Homework 5 Answers

1. What is TensorFlow? Which company is the leading contributor to TensorFlow?
    - **TensorFlow** is an open source Deep Learning library developed by Google that is **used** to perform complex numerical operations and several other tasks to model Deep Learning models. It's architecture allows easy deployment of computations across multiple platforms like CPU's, GPU's, etc.
2. What is TensorRT? How is it different from TensorFlow?
    - NVIDIA® TensorRT™ is a deep learning platform that optimizes neural network models and speeds up for inference across GPU-accelerated platforms running in the datacenter, embedded and automotive devices.
    - TensorRT deep learning inference optimizer and runtime is integration into TensorFlow, Google's open source machine learning framework to provide optimize performance of models using Tensorflow across GPU-accelerated platforms running in the datacenter.
3. What is ImageNet? How many images does it contain? How many classes?
    - **ImageNet** is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images.
    - Each meaningful concept in WordNet, possibly described by multiple words or word phrases, is called a "synonym set" or "synset". There are more than 100,000 synsets in WordNet, the majority of them are nouns (80,000+).
    - Total number of images: 14,197,122 (Source: www.image-net.org/about-stats)
    - Total number of non-empty synsets: 21841 (Source: www.image-net.org/about-stats)
    - The Tiny **ImageNet** dataset has 100,000 images across 200 **classes**. Each **class** has 500 training images, 50 validation images, and 50 test images. (Source:
4. Please research and explain the differences between MobileNet and GoogleNet (Inception) architectures.
    - MobileNet is a CNN architecture model for Image Classification and Mobile Vision. it uses very less computation power to run or apply transfer learning to.This makes it a perfect fit for Mobile devices,embedded systems and computers without GPU or low computational efficiency with compromising significantly with the accuracy of the results. The core layer of MobileNet is depthwise separable filters, named as Depthwise Separable Convolution. MobileNet also includes two simple global hyper-parameters that perform an efficient trade off between latency and accuracy.
    - Depthwise Separable Convolution is a form of factorized convolution which factorize a standard convolution into a depthwise convolution and a 1×1

convolution called a pointwise convolution. For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution.
    - GoogLeNet is also a convolutional neural network that is 22 layers deep. 1×1 convolution is inserted into the inception module for dimension reduction and global average pooling is used at the end of the network instead of using fully connected layers.
5. In your own words, what is a bottleneck?
    - The bottleneck in a neural network is just a layer with less neurons than the layer below or above it. Having such a layer encourages the network to compress feature representations to best fit in the available space, in order to get the best loss during training.
    - In a CNN (such as Google's Inception network), bottleneck layers are added to reduce the number of feature maps (aka "channels") in the network, which otherwise tend to increase in each layer. This is achieved by using 1x1 convolutions with less output channels than input channels.
6. How is a bottleneck different from the concept of layer freezing?
    - Freezing a layer in the context of neural networks is about controlling the way the weights are updated. When a layer is frozen, it means that the weights cannot be modified further. In case of bottleneck layer, number of features maps are compressed which otherwise tend to increase. For eg: 1x1 convolution
7. In the TF1 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?)
    - Previous layers are pre-trained and cached.It then finds their bottlenecks from the cache, and feeds them into the final layer to get predictions. Those predictions are then compared against the actual labels, and the results of this comparison is used to update the final layer's weights through a backpropagation process.
8. How does a low `--learning_rate` (step 7 of TF1) value (like 0.005) affect the precision? How much longer does training take?
    - Low `--learning_rate` value (like 0.005) increases the precision. It takes little more time to perform the training and precision does not increase as much.
9. How about a `--learning_rate` (step 7 of TF1) of 1.0? Is the precision still good enough to produce a usable graph?
    - Yes..increasing the learning rate to 1.0 will produce a usable graph as precision is lower, but still acceptable

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?
11. Run the TF1 script on the CPU (see instructions above) How does the training time compare to the default network training (section 4)? Why?
12. Try the training again, but this time do `export ARCHITECTURE="inception_v3"` Are CPU and GPU training times different?
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?