2d. Distributed training and monitoring

In this notebook, we refactor to call `train_and_evaluate` instead of hand-coding our ML pipeline. This allows us to carry out evaluation as part of our training loop instead of as a separate step. It also adds in failure-handling that is necessary for distributed training capabilities.

We also use TensorBoard to monitor the training.

In [1]:

```
import tensorflow as tf
import numpy as np
import shutil
print(tf.__version__)
```

```
1.15.0
```

Input

Read data created in Lab1a, but this time make it more general, so that we are reading in batches. Instead of using Pandas, we will use Datasets.

In [2]:

```
CSV_COLUMNS = ['fare_amount', 'pickuplon','pickuplat','dropofflon','dropofflat','passengers', 'key']
LABEL_COLUMN = 'fare_amount'
DEFAULTS = [[0.0], [-74.0], [40.0], [-74.0], [40.7], [1.0], ['nokey']]

def read_dataset(filename, mode, batch_size = 512):
  def _input_fn():
    def decode_csv(value_column):
      columns = tf.decode_csv(value_column, record_defaults = DEFAULTS)
      features = dict(zip(CSV_COLUMNS, columns))
      label = features.pop(LABEL_COLUMN)
      return features, label

    # 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)
    return dataset.make_one_shot_iterator().get_next()
  return _input_fn
```

Create features out of input data

For now, pass these through. (same as previous lab)

In [3]:

```
INPUT_COLUMNS = [
    tf.feature_column.numeric_column('pickuplon'),
    tf.feature_column.numeric_column('pickuplat'),
    tf.feature_column.numeric_column('dropofflat'),
    tf.feature_column.numeric_column('dropofflon'),
    tf.feature_column.numeric_column('passengers'),
]

def add_more_features(feats):
  # Nothing to add (yet!)
  return feats

feature_cols = add_more_features(INPUT_COLUMNS)
```

train_and_evaluate

In [4]:

```
def serving_input_fn():
  feature_placeholders = {
    'pickuplon' : tf.placeholder(tf.float32, [None]),
    'pickuplat' : tf.placeholder(tf.float32, [None]),
    'dropofflat' : tf.placeholder(tf.float32, [None]),
    'dropofflon' : tf.placeholder(tf.float32, [None]),
    'passengers' : tf.placeholder(tf.float32, [None]),
  }
  features = {
      key: tf.expand_dims(tensor, -1)
      for key, tensor in feature_placeholders.items()
  }
  return tf.estimator.export.ServingInputReceiver(features, feature_placeholders)
```

In [5]:

```
def train_and_evaluate(output_dir, num_train_steps):
  estimator = tf.estimator.LinearRegressor(
                       model_dir = output_dir,
                       feature_columns = feature_cols)
  train_spec=tf.estimator.TrainSpec(
                       input_fn = read_dataset('./taxi-train.csv', mode = tf.estimator.ModeKeys.TRAIN),
                       max_steps = num_train_steps)
  exporter = tf.estimator.LatestExporter('exporter', serving_input_fn)
  eval_spec=tf.estimator.EvalSpec(
                       input_fn = read_dataset('./taxi-valid.csv', mode = tf.estimator.ModeKeys.EVAL),
                       steps = None,
                       start_delay_secs = 1, # start evaluating after N seconds
                       throttle_secs = 10,  # evaluate every N seconds
                       exporters = exporter)
  tf.estimator.train_and_evaluate(estimator, train_spec, eval_spec)
```

In [6]:

```
# Run training
OUTDIR = 'taxi_trained'
shutil.rmtree(OUTDIR, ignore_errors = True) # start fresh each time
train_and_evaluate(OUTDIR, num_train_steps = 5000)
```

```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_tf_random_seed': None, '_train_distribute': None, '_session_creatio
n_timeout_secs': 7200, '_cluster_spec': <tensorflow.python.training.server_lib.ClusterSpec object at
0x7f6ddb1f5160>, '_task_type': 'worker', '_experimental_distribute': None, '_master': '', '_task_i
d': 0, '_protocol': None, '_save_checkpoints_steps': None, '_log_step_count_steps': 100, '_device_f
n': None, '_eval_distribute': None, '_num_worker_replicas': 1, '_evaluation_master': '', '_save_summ
ary_steps': 100, '_experimental_max_worker_delay_secs': None, '_model_dir': 'taxi_trained', '_keep_c
heckpoint_every_n_hours': 10000, '_keep_checkpoint_max': 5, '_save_checkpoints_secs': 600, '_session
_config': allow_soft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_num_ps_replicas': 0, '_service': None, '_global_id_in_cluster': 0, '_is_chief': True}
INFO:tensorflow:Not using Distribute Coordinator.
INFO:tensorflow:Running training and evaluation locally (non-distributed).
INFO:tensorflow:Start train and evaluate loop. The evaluate will happen after every checkpoint. Chec
kpoint frequency is determined based on RunConfig arguments: save_checkpoints_steps None or save_che
ckpoints_secs 600.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/training/train
ing_util.py:236: Variable.initialized_value (from tensorflow.python.ops.variables) is deprecated and
will be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatically both in eager and graph (ins
ide tf.defun) contexts.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/autograph/conv
erters/directives.py:119: The name tf.decode_csv is deprecated. Please use tf.io.decode_csv instead.

WARNING:tensorflow:From <ipython-input-2-9db945032f46>:26: DatasetV1.make_one_shot_iterator (from te
nsorflow.python.data.ops.dataset_ops) 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.estimator`, return the `Dataset` o
bject directly from your input function. As a last resort, you can use `tf.compat.v1.data.make_one_s
hot_iterator(dataset)`.
INFO:tensorflow:Calling model_fn.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/feature_colum
n/feature_column_v2.py:305: Layer.add_variable (from tensorflow.python.keras.engine.base_layer) is d
eprecated and will be removed in a future version.
Instructions for updating:
Please use `layer.add_weight` method instead.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/ops/resource_v
ariable_ops.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_vari
```

```
able_ops) with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_estimator/python/estimato
r/canned/linear.py:308: to_float (from tensorflow.python.ops.math_ops) is deprecated and will be rem
oved in a future version.
Instructions for updating:
Use `tf.cast` instead.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/ops/array_ops.
py:1475: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future
version.
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 taxi_trained/model.ckpt.
INFO:tensorflow:loss = 95557.62, step = 1
INFO:tensorflow:global_step/sec: 28.5132
INFO:tensorflow:loss = 46868.44, step = 101 (3.509 sec)
INFO:tensorflow:global_step/sec: 27.2148
INFO:tensorflow:loss = 71028.28, step = 201 (3.685 sec)
INFO:tensorflow:global_step/sec: 26.1641
INFO:tensorflow:loss = 63297.094, step = 301 (3.811 sec)
INFO:tensorflow:global_step/sec: 27.1608
INFO:tensorflow:loss = 42960.21, step = 401 (3.687 sec)
INFO:tensorflow:global_step/sec: 28.8093
INFO:tensorflow:loss = 39423.203, step = 501 (3.467 sec)
INFO:tensorflow:global_step/sec: 29.1613
INFO:tensorflow:loss = 66785.97, step = 601 (3.428 sec)
INFO:tensorflow:global_step/sec: 27.5579
INFO:tensorflow:loss = 64827.508, step = 701 (3.629 sec)
INFO:tensorflow:global_step/sec: 27.0409
INFO:tensorflow:loss = 35764.305, step = 801 (3.698 sec)
INFO:tensorflow:global_step/sec: 27.6022
INFO:tensorflow:loss = 62911.875, step = 901 (3.624 sec)
INFO:tensorflow:global_step/sec: 26.9141
INFO:tensorflow:loss = 40300.5, step = 1001 (3.715 sec)
INFO:tensorflow:global_step/sec: 28.5412
INFO:tensorflow:loss = 67090.34, step = 1101 (3.507 sec)
INFO:tensorflow:global_step/sec: 28.7928
```

```
INFO:tensorflow:loss = 84474.734, step = 1201 (3.469 sec)
INFO:tensorflow:global_step/sec: 32.0071
INFO:tensorflow:loss = 38200.45, step = 1301 (3.124 sec)
INFO:tensorflow:global_step/sec: 28.0899
INFO:tensorflow:loss = 35490.637, step = 1401 (3.563 sec)
INFO:tensorflow:global_step/sec: 25.4654
INFO:tensorflow:loss = 61773.055, step = 1501 (3.928 sec)
INFO:tensorflow:global_step/sec: 29.6727
INFO:tensorflow:loss = 68101.27, step = 1601 (3.367 sec)
INFO:tensorflow:global_step/sec: 29.8084
INFO:tensorflow:loss = 45069.07, step = 1701 (3.354 sec)
INFO:tensorflow:global_step/sec: 26.2312
INFO:tensorflow:loss = 46238.03, step = 1801 (3.816 sec)
INFO:tensorflow:global_step/sec: 27.1187
INFO:tensorflow:loss = 47554.867, step = 1901 (3.684 sec)
INFO:tensorflow:global_step/sec: 25.7383
INFO:tensorflow:loss = 54820.07, step = 2001 (3.885 sec)
INFO:tensorflow:global_step/sec: 30.0384
INFO:tensorflow:loss = 55158.492, step = 2101 (3.333 sec)
INFO:tensorflow:global_step/sec: 27.4852
INFO:tensorflow:loss = 43146.85, step = 2201 (3.638 sec)
INFO:tensorflow:global_step/sec: 26.3102
INFO:tensorflow:loss = 55931.977, step = 2301 (3.802 sec)
INFO:tensorflow:global_step/sec: 29.3007
INFO:tensorflow:loss = 56526.61, step = 2401 (3.412 sec)
INFO:tensorflow:global_step/sec: 25.7723
INFO:tensorflow:loss = 33954.336, step = 2501 (3.880 sec)
INFO:tensorflow:global_step/sec: 27.7556
INFO:tensorflow:loss = 44739.938, step = 2601 (3.599 sec)
INFO:tensorflow:global_step/sec: 27.1957
INFO:tensorflow:loss = 51653.47, step = 2701 (3.677 sec)
INFO:tensorflow:global_step/sec: 27.79
INFO:tensorflow:loss = 77831.555, step = 2801 (3.599 sec)
INFO:tensorflow:global_step/sec: 25.7523
INFO:tensorflow:loss = 43329.49, step = 2901 (3.883 sec)
INFO:tensorflow:global_step/sec: 25.6264
INFO:tensorflow:loss = 39390.812, step = 3001 (3.902 sec)
INFO:tensorflow:global_step/sec: 26.1875
INFO:tensorflow:loss = 47748.594, step = 3101 (3.819 sec)
INFO:tensorflow:global_step/sec: 26.8898
INFO:tensorflow:loss = 44505.227, step = 3201 (3.723 sec)
INFO:tensorflow:global_step/sec: 25.8756
INFO:tensorflow:loss = 41444.63, step = 3301 (3.865 sec)
```

```
INFO:tensorflow:global_step/sec: 25.5495
INFO:tensorflow:loss = 52066.375, step = 3401 (3.914 sec)
INFO:tensorflow:global_step/sec: 27.3561
INFO:tensorflow:loss = 30863.137, step = 3501 (3.655 sec)
INFO:tensorflow:global_step/sec: 27.3358
INFO:tensorflow:loss = 42025.797, step = 3601 (3.658 sec)
INFO:tensorflow:global_step/sec: 28.6355
INFO:tensorflow:loss = 54339.09, step = 3701 (3.490 sec)
INFO:tensorflow:global_step/sec: 26.7369
INFO:tensorflow:loss = 44323.74, step = 3801 (3.739 sec)
INFO:tensorflow:global_step/sec: 27.7
INFO:tensorflow:loss = 53061.47, step = 3901 (3.614 sec)
INFO:tensorflow:global_step/sec: 27.4454
INFO:tensorflow:loss = 34842.61, step = 4001 (3.640 sec)
INFO:tensorflow:global_step/sec: 27.4899
INFO:tensorflow:loss = 46118.594, step = 4101 (3.638 sec)
INFO:tensorflow:global_step/sec: 28.5328
INFO:tensorflow:loss = 37321.9, step = 4201 (3.506 sec)
INFO:tensorflow:global_step/sec: 27.3473
INFO:tensorflow:loss = 51358.574, step = 4301 (3.655 sec)
INFO:tensorflow:global_step/sec: 26.5859
INFO:tensorflow:loss = 89928.36, step = 4401 (3.761 sec)
INFO:tensorflow:global_step/sec: 29.0149
INFO:tensorflow:loss = 42115.945, step = 4501 (3.451 sec)
INFO:tensorflow:global_step/sec: 27.1411
INFO:tensorflow:loss = 44038.35, step = 4601 (3.680 sec)
INFO:tensorflow:global_step/sec: 28.0503
INFO:tensorflow:loss = 33126.336, step = 4701 (3.565 sec)
INFO:tensorflow:global_step/sec: 26.5597
INFO:tensorflow:loss = 30635.715, step = 4801 (3.765 sec)
INFO:tensorflow:global_step/sec: 27.7572
INFO:tensorflow:loss = 50354.914, step = 4901 (3.606 sec)
INFO:tensorflow:Saving checkpoints for 5000 into taxi_trained/model.ckpt.
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2020-01-17T20:24:03Z
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from taxi_trained/model.ckpt-5000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2020-01-17-20:24:04
INFO:tensorflow:Saving dict for global step 5000: average_loss = 86.9354, global_step = 5000, label/
mean = 11.419548, loss = 34143.88, prediction/mean = 11.086628
```

```
INFO:tensorflow:Saving 'checkpoint_path' summary for global step 5000: taxi_trained/model.ckpt-5000
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow_core/python/saved_model/si
gnature_def_utils_impl.py:201: build_tensor_info (from tensorflow.python.saved_model.utils_impl) is
deprecated and will be removed in a future version.
Instructions for updating:
This function will only be available through the v1 compatibility library as tf.compat.v1.saved_mode
l.utils.build_tensor_info or tf.compat.v1.saved_model.build_tensor_info.
INFO:tensorflow:Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['predict']
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures EXCLUDED from export because they cannot be be served via TensorFlow Serv
ing APIs:
INFO:tensorflow:'serving_default' : Regression input must be a single string Tensor; got {'pickupla
t': <tf.Tensor 'Placeholder_1:0' shape=(?,) dtype=float32>, 'dropofflon': <tf.Tensor 'Placeholder_3:
0' shape=(?,) dtype=float32>, 'passengers': <tf.Tensor 'Placeholder_4:0' shape=(?,) dtype=float32>,
'dropofflat': <tf.Tensor 'Placeholder_2:0' shape=(?,) dtype=float32>, 'pickuplon': <tf.Tensor 'Place
holder:0' shape=(?,) dtype=float32>}
INFO:tensorflow:'regression' : Regression input must be a single string Tensor; got {'pickuplat': <t
f.Tensor 'Placeholder_1:0' shape=(?,) dtype=float32>, 'dropofflon': <tf.Tensor 'Placeholder_3:0' sha
pe=(?,) dtype=float32>, 'passengers': <tf.Tensor 'Placeholder_4:0' shape=(?,) dtype=float32>, 'dropo
fflat': <tf.Tensor 'Placeholder_2:0' shape=(?,) dtype=float32>, 'pickuplon': <tf.Tensor 'Placeholde
r:0' shape=(?,) dtype=float32>}
WARNING:tensorflow:Export includes no default signature!
INFO:tensorflow:Restoring parameters from taxi_trained/model.ckpt-5000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: taxi_trained/export/exporter/temp-b'1579292644'/saved_model.p
b
INFO:tensorflow:Loss for final step: 55487.695.
```

Monitor training with TensorBoard

To activate TensorBoard within the JupyterLab UI navigate to "**File**" - "**New Launcher**". Then double-click the 'Tensorboard' icon on the bottom row.

TensorBoard 1 will appear in the new tab. Navigate through the three tabs to see the active TensorBoard. The 'Graphs' and 'Projector' tabs offer very interesting information including the ability to replay the tests.

You may close the TensorBoard tab when you are finished exploring.