

In order to successfully submit a job to Google Cloud's Machine Learning Engine, a `setup.py` file, which tells Google Cloud to install Keras and other dependencies to run the neural network. A `trainer` folder is created, which contains the main convolutional neural network program file, the google cloud configuration file for GPU and hyperparameter tuning and an empty `__init__.py` file. The `data` folder will contain the pickle file, which is the training dataset and validation dataset loader into the convolutional neural network file. To sync the local machine's files with Google Cloud's storage bucket, simply run `gsutil` command line tool provided by the google command line interface:

trainer folder

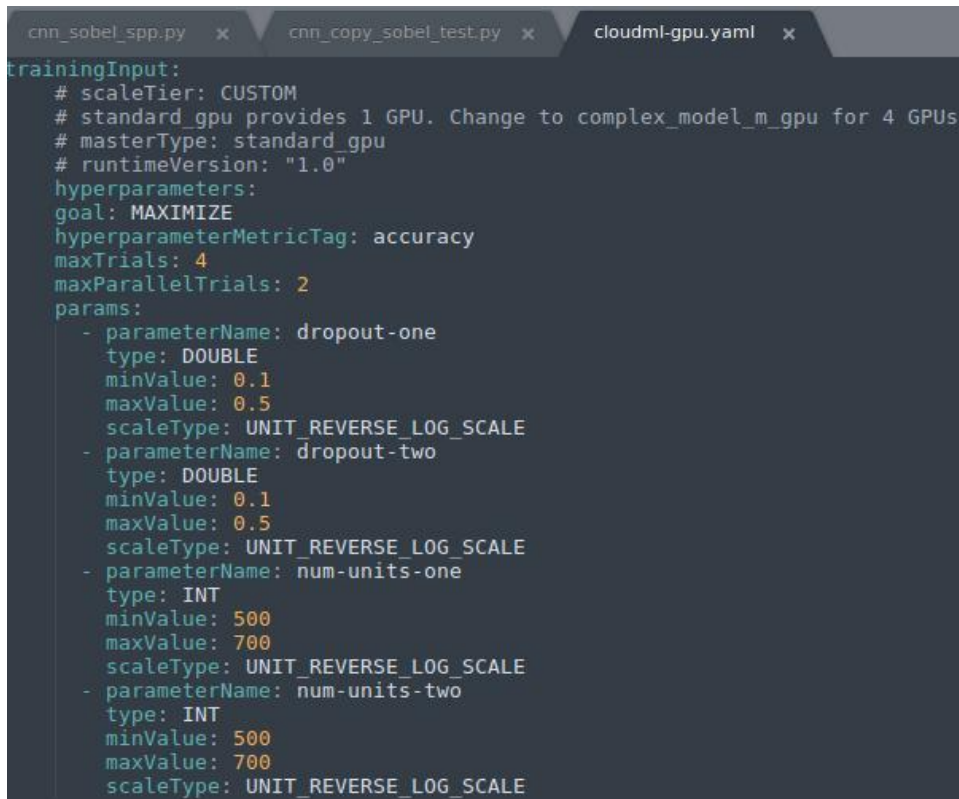
```
gsutil cp -r trainer/cnn_copy_sobel_test.py gs://cnntestbucket/trainer/cnn_copy_sobel_test.py
gsutil cp -r trainer/cloudml-gpu.yaml gs://cnntestbucket/trainer/cloudml-gpu.yaml
gsutil cp -r trainer/__init__.py gs://cnntestbucket/trainer/__init__.py
```

data folder

```
gsutil cp -r data/random_shapes_all.pkl gs://cnntestbucket/data/random_shapes_all.pkl
```

bucket folder

```
gsutil cp -r setup.py gs://cnntestbucket/setup.py
```



```
cnn_sobel_spp.py x  cnn_copy_sobel_test.py x  cloudml-gpu.yaml x
trainingInput:
# scaleTier: CUSTOM
# standard gpu provides 1 GPU. Change to complex_model_m_gpu for 4 GPUs
# masterType: standard gpu
# runtimeVersion: "1.0"
hyperparameters:
goal: MAXIMIZE
hyperparameterMetricTag: accuracy
maxTrials: 4
maxParallelTrials: 2
params:
- parameterName: dropout-one
  type: DOUBLE
  minValue: 0.1
  maxValue: 0.5
  scaleType: UNIT_REVERSE_LOG_SCALE
- parameterName: dropout-two
  type: DOUBLE
  minValue: 0.1
  maxValue: 0.5
  scaleType: UNIT_REVERSE_LOG_SCALE
- parameterName: num-units-one
  type: INT
  minValue: 500
  maxValue: 700
  scaleType: UNIT_REVERSE_LOG_SCALE
- parameterName: num-units-two
  type: INT
  minValue: 500
  maxValue: 700
  scaleType: UNIT_REVERSE_LOG_SCALE
```

```

["Cloud ML Engine package configuration."
from setuptools import setup, find_packages

setup(name='trainer',
      version='1.0',
      packages=find_packages(),
      include_package_data=True,
      description='MNIST MLP keras model on Cloud ML Engine',
      author='Maggie Cao',
      author_email='mahgieeee@hotmail.com',
      license='MIT',
      install_requires=[
          'keras',
          'h5py',
          'pillow',
          'joblib',
          'opencv-python'],
      zip_safe=False)

```

After the files are synced into Google Cloud's storage bucket export variables in the local machine:

```

export BUCKET_NAME=cnntestbucket
export JOB_NAME="cnn_sobel_tuning$(date +%Y%m%d_%H%M%S)"
export JOB_DIR=gs://$BUCKET_NAME/$JOB_NAME
export REGION=us-east1

```

The command line below will train on the local machine, which is not on google cloud. This example includes hyperparameter tuning:

```

gcloud ml-engine local train \
  --job-dir $JOB_DIR \
  --module-name trainer.cnn_copy_sobel_test \
  --package-path ./trainer \
  --configuration trainer/cloudml-gpu.yaml \
  -- \
  --train-file ./data/random_shapes_all.pkl \
  --dropout-one 0.2 \
  --dropout-two 0.2

```

This command will submit a job to Google Cloud's Machine Learning Engine with GPU support. If GPU support isn't needed just remove the line that contains config trainer/cloudml-gpu.yaml:

```

gcloud ml-engine jobs submit training $JOB_NAME \
  --job-dir $JOB_DIR \
  --runtime-version 1.0 \
  --module-name trainer.cnn_copy_sobel \
  --package-path ./trainer \

```

```
--region $REGION \  
--config trainer/cloudml-gpu.yaml \  
-- \  
--train-file gs://$BUCKET_NAME/data/random_shapes.pkl
```