# keras-callback

August 2, 2023

[ ]:
```
from google.colab import files
files=files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving data.csv to data.csv
```

[ ]:
```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
from keras.callbacks import Callback
from sklearn.metrics import roc_auc_score, f1_score
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import LearningRateScheduler
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import ReduceLROnPlateau
import os
import datetime
```

[ ]:
```
data=pd.read_csv("data.csv")
```

[ ]:
```
count_val=list(data["label"])
count_val.count(0)
count_val.count(1)
```

[ ]:
```
10000
```

[ ]:
```
x=(data[['f1','f2']]).to_numpy()
y=(data['label']).to_numpy()
```

[ ]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
validation_data=x_test,y_test
```

[ ]:
```
class performanceMetric(tf.keras.callbacks.Callback):
  def __init__(self):
    self.x_test = validation_data[0]
```

```
    self.y_test= validation_data[1]
  def on_epoch_end(self,epoch,logs={}):
    y_predict=self.model.predict(self.x_test)
    f1_score_val=f1_score(y_test,y_predict,average='micro')
    roc_score=roc_auc_score(y_test,y_predict)
    print("F1 score is :",f1_score_val," ROC_AUC score :",roc_score)
```

[ ]:
```
# class performance_metric(tf.ks.callbacks.Callback):
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=self.model.predict(self.validation_data[0])
    #  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(),average='micro')
     roc_val=roc_auc_score(val_targ, val_predict)
    #  self.val_f1s.append(val_f1)
     print("-f1 score :",val_f1 ,"-ROCValue :", roc_val)
```

[ ]:
```
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 {}".format(epoch))
                self.model.stop_training = True
        model_weights = self.model.get_weights()
        if model_weights is not None:
          if np.any([np.any(np.isnan(x)) for x in model_weights]):
            self.model.stop_training = True
```

[ ]:
```
def changeLearningRate(epoch, learning_rate):
    #here we are performing exponential decay of the learning rate
    if (epoch+1)%3==0:
      learning_rate=learning_rate*0.95
    return learning_rate
```

[ ]:
```
def model_1():
  return tf.keras.models.Sequential([tf.keras.layers.
 ↪Dense(2,activation="tanh",input_shape=(2,),kernel_initializer=keras.
 ↪initializers.RandomUniform(minval=0, maxval=1)),
```

```
                                    tf.keras.layers.Dense(8,␣
 ↪activation="tanh",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="tanh",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf.keras.layers.Dense(32,␣
 ↪activation="tanh",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="tanh",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf. keras.layers.Dense(8,␣
 ↪activation="tanh",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf.keras.layers.Dense(1,␣
 ↪activation='softmax',kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1))])
  # return model
```

[ ]:
```
filepath="model_save/weights-{epoch:02d}-{val_accuracy:.4f}.hdf5"
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.9, patience=2,␣
 ↪min_lr=(1*10^-4))
lrschedule = LearningRateScheduler(changeLearningRate, verbose=0)
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', ␣
 ↪verbose=1, save_best_only=True, mode='auto')
earlystop = EarlyStopping(monitor='val_accuracy', min_delta=0.35, patience=3,␣
 ↪verbose=1)
terminate= TerminateNaN()
metrics=performanceMetric()
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)
```

[ ]:
```
model1=model_1()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.1,␣
 ↪nesterov=False, name='SGD')        #https://www.tensorflow.org/api_docs/
 ↪python/tf/keras/optimizers/SGD
model1.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model1.fit(x=x_train, y=y_train, epochs=10,validation_data=(x_test,␣
 ↪y_test),callbacks=[metrics,checkpoint,terminate,lrschedule␣
 ↪,reduce_lr,tensorboard_callback,earlystop ])
# model1.fit(x=x_train, y=y_train, epochs=15,validation_data=(x_test,␣
 ↪y_test),callbacks=[checkpoint,metrics,terminate,reduce_lr,tensorboard_callback,lrschedule,e
 ↪])
```

```
Epoch 1/10
```

```
  1/469 […] - ETA: 8:26 - loss: 2.1677 - accuracy:
0.5625

WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared to the
batch time (batch time: 0.0029s vs `on_train_batch_end` time: 0.0082s). Check
your callbacks.

464/469 [============================>.] - ETA: 0s - loss: 0.8137 - accuracy:
0.4972F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 1: val_accuracy did not improve from 0.50720
469/469 [==============================] - 4s 7ms/step - loss: 0.8125 -
accuracy: 0.4976 - val_loss: 0.6931 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 2/10
462/469 [============================>.] - ETA: 0s - loss: 0.6934 - accuracy:
0.4976F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 2: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 3ms/step - loss: 0.6934 -
accuracy: 0.4976 - val_loss: 0.6931 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 3/10
451/469 [============================>..] - ETA: 0s - loss: 0.6933 - accuracy:
0.4978F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 3: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6933 -
accuracy: 0.4976 - val_loss: 0.6933 - val_accuracy: 0.5072 - lr: 0.0095
Epoch 4/10
447/469 [============================>..] - ETA: 0s - loss: 0.6933 - accuracy:
0.4969F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 4: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6933 -
accuracy: 0.4976 - val_loss: 0.6937 - val_accuracy: 0.5072 - lr: 0.0085
Epoch 4: early stopping
```

[ ]:
```
<keras.callbacks.History at 0x7efe1c7ca5d0>
```

[ ]:
```
# %tensorboard --logdir logs
%load_ext tensorboard
%tensorboard --logdir logs
```

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

Reusing TensorBoard on port 6006 (pid 199), started 0:34:33 ago. (Use '!kill␣
↪199' to kill it.)

<IPython.core.display.Javascript object>
```

[ ]:
```
!rm -rf ./logs/
```

[ ]:
```
def model_2():
  return tf.keras.models.Sequential([tf.keras.layers.
 ↪Dense(2,activation="relu",input_shape=(2,),kernel_initializer=keras.
 ↪initializers.RandomUniform(minval=0, maxval=1)),
                                    tf.keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf.keras.layers.Dense(32,␣
 ↪activation="relu",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1)),
                                    tf.keras.layers.Dense(1,␣
 ↪activation='softmax',kernel_initializer=keras.initializers.
 ↪RandomUniform(minval=0, maxval=1))])
  # return model
```

[ ]:
```
model2=model_2()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.2,␣
 ↪nesterov=False, name='SGD')        #https://www.tensorflow.org/api_docs/
 ↪python/tf/keras/optimizers/SGD
model2.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model2.fit(x=x_train, y=y_train, epochs=10,validation_data=(x_test,␣
 ↪y_test),callbacks=[metrics,checkpoint,terminate,lrschedule␣
 ↪,reduce_lr,tensorboard_callback,earlystop ])
```

```
Epoch 1/10
  1/469 […] - ETA: 5:38 - loss: 8003.9609 -
accuracy: 0.5625

WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared to the
batch time (batch time: 0.0016s vs `on_train_batch_end` time: 0.0033s). Check
your callbacks.

457/469 [============================>.] - ETA: 0s - loss: 18.2058 - accuracy:
0.4972F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 1: val_accuracy did not improve from 0.50720
469/469 [==============================] - 3s 4ms/step - loss: 17.7668 -
```

```
accuracy: 0.4976 - val_loss: 0.6932 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 2/10
467/469 [============================>.] - ETA: 0s - loss: 0.6932 - accuracy:
0.4971F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 2: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6932 -
accuracy: 0.4976 - val_loss: 0.6932 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 3/10
467/469 [============================>.] - ETA: 0s - loss: 0.6932 - accuracy:
0.4982F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 3: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6932 -
accuracy: 0.4976 - val_loss: 0.6933 - val_accuracy: 0.5072 - lr: 0.0095
Epoch 4/10
461/469 [============================>.] - ETA: 0s - loss: 0.6932 - accuracy:
0.4974F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 4: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6932 -
accuracy: 0.4976 - val_loss: 0.6932 - val_accuracy: 0.5072 - lr: 0.0085
Epoch 4: early stopping
```

[ ]:
```
<keras.callbacks.History at 0x7efe1b534810>
```

[ ]:
```
%tensorboard --logdir logs
```

```
Reusing TensorBoard on port 6006 (pid 199), started 0:35:32 ago. (Use '!kill␣
 ↪199' to kill it.)

<IPython.core.display.Javascript object>
```

[ ]:
```
!rm -rf ./logs/
```

[ ]:
```
def model_3():
  return tf.keras.models.Sequential([tf.keras.layers.
 ↪Dense(2,activation="relu",input_shape=(2,),kernel_initializer=keras.
 ↪initializers.he_uniform()),
                                    tf.keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.he_uniform()),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.he_uniform()),
                                    tf.keras.layers.Dense(32,␣
 ↪activation="relu",kernel_initializer=keras.initializers.he_uniform()),
                                    tf. keras.layers.Dense(16,␣
 ↪activation="relu",kernel_initializer=keras.initializers.he_uniform()),
```

```
                                    tf. keras.layers.Dense(16,␣
↪activation="relu",kernel_initializer=keras.initializers.he_uniform()),
                                    tf.keras.layers.Dense(1,␣
↪activation='softmax',kernel_initializer=keras.initializers.he_uniform())])
 # return model
```

[ ]:
```
model3=model_3()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.15,␣
↪nesterov=False, name='SGD')        #https://www.tensorflow.org/api_docs/
↪python/tf/keras/optimizers/SGD
model1.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model1.fit(x=x_train, y=y_train, epochs=10,validation_data=(x_test,␣
↪y_test),callbacks=[metrics,checkpoint,terminate,lrschedule␣
↪,reduce_lr,tensorboard_callback,earlystop ])
```

```
Epoch 1/10
  1/469 […] - ETA: 5:26 - loss: 0.6918 - accuracy:
0.4688

WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared to the
batch time (batch time: 0.0023s vs `on_train_batch_end` time: 0.0031s). Check
your callbacks.

459/469 [============================>.] - ETA: 0s - loss: 0.6933 - accuracy:
0.4984F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 1: val_accuracy did not improve from 0.50720
469/469 [==============================] - 3s 4ms/step - loss: 0.6933 -
accuracy: 0.4976 - val_loss: 0.6934 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 2/10
455/469 [============================>.] - ETA: 0s - loss: 0.6933 - accuracy:
0.4976F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 2: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6933 -
accuracy: 0.4976 - val_loss: 0.6936 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 3/10
454/469 [============================>.] - ETA: 0s - loss: 0.6932 - accuracy:
0.4970F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 3: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6932 -
accuracy: 0.4976 - val_loss: 0.6932 - val_accuracy: 0.5072 - lr: 0.0095
Epoch 4/10
454/469 [============================>.] - ETA: 0s - loss: 0.6933 - accuracy:
0.4980F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 4: val_accuracy did not improve from 0.50720
```

```
469/469 [==============================] - 2s 4ms/step - loss: 0.6933 -
accuracy: 0.4976 - val_loss: 0.6932 - val_accuracy: 0.5072 - lr: 0.0085
Epoch 4: early stopping
```

[ ]:
```
<keras.callbacks.History at 0x7efe1094e6d0>
```

[ ]:
```
%tensorboard --logdir logs
```

```
Reusing TensorBoard on port 6006 (pid 199), started 0:49:52 ago. (Use '!kill␣
↪199' to kill it.)
```

```
<IPython.core.display.Javascript object>
```

[ ]:
```
!rm -rf ./logs/
```

[ ]:
```
def model_4():
  return tf.keras.models.Sequential([tf.keras.layers.
↪Dense(2,activation="relu",input_shape=(2,),kernel_initializer=keras.
↪initializers.he_uniform()),
                                   tf.keras.layers.Dense(16,␣
↪activation="relu",kernel_initializer=keras.initializers.he_normal()),
                                   tf. keras.layers.Dense(32,␣
↪activation="relu",kernel_initializer=keras.initializers.he_normal()),
                                   tf.keras.layers.Dense(64,␣
↪activation="relu",kernel_initializer=keras.initializers.he_normal()),
                                   tf. keras.layers.Dense(32,␣
↪activation="relu",kernel_initializer=keras.initializers.he_normal()),
                                   tf. keras.layers.Dense(16,␣
↪activation="relu",kernel_initializer=keras.initializers.he_normal()),
                                   tf.keras.layers.Dense(1,␣
↪activation='softmax',kernel_initializer=keras.initializers.he_normal())])
```

[ ]:
```
model4=model_4()
optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.5,␣
↪nesterov=True, name='SGD')        #https://www.tensorflow.org/api_docs/
↪python/tf/keras/optimizers/SGD
model4.compile(optimizer,loss='BinaryCrossentropy',metrics=['accuracy'])
model4.fit(x=x_train, y=y_train, epochs=10,validation_data=(x_test,␣
↪y_test),callbacks=[metrics,checkpoint,terminate,lrschedule␣
↪,reduce_lr,tensorboard_callback,earlystop ])
```

```
Epoch 1/10
  1/469 […] - ETA: 7:49 - loss: 0.7184 - accuracy:
0.3438
```

```
WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared to the
batch time (batch time: 0.0015s vs `on_train_batch_end` time: 0.0041s). Check
your callbacks.
```

```
465/469 [============================>.] - ETA: 0s - loss: 0.6849 - accuracy:
0.4981F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 1: val_accuracy did not improve from 0.50720
469/469 [==============================] - 3s 4ms/step - loss: 0.6847 -
accuracy: 0.4976 - val_loss: 0.6943 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 2/10
462/469 [============================>.] - ETA: 0s - loss: 0.6762 - accuracy:
0.4970F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 2: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6761 -
accuracy: 0.4976 - val_loss: 0.6745 - val_accuracy: 0.5072 - lr: 0.0100
Epoch 3/10
460/469 [============================>.] - ETA: 0s - loss: 0.6708 - accuracy:
0.4980F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 3: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 3ms/step - loss: 0.6709 -
accuracy: 0.4976 - val_loss: 0.6706 - val_accuracy: 0.5072 - lr: 0.0095
Epoch 4/10
446/469 [============================>..] - ETA: 0s - loss: 0.6677 - accuracy:
0.4978F1 score is : 0.5072  ROC_AUC score : 0.5

Epoch 4: val_accuracy did not improve from 0.50720
469/469 [==============================] - 2s 4ms/step - loss: 0.6677 -
accuracy: 0.4976 - val_loss: 0.6694 - val_accuracy: 0.5072 - lr: 0.0085
Epoch 4: early stopping
```

[ ]:
```
<keras.callbacks.History at 0x7efe13ad67d0>
```

[ ]:
```
%tensorboard --logdir logs
```

```
Reusing TensorBoard on port 6006 (pid 199), started 0:41:27 ago. (Use '!kill␣
↪199' to kill it.)

<IPython.core.display.Javascript object>
```

[ ]: