To execute the code, run

```
source ./scripts/gpu-hyper-tune.sh
```

```
gs://iris-dataset/jobs/iris_20181112_211040
Job [iris_20181112_211040] submitted successfully.
...
INFO    2018-11-12 21:35:36 -0500   ps-replica-2    4   Module completed;
                                                        cleaning up.
INFO    2018-11-12 21:35:36 -0500   ps-replica-2    4   Clean up finished.
INFO    2018-11-12 21:36:18 -0500   service     Finished tearing down
                                                training program.
INFO    2018-11-12 21:36:25 -0500   service     Finished tearing down
                                                training program.
INFO    2018-11-12 21:37:11 -0500   service     Job completed successfully.
INFO    2018-11-12 21:37:11 -0500   service     Job completed successfully.
endTime: '2018-11-12T21:38:26'
jobId: iris_20181112_211040
startTime: '2018-11-12T21:10:47'
state: SUCCEEDED
```

Scikit-learn on Cloud MLE

This section will provide a walk-through of training a Scikit-learn model on Google Cloud MLE using the same Iris dataset example. We'll begin by moving the appropriate data files from the GitHub repository of this book to GCS.

Move the Data Files to GCS

Walk through the following steps to move the data files to GCS:

1. Create bucket to hold the datasets.

   ```
   gsutil mb gs://iris-sklearn
   ```

2. Run the following commands on the terminal to move the training and testing datasets to the buckets:

   Train set features.

   ```
   gsutil cp X_train.csv gs://iris-sklearn
   ```

   Train set targets.

   ```
   gsutil cp y_train.csv gs://iris-sklearn
   ```

   Test sample for online prediction.

   ```
   gsutil cp test-sample.json gs://iris-sklearn
   ```

Prepare the Training Scripts

The code for training a Scikit-learn model on Cloud MLE is also prepared as a python package. The project structure is as follows:

Iris_SklearnCloudML: [project name as parent folder]

- Trainer: [folder containing the model and execution code]
  - __init__.py: [an empty special python file indicating that the containing folder is a Python package]
  - model.py: [file contains the logic of the model written in Scikit-learn]
- scripts: [folder containing scripts to execute jobs on Cloud MLE]
  - single-instance-training.sh: [script to run a single instance training job on Cloud MLE]
  - online-prediction.sh: [script to execute an online prediction job on Cloud MLE]
  - create-prediction-service.sh: [script to create a prediction service on Cloud MLE]
- config.yaml: [configuration file for specifying model version]