Danny Hong
ECE 472: HW 4 Writeup

In this assignment, I attempted to classify both CIFAR-10 and CIFAR-100. The two datasets were downloaded from this link: https://www.cs.toronto.edu/~kriz/cifar.html. I first tried to reuse the model that I created in the last homework assignment for classifying MNIST. The results were terrible for both CIFAR-10 and CIFAR-100. In order to improve my accuracy results, I did some research and found that it was best to use pairs of double convolution blocks. This means that instead of having many blocks, with each block having different filtering values, there would be convolution blocks organized in sets of two, with there being identical filtering values for each set of two convolution blocks. I continued this layering process until I reached a filter size of 512. In addition, after each single layer, I then applied batch normalization. I also applied a dropout operation after every set of 2 convolution blocks, and imposed an L2 penalty at every single layer as opposed to just applying L2 regularization at the last classification layer like I had done in my MNIST program. Looking back, it is possible that my model had suffered from having too many layers, which would make the model unnecessarily complex and potentially lead to overfitting of the data.

In regards to data preprocessing, I decided to normalize the input image data before performing the training/validation data split. I also set up an image data generator to help with improving the accuracy as well.

In regards to learning rates, after the model had run for 75 epochs, I decided to reduce the learning rate for every following 25 epochs as well to allow for fine tuning of the weights towards the end of the learning process. I first trained the model for 125 epochs with a batch size of 512. This led to a test accuracy of about 85 pct. Therefore, I decided to increase the number of epochs (I experimented with 150, 200, and 225) until I hit an accuracy of 0.91 after 225 epochs.

I tried to use this same model to classify CIFAR-100, but I kept getting a top 5 accuracy of 86-87%, no matter how much I tried to increase the number of epochs or decrease my batch size. I therefore decided to implement an open source Inception (Google Net) model from this link:https://machinelearningknowledge.ai/googlenet-architecture-implementation-in-keras-with-cifar-10-dataset/. I also replaced my learning rate decay method with the learning rate decay formula that the google net program had in place. I first ran the model for 50 epochs with a batch size of 128, and I obtained a top 5 test accuracy of 89.7%. In order to achieve an accuracy of at least 90%, I decided to increase the number of epochs to 75 and decrease the batch size to 64, before running the model again. Thankfully, the resulting accuracy after this was about 90.1%, meaning that I did not need to do any more work.