```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Dense,Input,Activation
from tensorflow.keras.models import Model
import tensorflow as tf
from tensorflow import keras
from keras.callbacks import Callback
from sklearn.metrics import roc_auc_score,f1_score
import random
import datetime,os
```

```
'''!pip uninstall tensorflow'''
```

```
Found existing installation: tensorflow 2.8.0+zzzcolab20220506162203
Uninstalling tensorflow-2.8.0+zzzcolab20220506162203:
  Would remove:
    /usr/local/bin/estimator_ckpt_converter
    /usr/local/bin/import_pb_to_tensorboard
    /usr/local/bin/saved_model_cli
    /usr/local/bin/tensorboard
    /usr/local/bin/tf_upgrade_v2
    /usr/local/bin/tflite_convert
    /usr/local/bin/toco
    /usr/local/bin/toco_from_protos
    /usr/local/lib/python3.7/dist-packages/tensorflow-2.8.0+zzzcolab20220506162203.dist-
    /usr/local/lib/python3.7/dist-packages/tensorflow/*
Proceed (y/n)? ERROR: Operation cancelled by user
^C
```

```
'''!pip install tensorflow==2.7.0'''
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Collecting tensorflow==2.7.0
  Downloading https://us-python.pkg.dev/colab-wheels/public/tensorflow/tensorflow-2.7
     \ 665.5 MB 113.6 MB/s
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/d
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-p
Collecting tensorflow-estimator<2.8,~=2.7.0rc0
  Downloading tensorflow_estimator-2.7.0-py2.py3-none-any.whl (463 kB)
     |████████████████████████████████| 463 kB 6.6 MB/s
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7
Requirement already satisfied: wheel<1.0,>=0.32.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in /usr/local/lib
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /usr/local/lib/python3.7/dis
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tensorboard~=2.6 in /usr/local/lib/python3.7/dist-pack
Collecting keras<2.8,>=2.7.0rc0
  Downloading keras-2.7.0-py2.py3-none-any.whl (1.3 MB)
     |████████████████████████████████| 1.3 MB 44.8 MB/s
Collecting gast<0.5.0,>=0.2.1
  Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/pyt
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/li
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/d
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/di
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packa
Installing collected packages: tensorflow-estimator, keras, gast, tensorflow
```

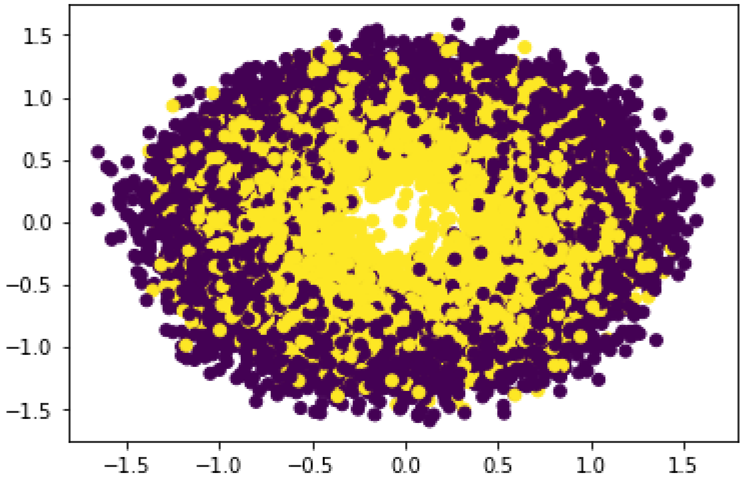
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.moun
```

```
df = pd.read_csv('/content/drive/My Drive/data.csv')
print(df.head(2))
x=df[['f1','f2']].values
y=df['label'].values
```

```
         f1        f2  label
0  0.450564  1.074305    0.0
1  0.085632  0.967682    0.0
```

```
plt.scatter(df['f1'],df['f2'],c=y)
plt.show()
```



```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.25) #train test split for
```

```
class Metrics(tf.keras.callbacks.Callback):
  def __init__(self):
    self.validation_data=(x_test,y_test)
  def on_train_begin(self, logs={}):
    self.val_f1s = []
  def on_epoch_end(self, epoch, logs={}):
     val_predict = (np.asarray(self.model.predict(self.validation_data[0]))).round()
     val_targ = self.validation_data[1]
     val_f1 = f1_score(val_targ, val_predict.round())
     roc_val=roc_auc_score(val_targ, val_predict)
     self.val_f1s.append(val_f1)
     print("-f1 score :",val_f1,"-ROCValue :", roc_val)
```

```
'''class Metrics(tf.keras.callbacks.Callback):
  def _init_(self):
    self.validation_data=(x_test,y_test)
  def on_train_begin(self,logs={}):
    self.val_f1score = []
  def on_epoch_end(self,epoch,logs={}):
    val_predict = (np.asarray(self.model.predict(self.validation_data[0]))).round()
    print(val_predict)
    val_targ = self.validation_data[1]
    val_f1 = f1_score(val_targ,val_predict.round())
    roc_val = roc_auc_score(val_targ,val_predict.round())
    self.val_f1score.append(val_f1)

    print(val_targ)
    print(val_f1)
    print("f1 score: ",val_f1)
    print("roc-auc score: ",roc_val)'''
```

```
'class Metrics(tf.keras.callbacks.Callback):\n  def _init_(self):\n    self.validation_
data=(x_test,y_test)\n  def on_train_begin(self,logs={}):\n    self.val_f1score = [] \n
def on_epoch_end(self,epoch,logs={}):\n    val_predict = (np.asarray(self.model.predict
(self.validation_data[0]))).round()   \n    print(val_predict)\n    val_targ = self.val
idation_data[1]\n    val_f1 = f1_score(val_targ,val_predict.round())\n    roc_val = roc
```

```
class TerminateNaN(tf.keras.callbacks.Callback):
  def on_epoch_end(self,epoch,logs={}):
    loss = logs.get('loss')
    if loss is not None:
      if np.isnan(loss) or np.isinf(loss):
        print("invalid loss and terminated at epoch",epoch)
        self.model.stop_training = True
```

```
def lr_scheduler(epoch,lr):
  decay_rate = 0.95
  decay_step = 3
  if (epoch+1) % decay_step == 0:
    return lr*decay_rate
  return lr
```

```
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.callbacks import LearningRateScheduler
```

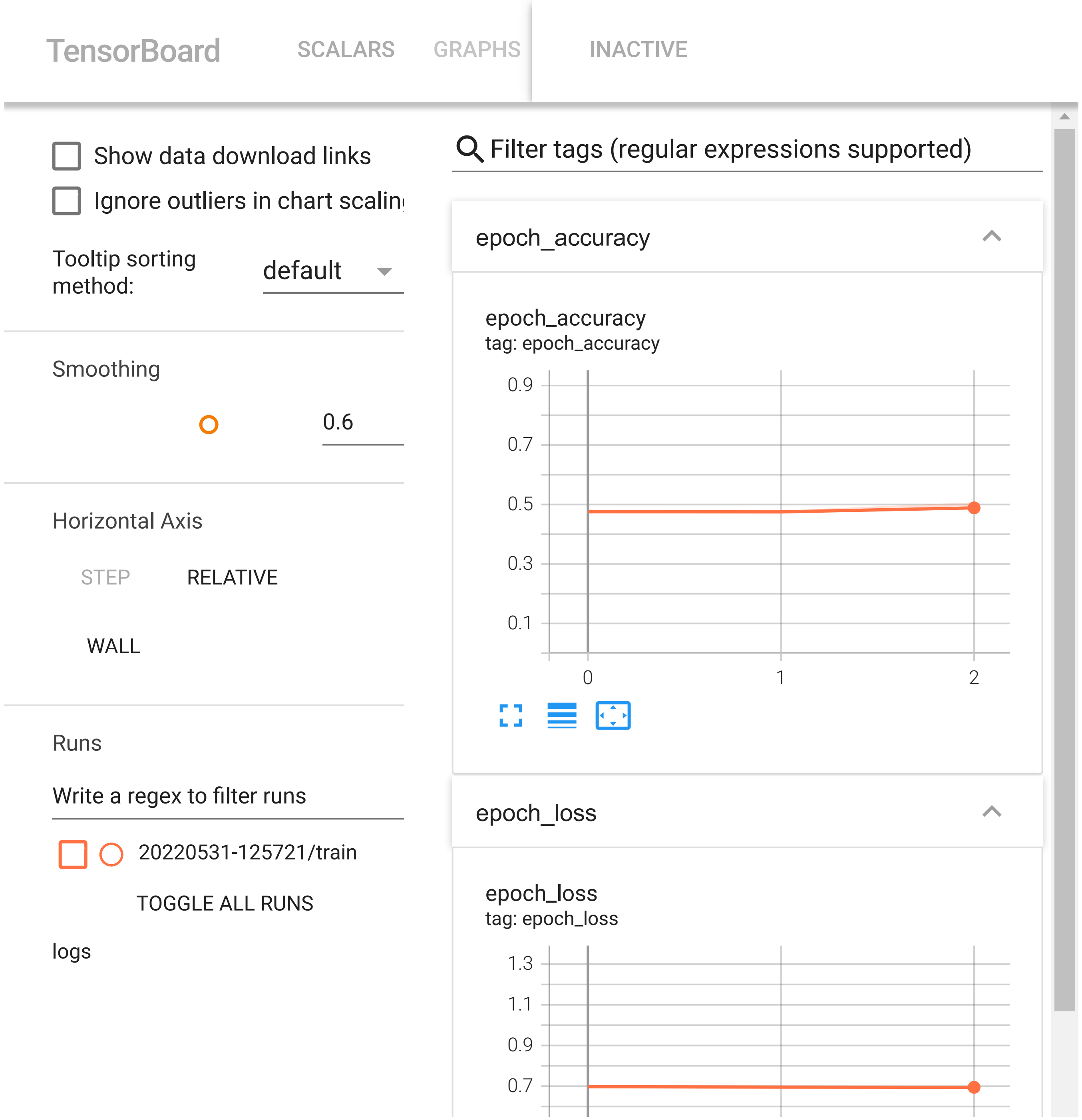
```
%load_ext tensorboard
```

```
from keras.initializers import RandomUniform
def create_model1():
  return tf.keras.models.Sequential([
                                   tf.keras.layers.Dense(2,activation='tanh',input_shape=(2
                                   tf.keras.layers.Dense(16,activation='tanh',kernel_initia
                                   tf.keras.layers.Dense(16,activation='tanh',kernel_initia
                                   tf.keras.layers.Dense(16,activation ='tanh',kernel_initi
                                   tf.keras.layers.Dense(16,activation = 'tanh',kernel_init
                                     tf.keras.layers.Dense(16,activation='tanh',kernel_initia
                                     tf.keras.layers.Dense(1,activation = 'softmax',kernel_in
  ])


filepath="/content/drive/My Drive/weights_CB.hdf5"
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.9, patience=1, min_lr=0.0001)
lrschedule = LearningRateScheduler(lr_scheduler, verbose=0)
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy',  verbose=1, save_best
earlystop = EarlyStopping(monitor='val_accuracy', min_delta=0.35, patience=2, verbose=1)
terminate= TerminateNaN()
metrics=Metrics()
logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)
```

```
model_1 = create_model1()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01,momentum = 0.0,nesterov=False,name='SGD'
model_1.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model_1.fit(x=x_train,y=y_train,epochs=15,validation_data=(x_test,y_test),callbacks=[metrics,
```

```
Epoch 1/15
  1/469 [..............................] - ETA: 4:40 - loss: 3.3464 - accuracy: 0.5625WA
463/469 [============================>.] - ETA: 0s - loss: 1.0236 - accuracy: 0.5005-f1

Epoch 00001: val_accuracy improved from -inf to 0.49900, saving model to /content/drive/
469/469 [==============================] - 2s 3ms/step - loss: 1.0195 - accuracy: 0.500
Epoch 2/15
463/469 [============================>.] - ETA: 0s - loss: 0.6936 - accuracy: 0.5000-f1

Epoch 00002: val_accuracy did not improve from 0.49900
469/469 [==============================] - 1s 3ms/step - loss: 0.6936 - accuracy: 0.500
Epoch 3/15
466/469 [============================>.] - ETA: 0s - loss: 0.6933 - accuracy: 0.5005-f1

Epoch 00003: val_accuracy did not improve from 0.49900
469/469 [==============================] - 1s 3ms/step - loss: 0.6933 - accuracy: 0.500
Epoch 00003: early stopping
<keras.callbacks.History at 0x7fb9f3118550>
```

```
%tensorboard --logdir logs
```



□ 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

□ ○ 20220531-125721/train

TOGGLE ALL RUNS

logs

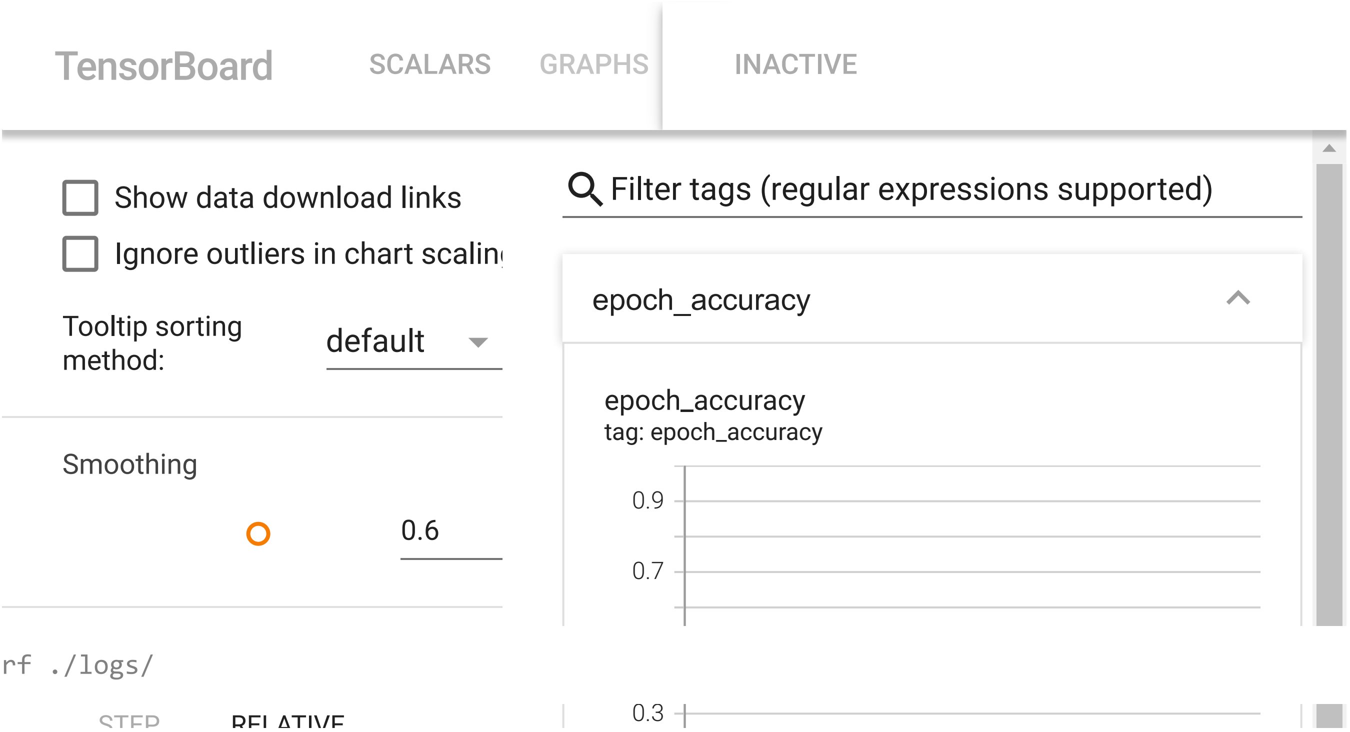
```
def create_model2():
  return tf.keras.models.Sequential([
                                   tf.keras.layers.Dense(2,activation='relu',input_shape=(2
                                   tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                   tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                   tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                   tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                   tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                   tf.keras.layers.Dense(1,activation='softmax',kernel_init


  ])
```

```
model_2 = create_model2()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01,momentum = 0.0,nesterov=False,name='SGD'
model_2.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model_2.fit(x=x_train,y=y_train,epochs=15,validation_data=(x_test,y_test),callbacks=[metrics,
```

```
Epoch 1/15
  1/469 [..............................] - ETA: 4:04 - loss: 22.5956 - accuracy: 0.4062W
468/469 [============================>.] - ETA: 0s - loss: 0.7342 - accuracy: 0.5006-f1

Epoch 00001: val_accuracy did not improve from 0.49900
469/469 [==============================] - 2s 3ms/step - loss: 0.7340 - accuracy: 0.5003
Epoch 2/15
465/469 [============================>.] - ETA: 0s - loss: 0.6851 - accuracy: 0.5007-f1

Epoch 00002: val_accuracy did not improve from 0.49900
469/469 [==============================] - 1s 3ms/step - loss: 0.6854 - accuracy: 0.5003
Epoch 3/15
462/469 [============================>.] - ETA: 0s - loss: 0.6845 - accuracy: 0.5004-f1

Epoch 00003: val_accuracy did not improve from 0.49900
469/469 [==============================] - 1s 3ms/step - loss: 0.6843 - accuracy: 0.5003
Epoch 00003: early stopping
<keras.callbacks.History at 0x7fba6b6045d0>
```

```
%tensorboard --logdir logs
```

```
Reusing TensorBoard on port 6006 (pid 1025), started 0:00:06 ago. (Use '!kill 1025' to
kill it.)
```



☐ Show data download links

☐ Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing 0.6

epoch_accuracy
tag: epoch_accuracy

```
!rm -rf ./logs/
```

```
def create_model3():
  return tf.keras.models.Sequential([
                                     tf.keras.layers.Dense(2,activation='relu',input_shape=(2
                                     tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                     tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                     tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                     tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                     tf.keras.layers.Dense(16,activation='relu',kernel_initia
                                     tf.keras.layers.Dense(1,activation='softmax',kernel_init


  ])
```

```
model_3 = create_model3()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01,momentum = 0.0,nesterov=False,name='SGD'
model_3.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model_3.fit(x=x_train,y=y_train,epochs=15,validation_data=(x_test,y_test),callbacks=[metrics,
```

```
Epoch 1/15
  1/469 [..............................] - ETA: 5:28 - loss: 1.2137 - accuracy: 0.3438W
461/469 [============================>.] - ETA: 0s - loss: 0.7057 - accuracy: 0.5005-f1

Epoch 00001: val_accuracy did not improve from 0.49900
469/469 [==============================] - 2s 3ms/step - loss: 0.7054 - accuracy: 0.500
Epoch 2/15
463/469 [============================>.] - ETA: 0s - loss: 0.6863 - accuracy: 0.5007-f1

Epoch 00002: val_accuracy did not improve from 0.49900
469/469 [==============================] - 1s 3ms/step - loss: 0.6863 - accuracy: 0.500
Epoch 3/15
```

```
457/469 [============================>.] - ETA: 0s - loss: 0.6807 - accuracy: 0.4997-f1

Epoch 00003: val_accuracy did not improve from 0.49900
469/469 [==============================] - 1s 2ms/step - loss: 0.6808 - accuracy: 0.5003
Epoch 00003: early stopping
<keras.callbacks.History at 0x7fba6af72150>
```

```
%tensorboard --logdir logs
```

```
     Reusing TensorBoard on port 6006 (pid 1025), started 0:00:11 ago. (Use '!kill 1025' to
     kill it.)
```

```
!rm -rf ./logs/
```

```
def create_model4():
  return tf.keras.models.Sequential([
                                    tf.keras.layers.Dense(4,activation='relu',input_shape=(2
                                    tf.keras.layers.Dense(8,activation='relu',kernel_initial
                                    tf.keras.layers.Dense(8,activation='relu',kernel_initial
                                    tf.keras.layers.Dense(8,activation='relu',kernel_initial
                                    tf.keras.layers.Dense(8,activation='relu',kernel_initial
                                    tf.keras.layers.Dense(8,activation='relu',kernel_initial
                                    tf.keras.layers.Dense(1,activation='sigmoid',kernel_init



  ])
```

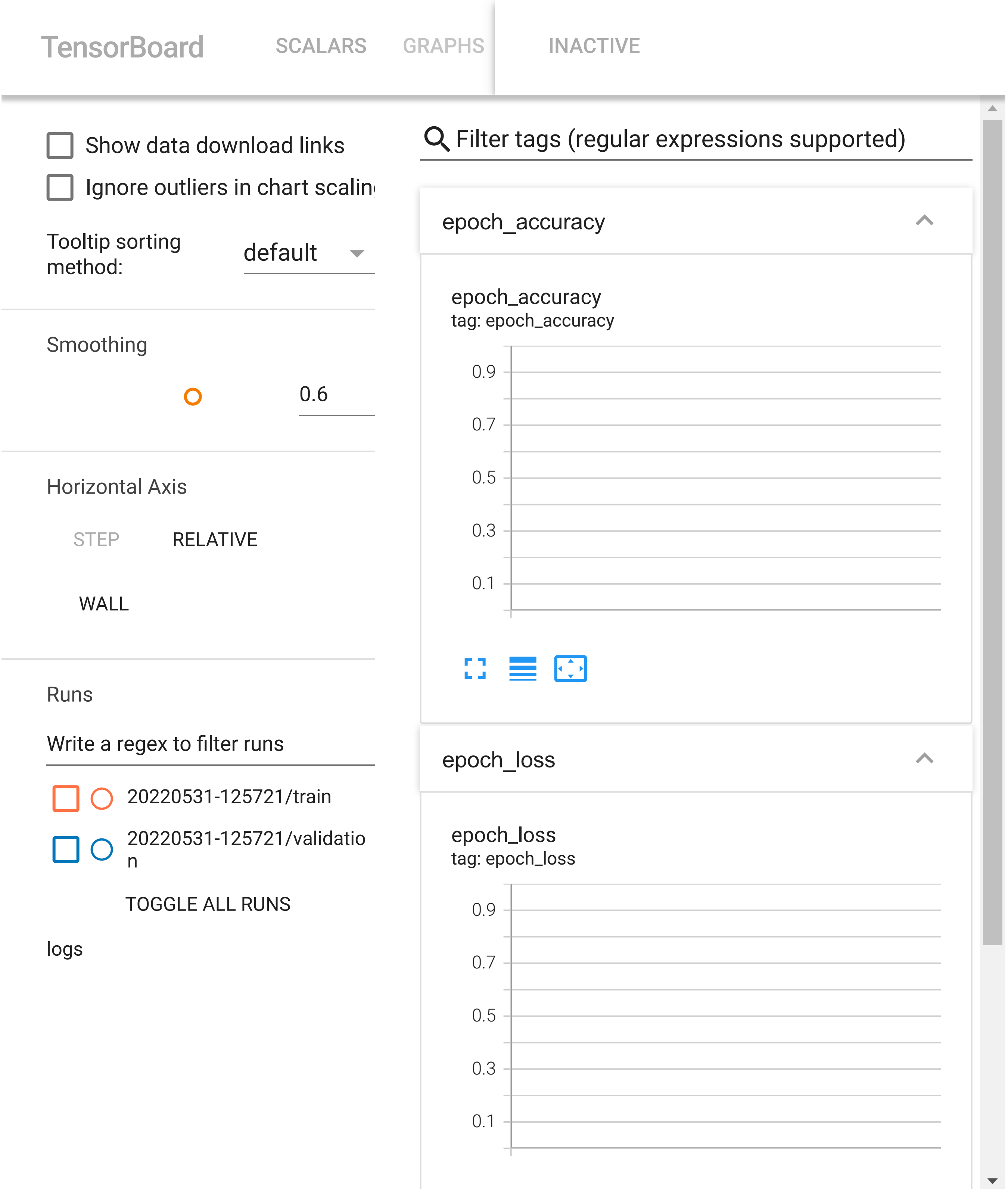
Horizontal Axis

```
model_4 = create_model4()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01,momentum = 0.0,nesterov=False,name='adam
model_4.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model_4.fit(x=x_train,y=y_train,epochs=15,validation_data=(x_test,y_test),callbacks=[metrics,
```

```
     Epoch 1/15
       1/469 [..............................] - ETA: 4:02 - loss: 0.7184 - accuracy: 0.5938W
     438/469 [===========================>..] - ETA: 0s - loss: 0.6974 - accuracy: 0.4774-f1

     Epoch 00001: val_accuracy did not improve from 0.49900
     469/469 [==============================] - 2s 3ms/step - loss: 0.6973 - accuracy: 0.475
     Epoch 2/15
     439/469 [===========================>..] - ETA: 0s - loss: 0.6942 - accuracy: 0.4751-f1

     Epoch 00002: val_accuracy did not improve from 0.49900
     469/469 [==============================] - 1s 3ms/step - loss: 0.6942 - accuracy: 0.474
     Epoch 3/15
     454/469 [============================>.] - ETA: 0s - loss: 0.6934 - accuracy: 0.5016-f1

     Epoch 00003: val_accuracy improved from 0.49900 to 0.51980, saving model to /content/dr
     469/469 [==============================] - 2s 4ms/step - loss: 0.6934 - accuracy: 0.501
     Epoch 00003: early stopping
     <keras.callbacks.History at 0x7fba6af48550>
```

```
%tensorboard --logdir logs
```

```
Reusing TensorBoard on port 6006 (pid 1025), started 0:00:18 ago. (Use '!kill 1025' to
kill it.)
```



□ Show data download links

□ Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing

○ 0.6

Horizontal Axis

Runs

Write a regex to filter runs

□ ○ 20220531-125721/train

□ ○ 20220531-125721/validation

logs

Filter tags (regular expressions supported)

epoch_accuracy

epoch_accuracy
tag: epoch_accuracy

epoch_loss

epoch_loss
tag: epoch_loss