```
    os.environ['TF_CPP_MIN_LOG_LEVEL'] = str(
        tf.logging.__dict__[args.verbosity] / 10)

    # Run the training job
    hparams = hparam.HParams(**args.__dict__)
    train_and_evaluate(hparams)
```

Note the following in the preceding code:

- The method '_get_session_config_from_env_var()' defines the configuration for the runtime environment on Cloud MLE for the Estimator.
- The method 'train_and_evaluate()' does a number of orchestration events including
  - Routing training and evaluation datasets to the model function in 'model.py'
  - Setting up the runtime environment of the Estimator
  - Passing hyper-parameters to the Estimator model
- The line of code "if __name__ == '__main__':" defines the entry point of the Python script via the terminal session. In this script, the code will receive inputs from the terminal through the 'argparse.ArgumentParser()' method.

Training on Cloud MLE

The training execution codes are bash commands stored in a shell script. Shell scripts end with the suffix '.sh.'

Running a Single Instance Training Job

The bash codes for executing training on a single instance on Cloud MLE is shown in the following. Change the bucket names accordingly.

```
DATE=`date '+%Y%m%d_%H%M%S'`
export JOB_NAME=iris_$DATE
```

```
export GCS_JOB_DIR=gs://iris-dataset/jobs/$JOB_NAME
export TRAIN_FILE=gs://iris-dataset/train_data.csv
export EVAL_FILE=gs://iris-dataset/test_data.csv

echo $GCS_JOB_DIR

gcloud ai-platform jobs submit training $JOB_NAME \
                                    --stream-logs \
                                    --runtime-version 1.8 \
                                    --job-dir $GCS_JOB_DIR \
                                    --module-name trainer.task \
                                    --package-path trainer/ \
                                    --region us-central1 \
                                    -- \
                                    --train-files $TRAIN_FILE \
                                    --eval-files $EVAL_FILE \
                                    --train-steps 5000 \
                                    --eval-steps 100
```

This code is stored in the file 'single-instance-training.sh' and executed by running the command on the terminal.

```
source ./scripts/single-instance-training.sh

'Output:'
gs://iris-dataset/jobs/iris_20181112_010123
Job [iris_20181112_010123] submitted successfully.
INFO    2018-11-12 01:01:25 -0500   service     Validating job
                                                requirements...
INFO    2018-11-12 01:01:26 -0500   service     Job creation request
                                                has been successfully
                                                validated.
INFO    2018-11-12 01:01:26 -0500   service     Job iris_20181112_010123 is
                                                queued.
INFO    2018-11-12 01:01:26 -0500   service     Waiting for job to be
                                                provisioned.
```

```
INFO    2018-11-12 01:05:32 -0500   service     Waiting for training
                                                program to start.
...
INFO    2018-11-12 01:09:05 -0500   ps-replica-2        Module completed;
                                                        cleaning up.
INFO    2018-11-12 01:09:05 -0500   ps-replica-2        Clean up finished.
INFO    2018-11-12 01:09:55 -0500   service             Finished tearing
                                                        down training
                                                        program.
INFO    2018-11-12 01:10:53 -0500   service             Job completed
                                                        successfully.
endTime: '2018-11-12T01:08:35'
jobId: iris_20181112_010123
startTime: '2018-11-12T01:07:34'
state: SUCCEEDED
```

Running a Distributed Training Job

The code for initiating distributed training on Cloud MLE is shown in the following, and the code is stored in the file 'distributed-training.sh'. For a distributed job, the attribute '- -scale-tier' is set to a tier above the basic machine type. Change the bucket names accordingly.

```
export SCALE_TIER=STANDARD_1 # BASIC | BASIC_GPU | STANDARD_1 | PREMIUM_1 |
BASIC_TPU
DATE=`date '+%Y%m%d_%H%M%S'`
export JOB_NAME=iris_$DATE
export GCS_JOB_DIR=gs://iris-dataset/jobs/$JOB_NAME
export TRAIN_FILE=gs://iris-dataset/train_data.csv
export EVAL_FILE=gs://iris-dataset/test_data.csv

echo $GCS_JOB_DIR

gcloud ai-platform jobs submit training $JOB_NAME \
                                    --stream-logs \
                                    --scale-tier $SCALE_TIER \
                                    --runtime-version 1.8 \
```