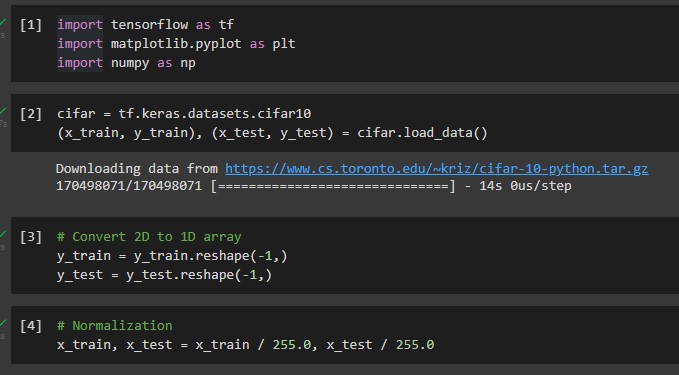
**Homework 3**

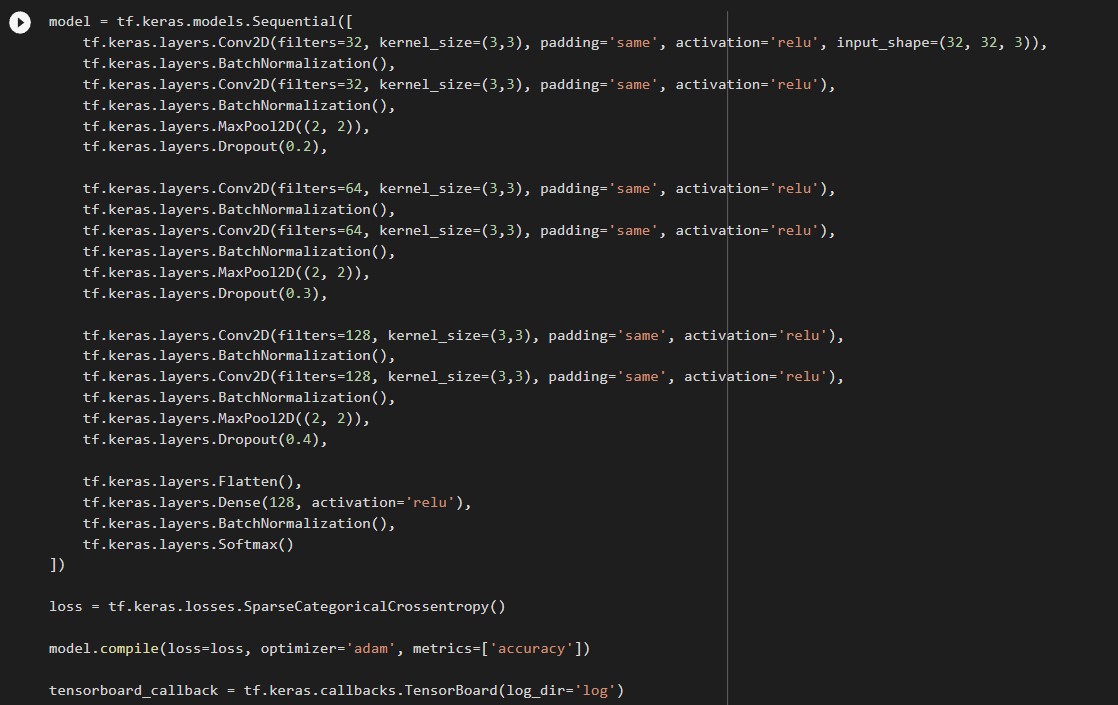


[1] I imported necessary packages.

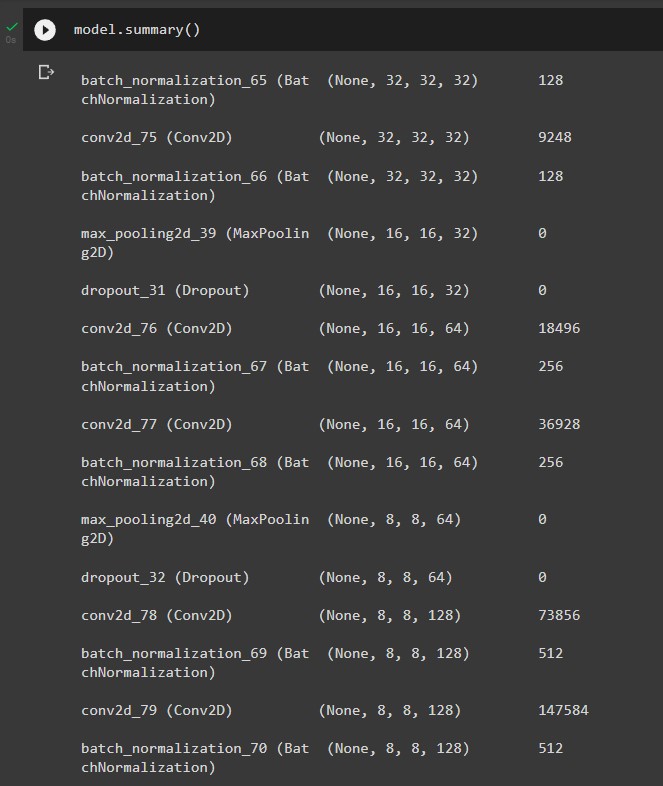
[2] Download CIFAR10 dataset and assign it to cifar variable. It already has train and test data separated, so assign those to the x and y variables.

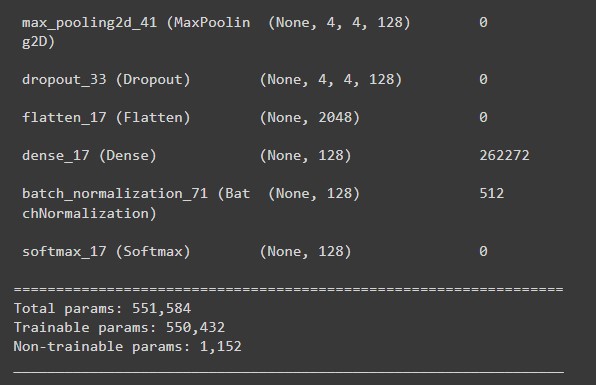
[3] Initially, y is 2D array, I do reshape(-1,) to convert it to a 1D array.

[4] Divide the x by 255.0 for a normalization.

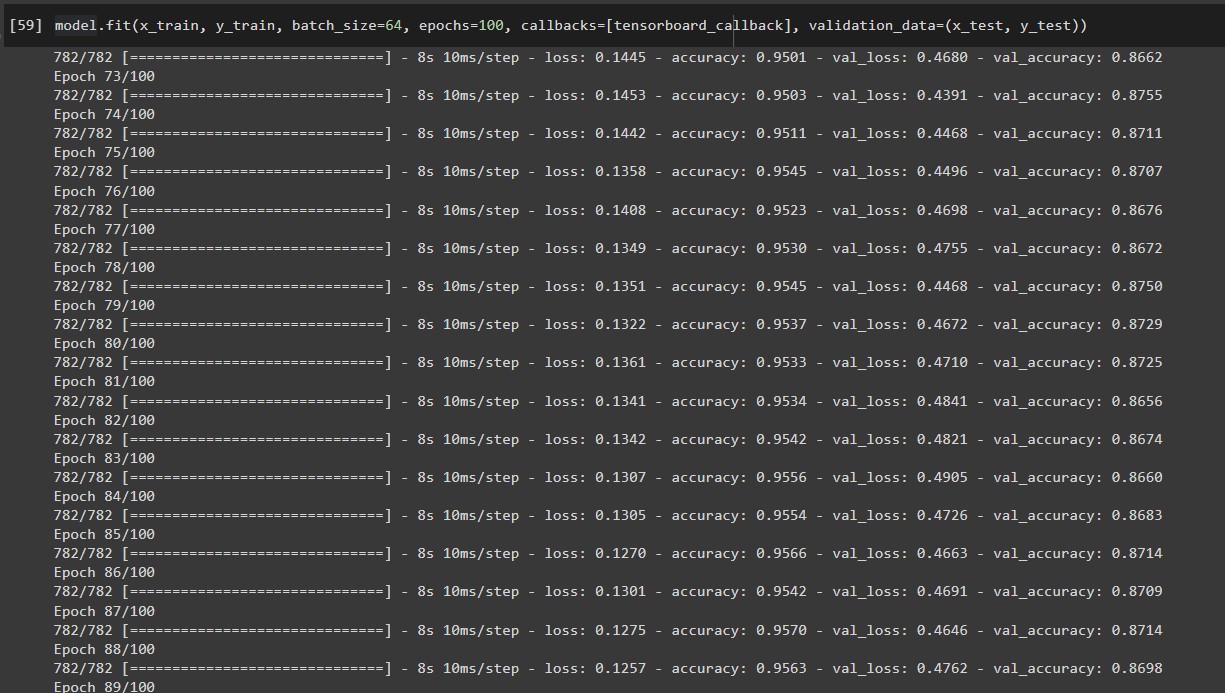


This is the Keras CNN model. The input shape is (32, 32, 3). Since I will train in batch, so I have to add batch normalizations. Batch normalization has to be after Conv2D but before MaxPool2D. I train Conv2D 🡪 BatchNormalization 🡪 Conv2D 🡪 Batch Normalization 🡪 MaxPool2D 🡪 Dropout, and increase the filters size from time to time. I have to deal with the overfitting on training data problem, so I add Dropout layer to my neural network. Then, flatten the layer, add one normal hidden layer, do the batch normalization again, then add the Softmax layer lastly. The loss function is Sparse categorical crossentropy. Optimizer will be AdamOptimizer. Use accuracy metrics for measuring training accuracy.

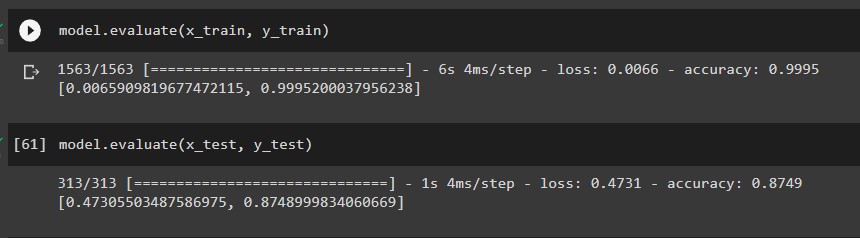




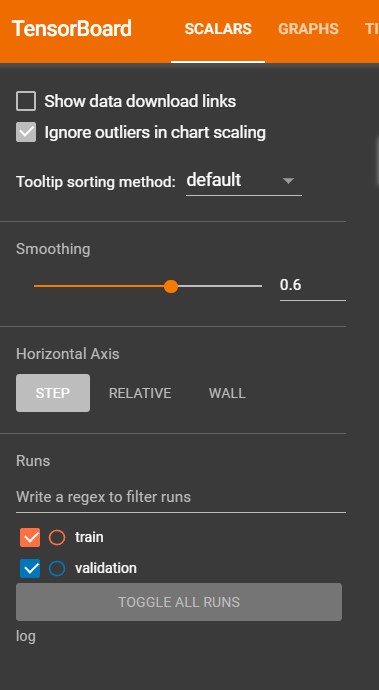
These figures show the model summary.



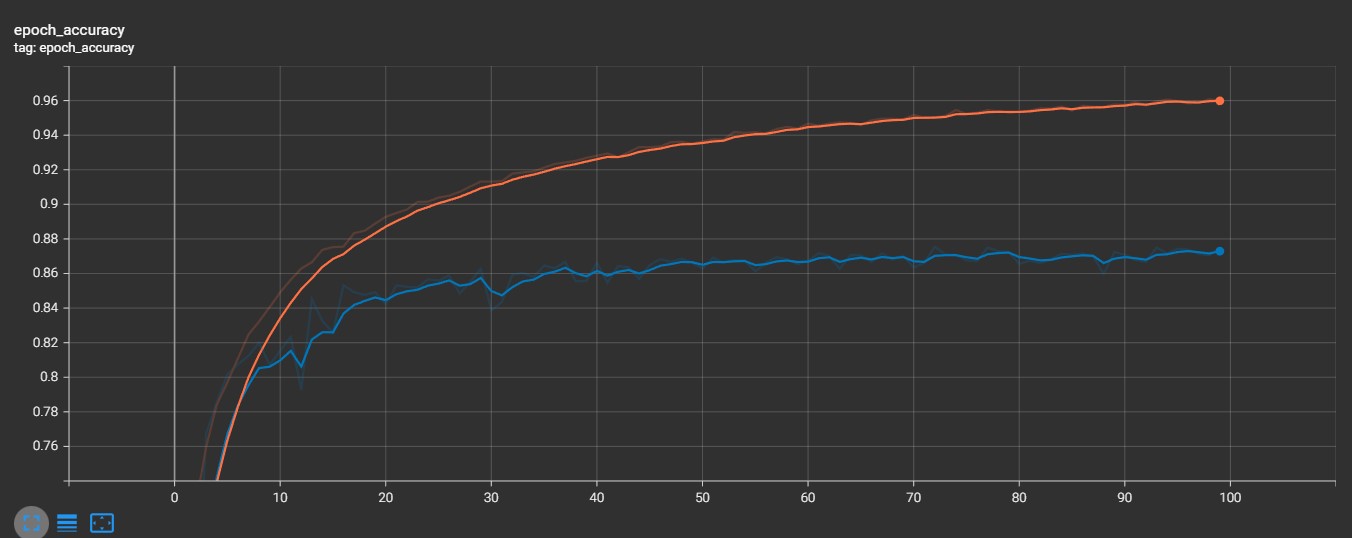
Then, fit the model. My batch size is 64, with 100 epochs. The validation data parameter here uses the variable x\_test and y\_test that we assigned it beforehand. Also use Google Colab’s GPU because if not, its going to take too long time. After the training is completed, we will check the results.



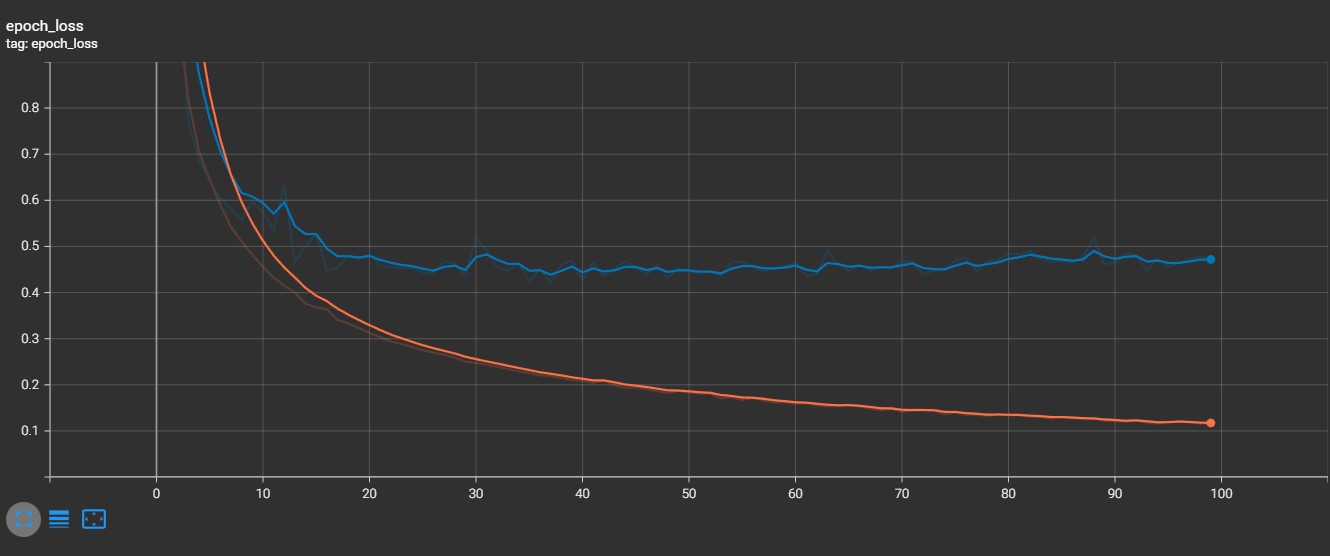
Here you can see that the training loss is 0.0066, the training accuracy is 0.9995, the testing loss is 0.4731, and the testing accuracy is 0.8749. I tried changing parameters for a several times, and this one was my best result. Now, let’s see the Tensorboard graph results.



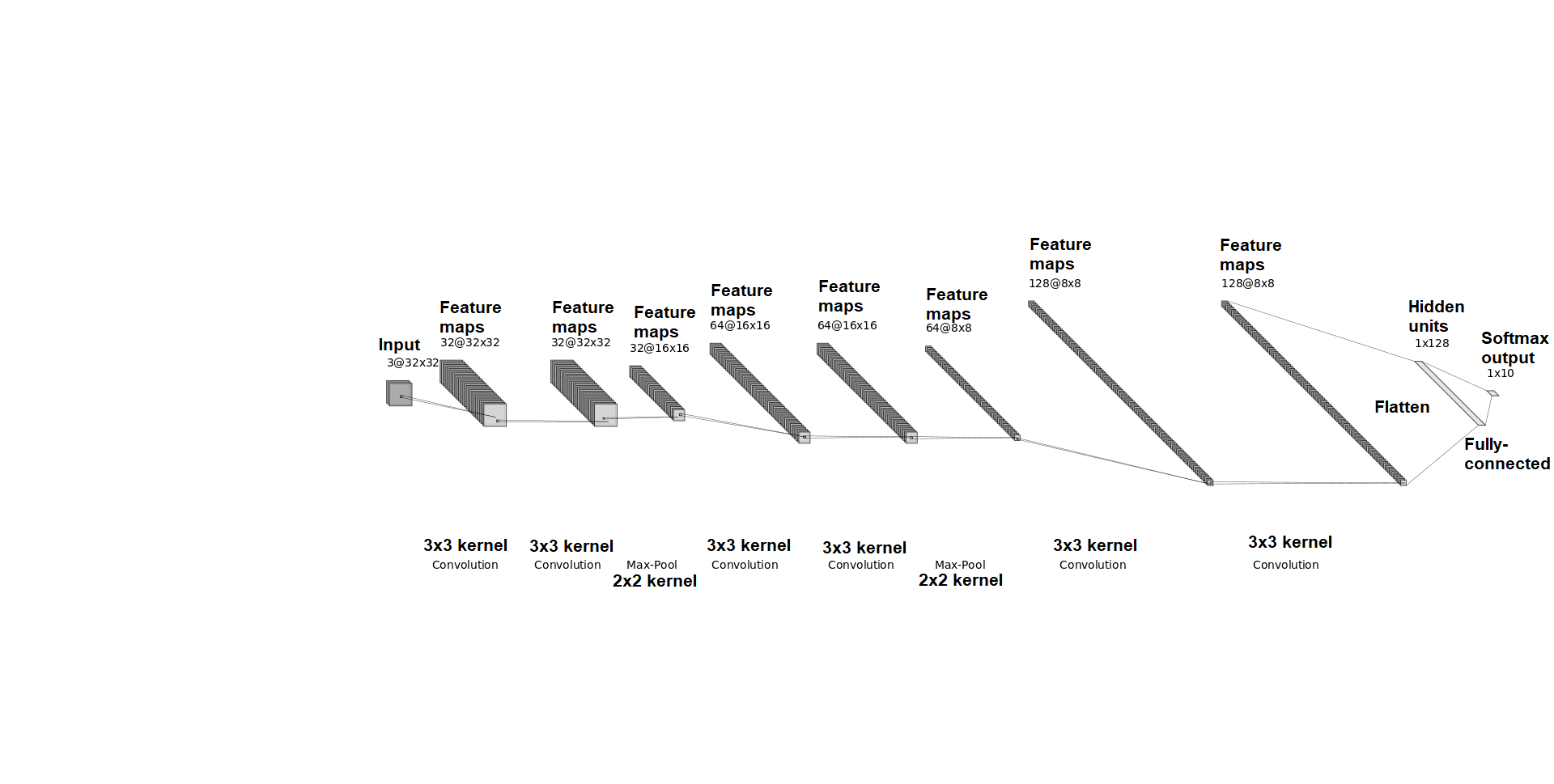
Orange line means training, blue line means validation (testing)



Here is the accuracy in each epoch of training (orange) and testing (blue) data.



Here is the loss in each epoch of training (orange) and testing (blue) data.

Here is the visualization of CNN architecture.