixel 0, after resizing the image to add four extra pixels on each side. After that, I randomly cropped a [32, 32] section of the image.

As for Hyperparameters, I have used weight decay as 0.0005, learning rate as 0.01, momentum 0.7, max epochs as 200. Also for the trainig paramters I have used batch size = 256.

As expected as epochs grows higher, the accuracy and loss improved. However, I was able to see that the improvement rate was exponentially decreasing. In other words, from epoch 0 to 50 about 20% accuracy has improved while 150 to 200 0.5% accuracy improved

This is the Test Accuracy after 200 Epochs:

```
Test loss:  0.5907596349716187
Previous best:  0
Current best:  0.9043999910354614
Saved the best model.

score = 0.904400 in validation set.
```

3 Conclusion

As working on this project, I tried to compare the accuracy and speed to the ResNet that was done from Homework2.

One observation that I made was that speed for the Dual Path Network was significantly slower. This was expected as we are embedding DenseNet into our model which has weakness in memory and computational complexity compared to ResNet. However, the difference between the was quite unexpected. Training 200 epochs for Dual Path Network took nearly 5 hours, while ResNet took 2 hours.

Another observation that I made is that the accuracy did not quite differ. As you can see the Testing accuracy for Dual Path Network model was 90.4% while ResNet most of the time resulted in range between 89-90%.