

R05921034 吳明憲

1-1. Describe how you preprocess your data and the model architecture. (30%)

```
def convert_images(raw_images):  
    raw = np.array(raw_images, dtype = float)/255.0
```

```
datagen = ImageDataGenerator(  
    featurewise_center=False, # set input mean to 0 over the dataset  
    samplewise_center=False, # set each sample mean to 0  
    featurewise_std_normalization=False, # divide inputs by std of the dataset  
    samplewise_std_normalization=False, # divide each input by its std  
    zca_whitening=False, # apply ZCA whitening  
    rotation_range=15, # randomly rotate images in the range (degrees, 0 to 180)  
    width_shift_range=0.1, # randomly shift images horizontally (fraction of total width)  
    height_shift_range=0.1, # randomly shift images vertically (fraction of total height)  
    horizontal_flip=True, # randomly flip images  
    vertical_flip=False) # randomly flip images
```

我利用 imagedatagenerator 去讓圖片有旋轉，以及水平翻轉,並且去同除 255 去做預處理。

```
model = Sequential()  
  
#model.add(Conv2D(32, (3,3), padding='same', input_shape= x_train.shape[1:]))  
model.add(Conv2D(32, kernel_size=(3, 3), padding='same', input_shape=(32,32,3)))  
model.add(Activation('relu'))  
model.add(Conv2D(32, (3, 3)))  
model.add(Activation('relu'))  
model.add(Dropout(0.25))  
  
model.add(Conv2D(64, (3, 3), padding='same'))  
model.add(Activation('relu'))  
model.add(Conv2D(64, (3, 3)))  
model.add(Activation('relu'))  
model.add(AveragePooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
  
model.add(Conv2D(128, (3, 3), padding='same'))  
model.add(Activation('relu'))  
model.add(Conv2D(128, (3, 3)))  
model.add(Activation('relu'))  
model.add(AveragePooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
  
model.add(Conv2D(256, (3, 3), padding='same'))  
model.add(Activation('relu'))  
model.add(Conv2D(256, (1, 1)))  
model.add(Activation('relu'))  
model.add(AveragePooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
  
model.add(Flatten())  
model.add(Dense(512))  
model.add(Activation('relu'))  
model.add(Dropout(0.5))  
model.add(Dense(10))  
model.add(Activation('softmax'))
```

上圖是我最佳的 model，我用了 6 層的 convolution 2D,並在最後面接一個 classifier 和 softmax 去分類，那中間我是用 average pooling 去處理，並加上 dropout 去訓練。

## 1-2. Compare different architectures or parameters and record their performance (30%)

我做了不同深度的 cnn 的比較，最佳的是六層去 train，下圖是四層 cnn model 以及結果。我跑了 300 個 epochs 之後 acc 大概是在 0.86 左右從圖形看來，雖然成長緩慢但是仍是還可以將 epochs 加大。

```
model = Sequential()

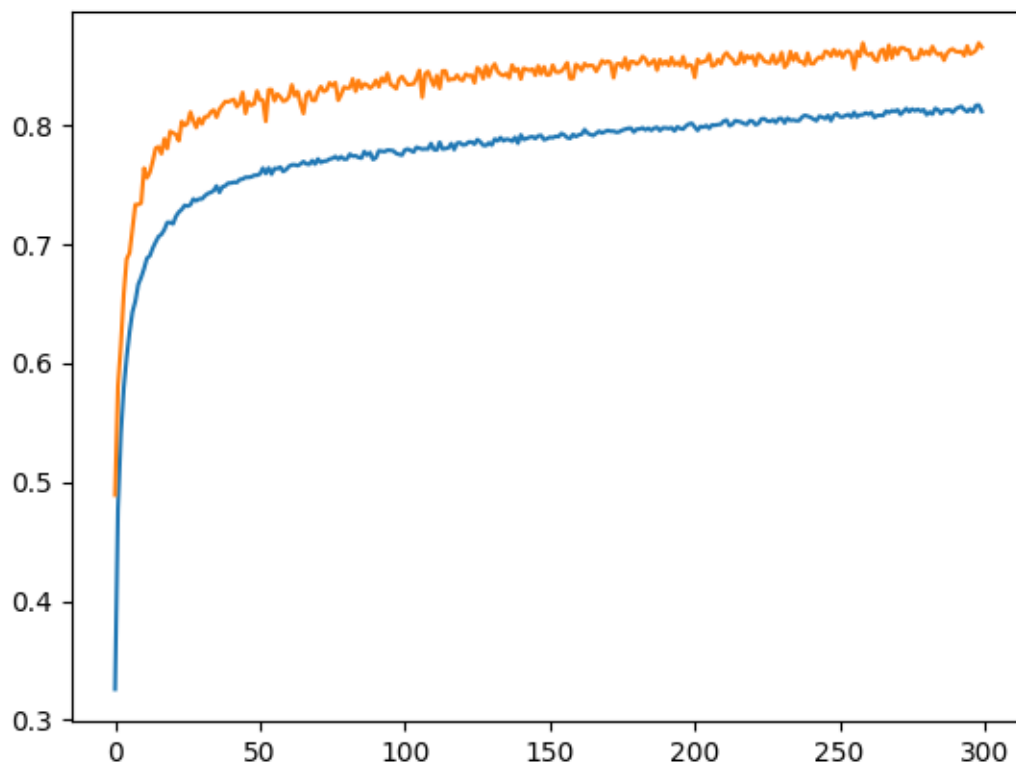
#model.add(Conv2D(32, (3,3), padding='same', input_shape= x_train.shape[1:]))
model.add(Conv2D(32, kernel_size=(3, 3), padding='same', input_shape=(32,32,3)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(AveragePooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(AveragePooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

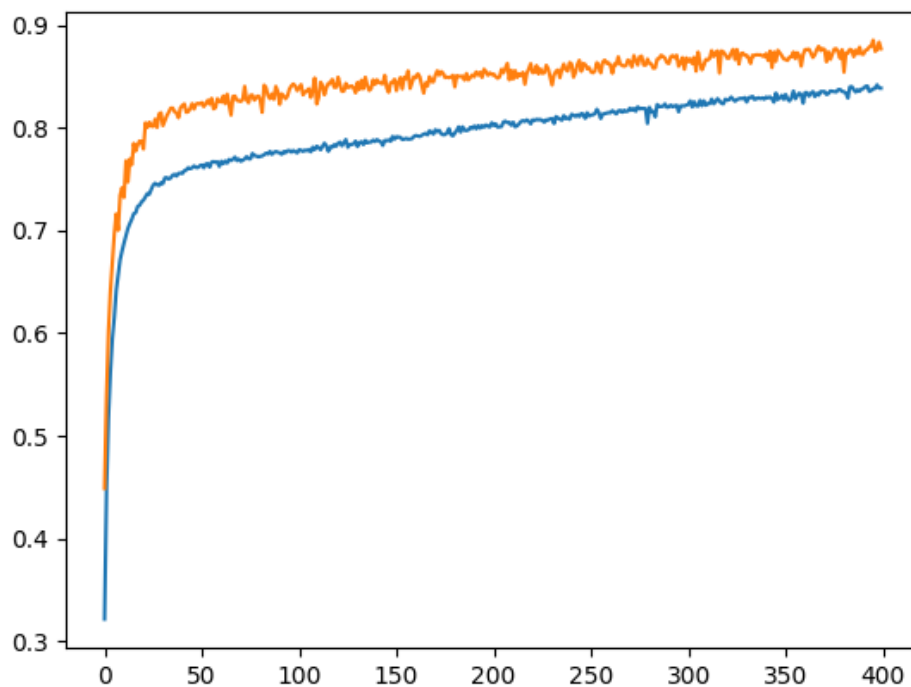
model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(10))
model.add(Activation('softmax'))

9248/10000 [=====>...] - ETA: 0s
9536/10000 [=====>..] - ETA: 0s
9824/10000 [=====>.] - ETA: 0s
10000/10000 [=====] - 2s 184us/step
Using TensorFlow backend.
Test loss: 0.416846414292
Test accuracy: 0.8661
```



2. screenshot.png: A screenshot of the best testing set accuracy on terminal. (20%)

```
epoch 398/400
1563/1562 [=====] - 18s 12ms/step - loss: 0.4812 - acc: 0.8417 - val_loss: 0.3877 - val_acc: 0.8757
epoch 399/400
1563/1562 [=====] - 18s 12ms/step - loss: 0.4791 - acc: 0.8385 - val_loss: 0.3744 - val_acc: 0.8833
epoch 400/400
1563/1562 [=====] - 18s 12ms/step - loss: 0.4850 - acc: 0.8386 - val_loss: 0.3700 - val_acc: 0.8768
10000/10000 [=====] - 2s 192us/step
test loss: 0.369957184339
test accuracy: 0.8768
```



上面兩張圖片是我最佳的 model 的圖片，

我最好的 model 大概是跑出 0.88 左右的 accuracy,從下圖中可以看出

大概到 50 epochs 之後就就開始有趨於平緩，我總共是跑 400epochs 不過仍然有再上升的趨勢

### 3.hw3.py: A script that you train and test your model (20%)

請將 cifar10 的 data\_batch\_1 ~ data\_batch\_5 還有 test\_batch 跟 hw3.py 放在同一資料夾下，然後直接鍵入 python3 hw3.py 即可開始 train