```
        maxValue: 15
        scaleType: UNIT_LINEAR_SCALE
      - parameterName: scale-factor
        type: DOUBLE
        minValue: 0.1
        maxValue: 1.0
        scaleType: UNIT_REVERSE_LOG_SCALE
```

Execute Training Job with Hyper-parameter Tuning

Run the following code on the terminal to launch a distributed training job.

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

```
gs://iris-dataset/jobs/iris_20181114_190121
Job [iris_20181114_190121] submitted successfully.
INFO    2018-11-14 12:41:07 -0500   service     Validating job
                                                requirements...
INFO    2018-11-14 12:41:07 -0500   service     Job creation request
                                                has been successfully
                                                validated.
INFO    2018-11-14 12:41:08 -0500   service     Job iris_20181114_190121 is
                                                queued.
INFO    2018-11-14 12:41:18 -0500   service     Waiting for job to be
                                                provisioned.
INFO    2018-11-14 12:41:18 -0500   service     Waiting for job to be
                                                provisioned.
...
INFO    2018-11-14 12:56:38 -0500   service     Finished tearing down
                                                training program.
INFO    2018-11-14 12:56:45 -0500   service     Finished tearing down
                                                training program.
INFO    2018-11-14 12:57:37 -0500   service     Job completed successfully.
INFO    2018-11-14 12:57:43 -0500   service     Job completed successfully.
endTime: '2018-11-14T13:04:34'
jobId: iris_20181114_190121
```

```
startTime: '2018-11-14T12:41:12'
state: SUCCEEDED
```

The job details of the hyper-parameter training job is shown in Figure 41-3.

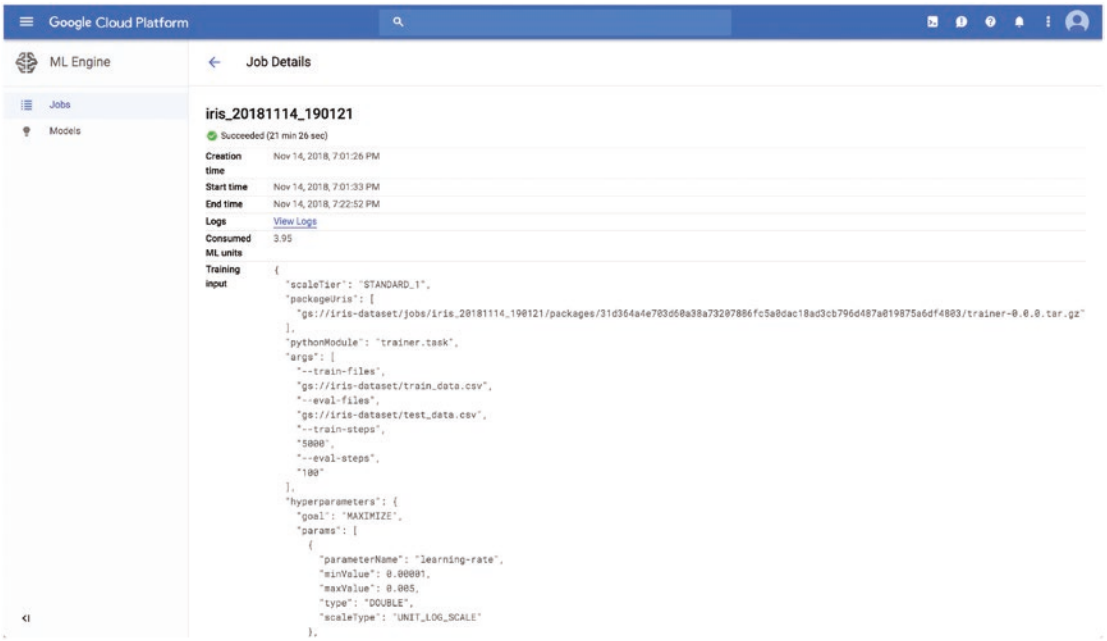

Figure 41-3. *Job details: Hyper-parameter distributed training job on Cloud MLE*

Under **'Training output'**, the first **'trialID'** contains the hyper-parameter set that minimizes the cost function and performs best on the evaluation metric. Observe that the trial run within the red box has the highest accuracy value in the **'objectiveValue'** attribute. This is illustrated in Figure 41-4.

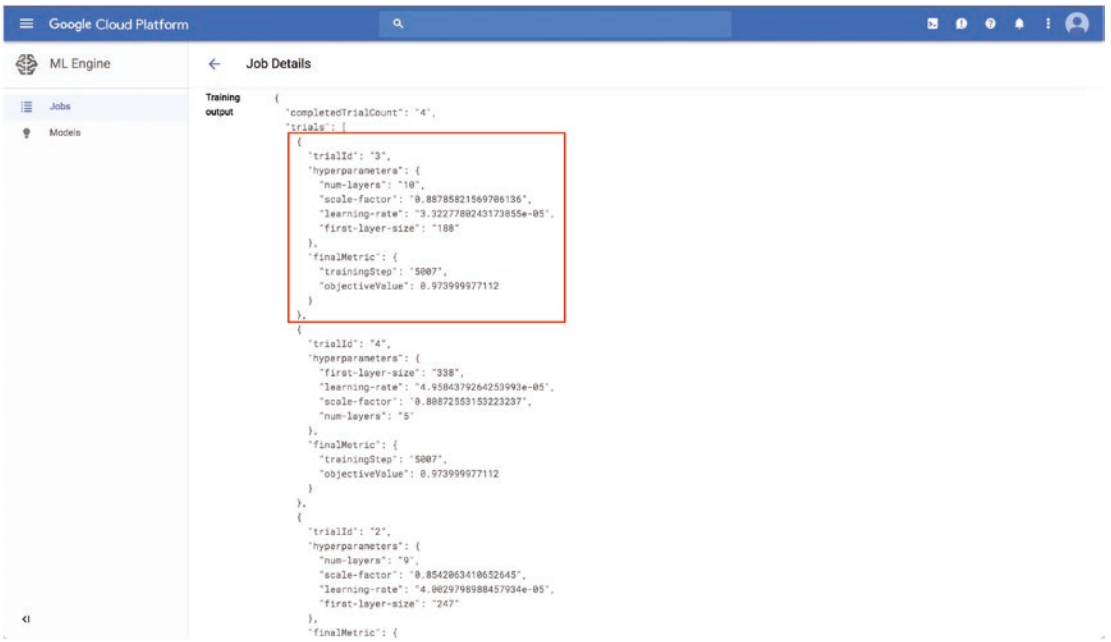

Figure 41-4. *Choosing the best hyper-parameter set*

Making Predictions on Cloud MLE

To make predictions on Cloud MLE, we first create a prediction instance. To do this, run the code in 'create-prediction-service.sh' as shown in the following. The variable 'MODEL_BINARIES' points to the folder location on GCS that stores the trained model for the hyper-parameter setting with '**trialID** = 2'.

```
export MODEL_VERSION=v1
export MODEL_NAME=iris
export MODEL_BINARIES=$GCS_JOB_DIR/3/export/iris/1542241126

# Create a Cloud ML Engine model
gcloud ai-platform models create $MODEL_NAME

# Create a model version
gcloud ai-platform versions create $MODEL_VERSION \
    --model $MODEL_NAME \
    --origin $MODEL_BINARIES \
    --runtime-version 1.8
```