Tensorboard Mnist - 1

TensorFlow + Keras 2.0 Intro
2017-4-1

Like last time, but

- Add better logging
- Fancy graphs
- Epochs clearly delineated in training

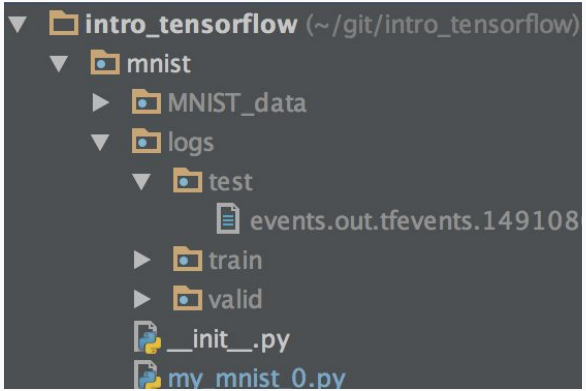
Tensorflow Logger

`tf.summary.Filewriter`

- One type of data stream → one log directory
  - Since we can't name the log files themselves

Example: Finding loss and accuracy on train, validation, and test data:

→ Write to a different directory for each of train, validation, and test.



Make the Filewriters

```
log_path = os.path.join(PATH, 'logs')
```

- set log path

```
train_writer = tf.summary.FileWriter(log_path + '/train', graph=sess.graph)
valid_writer = tf.summary.FileWriter(log_path + '/valid', graph=sess.graph)
test_writer = tf.summary.FileWriter(log_path + '/test', graph=sess.graph)
```

- one `Filewriter` for each directory

Set epoch size and number

```
samples_per_epoch = mnist_data.train.labels.shape[0]  # 1 pass over dataset
```

- `labels.shape` = (total training samples, 10)

```
epochs = 10
assert samples_per_epoch % batch_size == 0, \
  'batch size {} does not divide epoch size {}'.format(batch_size, samples_per_epoch)
```

- Ensure that each epoch is an integer number of batches

The usual epoch format

```
epoch = 0
while epoch < epochs:
  print('Starting epoch {}'.format(epoch))
  sampled = 0
  while sampled < samples_per_epoch:
          . . .
    <training magic>
          . . .
    sampled += batch_size
  epoch += 1
```

Training magic + output

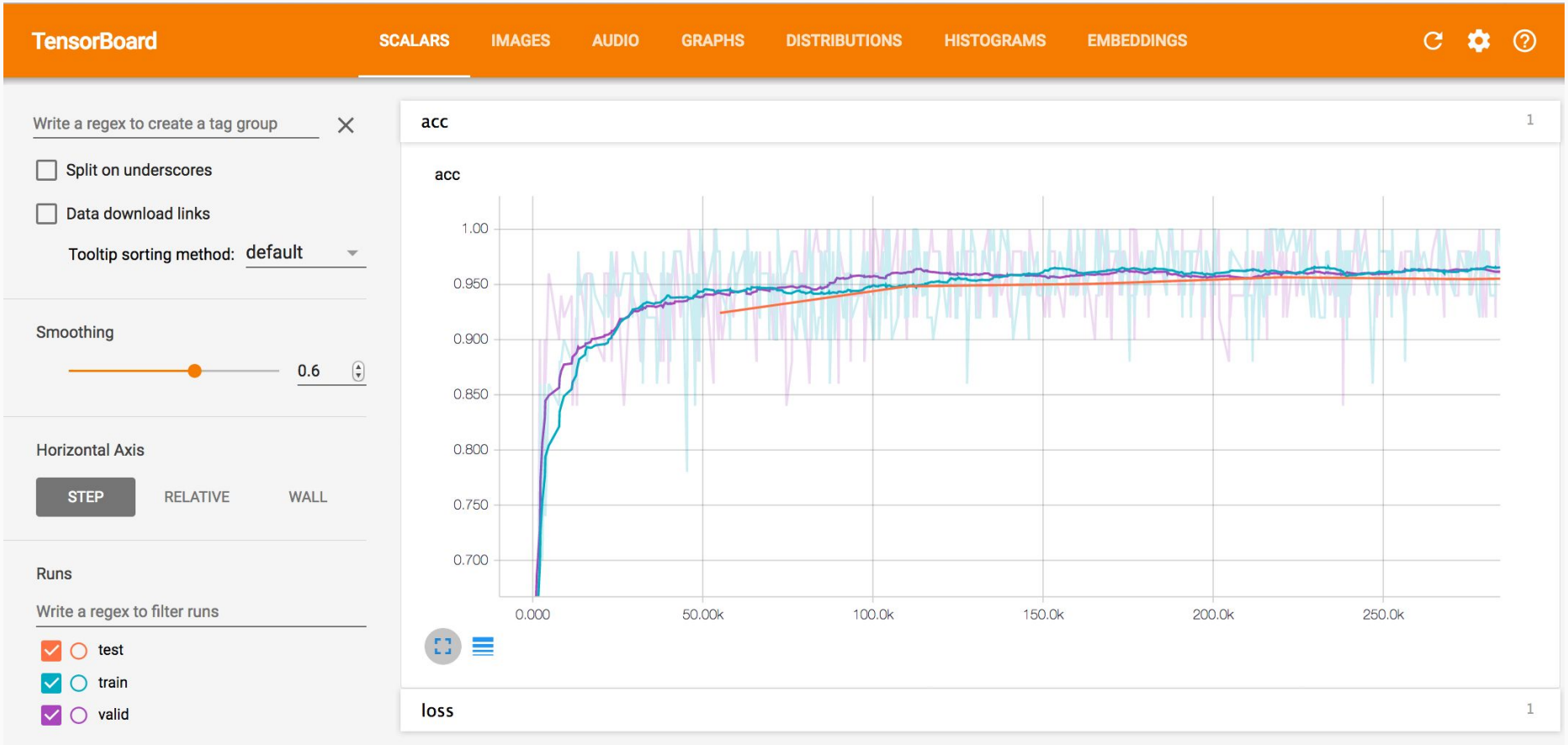
```
# train
batch = mnist_data.train.next_batch(batch_size)
summary, _ = sess.run([summary_op, train_step], feed_dict={
  img: batch[0],
  labels: batch[1]
})
train_writer.add_summary(summary, epoch*samples_per_epoch + sampled)

# validation
valid_batch = mnist_data.validation.next_batch(batch_size)
summary, _, _ = sess.run([summary_op, acc_value, loss], feed_dict={
  img: valid_batch[0],
  labels: valid_batch[1]
})
valid_writer.add_summary(summary, epoch*samples_per_epoch + sampled)
```

Tensorboard

- Run the new mnist program
- In top directory of the git repository, in terminal:
- `tensorboard --logdir=`pwd`/mnist/logs`
- In browser, go to `localhost:6006`

Tensorboard Interface

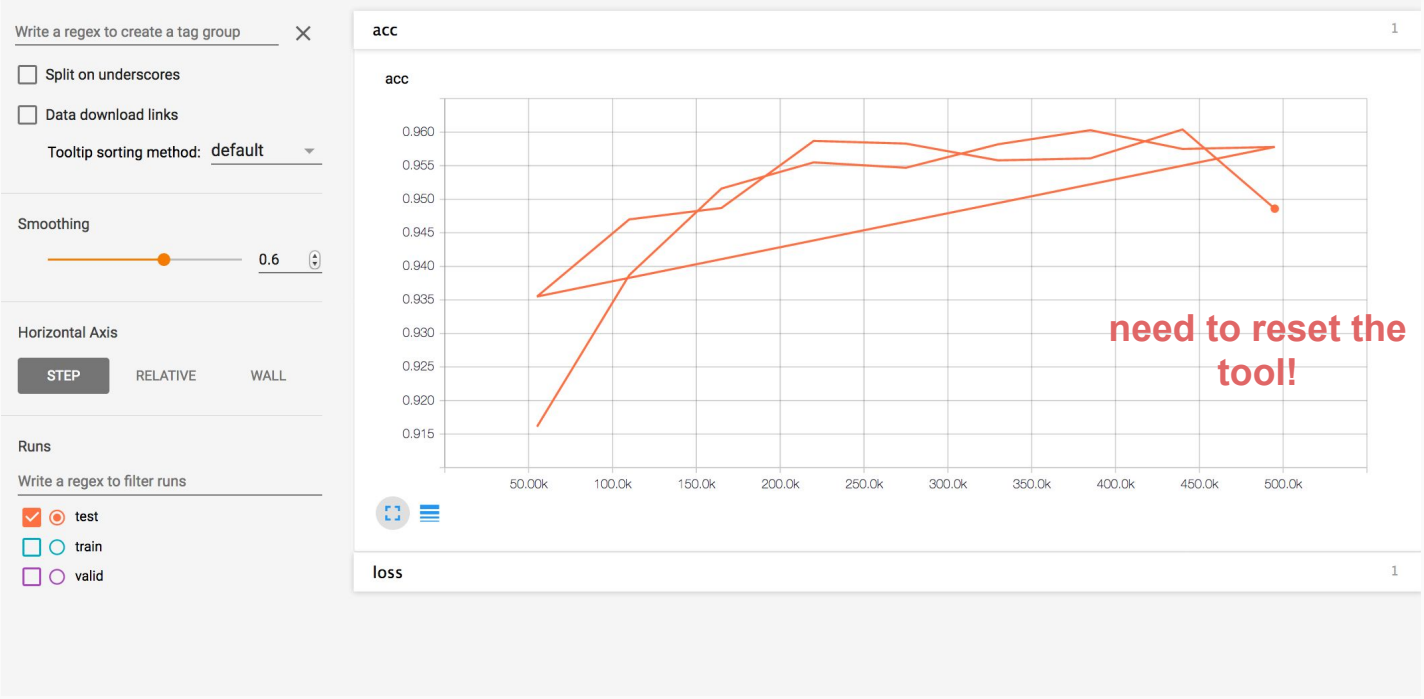


Tensorboard stuff

Using the graph:

- draw rectangle on the graph to zoom to that selection
- double click to zoom out

If multiple logs are written to same folder



Resetting a Tensorboard log

- Delete the log folder
  - Destroys all history, including the appended new run
- Rerun the program that makes the log
- Restart the tensorboard process running in terminal

- Now we have the basic tools need to use Tensorboard!
  - It can also be used to record non-Tensorflow info, by converting the values to Tensorflow variables/operations and printing those with the `Filewriter`