1. Download all the data in this folder https://drive.google.com/open?id=1Z4TyI7FcFVEx8q

path/to/the/image.tif,category

where the categories are numbered 0 to 15, in the following order:

0 letter
1 form
2 email
3 handwritten
4 advertisement
5 scientific report
6 scientific publication
7 specification
8 file folder
9 news article
10 budget
11 invoice
12 presentation
13 questionnaire
14 resume
15 memo

2. On this image data, you have to train 3 types of models as given below. You have to s

Automatic saving failed. This file was updated remotely or in another tab. Show diff

have given th

or you can use this method also
https://medium.com/@vijayabhaskar96/tutorial-on-keras-imagedatagenerator-with-flow-from-

https://medium.com/@vijayabhaskar96/tutorial-on-keras-flow-from-dataframe-1fd4493d237c

4. You are free to choose Learning rate, optimizer, loss function, image augmentation, a

5. Use tensorboard for every model and analyse your gradients. (you need to upload the s

Note: fit_genarator() method will have problems with the tensorboard histograms, try to

6. You can check about Transfer Learning in this link - https://blog.keras.io/building-p

06750000/00484516897554883881/03543900857199698311/1Z4TyI7FcFVEx8qdl4jO9qxvxaqLSqoEu?e=dow

```
--2020-07-08 11:15:56--  https://doc-0c-0g-docs.googleusercontent.com/docs/securesc/
Resolving doc-0c-0g-docs.googleusercontent.com (doc-0c-0g-docs.googleusercontent.com)
Connecting to doc-0c-0g-docs.googleusercontent.com (doc-0c-0g-docs.googleusercontent
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/rar]
Saving to: ‘rvl-cdip.rar’

rvl-cdip.rar            [    <=>             ]   4.34G  80.3MB/s    in 85s

2020-07-08 11:17:22 (52.1 MB/s) - ‘rvl-cdip.rar’ saved [4660541790]
```

```
!unrar x '/content/rvl-cdip.rar'
```

```
import pandas as pd
data = pd.read_csv("labels_final.csv")
```

```
data.head()
```

	path	label
0	imagesv/v/o/h/voh71d00/509132755+-2755.tif	3
1	imagesl/l/x/t/lxt19d00/502213303.tif	3
2	imagesx/x/e/d/xed05a00/2075325674.tif	2
3	imageso/o/j/b/ojb60d00/517511301+-1301.tif	3
4	imagesq/q/z/k/qzk17e00/2031320195.tif	7

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
# separating data into train and test.
from sklearn.model_selection import train_test_split
train, test = train_test_split(data, test_size=0.2, random_state=42)
```

```
import tensorflow as tf
import os
import numpy as np
import pandas as pd
```

```
print(train.shape)
print(test.shape)
```

```
(38400, 2)
(9600, 2)
```

Model-1

1. Use VGG-16 pretrained network without Fully Connected layers and initilize all the we
2. After VGG-16 network without FC layers, add a new Conv block (1 Conv layer and 1 Max
3. Final architecture will be **INPUT --> VGG-16 without Top layers(FC) --> Conv Layer -->**
4. Train only new Conv block, FC layers, output layer. Don't train the VGG-16 network.

```
from keras import applications
from keras.preprocessing.image import ImageDataGenerator
from keras import optimizers
from keras.models import Sequential
from keras.layers import Dropout, Flatten, Dense
from tensorflow.keras.layers import Dense,Input,Conv2D,MaxPool2D,Activation,Dropout,Flatte
from tensorflow.keras.models import Model
import random as rn
import os
import datetime
datagen=ImageDataGenerator(rescale=1./255)
```

```
Using TensorFlow backend.
```

```
train_generator=datagen.flow_from_dataframe(dataframe=train, directory="data_final", x_col
                                                                              class_mode="
test_generator=datagen.flow_from_dataframe(dataframe=test, directory="/content/data_final"
                                                                               class_mode=
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
train_size=int(train.shape[0]/32)
test_size = int(test.shape[0]/32)
print(train_size)
print(test_size)
```

```
1200
300
```

```
os.environ['PYTHONHASHSEED'] = '0'

##https://keras.io/getting-started/faq/#how-can-i-obtain-reproducible-results-using-keras-
## Have to clear the session. If you are not clearing, Graph will create again and again a
## Varibles will also set to some value from before session
tf.keras.backend.clear_session()

## Set the random seed values to regenerate the model.
np.random.seed(0)
rn.seed(0)

#Input layer
input_layer = Input(shape=(224,224,3,),name='Input_Layer')
input_layer = Input(shape=(224,224,3),name='input_layer')

#VGG model
vgg_model = tf.keras.applications.VGG16(include_top=False, weights='imagenet',)
print('Model loaded.')
vgg_model.trainable=False
vgg= vgg_model(input_layer)


#Conv Layer
Conv1 = Conv2D(filters=128,kernel_size=(3,3),strides=(1,1),padding='valid',data_format='ch
              activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0)
#MaxPool Layer
Pool1 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='valid',data_format='channels_last

#Flatten
flatten = Flatten(data_format='channels_last',name='Flatten')(Pool1)

#FC layer
FC1 = Dense(units=64,activation='relu',kernel_initializer=tf.keras.initializers.glorot_nor

#FC layer
FC2 = Dense(units=32,activation='relu',kernel_initializer=tf.keras.initializers.glorot_nor

#output layer
Out = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.initializers.glorot_

#Creating a model
model1= Model(inputs=input_layer,outputs=Out)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
pplications/vg
    Model loaded.
```

```
model1.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
#compiling
model1.compile(optimizer=tf.keras.optimizers.Adam(lr=0.01),loss='categorical_crossentropy'
```

```
vgg16 (Model)                multiple                  14714688
```

```
%load_ext tensorboard
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)
```

```
_________________________________________________________________
Flatten (Flatten)            (None, 512)               0
```

```
%tensorboard --logdir $logdir
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

TensorBoard SCALARS GRAPHS INACTIVE

```
model1.fit_generator(generator=train_generator,steps_per_epoch=train_size,validation_data=
```

```
WARNING:tensorflow:From <ipython-input-17-e882bde338b6>:1: Model.fit_generator (from
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/15
1200/1200 [==============================] - 296s 247ms/step - loss: 2.0599 - accurac
Epoch 2/15
1200/1200 [==============================] - 289s 241ms/step - loss: 1.6197 - accurac
Epoch 3/15
1200/1200 [==============================] - 278s 232ms/step - loss: 1.4799 - accurac
Epoch 4/15
1200/1200 [==============================] - 274s 229ms/step - loss: 1.4088 - accurac
Epoch 5/15
1200/1200 [==============================] - 272s 227ms/step - loss: 1.3690 - accurac
Epoch 6/15
1200/1200 [==============================] - 270s 225ms/step - loss: 1.3351 - accurac
Epoch 7/15
1200/1200 [==============================] - 271s 226ms/step - loss: 1.3008 - accurac
Epoch 8/15
1200/1200 [==============================] - 268s 224ms/step - loss: 1.2787 - accurac
Epoch 9/15
1200/1200 [==============================] - 270s 225ms/step - loss: 1.2593 - accurac
Epoch 10/15
1200/1200 [==============================] - 270s 225ms/step - loss: 1.2400 - accurac
Epoch 11/15
1200/1200 [==============================] - 274s 228ms/step - loss: 1.2106 - accurac
Epoch 12/15
1200/1200 [                              ] - 273s 228ms/step - loss: 1.1952 - accurac
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
                                                                     1876 - accurac
Epoch 14/15
1200/1200 [==============================] - 259s 216ms/step - loss: 1.1830 - accurac
Epoch 15/15
1200/1200 [==============================] - 265s 221ms/step - loss: 1.1563 - accurac
<tensorflow.python.keras.callbacks.History at 0x7fcde3f92e80>
```

Model-2

```
1. Use VGG-16 pretrained network without Fully Connected layers and initilize all the we
2. After VGG-16 network without FC layers, don't use FC layers, use conv layers only as
3. Final architecture will be VGG-16 without FC layers(without top), 2 Conv layers ident
3. Train only last 2 Conv layers identical to FC layers, 1 output layer. Don't train the
```

```
tf.keras.backend.clear_session()
## Set the random seed values to regenerate the model.
np.random.seed(0)
rn.seed(0)
```

```
#Input layer
input_layer = Input(shape=(224,224,3,))

#VGG model
vgg_model = tf.keras.applications.VGG16(include_top=False, weights='imagenet',input_shape=
vgg_model.trainable=False
```

```
vgg= vgg_model(input_layer)
#Conv Layer
FCConv1 = Conv2D(filters=4096,kernel_size=(7,7),strides=(1,1),padding='valid',data_format=
              activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0)

FCConv2 = Conv2D(filters=4096,kernel_size=(1,1),strides=(1,1),padding='valid',data_format=
              activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=30
#Flatten
flatten = Flatten(data_format='channels_last',name='Flatten')(FCConv2)
#output layer
Out = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.initializers.glorot_

#Creating a model
model2_new= Model(inputs=input_layer,outputs=Out)
```

```
model2_new.summary()
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

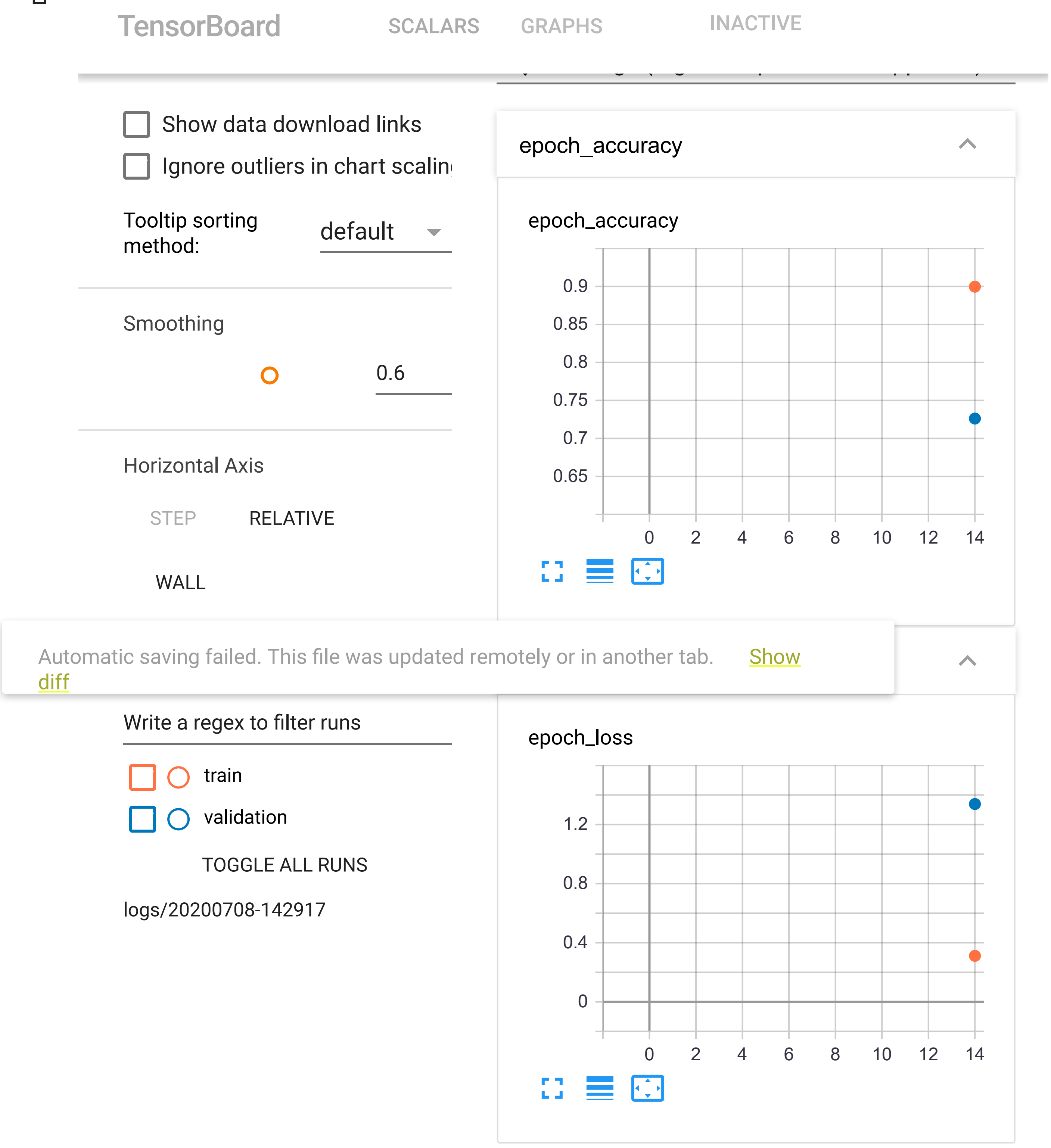
```
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
vgg16 (Model)                (None, 7, 7, 512)         14714688
_________________________________________________________________
conv2d (Conv2D)              (None, 1, 1, 4096)        102764544
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 1, 1, 4096)        16781312
_________________________________________________________________
Flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
Output (Dense)               (None, 16)                65552
=================================================================
Total params: 134,326,096
Trainable params: 119,611,408
Non-trainable params: 14,714,688
_________________________________________________________________
```

```
#compiling
model2_new.compile(optimizer=tf.keras.optimizers.Adam(lr=0.001),loss='categorical_crossent
```

```
%load_ext tensorboard
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)
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```

```
%tensorboard --logdir $logdir
```



☐ Show data download links
☐ Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis
STEP RELATIVE
WALL

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Write a regex to filter runs
☐ ○ train
☐ ○ validation
TOGGLE ALL RUNS
logs/20200708-142917

epoch_accuracy

```
model2_new.fit_generator(generator=train_generator,steps_per_epoch=train_size,validation_d
Epoch 1/15
1200/1200 [==============================] - 425s 354ms/step - loss: 1.4449 - accura
Epoch 2/15
1200/1200 [==============================] - 425s 354ms/step - loss: 0.9344 - accura
Epoch 3/15
1200/1200 [==============================] - 425s 354ms/step - loss: 0.7969 - accura
Epoch 4/15
1200/1200 [==============================] - 425s 354ms/step - loss: 0.6980 - accura
Epoch 5/15
1200/1200 [==============================] - 425s 354ms/step - loss: 0.6260 - accura
Epoch 6/15
1200/1200 [==============================] - 425s 354ms/step - loss: 0.5546 - accura
Epoch 7/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.5096 - accura
Epoch 8/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.4589 - accura
Epoch 9/15
1200/1200 [==============================] - 427s 356ms/step - loss: 0.4179 - accura
Epoch 10/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.3950 - accura
Epoch 11/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.3611 - accura
Epoch 12/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.3401 - accura
Epoch 13/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.3119 - accura
Epoch 14/15
1200/1200 [==============================] - 426s 355ms/step - loss: 0.2984 - accura
```

Model-3

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
1. Use same network as Model-2 'INPUT --> VGG-16 without Top layers(FC) --> 2 Conv Layer
```

```
tf.keras.backend.clear_session()
## Set the random seed values to regenerate the model.
np.random.seed(0)
rn.seed(0)
#Input layer
input_layer = Input(shape=(224,224,3,))

#VGG model
vgg_model = tf.keras.applications.VGG16(include_top=False, weights='imagenet',input_shape=
for i in range(0,19):
  if i<13:
    vgg_model.layers[i].trainable=False
  else:
    vgg_model.layers[i].trainable=True
```

```
vgg= vgg_model(input_layer)
#Conv Layer
FCConv1 = Conv2D(filters=128,kernel_size=(7,7),strides=(1,1),padding='valid',data_format='
              activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0)

FCConv2 = Conv2D(filters=64,kernel_size=(1,1),strides=(1,1),padding='valid',data_format='c
              activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=30
#Flatten
flatten = Flatten(data_format='channels_last',name='Flatten')(FCConv2)
#output layer
Out = Dense(units=16,activation='softmax',kernel_initializer=tf.keras.initializers.glorot_

#Creating a model
model3_new= Model(inputs=input_layer,outputs=Out)
```

```
model3_new.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
vgg16 (Model)                (None, 7, 7, 512)         14714688
_________________________________________________________________
conv2d (Conv2D)              (None, 1, 1, 128)         3211392
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 1, 1, 64)          8256
```

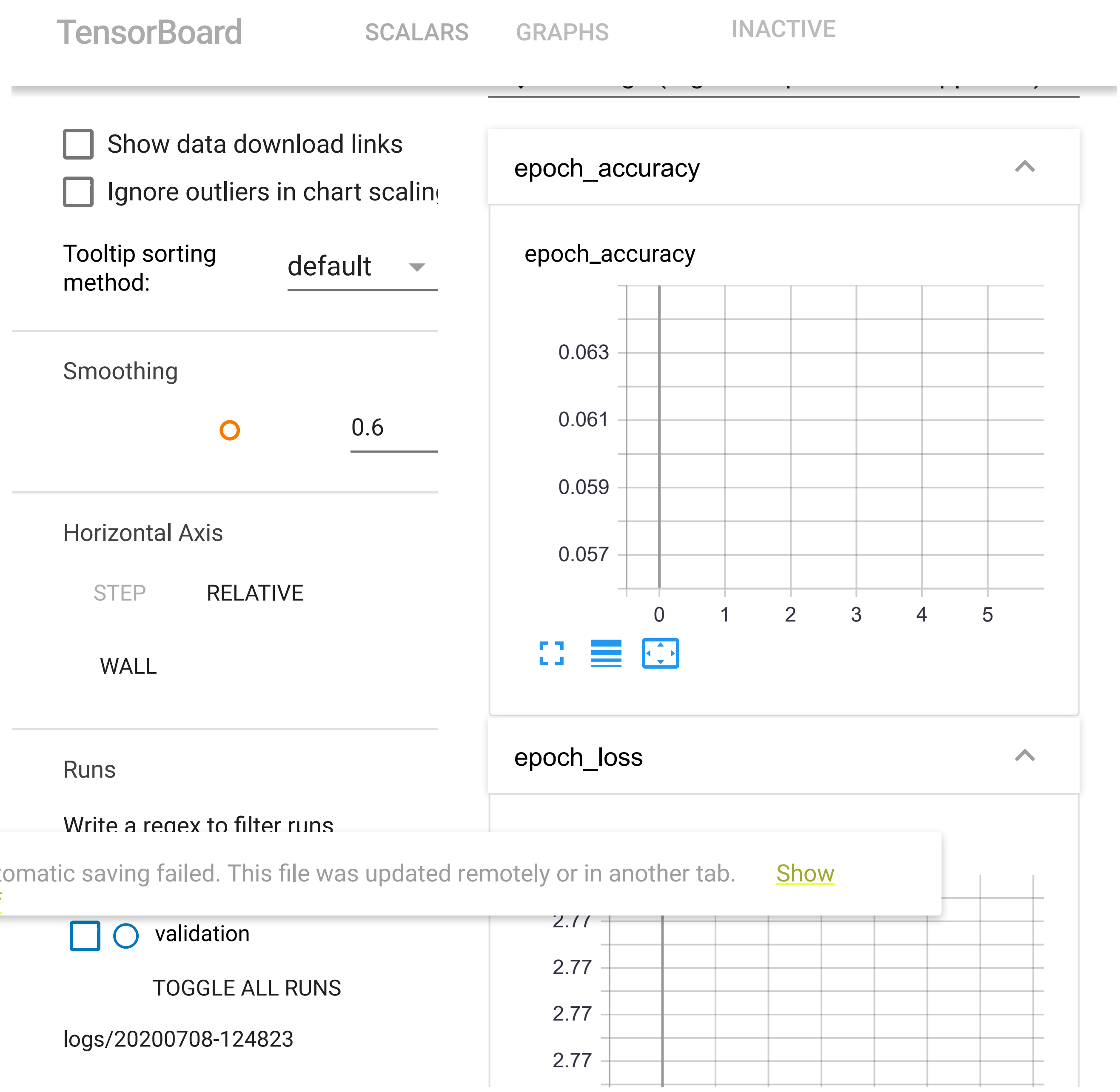
Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
=================================================================
Total params: 17,935,376
Trainable params: 12,659,920
Non-trainable params: 5,275,456
_________________________________________________________________
```

```
#compiling
model3_new.compile(optimizer=tf.keras.optimizers.Adam(lr=0.001),loss='categorical_crossent
```

```
%load_ext tensorboard
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)
%tensorboard --logdir $logdir
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```



Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
model3_new.fit_generator(generator=train_generator,steps_per_epoch=train_size,validation_d
Epoch 1/15
1200/1200 [==============================] - 294s 245ms/step - loss: 2.7774 - accura
Epoch 2/15
1200/1200 [==============================] - 292s 243ms/step - loss: 2.7728 - accura
Epoch 3/15
1200/1200 [==============================] - 292s 244ms/step - loss: 2.7728 - accura
Epoch 4/15
1200/1200 [==============================] - 294s 245ms/step - loss: 2.7728 - accura
Epoch 5/15
1200/1200 [==============================] - 290s 242ms/step - loss: 2.7728 - accura
Epoch 6/15
1200/1200 [==============================] - 293s 245ms/step - loss: 2.7728 - accura
Epoch 7/15
1200/1200 [==============================] - 290s 242ms/step - loss: 2.7728 - accura
Epoch 8/15
1200/1200 [==============================] - 293s 244ms/step - loss: 2.7728 - accura
Epoch 9/15
1200/1200 [==============================] - 290s 242ms/step - loss: 2.7728 - accura
Epoch 10/15
1200/1200 [==============================] - 293s 244ms/step - loss: 2.7728 - accura
Epoch 11/15
1200/1200 [==============================] - 290s 242ms/step - loss: 2.7728 - accura
Epoch 12/15
1200/1200 [==============================] - 293s 244ms/step - loss: 2.7728 - accura
Epoch 13/15
1200/1200 [==============================] - 290s 242ms/step - loss: 2.7728 - accura
Epoch 14/15
1200/1200 [==============================] - 292s 243ms/step - loss: 2.7728 - accura
Epoch 15/15
1200/1200 [==============================] - 290s 242ms/step - loss: 2.7728 - accura
<tensorflow.python.keras.callbacks.History at 0x7fcdd838a860>
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff