```python
import numpy as np
import pandas as pd
import tensorflow as tf
import datetime
```

Loading and running our previous MNIST handwritten digits dataset

- Using 20 Epochs

```python
from keras.datasets import mnist
from keras.utils import to_categorical

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

from keras import layers
from keras import models
from keras.layers import Dropout

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(Dropout(0.5))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(16, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(16, activation='relu'))
model.add(Dropout(0.5))
model.add(layers.Dense(10, activation='sigmoid'))

model.summary()

model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.fit(train_images, train_labels,
          epochs=20,
          batch_size=64,
          validation_split = 0.2)
```

```
dropout (Dropout)            (None, 26, 26, 32)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 16)        4624
_________________________________________________________________
flatten (Flatten)            (None, 1936)              0
_________________________________________________________________
dense (Dense)                (None, 16)                30992
_________________________________________________________________
dropout_1 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                170
=================================================================
Total params: 36,106
Trainable params: 36,106
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
750/750 [==============================] - 44s 57ms/step - loss: 1.3461 - accura
Epoch 2/20
750/750 [==============================] - 42s 57ms/step - loss: 0.5766 - accura
Epoch 3/20
750/750 [==============================] - 43s 57ms/step - loss: 0.4781 - accura
Epoch 4/20
750/750 [==============================] - 43s 57ms/step - loss: 0.4366 - accura
Epoch 5/20
750/750 [==============================] - 43s 58ms/step - loss: 0.4164 - accura
Epoch 6/20
750/750 [==============================] - 44s 59ms/step - loss: 0.4172 - accura
Epoch 7/20
750/750 [==============================] - 44s 59ms/step - loss: 0.4150 - accura
Epoch 8/20
750/750 [==============================] - 44s 59ms/step - loss: 0.4064 - accura
Epoch 9/20
750/750 [==============================] - 45s 60ms/step - loss: 0.4108 - accura
Epoch 10/20
750/750 [==============================] - 45s 60ms/step - loss: 0.4087 - accura
Epoch 11/20
750/750 [==============================] - 44s 59ms/step - loss: 0.4174 - accura
Epoch 12/20
750/750 [==============================] - 45s 60ms/step - loss: 0.4185 - accura
Epoch 13/20
750/750 [==============================] - 46s 61ms/step - loss: 0.4139 - accura
Epoch 14/20
750/750 [==============================] - 42s 57ms/step - loss: 0.4172 - accura
Epoch 15/20
750/750 [==============================] - 43s 58ms/step - loss: 0.4173 - accura
Epoch 16/20

750/750 [==============================] - 43s 58ms/step - loss: 0.4192 - accura
Epoch 17/20
750/750 [==============================] - 44s 59ms/step - loss: 0.4132 - accura
Epoch 18/20
750/750 [==============================] - 46s 61ms/step - loss: 0.4186 - accura
Epoch 19/20
750/750 [==============================] - 43s 57ms/step - loss: 0.4178 - accura
Epoch 20/20
```

```
750/750 [==============================] - 44s 59ms/step - loss: 0.4227 - accura
<tensorflow.python.keras.callbacks.History at 0x7f15a4f7f990>
```

Evaluating the model on the test set

```
model.save_weights('model.h1')
model.load_weights('model.h1')
model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 2s 8ms/step - loss: 0.2068 - acc: 0.6
[0.2068410962820053, 0.6001999974250793]
```

Automatically reducing overall runtime by usning earlystopping

- we can use callback_early_stopping to interrupt training once a target metric being monitored has stopped improving for a fixed number of epochs.
- this callback allows us to interrupt training as soon as we start overfitting, thus avoiding having to retrain our model for a smaller number of epochs.

MODEL CHECKPOINT AND EARLY STOPPING CALLBACKS

```
import keras

callbacks_list = [
    keras.callbacks.EarlyStopping(
        monitor='acc',
        patience=1,
    ),
    keras.callbacks.ModelCheckpoint(
        filepath='my_model.h5',
        monitor='val_loss',
        save_best_only=True,
    )
]

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['acc'])

model.fit(train_images, train_labels,
          epochs=20,
          batch_size=64,
          callbacks=callbacks_list,
          validation_data=(test_images, test_labels))
```

```
Epoch 1/20
938/938 [==============================] - 57s 60ms/step - loss: 0.0984 - acc: 0
Epoch 2/20
```

```
938/938 [==============================] - 57s 61ms/step - loss: 0.0951 - acc: 0
<tensorflow.python.keras.callbacks.History at 0x7f159f3fc690>
```

REDUCING LEARNING RATE ON PLATEAU CALLBACK

```
callbacks_list = [
    keras.callbacks.ReduceLROnPlateau(
        monitor='val_loss',
        factor=0.1,
        patience=10,
    )
]

model.fit(train_images, train_labels,
          epochs=20,
          batch_size=32,
          callbacks=callbacks_list,
          validation_data=(test_images, test_labels))
```

```
Epoch 1/20
1875/1875 [==============================] - 64s 34ms/step - loss: 0.0974 - acc:
Epoch 2/20
1875/1875 [==============================] - 62s 33ms/step - loss: 0.0987 - acc:
Epoch 3/20
1875/1875 [==============================] - 63s 33ms/step - loss: 0.0994 - acc:
Epoch 4/20
1875/1875 [==============================] - 65s 35ms/step - loss: 0.0992 - acc:
Epoch 5/20
1875/1875 [==============================] - 65s 35ms/step - loss: 0.0994 - acc:
Epoch 6/20
1875/1875 [==============================] - 66s 35ms/step - loss: 0.0998 - acc:
Epoch 7/20
1875/1875 [==============================] - 68s 36ms/step - loss: 0.1001 - acc:
Epoch 8/20
1875/1875 [==============================] - 67s 36ms/step - loss: 0.0991 - acc:
Epoch 9/20
1875/1875 [==============================] - 65s 35ms/step - loss: 0.1000 - acc:
Epoch 10/20
1875/1875 [==============================] - 65s 34ms/step - loss: 0.1004 - acc:
Epoch 11/20
1875/1875 [==============================] - 64s 34ms/step - loss: 0.1000 - acc:
Epoch 12/20
1875/1875 [==============================] - 62s 33ms/step - loss: 0.0997 - acc:
Epoch 13/20
1875/1875 [==============================] - 62s 33ms/step - loss: 0.0999 - acc:
Epoch 14/20
1875/1875 [==============================] - 60s 32ms/step - loss: 0.0990 - acc:
Epoch 15/20
1875/1875 [==============================] - 61s 33ms/step - loss: 0.0994 - acc:
Epoch 16/20
1875/1875 [==============================] - 61s 33ms/step - loss: 0.0996 - acc:
Epoch 17/20
1875/1875 [==============================] - 62s 33ms/step - loss: 0.1004 - acc:
Epoch 18/20
```

```
1875/1875 [==============================] - 62s 33ms/step - loss: 0.0994 - acc:
Epoch 19/20
1875/1875 [==============================] - 62s 33ms/step - loss: 0.1002 - acc:
Epoch 20/20
1875/1875 [==============================] - 61s 33ms/step - loss: 0.1009 - acc:
<tensorflow.python.keras.callbacks.History at 0x7f15a26c9b90>
```

```
import keras
import numpy as np

class ActivationLogger(keras.callbacks.Callback):

    def set_model(self, model):
        self.model = model
        layer_outputs = [layer.output for layer in model.layers]
        self.activations_model = keras.models.Model(model.input,
                                                    layer_outputs)

    def on_epoch_end(self, epoch, logs=None):
        if self.validation_data is None:
            raise RuntimeError('Requires validation_data.')

        validation_sample = self.validation_data[0][0:1]
        activations = self.activations_model.predict(validation_sample)
        f = open('activations_at_epoch_' + str(epoch) + '.npz', 'w')
        np.savez(f, activations)
        f.close()
```

Training the model with a TensorBoard callback

```
from keras.callbacks import TensorBoard
tbCallBack = TensorBoard(log_dir='./log', histogram_freq=1,
                         write_graph=True,
                         write_grads=True,
                         batch_size=128,
                         write_images=True)

model.fit(train_images, train_labels,
          batch_size=128,
          epochs=20,
          verbose=1,
          validation_data=(test_images, test_labels),
          callbacks=[tbCallBack])
```

```
WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the `Tens
WARNING:tensorflow:`batch_size` is no longer needed in the `TensorBoard` Callbac
Epoch 1/20
469/469 [==============================] - 48s 102ms/step - loss: 0.0944 - acc:
Epoch 2/20
469/469 [==============================] - 48s 102ms/step - loss: 0.0941 - acc:
```

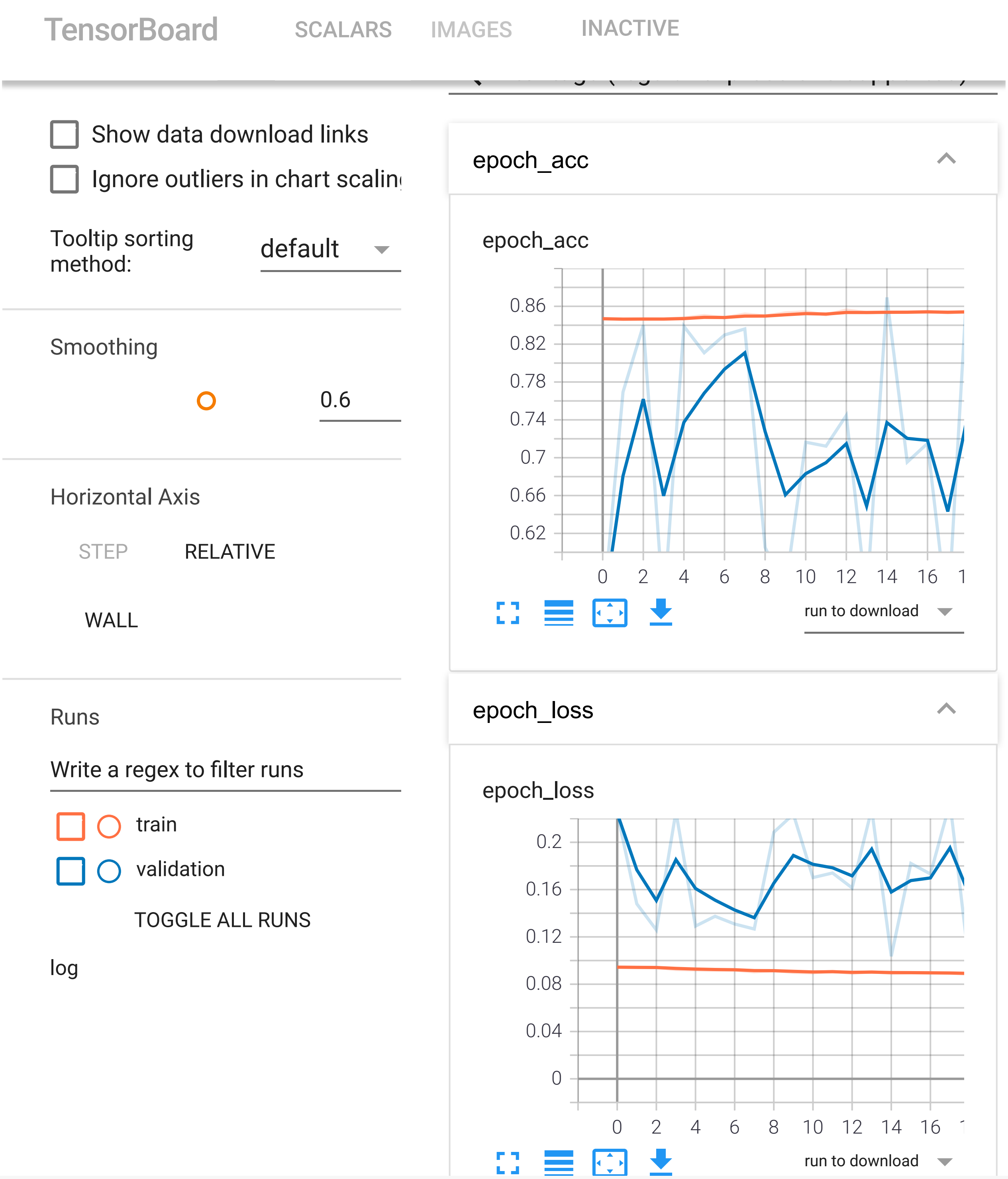
```
Epoch 3/20
469/469 [==============================] - 48s 102ms/step - loss: 0.0940 - acc:
Epoch 4/20
469/469 [==============================] - 47s 101ms/step - loss: 0.0925 - acc:
Epoch 5/20
469/469 [==============================] - 48s 102ms/step - loss: 0.0921 - acc:
Epoch 6/20
469/469 [==============================] - 48s 103ms/step - loss: 0.0919 - acc:
Epoch 7/20
469/469 [==============================] - 49s 104ms/step - loss: 0.0919 - acc:
Epoch 8/20
469/469 [==============================] - 48s 102ms/step - loss: 0.0905 - acc:
Epoch 9/20
469/469 [==============================] - 47s 100ms/step - loss: 0.0913 - acc:
Epoch 10/20
469/469 [==============================] - 47s 100ms/step - loss: 0.0900 - acc:
Epoch 11/20
469/469 [==============================] - 47s 100ms/step - loss: 0.0896 - acc:
Epoch 12/20
469/469 [==============================] - 47s 100ms/step - loss: 0.0909 - acc:
Epoch 13/20
469/469 [==============================] - 47s 99ms/step - loss: 0.0893 - acc: 0
Epoch 14/20
469/469 [==============================] - 48s 101ms/step - loss: 0.0905 - acc:
Epoch 15/20
469/469 [==============================] - 48s 102ms/step - loss: 0.0893 - acc:
Epoch 16/20
469/469 [==============================] - 47s 99ms/step - loss: 0.0897 - acc: 0
Epoch 17/20
469/469 [==============================] - 46s 99ms/step - loss: 0.0894 - acc: 0
Epoch 18/20
469/469 [==============================] - 46s 99ms/step - loss: 0.0893 - acc: 0
Epoch 19/20
469/469 [==============================] - 46s 99ms/step - loss: 0.0886 - acc: 0
Epoch 20/20
469/469 [==============================] - 47s 100ms/step - loss: 0.0901 - acc:
<tensorflow.python.keras.callbacks.History at 0x7f159fa4a890>
```

```
model.save_weights('model.h2')
model.load_weights('model.h2')
model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 3s 8ms/step - loss: 0.2068 - acc: 0.6
[0.2068410962820053, 0.6001999974250793]
```

Visualizing the Tensorboard outcome

```
%load_ext tensorboard
%tensorboard --logdir log
```



TensorBoard SCALARS IMAGES INACTIVE

☐ Show data download links

☐ Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

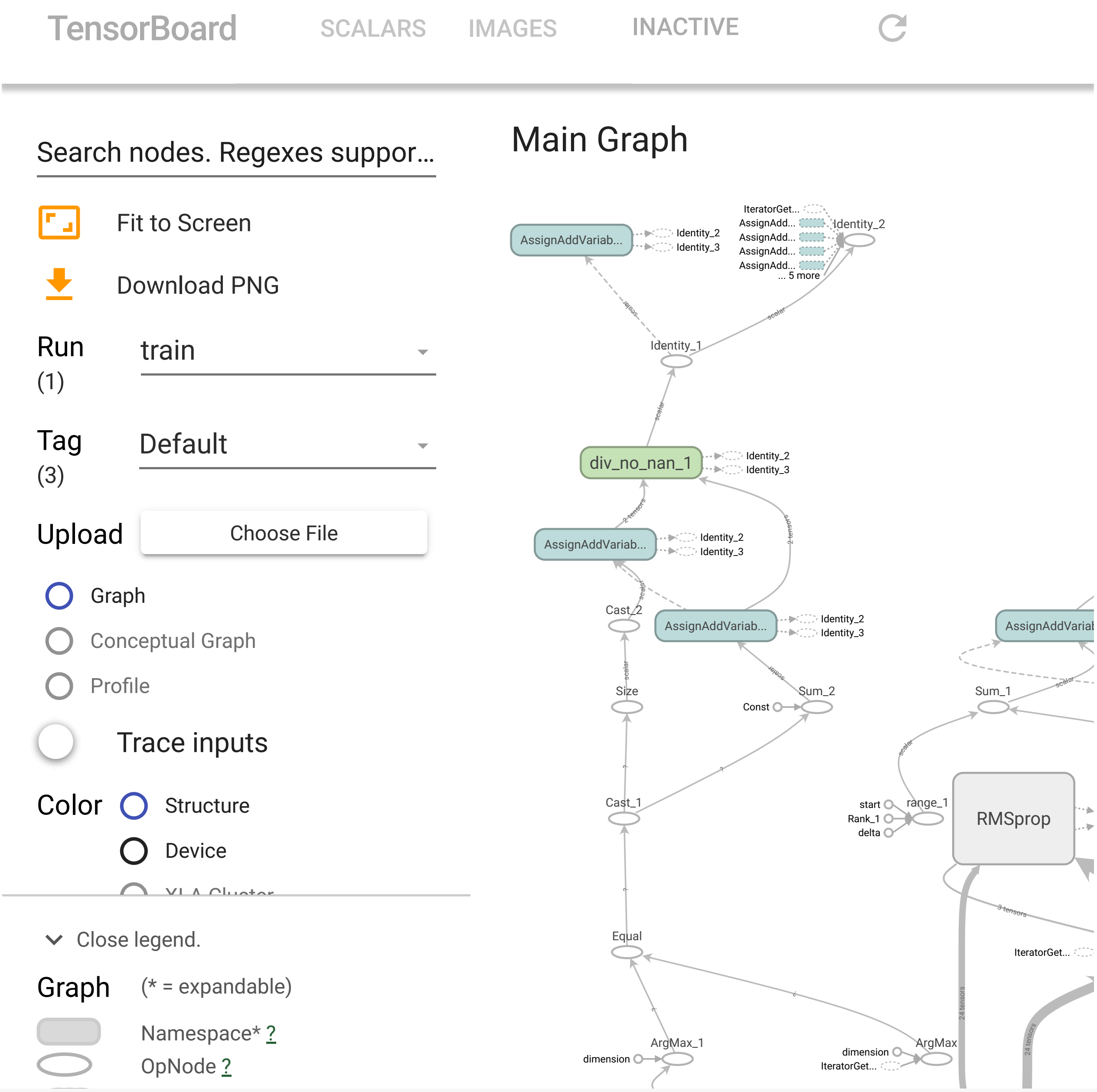
☐ ○ train

☐ ○ validation

TOGGLE ALL RUNS

log

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 969), started 0:17:22 ago. (Use '!kill 969
```



Run (1) train

Tag (3) Default

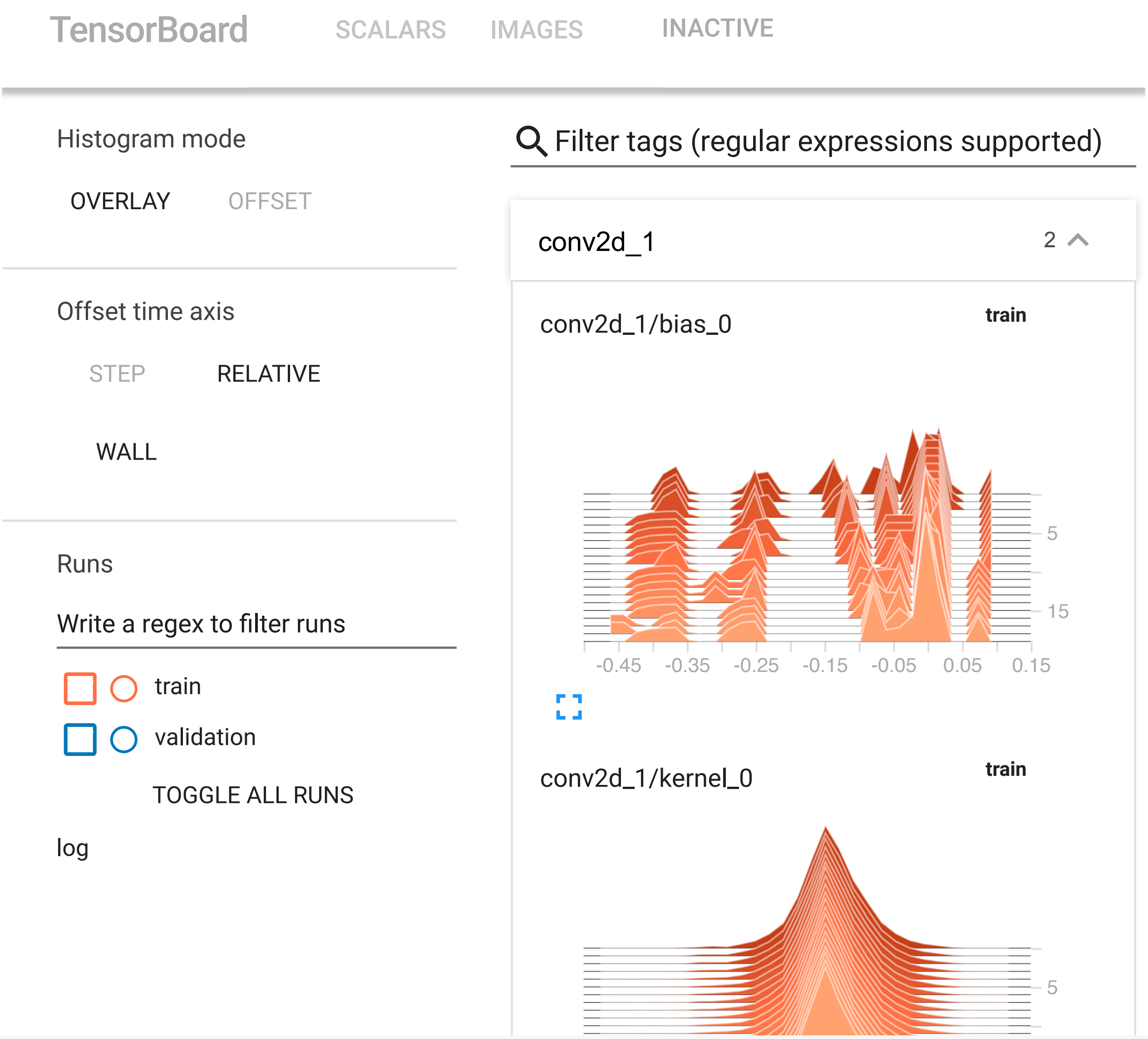
○ Graph ○ Conceptual Graph ○ Profile

○ Trace inputs

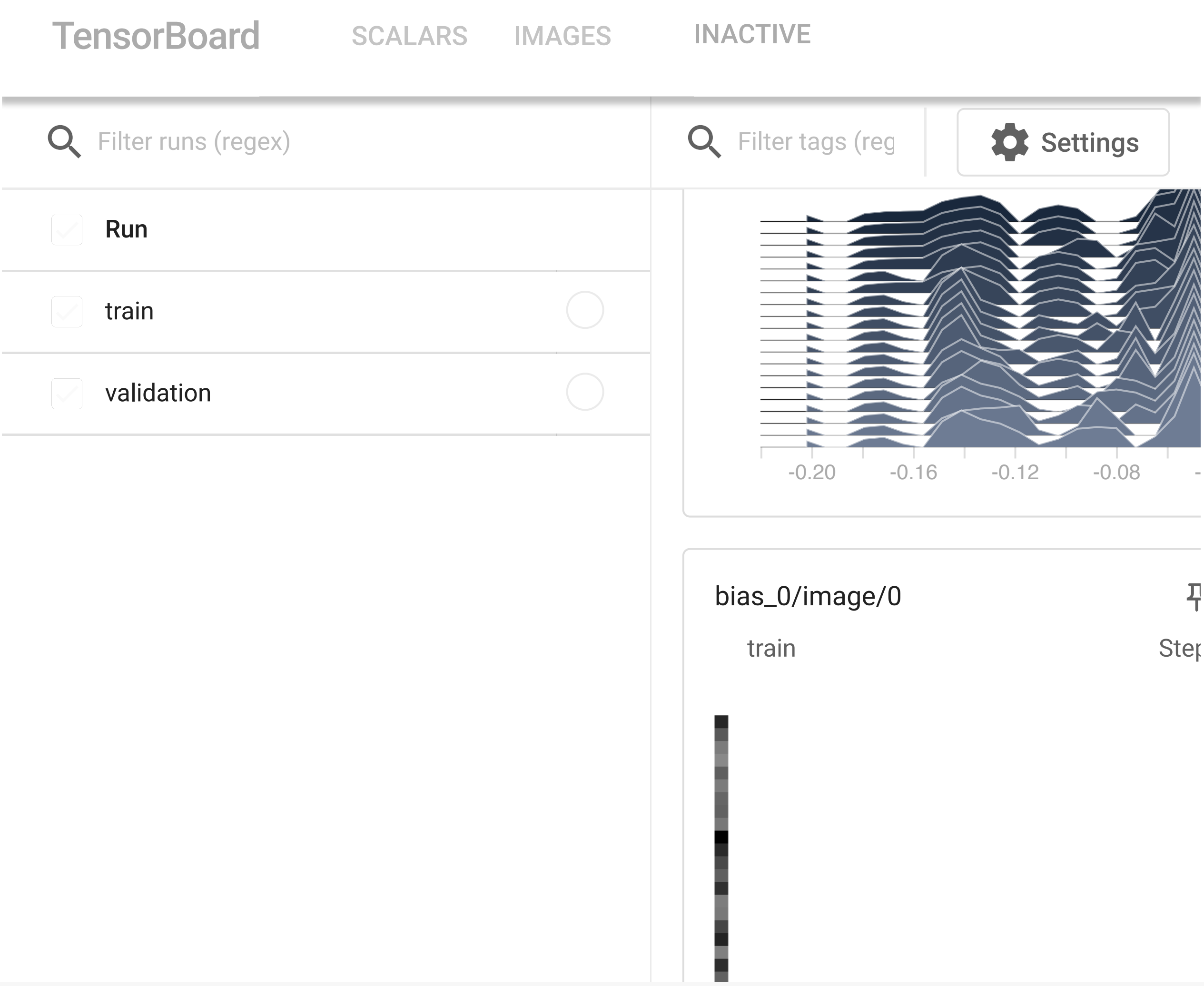
Color ○ Structure ○ Device ○ XLA Cluster

Graph (* = expandable)

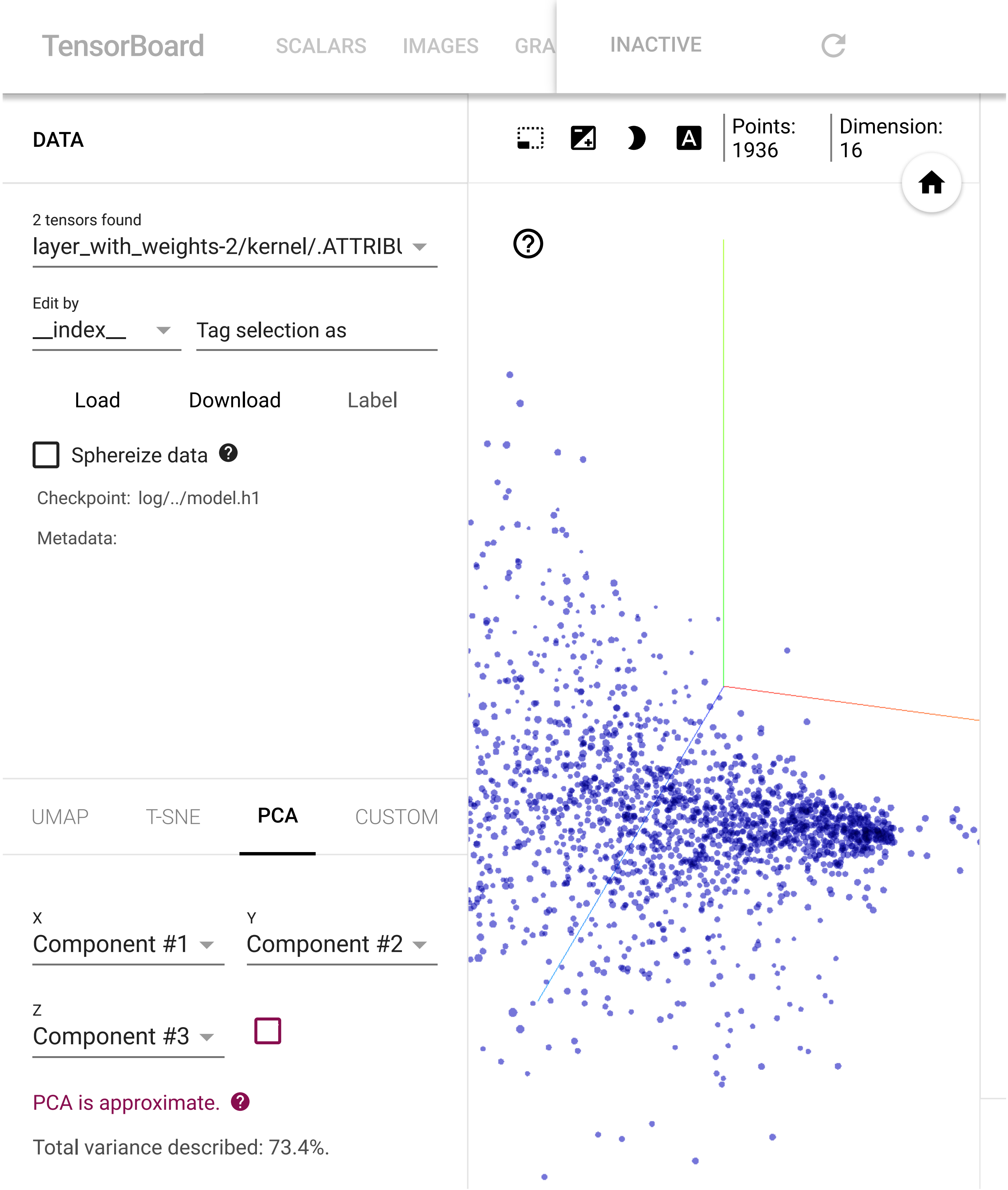
```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 969), started 0:17:26 ago. (Use '!kill 969
```



```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 969), started 0:23:00 ago. (Use '!kill 969
```



```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 969), started 0:23:09 ago. (Use '!kill 969
```



Points: 1936

Dimension: 16

2 tensors found
layer_with_weights-2/kernel/.ATTRIBl

Edit by
__index__

Checkpoint: log/../model.h1

Metadata:

X
Component #1

Y
Component #2

Z
Component #3

PCA is approximate.

Total variance described: 73.4%.

✓ 0s completed at 8:05 PM