Home Assignment

Background

MNIST is a ML dataset often used for research and learning purposes. It consists of images of handwritten numbers, together with their labels.

One of the problems faced in training ML models, is delivering data to the model, quick enough. Training data needs to be loaded from a resting state, such as S3, and then processed so that it is ready for training, for example cropping an image.

The compute power on a single machine may not be enough, to both train a model and perform the data loading, therefore one way to train is via a heterogeneous training cluster. A heterogeneous cluster is one where there is both a GPU instance and CPU-only instance available during training time. The CPU-only instance loads the dataset and readies it for training and delivers it to the GPU-instance, where it is fed into an ML model.

One way to communicate between the GPU-instance and CPU-instance is via a GRPC service.

Task

Implement a GRPC service, which streams MNIST samples to a client.

Mnist Service

The GRPC service should load the MNIST examples and **stream** them to the client. Each response should include the image, together with its label.

The base service should implement the following proto, you can extend the proto if required.

```
syntax = "proto3";

service MnistService {
    rpc GetTrainingSamples(DataRequest) returns (stream Sample) {}
}

message DataRequest {
}

message Sample {
    bytes image = 1;
    int64 label = 2;
}
```

Client

The client should connect with the running Mnist service and open a stream to get training samples.

Expectations

- **Make sure to include a README, on how to setup and run the service.**
- Both the service and client should run in separate containers
- While you may complete this example in any language, **Python is recommended**.

Some useful links

- https://grpc.io/docs/languages/python/quickstart/
- http://yann.lecun.com/exdb/mnist/
- https://www.tensorflow.org/datasets/catalog/mnist

Things to think about:

- How are you testing?
- How would this service be deployed?
- Does the code conform to the languages coding style?