Training with GPUs on Cloud MLE

Training models on GPUs can greatly reduce the processing time. In order to use GPUs on Cloud MLE, we make the following changes to our code example:

1. Change the scale tier to **'CUSTOM'.** The CUSTOM tier makes a number of GPU accelerators available, namely:

   a. standard_gpu: A single NVIDIA Tesla K80 GPU

   b. complex_model_m_gpu: Four NVIDIA Tesla K80 GPUs

   c. complex_model_l_gpu: Eight NVIDIA Tesla K80 GPUs

   d. standard_p100: A single NVIDIA Tesla P100 GPU

   e. complex_model_m_p100: Four NVIDIA Tesla P100 GPUs

   f. standard_v100: A single NVIDIA Tesla V100 GPU

   g. large_model_v100: A single NVIDIA Tesla V100 GPU

   h. complex_model_m_v100: Four NVIDIA Tesla V100 GPUs

   i. complex_model_l_v100: Eight NVIDIA Tesla V100 GPUs

2. Add the following parameters to the '.yaml' file to configure the GPU instance.

```
trainingInput:
  scaleTier: CUSTOM
  masterType: complex_model_m_gpu
  workerType: complex_model_m_gpu
  parameterServerType: large_model
  workerCount: 2
  parameterServerCount: 3
```

3. The full configuration file in 'gpu_hptuning_config.yaml' now looks like this:

```
trainingInput:
  scaleTier: CUSTOM
  masterType: complex_model_m_gpu
  workerType: complex_model_m_gpu
  parameterServerType: large_model
  workerCount: 2
  parameterServerCount: 3
  hyperparameters:
    goal: MAXIMIZE
    hyperparameterMetricTag: accuracy
    maxTrials: 4
    maxParallelTrials: 2
    params:
      - parameterName: learning-rate
        type: DOUBLE
        minValue: 0.00001
        maxValue: 0.005
        scaleType: UNIT_LOG_SCALE
      - parameterName: first-layer-size
        type: INTEGER
        minValue: 50
        maxValue: 500
        scaleType: UNIT_LINEAR_SCALE
      - parameterName: num-layers
        type: INTEGER
        minValue: 1
        maxValue: 15
        scaleType: UNIT_LINEAR_SCALE
      - parameterName: scale-factor
        type: DOUBLE
        minValue: 0.1
        maxValue: 1.0
        scaleType: UNIT_REVERSE_LOG_SCALE
```

Note that running GPUs on Cloud MLE is only available in the following regions:

- us-east1
- us-central1
- us-west1

- asia-east1
- europe-west1
- europe-west4

The updated execution code for training with GPUs on Cloud MLE is saved as 'gpu-hyper-tune.sh' (code shown in the following).

```
export SCALE_TIER=CUSTOM
DATE=`date '+%Y%m%d_%H%M%S'`
export JOB_NAME=iris_$DATE
export HPTUNING_CONFIG=gpu_hptuning_config.yaml
export GCS_JOB_DIR=gs://iris-dataset/jobs/$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 \
                                    --config $HPTUNING_CONFIG \
                                    --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
```

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
   ```