Rishikesh_Majumder_w251_HW5

1. What is TensorFlow? Which company is the leading contributor to TensorFlow?
TensorFlow is an open source library that is used in machine learning/deep learning for numerical tasks.
Alphabet/Google
2. What is TensorRT? How is it different from TensorFlow?
NVIDIA TensorRT is an efficient GPU based platform to get deep learning inferences with high throughput and low latency.
TensorRT can be integrated with Tensorflow and it boosts the performance of deep learning inference tasks with efficient GPU usage.
3. What is ImageNet? How many images does it contain? How many classes?
ImageNet is a hierarchical database for images and it follows WordNet noun structure.
14,197,122 images and 21841 classes
4. Please research and explain the differences between MobileNet and GoogleNet (Inception) architectures.
MobileNet (CNN) architecture is a light-weighted and optimized for vision applications in mobile devices.
GoogleNet is an architecture with wider and larger nets and it uses a combination of pooling, convolutions and inception modules.
5. In your own words, what is a bottleneck?
In a neural network, the bottleneck is the layer before the last output layer. It has a more compact structure than the internal layers.
6. How is a bottleneck different from the concept of layer freezing?
The bottleneck can be responsible for dimensionality reduction before the classification layer (output), while layer freezing is used to keep weights fixed in one or more layer(s) to accelerate the training process.
7. In this lab, you trained the last layer (all the previous layers retain their already-trained state). Explain how the lab used the previous layers (where did they come from? how were they used in the process?)
This lab uses the concept of transfer learning. There is already a pre-trained image classification model which is generated from ImageNet Large Visual Recognition

Challenge dataset. This pre-trained model was not created with flowers' data input, but the by adding last image classification layer, the image classifying output from bottleneck layer will be the input to the newly added layer and that helps to classify flower images.

8. How does a low `--learning_rate` (step 7) value (like 0.005) affect the precision? How much longer does training take?

With the learning rate 0.005, the precision goes down from 91% to 89%.

It takes almost the same amount of time.

9. How about a `--learning_rate` (step 7) of 1.0? Is the precision still good enough to produce a usable graph?

Yes, precision is still 88%.

10. For step 8, you can use any images you like. Pictures of food, people, or animals work well. You can even use ImageNet images. How accurate was your model? Were you able to train it using a few images, or did you need a lot?

When used 3 and 6 images in two classes – tiger and elephant, it threw an exception Label elephant has no images in the category validation. According to the suggested minimum count, it should be at least 20.

When used more than 50 images, the training went fine and the accuracy level is also significantly high (90%).

11. Run the script on the CPU (see instructions above) How does the training time compare to the default network training (section 4)? Why?

On CPU, it took about 11 min 30 seconds where the default training time was 10 min 40 seconds.

With multiple cores and concurrent thread calls, GPU can handle vector/matrix operations in parallel, but since MobileNet was used as architecture, it did not fully utilize GPU potential.

12. Try the training again, but this time do `export ARCHITECTURE="inception_v3"` Are CPU and GPU training times different?

With inception_v3 architecture option, CPU takes longer than an hour and GPU takes about 17 mins.

13. Given the hints under the notes section, if we trained Inception_v3, what do we need to pass to replace ??? below to the label_image script? Can we also glean the answer from examining TensorBoard?

```
python -m scripts.label_image --input_layer=27 --input_height=299 --input_width=299
--graph=tf_files/retrained_graph.pb
--image=tf_files/flower_photos/daisy/21652746_cc379e0eea_m.jpg
```