Tensorboard工具与模型优化

七月在线 张雨石

2017年8月13日

http://blog.csdn.net/stdcoutzyx

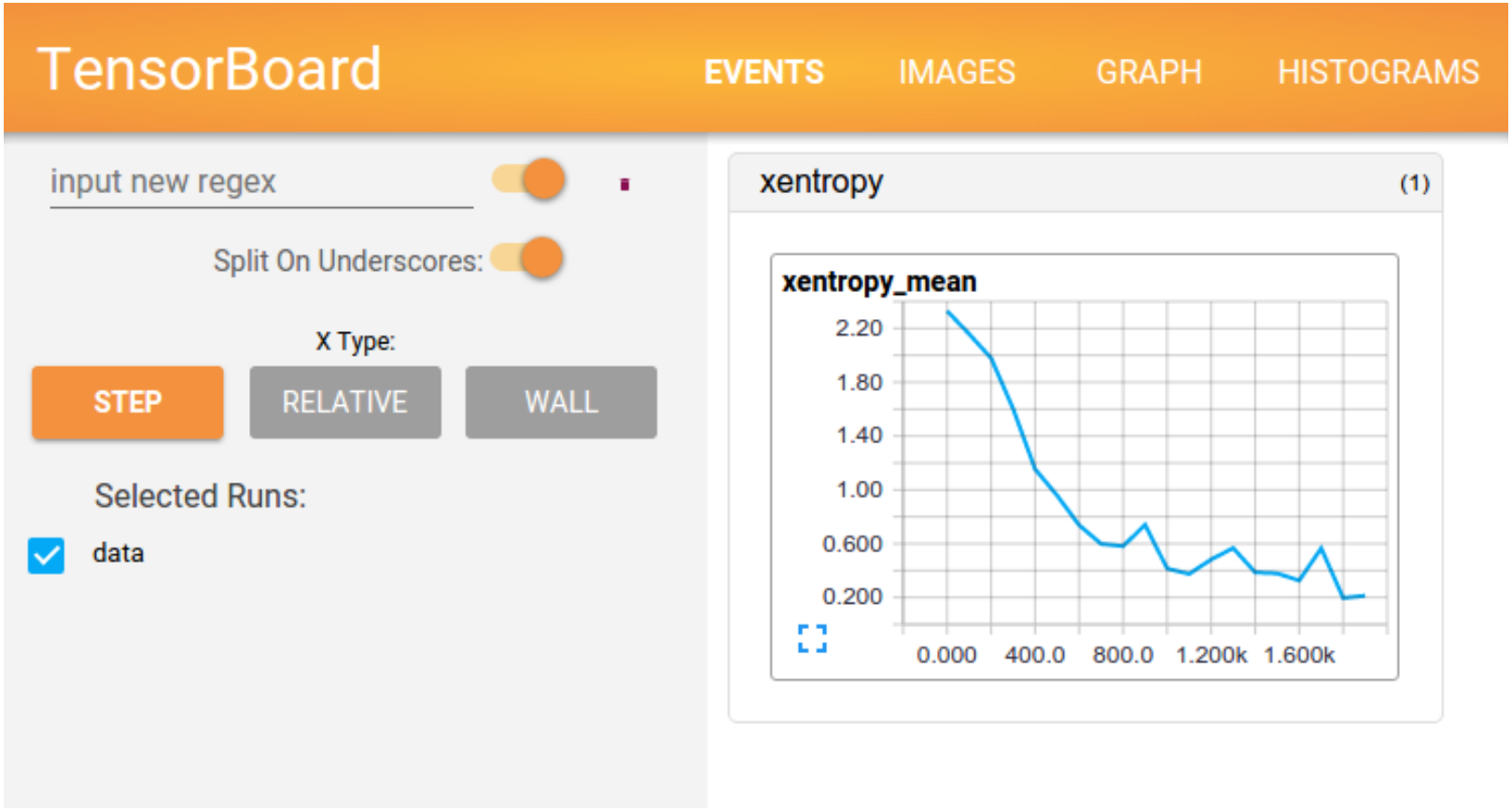
目录

- Tensorboard简介
- Tensorboard使用
- 调参手段
- 调优案例分析

Tensorboard简介

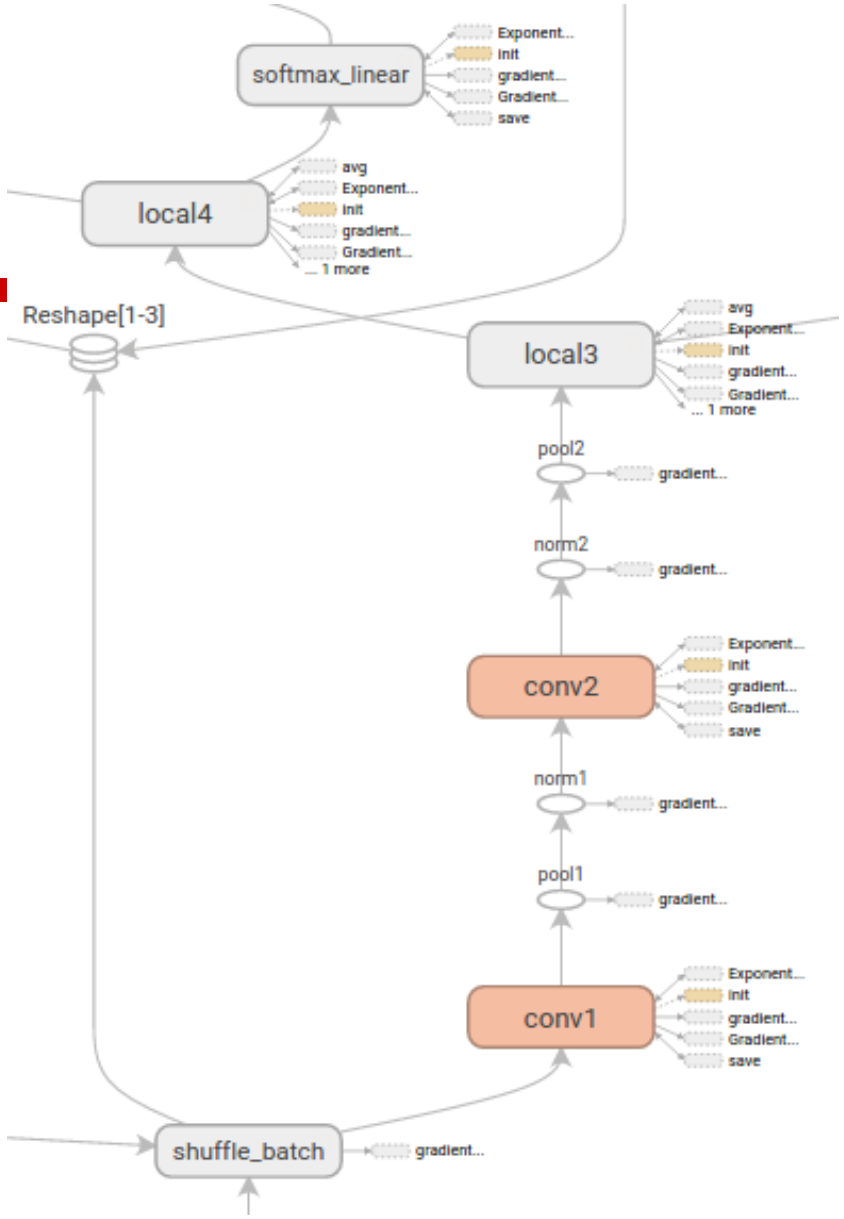
Deep Learning model == Black box

Tensorboard简介



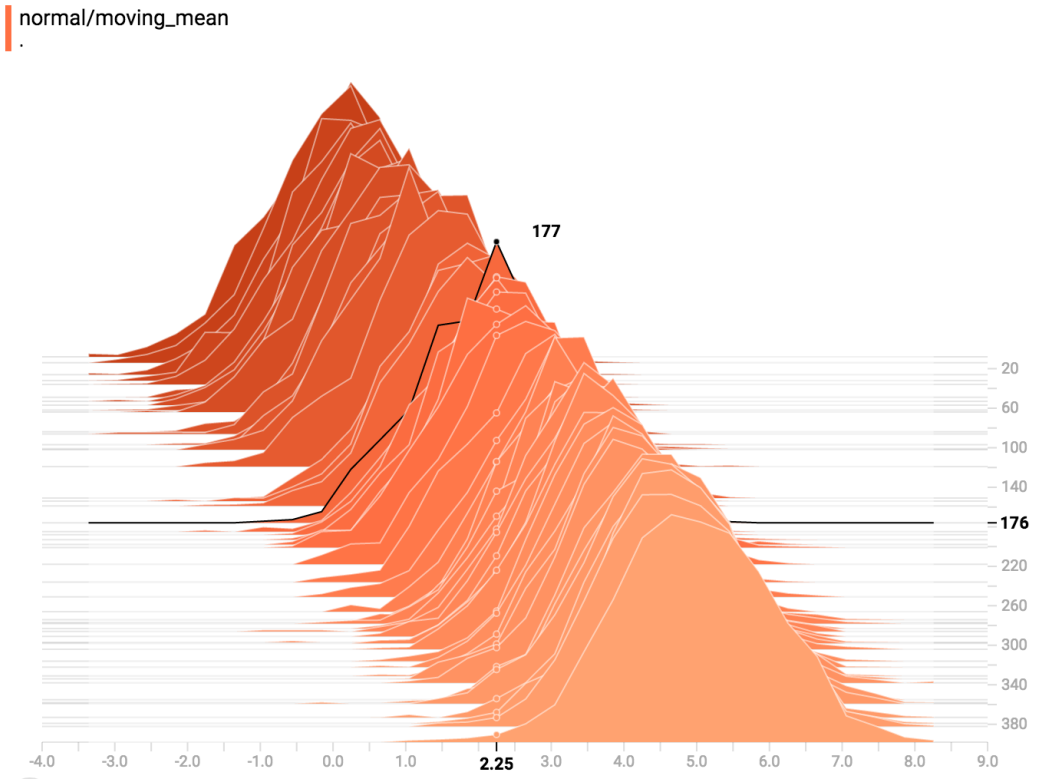
Tensorboard简介

- 可视化
  - 模型结构



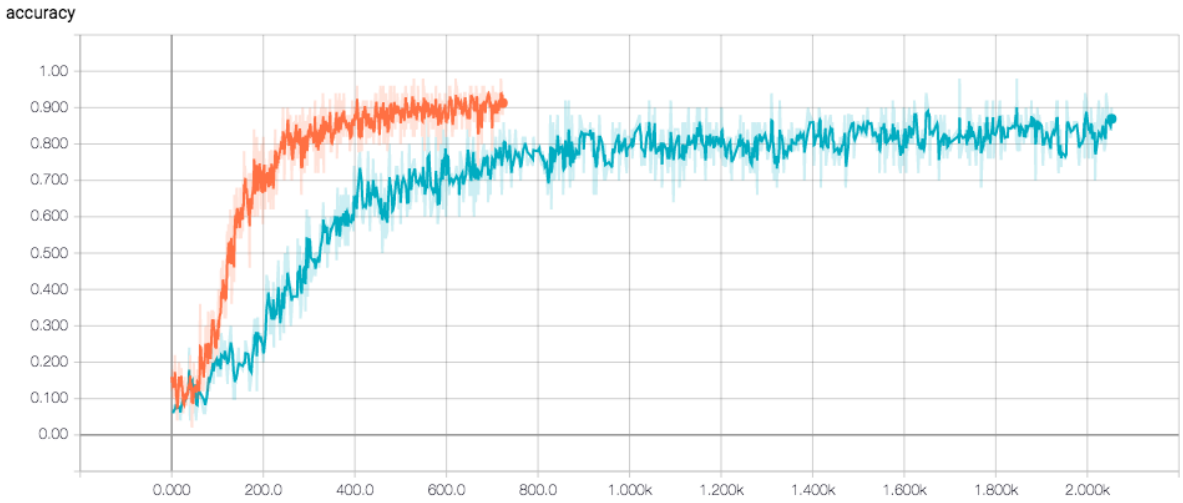
Tensorboard简介

- 可视化
  - 模型结构
  - 参数分布



Tensorboard简介

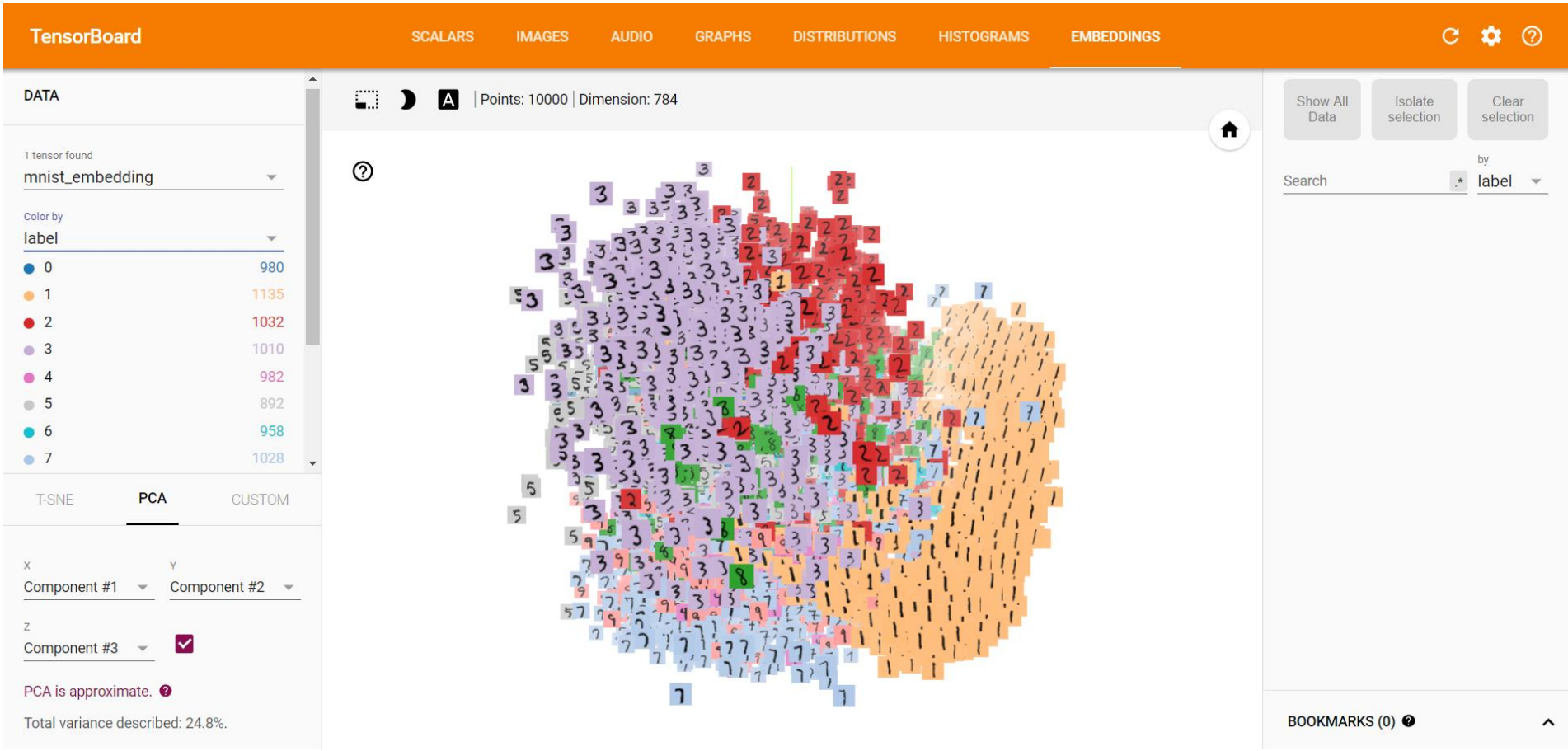
- 可视化
  - 模型结构
  - 参数分布
  - 训练过程



Tensorboard简介

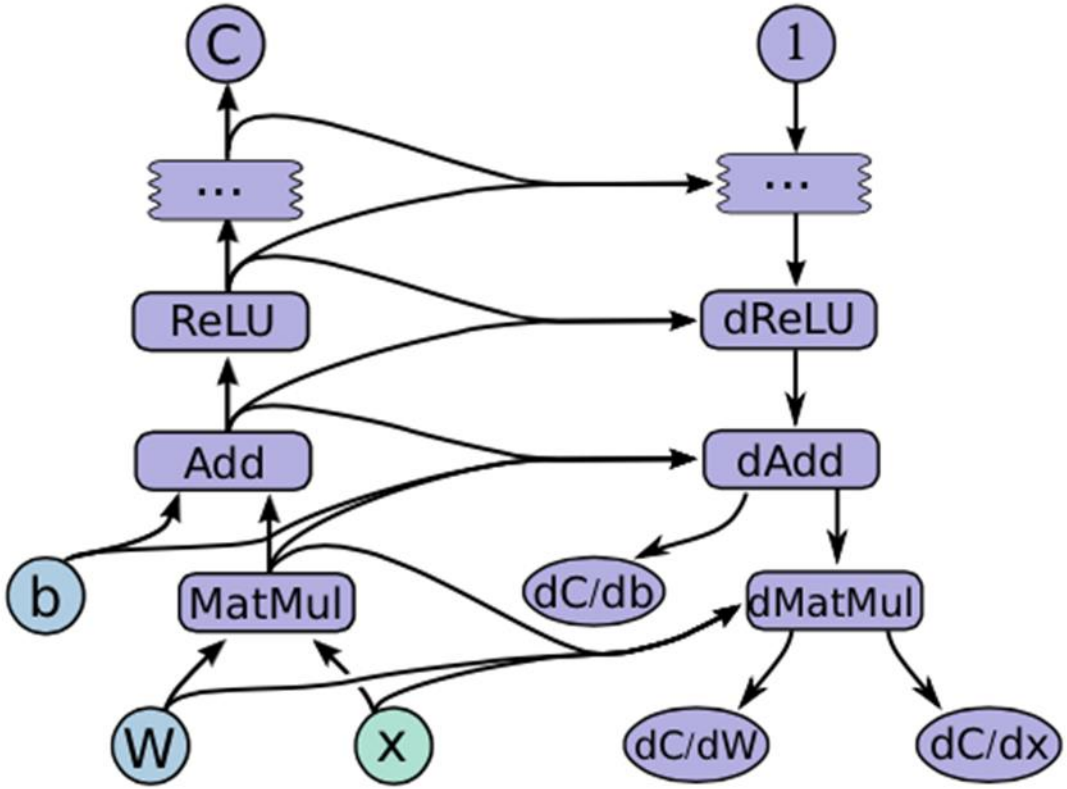
- 可视化
  - 模型结构
  - 参数分布
  - 训练过程
  - Embedding数据分析

Tensorboard简介



Tensorboard使用-Big Picture

- Tensorflow 概念回顾



Tensorboard使用-Big Picture

- Summary_ops
  - tf.summary.tensor_summary
  - tf.summary.scalar
  - tf.summary.histogram
  - tf.summary.audio
  - tf.summary.image
  - tf.summary.merge
  - tf.summary.merge_all

Tensorboard使用-Big Picture

- Summary_ops
  - 数据序列化
  - 写到硬盘/发给tensorboard
- 其他API
  - tf.summary.FileWriter
  - tf.summary.FileWriterCache

Tensorboard使用-Big Picture

- 回顾：可以做什么
  - 输出模型结构
  - 跟踪训练状态
  - 分析参数
  - 分析数据

Tensorboard使用-base code

```
11 def conv2d(x, output_dim, k_h=5, k_w=5, s_h=1, s_w=1, stddev=0.02, name="conv2d"):
12     with tf.variable_scope(name):
13         w = tf.get_variable('w', [k_h, k_w, x.get_shape()[-1], output_dim],
14                 initializer = tf.truncated_normal_initializer(stddev=stddev))
15         conv = tf.nn.conv2d(x, w, strides=[1, s_h, s_w, 1], padding="SAME")
16         biases = tf.get_variable('biases', [output_dim],
17                 initializer = tf.constant_initializer(0.0))
18         conv = conv + biases
19         return conv
20
21 def max_pool_2d(x, k_h=2, k_w=2, s_h=2, s_w=2, name="max_pool"):
22     return tf.nn.max_pool(
23             x,
24             ksize=[1, k_h, k_w, 1],
25             strides=[1, s_h, s_w, 1],
26             padding="SAME",
27             name=name)
```

Tensorboard使用-base code

```
29 def dense(x, output_dim, stddev=0.02, name="dense"):
30     with tf.variable_scope(name):
31         w = tf.get_variable('w', [x.get_shape()[-1], output_dim],
32                 initializer = tf.truncated_normal_initializer(stddev=stddev))
33         biases = tf.get_variable('biases', [output_dim],
34                 initializer = tf.constant_initializer(0.0))
35         return tf.matmul(x, w) + biases
36
37 def lrelu(x, name="lrelu"):
38     return tf.maximum(x, 0.2 * x)
```

Tensorboard使用-base code

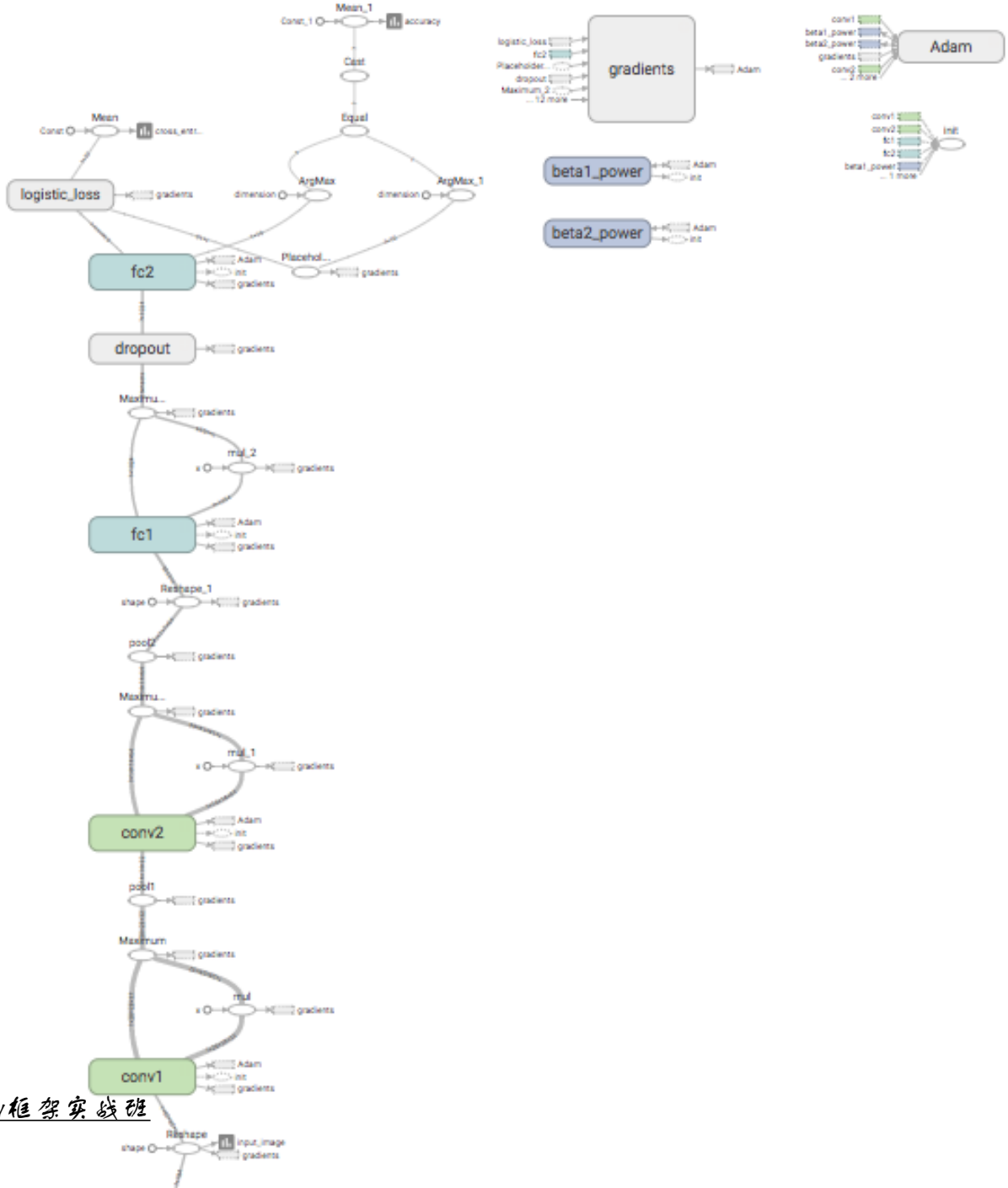
```
40 x = tf.placeholder(tf.float32, [None, 784])
41 x_image = tf.reshape(x, [-1, 28, 28, 1])
42 conv1 = lrelu(conv2d(x_image, 32, name="conv1"))
43 pool1 = max_pool_2d(conv1, name="pool1")
44 conv2 = lrelu(conv2d(pool1, 64, name="conv2"))
45 pool2 = max_pool_2d(conv2, name="pool2")
46 pool2_flatten = tf.reshape(pool2, [-1, 7*7*64])
47 fc1 = lrelu(dense(pool2_flatten, 1024, name="fc1"))
48 dropout_fc1 = tf.nn.dropout(fc1, 0.5)
49 logits = dense(dropout_fc1, 10, name="fc2")
```

Tensorboard使用-模型结构

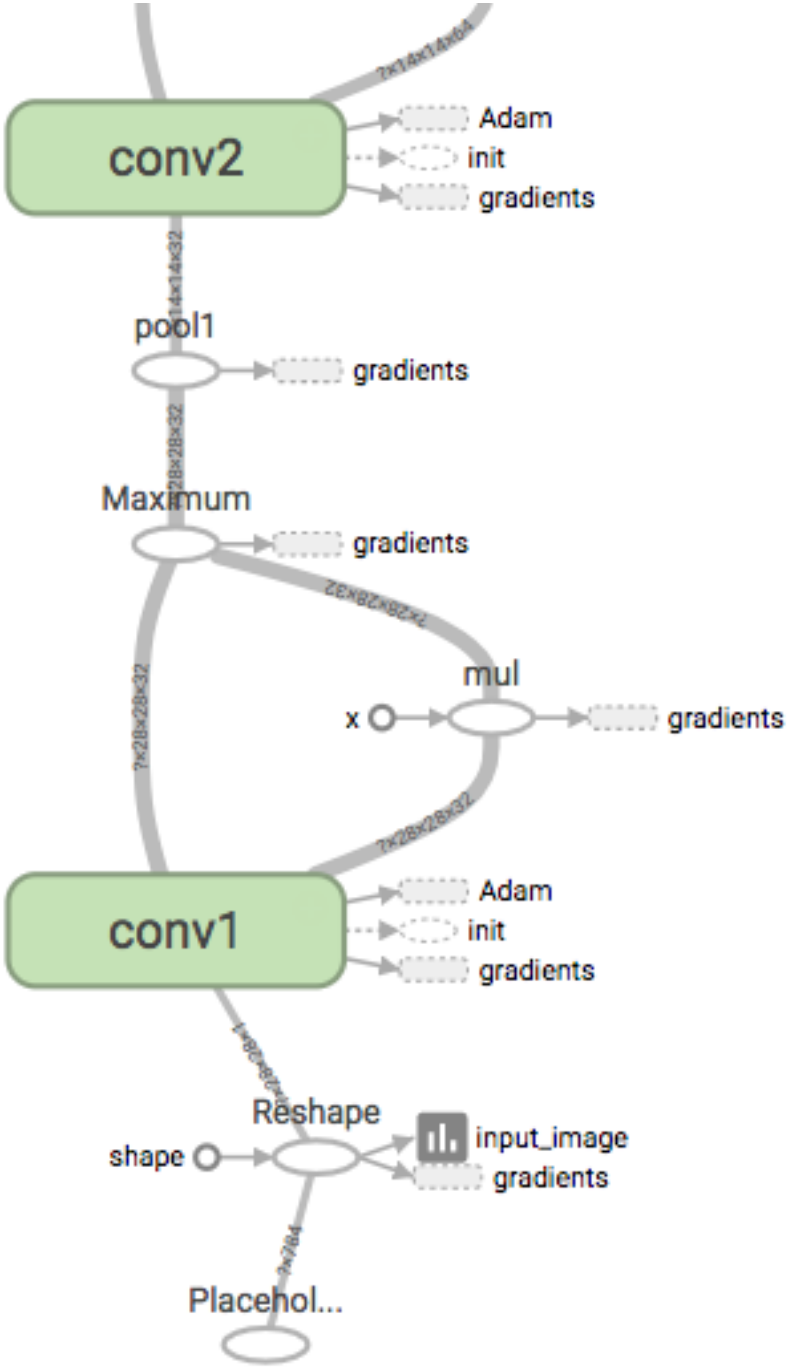
```
62 init = tf.global_variables_initializer()
63
64 with tf.Session() as sess:
65     summary_writer = tf.summary.FileWriter(FLAGS.ckp_dir, sess.graph)
66     sess.run(init)
67     for i in range(100):
68         batch_xs, batch_ys = mnist.train.next_batch(50)
69         cross_entropy_val, _, summary_str = sess.run(
70                 [cross_entropy, train_step, merged_summary],
71                 feed_dict={x: batch_xs, y_: batch_ys})
72         summary_writer.add_summary(summary_str, i)
73         if i % 100 == 0:
74             accuracy_val = sess.run(
75                     accuracy,
76                     feed_dict={x: mnist.test.images[0:1000], y_: mnist.test.
77             print "Epoch %4d: cross_entropy: %4.8f, accuracy: %4.8f" % (i, c
78         else:
79             print "Epoch %4d: cross_entropy: %4.8f" % (i, cross_entropy_val)
80
```

Tensorboard使用-模型结构

- Launch Tensorboard
  - Tensorboard –logdir=dir_path
- Dir_path
  - 单次训练dir
  - 多次训练同级dir的父目录
  - 两个不相干的dir
    - --logdir=name1:path1,name2:path2

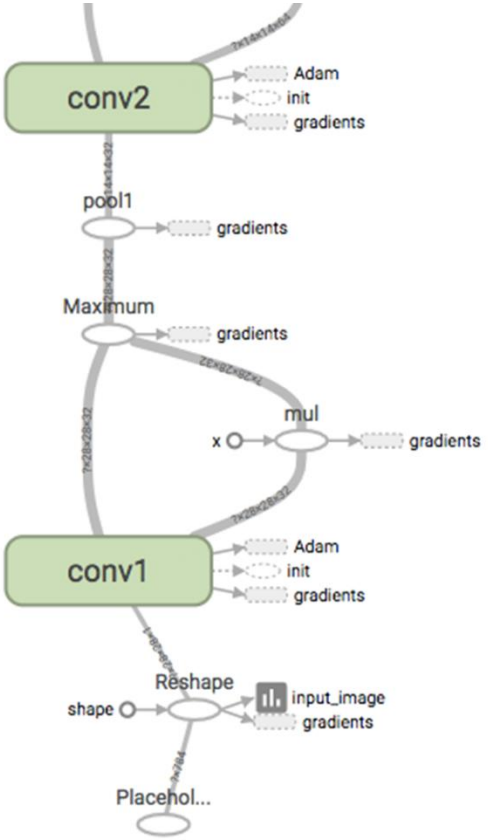

Mean_1
accuracy
Cast
Equal
Mean
logistic_loss
gradients
ArgMax
ArgMax_1
fc2
dropout
fc1
Reshape_1
pool2
conv2
pool1
conv1
Reshape
input_image
gradients
Adam
beta1_power
beta2_power
Adam
init

Ten


conv2
Adam
init
gradients
pool1
gradients
Maximum
gradients
mul
x
gradients
conv1
Adam
init
gradients
Reshape
shape
input_image
gradients
Placehol...

Tensorboard使用-模型结构

- 图很乱



- 命名空间
  - tf.name_scope
  - tf.variable_scope

Tensorboard-模型结构

- 命名空间
  - tf.variable_scope

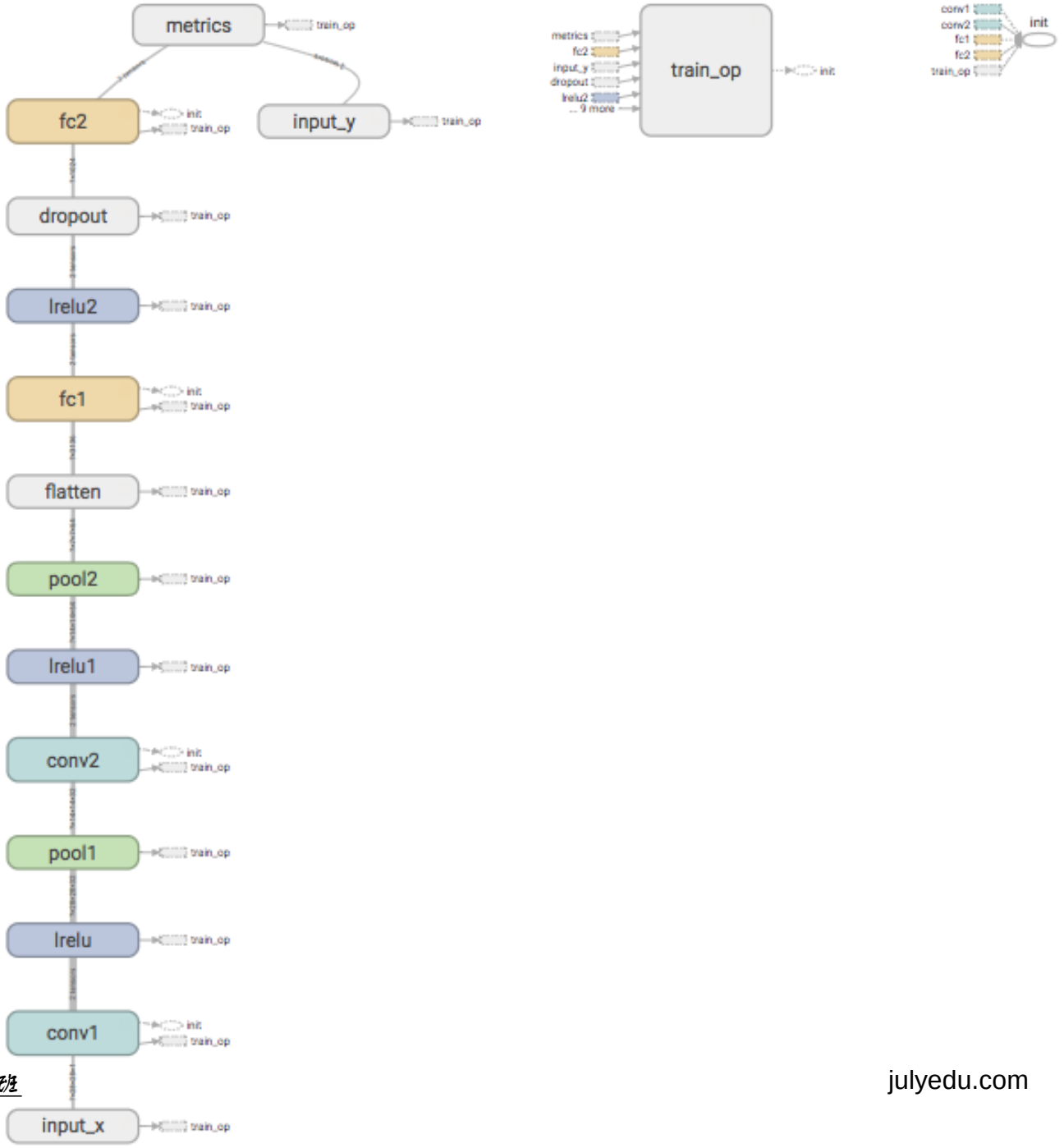
```
with tf.variable_scope("foo"):
    with tf.variable_scope("bar"):
        v = tf.get_variable("v", [1])
        assert v.name == "foo/bar/v:0"

with tf.variable_scope("foo"):
    v = tf.get_variable("v", [1])
with tf.variable_scope("foo", reuse=True):
    v1 = tf.get_variable("v", [1])
assert v1 == v


with tf.variable_scope("foo") as scope:
    v = tf.get_variable("v", [1])
    scope.reuse_variables()
    v1 = tf.get_variable("v", [1])
assert v1 == v
```

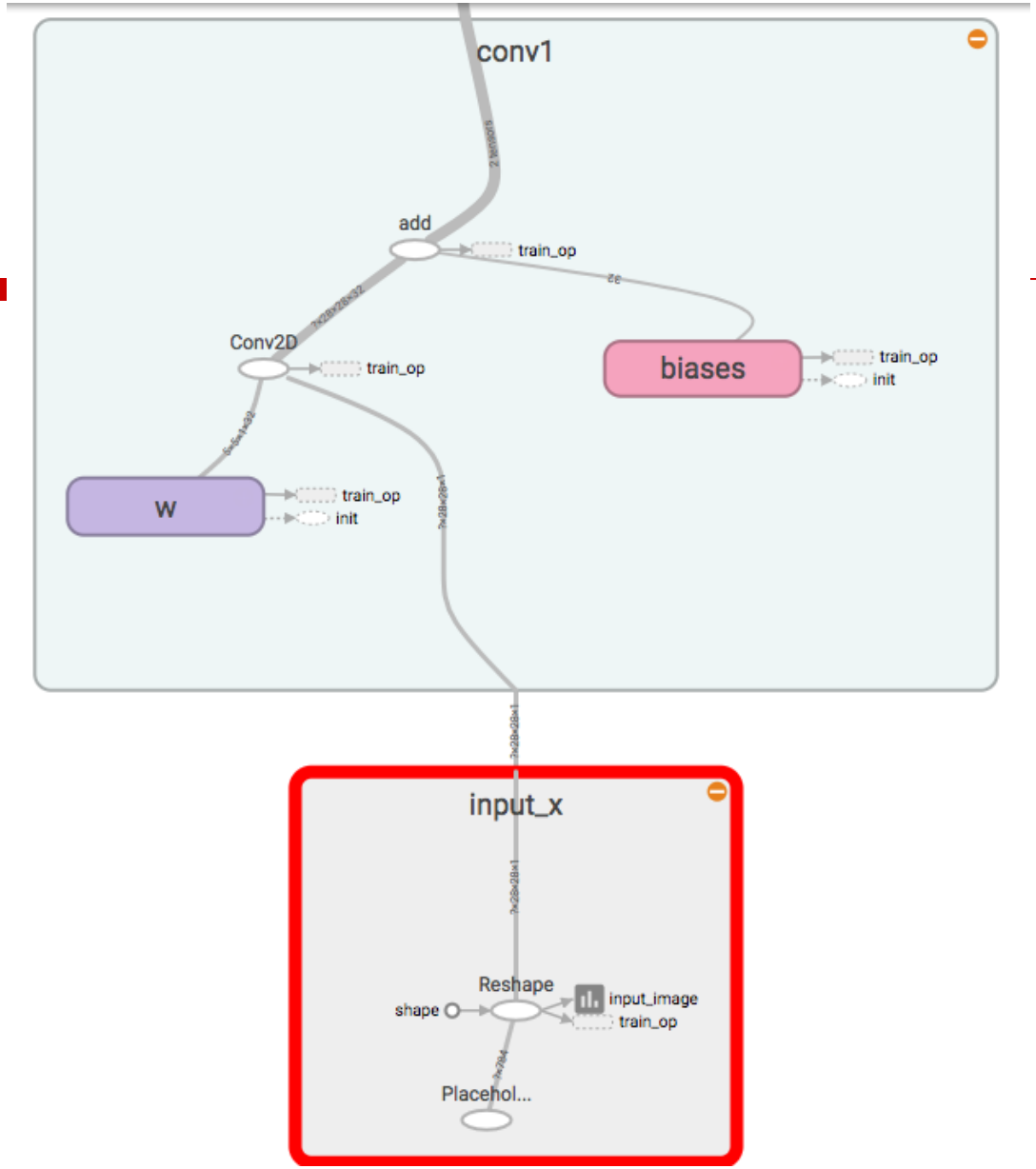
Tensorboard使用-模型结构

```
21 def max_pool_2d(x, k_h=2, k_w=2, s_h=2, s_w=2, name="max_pool"):
22     with tf.name_scope(name):
23         return tf.nn.max_pool(
24                 x,
25                 ksize=[1, k_h, k_w, 1],
26                 strides=[1, s_h, s_w, 1],
27                 padding="SAME",
28                 name='pool')
29
30 def dense(x, output_dim, stddev=0.02, name="dense"):
31     with tf.variable_scope(name):
32         w = tf.get_variable('w', [x.get_shape()[-1], output_dim],
33                 initializer = tf.truncated_normal_initializer(stddev=stddev))
34         biases = tf.get_variable('biases', [output_dim],
35                 initializer = tf.constant_initializer(0.0))
36         return tf.matmul(x, w) + biases
37
38 def lrelu(x, name="lrelu"):
39     with tf.name_scope(name):
40         return tf.maximum(x, 0.2 * x)
```

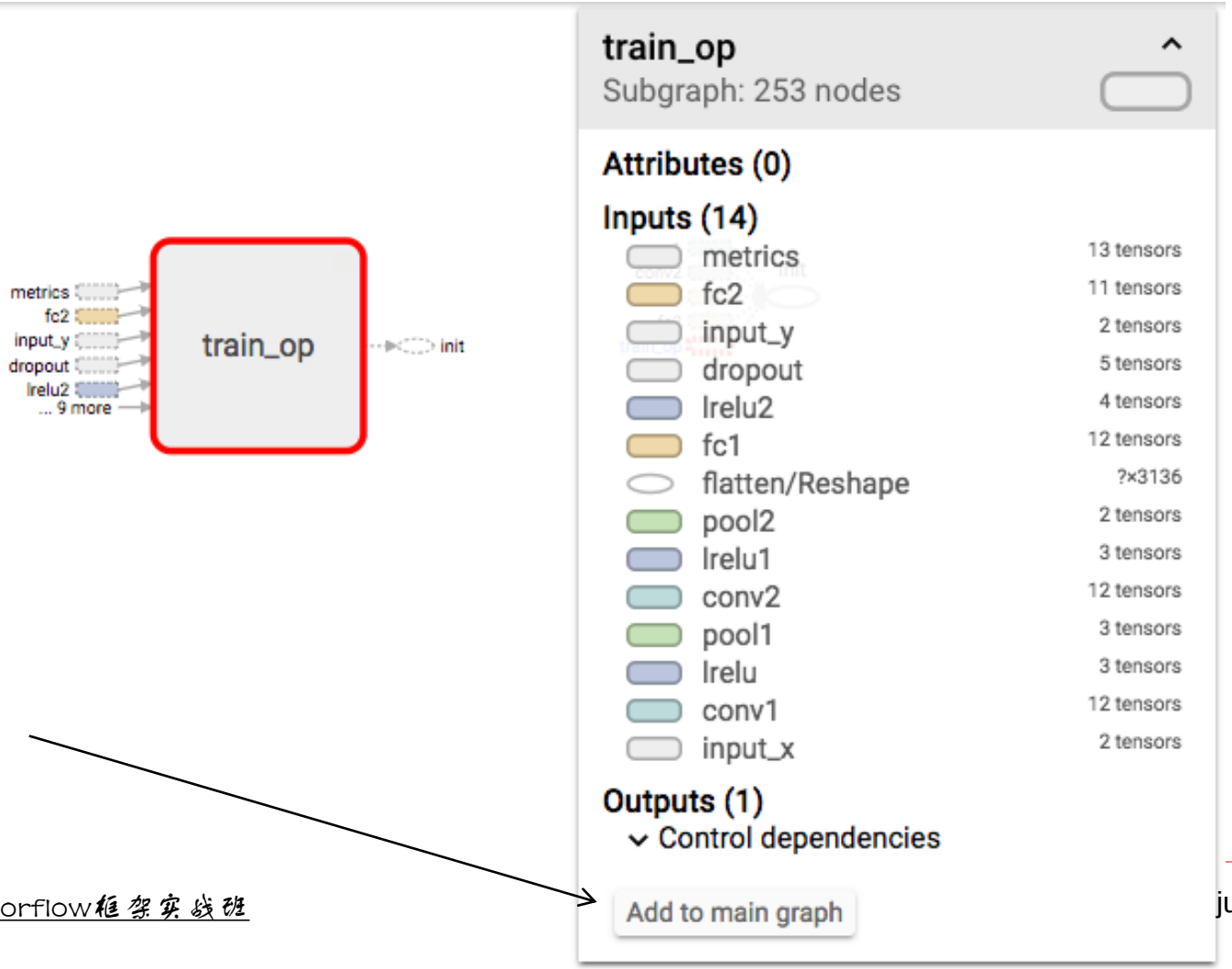
Tensorboard使用-模型结构

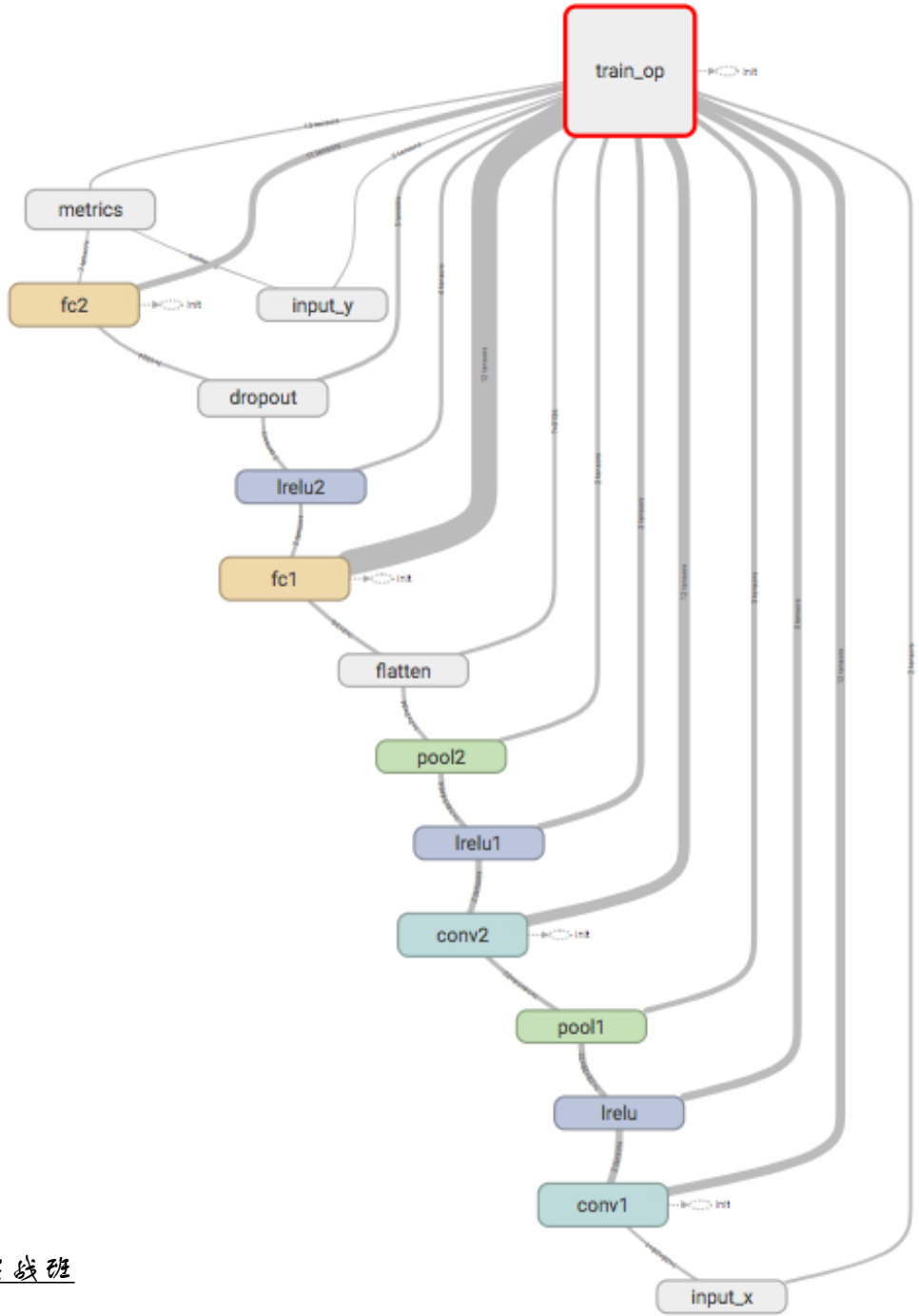
```
42 with tf.name_scope("input_x"):
43     x = tf.placeholder(tf.float32, [None, 784])
44     x_image = tf.reshape(x, [-1, 28, 28, 1])
45 conv1 = lrelu(conv2d(x_image, 32, name="conv1"))
46 pool1 = max_pool_2d(conv1, name="pool1")
47 conv2 = lrelu(conv2d(pool1, 64, name="conv2"), name='lrelu1')
48 pool2 = max_pool_2d(conv2, name="pool2")
49 with tf.name_scope("flatten"):
50     pool2_flatten = tf.reshape(pool2, [-1, 7*7*64])
51 fc1 = lrelu(dense(pool2_flatten, 1024, name="fc1"), name='lrelu2')
52 dropout_fc1 = tf.nn.dropout(fc1, 0.5)
53 logits = dense(dropout_fc1, 10, name="fc2")
54
55 with tf.name_scope("input_y"):
56     y_ = tf.placeholder(tf.float32, [None, 10])
57 with tf.name_scope("metrics"):
58     cross_entropy = tf.reduce_mean(
59         tf.nn.sigmoid_cross_entropy_with_logits(logits=logits, labels=y_))
60     correct_prediction = tf.equal(tf.argmax(logits,1), tf.argmax(y_,1))
61     accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
```


metrics
train_op
fc2
init
train_op
input_y
train_op
dropout
train_op
lrelu2
train_op
fc1
init
train_op
flatten
train_op
pool2
train_op
lrelu1
train_op
conv2
init
train_op
pool1
train_op
lrelu
train_op
conv1
init
train_op
input_x
train_op
metrics
fc2
input_y
dropout
lrelu2
... 9 more
train_op
init
conv1
conv2
fc1
fc2
train_op
init

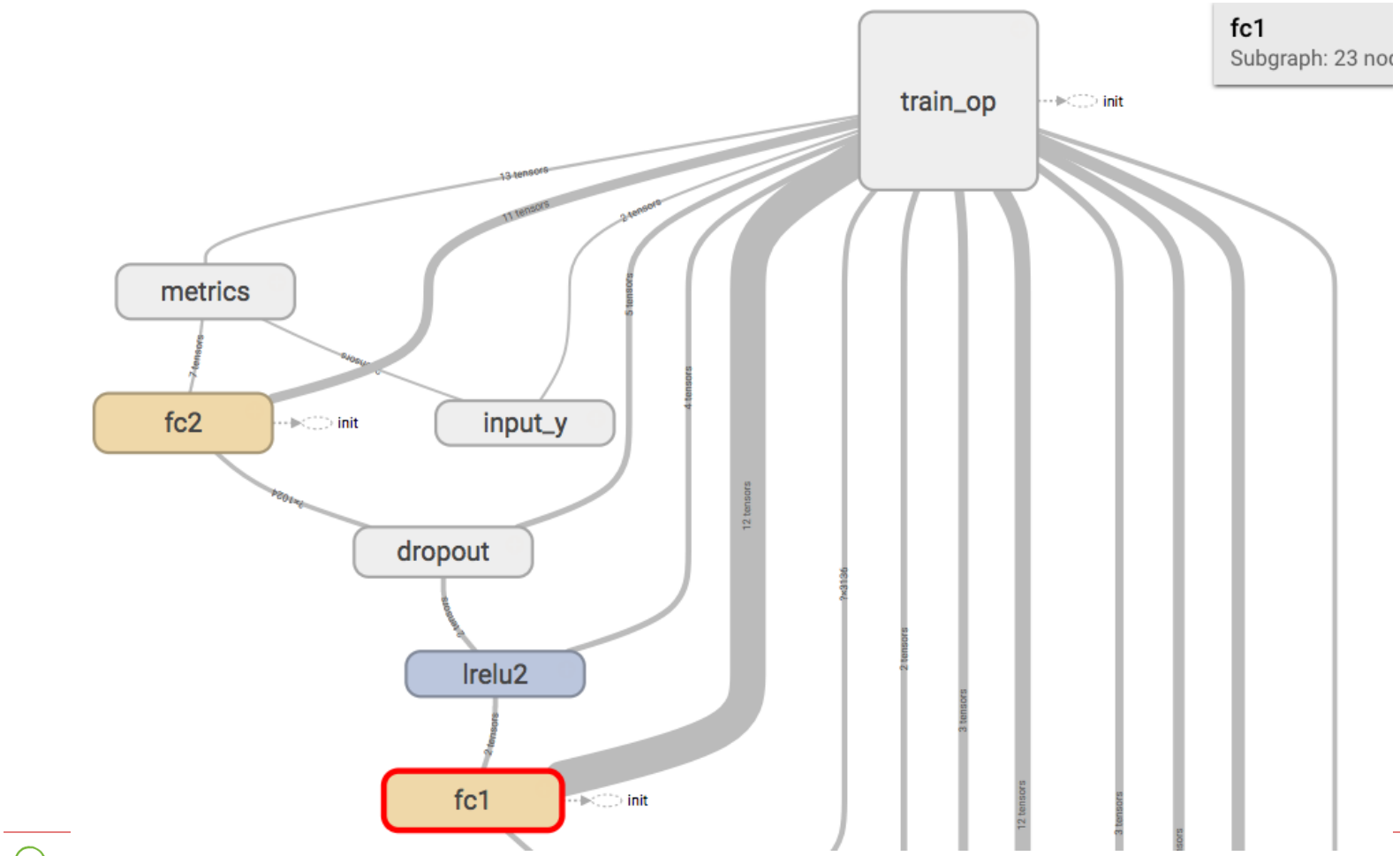

conv1
add
train_op
Conv2D
train_op
biases
train_op
init
W
train_op
init
input_x
Reshape
shape
input_image
train_op
Placehol...

Tensorboard使用-模型结构




train_op
metrics
fc2
input_y
dropout
lrelu2
fc1
flatten
pool2
lrelu1
conv2
pool1
lrelu
conv1
input_x

Tensorboard使用-模型结构



Tensorboard使用-模型结构

Symbol	Meaning
	High-level node representing a name scope. Double-click to expand a high-level node.
	Sequence of numbered nodes that are not connected to each other.
	Sequence of numbered nodes that are connected to each other.
	An individual operation node.
	A constant.
	A summary node.
	Edge showing the data flow between operations.
	Edge showing the control dependency between operations.
	A reference edge showing that the outgoing operation node can mutate the incoming tensor.

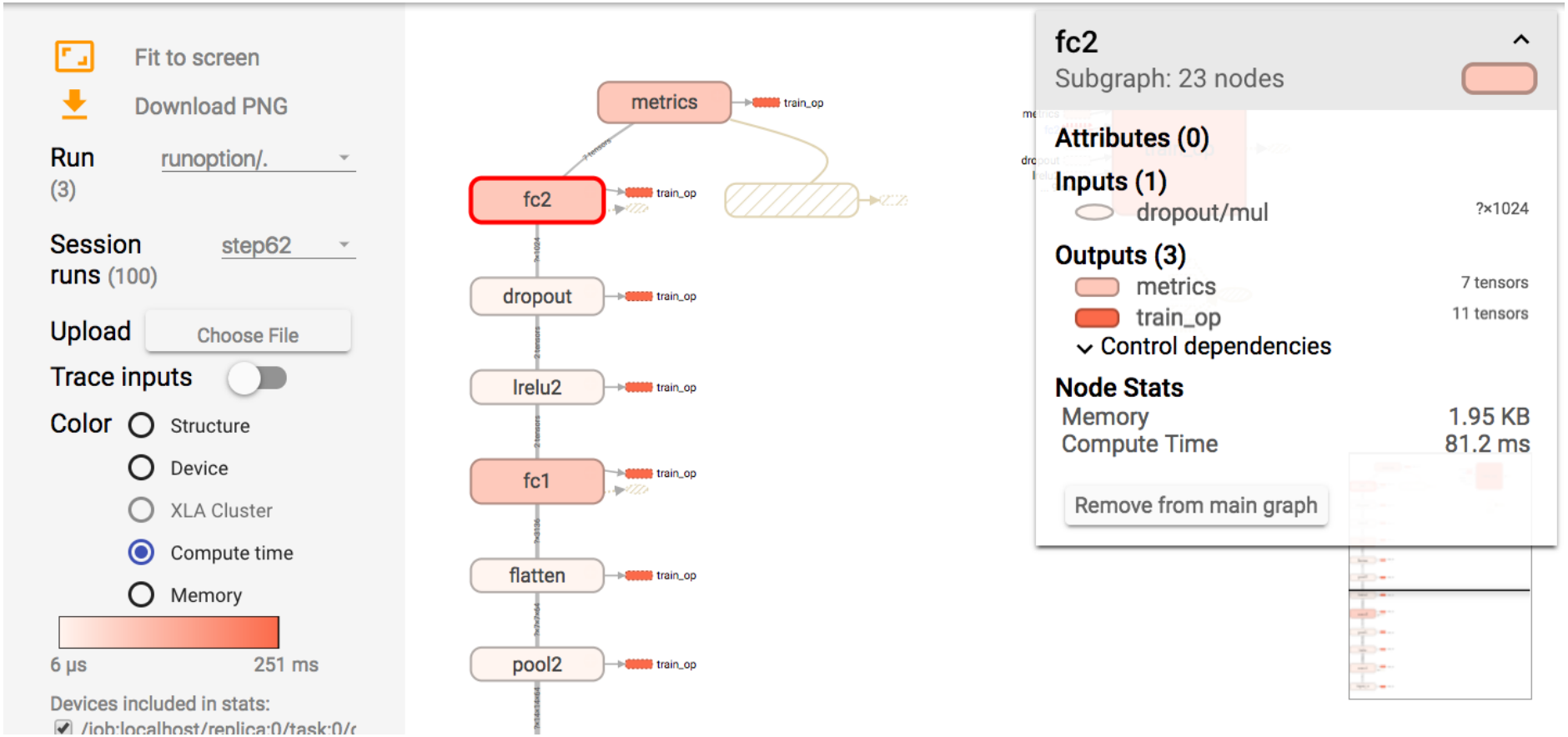
Tensorboard使用-模型结构

- 就这样？
- System trace
  - Memory
  - Compute time

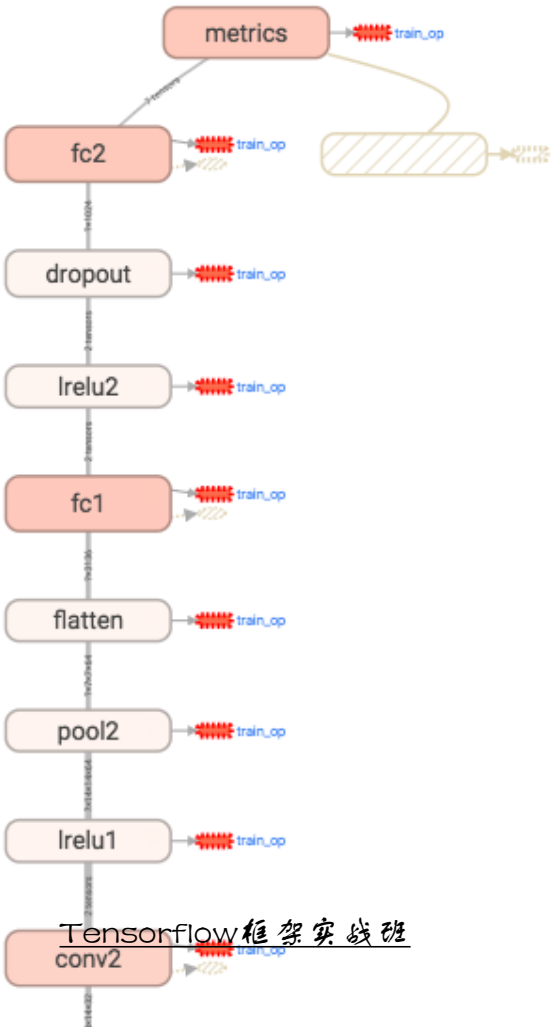
Tensorboard使用-模型结构

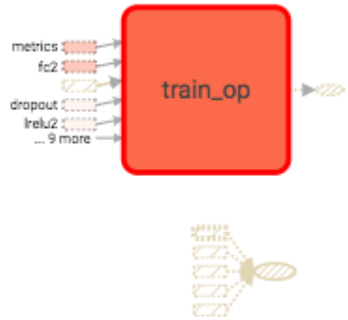
```
71 with tf.Session() as sess:
72     summary_writer = tf.summary.FileWriter(FLAGS.ckp_dir, sess.graph)
73     sess.run(init)
74     for i in range(100):
75         batch_xs, batch_ys = mnist.train.next_batch(50)
76         run_options = tf.RunOptions(trace_level=tf.RunOptions.FULL_TRACE)
77         run_metadata = tf.RunMetadata()
78         cross_entropy_val, _, summary_str = sess.run(
79                 [cross_entropy, train_step, merged_summary],
80                 feed_dict={x: batch_xs, y_: batch_ys},
81                 options=run_options,
82                 run_metadata=run_metadata)
83         summary_writer.add_run_metadata(run_metadata, 'step%d' % i)
84         summary_writer.add_summary(summary_str, i)
85         if i % 100 == 0:
86             accuracy_val = sess.run(
87                     accuracy,
88                     feed_dict={x: mnist.test.images[0:1000], y_: mnist.test.l
89             print "Epoch %4d: cross_entropy: %4.8f, accuracy: %4.8f" % (i, cr
90         else:
91             print "Epoch %4d: cross_entropy: %4.8f" % (i, cross_entropy_val)
92
```

Tensorboard使用-模型结构



Tensorboard使用-模型结构





train_op

Subgraph: 253 nodes

Attributes (0)

Inputs (14)

metrics	13 tensors
fc2	11 tensors
input_y	2 tensors
dropout	5 tensors
lrelu2	4 tensors
fc1	12 tensors
flatten/Reshape	?×3136
pool2	2 tensors
lrelu1	3 tensors
conv2	12 tensors
pool1	3 tensors
lrelu	3 tensors
conv1	12 tensors
input_x	2 tensors

Outputs (1)

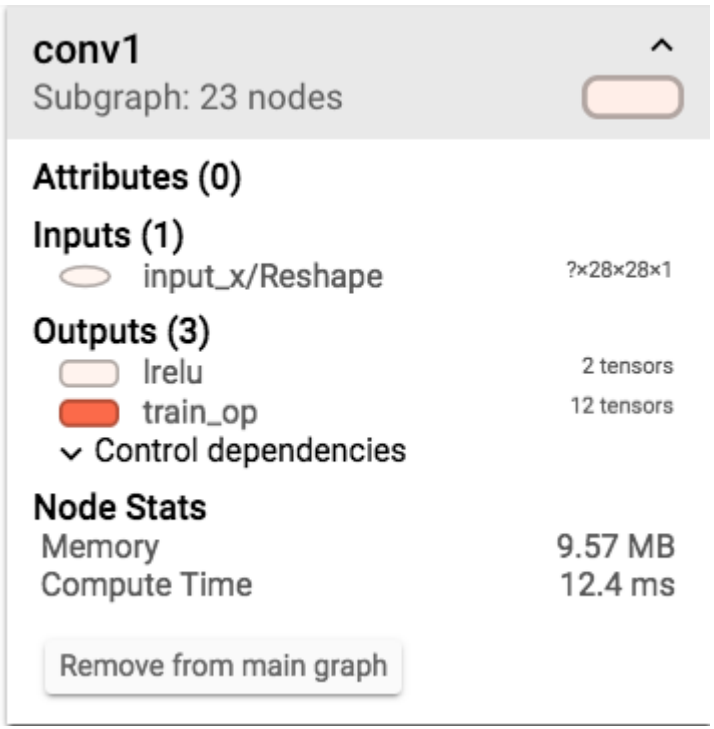
Control dependencies

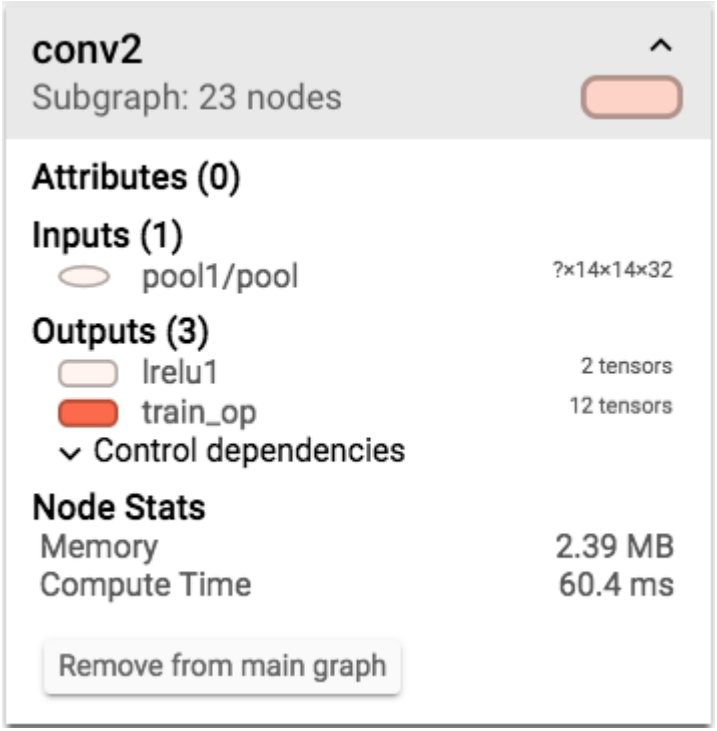
Node Stats

Memory	121 MB
Compute Time	246 ms

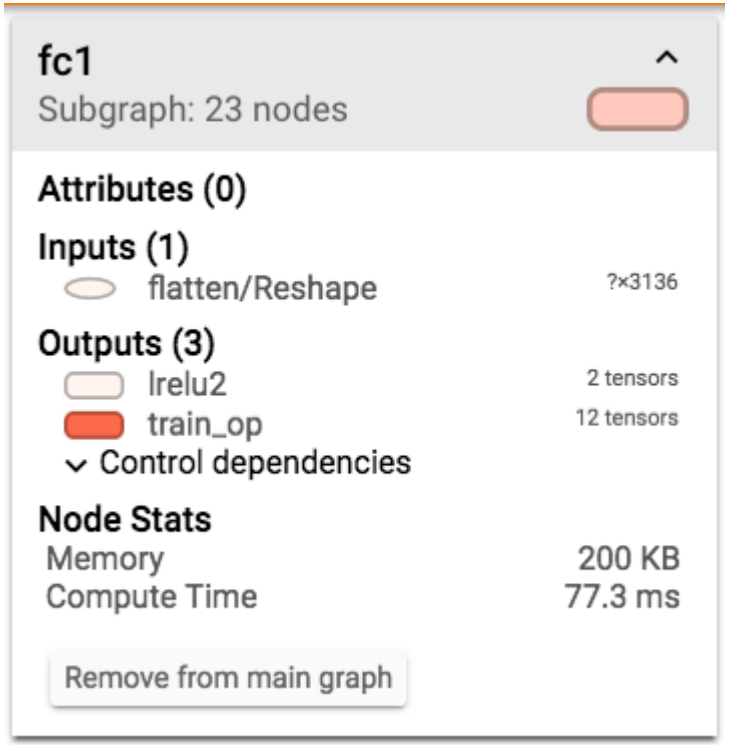
Add to main graph

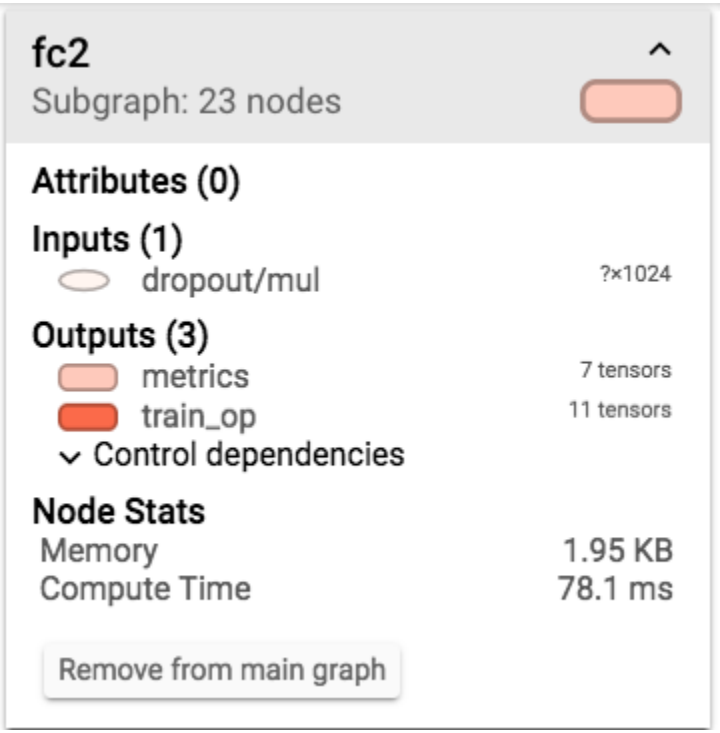
Tensorboard使用-模型结构



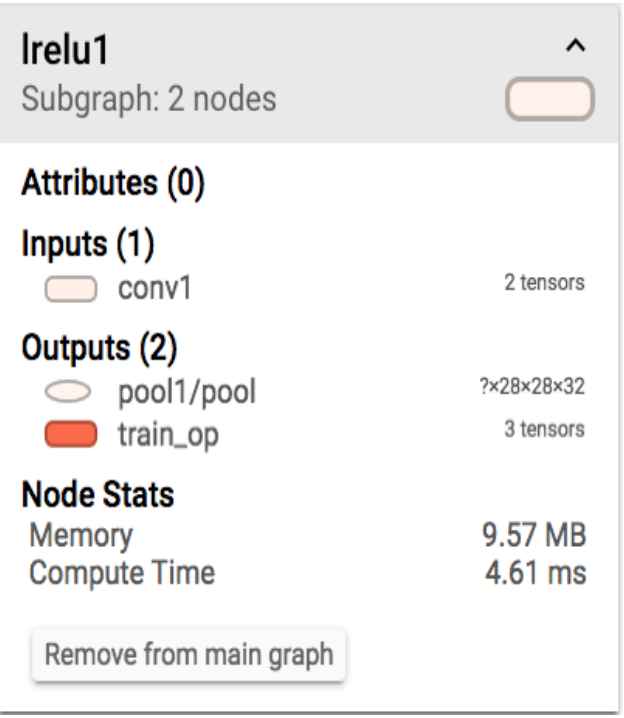


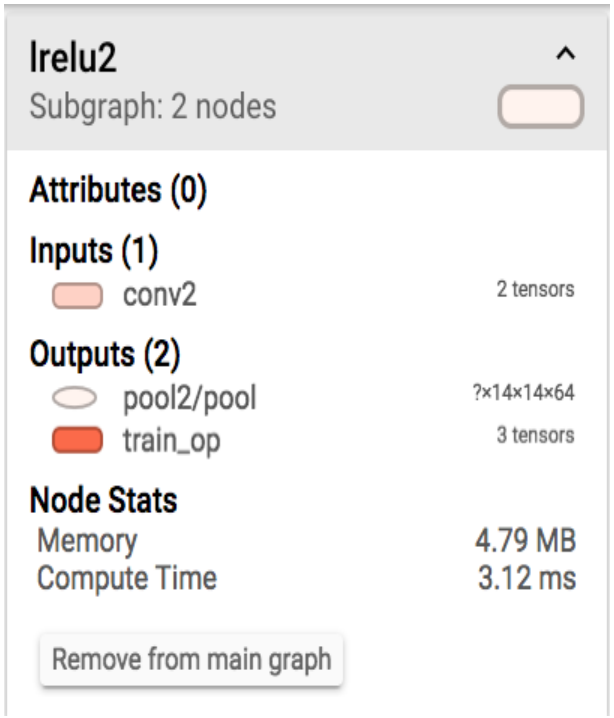
Tensorboard使用-模型结构

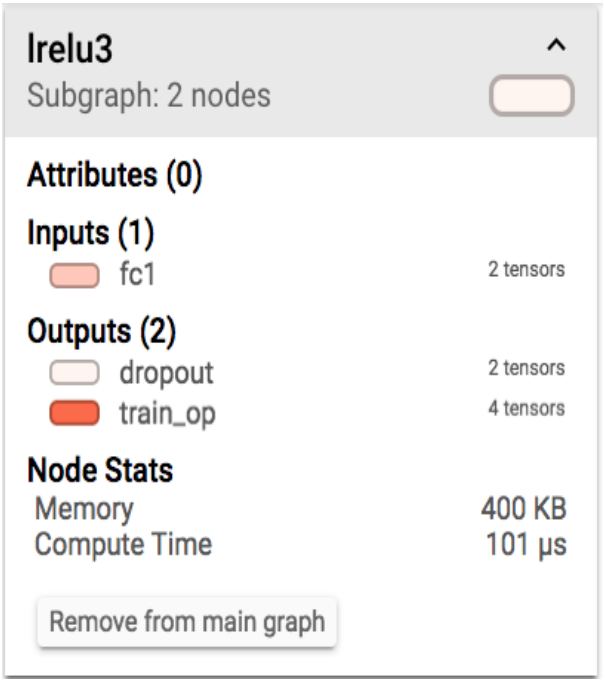




Tensorboard使用-模型结构





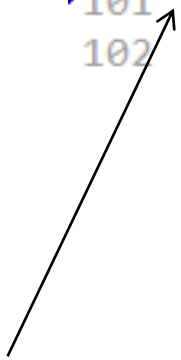


Tensorboard使用-训练状态

- 回顾
  - tf.summary.tensor_summary
  - tf.summary.scalar
  - tf.summary.histogram
  - tf.summary.audio
  - tf.summary.image
  - tf.summary.merge
  - tf.summary.merge_all

Tensorboard使用-训练状态

```
93  with tf.name_scope("metrics"):
94      cross_entropy = tf.reduce_mean(
95          tf.nn.sigmoid_cross_entropy_with_logits(logits=logits, labels=y
96      correct_prediction = tf.equal(tf.argmax(logits,1), tf.argmax(y_,1))
97      accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
98  cross_entropy_summary = tf.summary.scalar('cross_entropy', cross_entrop
99  accuracy_summary = tf.summary.scalar('accuracy', accuracy)
100 tf.summary.image("input_image", x_image)
101 merged_summary = tf.summary.merge_all()
102 merged_summary_test = tf.summary.merge([cross_entropy_summary, accuracy
```



Tensorboard使用-训练状态

```
train_writer = tf.summary.FileWriter(train_log_dir, sess.graph)
test_writer  = tf.summary.FileWriter(test_log_dir)
sess.run(init)
for i in range(100):
    batch_xs, batch_ys = mnist.train.next_batch(50)
    run_options = tf.RunOptions(trace_level=tf.RunOptions.FULL_TRACE)
    run_metadata = tf.RunMetadata()
    cross_entropy_val, _, summary_str = sess.run(
            [cross_entropy, train_step, merged_summary],
            feed_dict={x: batch_xs, y_: batch_ys},
            options=run_options,
            run_metadata=run_metadata)
    train_writer.add_run_metadata(run_metadata, 'step%d' % i)
    train_writer.add_summary(summary_str, i)
    if (i+1) % 10 == 0:
        accuracy_val, summary_str = sess.run(
                [accuracy, merged_summary_test],
                feed_dict={x: mnist.test.images[0:1000], y_: mnist.test.labels[0:1000]})
        test_writer.add_summary(summary_str, i)
        print "Epoch %4d: cross_entropy: %4.8f, accuracy: %4.8f" % (i, cross_entropy_val,
    else:
        print "Epoch %4d: cross_entropy: %4.8f" % (i, cross_entropy_val)
```

Tensorboard使用-训练状态

```
12 def variable_summaries(var):
13     with tf.name_scope('summaries'):
14         mean = tf.reduce_mean(var)
15         with tf.name_scope('stddev'):
16             stddev = tf.sqrt(tf.reduce_mean(tf.square(var - mean)))
17         tf.summary.scalar('mean', mean)
18         tf.summary.scalar('stddev', stddev)
19         tf.summary.scalar('min', tf.reduce_min(var))
20         tf.summary.scalar('max', tf.reduce_max(var))
21         tf.summary.histogram('histogram', var)
```

Tensorboard使用-训练状态

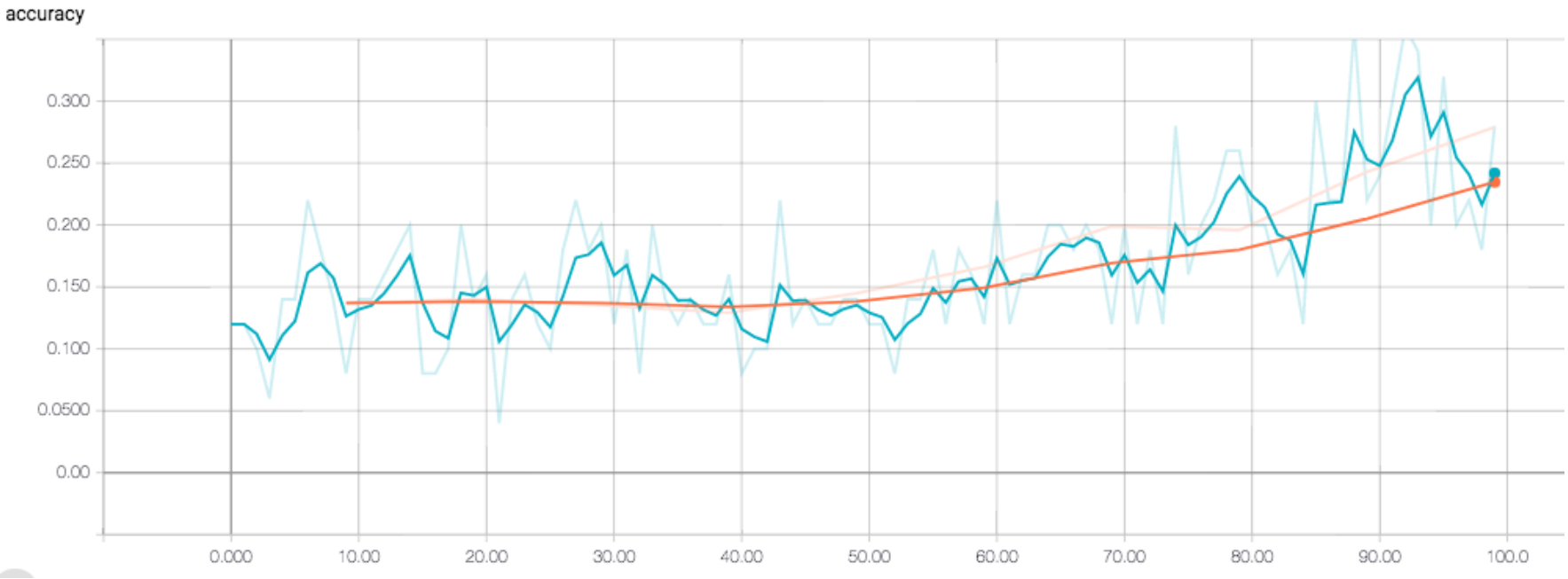
```
33 def conv2d(x, output_dim, k_h=5, k_w=5, s_h=1, s_w=1, stddev=0.02, name="conv2d"):
34     with tf.variable_scope(name):
35         w = tf.get_variable('w', [k_h, k_w, x.get_shape()[-1], output_dim],
36                 initializer = tf.truncated_normal_initializer(stddev=stddev))
37         conv = tf.nn.conv2d(x, w, strides=[1, s_h, s_w, 1], padding="SAME")
38         biases = tf.get_variable('biases', [output_dim],
39                 initializer = tf.constant_initializer(0.0))
40         conv = conv + biases
41         with tf.name_scope('w'):
42             variable_summaries(w)
43         with tf.name_scope('biases'):
44             variable_summaries(biases)
45         with tf.name_scope('conv'):
46             variable_summaries(conv)
47         return conv
```

Tensorboard使用-训练状态

```
60 def dense(x, output_dim, stddev=0.02, name="dense"):
61     with tf.variable_scope(name):
62         w = tf.get_variable('w', [x.get_shape()[-1], output_dim],
63                 initializer = tf.truncated_normal_initializer(stddev=stddev))
64         biases = tf.get_variable('biases', [output_dim],
65                 initializer = tf.constant_initializer(0.0))
66         result = tf.matmul(x, w) + biases
67         with tf.name_scope('w'):
68             variable_summaries(w)
69         with tf.name_scope('biases'):
70             variable_summaries(biases)
71         with tf.name_scope('result'):
72             variable_summaries(result)
73         return result


75 def lrelu(x, name="lrelu"):
76     with tf.name_scope(name):
77         result = tf.maximum(x, 0.2 * x)
78         variable_summaries(result)
79         return result
```

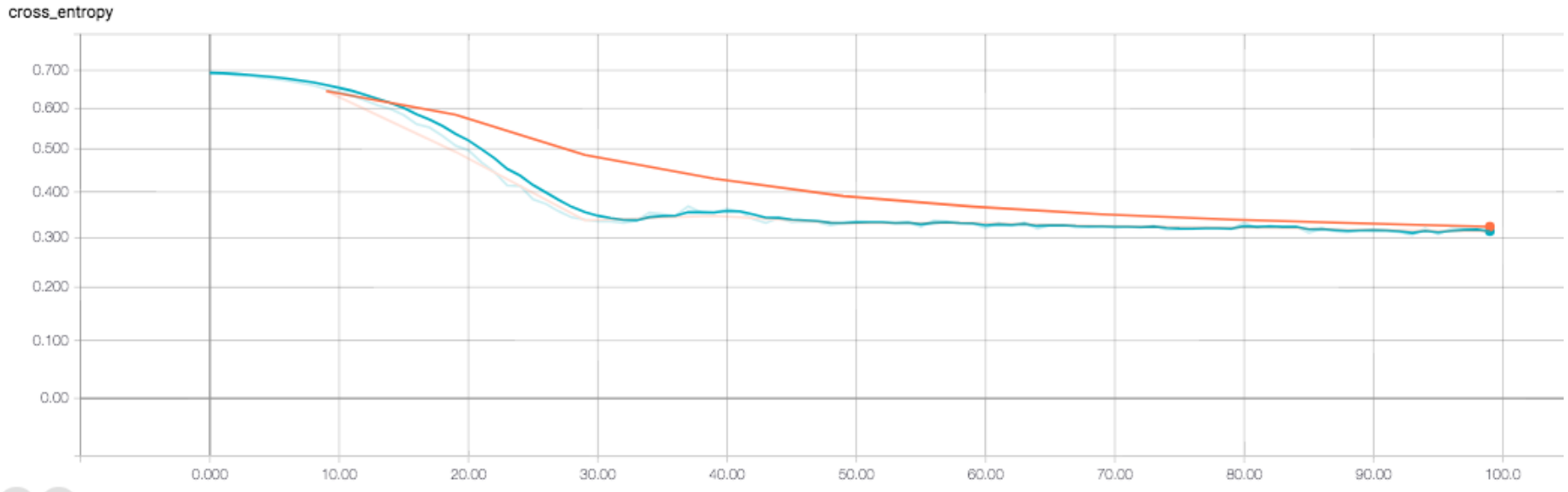
Tensorboard使用-训练状态

Accuracy随迭代次数变化图



Tensorboard使用-训练状态

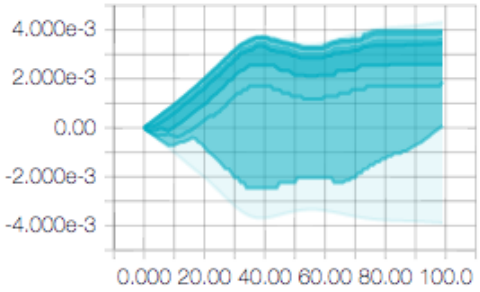
Cross_entropy随迭代次数变化图



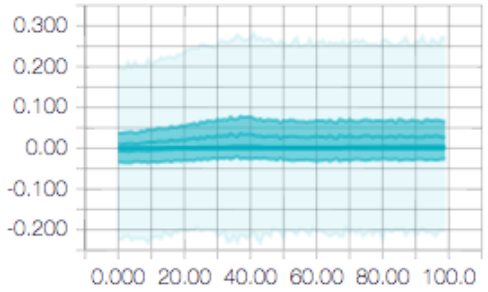
参数Distribution图

conv1

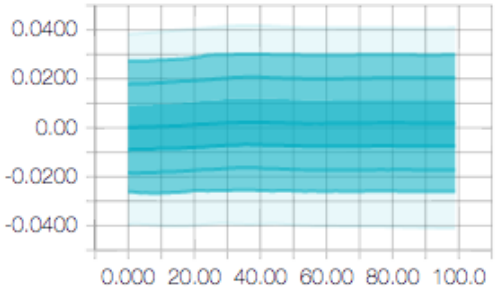
conv1/biases_1/summaries/histogram
train



conv1/conv/summaries/histogram
train

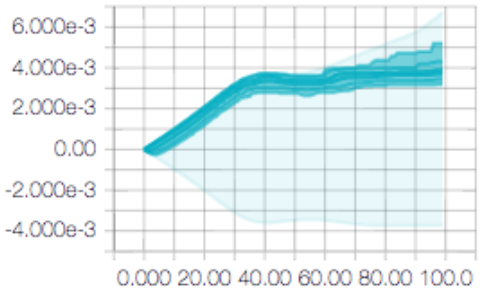


conv1/w_1/summaries/histogram
train

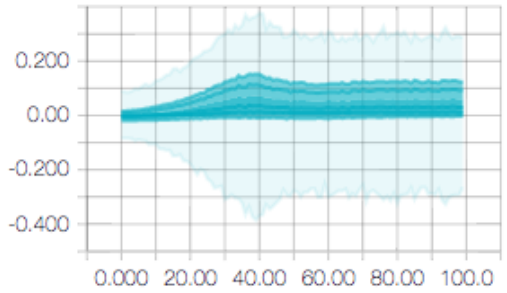


conv2

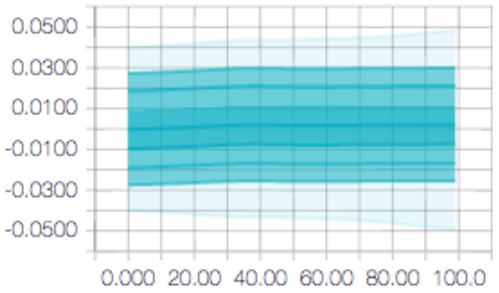
conv2/biases_1/summaries/histogram
train



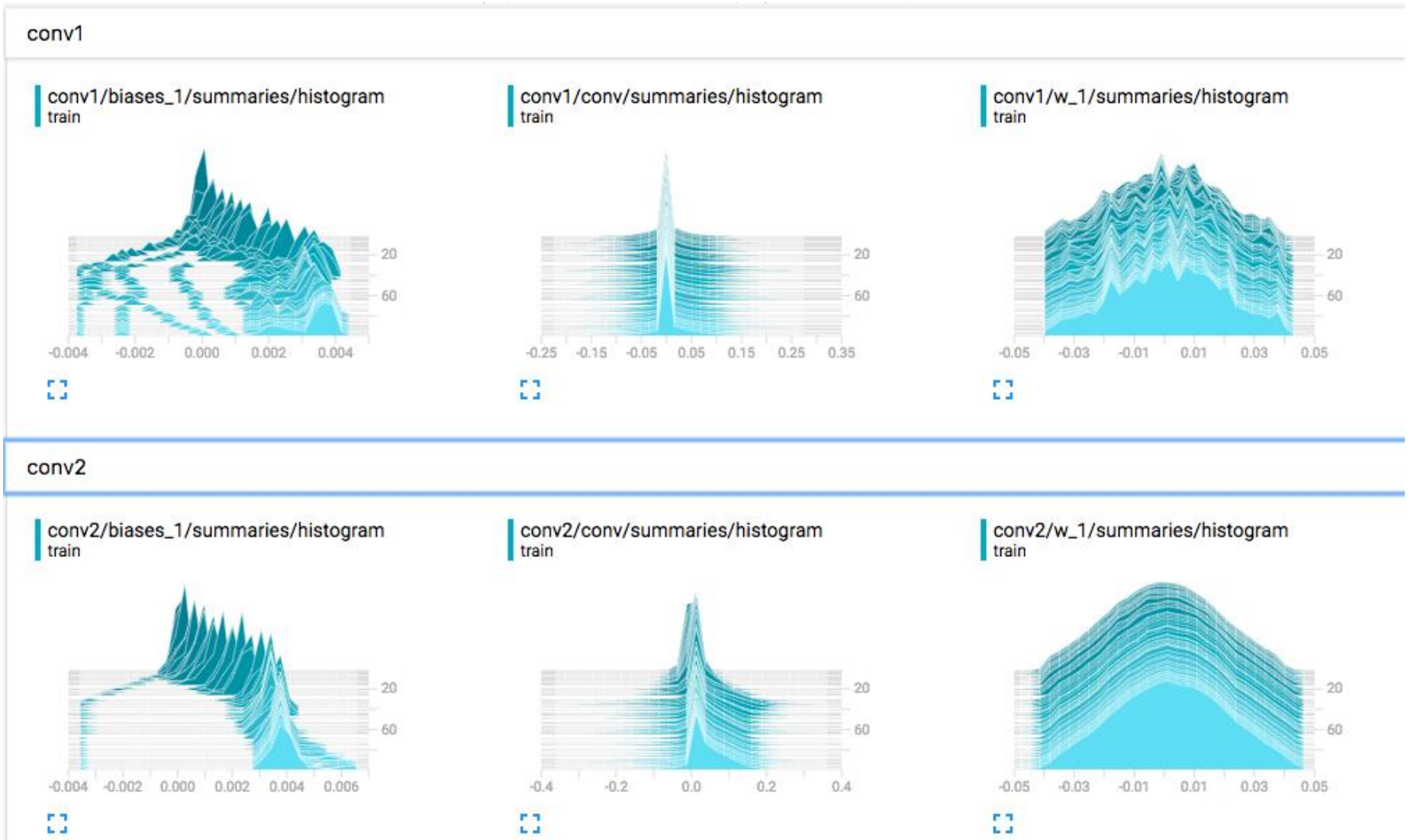
conv2/conv/summaries/histogram
train

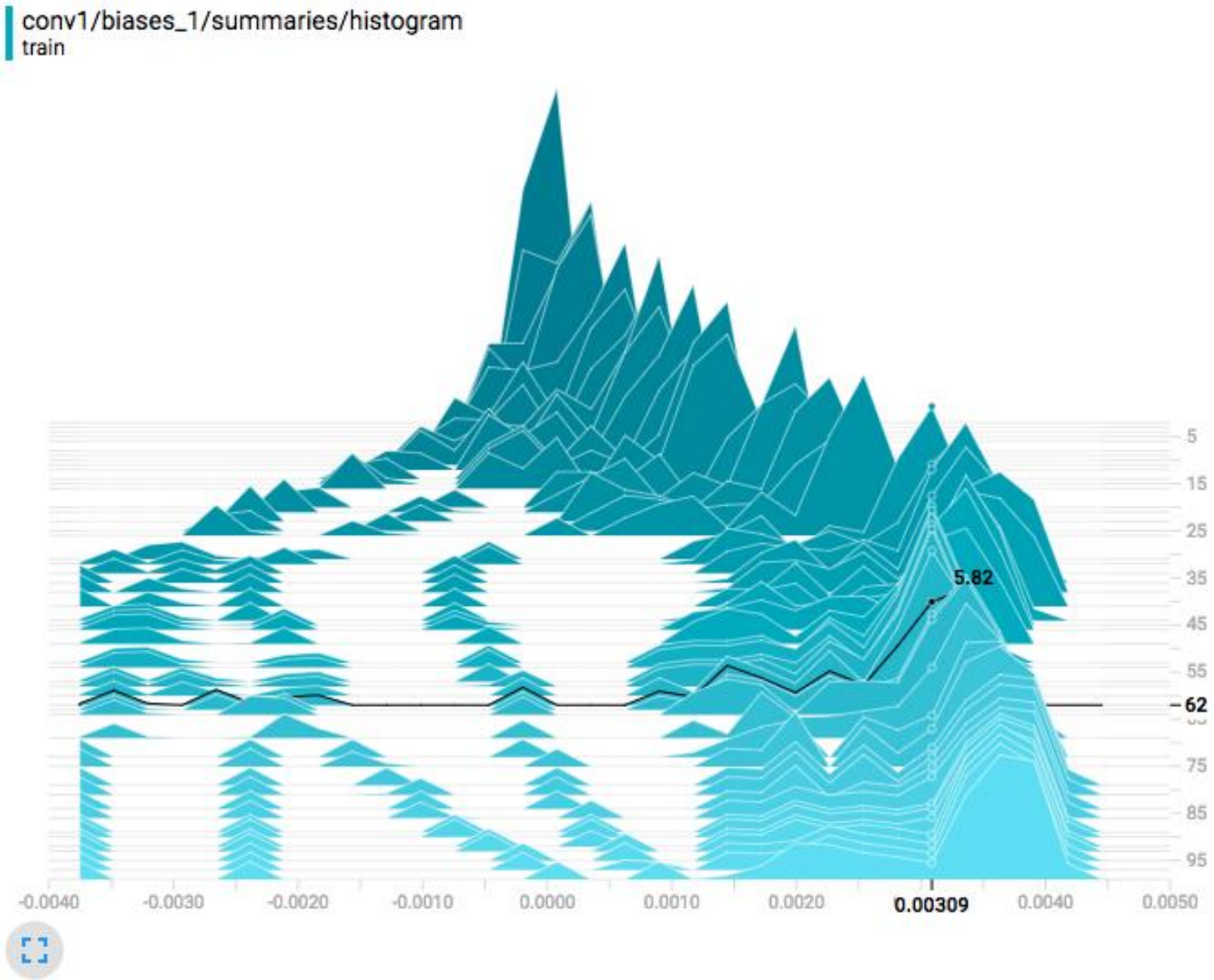


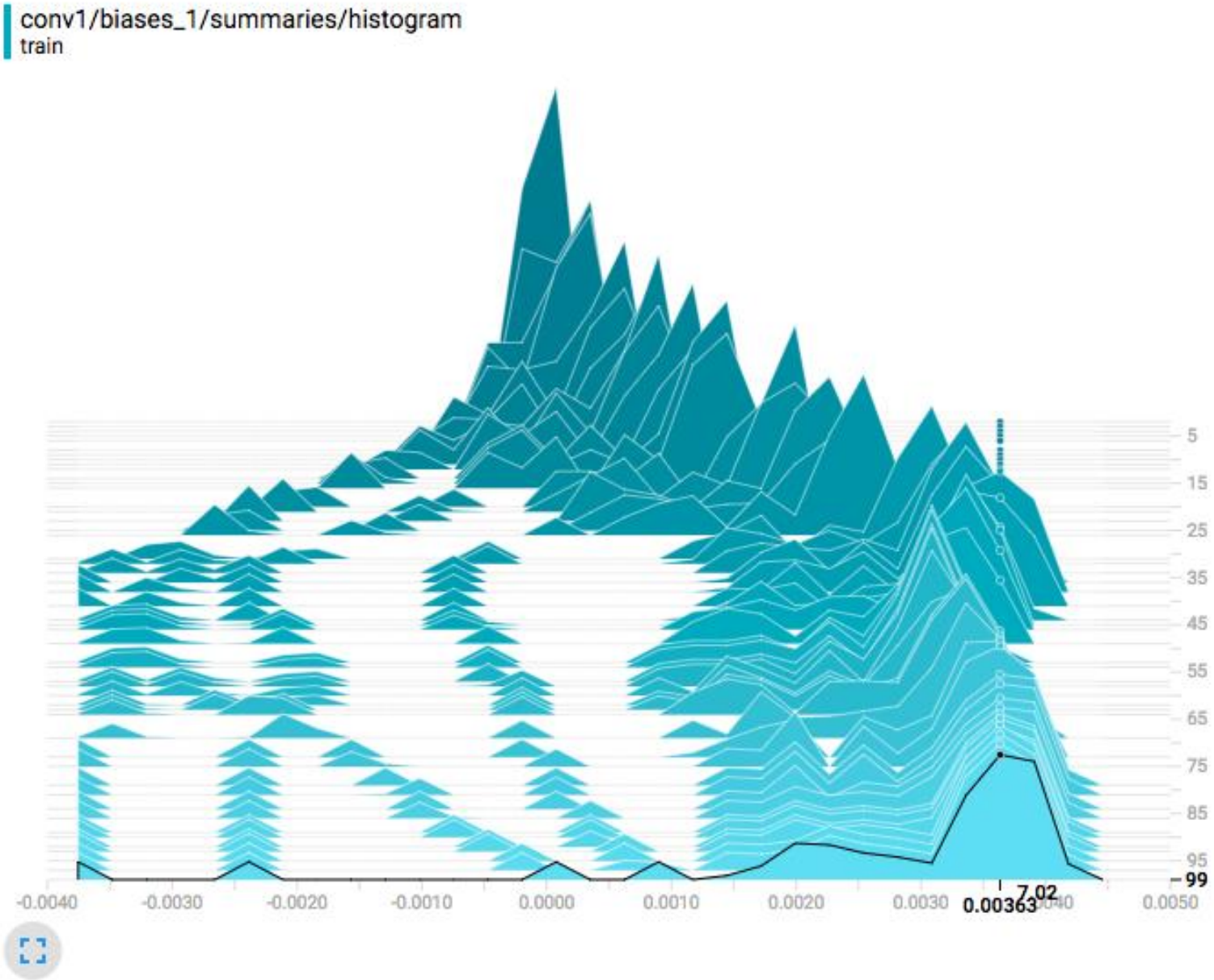
conv2/w_1/summaries/histogram
train

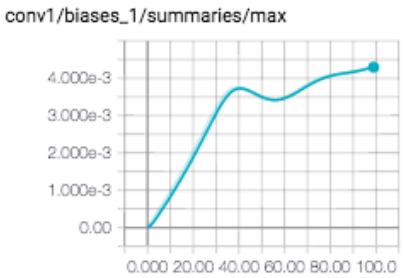


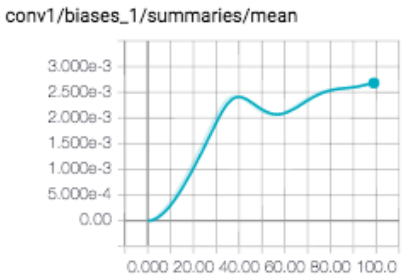
参数histogram

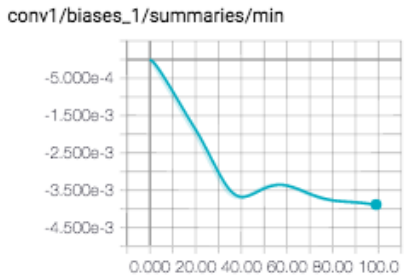


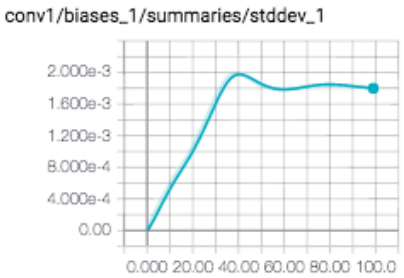

conv1/biases_1/summaries/histogram
train
5
15
25
35
45
55
62
75
85
95
5.82
-0.0040
-0.0030
-0.0020
-0.0010
0.0000
0.0010
0.0020
0.00309
0.0040
0.0050

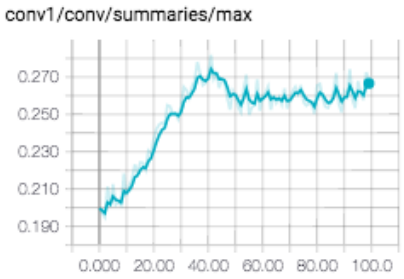

conv1/biases_1/summaries/histogram
train
5
15
25
35
45
55
65
75
85
95
99
-0.0040
-0.0030
-0.0020
-0.0010
0.0000
0.0010
0.0020
0.0030
0.0050
7.02
0.00363

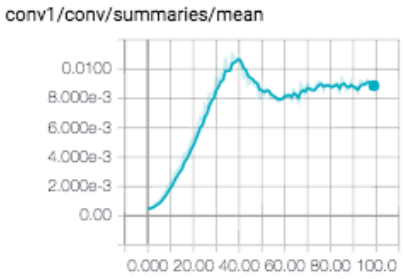

conv1/biases_1/summaries/max
4.000e-3
3.000e-3
2.000e-3
1.000e-3
0.00
0.000 20.00 40.00 60.00 80.00 100.0

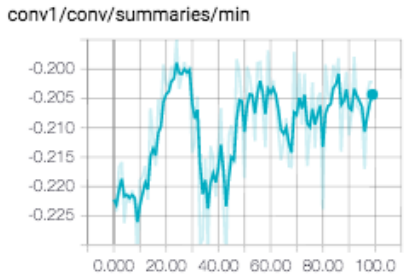

conv1/biases_1/summaries/mean
3.000e-3
2.500e-3
2.000e-3
1.500e-3
1.000e-3
5.000e-4
0.00
0.000 20.00 40.00 60.00 80.00 100.0

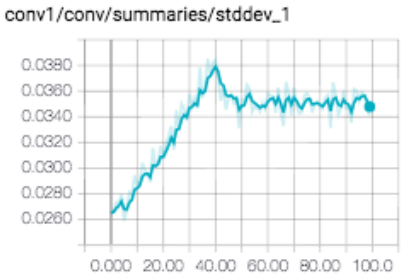

conv1/biases_1/summaries/min
-5.000e-4
-1.500e-3
-2.500e-3
-3.500e-3
-4.500e-3
0.000 20.00 40.00 60.00 80.00 100.0

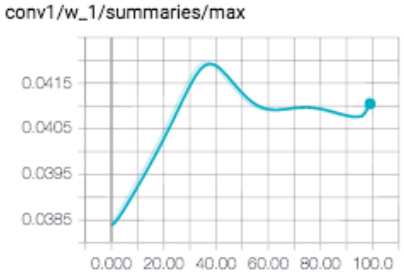

conv1/biases_1/summaries/stddev_1
2.000e-3
1.600e-3
1.200e-3
8.000e-4
4.000e-4
0.00
0.000 20.00 40.00 60.00 80.00 100.0

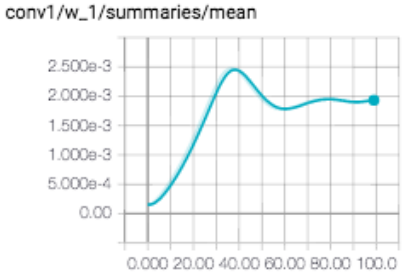

conv1/conv/summaries/max
0.270
0.250
0.230
0.210
0.190
0.000 20.00 40.00 60.00 80.00 100.0

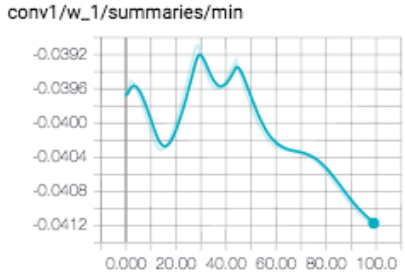

conv1/conv/summaries/mean
0.0100
8.000e-3
6.000e-3
4.000e-3
2.000e-3
0.00
0.000 20.00 40.00 60.00 80.00 100.0

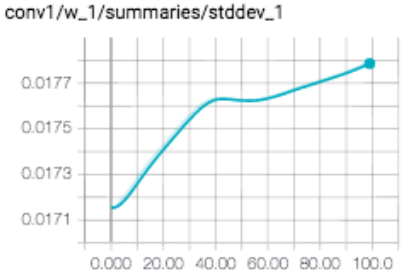

conv1/conv/summaries/min
-0.200
-0.205
-0.210
-0.215
-0.220
-0.225
0.000 20.00 40.00 60.00 80.00 100.0


conv1/conv/summaries/stddev_1
0.0380
0.0360
0.0340
0.0320
0.0300
0.0280
0.0260
0.000 20.00 40.00 60.00 80.00 100.0


conv1/w_1/summaries/max
0.0415
0.0405
0.0395
0.0385
0.000 20.00 40.00 60.00 80.00 100.0


conv1/w_1/summaries/mean
2.500e-3
2.000e-3
1.500e-3
1.000e-3
5.000e-4
0.00
0.000 20.00 40.00 60.00 80.00 100.0


conv1/w_1/summaries/min
-0.0392
-0.0396
-0.0400
-0.0404
-0.0408
-0.0412
0.000 20.00 40.00 60.00 80.00 100.0


conv1/w_1/summaries/stddev_1
0.0177
0.0175
0.0173
0.0171
0.000 20.00 40.00 60.00 80.00 100.0

Tensorboard使用-模型状态

```
23 def feature_map_summary(conv, dim, name):
24     conv_channels = tf.split(conv,
25         num_or_size_splits=dim, axis=3)
26     with tf.name_scope(name):
27         len_channel = len(conv_channels)
28         for i in range(len_channel):
29             tf.summary.image('channel-%d' % (i),
30                 conv_channels[i], max_outputs=1)
```

Tensorboard使用-模型状态

```
84 conv1 = lrelu(conv2d(x_image, 32, name="conv1"), name='lrelu1')
85 feature_map_summary(conv1, 32, "conv1_feature_map")
86 pool1 = max_pool_2d(conv1, name="pool1")
87 conv2 = lrelu(conv2d(pool1, 64, name="conv2"), name='lrelu2')
88 feature_map_summary(conv2, 64, "conv2_feature_map")
89 pool2 = max_pool_2d(conv2, name="pool2")
90 with tf.name_scope("flatten"):
91     pool2_flatten = tf.reshape(pool2, [-1, 7*7*64])
92 fc1 = lrelu(dense(pool2_flatten, 1024, name="fc1"), name='lrelu3')
93 dropout_fc1 = tf.nn.dropout(fc1, 0.5)
94 logits = dense(dropout_fc1, 10, name="fc2")
```

conv1_feature_map/channel-6/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

conv1_feature_map/channel-7/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

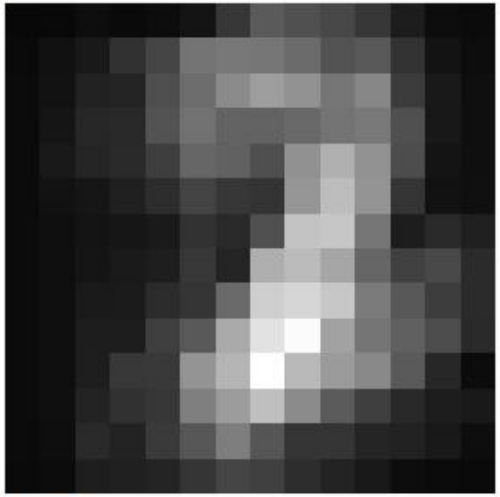
conv1_feature_map/channel-8/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

conv1_feature_map/channel-9/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

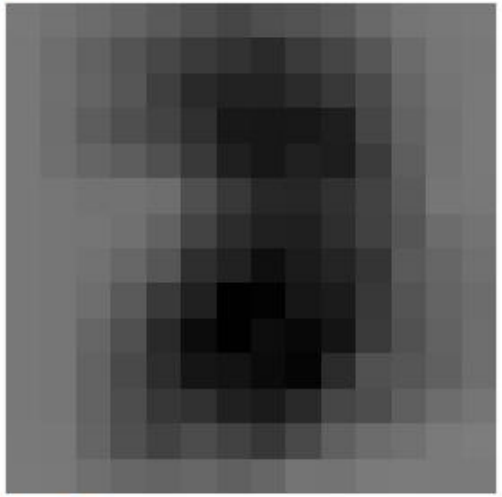
conv1_feature_map/channel-10/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

conv1_feature_map/channel-11/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

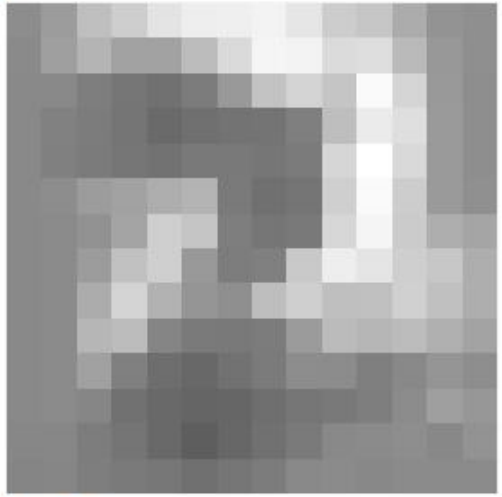
conv2_feature_map/channel-39/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))



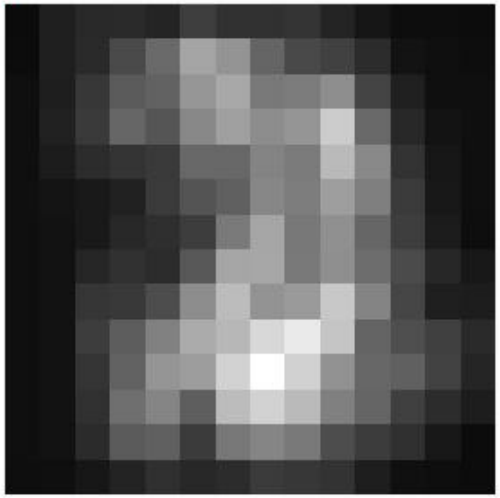
conv2_feature_map/channel-40/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))



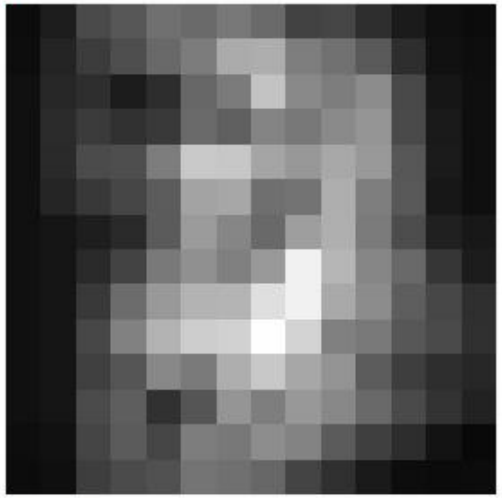
conv2_feature_map/channel-41/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))



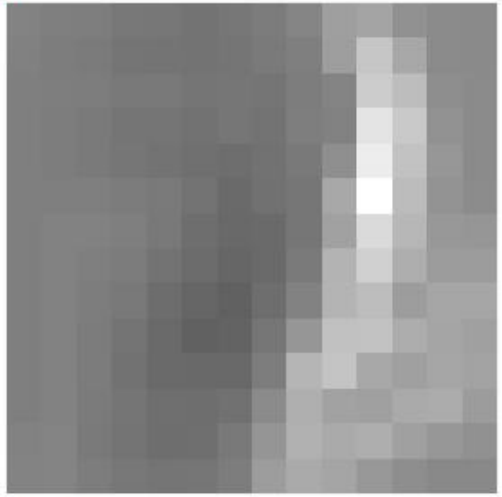
conv2_feature_map/channel-42/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))



conv2_feature_map/channel-43/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))



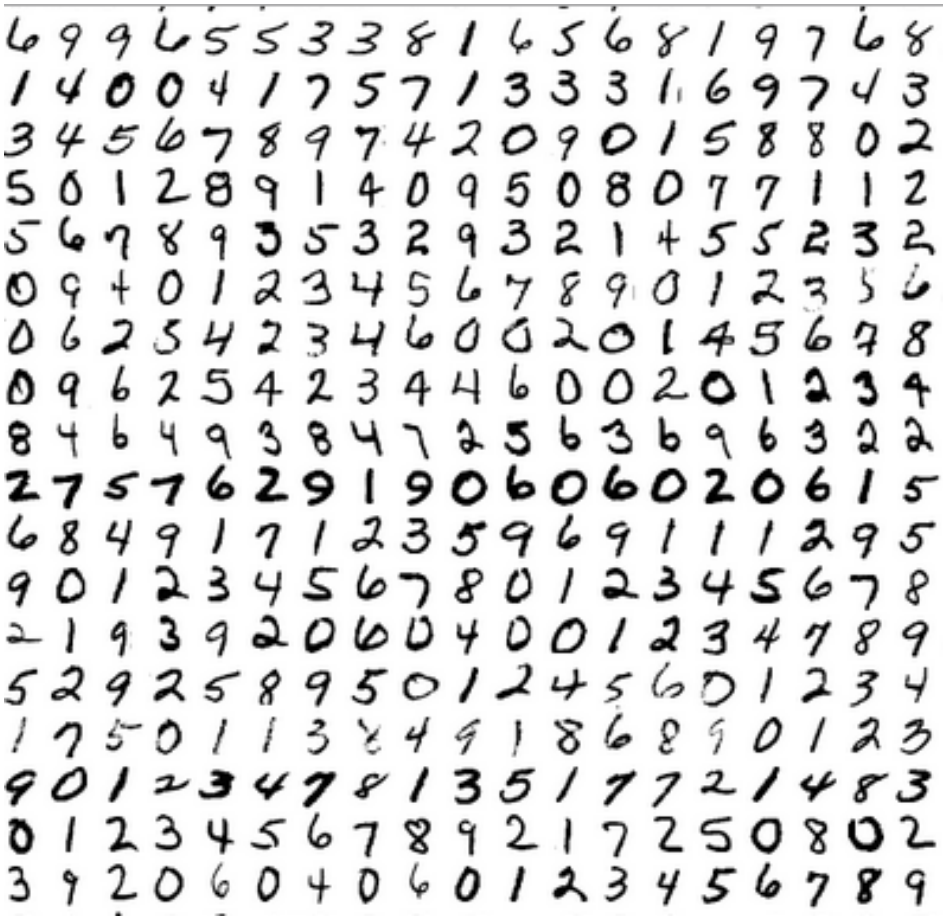
conv2_feature_map/channel-44/image
train
step **22** (Sat Aug 12 2017 11:26:37 GMT+0800 (CST))

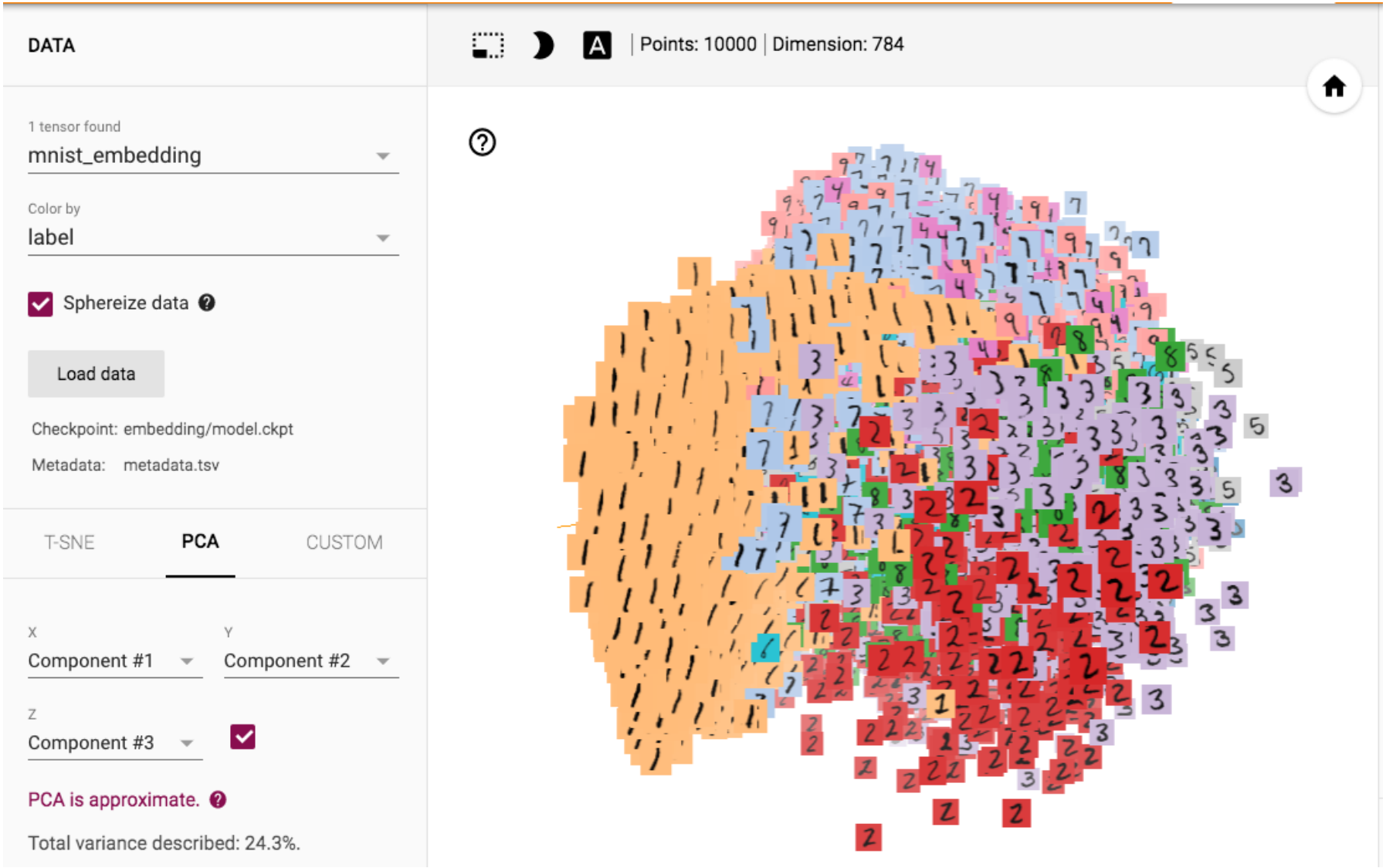


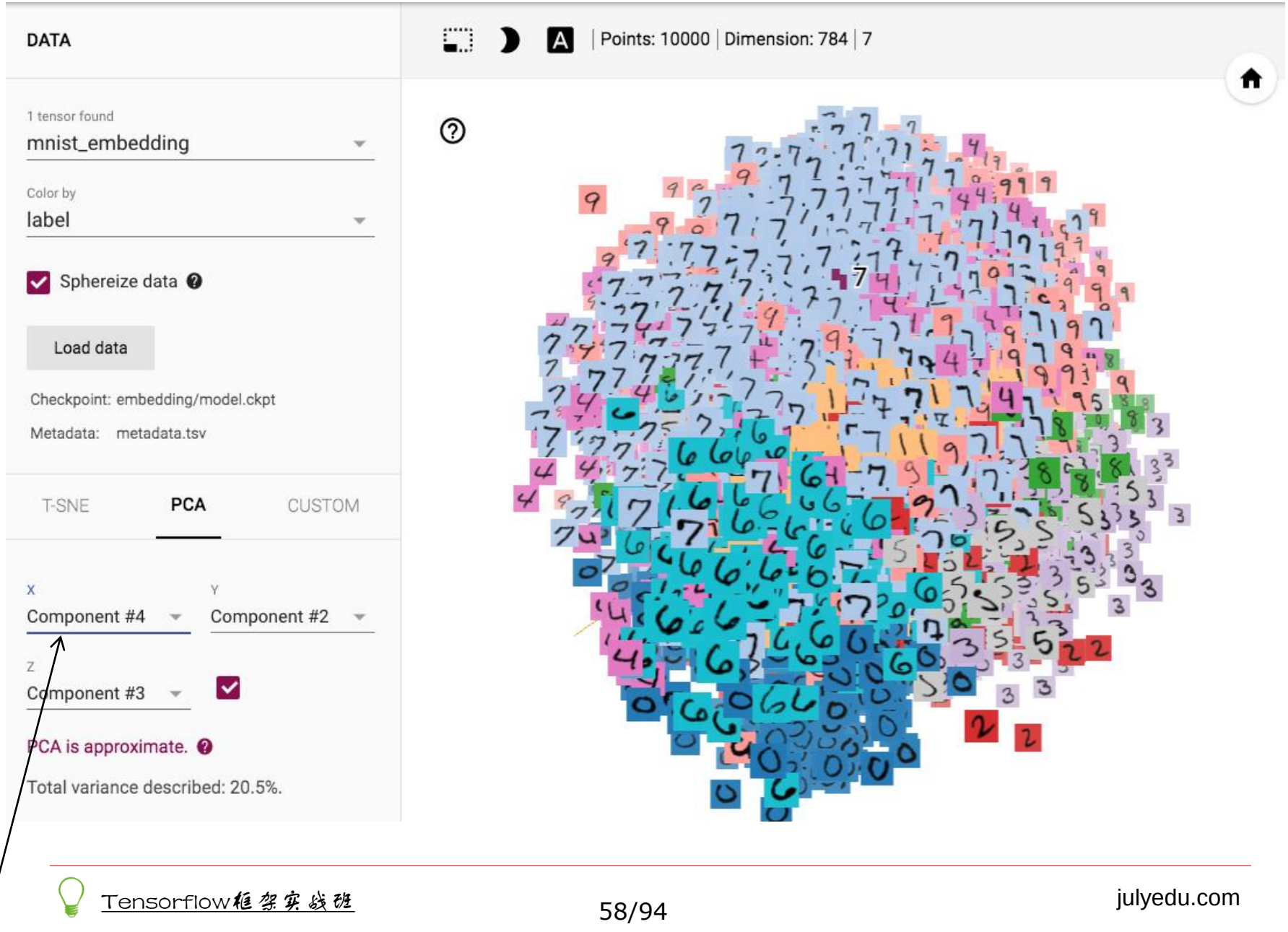
Tensorboard使用-分析数据

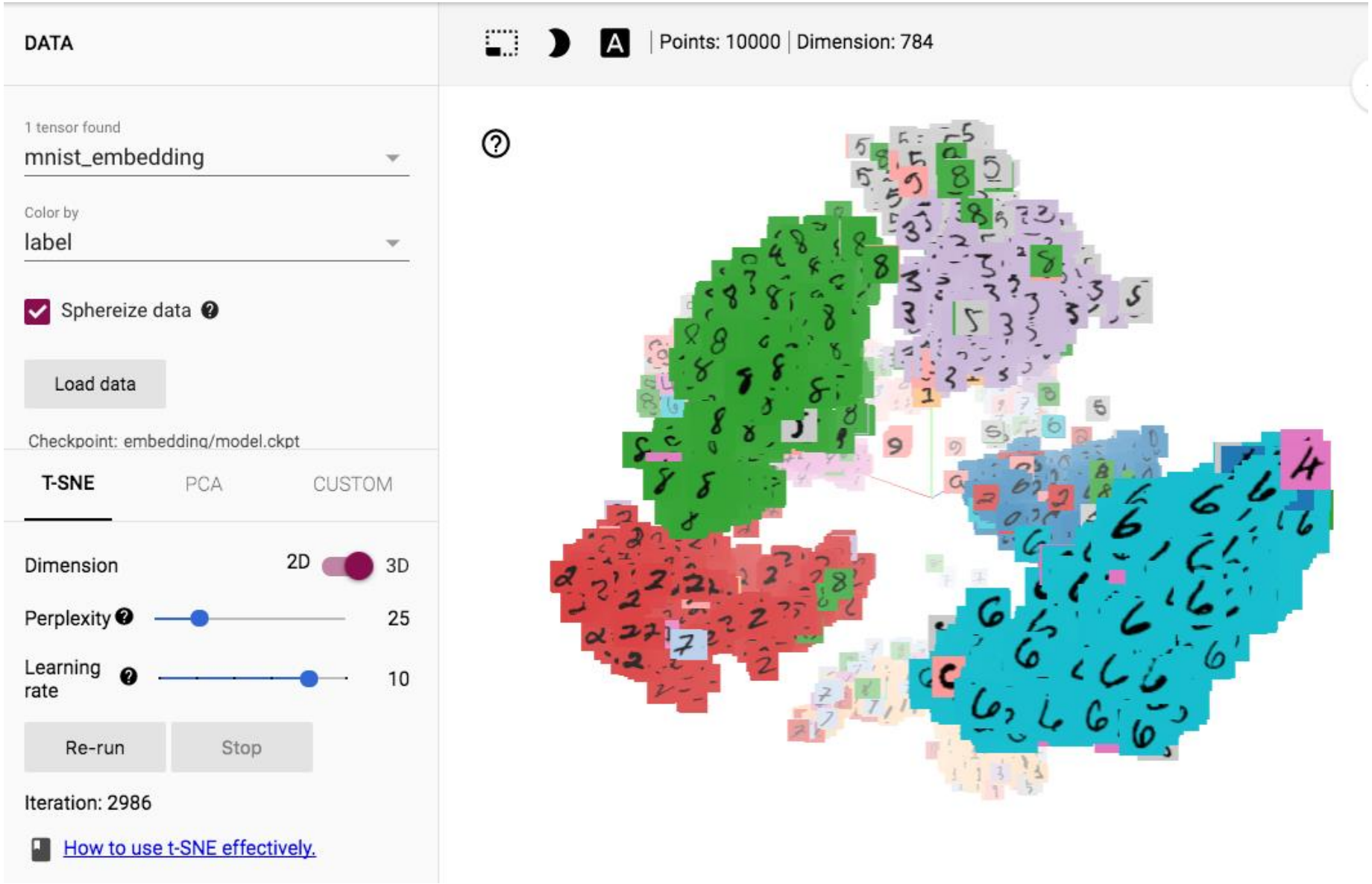
```
21 embedding_var = tf.Variable(mnist.test.images, name="mnist_embedding")
22 saver = tf.train.Saver()
23 sess = tf.Session()
24
25 sess.run(tf.global_variables_initializer())
26 saver.save(sess, os.path.join(path, 'model.ckpt'))
27
28 config = projector.ProjectorConfig()
29
30 embedding = config.embeddings.add()
31 embedding.tensor_name = embedding_var.name
32 embedding.metadata_path = os.path.join(path, 'metadata.tsv')
33 embedding.sprite.image_path = '/Users/zhangyx/workspace/mnist/MNIST_data/mnist_10k_sp
34 embedding.sprite.single_image_dim.extend([28, 28])
35
36 summary_writer = tf.summary.FileWriter(path)
37 projector.visualize_embeddings(summary_writer, config)
```

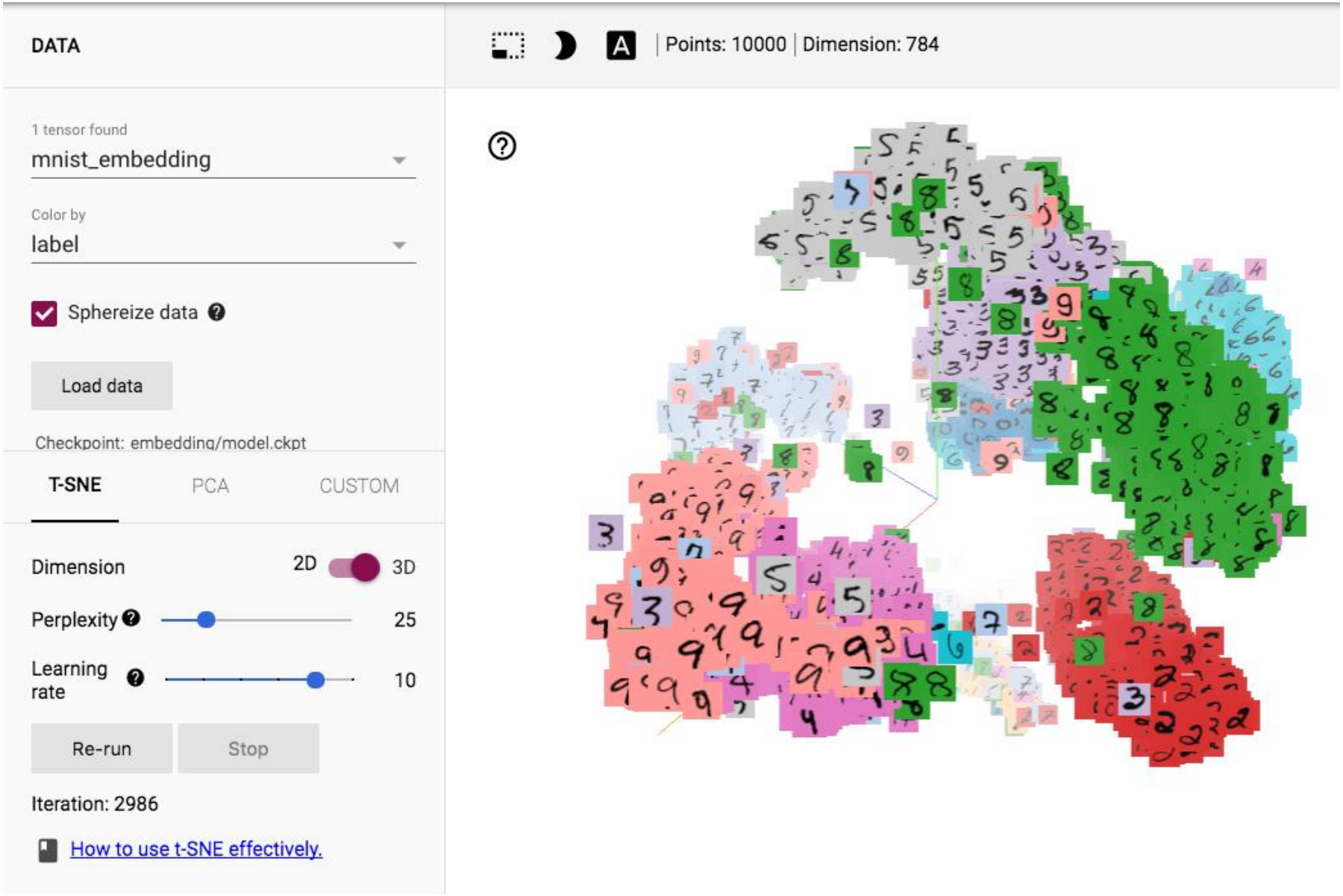
Tensorboard使用-分析数据

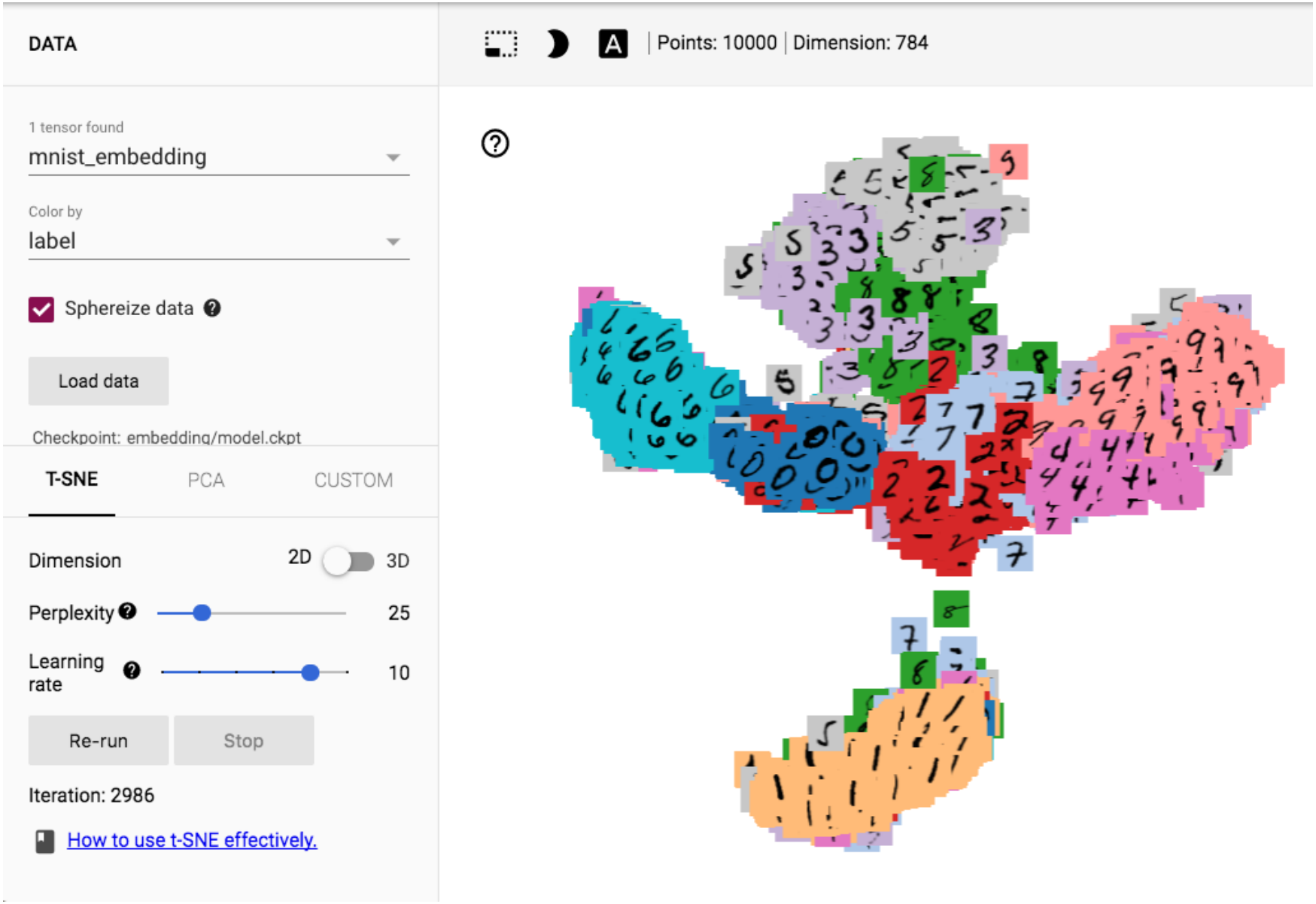



DATA
Points: 10000 | Dimension: 784
1 tensor found
mnist_embedding
Color by
label
Sphereize data
Load data
Checkpoint: embedding/model.ckpt
Metadata: metadata.tsv
T-SNE
PCA
CUSTOM
X
Component #1
Y
Component #2
Z
Component #3
PCA is approximate.
Total variance described: 24.3%.


DATA
Points: 10000 | Dimension: 784 | 7
1 tensor found
mnist_embedding
Color by
label
Sphereize data
Load data
Checkpoint: embedding/model.ckpt
Metadata: metadata.tsv
T-SNE
PCA
CUSTOM
X
Component #4
Y
Component #2
Z
Component #3
PCA is approximate.
Total variance described: 20.5%.


DATA
Points: 10000 | Dimension: 784
1 tensor found
mnist_embedding
Color by
label
Sphereize data
Load data
Checkpoint: embedding/model.ckpt
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Perplexity
25
Learning rate
10
Re-run
Stop
Iteration: 2986
How to use t-SNE effectively.


DATA
Points: 10000 | Dimension: 784
1 tensor found
mnist_embedding
Color by
label
Sphereize data
Load data
Checkpoint: embedding/model.ckpt
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Perplexity
25
Learning rate
10
Re-run
Stop
Iteration: 2986
How to use t-SNE effectively.


DATA
Points: 10000 | Dimension: 784
1 tensor found
mnist_embedding
Color by
label
Sphereize data
Load data
Checkpoint: embedding/model.ckpt
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Perplexity
25
Learning rate
10
Re-run
Stop
Iteration: 2986
How to use t-SNE effectively.

Break!

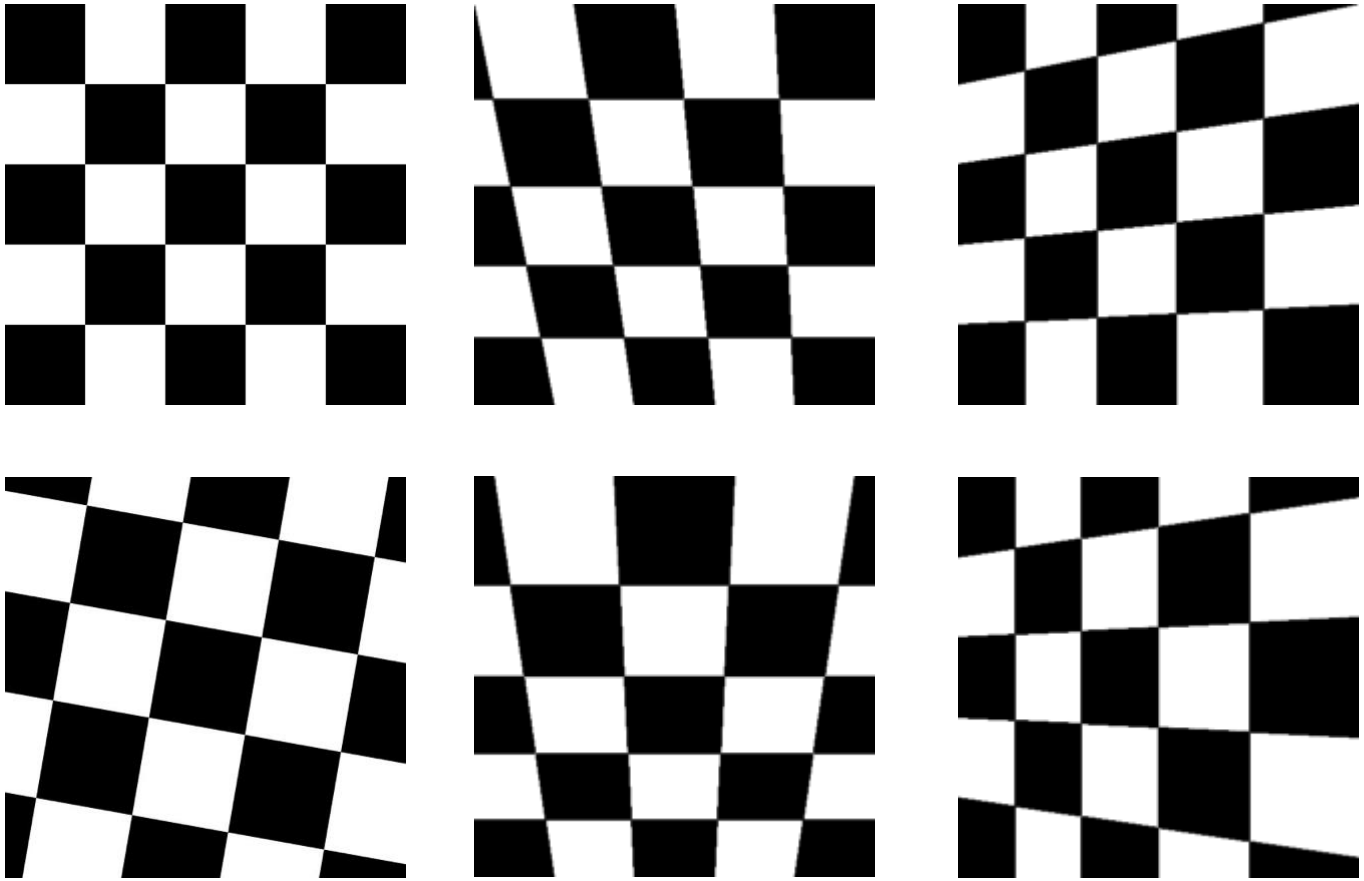
Tensorboard使用-调优手段

- 收集高质量数据
- Data Augmentation
  - Distortion
  - Crop
  - Reflection
  - ……
  - https://github.com/mdbloice/Augmentor
- 数据归一化

Tensorboard使用-调优手段

- tf.image.flip_left_right
- tf.image.flip_up_down
- tf.image.random_brightness
- tf.image.random_contrast
- tf.image.random_flip_left_right
- tf.image.random_flip_up_down
- tf.image.random_hue
- tf.image.random_saturation

Tensorboard使用-调优手段



Tensorboard使用-调优手段

- 参数初始化
  - 均匀分布初始化[-scale, scale]
    - Xavier: Scale = np.sqrt(3/n)
    - He: Sclae = np.sqrt(6/n)
  - 高斯分布初始化
    - Xavier: stddev = np.sqrt(1/n)
    - He: stddev = np.sqrt(2/n)
  - N
    - Xavier: n=(fan_in+fan_out)/2
    - He: n=(fan_in)

Tensorboard使用-调优手段

- 参数初始化
  - tf.contrib.keras.initializers.glorot_normal
  - tf.contrib.keras.initializers.glorot_uniform
  - tf.contrib.keras.initializers.he_normal
  - tf.contrib.keras.initializers.he_uniform
  - tf.contrib.keras.initializers.lecun_uniform

Tensorboard使用-调优手段

- 优化方法
  - Adam
  - Adadelta
  - Sgd
  - Sgd+momentum

Tensorboard使用-调优手段

- 归一化
  - 数据归一化
  - 梯度归一化
    - 梯度/mini_size
  - Batch Normalization

Tensorboard使用-调优手段

- Learning rate
  - 太大
    - Loss爆炸或者nan
  - 太小
    - 没有反应
  - 初始设置小的，后期慢慢调大
    - 1，0.1， 0.01， 0.001
    - 梯度截断

Tensorboard使用-调优手段

- Batch size
  - =1
  - 大
    - 吞吐量大
    - 内存计算资源占用多

Tensorboard使用-调优手段

- Ensemble
  - 同样模型结构，不同初始化方法
  - 不同的模型结构，选取最好的几个model
  - 同样的模型，不同的训练阶段

Tensorboard使用-调优手段

- 模型结构
  - 模型深度
  - 每一层的filter数目
  - Dropout
  - 激活函数

Tensorboard使用-调优手段

- Eearly stopping
  - Val accuracy一下降就停
  - 连续多次val accuracy下降
  - 记录最优值，连续多次val accuracy没达到最优值
  - 每一段时间对val accuracy计算平均值，平均值下降即停

Tensorboard使用-调优手段

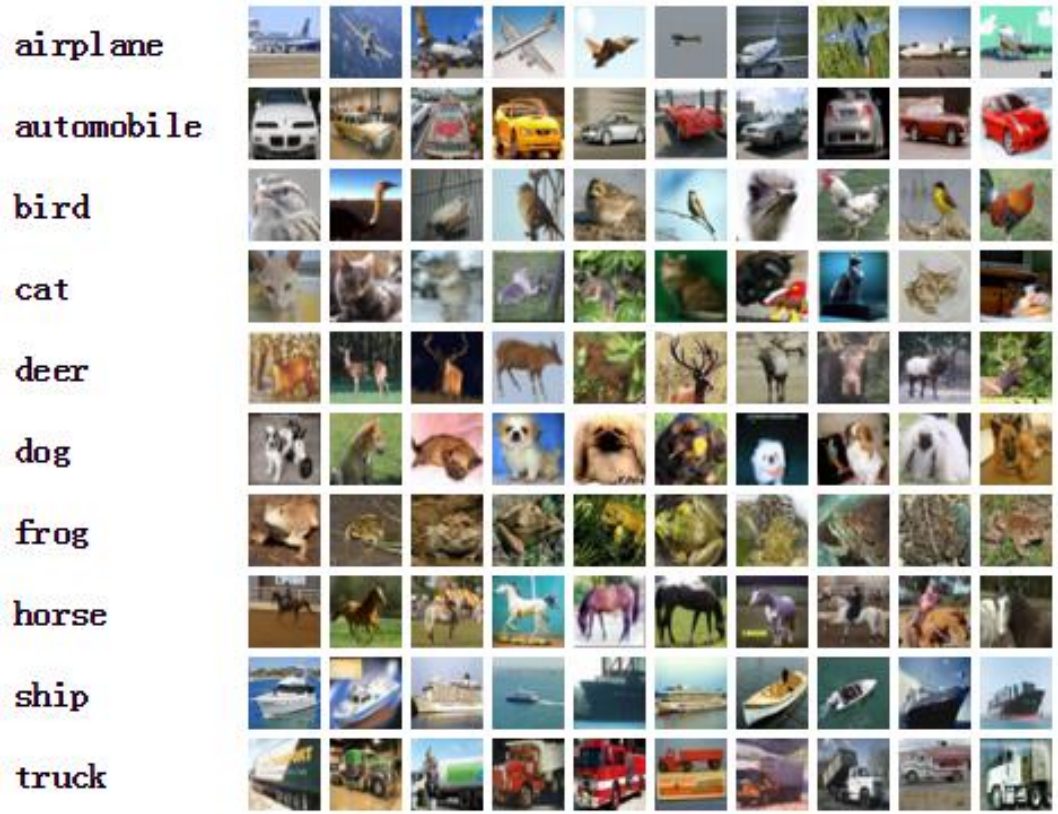
- 欠拟合
  - 增大网络复杂度
  - 降低learning rate
  - Batch normalization

Tensorboard使用-调优手段

- 过拟合
  - 丰富数据
  - 降低网络复杂度
  - 正则化
    - L1
    - L2
  - Dropout
  - Eearly stopping
  - 降低learning rate

Tensorboard使用-Cifar

- Cifar数据集
  - 32 * 32
  - Trainset
    - 50000
  - Testset
    - 10000



who is the best in CIFAR-10 ?

CIFAR-10 49 results collected

Units: accuracy %

> Classify 32x32 colour images.

Result	Method	Venue	Details
96.53%	Fractional Max-Pooling	arXiv 2015	Details
95.59%	Striving for Simplicity: The All Convolutional Net	ICLR 2015	Details
94.16%	All you need is a good init	ICLR 2016	Details
94%	Lessons learned from manually classifying CIFAR-10	unpublished 2011	Details
93.95%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
93.72%	Spatially-sparse convolutional neural networks	arXiv 2014	
93.63%	Scalable Bayesian Optimization Using Deep Neural Networks	ICML 2015	

Tensorboard使用-Cifar

```
170    # Randomly crop a [height, width] section of the image.
171    distorted_image = tf.random_crop(reshaped_image, [height, width, 3])
172
173    # Randomly flip the image horizontally.
174    distorted_image = tf.image.random_flip_left_right(distorted_image)
175
176    # Because these operations are not commutative, consider randomizing
177    # the order their operation.
178    # NOTE: since per_image_standardization zeros the mean and makes
179    # the stddev unit, this likely has no effect see tensorflow#1458.
180    distorted_image = tf.image.random_brightness(distorted_image,
181                                                 max_delta=63)
182    distorted_image = tf.image.random_contrast(distorted_image,
183                                               lower=0.2, upper=1.8)
184
185    # Subtract off the mean and divide by the variance of the pixels.
186    float_image = tf.image.per_image_standardization(distorted_image)
```

Tensorboard使用-Cifar

```
115 def _variable_with_weight_decay(name, shape, stddev, wd):
116   dtype = tf.float16 if FLAGS.use_fp16 else tf.float32
117   var = _variable_on_cpu(
118       name,
119       shape,
120       tf.truncated_normal_initializer(stddev=stddev, dtype=dtype))
121   if wd is not None:
122     weight_decay = tf.multiply(tf.nn.l2_loss(var), wd, name='weight_loss')
123     tf.add_to_collection('losses', weight_decay)
124   return var
```

```
268   labels = tf.cast(labels, tf.int64)
269   cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(
270       labels=labels, logits=logits, name='cross_entropy_per_example')
271   cross_entropy_mean = tf.reduce_mean(cross_entropy, name='cross_entropy')
272   tf.add_to_collection('losses', cross_entropy_mean)
273
274   # The total loss is defined as the cross entropy loss plus all of the weight
275   # decay terms (L2 loss).
276   return tf.add_n(tf.get_collection('losses'), name='total_loss')
```

Tensorboard使用-Cifar

```
188   with tf.variable_scope('conv1') as scope:
189     kernel = _variable_with_weight_decay('weights',
190                                          shape=[5, 5, 3, 64],
191                                          stddev=5e-2,
192                                          wd=0.5)
193     conv = tf.nn.conv2d(images, kernel, [1, 1, 1, 1], padding='SAME')
194     biases = _variable_on_cpu('biases', [64], tf.constant_initializer(0.0))
195     pre_activation = tf.nn.bias_add(conv, biases)
196     conv1 = tf.nn.relu(pre_activation, name=scope.name)
197     _activation_summary(conv1)
198
199   # pool1
200   pool1 = tf.nn.max_pool(conv1, ksize=[1, 3, 3, 1], strides=[1, 2, 2, 1],
201                          padding='SAME', name='pool1')
202   # norm1
203   norm1 = tf.nn.lrn(pool1, 4, bias=1.0, alpha=0.001 / 9.0, beta=0.75,
204                     name='norm1')
```

Tensorboard使用-Cifar

```
206   # conv2
207   with tf.variable_scope('conv2') as scope:
208     kernel = _variable_with_weight_decay('weights',
209                                          shape=[5, 5, 64, 64],
210                                          stddev=5e-2,
211                                          wd=0.5)
212     conv = tf.nn.conv2d(norm1, kernel, [1, 1, 1, 1], padding='SAME')
213     biases = _variable_on_cpu('biases', [64], tf.constant_initializer(0.1))
214     pre_activation = tf.nn.bias_add(conv, biases)
215     conv2 = tf.nn.relu(pre_activation, name=scope.name)
216     _activation_summary(conv2)
217
218   # norm2
219   norm2 = tf.nn.lrn(conv2, 4, bias=1.0, alpha=0.001 / 9.0, beta=0.75,
220                     name='norm2')
221   # pool2
222   pool2 = tf.nn.max_pool(norm2, ksize=[1, 3, 3, 1],
223                          strides=[1, 2, 2, 1], padding='SAME', name='pool2')
```

```
225   # local3
226   with tf.variable_scope('local3') as scope:
227     # Move everything into depth so we can perform a single matrix multiply.
228     reshape = tf.reshape(pool2, [FLAGS.batch_size, -1])
229     dim = reshape.get_shape()[1].value
230     weights = _variable_with_weight_decay('weights', shape=[dim, 384],
231                                           stddev=0.04, wd=0.004)
232     biases = _variable_on_cpu('biases', [384], tf.constant_initializer(0.1))
233     local3 = tf.nn.relu(tf.matmul(reshape, weights) + biases, name=scope.name)
234     _activation_summary(local3)
235
236   # local4
237   with tf.variable_scope('local4') as scope:
238     weights = _variable_with_weight_decay('weights', shape=[384, 192],
239                                           stddev=0.04, wd=0.004)
240     biases = _variable_on_cpu('biases', [192], tf.constant_initializer(0.1))
241     local4 = tf.nn.relu(tf.matmul(local3, weights) + biases, name=scope.name)
242     _activation_summary(local4)
243
244   with tf.variable_scope('softmax_linear') as scope:
245     weights = _variable_with_weight_decay('weights', [192, NUM_CLASSES],
246                                           stddev=1/192.0, wd=0.0)
247     biases = _variable_on_cpu('biases', [NUM_CLASSES],
248                               tf.constant_initializer(0.0))
249     softmax_linear = tf.add(tf.matmul(local4, weights), biases, name=scope.name)
250     _activation_summary(softmax_linear)
```

Tensorboard使用-Cifar

```
324   num_batches_per_epoch = NUM_EXAMPLES_PER_EPOCH_FOR_TRAIN / FLAGS.batch_size
325   decay_steps = int(num_batches_per_epoch * NUM_EPOCHS_PER_DECAY)
326
327   # Decay the learning rate exponentially based on the number of steps.
328   lr = tf.train.exponential_decay(INITIAL_LEARNING_RATE,
329                                   global_step,
330                                   decay_steps,
331                                   LEARNING_RATE_DECAY_FACTOR,
332                                   staircase=True)
333   tf.summary.scalar('learning_rate', lr)
334
335   # Generate moving averages of all losses and associated summaries.
336   loss_averages_op = _add_loss_summaries(total_loss)
337
338   # Compute gradients.
339   with tf.control_dependencies([loss_averages_op]):
340     opt = tf.train.GradientDescentOptimizer(lr)
341     grads = opt.compute_gradients(total_loss)
```

Tensorboard使用-Cifar

```
343   # Apply gradients.
344   apply_gradient_op = opt.apply_gradients(grads, global_step=global_step)
345
346   # Add histograms for trainable variables.
347   for var in tf.trainable_variables():
348     tf.summary.histogram(var.op.name, var)
349
350   # Add histograms for gradients.
351   for grad, var in grads:
352     if grad is not None:
353       tf.summary.histogram(var.op.name + '/gradients', grad)
354
355   # Track the moving averages of all trainable variables.
356   variable_averages = tf.train.ExponentialMovingAverage(
357       MOVING_AVERAGE_DECAY, global_step)
358   variables_averages_op = variable_averages.apply(tf.trainable_variables())
359
360   with tf.control_dependencies([apply_gradient_op, variables_averages_op]):
361     train_op = tf.no_op(name='train')
```

Tensorboard使用-Cifar

- 86%
  - 100k steps
  - Batch_size=128

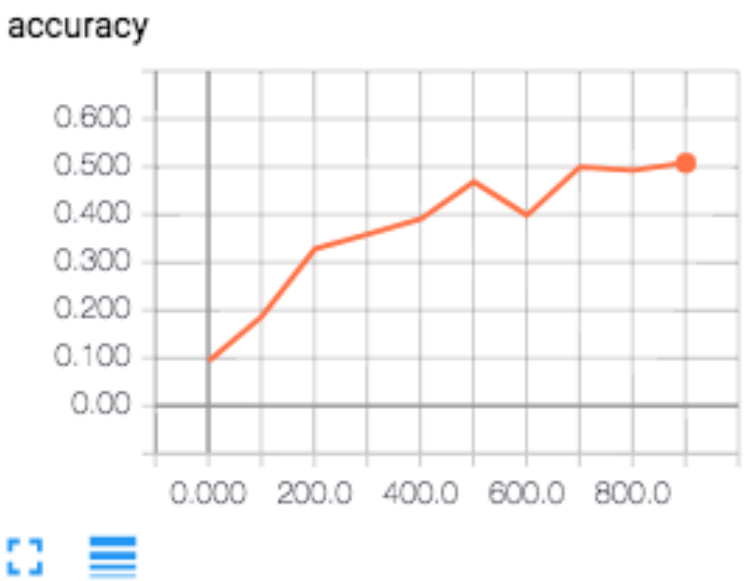
System	Step Time (sec/batch)	Accuracy
1 Tesla K20m	0.35-0.60	~86% at 60K steps (5 hours)
1 Tesla K40m	0.25-0.35	~86% at 100K steps (4 hours)

Tensorboard使用-Cifar

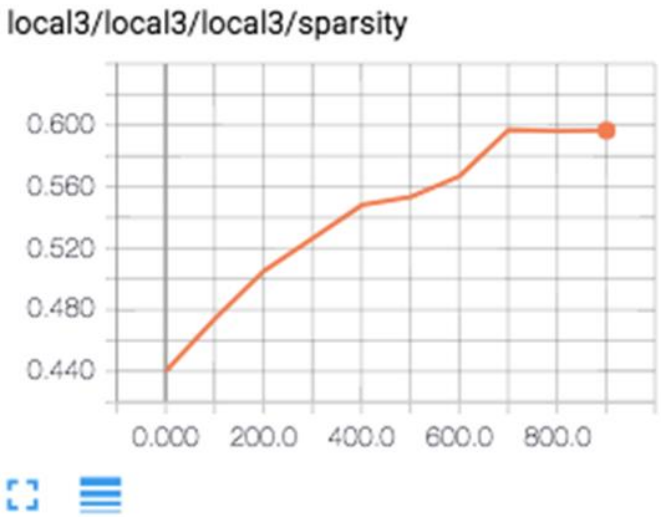
```
79 def _activation_summary(x):
80   tensor_name = re.sub('%s_[0-9]*/' % TOWER_NAME, '', x.op.name)
81   tf.summary.histogram(tensor_name + '/activations', x)
82   tf.summary.scalar(tensor_name + '/sparsity',
83                                        tf.nn.zero_fraction(x))
```

Tensorboard使用-Cifar

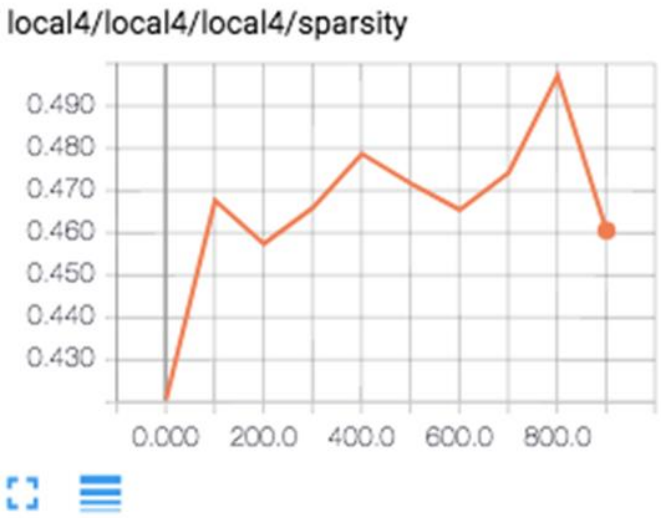
accuracy



local3

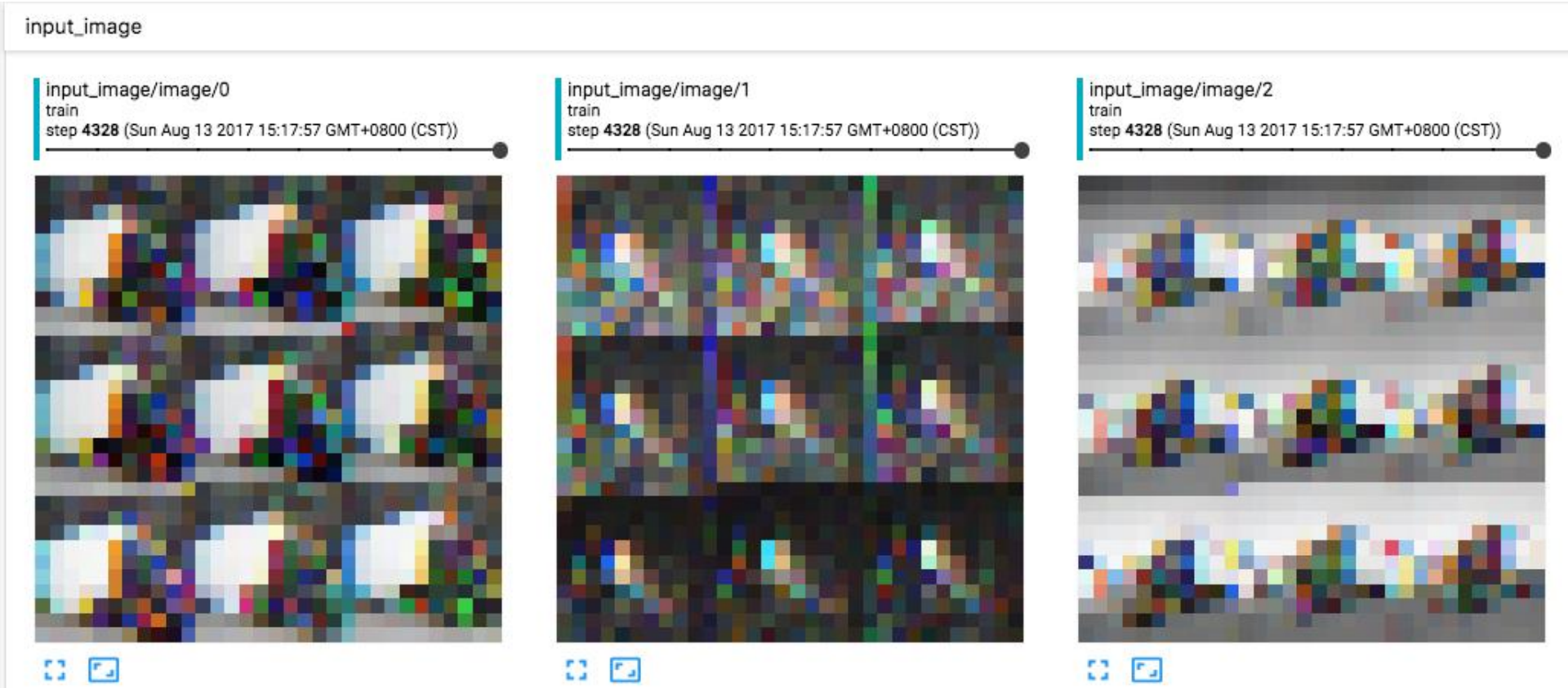


local4

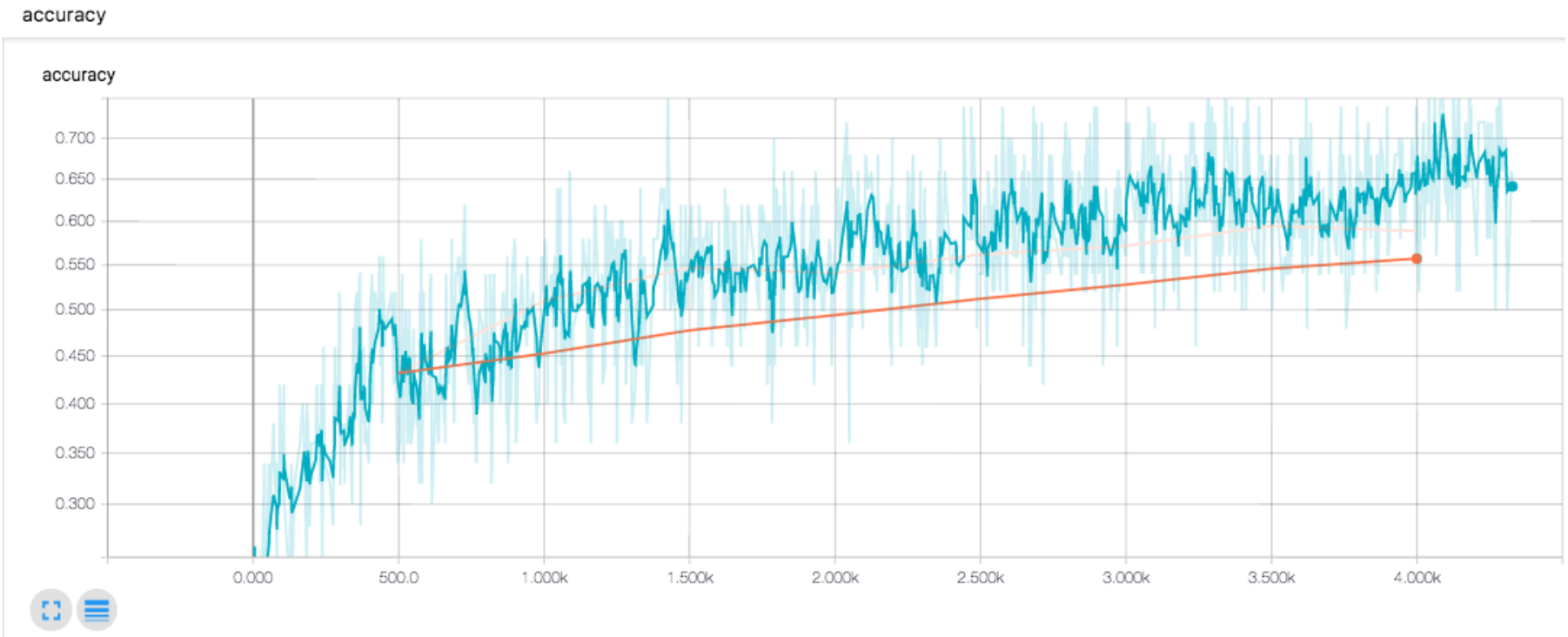


Tensorboard使用-Cifar快速实验

- 错误的转换图片

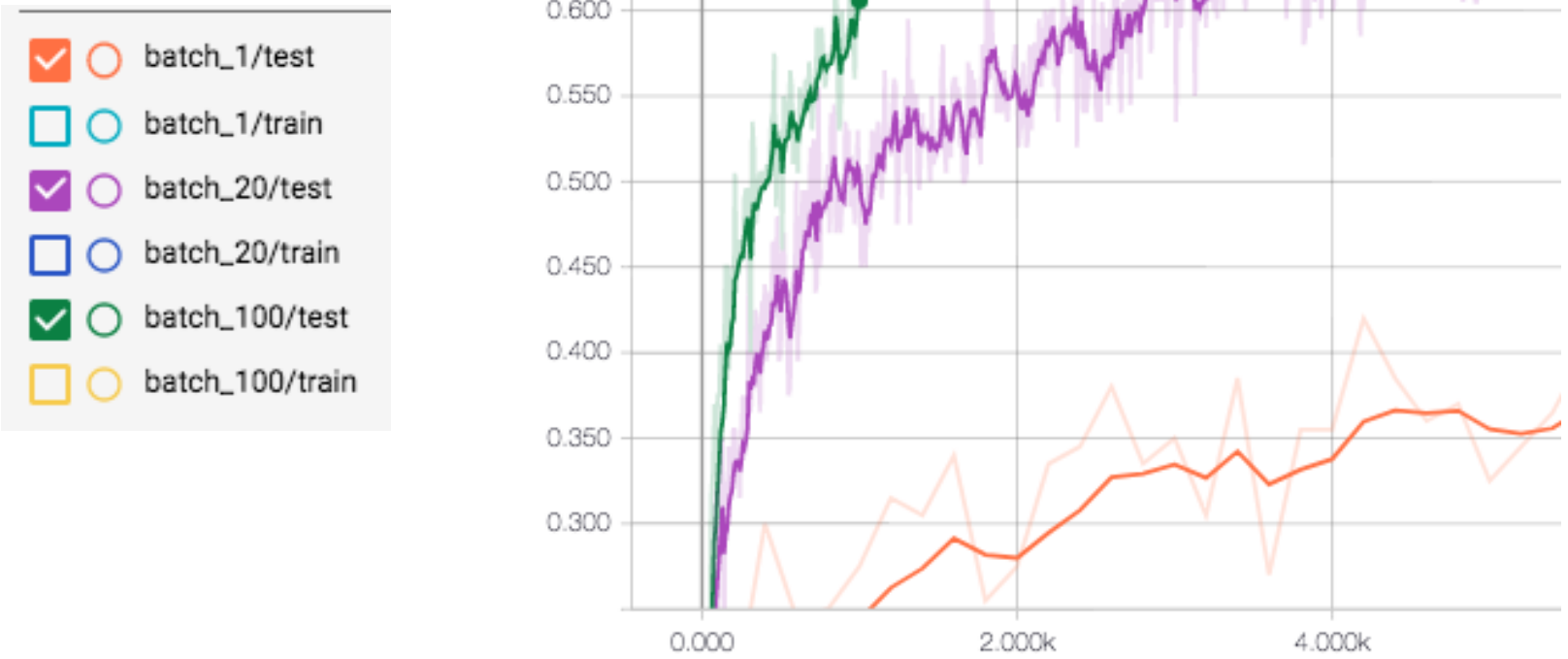


Tensorboard使用-Cifar快速实验



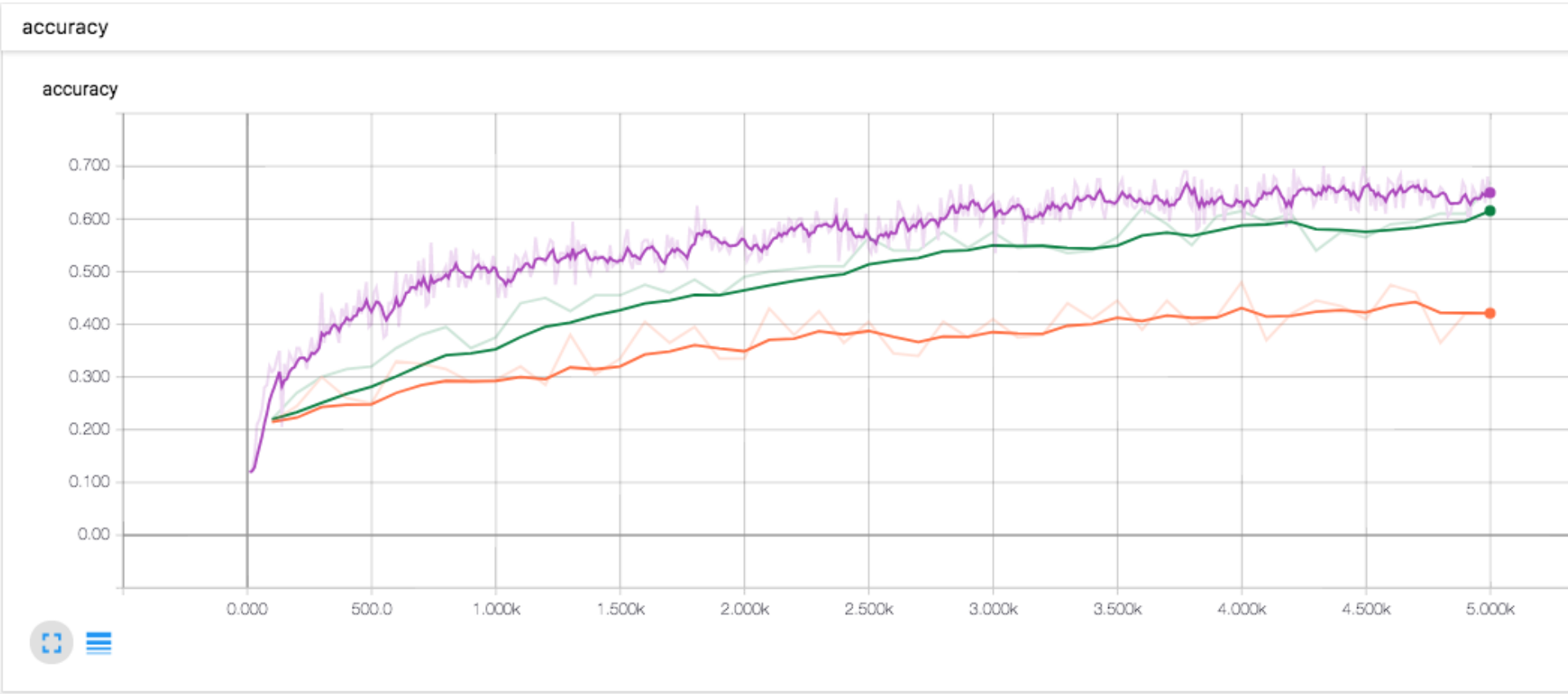
Tensorboard使用-Cifar快速实验

- Batch_size



Tensorboard使用-Cifar

- Initialize



Summary

- Tensorboard介绍
- 调优手段
- Cifar10实例

谢谢！