Time series prediction using RNNs, with TensorFlow and Cloud ML Engine

This notebook illustrates:

1. Creating a Recurrent Neural Network in TensorFlow
2. Creating a Custom Estimator in tf.estimator
3. Training on Cloud ML Engine

Simulate some time-series data

Essentially a set of sinusoids with random amplitudes and frequencies.

In [1]:

```
import os
PROJECT = 'qwiklabs-gcp-01-9d4e02eca3d3' # REPLACE WITH YOUR PROJECT ID
BUCKET = 'qwiklabs-gcp-01-9d4e02eca3d3' # REPLACE WITH YOUR BUCKET NAME
REGION = 'us-central1' # REPLACE WITH YOUR BUCKET REGION e.g. us-central1
os.environ['TFVERSION'] = '1.8'  # Tensorflow version
```

In [2]:

```
# for bash
os.environ['PROJECT'] = PROJECT
os.environ['BUCKET'] = BUCKET
os.environ['REGION'] = REGION
```

In [3]:

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

```
Updated property [core/project].
Updated property [compute/region].
```

In [4]:

```
import tensorflow as tf
print(tf.__version__)
```

```
1.15.2
```

In [5]:

```
import numpy as np
import seaborn as sns
import pandas as pd

SEQ_LEN = 10
def create_time_series():
  freq = (np.random.random() * 0.5) + 0.1  # 0.1 to 0.6
  ampl = np.random.random() + 0.5  # 0.5 to 1.5
  x = np.sin(np.arange(0, SEQ_LEN) * freq) * ampl
  return x

for i in range(0, 5):
  sns.tsplot( create_time_series() );  # 5 series
```

```
---------------------------------------------------------------------------
-------
AttributeError                            Traceback (most recent cal
l last)
<ipython-input-5-17b3327d14b1> in <module>
     11
     12 for i in range(0, 5):
---> 13   sns.tsplot( create_time_series() );  # 5 series

AttributeError: module 'seaborn' has no attribute 'tsplot'
```

In [6]:

```
def to_csv(filename, N):
  with open(filename, 'w') as ofp:
    for lineno in range(0, N):
      seq = create_time_series()
      line = ",".join(map(str, seq))
      ofp.write(line + '\n')

to_csv('train.csv', 1000)  # 1000 sequences
to_csv('valid.csv',  50)
```

In [7]:

```
!head -5 train.csv valid.csv
```

```
==> train.csv <==
0.0,0.7835257568060624,1.3028590345672058,1.3828888750150852,0.99663
04831230729,0.27432407271877035,-0.540480071427818,-1.1730428933146
8,-1.4100742252424463,-1.171650855179664
0.0,0.17322099198886648,0.3429575185097825,0.5057952066364573,0.6584
584585637777,0.7978763426254479,0.9212443673043033,1.026080895606425
8,1.110277064983147,1.172139208625621
0.0,0.2815890058048289,0.5402251254962078,0.7548264088639949,0.90790
02737568997,0.9869693616296545,0.9855885921288892,0.903870514553214
9,0.7484761337335679,0.5320719581347553
0.0,0.11308511436789251,0.2248347065778442,0.33392902683456616,0.439
0796837982897,0.5390448603378915,0.6326439792479864,0.71877164572956
04,0.7964107019750557,0.8646442396847629
0.0,0.2150923017693087,0.40513266736660974,0.5479869754408063,0.6270
16894548454,0.6330177601205385,0.5652906475556116,0.431723776409344
4,0.2478737644831486,0.03515373872815868

==> valid.csv <==
0.0,0.32479429404936555,0.6091205572484247,0.8175529086580774,0.9241
215360569126,0.9155484275455114,0.7929017571222435,0.571462794604065
3,0.27882192236784037,-0.04855901261840499
0.0,0.10268709494441687,0.20340192132594506,0.30021009112828506,0.39
12522498723245,0.47477978846861935,0.5491884280289793,0.613049032587
5687,0.665135057924336,0.7044461092925692
0.0,0.18718921679164938,0.3636714689204388,0.5193522153142289,0.6453
267322987856,0.7343894514931276,0.781446108368901 3,0.783805127145424
5,0.7413315752068546,0.6564548810614512
0.0,0.42737469522168536,0.818268217424057,1.1393134665540883,1.36310
5667805292,1.4705416732713952,1.452450627616941,1.3103768023535558,
1.0564477751289063,0.7123392063889237
0.0,0.3782568175050376,0.6930839204531855,0.8916878580510053,0.9407
648279342158,0.8320851223356872,0.5838732204647514,0.237751729046770
36,-0.14823830470055482,-0.5093702852615594
```

RNN

For more info, see:

1. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ for the theory
2. https://www.tensorflow.org/tutorials/recurrent for explanations
3. https://github.com/tensorflow/models/tree/master/tutorials/rnn/ptb for sample code

Here, we are trying to predict from 9 values of a timeseries, the tenth value.

Imports

Several tensorflow packages and shutil

In [8]:

```
import tensorflow as tf
import shutil
import tensorflow.contrib.metrics as metrics
import tensorflow.contrib.rnn as rnn
```

```
WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.
0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/2018090
7-contrib-sunset.md
  * https://github.com/tensorflow/addons
  * https://github.com/tensorflow/io (for I/O related ops)
If you depend on functionality not listed there, please file an issu
e.
```

Input Fn to read CSV

Our CSV file structure is quite simple -- a bunch of floating point numbers (note the type of DEFAULTS). We ask for the data to be read BATCH_SIZE sequences at a time. The Estimator API in tf.contrib.learn wants the features returned as a dict. We'll just call this timeseries column 'rawdata'.

Our CSV file sequences consist of 10 numbers. We'll assume that 9 of them are inputs and we need to predict the last one.

In [9]:

```
DEFAULTS = [[0.0] for x in range(0, SEQ_LEN)]
BATCH_SIZE = 20
TIMESERIES_COL = 'rawdata'
# In each sequence, column index 0 to N_INPUTS - 1 are features, and column inde
x N_INPUTS to SEQ_LEN are labels
N_OUTPUTS = 1
N_INPUTS = SEQ_LEN - N_OUTPUTS
```

Reading data using the Estimator API in tf.estimator requires an input_fn. This input_fn needs to return a dict of features and the corresponding labels.

So, we read the CSV file. The Tensor format here will be a scalar -- entire line. We then decode the CSV. At this point, all_data will contain a list of scalar Tensors. There will be SEQ_LEN of these tensors.

We split this list of SEQ_LEN tensors into a list of N_INPUTS Tensors and a list of N_OUTPUTS Tensors. We stack them along the first dimension to then get a vector Tensor for each. We then put the inputs into a dict and call it features. The other is the ground truth, so labels.

In [10]:

```
# Read data and convert to needed format
def read_dataset(filename, mode, batch_size = 512):
  def _input_fn():
    # Provide the ability to decode a CSV
    def decode_csv(line):
      # all_data is a list of scalar tensors
      all_data = tf.decode_csv(line, record_defaults = DEFAULTS)
      inputs = all_data[:len(all_data) - N_OUTPUTS]  # first N_INPUTS values
      labels = all_data[len(all_data) - N_OUTPUTS:] # last N_OUTPUTS values

      # Convert each list of rank R tensors to one rank R+1 tensor
      inputs = tf.stack(inputs, axis = 0)
      labels = tf.stack(labels, axis = 0)

      # Convert input R+1 tensor into a feature dictionary of one R+1 tensor
      features = {TIMESERIES_COL: inputs}

      return features, labels

    # Create list of files that match pattern
    file_list = tf.gfile.Glob(filename)

    # Create dataset from file list
    dataset = tf.data.TextLineDataset(file_list).map(decode_csv)

    if mode == tf.estimator.ModeKeys.TRAIN:
        num_epochs = None # indefinitely
        dataset = dataset.shuffle(buffer_size = 10 * batch_size)
    else:
        num_epochs = 1 # end-of-input after this

    dataset = dataset.repeat(num_epochs).batch(batch_size)

    iterator = dataset.make_one_shot_iterator()
    batch_features, batch_labels = iterator.get_next()
    return batch_features, batch_labels
  return _input_fn
```

Define RNN

A recursive neural network consists of possibly stacked LSTM cells.

The RNN has one output per input, so it will have 8 output cells. We use only the last output cell, but rather use it directly, we do a matrix multiplication of that cell by a set of weights to get the actual predictions. This allows for a degree of scaling between inputs and predictions if necessary (we don't really need it in this problem).

Finally, to supply a model function to the Estimator API, you need to return a EstimatorSpec. The rest of the function creates the necessary objects.

In [11]:

```
LSTM_SIZE = 3  # number of hidden layers in each of the LSTM cells

# Create the inference model
def simple_rnn(features, labels, mode):
  # 0. Reformat input shape to become a sequence
  x = tf.split(features[TIMESERIES_COL], N_INPUTS, 1)

  # 1. Configure the RNN
  lstm_cell = rnn.BasicLSTMCell(LSTM_SIZE, forget_bias = 1.0)
  outputs, _ = rnn.static_rnn(lstm_cell, x, dtype = tf.float32)

  # Slice to keep only the last cell of the RNN
  outputs = outputs[-1]

  # Output is result of linear activation of last layer of RNN
  weight = tf.get_variable("weight", initializer=tf.initializers.random_normal,
shape=[LSTM_SIZE, N_OUTPUTS])
  bias = tf.get_variable("bias", initializer=tf.initializers.random_normal, shap
e=[N_OUTPUTS])
  predictions = tf.matmul(outputs, weight) + bias

  # 2. Loss function, training/eval ops
  if mode == tf.estimator.ModeKeys.TRAIN or mode == tf.estimator.ModeKeys.EVAL:
    loss = tf.losses.mean_squared_error(labels, predictions)
    train_op = tf.contrib.layers.optimize_loss(
      loss = loss,
      global_step = tf.train.get_global_step(),
      learning_rate = 0.01,
      optimizer = "SGD")
    eval_metric_ops = {
      "rmse": tf.metrics.root_mean_squared_error(labels, predictions)
    }
  else:
    loss = None
    train_op = None
    eval_metric_ops = None

  # 3. Create predictions
  predictions_dict = {"predicted": predictions}

  # 4. Create export outputs
  export_outputs = {"predict_export_outputs": tf.estimator.export.PredictOutput(
outputs = predictions)}

  # 5. Return EstimatorSpec
  return tf.estimator.EstimatorSpec(
      mode = mode,
      predictions = predictions_dict,
      loss = loss,
      train_op = train_op,
      eval_metric_ops = eval_metric_ops,
      export_outputs = export_outputs)
```

Estimator

Distributed training is launched off using an Estimator. The key line here is that we use tf.estimator.Estimator rather than, say tf.estimator.DNNRegressor. This allows us to provide a model_fn, which will be our RNN defined above. Note also that we specify a serving_input_fn -- this is how we parse the input data provided to us at prediction time.

In [12]:

```
# Create functions to read in respective datasets
def get_train():
  return read_dataset(filename = 'train.csv', mode = tf.estimator.ModeKeys.TRAIN
, batch_size = 512)

def get_valid():
  return read_dataset(filename = 'valid.csv', mode = tf.estimator.ModeKeys.EVAL,
batch_size = 512)
```

In [13]:

```
# Create serving input function
def serving_input_fn():
  feature_placeholders = {
      TIMESERIES_COL: tf.placeholder(tf.float32, [None, N_INPUTS])
  }

  features = {
    key: tf.expand_dims(tensor, -1)
    for key, tensor in feature_placeholders.items()
  }
  features[TIMESERIES_COL] = tf.squeeze(features[TIMESERIES_COL], axis = [2])

  return tf.estimator.export.ServingInputReceiver(features, feature_placeholders
)
```

In [14]:

```
# Create custom estimator's train and evaluate function
def train_and_evaluate(output_dir):
  estimator = tf.estimator.Estimator(model_fn = simple_rnn,
                                     model_dir = output_dir)
  train_spec = tf.estimator.TrainSpec(input_fn = get_train(),
                                      max_steps = 1000)
  exporter = tf.estimator.LatestExporter('exporter', serving_input_fn)
  eval_spec = tf.estimator.EvalSpec(input_fn = get_valid(),
                                    steps = None,
                                    exporters = exporter)
  tf.estimator.train_and_evaluate(estimator, train_spec, eval_spec)
```

In [15]:

```
# Run the model
shutil.rmtree('outputdir', ignore_errors = True) # start fresh each time
train_and_evaluate('outputdir')
```

```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_model_dir': 'outputdir', '_tf_rando
m_seed': None, '_save_summary_steps': 100, '_save_checkpoints_step
s': None, '_save_checkpoints_secs': 600, '_session_config': allow_so
ft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 1000
0, '_log_step_count_steps': 100, '_train_distribute': None, '_device
_fn': None, '_protocol': None, '_eval_distribute': None, '_experimen
tal_distribute': None, '_experimental_max_worker_delay_secs': None,
'_session_creation_timeout_secs': 7200, '_service': None, '_cluster_
spec': <tensorflow.python.training.server_lib.ClusterSpec object at
0x7f768eaa5490>, '_task_type': 'worker', '_task_id': 0, '_global_id_
in_cluster': 0, '_master': '', '_evaluation_master': '', '_is_chie
f': True, '_num_ps_replicas': 0, '_num_worker_replicas': 1}
INFO:tensorflow:Not using Distribute Coordinator.
INFO:tensorflow:Running training and evaluation locally (non-distrib
uted).
INFO:tensorflow:Start train and evaluate loop. The evaluate will hap
pen after every checkpoint. Checkpoint frequency is determined based
on RunConfig arguments: save_checkpoints_steps None or save_checkpoi
nts_secs 600.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/training/training_util.py:236: Variable.initialize
d_value (from tensorflow.python.ops.variables) is deprecated and wil
l be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatica
lly both in eager and graph (inside tf.defun) contexts.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/autograph/converters/directives.py:119: The name t
f.decode_csv is deprecated. Please use tf.io.decode_csv instead.

WARNING:tensorflow:From <ipython-input-10-069415855dc2>:34: DatasetV
1.make_one_shot_iterator (from tensorflow.python.data.ops.dataset_op
s) is deprecated and will be removed in a future version.
Instructions for updating:
Use `for ... in dataset:` to iterate over a dataset. If using `tf.es
timator`, return the `Dataset` object directly from your input funct
ion. As a last resort, you can use `tf.compat.v1.data.make_one_shot_
iterator(dataset)`.
INFO:tensorflow:Calling model_fn.
WARNING:tensorflow:From <ipython-input-11-a1f8e62ab687>:9: BasicLSTM
Cell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecat
ed and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be re
placed by that in Tensorflow 2.0.
WARNING:tensorflow:From <ipython-input-11-a1f8e62ab687>:10: static_r
nn (from tensorflow.python.ops.rnn) is deprecated and will be remove
d in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell, unroll=True)`, which is equivalen
t to this API
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/rnn_cell_impl.py:735: Layer.add_variable (from
tensorflow.python.keras.engine.base_layer) is deprecated and will be
```

```
removed in a future version.
Instructions for updating:
Please use `layer.add_weight` method instead.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/rnn_cell_impl.py:739: calling Zeros.__init__
(from tensorflow.python.ops.init_ops) with dtype is deprecated and w
ill be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/losses/losses_impl.py:121: where (from tensorf
low.python.ops.array_ops) is deprecated and will be removed in a fut
ure version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
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 0 into outputdir/model.ckpt.
INFO:tensorflow:loss = 0.89921373, step = 1
INFO:tensorflow:global_step/sec: 13.5125
INFO:tensorflow:loss = 0.4634918, step = 101 (7.407 sec)
INFO:tensorflow:global_step/sec: 19.5906
INFO:tensorflow:loss = 0.2396803, step = 201 (5.103 sec)
INFO:tensorflow:global_step/sec: 17.8986
INFO:tensorflow:loss = 0.15136671, step = 301 (5.588 sec)
INFO:tensorflow:global_step/sec: 20.3024
INFO:tensorflow:loss = 0.123387426, step = 401 (4.922 sec)
INFO:tensorflow:global_step/sec: 19.8931
INFO:tensorflow:loss = 0.0975287, step = 501 (5.028 sec)
INFO:tensorflow:global_step/sec: 20.1018
INFO:tensorflow:loss = 0.08157629, step = 601 (4.982 sec)
INFO:tensorflow:global_step/sec: 19.7769
INFO:tensorflow:loss = 0.07310486, step = 701 (5.051 sec)
INFO:tensorflow:global_step/sec: 19.4965
INFO:tensorflow:loss = 0.06913547, step = 801 (5.129 sec)
INFO:tensorflow:global_step/sec: 19.5254
INFO:tensorflow:loss = 0.064153396, step = 901 (5.118 sec)
INFO:tensorflow:Saving checkpoints for 1000 into outputdir/model.ckp
t.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-04-14T22:07:16Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from outputdir/model.ckpt-1000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-04-14-22:07:17
INFO:tensorflow:Saving dict for global step 1000: global_step = 100
0, loss = 0.049876522, rmse = 0.22333053
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 100
0: outputdir/model.ckpt-1000
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/saved_model/signature_def_utils_impl.py:201: build
_tensor_info (from tensorflow.python.saved_model.utils_impl) is depr
ecated and will be removed in a future version.
```

```
Instructions for updating:
This function will only be available through the v1 compatibility li
brary as tf.compat.v1.saved_model.utils.build_tensor_info or tf.comp
at.v1.saved_model.build_tensor_info.
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['predict
_export_outputs', 'serving_default']
INFO:tensorflow:Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
INFO:tensorflow:Restoring parameters from outputdir/model.ckpt-1000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: outputdir/export/exporter/tem
p-b'1586902037'/saved_model.pb
INFO:tensorflow:Loss for final step: 0.05545029.
```

Standalone Python module

To train this on Cloud ML Engine, we take the code in this notebook and make a standalone Python module.

In [16]:

```
%%bash
# Run module as-is
echo $PWD
rm -rf outputdir
export PYTHONPATH=${PYTHONPATH}:${PWD}/simplernn
python -m trainer.task \
  --train_data_paths="${PWD}/train.csv*" \
  --eval_data_paths="${PWD}/valid.csv*"  \
  --output_dir=outputdir \
  --job-dir=./tmp
```

```
/home/jupyter/training-data-analyst/courses/machine_learning/deepdiv
e/05_artandscience
```

```
WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.
0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/2018090
7-contrib-sunset.md
  * https://github.com/tensorflow/addons
  * https://github.com/tensorflow/io (for I/O related ops)
If you depend on functionality not listed there, please file an issu
e.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:21: The name tf.logging.set_verbosity is deprecated. Please use t
f.compat.v1.logging.set_verbosity instead.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:21: The name tf.logging.INFO is deprecated. Please use tf.compat.v
1.logging.INFO instead.

INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_model_dir': 'outputdir/', '_tf_rand
om_seed': None, '_save_summary_steps': 100, '_save_checkpoints_step
s': None, '_save_checkpoints_secs': 600, '_session_config': allow_so
ft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 1000
0, '_log_step_count_steps': 100, '_train_distribute': None, '_device
_fn': None, '_protocol': None, '_eval_distribute': None, '_experimen
tal_distribute': None, '_experimental_max_worker_delay_secs': None,
'_session_creation_timeout_secs': 7200, '_service': None, '_cluster_
spec': <tensorflow.python.training.server_lib.ClusterSpec object at
0x7f2d8c83a0d0>, '_task_type': 'worker', '_task_id': 0, '_global_id_
in_cluster': 0, '_master': '', '_evaluation_master': '', '_is_chie
f': True, '_num_ps_replicas': 0, '_num_worker_replicas': 1}
INFO:tensorflow:Not using Distribute Coordinator.
INFO:tensorflow:Running training and evaluation locally (non-distrib
uted).
INFO:tensorflow:Start train and evaluate loop. The evaluate will hap
pen after every checkpoint. Checkpoint frequency is determined based
on RunConfig arguments: save_checkpoints_steps None or save_checkpoi
nts_secs 600.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/training/training_util.py:236: Variable.initialize
d_value (from tensorflow.python.ops.variables) is deprecated and wil
l be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatica
lly both in eager and graph (inside tf.defun) contexts.
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:53: The name tf.gfile.Glob is deprecated. Please use tf.io.gfile.g
lob instead.

WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/autograph/converters/directives.py:119: The name t
```

```
f.decode_csv is deprecated. Please use tf.io.decode_csv instead.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:66: DatasetV1.make_one_shot_iterator (from tensorflow.python.data.
ops.dataset_ops) is deprecated and will be removed in a future versi
on.
Instructions for updating:
Use `for ... in dataset:` to iterate over a dataset. If using `tf.es
timator`, return the `Dataset` object directly from your input funct
ion. As a last resort, you can use `tf.compat.v1.data.make_one_shot_
iterator(dataset)`.
INFO:tensorflow:Calling model_fn.
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:90: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_im
pl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be re
placed by that in Tensorflow 2.0.
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:91: static_rnn (from tensorflow.python.ops.rnn) is deprecated and
will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell, unroll=True)`, which is equivalen
t to this API
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/rnn_cell_impl.py:735: Layer.add_variable (from
tensorflow.python.keras.engine.base_layer) is deprecated and will be
removed in a future version.
Instructions for updating:
Please use `layer.add_weight` method instead.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/rnn_cell_impl.py:739: calling Zeros.__init__
(from tensorflow.python.ops.init_ops) with dtype is deprecated and w
ill be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:98: The name tf.get_variable is deprecated. Please use tf.compat.v
1.get_variable instead.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:106: The name tf.losses.mean_squared_error is deprecated. Please u
se tf.compat.v1.losses.mean_squared_error instead.

WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/losses/losses_impl.py:121: where (from tensorf
low.python.ops.array_ops) is deprecated and will be removed in a fut
ure version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:109: The name tf.train.get_global_step is deprecated. Please use t
f.compat.v1.train.get_global_step instead.
```

```
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:113: The name tf.metrics.root_mean_squared_error is deprecated. Pl
ease use tf.compat.v1.metrics.root_mean_squared_error instead.

INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
2020-04-14 22:07:37.413442: I tensorflow/core/platform/profile_util
s/cpu_utils.cc:94] CPU Frequency: 2200000000 Hz
2020-04-14 22:07:37.413808: I tensorflow/compiler/xla/service/servic
e.cc:168] XLA service 0x558c7bb1ed50 initialized for platform Host
(this does not guarantee that XLA will be used). Devices:
2020-04-14 22:07:37.413841: I tensorflow/compiler/xla/service/servic
e.cc:176]   StreamExecutor device (0): Host, Default Version
2020-04-14 22:07:37.413963: I tensorflow/core/common_runtime/process
_util.cc:136] Creating new thread pool with default inter op settin
g: 2. Tune using inter_op_parallelism_threads for best performance.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 0 into outputdir/model.ckpt.
INFO:tensorflow:loss = 3.4759233, step = 1
INFO:tensorflow:global_step/sec: 13.4466
INFO:tensorflow:loss = 0.13964126, step = 101 (7.437 sec)
INFO:tensorflow:global_step/sec: 17.4043
INFO:tensorflow:loss = 0.086861, step = 201 (5.746 sec)
INFO:tensorflow:global_step/sec: 19.6483
INFO:tensorflow:loss = 0.071839646, step = 301 (5.089 sec)
INFO:tensorflow:global_step/sec: 19.7637
INFO:tensorflow:loss = 0.05955345, step = 401 (5.060 sec)
INFO:tensorflow:global_step/sec: 17.9131
INFO:tensorflow:loss = 0.04962764, step = 501 (5.582 sec)
INFO:tensorflow:global_step/sec: 17.8863
INFO:tensorflow:loss = 0.04475543, step = 601 (5.591 sec)
INFO:tensorflow:global_step/sec: 18.1975
INFO:tensorflow:loss = 0.043421205, step = 701 (5.495 sec)
INFO:tensorflow:global_step/sec: 18.5114
INFO:tensorflow:loss = 0.036736704, step = 801 (5.402 sec)
INFO:tensorflow:global_step/sec: 18.9877
INFO:tensorflow:loss = 0.035597403, step = 901 (5.267 sec)
INFO:tensorflow:Saving checkpoints for 1000 into outputdir/model.ckp
t.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-04-14T22:08:36Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from outputdir/model.ckpt-1000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-04-14-22:08:37
INFO:tensorflow:Saving dict for global step 1000: global_step = 100
0, loss = 0.035899792, rmse = 0.1894724
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 100
0: outputdir/model.ckpt-1000
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:138: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/saved_model/signature_def_utils_impl.py:201: build
_tensor_info (from tensorflow.python.saved_model.utils_impl) is depr
ecated and will be removed in a future version.
Instructions for updating:
This function will only be available through the v1 compatibility li
brary as tf.compat.v1.saved_model.utils.build_tensor_info or tf.comp
at.v1.saved_model.build_tensor_info.
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['predict
_export_outputs', 'serving_default']
INFO:tensorflow:Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
INFO:tensorflow:Restoring parameters from outputdir/model.ckpt-1000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: outputdir/export/exporter/tem
p-b'1586902117'/saved_model.pb
INFO:tensorflow:Loss for final step: 0.034011565.
```

Try out online prediction. This is how the REST API will work after you train on Cloud ML Engine

In [17]:

```
%%writefile test.json
{"rawdata_input": [0,0.214,0.406,0.558,0.655,0.687,0.65,0.549,0.393]}
```

```
Writing test.json
```

In [40]:

```
# local predict doesn't work with Python 3 yet.
# %%bash
# MODEL_DIR=$(ls ./outputdir/export/exporter/)
# gcloud ml-engine local predict --model-dir=./outputdir/export/exporter/$MODEL_
DIR --json-instances=test.json
```

Cloud ML Engine

Now to train on Cloud ML Engine.

In [24]:

```
%%bash
# Run module on Cloud ML Engine
OUTDIR=gs://${BUCKET}/simplernn/model_trained
JOBNAME=simplernn_$(date -u +%y%m%d_%H%M%S)
gsutil -m rm -rf $OUTDIR
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   --package-path=${PWD}/simplernn/trainer \
   --job-dir=$OUTDIR \
   --staging-bucket=gs://$BUCKET \
   --scale-tier=BASIC \
   --runtime-version=1.4 \
   -- \
   --train_data_paths="gs://${BUCKET}/train.csv*" \
   --eval_data_paths="gs://${BUCKET}/valid.csv*"  \
   --output_dir=$OUTDIR
```

```
CommandException: 1 files/objects could not be removed.
WARNING: The `gcloud ml-engine` commands have been renamed and will
soon be removed. Please use `gcloud ai-platform` instead.
ERROR: (gcloud.ml-engine.jobs.submit.training) INVALID_ARGUMENT: Fie
ld: runtime_version Error: The specified runtime version '1.4' with
the Python version '' is not supported or is deprecated.  Please spe
cify a different runtime version. See https://cloud.google.com/ml-en
gine/docs/runtime-version-list for a list of supported versions
- '@type': type.googleapis.com/google.rpc.BadRequest
  fieldViolations:
  - description: The specified runtime version '1.4' with the Python
version '' is
      not supported or is deprecated.  Please specify a different ru
ntime version.
      See https://cloud.google.com/ml-engine/docs/runtime-version-li
st for a list
      of supported versions
    field: runtime_version
```

```
---------------------------------------------------------------------
-------
CalledProcessError                        Traceback (most recent cal
l last)
<ipython-input-24-ee5350fe151e> in <module>
----> 1 get_ipython().run_cell_magic('bash', '', '# Run module on Cl
oud ML Engine\nOUTDIR=gs://${BUCKET}/simplernn/model_trained\nJOBNAM
E=simplernn_$(date -u +%y%m%d_%H%M%S)\ngsutil -m rm -rf $OUTDIR\ngcl
oud ml-engine jobs submit training $JOBNAME \\\n   --region=$REGION
 \\\n   --module-name=trainer.task \\\n   --package-path=${PWD}/simp
lernn/trainer \\\n   --job-dir=$OUTDIR \\\n   --staging-bucket=gs://
$BUCKET \\\n   --scale-tier=BASIC \\\n   --runtime-version=1.4 \\\n
-- \\\n   --train_data_paths="gs://${BUCKET}/train.csv*" \\\n   --ev
al_data_paths="gs://${BUCKET}/valid.csv*"  \\\n   --output_dir=$OUTD
IR\n')

/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshel
l.py in run_cell_magic(self, magic_name, line, cell)
   2360             with self.builtin_trap:
   2361                 args = (magic_arg_s, cell)
-> 2362                 result = fn(*args, **kwargs)
   2363             return result
   2364

/opt/conda/lib/python3.7/site-packages/IPython/core/magics/script.py
in named_script_magic(line, cell)
    140             else:
    141                 line = script
--> 142             return self.shebang(line, cell)
    143
    144         # write a basic docstring:

<decorator-gen-110> in shebang(self, line, cell)

/opt/conda/lib/python3.7/site-packages/IPython/core/magic.py in <lam
bda>(f, *a, **k)
    185     # but it's overkill for just that one bit of state.
    186     def magic_deco(arg):
--> 187         call = lambda f, *a, **k: f(*a, **k)
    188
    189         if callable(arg):

/opt/conda/lib/python3.7/site-packages/IPython/core/magics/script.py
in shebang(self, line, cell)
    243             sys.stderr.flush()
    244         if args.raise_error and p.returncode!=0:
--> 245             raise CalledProcessError(p.returncode, cell, out
put=out, stderr=err)
    246
    247     def _run_script(self, p, cell, to_close):

CalledProcessError: Command 'b'# Run module on Cloud ML Engine\nOUTD
IR=gs://${BUCKET}/simplernn/model_trained\nJOBNAME=simplernn_$(date
 -u +%y%m%d_%H%M%S)\ngsutil -m rm -rf $OUTDIR\ngcloud ml-engine jobs
submit training $JOBNAME \\\n   --region=$REGION \\\n   --module-nam
e=trainer.task \\\n   --package-path=${PWD}/simplernn/trainer \\\n
--job-dir=$OUTDIR \\\n   --staging-bucket=gs://$BUCKET \\\n   --scal
e-tier=BASIC \\\n   --runtime-version=1.4 \\\n   -- \\\n   --train_d
ata_paths="gs://${BUCKET}/train.csv*" \\\n   --eval_data_paths="g
s://${BUCKET}/valid.csv*"  \\\n   --output_dir=$OUTDIR\n'' returned
 non-zero exit status 1.
```

Variant: long sequence

To create short sequences from a very long sequence.

In [20]:

```
import tensorflow as tf
import numpy as np

def breakup(sess, x, lookback_len):
  N = sess.run(tf.size(x))
  windows = [tf.slice(x, [b], [lookback_len]) for b in range(0, N-lookback_len)]
  windows = tf.stack(windows)
  return windows

x = tf.constant(np.arange(1,11, dtype=np.float32))
with tf.Session() as sess:
    print('input=', x.eval())
    seqx = breakup(sess, x, 5)
    print('output=', seqx.eval())
```

```
input= [ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
output= [[1. 2. 3. 4. 5.]
 [2. 3. 4. 5. 6.]
 [3. 4. 5. 6. 7.]
 [4. 5. 6. 7. 8.]
 [5. 6. 7. 8. 9.]]
```

Variant: Keras

You can also invoke a Keras model from within the Estimator framework by creating an estimator from the compiled Keras model:

In [21]:

```
def make_keras_estimator(output_dir):
  from tensorflow import keras
  model = keras.models.Sequential()
  model.add(keras.layers.Dense(32, input_shape=(N_INPUTS,), name=TIMESERIES_INPU
T_LAYER))
  model.add(keras.layers.Activation('relu'))
  model.add(keras.layers.Dense(1))
  model.compile(loss = 'mean_squared_error',
                optimizer = 'adam',
                metrics = ['mae', 'mape']) # mean absolute [percentage] error
  return keras.estimator.model_to_estimator(model)
```

In [22]:

```
%%bash
# Run module as-is
echo $PWD
rm -rf outputdir
export PYTHONPATH=${PYTHONPATH}:${PWD}/simplernn
python -m trainer.task \
  --train_data_paths="${PWD}/train.csv*" \
  --eval_data_paths="${PWD}/valid.csv*"  \
  --output_dir=${PWD}/outputdir \
  --job-dir=./tmp --keras
```

```
/home/jupyter/training-data-analyst/courses/machine_learning/deepdiv
e/05_artandscience
```

```
WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.
0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/2018090
7-contrib-sunset.md
  * https://github.com/tensorflow/addons
  * https://github.com/tensorflow/io (for I/O related ops)
If you depend on functionality not listed there, please file an issu
e.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:21: The name tf.logging.set_verbosity is deprecated. Please use t
f.compat.v1.logging.set_verbosity instead.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:21: The name tf.logging.INFO is deprecated. Please use tf.compat.v
1.logging.INFO instead.

WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/resource_variable_ops.py:1630: calling BaseRes
ourceVariable.__init__ (from tensorflow.python.ops.resource_variable
_ops) with constraint is deprecated and will be removed in a future
version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
INFO:tensorflow:Using default config.
INFO:tensorflow:Using the Keras model provided.
2020-04-14 22:12:34.317781: I tensorflow/core/platform/profile_util
s/cpu_utils.cc:94] CPU Frequency: 2200000000 Hz
2020-04-14 22:12:34.318175: I tensorflow/compiler/xla/service/servic
e.cc:168] XLA service 0x55a886679ee0 initialized for platform Host
(this does not guarantee that XLA will be used). Devices:
2020-04-14 22:12:34.318210: I tensorflow/compiler/xla/service/servic
e.cc:176]   StreamExecutor device (0): Host, Default Version
2020-04-14 22:12:34.318342: I tensorflow/core/common_runtime/process
_util.cc:136] Creating new thread pool with default inter op settin
g: 2. Tune using inter_op_parallelism_threads for best performance.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/init_ops.py:97: calling GlorotUniform.__init__
(from tensorflow.python.ops.init_ops) with dtype is deprecated and w
ill be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/init_ops.py:97: calling Zeros.__init__ (from t
ensorflow.python.ops.init_ops) with dtype is deprecated and will be
removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing
it to the constructor
INFO:tensorflow:Using config: {'_model_dir': '/home/jupyter/training
-data-analyst/courses/machine_learning/deepdive/05_artandscience/out
putdir/', '_tf_random_seed': None, '_save_summary_steps': 100, '_sav
e_checkpoints_steps': None, '_save_checkpoints_secs': 600, '_session
_config': allow_soft_placement: true
graph_options {
  rewrite_options {
```

```
    meta_optimizer_iterations: ONE
  }
}
, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 1000
0, '_log_step_count_steps': 100, '_train_distribute': None, '_device
_fn': None, '_protocol': None, '_eval_distribute': None, '_experimen
tal_distribute': None, '_experimental_max_worker_delay_secs': None,
'_session_creation_timeout_secs': 7200, '_service': None, '_cluster_
spec': <tensorflow.python.training.server_lib.ClusterSpec object at
0x7fd89d7bdfd0>, '_task_type': 'worker', '_task_id': 0, '_global_id_
in_cluster': 0, '_master': '', '_evaluation_master': '', '_is_chie
f': True, '_num_ps_replicas': 0, '_num_worker_replicas': 1}
INFO:tensorflow:Not using Distribute Coordinator.
INFO:tensorflow:Running training and evaluation locally (non-distrib
uted).
INFO:tensorflow:Start train and evaluate loop. The evaluate will hap
pen after every checkpoint. Checkpoint frequency is determined based
on RunConfig arguments: save_checkpoints_steps None or save_checkpoi
nts_secs 600.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/training/training_util.py:236: Variable.initialize
d_value (from tensorflow.python.ops.variables) is deprecated and wil
l be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatica
lly both in eager and graph (inside tf.defun) contexts.
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:53: The name tf.gfile.Glob is deprecated. Please use tf.io.gfile.g
lob instead.

WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/autograph/converters/directives.py:119: The name t
f.decode_csv is deprecated. Please use tf.io.decode_csv instead.

WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:66: DatasetV1.make_one_shot_iterator (from tensorflow.python.data.
ops.dataset_ops) is deprecated and will be removed in a future versi
on.
Instructions for updating:
Use `for ... in dataset:` to iterate over a dataset. If using `tf.es
timator`, return the `Dataset` object directly from your input funct
ion. As a last resort, you can use `tf.compat.v1.data.make_one_shot_
iterator(dataset)`.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Warm-starting with WarmStartSettings: WarmStartSetti
ngs(ckpt_to_initialize_from='/home/jupyter/training-data-analyst/cou
rses/machine_learning/deepdive/05_artandscience/outputdir/keras/kera
s_model.ckpt', vars_to_warm_start='.*', var_name_to_vocab_info={}, v
ar_name_to_prev_var_name={})
INFO:tensorflow:Warm-starting from: /home/jupyter/training-data-anal
yst/courses/machine_learning/deepdive/05_artandscience/outputdir/ker
as/keras_model.ckpt
INFO:tensorflow:Warm-starting variables only in TRAINABLE_VARIABLES.
INFO:tensorflow:Warm-started 4 variables.
INFO:tensorflow:Create CheckpointSaverHook.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/ops/array_ops.py:1475: where (from tensorflow.pyth
on.ops.array_ops) is deprecated and will be removed in a future vers
```

```
ion.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 0 into /home/jupyter/training
-data-analyst/courses/machine_learning/deepdive/05_artandscience/out
putdir/model.ckpt.
INFO:tensorflow:loss = 1.0677292, step = 1
INFO:tensorflow:global_step/sec: 22
INFO:tensorflow:loss = 0.037802376, step = 101 (4.546 sec)
INFO:tensorflow:global_step/sec: 25.0002
INFO:tensorflow:loss = 0.007338594, step = 201 (4.000 sec)
INFO:tensorflow:global_step/sec: 23.3402
INFO:tensorflow:loss = 0.0024417578, step = 301 (4.285 sec)
INFO:tensorflow:global_step/sec: 24.5846
INFO:tensorflow:loss = 0.0012697165, step = 401 (4.068 sec)
INFO:tensorflow:global_step/sec: 26.92
INFO:tensorflow:loss = 0.0008147394, step = 501 (3.715 sec)
INFO:tensorflow:global_step/sec: 24.7521
INFO:tensorflow:loss = 0.00057212624, step = 601 (4.040 sec)
INFO:tensorflow:global_step/sec: 25.8129
INFO:tensorflow:loss = 0.00044583943, step = 701 (3.874 sec)
INFO:tensorflow:global_step/sec: 26.3695
INFO:tensorflow:loss = 0.0003300052, step = 801 (3.792 sec)
INFO:tensorflow:global_step/sec: 26.4128
INFO:tensorflow:loss = 0.00026902207, step = 901 (3.786 sec)
INFO:tensorflow:Saving checkpoints for 1000 into /home/jupyter/train
ing-data-analyst/courses/machine_learning/deepdive/05_artandscience/
outputdir/model.ckpt.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-04-14T22:13:16Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from /home/jupyter/training-dat
a-analyst/courses/machine_learning/deepdive/05_artandscience/outputd
ir/model.ckpt-1000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-04-14-22:13:16
INFO:tensorflow:Saving dict for global step 1000: global_step = 100
0, loss = 0.00023269924, mean_absolute_error = 0.012297125, mean_abs
olute_percentage_error = 4.200561
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 100
0: /home/jupyter/training-data-analyst/courses/machine_learning/deep
dive/05_artandscience/outputdir/model.ckpt-1000
WARNING:tensorflow:From /home/jupyter/training-data-analyst/courses/
machine_learning/deepdive/05_artandscience/simplernn/trainer/model.p
y:138: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
WARNING:tensorflow:From /opt/conda/lib/python3.7/site-packages/tenso
rflow_core/python/saved_model/signature_def_utils_impl.py:201: build
_tensor_info (from tensorflow.python.saved_model.utils_impl) is depr
ecated and will be removed in a future version.
Instructions for updating:
This function will only be available through the v1 compatibility li
brary as tf.compat.v1.saved_model.utils.build_tensor_info or tf.comp
```

```
at.v1.saved_model.build_tensor_info.
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['serving
_default']
INFO:tensorflow:Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
INFO:tensorflow:Restoring parameters from /home/jupyter/training-dat
a-analyst/courses/machine_learning/deepdive/05_artandscience/outputd
ir/model.ckpt-1000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: /home/jupyter/training-data-a
nalyst/courses/machine_learning/deepdive/05_artandscience/outputdir/
export/exporter/temp-b'1586902396'/saved_model.pb
INFO:tensorflow:Loss for final step: 0.00023530833.
```