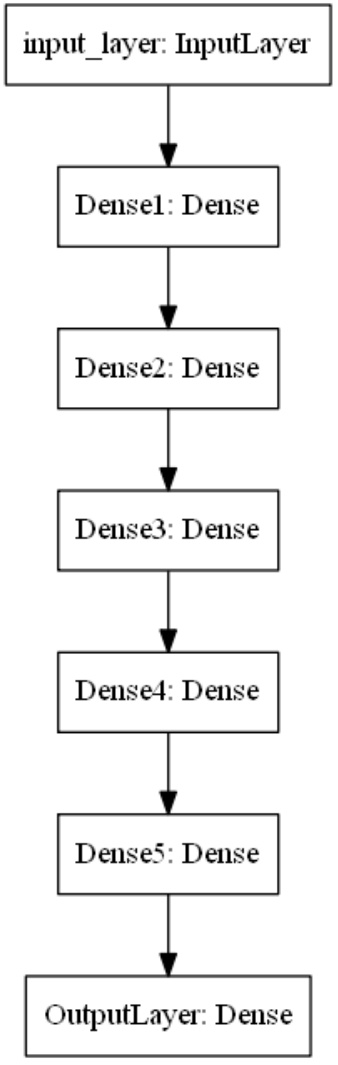
1. Download the data from here

2. Code the model to classify data like below image



3. Write your own callback function, that has to print the micro F1 score and AUC score after each epoch. **DONE**

4. Save your model at every epoch if your validation accuracy is improved from previous epoch. ** KINDA DONE**

5. you have to decay learning based on below conditions
    Cond1. If your validation accuracy at that epoch is less than previous epoch accuracy, you have to decrese the learning rate by 10%. DONE
    Cond2. For every 3rd epoch, decay your learning rate by 5%.DONE

6. If you are getting any NaN values(either weigths or loss) while training, you have to terminate your training. DONE

7. You have to stop the training if your validation accuracy is not increased in last 2 epochs. DONE

8. Use tensorboard for every model and analyse your gradients. (you need to upload the screenshots for each model for evaluation)

9. use cross entropy as loss function

10. Try the architecture params as given below.

Model-1

1. Use tanh as an activation for every layer except output layer.
2. use SGD with momentum as optimizer.
3. use RandomUniform(0,1) as initilizer.
3. Analyze your output and training process.

Model-2

1. Use relu as an activation for every layer except output layer.
2. use SGD with momentum as optimizer.
3. use RandomUniform(0,1) as initilizer.
3. Analyze your output and training process.

Model-3

1. Use relu as an activation for every layer except output layer.
2. use SGD with momentum as optimizer.
3. use he_uniform() as initilizer.
3. Analyze your output and training process.

Model-4

1. Try with any values to get better accuracy/f1 score.

```
%load_ext tensorboard
```

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

```
pip install tensorflow-addons
```

```
     Collecting tensorflow-addons
       Downloading tensorflow_addons-0.14.0-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)
          |████████████████████████████████| 1.1 MB 5.2 MB/s
     Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-packages (from tensorflow-addons) (2.7.1)
     Installing collected packages: tensorflow-addons
     Successfully installed tensorflow-addons-0.14.0
```

```
import tensorflow as tf
import pandas as pd
from sklearn.model_selection import train_test_split
import datetime

from tensorflow.keras.layers import Dense, Input, Activation
import keras
from tensorflow.keras.models import Model
import tensorflow_addons as tfa
from keras import metrics
from tensorflow.keras.callbacks import EarlyStopping, LearningRateScheduler, ReduceLROnPlateau, TerminateOnNaN, ModelCheckpoint
import random as rn
```

```
import numpy as np
from keras.callbacks import Callback
from sklearn.metrics import confusion_matrix, f1_score, precision_score, recall_score, roc_auc_score




data = pd.read_csv('data.csv')


X = data[['f1', 'f2']]
y = data['label']

print(X.shape)
print(y.shape)

     (20000, 2)
     (20000,)


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)


def changedLearningRate(epoch, lr):
  """In this function, we decrease the learning rate by 5% every third epoch"""
  if ((epoch+1)%3 == 0):
    lr = 0.95 * lr
  return float(lr)


class Metrics(Callback):

  def on_train_begin(self, logs={}):
    self.val_f1s = []
    self.val_recalls = []
    self.val_precisions = []
    self.val_aucs = []

  def on_epoch_end(self, epoch, logs={}):
    val_predict = (np.asarray(self.model.predict(X_test))).round()
    val_targ = y_test
    _val_f1 = f1_score(val_targ, val_predict)
    _val_recall = recall_score(val_targ, val_predict)
    _val_precision = precision_score(val_targ, val_predict)
    _val_auc = roc_auc_score(val_targ, val_predict)
    self.val_f1s.append(_val_f1)
    self.val_recalls.append(_val_recall)
    self.val_precisions.append( val_precision)
```

```
    self.val_aucs.append(_val_auc)

    print("val_f1: %f — val_precision: %f — val_recall %f — val_auc %f" %(_val_f1, _val_precision, _val_recall, _val_auc))

    return

metrics = Metrics()
#source: https://medium.com/@thongonary/how-to-compute-f1-score-for-each-epoch-in-keras-a1acd17715a2
```

Model 1

```
input_layer = Input(shape=(2, ))

layer1 = Dense(784, activation='tanh', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(input_layer)
layer2 = Dense(512, activation='tanh', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer1)
layer3 = Dense(256, activation='tanh', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer2)
layer4 = Dense(64, activation='tanh', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer3)
layer5 = Dense(32, activation='tanh', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer4)


output = Dense(1, activation='sigmoid', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer5)

model1 = Model(inputs = input_layer, outputs = output)
optimizer = tf.keras.optimizers.SGD(learning_rate=0.1, momentum=0.1, nesterov=True)

model1.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

log_dir="logs/fit/model1/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True, write_grads=True)

filepath = "model_save/weights-{epoch:02d}-{val_accuracy:.4f}.hdf5"

checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
reduce_lr = ReduceLROnPlateau(monitor = 'val_accuracy', factor=0.9, patience=1, verbose=1, mode='auto')

terminate_tr = TerminateOnNaN()
earlystop = EarlyStopping(monitor='val_accuracy', patience=2, verbose=1, mode='max')

lrschedule = LearningRateScheduler(changedLearningRate, verbose=1)
model1.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), validation_split=0.2, callbacks=[checkpoint, reduce_lr, terminate_tr,

     WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the `TensorBoard` Callback.
     Epoch 1/10
```

```
Epoch 00001: LearningRateScheduler setting learning rate to 0.10000000149011612.
  3/335 [..............................] - ETA: 43s - loss: 6.1710 - accuracy: 0.5521WARNING:tensorflow:Callback method `on_train_batch_begin` is
WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared to the batch time (batch time: 0.0048s vs `on_train_batch_end` time: 0.0
335/335 [==============================] - 5s 6ms/step - loss: 0.7967 - accuracy: 0.5089 - val_loss: 0.7089 - val_accuracy: 0.5086

Epoch 00001: val_accuracy improved from -inf to 0.50858, saving model to model_save/weights-01-0.5086.hdf5
val_f1: 0.491618 — val_precision: 0.491768 — val_recall 0.491469 — val_auc 0.494529
Epoch 2/10

Epoch 00002: LearningRateScheduler setting learning rate to 0.10000000149011612.
335/335 [==============================] - 1s 4ms/step - loss: 0.7062 - accuracy: 0.4989 - val_loss: 0.6951 - val_accuracy: 0.5086

Epoch 00002: val_accuracy did not improve from 0.50858

Epoch 00002: ReduceLROnPlateau reducing learning rate to 0.09000000134110452.
val_f1: 0.491618 — val_precision: 0.491768 — val_recall 0.491469 — val_auc 0.494529
Epoch 3/10

Epoch 00003: LearningRateScheduler setting learning rate to 0.08550000339746475.
335/335 [==============================] - 1s 4ms/step - loss: 0.7037 - accuracy: 0.4998 - val_loss: 0.6948 - val_accuracy: 0.5086

Epoch 00003: val_accuracy did not improve from 0.50858

Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.07695000171661377.
val_f1: 0.491618 — val_precision: 0.491768 — val_recall 0.491469 — val_auc 0.494529
Epoch 00003: early stopping
<keras.callbacks.History at 0x7fa011530c50>
```

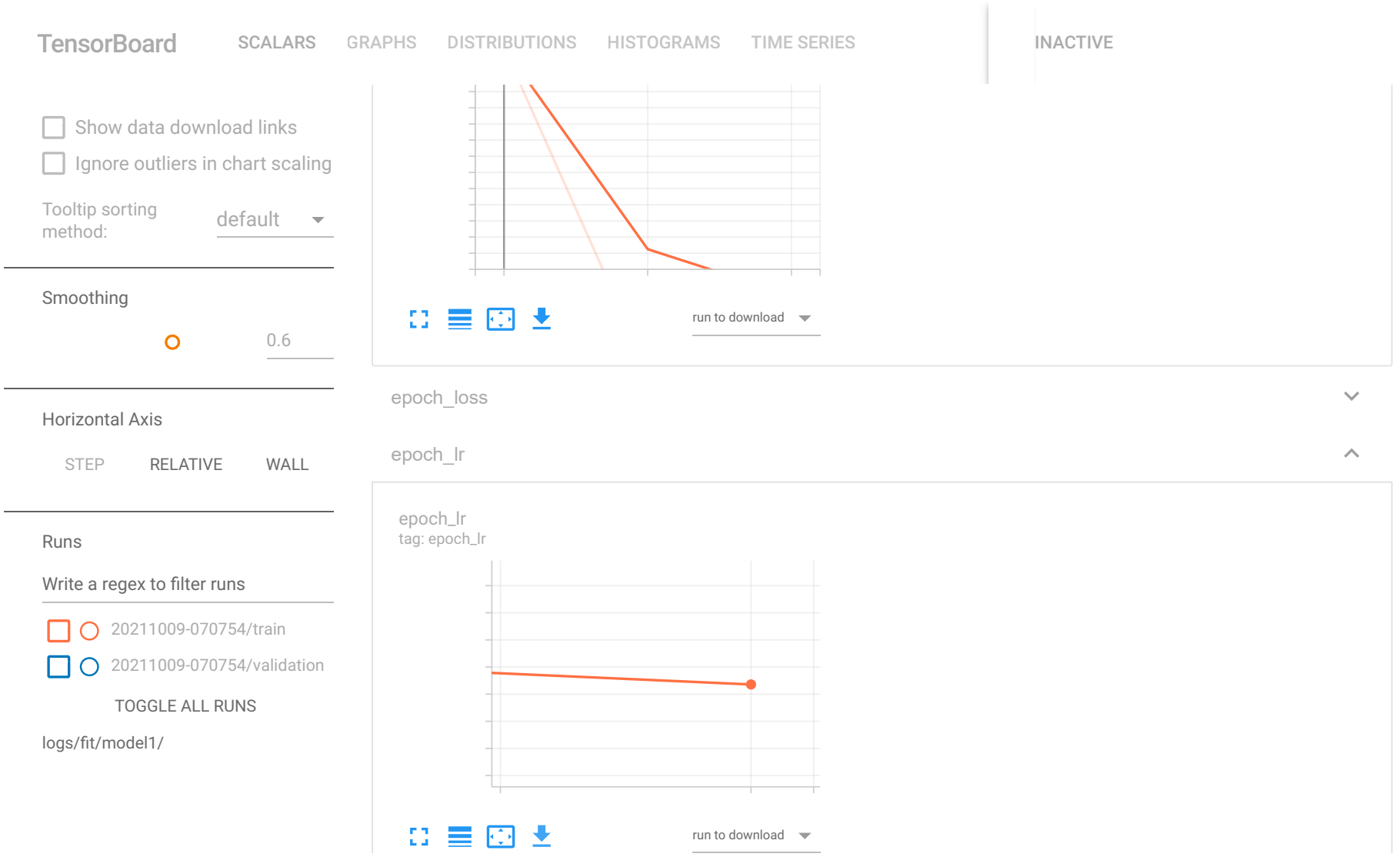
```
%load_ext tensorboard
```

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

```
!kill 197
```

```
/bin/bash: line 0: kill: (197) - No such process
```

```
%tensorboard --logdir logs/fit/model1/
```



Model 2

```
input_layer = Input(shape=(2, ))

layer1 = Dense(784, activation='relu', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(input_layer)
layer2 = Dense(512, activation='relu', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer1)
layer3 = Dense(256, activation='relu', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer2)
layer4 = Dense(64, activation='relu', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer3)
layer5 = Dense(32, activation='relu', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer4)
```

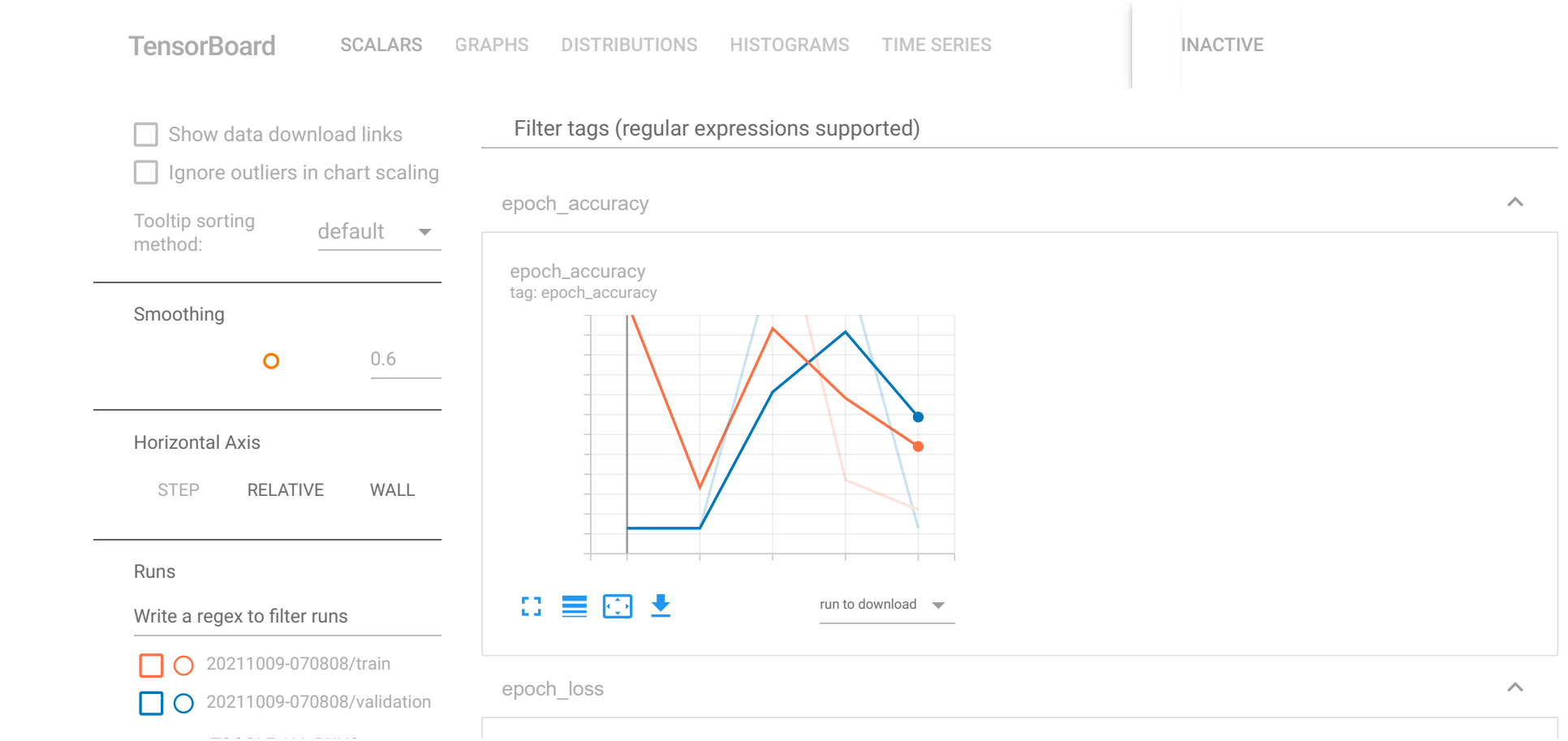
```
layer5 = Dense(32, activation='relu', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer4)


output = Dense(1, activation='sigmoid', kernel_initializer=tf.keras.initializers.random_uniform(0, 1))(layer5)

model2 = Model(inputs = input_layer, outputs = output)
optimizer = tf.keras.optimizers.SGD(learning_rate=0.1, momentum=0.9, nesterov=True)

model2.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

log_dir="logs/fit/model2/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True, write_grads=True)

filepath = "model_save/weights-{epoch:02d}-{val_accuracy:.4f}.hdf5"

checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
reduce_lr = ReduceLROnPlateau(monitor = 'val_accuracy', factor=0.9, patience=1, verbose=1, mode='auto')

terminate_tr = TerminateOnNaN()
earlystop = EarlyStopping(monitor='val_accuracy', patience=2, verbose=1, mode='max')

lrschedule = LearningRateScheduler(changedLearningRate, verbose=1)
model2.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), validation_split=0.2, callbacks=[checkpoint, reduce_lr, terminate_tr,

    WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the `TensorBoard` Callback.
    Epoch 1/10

    Epoch 00001: LearningRateScheduler setting learning rate to 0.100000000149011612.
      3/335 [..............................] - ETA: 37s - loss: 271526528.0000 - accuracy: 0.5000WARNING:tensorflow:Callback method `on_train_batch_b
    WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared to the batch time (batch time: 0.0039s vs `on_train_batch_end` time: 0.0
    335/335 [==============================] - 2s 6ms/step - loss: 2431580.7500 - accuracy: 0.5023 - val_loss: 0.6955 - val_accuracy: 0.4966

    Epoch 00001: val_accuracy improved from -inf to 0.49664, saving model to model_save/weights-01-0.4966.hdf5
    val_f1: 0.000000 — val_precision: 0.000000 — val_recall 0.000000 — val_auc 0.500000
    Epoch 2/10

    Epoch 00002: LearningRateScheduler setting learning rate to 0.100000000149011612.
     16/335 [>.............................] - ETA: 1s - loss: 0.6947 - accuracy: 0.4824/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_class
      _warn_prf(average, modifier, msg_start, len(result))
    335/335 [==============================] - 2s 4ms/step - loss: 0.6949 - accuracy: 0.4949 - val_loss: 0.6958 - val_accuracy: 0.4966

    Epoch 00002: val_accuracy did not improve from 0.49664

    Epoch 00002: ReduceLROnPlateau reducing learning rate to 0.090000000134110452.
    val_f1: 0.000000 — val_precision: 0.000000 — val_recall 0.000000 — val_auc 0.500000
    Epoch 3/10

    Epoch 00003: LearningRateScheduler setting learning rate to 0.085500000339746475.
     15/335 [>.............................] - ETA: 1s - loss: 0.6925 - accuracy: 0.5271/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_class
      _warn_prf(average, modifier, msg_start, len(result))
```

```
335/335 [==============================] - 1s 4ms/step - loss: 0.6943 - accuracy: 0.5055 - val_loss: 0.6931 - val_accuracy: 0.5034

Epoch 00003: val_accuracy improved from 0.49664 to 0.50336, saving model to model_save/weights-03-0.5034.hdf5
val_f1: 0.664238 — val_precision: 0.497273 — val_recall 1.000000 — val_auc 0.500000
Epoch 4/10

Epoch 00004: LearningRateScheduler setting learning rate to 0.08550000190734863.
335/335 [==============================] - 1s 4ms/step - loss: 0.6944 - accuracy: 0.4979 - val_loss: 0.6931 - val_accuracy: 0.5034

Epoch 00004: val_accuracy did not improve from 0.50336

Epoch 00004: ReduceLROnPlateau reducing learning rate to 0.07695000171661377.
val_f1: 0.664238 — val_precision: 0.497273 — val_recall 1.000000 — val_auc 0.500000
Epoch 5/10

Epoch 00005: LearningRateScheduler setting learning rate to 0.07694999873638153.
335/335 [==============================] - 1s 4ms/step - loss: 0.6944 - accuracy: 0.4971 - val_loss: 0.6961 - val_accuracy: 0.4966

Epoch 00005: val_accuracy did not improve from 0.50336

Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.06925499886274337.
val_f1: 0.000000 — val_precision: 0.000000 — val_recall 0.000000 — val_auc 0.500000
Epoch 00005: early stopping
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: UndefinedMetricWarning: Precision is ill-defined and being set to
  _warn_prf(average, modifier, msg_start, len(result))
<keras.callbacks.History at 0x7fa011cd4a50>
```

```
!kill 198
```

```
/bin/bash: line 0: kill: (198) - No such process
```

```
%tensorboard --logdir logs/fit/model2/
```



Model 3

```
input_layer = Input(shape=(2, ))

layer1 = Dense(784, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(input_layer)
layer2 = Dense(512, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer1)
layer3 = Dense(256, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer2)
layer4 = Dense(64, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer3)
layer5 = Dense(32, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer4)


output = Dense(1, activation='sigmoid', kernel_initializer=tf.keras.initializers.HeUniform)(layer5)

model3 = Model(inputs = input_layer, outputs = output)
optimizer = tf.keras.optimizers.SGD(learning_rate=0.1, momentum=0.9, nesterov=True)
```

```
model3.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

log_dir="logs/fit/model3/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True, write_grads=True)

filepath = "model_save/weights-{epoch:02d}-{val_accuracy:.4f}.hdf5"

checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
reduce_lr = ReduceLROnPlateau(monitor = 'val_accuracy', factor=0.9, patience=1, verbose=1, mode='auto')

terminate_tr = TerminateOnNaN()
earlystop = EarlyStopping(monitor='val_accuracy', patience=2, verbose=1, mode='max')

lrschedule = LearningRateScheduler(changedLearningRate, verbose=1)
model3.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), validation_split=0.2, callbacks=[checkpoint, reduce_lr, terminate_tr,

    Epoch 00003: LearningRateScheduler setting learning rate to 0.09500000141561031.
    335/335 [==============================] - 1s 4ms/step - loss: 0.6535 - accuracy: 0.6154 - val_loss: 0.7362 - val_accuracy: 0.5381

    Epoch 00003: val_accuracy did not improve from 0.58843

    Epoch 00003: ReduceLROnPlateau reducing learning rate to 0.08549999892711639.
    val_f1: 0.679632 — val_precision: 0.520311 — val_recall 0.979586 — val_auc 0.543138
    Epoch 4/10

    Epoch 00004: LearningRateScheduler setting learning rate to 0.08550000190734863.
    335/335 [==============================] - 1s 4ms/step - loss: 0.6412 - accuracy: 0.6327 - val_loss: 0.6318 - val_accuracy: 0.6455

    Epoch 00004: val_accuracy improved from 0.58843 to 0.64552, saving model to model_save/weights-04-0.6455.hdf5
    val_f1: 0.665851 — val_precision: 0.668407 — val_recall 0.663315 — val_auc 0.668909
    Epoch 5/10

    Epoch 00005: LearningRateScheduler setting learning rate to 0.08550000190734863.
    335/335 [==============================] - 1s 4ms/step - loss: 0.6328 - accuracy: 0.6367 - val_loss: 0.6254 - val_accuracy: 0.6504

    Epoch 00005: val_accuracy improved from 0.64552 to 0.65037, saving model to model_save/weights-05-0.6504.hdf5
    val_f1: 0.688681 — val_precision: 0.653709 — val_recall 0.727605 — val_auc 0.673176
    Epoch 6/10

    Epoch 00006: LearningRateScheduler setting learning rate to 0.0812250018119812.
    335/335 [==============================] - 1s 4ms/step - loss: 0.6284 - accuracy: 0.6466 - val_loss: 0.6320 - val_accuracy: 0.6478

    Epoch 00006: val_accuracy did not improve from 0.65037

    Epoch 00006: ReduceLROnPlateau reducing learning rate to 0.07310250028967857.
    val_f1: 0.618842 — val_precision: 0.715142 — val_recall 0.545399 — val_auc 0.665255
    Epoch 7/10

    Epoch 00007: LearningRateScheduler setting learning rate to 0.07310250401496887.
```

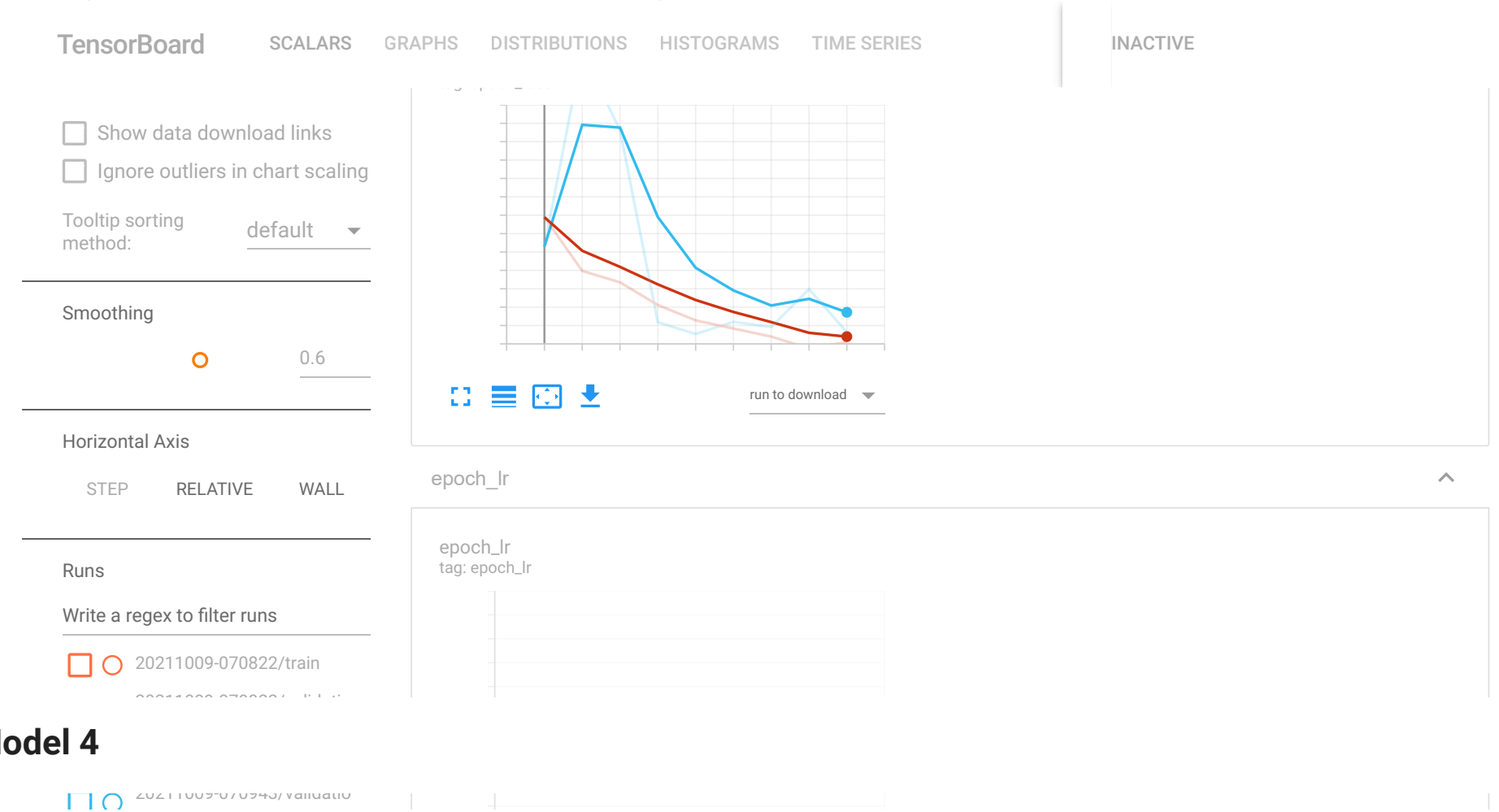
```
335/335 [==============================] - 1s 4ms/step - loss: 0.6240 - accuracy: 0.6566 - val_loss: 0.6292 - val_accuracy: 0.6541

Epoch 00007: val_accuracy improved from 0.65037 to 0.65410, saving model to model_save/weights-07-0.6541.hdf5
val_f1: 0.663225 — val_precision: 0.681453 — val_recall 0.645948 — val_auc 0.673637
Epoch 8/10

Epoch 00008: LearningRateScheduler setting learning rate to 0.07310250401496887.
335/335 [==============================] - 1s 4ms/step - loss: 0.6175 - accuracy: 0.6563 - val_loss: 0.6498 - val_accuracy: 0.5873

Epoch 00008: val_accuracy did not improve from 0.65410

Epoch 00008: ReduceLROnPlateau reducing learning rate to 0.06579225361347199.
val_f1: 0.430323 — val_precision: 0.802161 — val_recall 0.294028 — val_auc 0.611149
Epoch 9/10

Epoch 00009: LearningRateScheduler setting learning rate to 0.0625026423484087.
335/335 [==============================] - 1s 4ms/step - loss: 0.6209 - accuracy: 0.6587 - val_loss: 0.6264 - val_accuracy: 0.6511

Epoch 00009: val_accuracy did not improve from 0.65410

Epoch 00009: ReduceLROnPlateau reducing learning rate to 0.05625238046050072.
val_f1: 0.643550 — val_precision: 0.698537 — val_recall 0.596587 — val_auc 0.670958
Epoch 00009: early stopping
<keras.callbacks.History at 0x7fa0e4de73d0>
```

```
!kill 197
```

```
/bin/bash: line 0: kill: (197) - No such process
```

```
%tensorboard --logdir logs/fit/model3/
```

```
Reusing TensorBoard on port 6008 (pid 380), started 0:01:31 ago. (Use '!kill 380' to kill it.)
```



Model 4

```
input_layer = Input(shape=(2, ))

layer1 = Dense(784, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(input_layer)
layer2 = Dense(512, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer1)
layer3 = Dense(256, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer2)
layer4 = Dense(64, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer3)
layer5 = Dense(32, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer4)
# layer6 = Dense(16, activation='relu', kernel_initializer=tf.keras.initializers.HeUniform)(layer5)


output = Dense(1, activation='sigmoid', kernel_initializer=tf.keras.initializers.HeUniform)(layer5)

model4 = Model(inputs=input_layer, outputs=output)
```

```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.01, beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=False, name="Adam",)

model4.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

log_dir="logs/fit/model4/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True, write_grads=True)

filepath = "model_save/weights-{epoch:02d}-{val_accuracy:.4f}.hdf5"

checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.9, patience=1, verbose=1, mode='auto')

terminate_tr = TerminateOnNaN()
earlystop = EarlyStopping(monitor='val_accuracy', patience=2, verbose=1, mode='max')

lrschedule = LearningRateScheduler(changedLearningRate, verbose=1)
model4.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), validation_split=0.2, callbacks=[checkpoint, reduce_lr, terminate_tr,

    w 2.0 for the `TensorBoard` Callback.


    .009999999776482582.
    .4966 - accuracy: 0.5312WARNING:tensorflow:Callback method `on_train_batch_begin` is slow compared to the batch time (batch time: 0.0050s vs `o
    ow compared to the batch time (batch time: 0.0050s vs `on_train_batch_end` time: 0.0196s). Check your callbacks.
    : 0.7274 - accuracy: 0.5904 - val_loss: 0.6610 - val_accuracy: 0.5955

    ng model to model_save/weights-01-0.5955.hdf5
    056 — val_auc 0.579095


    .009999999776482582.
    : 0.6340 - accuracy: 0.6368 - val_loss: 0.6334 - val_accuracy: 0.6466

    aving model to model_save/weights-02-0.6466.hdf5
    910 — val_auc 0.633827


    .009499999787658453.
    : 0.6326 - accuracy: 0.6509 - val_loss: 0.6178 - val_accuracy: 0.6638

    aving model to model_save/weights-03-0.6638.hdf5
    745 — val_auc 0.673258


    .009499999694526196.
    : 0.6119 - accuracy: 0.6605 - val_loss: 0.6070 - val_accuracy: 0.6728

    aving model to model_save/weights-04-0.6728.hdf5
    916 — val_auc 0.684422
```

```
.009499999694526196.
: 0.6159 - accuracy: 0.6558 - val_loss: 0.6145 - val_accuracy: 0.6731

aving model to model_save/weights-05-0.6731.hdf5
602 — val_auc 0.664810


.009024999709799886.
: 0.6092 - accuracy: 0.6591 - val_loss: 0.6104 - val_accuracy: 0.6660



8122500032186508.
553 — val_auc 0.674970


.008122500032186508.
: 0.6084 - accuracy: 0.6582 - val_loss: 0.6056 - val_accuracy: 0.6724



73102500289678575.
721 — val_auc 0.678912
```

```
!kill 197
```

```
 /bin/bash: line 0: kill: (197) - No such process
```

```
%tensorboard --logdir logs/fit/model4/
```

```
Reusing TensorBoard on port 6006 (pid 378), started 0:01:07 ago. (Use '!kill 378' to kill it.)
```

TensorBoard SCALARS GRAPHS DISTRIBUTIONS HISTOGRAMS TIME SERIES INACTIVE

Show data download links

Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

0.6

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

20211009-093418/train

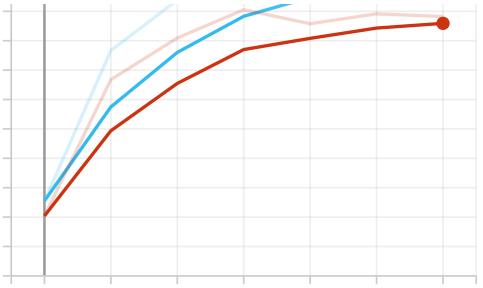
20211009-093418/validation

20211009-093532/train

20211009-093532/validation

TOGGLE ALL RUNS

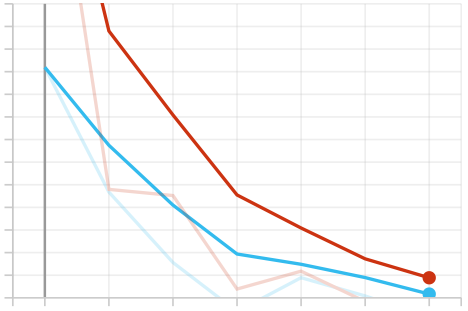
logs/fit/model4/



run to download

epoch_loss

epoch_loss
tag: epoch_loss



run to download

epoch_lr

epoch_lr
tag: epoch_lr