

MNIST Image Classification with TensorFlow on Cloud ML Engine

This notebook demonstrates how to implement different image models on MNIST using Estimator.

Note the MODEL_TYPE; change it to try out different models

```
import os
PROJECT = 'qwiklabs-gcp-32b5e449498c3d9f' # REPLACE WITH YOUR PROJECT ID
BUCKET = 'qwiklabs-gcp-32b5e449498c3d9f' # REPLACE WITH YOUR BUCKET NAME
REGION = 'us-central1' # REPLACE WITH YOUR BUCKET REGION e.g. us-central1
MODEL_TYPE='dnn_dropout' # 'linear', 'dnn', 'dnn_dropout', or 'cnn'
```

do not change these

```
os.environ['PROJECT'] = PROJECT
os.environ['BUCKET'] = BUCKET
os.environ['REGION'] = REGION
os.environ['MODEL_TYPE'] = MODEL_TYPE
os.environ['TFVERSION'] = '1.8' # Tensorflow version
```

```
%%bash
gcloud config set project $PROJECT
gcloud config set compute/region $REGION
```

Updated property [core/project].

Updated property [compute/region].

Run as a Python module

In the previous notebook (mnist_linear.ipynb) we ran our code directly from the notebook.

Now since we want to run our code on Cloud ML Engine, we've packaged it as a python module.

The model.py and task.py containing the model code is in mnistmodel/trainer (mnistmodel/trainer)

Complete the TODOs in model.py before proceeding!

Once you've completed the TODOs, set MODEL_TYPE and run it locally for a few steps to test the code.

```
%%bash
rm -rf mnistmodel.tar.gz mnist_trained
gcloud ml-engine local train \
  --module-name=trainer.task \
  --package-path=${PWD}/mnistmodel/trainer \
  -- \
  --output_dir=${PWD}/mnist_trained \
  --train_steps=100 \
  --learning_rate=0.01 \
  --model=$MODEL_TYPE
```

Extracting mnist/data/train-images-idx3-ubyte.gz

Extracting mnist/data/train-labels-idx1-ubyte.gz

Extracting mnist/data/t10k-images-idx3-ubyte.gz

Extracting mnist/data/t10k-labels-idx1-ubyte.gz

/usr/local/envs/py3env/lib/python3.5/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.

from ._conv import register_converters as _register_converters

WARNING:tensorflow:From /content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnistmodel/trainer/model.py:128: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.

Instructions for updating:

Please use alternatives such as official/mnist/dataset.py from tensorflow/models.

WARNING:tensorflow:From /usr/local/envs/py3env/lib/python3.5/site-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in a future version.

Instructions for updating:

Please write your own downloading logic.

WARNING:tensorflow:From /usr/local/envs/py3env/lib/python3.5/site-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.

Instructions for updating:

Please use tf.data to implement this functionality.

WARNING:tensorflow:From /usr/local/envs/py3env/lib/python3.5/site-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.

Instructions for updating:

Please use tf.data to implement this functionality.

WARNING:tensorflow:From /usr/local/envs/py3env/lib/python3.5/site-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.

Instructions for updating:

Please use tf.one_hot on tensors.

WARNING:tensorflow:From /usr/local/envs/py3env/lib/python3.5/site-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: DataSet.__init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.

Instructions for updating:

Please use alternatives such as official/mnist/dataset.py from tensorflow/models.

INFO:tensorflow:TF_CONFIG environment variable: {'environment': 'cloud', 'cluster': {}, 'job': {'args': ['--output_dir=/content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained', '--train_steps=100', '--learning_rate=0.01', '--model=dnn_dropout'], 'job_name': 'trainer.task'}, 'task': {}}

INFO:tensorflow:Using config: {'_num_worker_replicas': 1, '_task_id': 0, '_task_type': 'worker', '_session_config': None, '_cluster_spec': <tensorflow.python.training.server_lib.ClusterSpec object at 0x7f08109f1828>, '_global_id_in_cluster': 0, '_evaluation_master': '', '_service': None, '_save_checkpoints_secs': 60, '_is_chief': True, '_train_distribute': None, '_keep_checkpoint_every_n_hours': 10000, '_num_ps_replicas': 0, '_model_dir': '/content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained/', '_save_summary_steps': 100, '_tf_random_seed': None, '_log_step_count_steps': 100, '_master': '', '_save_checkpoints_steps': None, '_keep_checkpoint_max': 5}

INFO:tensorflow:Running training and evaluation locally (non-distributed).
INFO:tensorflow:Start train and evaluate loop. The evaluate will happen after 60 secs (eval_spec.throttle_secs) or training is finished.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 1 into /content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained/model.ckpt.
INFO:tensorflow:step = 1, loss = 2.3723686
INFO:tensorflow:Saving checkpoints for 100 into /content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained/model.ckpt.
INFO:tensorflow:Loss for final step: 0.21053186.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2019-04-03-18:04:32
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from /content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained/model.ckpt-100
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2019-04-03-18:04:32
INFO:tensorflow:Saving dict for global step 100: accuracy = 0.9215, global_step = 100, loss = 0.24753337
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['classes', 'serving_default']
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Restoring parameters from /content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained/model.ckpt-100
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: b"/content/datalab/notebooks/training-data-analyst/courses/machine_learning/deepdive/08_image/labs/mnist_trained/export/exporter/temp-b'1554314672'/saved_model.pb"

Now, let's do it on Cloud ML Engine so we can train on GPU: `--scale-tier=BASIC_GPU`

Note the GPU speed up depends on the model type. You'll notice the more complex CNN model trains significantly faster on GPU, however the speed up on the simpler models is not as pronounced.

```
%%bash
OUTDIR=gs://${BUCKET}/mnist/trained_${MODEL_TYPE}
JOBNAME=mnist_${MODEL_TYPE}_${date -u +%y%m%d_%H%M%S}
echo $OUTDIR $REGION $JOBNAME
gsutil -m rm -rf $OUTDIR
gcloud ml-engine jobs submit training $JOBNAME \
  --region=$REGION \
  --module-name=trainer.task \
  --package-path=${PWD}/mnistmodel/trainer \
  --job-dir=$OUTDIR \
  --staging-bucket=gs://${BUCKET} \
  --scale-tier=BASIC_GPU \
  --runtime-version=$TFVERSION \
  -- \
  --output_dir=$OUTDIR \
  --train_steps=10000 --learning_rate=0.01 --train_batch_size=512 \
  --model=$MODEL_TYPE --batch_norm
```


gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout us-central1 mnist_dnn_dropout_190403_180816
jobId: mnist_dnn_dropout_190403_180816
state: QUEUED

Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/#1554314881213689...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/checkpoint#1554314882816805...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/eval/#1554314885635371...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/eval/events.out.tfevents.1554314885.cmle-training-100988377467864037#1554314886388298...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/events.out.tfevents.1554314875.cmle-training-100988377467864037#1554314876333539...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/#1554314887653893...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/#1554314887817309...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554314886/#1554314891750677...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554314886/saved_model.pb#1554314891894163...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554314886/variables/#1554314892104771...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554314886/variables/variables.data-00000-of-00001#1554314892341332...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554314886/variables/variables.index#1554314892518313...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/graph.pbtxt#1554314878510706...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/model.ckpt-1.data-00000-of-00001#1554314881672733...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/model.ckpt-1.index#1554314881924447...
Removing gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/model.ckpt-1.meta#1554314883521279...
/ [16/16 objects] 100% Done
Operation completed over 16 objects.
Job [mnist_dnn_dropout_190403_180816] submitted successfully.
Your job is still active. You may view the status of your job with the command

```
$ gcloud ml-engine jobs describe mnist_dnn_dropout_190403_180816
```

or continue streaming the logs with the command

```
$ gcloud ml-engine jobs stream-logs mnist_dnn_dropout_190403_180816
```

Monitoring training with TensorBoard

Use this cell to launch tensorboard

```
from google.datalab.ml import TensorBoard
TensorBoard().start('gs://{}/mnist/trained_{}'.format(BUCKET, MODEL_TYPE))
```

TensorBoard was started successfully with pid 5119. Click [here \(/ _proxy/38601/\)](#) to access it.

5119

```
for pid in TensorBoard.list()['pid']:
    TensorBoard().stop(pid)
    print('Stopped TensorBoard with pid {}'.format(pid))
```

Stopped TensorBoard with pid 5119

Here are my results:

Model	Accuracy	Time taken	Model description	Run time parameters
linear	91.53	3 min	linear	100 steps, LR=0.01, Batch=512
linear	92.73	8 min	linear	1000 steps, LR=0.01, Batch=512
linear	92.29	18 min	linear	10000 steps, LR=0.01, Batch=512
dnn	98.14	15 min	300-100-30 nodes fully connected	10000 steps, LR=0.01, Batch=512
dnn	97.99	48 min	300-100-30 nodes fully connected	100000 steps, LR=0.01, Batch=512
dnn_dropout	97.84	29 min	300-100-30-DL(0.1)- nodes	20000 steps, LR=0.01, Batch=512
cnn	98.97	35 min	maxpool(10 5x5 cnn, 2)-maxpool(20 5x5 cnn, 2)-300-DL(0.25)	20000 steps, LR=0.01, Batch=512
cnn	98.93	35 min	maxpool(10 11x11 cnn, 2)-maxpool(20 3x3 cnn, 2)-300-DL(0.25)	20000 steps, LR=0.01, Batch=512
cnn	99.17	35 min	maxpool(10 11x11 cnn, 2)-maxpool(20 3x3 cnn, 2)-300-DL(0.25), batch_norm (logits only)	20000 steps, LR=0.01, Batch=512
cnn	99.27	35 min	maxpool(10 11x11 cnn, 2)-maxpool(20 3x3 cnn, 2)-300-DL(0.25), batch_norm (logits, deep)	10000 steps, LR=0.01, Batch=512
cnn	99.48	12 hr	as-above but nfil1=20, nfil2=27, dprob=0.1, lr=0.001, batchsize=233	(hyperparameter optimization)

Create a table to keep track of your own results as you experiment with model type and hyperparameters!

Deploying and predicting with model

Deploy the model:

```
%%bash
MODEL_NAME="mnist2"
MODEL_VERSION=${MODEL_TYPE}
MODEL_LOCATION=$(gsutil ls gs://${BUCKET}/mnist/trained_${MODEL_TYPE}/export/exporter | tail -1)
echo $MODEL_LOCATION
echo "Deleting and deploying $MODEL_NAME $MODEL_VERSION from $MODEL_LOCATION ... this will take a few minutes"
#gcloud ml-engine versions delete ${MODEL_VERSION} --model ${MODEL_NAME}
#gcloud ml-engine models delete ${MODEL_NAME}
gcloud ml-engine models create ${MODEL_NAME} --regions $REGION
gcloud ml-engine versions create ${MODEL_VERSION} --model ${MODEL_NAME} --origin ${MODEL_LOCATION} --runtime-version=$TFVERSIO
N

gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554315054/
Deleting and deploying mnist2 dnn_dropout from gs://qwiklabs-gcp-32b5e449498c3d9f/mnist/trained_dnn_dropout/export/exporter/1554315054/ ... t
his will take a few minutes

Created ml engine model [projects/qwiklabs-gcp-32b5e449498c3d9f/models/mnist2].
Creating version (this might take a few minutes).....
.....
done.
```

To predict with the model, let's take one of the example images.

```
import json, codecs
import matplotlib.pyplot as plt
from tensorflow.examples.tutorials.mnist import input_data

HEIGHT=28
WIDTH=28

mnist = input_data.read_data_sets('mnist/data', one_hot=True, reshape=False)
IMGNO=45 #CHANGE THIS to get different images
jsondata = {'image': mnist.test.images[IMGNO].reshape(HEIGHT, WIDTH).tolist()}
json.dump(jsondata, codecs.open('test.json', 'w', encoding='utf-8'))
plt.imshow(mnist.test.images[IMGNO].reshape(HEIGHT, WIDTH));
```

Extracting mnist/data/train-images-idx3-ubyte.gz

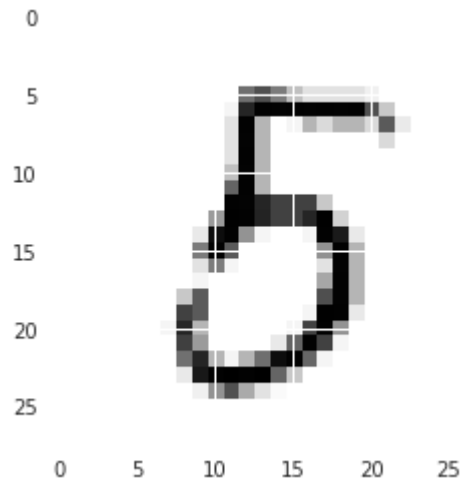
Extracting mnist/data/train-labels-idx1-ubyte.gz

Extracting mnist/data/t10k-images-idx3-ubyte.gz

Extracting mnist/data/t10k-labels-idx1-ubyte.gz

/usr/local/envs/py3env/lib/python3.5/site-packages/matplotlib/font_manager.py:1320: UserWarning: findfont: Font family ['sans-serif'] not found. Falling back to DejaVu Sans

(prop.get_family(), self.defaultFamily[fonttext]))



Send it to the prediction service

```
%%bash
gcloud ml-engine predict \
  --model=mnist2 \
  --version=${MODEL_TYPE} \
  --json-instances=./test.json
```

CLASSES PROBABILITIES

5 [1.711910326768607e-36, 0.0, 0.0, 1.7596145054468512e-21, 0.0, 1.0, 1.021857953419327e-21, 0.0, 2.767283688023675e-26, 5.796368521376454e-32]

Copyright 2017 Google Inc. All Rights Reserved.

#

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

#

<http://www.apache.org/licenses/LICENSE-2.0>

#

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.