MSBD 6000B Deep Learning

Project 2

Student Name: Jingyi Wang
Student ID: 20469513
Due date: 11:59pm, Nov 23, 2017

1. Overview

This project is required to train a deep model and then test on a test set including 5 kinds of flowers.

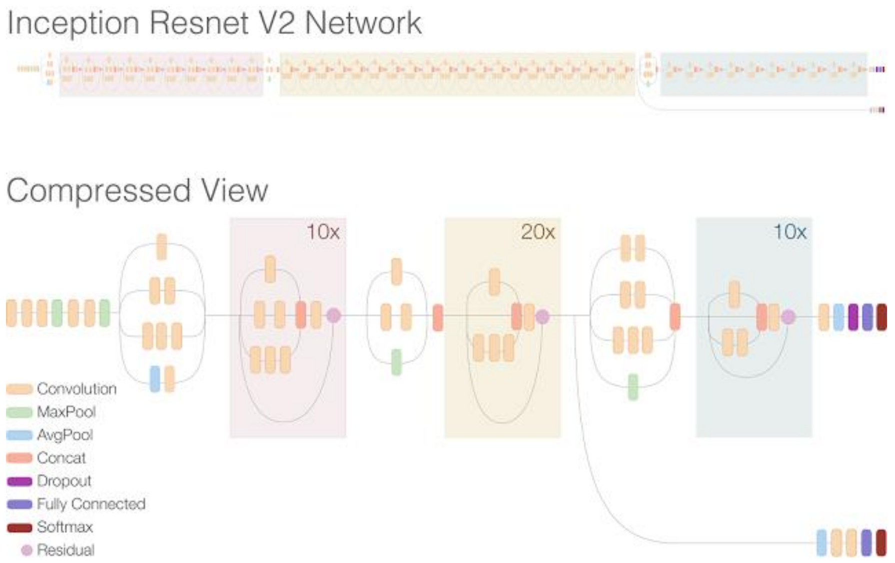
2. Experiment

Experiment has been conducted by using an Inception-ResNet-v2 model in Tensorflow-slim and fine-tuning some parameters. First train the model on the training set with 2569 instances whose paths and labels are stated in 'train.txt', and then test on the validation set with 550 instances whose paths and labels are stated in 'val.txt'. The final model is derived by training with both train and validation sets and would be tested using another set of 551 instances in 'text.txt'.

2.0 Prepare the dataset

Initially, the flower dataset is converted into the format of TF-records and separated in two directories of 'train' and 'val' according to the provided .txt files. Few modifications in classification module including 'train_image_classifier.py' and 'dataset_factory.py' are necessary in order to make the model adapted to the required dataset.

2.1 Inception-ResNet-v2



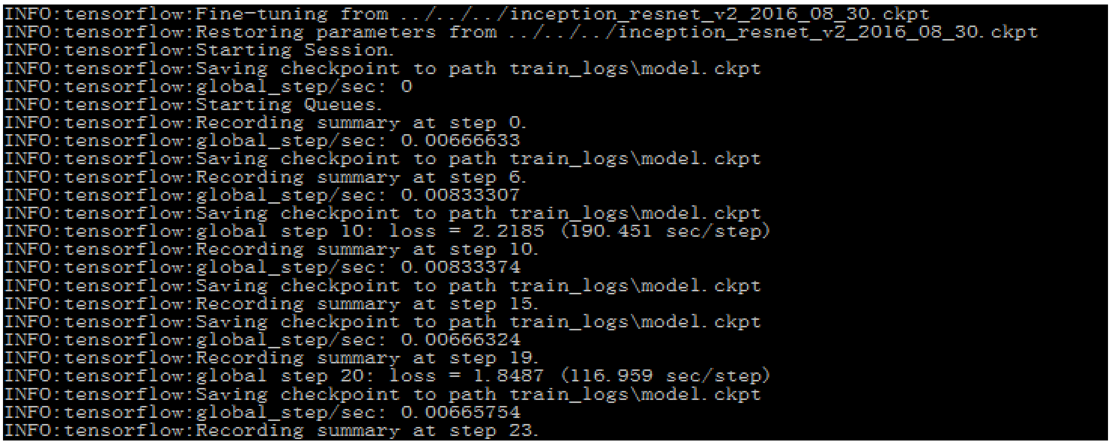
2.2 Classification

On training set, construct the Inception-ResNet-v2 model with the following parameters in command.

Command line:

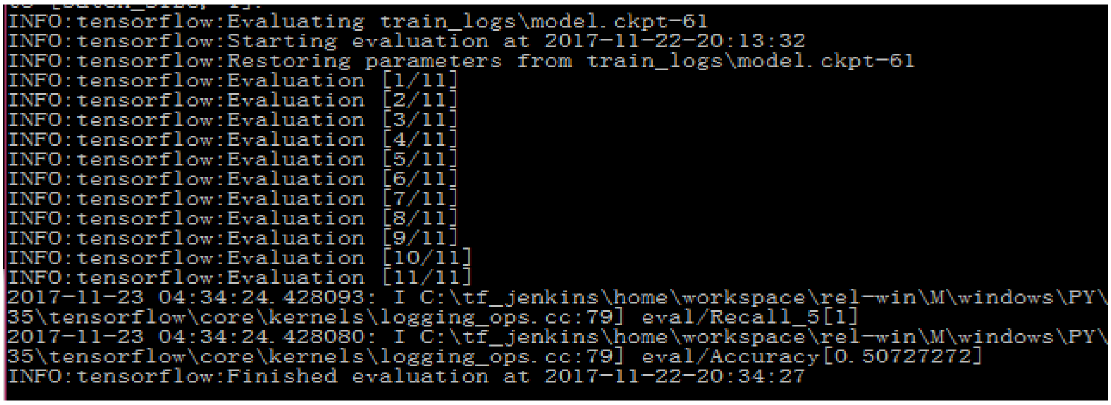
```
python train_image_classifier.py --clone_on_cpu=True --train_dir=train_logs --dataset_dir=../../../train
--num_samples=2569 --num_classes=5 --labels_to_names_path=../../../labels.txt --
model_name=inception_resnet_v2 --checkpoint_path=../../../inception_resnet_v2_2016_08_30.ckpt --
checkpoint_exclude_scopes=InceptionResnetV2/Logits,InceptionResnetV2/AuxLogits --
trainable_scopes=InceptionResnetV2/Logits,InceptionResnetV2/AuxLogits.
```

Output:

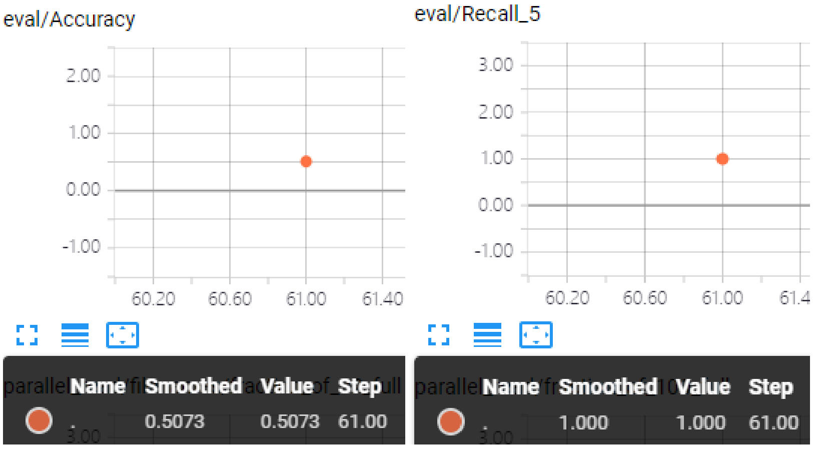


```
INFO:tensorflow:Fine-tuning from ../../../inception_resnet_v2_2016_08_30.ckpt
INFO:tensorflow:Restoring parameters from ../../../inception_resnet_v2_2016_08_30.ckpt
INFO:tensorflow:Starting Session.
INFO:tensorflow:Saving checkpoint to path train_logs\model.ckpt
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:Starting Queues.
INFO:tensorflow:Recording summary at step 0.
INFO:tensorflow:global_step/sec: 0.00666633
INFO:tensorflow:Saving checkpoint to path train_logs\model.ckpt
INFO:tensorflow:Recording summary at step 6.
INFO:tensorflow:global_step/sec: 0.00833307
INFO:tensorflow:Saving checkpoint to path train_logs\model.ckpt
INFO:tensorflow:global step 10: loss = 2.2185 (190.451 sec/step)
INFO:tensorflow:Recording summary at step 10.
INFO:tensorflow:global_step/sec: 0.00833374
INFO:tensorflow:Saving checkpoint to path train_logs\model.ckpt
INFO:tensorflow:Recording summary at step 15.
INFO:tensorflow:Saving checkpoint to path train_logs\model.ckpt
INFO:tensorflow:global_step/sec: 0.00666324
INFO:tensorflow:Recording summary at step 19.
INFO:tensorflow:global step 20: loss = 1.8487 (116.959 sec/step)
INFO:tensorflow:Saving checkpoint to path train_logs\model.ckpt
INFO:tensorflow:global_step/sec: 0.00665754
INFO:tensorflow:Recording summary at step 23.
```

Basically, the cross-entropy loss decreases as the number of training steps increase. Since the training process is pretty time-consuming, here only use the model trained after the step 61 to make validation. Obtain the following output on validation set. By exploring graphs on Tensorboard, notice the accuracy and recall of predictions on validation set reach about 50.73% and 100% respectively.



```
INFO:tensorflow:Evaluating train_logs\model.ckpt-61
INFO:tensorflow:Starting evaluation at 2017-11-22-20:13:32
INFO:tensorflow:Restoring parameters from train_logs\model.ckpt-61
INFO:tensorflow:Evaluation [1/11]
INFO:tensorflow:Evaluation [2/11]
INFO:tensorflow:Evaluation [3/11]
INFO:tensorflow:Evaluation [4/11]
INFO:tensorflow:Evaluation [5/11]
INFO:tensorflow:Evaluation [6/11]
INFO:tensorflow:Evaluation [7/11]
INFO:tensorflow:Evaluation [8/11]
INFO:tensorflow:Evaluation [9/11]
INFO:tensorflow:Evaluation [10/11]
INFO:tensorflow:Evaluation [11/11]
2017-11-23 04:34:24.428093: I C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\
35\tensorflow\core\kernels\logging_ops.cc:79] eval/Recall_5[1]
2017-11-23 04:34:24.428080: I C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\
35\tensorflow\core\kernels\logging_ops.cc:79] eval/Accuracy[0.50727272]
INFO:tensorflow:Finished evaluation at 2017-11-22-20:34:27
```



3. Conclusion

Although the pre-trained model has been applied to the provided training set to get fine-tuning, the losses tend not to decrease significantly within very limited number of training steps. With the current model, results on validation set is not satisfactory as expected. It is possible that performance would get improved with a fully trained Inception-ResNet-v2 model.

Reference:

1. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, 770-778.