Assignment: Deep Learning Containers Training and Deployment

For your final assignment, this will be similar to your last assignment where you trained an ML model and deployed it in a container.

Requirements

The requirements for the assignment will be A:

1. Create an image for training a Deep Learning model:

- Should be a binary image classifier
- Image should be compatible with GPU
- Training code should use GPU
- Push image to Dockerhub

2. Create an image for deploying model created from the previous step.

- Should be a completely separate image from the one you used to train.
- This should just be a normal CPU image
- You can use `flask` to accept the request and pass the image to the model to be inferred.
- Should use `gunicorn` and/or `nginx` with `flask` and be ready for production.
- Inference should be done on the `/invocations` endpoint.
- Response should be in JSON `{"prediction": <class>}`
- Push image to Dockerhub

Categories for classification

The binary class you've picked for classification should be interesting and not too easy. There should be some level of similarity between the two classes such that training the ML model should not be too easy.

Get your instructor's approval before proceeding with training and submission of this assignment.

It should also be easy for your instructor to find images on the internet to test your model.

Training model

1. You don't necessarily need to do your training/evaluation process on a container. You can iterate on a Sagemaker instance to develop your model. Only requirement is that the final training code should use GPU and be packaged in a GPU enabled image.

2. You can use any other deep learning framework if you want. I used `fastai` for our exercise since it is the fastest way to get started and has online classes if you want to go deeper : course.fast.ai.

3. If you find that you are taking 30 minutes or up to an hour to get a performant model, re-evaluate your approach. The requirements for this exercise should not too difficult that training should take too long or even need hyperparamter tuning (hence why I recommended `fastai`). AWS `p2` instances (or any

instance that uses a `GPU`) are already pretty expensive. If you want to continue practicing deep learning on a GPU for free, I recommend Google Colab which offers free GPU for its free notebooks.

Evaluation

1. Best practices for building images.

- Proper order of `Dockerfile` commands
- Pinned dependency versions (pip/conda)
- Pinned base image version (unless your code only works on the latest base image version, in which case inform your instructor)
- Download security updates and use non root user.
- Model deployment image should only include packages/dependencies needed. Use a smaller base image.

2. Performance of model:

- Instructor will test 10 images for your model on the deployed model on flask. Passing is predicting 6 images out of 10. Higher score if you can get more predictions.