In [ ]:
```
# set tf 1.x for colab
%tensorflow_version 1.x
```

Video "What is TensorFlow"

In [1]:
```
%config IPCompleter.greedy=True
```

In [2]:
```
import tensorflow as tf
import numpy as np
print(tf.__version__)
```

```
1.3.0
```

In [3]:
```
tf.reset_default_graph()
a = tf.placeholder(np.float32, (2, 2))
b = tf.Variable(tf.ones((2, 2)))
c = a @ b
```

In [4]:
```
print(c)
```

```
Tensor("matmul:0", shape=(2, 2), dtype=float32)
```

In [5]:
```
s = tf.InteractiveSession()
```

In [6]:
```
s.run(tf.global_variables_initializer())
s.run(c, feed_dict={a: np.ones((2, 2))})
```

Out[6]:
```
array([[ 2.,  2.],
       [ 2.,  2.]], dtype=float32)
```

In [7]:
```
s.close()
```

Video "Our first model in TensorFlow"

Simple optimization (with simple prints)

In [8]:
```
tf.reset_default_graph()
x = tf.get_variable("x", shape=(), dtype=tf.float32, trainable=True)
f = x ** 2
```

In [9]:
```
optimizer = tf.train.GradientDescentOptimizer(0.1)
step = optimizer.minimize(f, var_list=[x])
```

In [10]:
```
tf.trainable_variables()
```

Out[10]:
```
[<tf.Variable 'x:0' shape=() dtype=float32_ref>]
```

In [11]:
```
with tf.Session() as s:  # in this way session will be closed automatically
    s.run(tf.global_variables_initializer())
    for i in range(10):
        _, curr_x, curr_f = s.run([step, x, f])
        print(curr_x, curr_f)
```

```
-0.865988 1.17177
-0.69279 0.749935
-0.554232 0.479959
-0.443386 0.307174
-0.354709 0.196591
-0.283767 0.125818
-0.227014 0.0805237
-0.181611 0.0515352
-0.145289 0.0329825
-0.116231 0.0211088
```

Simple optimization (with tf.Print)

In [12]:
```
tf.reset_default_graph()
x = tf.get_variable("x", shape=(), dtype=tf.float32)
f = x ** 2
f = tf.Print(f, [x, f], "x, f:")
```

In [13]:
```
optimizer = tf.train.GradientDescentOptimizer(0.1)
step = optimizer.minimize(f)
```

In [14]:
```
with tf.Session() as s:
    s.run(tf.global_variables_initializer())
    for i in range(10):
        s.run([step, f])
```

In [15]:
```
# Prints to jupyter server stdout (not available in Coursera Notebooks):
# 2018-07-21 18:01:27.308270: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-1.0670249][1.1385423]
# 2018-07-21 18:01:27.308809: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.85361993][0.72866696]
# 2018-07-21 18:01:27.309116: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.68289596][0.46634689]
# 2018-07-21 18:01:27.309388: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.54631674][0.29846197]
# 2018-07-21 18:01:27.309678: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.43705338][0.19101566]
# 2018-07-21 18:01:27.309889: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.34964269][0.12225001]
# 2018-07-21 18:01:27.310213: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.27971417][0.078240015]
# 2018-07-21 18:01:27.310475: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.22377133][0.050073609]
# 2018-07-21 18:01:27.310751: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.17901707][0.032047112]
# 2018-07-21 18:01:27.310963: I tensorflow/core/kernels/logging_ops.cc:79] x, f:[-0.14321366][0.020510152]
```

Simple optimization (with TensorBoard logging)

In [16]:
```
tf.reset_default_graph()
x = tf.get_variable("x", shape=(), dtype=tf.float32)
f = x ** 2
```

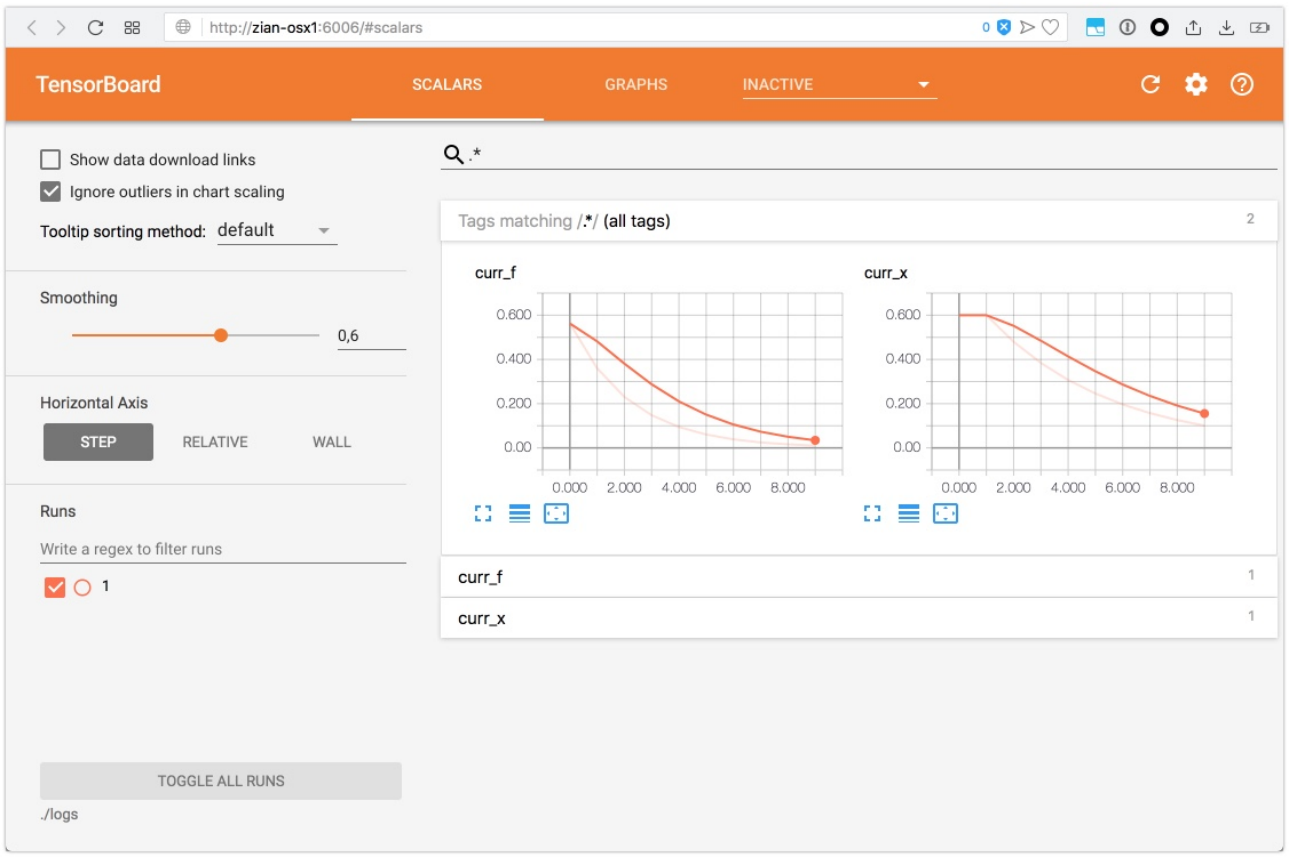
In [17]:
```
optimizer = tf.train.GradientDescentOptimizer(0.1)
step = optimizer.minimize(f)
```

In [18]:
```
tf.summary.scalar('curr_x', x)
tf.summary.scalar('curr_f', f)
summaries = tf.summary.merge_all()
```

In [19]:
```
s = tf.InteractiveSession()
summary_writer = tf.summary.FileWriter("logs/1", s.graph)
s.run(tf.global_variables_initializer())
for i in range(10):
    _, curr_summaries = s.run([step, summaries])
    summary_writer.add_summary(curr_summaries, i)
    summary_writer.flush()
```

Run `tensorboard --logdir=./logs` in bash

This is what you can see in your browser **(not available in Coursera Notebooks)**



If you're running on Google Colab you can still run TensorBoard!

In [ ]:
```
# !!! RUN THIS CELL ONLY ON GOOGLE COLAB !!!
! wget https://raw.githubusercontent.com/hse-aml/intro-to-dl/master/setup_google_colab.py -O setup_google_colab.py
import setup_google_colab

# run tensorboard in background
import os
os.system("tensorboard --logdir=./logs --host 0.0.0.0 --port 6006 &")

# expose port and show the link
setup_google_colab.expose_port_on_colab(6006)
```

In [20]:
```
s.close()
```

Training a linear model

In [21]:
```
# generate model data
N = 1000
D = 3
x = np.random.random((N, D))
w = np.random.random((D, 1))
y = x @ w + np.random.randn(N, 1) * 0.20

print(x.shape, y.shape)
print(w.T)
```

```
(1000, 3) (1000, 1)
[[ 0.09498027  0.48793618  0.39011257]]
```

In [22]:
```
tf.reset_default_graph()

features = tf.placeholder(tf.float32, shape=(None, D))
target = tf.placeholder(tf.float32, shape=(None, 1))

weights = tf.get_variable("weights", shape=(D, 1), dtype=tf.float32)
predictions = features @ weights

loss = tf.reduce_mean((target - predictions) ** 2)

print(target.shape, predictions.shape, loss.shape)
```

```
(?, 1) (?, 1) ()
```

In [23]:
```
optimizer = tf.train.GradientDescentOptimizer(0.1)
step = optimizer.minimize(loss)
```

In [24]:
```
with tf.Session() as s:
    s.run(tf.global_variables_initializer())
    for i in range(300):
        _, curr_loss, curr_weights = s.run([step, loss, weights],
                                           feed_dict={features: x, target: y})
        if i % 50 == 0:
            print(curr_loss)
```

```
0.532865
0.0458802
0.0410158
0.040087
0.0399092
0.0398751
```

In [25]:
```
# found weights
curr_weights.T
```

Out[25]:
```
array([[ 0.11388827,  0.4882018 ,  0.36716884]], dtype=float32)
```

In [26]:
```
# true weights
w.T
```

Out[26]:
```
array([[ 0.09498027,  0.48793618,  0.39011257]])
```