Technical Questions

(10 points) **Question 1**:
You have access to an infinite amount of GPUs that have 8 GBs of Memory each, with a deep learning model that you want to train. You try training your model on a batch of size 32, but to your dismay, you get an OOM error (out of memory error) from your GPU. You keep decreasing the batch size, to find that you can only train on a batch size of 1 without going out of memory.

a) (2 points) How many GPUs will you need to train your model for a batch size of 32 ?
‘

Ans: 32 GPUs

b) (4 points) What is the specific PyTorch or TensorFlow module you will use to conduct such a type of training? You may write a maximum of 280 characters about how you will use it.

Ans: In PyTorch we can use nn.DataParallel class which helps distribute training into multiple GPUs in a single machine. An object of the class will be initialized with a nn.Module object representing the network, and a list of GPU IDs, across which the mini-batches have to be parallelized.

c) (4 points) You connect the required GPUs, and try to train your model on them, but you face the exact OOM error again when you go beyond a batch size of 1. When you check the code, you realize the code has access only to one GPU. Given that the hardware is connected OK, what do you check next ? Hint: It’s probably a software issue outside your deep learning library. Write a maximum of 280 characters elaborating on what you check next.

Ans: CUDA_VISIBLE_DEVICES should be checked that it is not set to 0 which means the code has access to only one GPU. By default it can see all GPUs unless it’s assigned to certain gpus. So, to solve this problem we wouldn’t assign CUDA_VISIBLE_DEVICES to any gpus so the code would automatically see all gpus.

(10 points) **Question 2**:
A deep learning developer is working her way through the Simple MNIST Convnet Keras Tutorial. She opens the google colab from the tutorial, and runs the code. But to her surprise: she gets better accuracy than the tutorial from the first epoch! “Wonderful!”, she thinks. On the next day, she tries the exact same code, but gets worse accuracy than the tutorial on the first epoch. She comes to you, the deep learning expert she knows, asking for an explanation for why this happened: why does the exact same code produce different results every time she runs it ?
Write your explanation in no more than 400 characters.
Hint: This is not something specific to the Keras framework, it has to do with how deep learning libraries work in general.

Ans: Completely reproducible results are not guaranteed across deep learning frameworks. Hyperparameter sampling decisions, data shuffling, dropout and initial weights are some

sources of nondeterministic behavior for the platform/framework. A way to control the sources of randomness is to set consistent seeds to seed the random number generator for all packages.