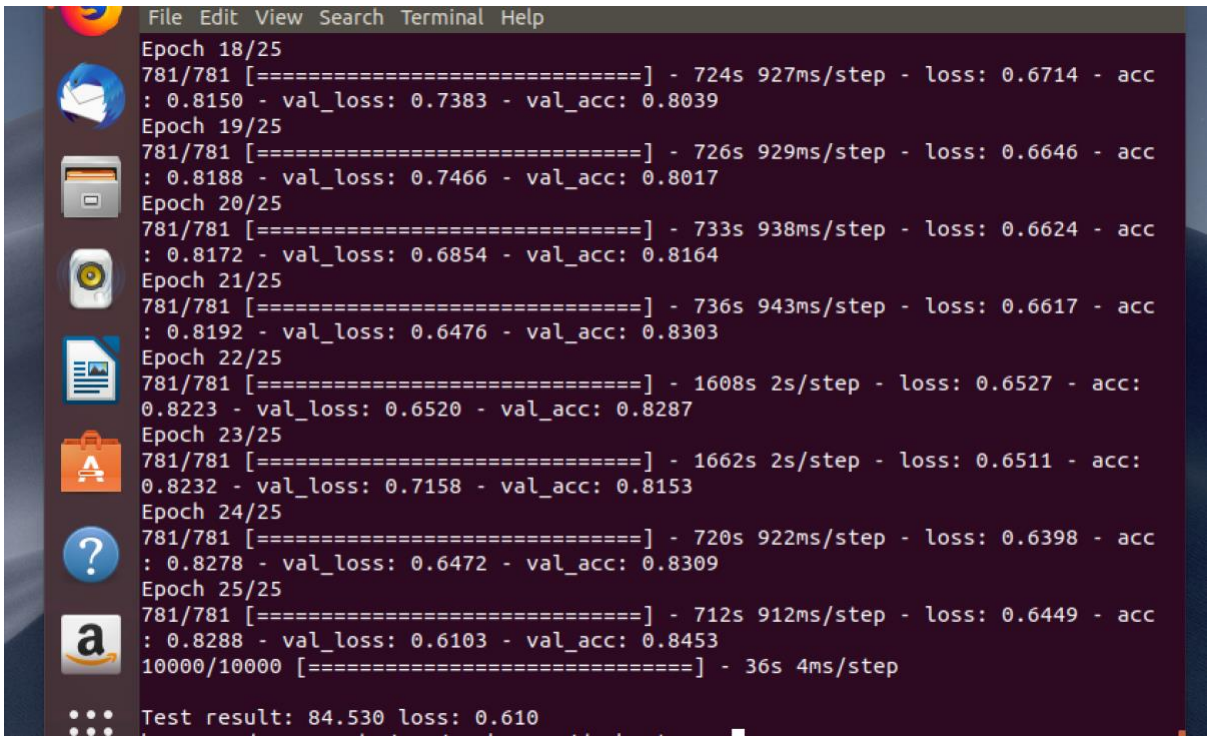
CLOUD FOG COMPUTING AND BIG DATA ANALYSIS

HOMEWORK 6: ***Deep Learning with TensorFlow and Keras on Docker***

By 0516251 費蓋德

Part I: cifar10_cnn with more than 80% accuracy on the test set (after just 25 epochs):



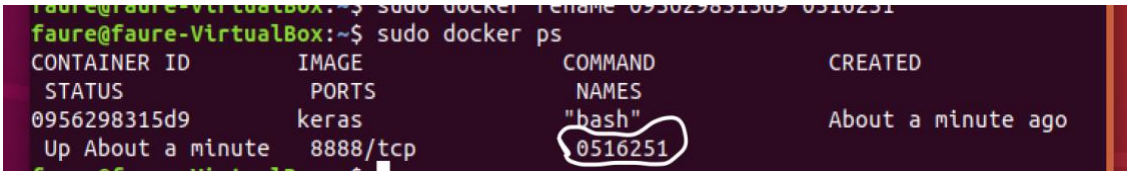
```
File  Edit  View  Search  Terminal  Help
Epoch 18/25
781/781 [==============================] - 724s 927ms/step - loss: 0.6714 - acc
: 0.8150 - val_loss: 0.7383 - val_acc: 0.8039
Epoch 19/25
781/781 [==============================] - 726s 929ms/step - loss: 0.6646 - acc
: 0.8188 - val_loss: 0.7466 - val_acc: 0.8017
Epoch 20/25
781/781 [==============================] - 733s 938ms/step - loss: 0.6624 - acc
: 0.8172 - val_loss: 0.6854 - val_acc: 0.8164
Epoch 21/25
781/781 [==============================] - 736s 943ms/step - loss: 0.6617 - acc
: 0.8192 - val_loss: 0.6476 - val_acc: 0.8303
Epoch 22/25
781/781 [==============================] - 1608s 2s/step - loss: 0.6527 - acc:
0.8223 - val_loss: 0.6520 - val_acc: 0.8287
Epoch 23/25
781/781 [==============================] - 1662s 2s/step - loss: 0.6511 - acc:
0.8232 - val_loss: 0.7158 - val_acc: 0.8153
Epoch 24/25
781/781 [==============================] - 720s 922ms/step - loss: 0.6398 - acc
: 0.8278 - val_loss: 0.6472 - val_acc: 0.8309
Epoch 25/25
781/781 [==============================] - 712s 912ms/step - loss: 0.6449 - acc
: 0.8288 - val_loss: 0.6103 - val_acc: 0.8453
10000/10000 [==============================] - 36s 4ms/step

Test result: 84.530 loss: 0.610
```

I could probably reach 90 percent accuracy with more epochs but I don't have GPU so the training is very slow.

Part II

1- **My Docker environment:**



```
faure@faure-VirtualBox:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
 STATUS              PORTS               NAMES
0956298315d9        keras               "bash"              About a minute ago
 Up About a minute   8888/tcp           0516251
```

2- **Architecture of my program**

The architecture contains 6-layered CNN, 2D pooling layers, a flatten and a dense layer. Here is the summary:

```
2018-12-17 10:29:37.673001: I tensorflow/core/platform/cpu_feature_guard.cc:141] Yo
s that this TensorFlow binary was not compiled to use: AVX2
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
activation_1 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 16, 64)        256
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 64)        36928
_________________________________________________________________
activation_4 (Activation)    (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_4 (Batch (None, 16, 16, 64)        256
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
activation_5 (Activation)    (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_5 (Batch (None, 8, 8, 128)         512
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 8, 8, 128)         147584
_________________________________________________________________
activation_6 (Activation)    (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_6 (Batch (None, 8, 8, 128)         512
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                20490
=================================================================
```

3- My experiences in doing this homework

The first problem I encountered has something to do with the performance of my computer. I first ran the provided cifar10_cnn.py file and it took 2 days to finish but the accuracy does not meet the requirements.

After analysis of the dataset on hand, I realized that these images cannot be classified properly with so few convolutional layers, so I added more and also increased the number of epochs. The program was running correctly this time till I realized there will be overfitting because of the huge number of epochs I tried to use. With the number of layers I was using, 25 epochs can give around 85% accuracy, I decided to stop at this point.

In this assignment, I have learned about keras library especially how to train a CNN model. I have learned about different kinds of layers (Dense, Conv2D…) , different kinds of activation function (relu, softmax…) and also how to prevent underfitting or overfitting when building neural nets.