custom_callback_final

March 22, 2021

```
[ ]: # !pip install --upgrade tensorflow==2.2.0
```

```
[1]: from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense
     from tensorflow.keras.utils import to_categorical
     from sklearn.model_selection import train_test_split
     import numpy as np
     import tensorflow as tf
     import pandas as pd
     import matplotlib.pyplot as plt
     from sklearn import metrics
     from tensorflow.keras.optimizers import SGD
     from tensorflow.keras.callbacks import LearningRateScheduler, ReduceLROnPlateau
     import tensorflow.keras.callbacks as cbks
     from tensorflow.keras.models import load_model
     import os

     import warnings
     warnings.filterwarnings("ignore")

     # Configuration options
     feature_vector_length = 2
     num_classes = 1
     %load_ext tensorboard
```

0.1 1. Feature Engineering

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

     data['label'] = data['label'].astype('int32')

     data['2f1'] = data['f1'] * 2
     data['f1^2'] = data['f1'] ** 2
     data['sqrt_f1'] = np.sqrt(data['f1'])
     data['log_f1'] = np.log(data['f1'])
     data['exp_f1'] = np.exp(data['f1'])
     data['sin_f1'] = np.sin(data['f1'])
     data['tan_f1'] = np.tan(data['f1'])
```

```
data['2f2'] = data['f2'] * 2
data['f2^2'] = data['f2'] ** 2
data['sqrt_f2'] = np.sqrt(data['f2'])
data['log_f2'] = np.log(data['f2'])
data['exp_f2'] = np.exp(data['f2'])
data['sin_f2'] = np.sin(data['f2'])
data['tan_f2'] = np.tan(data['f2'])

data['f1f2'] = data['f1'] * data['f2']
data['2_f1f2'] = data['2f1'] * data['2f2']
data['sqrt_f1f2'] = data['sqrt_f1'] * data['sqrt_f2']
data['2_log_f1f2'] = data['log_f1'] * data['log_f2']
data['2_f1^2'] = data['f1^2'] * data['f2^2']
data['tan_f1f2'] = data['tan_f1'] * data['tan_f2']
data['sin_f1f2'] = data['sin_f1'] * data['sin_f2']
data['exp_f1f2'] = data['exp_f1'] * data['exp_f2']

data.fillna(0, inplace=True)

data.head()
```

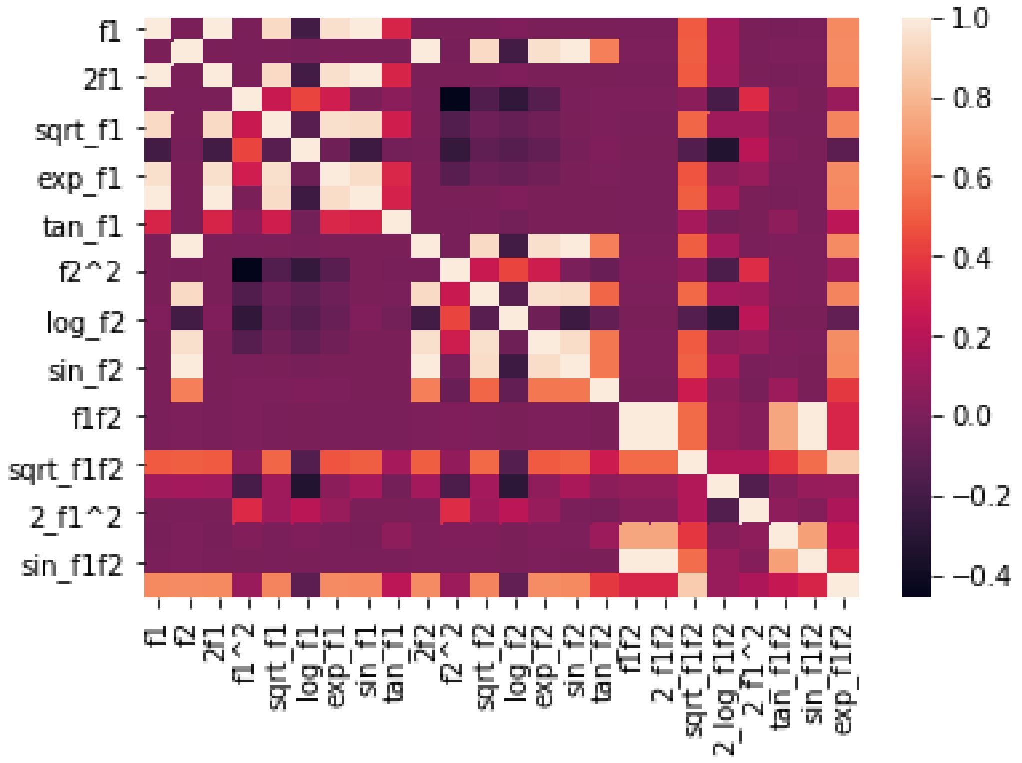
[2]:
```
         f1        f2  label       2f1  …    2_f1^2  tan_f1f2  sin_f1f2
exp_f1f2
0  0.450564  1.074305      0  0.901127  …  0.234297  0.892931  0.382894
4.594539
1  0.085632  0.967682      0  0.171263  …  0.006866  0.124639  0.070438
2.867135
2  0.117326  0.971521      1  0.234652  …  0.012992  0.172554  0.096659
2.970845
3  0.982179 -0.380408      0  1.964358  …  0.139599 -0.599032 -0.308813
1.825348
4 -0.720352  0.955850      0 -1.440705  …  0.474099 -1.242647 -0.538804
1.265538

[5 rows x 25 columns]
```

0.2 2. Data Splitting

[3]:
```
target = data['label']

data = data.drop(['label'], axis=1)

X_train, X_test, Y_train, Y_test = train_test_split(data, target, test_size=0.
 ↪20, random_state=3, shuffle=True, stratify=target)

print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(16000, 24) (4000, 24) (16000,) (4000,)
```

[4]:
```python
import seaborn as sns

sns.heatmap(data.corr());
```



0.3 3. Callbacks

[33]:
```python
from sklearn.metrics import roc_auc_score
from keras.callbacks import Callback
from datetime import datetime

''' AUC Ref "" https://stackoverflow.com/questions/41032551/
how-to-compute-receiving-operating-characteristic-roc-and-auc-in-keras'''

class RocCallback(Callback):
    def __init__(self,training_data,validation_data):
        self.x = training_data[0]
        self.y = training_data[1]
        self.x_val = validation_data[0]
        self.y_val = validation_data[1]
    def on_train_begin(self, logs={}):
        return

    def on_train_end(self, logs={}):
        return

    def on_epoch_begin(self, epoch, logs={}):
        return

    def on_epoch_end(self, epoch, logs={}):
        y_pred_train = self.model.predict_proba(self.x)
        roc_train = roc_auc_score(self.y, y_pred_train)
        y_pred_val = self.model.predict_proba(self.x_val)
        roc_val = roc_auc_score(self.y_val, y_pred_val)
        print('\rroc-auc_train: %s - roc-auc_val: %s' %␣
↪(str(round(roc_train,4)),str(round(roc_val,4))),end=100*' '+'\n')

        tf.summary.scalar('train auc', data=round(roc_train,4), step=epoch)
        tf.summary.scalar('valid auc', data=round(roc_val,4), step=epoch)
        print("\n", "#"*200, "\n")

        return

    def on_batch_begin(self, batch, logs={}):
        return

    def on_batch_end(self, batch, logs={}):
        return

roc = RocCallback(training_data=(X_train, Y_train),
                  validation_data=(X_test, Y_test))
```

[34]:
```
from keras import backend as K

def micro_f1(y_true, y_pred):
  def recall(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

  def precision(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision
  precision = precision(y_true, y_pred)
  recall = recall(y_true, y_pred)
  #2*pi*row / (pi + row): micro-f1
  micro_f1 = (2*precision*recall)/(precision+recall+K.epsilon())
  return micro_f1

def get_model(act_fn, initializer, metrics=None):
  # Set the input shape
  input_shape = (X_train.shape[1],)
  print(f'Feature shape: {input_shape}')
  # Create the model
  model = Sequential()
  model.add(Dense(50, input_shape=input_shape, activation=act_fn,␣
 ↪kernel_initializer=initializer))
  model.add(Dense(25, activation=act_fn))
  model.add(Dense(10, activation=act_fn))
  model.add(Dense(5, activation=act_fn))
  model.add(Dense(3, activation=act_fn))
  model.add(Dense(num_classes, activation='sigmoid'))
  sgd = SGD(lr=0.01, momentum=0.9)
  # Configure the model and start training
  model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=[micro_f1],␣
 ↪run_eagerly=True)

  return model
```

[35]:
```
class CustomCallback(tf.keras.callbacks.Callback):
    def __init__(self):
        self.val_acc = dict()
        self.lr = dict()
        self.val_loss = dict()

    def on_train_begin(self, logs=None):
        keys = list(logs.keys())

    def on_train_end(self, logs=None):
        keys = list(logs.keys())
        title="Learning Rate Schedule"
        epochs = list(self.lr.keys())
        lrs = list(self.lr.values())

        # the learning rate schedule
        plt.style.use("ggplot")
        plt.figure()
        plt.plot(epochs, lrs, color='green', marker='o', linestyle='dashed')
        plt.title(title)
        plt.xlabel("Epoch #")
        plt.ylabel("Learning Rate")
        plt.savefig("lrng_rate.png")
        print("Stop training; got log keys: {}".format(keys))

    def on_epoch_begin(self, epoch, logs=None):
        keys = list(logs.keys())

    def on_epoch_end(self, epoch, logs=None):
        keys = list(logs.keys())
        lr_flag = False
        print("\nEnd epoch {} of training; got log keys: {}".format(epoch,␣
↪logs))

        try :
          if logs['val_micro_f1']:
            self.val_acc[str(epoch)] = logs['val_micro_f1']
            val_loss = logs['val_loss']
            self.val_loss[epoch] = val_loss
          elif logs['AUC_custom']:
            self.val_acc[str(epoch)] = logs['val_AUC_custom']
            val_loss = logs['val_loss']
            self.val_loss[epoch] = val_loss

        except:
          pass

        if epoch>=1:
          #learning rate reduce if prev_val_acc > next_val_acc epoch drop 10%
          if self.val_acc[str(epoch-1)] > self.val_acc[str(epoch)]:
            lr_flag = True
            lr = float(tf.keras.backend.get_value(self.model.optimizer.
↪learning_rate))
            scheduled_lr = lr * 0.90
            tf.keras.backend.set_value(self.model.optimizer.lr, scheduled_lr)

            print(f"\n custom learning rate by val_accracy if prev_val_acc >␣
↪next_val_acc epoch drop 10% : {lr}\n")
            self.lr[str(epoch+1)]= lr

          #learning rate for every 3 epoch drop 5%
          elif ((epoch+1) % 3 == 0) and lr_flag==False:
            lr = float(tf.keras.backend.get_value(self.model.optimizer.
↪learning_rate))
            schedule = float(lr * 0.95)
            tf.keras.backend.set_value(self.model.optimizer.lr, schedule)
            print(f"\n custom learning rate by every 3rd epoch {lr}\n")
            self.lr[str(epoch+1)]= lr
          #save model if prev_val_acc < next_val_acc BEST_MODEL = TRUE
          elif self.val_acc[str(epoch-1)] < self.val_acc[str(epoch)]:
            try:
              os.makedirs(("/content/best_model"))
            except:
              pass

            name = '/content/best_model/weights%08d.h5' % epoch

            # Record the best weights if current results is better.
            model = self.model.get_weights()
            print("\n", "#"*100, "\n", "Best Weights Saved : ", "\n", "#"*100)
            # model = load_model(model)
            self.model.save(name)

            # self.model.savefig("/content/best_model/best_model.png")
            with open("/content/best_model/best_weights.txt", "w+") as f:
              f.write(str(model))

          #if val loss is nan, stop_training
          elif val_loss != 'int' or val_loss != 'float':
            print("\n", "#"*100, "\n", "val loss is nan - Train Termination␣
↪Execcuted: ", "\n", "#"*100)
            self.model.stop_training = True

          # early stopping if val_acc is not improved for last 2 epochs
          elif epoch >= 2:
            val_loss = type(logs['val_loss'])
            if round(self.val_acc[str(epoch-1)] - self.val_acc[str(epoch-2)],␣
↪4) == 0:
              print("\n", "#"*100, "\n", "Early Stopping Executed: ", "\n",␣
↪"#"*100)
              self.model.stop_training = True

    def on_test_begin(self, logs=None):
        keys = list(logs.keys())

    def on_test_end(self, logs=None):
        keys = list(logs.keys())
        print("\nStop testing; got log keys: {}".format(logs))

    def on_predict_begin(self, logs=None):
        keys = list(logs.keys())

    def on_predict_end(self, logs=None):
        keys = list(logs.keys())
```

```
    def on_train_batch_begin(self, batch, logs=None):
        keys = list(logs.keys())

    def on_train_batch_end(self, batch, logs=None):
        keys = list(logs.keys())

    def on_test_batch_begin(self, batch, logs=None):
        keys = list(logs.keys())

    def on_test_batch_end(self, batch, logs=None):
        keys = list(logs.keys())

    def on_predict_batch_begin(self, batch, logs=None):
        keys = list(logs.keys())

    def on_predict_batch_end(self, batch, logs=None):
        keys = list(logs.keys())
```

0.4 4. Model

0.4.1 Model 1

[ ]:
```
from datetime import datetime
import os

logdir = "logs1/scalars/" + datetime.now().strftime("%Y%m%d-%H%M%S")

file_writer = tf.summary.create_file_writer(logdir + "/metrics")
file_writer.set_as_default()

tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir,␣
 ↪histogram_freq=1, write_graph=True, write_grads=True)
```

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

[ ]:
```
%tensorboard --logdir logs1
```

[ ]:
```
model1 = get_model('tanh', 'RandomUniform')

history_tanh = model1.fit(X_train, Y_train.ravel(), epochs=20, batch_size=250,
                          verbose=1, validation_split=0.2,
                          callbacks=[CustomCallback(), tensorboard_callback,␣
 ↪roc])

# Test the model after training
test_results = model1.evaluate(X_test, Y_test, verbose=1)
```

```
print(f'\nTest results - Loss: {test_results[0]} - micro_f1:␣
↪{test_results[1]}%')
```

```
Feature shape: (24,)
Epoch 1/20
 6/52 [==>…] - ETA: 4s - loss: 0.6980 - micro_f1:
0.4895WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared
to the batch time (batch time: 0.0239s vs `on_train_batch_end` time: 0.0324s).
Check your callbacks.
52/52 [==============================] - ETA: 0s - loss: 0.6822 - micro_f1:
0.5514
Stop testing; got log keys: {'loss': 0.6399805545806885, 'micro_f1':
0.531017541885376}
52/52 [==============================] - 5s 40ms/step - loss: 0.6819 - micro_f1:
0.5519 - val_loss: 0.6400 - val_micro_f1: 0.5310

End epoch 0 of training; got log keys: {'loss': 0.6662553548812866, 'micro_f1':
0.5798256397247314, 'val_loss': 0.6399805545806885, 'val_micro_f1':
0.531017541885376}
roc-auc_train: 0.7093 - roc-auc_val: 0.7167

 ###############################################################################
################################################################################
#########################################

Epoch 2/20
52/52 [==============================] - ETA: 0s - loss: 0.6309 - micro_f1:
0.6386
Stop testing; got log keys: {'loss': 0.6120874285697937, 'micro_f1':
0.6932922005653381}
52/52 [==============================] - 1s 25ms/step - loss: 0.6307 - micro_f1:
0.6389 - val_loss: 0.6121 - val_micro_f1: 0.6933

End epoch 1 of training; got log keys: {'loss': 0.6223677396774292, 'micro_f1':
0.6551369428634644, 'val_loss': 0.6120874285697937, 'val_micro_f1':
0.6932922005653381}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7288 - roc-auc_val: 0.7327

 ###############################################################################
################################################################################
#########################################
```

```
Epoch 3/20
51/52 [============================>.] - ETA: 0s - loss: 0.6136 - micro_f1:
0.6692
Stop testing; got log keys: {'loss': 0.6097702980041504, 'micro_f1':
0.6353364586830139}
52/52 [==============================] - 1s 26ms/step - loss: 0.6136 - micro_f1:
0.6689 - val_loss: 0.6098 - val_micro_f1: 0.6353

End epoch 2 of training; got log keys: {'loss': 0.6143127083778381, 'micro_f1':
0.6612235307693481, 'val_loss': 0.6097702980041504, 'val_micro_f1':
0.6353364586830139}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.009999999776482582

roc-auc_train: 0.7313 - roc-auc_val: 0.7342

 ###############################################################################
################################################################################
#########################################

Epoch 4/20
52/52 [==============================] - ETA: 0s - loss: 0.6079 - micro_f1:
0.6680
Stop testing; got log keys: {'loss': 0.6019672155380249, 'micro_f1':
0.6725978851318359}
52/52 [==============================] - 1s 26ms/step - loss: 0.6079 - micro_f1:
0.6680 - val_loss: 0.6020 - val_micro_f1: 0.6726

End epoch 3 of training; got log keys: {'loss': 0.6096696257591248, 'micro_f1':
0.6686173677444458, 'val_loss': 0.6019672155380249, 'val_micro_f1':
0.6725978851318359}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7366 - roc-auc_val: 0.7397

 ###############################################################################
################################################################################
#########################################

Epoch 5/20
52/52 [==============================] - ETA: 0s - loss: 0.6107 - micro_f1:
0.6599
```

```
Stop testing; got log keys: {'loss': 0.6105101108551025, 'micro_f1':
0.6820487380027771}
52/52 [==============================] - 1s 26ms/step - loss: 0.6107 - micro_f1:
0.6599 - val_loss: 0.6105 - val_micro_f1: 0.6820

End epoch 4 of training; got log keys: {'loss': 0.6093389987945557, 'micro_f1':
0.6606200337409973, 'val_loss': 0.6105101108551025, 'val_micro_f1':
0.6820487380027771}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7308 - roc-auc_val: 0.7329

 ################################################################################
################################################################################
#########################################

Epoch 6/20
52/52 [==============================] - ETA: 0s - loss: 0.6113 - micro_f1:
0.6684
Stop testing; got log keys: {'loss': 0.6145884990692139, 'micro_f1':
0.6078057289123535}
52/52 [==============================] - 1s 26ms/step - loss: 0.6112 - micro_f1:
0.6684 - val_loss: 0.6146 - val_micro_f1: 0.6078

End epoch 5 of training; got log keys: {'loss': 0.6070948243141174, 'micro_f1':
0.6664572358131409, 'val_loss': 0.6145884990692139, 'val_micro_f1':
0.6078057289123535}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.008999999612569809

roc-auc_train: 0.7357 - roc-auc_val: 0.7387

 ################################################################################
################################################################################
#########################################

Epoch 7/20
51/52 [============================>.] - ETA: 0s - loss: 0.6135 - micro_f1:
0.6449
Stop testing; got log keys: {'loss': 0.6056390404701233, 'micro_f1':
0.6299196481704712}
52/52 [==============================] - 1s 25ms/step - loss: 0.6133 - micro_f1:
0.6453 - val_loss: 0.6056 - val_micro_f1: 0.6299
```

```
End epoch 6 of training; got log keys: {'loss': 0.6093505024909973, 'micro_f1':
0.6543399095535278, 'val_loss': 0.6056390404701233, 'val_micro_f1':
0.6299196481704712}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7361 - roc-auc_val: 0.7408

 ###############################################################################
################################################################################
#########################################

Epoch 8/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6079 - micro_f1:
0.6588
Stop testing; got log keys: {'loss': 0.6044343113899231, 'micro_f1':
0.6791900396347046}
52/52 [==============================] - 1s 26ms/step - loss: 0.6079 - micro_f1:
0.6590 - val_loss: 0.6044 - val_micro_f1: 0.6792

End epoch 7 of training; got log keys: {'loss': 0.6069324612617493, 'micro_f1':
0.6637423634529114, 'val_loss': 0.6044343113899231, 'val_micro_f1':
0.6791900396347046}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7353 - roc-auc_val: 0.7377

 ###############################################################################
################################################################################
#########################################

Epoch 9/20
52/52 [==============================] - ETA: 0s - loss: 0.6072 - micro_f1:
0.6673
Stop testing; got log keys: {'loss': 0.6031242609024048, 'micro_f1':
0.6667574644088745}
52/52 [==============================] - 1s 26ms/step - loss: 0.6072 - micro_f1:
0.6672 - val_loss: 0.6031 - val_micro_f1: 0.6668

End epoch 8 of training; got log keys: {'loss': 0.6074584722518921, 'micro_f1':
```

```
0.6619762182235718, 'val_loss': 0.6031242609024048, 'val_micro_f1':
0.6667574644088745}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.008099999278783798

roc-auc_train: 0.7352 - roc-auc_val: 0.7386

 ################################################################################
################################################################################
#########################################

Epoch 10/20
52/52 [==============================] - ETA: 0s - loss: 0.6066 - micro_f1:
0.6698
Stop testing; got log keys: {'loss': 0.6040844917297363, 'micro_f1':
0.6463684439659119}
52/52 [==============================] - 1s 26ms/step - loss: 0.6066 - micro_f1:
0.6697 - val_loss: 0.6041 - val_micro_f1: 0.6464

End epoch 9 of training; got log keys: {'loss': 0.6053555011749268, 'micro_f1':
0.6638064980506897, 'val_loss': 0.6040844917297363, 'val_micro_f1':
0.6463684439659119}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.0072899991646409035

roc-auc_train: 0.7347 - roc-auc_val: 0.7392

 ################################################################################
################################################################################
#########################################

Epoch 11/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6077 - micro_f1:
0.6643
Stop testing; got log keys: {'loss': 0.6014505624771118, 'micro_f1':
0.6594324707984924}
52/52 [==============================] - 1s 26ms/step - loss: 0.6075 - micro_f1:
0.6644 - val_loss: 0.6015 - val_micro_f1: 0.6594

End epoch 10 of training; got log keys: {'loss': 0.604322612285614, 'micro_f1':
0.6655615568161011, 'val_loss': 0.6014505624771118, 'val_micro_f1':
0.6594324707984924}

 ################################################################################
#####################
 Best Weights Saved :
```

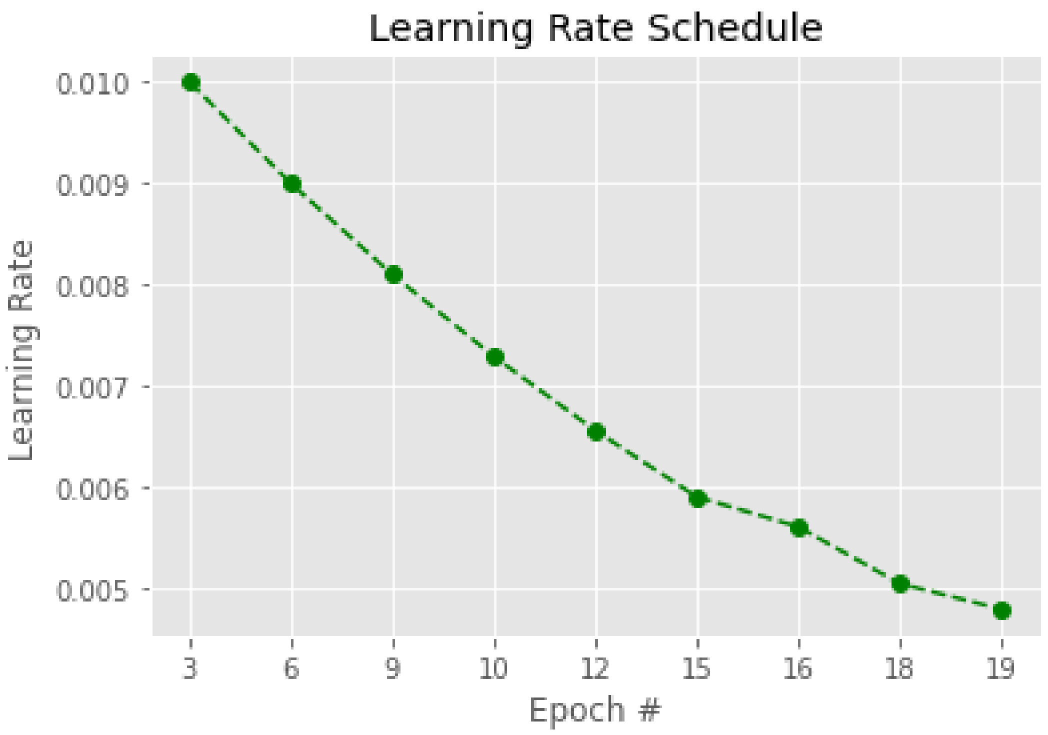
```
 ################################################################################
#####################
roc-auc_train: 0.737 - roc-auc_val: 0.7401

 ################################################################################
################################################################################
#########################################

Epoch 12/20
51/52 [============================>.] - ETA: 0s - loss: 0.6069 - micro_f1:
0.6580
Stop testing; got log keys: {'loss': 0.6026365756988525, 'micro_f1':
0.6481301784515381}
52/52 [==============================] - 1s 26ms/step - loss: 0.6069 - micro_f1:
0.6582 - val_loss: 0.6026 - val_micro_f1: 0.6481

End epoch 11 of training; got log keys: {'loss': 0.6049433350563049, 'micro_f1':
0.6612739562988281, 'val_loss': 0.6026365756988525, 'val_micro_f1':
0.6481301784515381}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.006560999434441328

roc-auc_train: 0.7368 - roc-auc_val: 0.7413

 ################################################################################
################################################################################
#########################################

Epoch 13/20
52/52 [==============================] - ETA: 0s - loss: 0.6000 - micro_f1:
0.6681
Stop testing; got log keys: {'loss': 0.6026133298873901, 'micro_f1':
0.6675258874893188}
52/52 [==============================] - 1s 26ms/step - loss: 0.6001 - micro_f1:
0.6680 - val_loss: 0.6026 - val_micro_f1: 0.6675

End epoch 12 of training; got log keys: {'loss': 0.6053372025489807, 'micro_f1':
0.6670522689819336, 'val_loss': 0.6026133298873901, 'val_micro_f1':
0.6675258874893188}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7371 - roc-auc_val: 0.7409
```

```
 ###############################################################################
################################################################################
#########################################

Epoch 14/20
51/52 [============================>.] - ETA: 0s - loss: 0.6039 - micro_f1:
0.6655
Stop testing; got log keys: {'loss': 0.6011123061180115, 'micro_f1':
0.6731069087982178}
52/52 [==============================] - 1s 26ms/step - loss: 0.6039 - micro_f1:
0.6655 - val_loss: 0.6011 - val_micro_f1: 0.6731

End epoch 13 of training; got log keys: {'loss': 0.6041221022605896, 'micro_f1':
0.6644833087921143, 'val_loss': 0.6011123061180115, 'val_micro_f1':
0.6731069087982178}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7372 - roc-auc_val: 0.7413

 ###############################################################################
################################################################################
#########################################

Epoch 15/20
51/52 [============================>.] - ETA: 0s - loss: 0.6062 - micro_f1:
0.6590
Stop testing; got log keys: {'loss': 0.6030179858207703, 'micro_f1':
0.6775444746017456}
52/52 [==============================] - 1s 26ms/step - loss: 0.6062 - micro_f1:
0.6591 - val_loss: 0.6030 - val_micro_f1: 0.6775

End epoch 14 of training; got log keys: {'loss': 0.6055681109428406, 'micro_f1':
0.6629765033721924, 'val_loss': 0.6030179858207703, 'val_micro_f1':
0.6775444746017456}

 custom learning rate by every 3rd epoch 0.0059048994444310665

roc-auc_train: 0.7369 - roc-auc_val: 0.7401

 ###############################################################################
################################################################################
#########################################

Epoch 16/20
```

```
51/52 [============================>.] - ETA: 0s - loss: 0.6044 - micro_f1:
0.6752
Stop testing; got log keys: {'loss': 0.599791944026947, 'micro_f1':
0.6681127548217773}
52/52 [==============================] - 1s 27ms/step - loss: 0.6044 - micro_f1:
0.6748 - val_loss: 0.5998 - val_micro_f1: 0.6681

End epoch 15 of training; got log keys: {'loss': 0.6048495769500732, 'micro_f1':
0.6659899950027466, 'val_loss': 0.599791944026947, 'val_micro_f1':
0.6681127548217773}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.005609654355794191

roc-auc_train: 0.7382 - roc-auc_val: 0.7413

 ###############################################################################
################################################################################
#########################################

Epoch 17/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6018 - micro_f1:
0.6617
Stop testing; got log keys: {'loss': 0.6009876728057861, 'micro_f1':
0.673945426940918}
52/52 [==============================] - 1s 26ms/step - loss: 0.6020 - micro_f1:
0.6621 - val_loss: 0.6010 - val_micro_f1: 0.6739

End epoch 16 of training; got log keys: {'loss': 0.6040785908699036, 'micro_f1':
0.6684337854385376, 'val_loss': 0.6009876728057861, 'val_micro_f1':
0.673945426940918}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7352 - roc-auc_val: 0.74

 ###############################################################################
################################################################################
#########################################

Epoch 18/20
51/52 [============================>.] - ETA: 0s - loss: 0.6056 - micro_f1:
0.6682
Stop testing; got log keys: {'loss': 0.6024540066719055, 'micro_f1':
0.6874932050704956}
```

```
52/52 [==============================] - 1s 27ms/step - loss: 0.6055 - micro_f1:
0.6681 - val_loss: 0.6025 - val_micro_f1: 0.6875

End epoch 17 of training; got log keys: {'loss': 0.6040064096450806, 'micro_f1':
0.6658637523651123, 'val_loss': 0.6024540066719055, 'val_micro_f1':
0.6874932050704956}

 custom learning rate by every 3rd epoch 0.005048688966780901

roc-auc_train: 0.7377 - roc-auc_val: 0.7409

 ################################################################################
################################################################################
#########################################

Epoch 19/20
52/52 [==============================] - ETA: 0s - loss: 0.5988 - micro_f1:
0.6703
Stop testing; got log keys: {'loss': 0.5998003482818604, 'micro_f1':
0.6763060688972473}
52/52 [==============================] - 1s 26ms/step - loss: 0.5989 - micro_f1:
0.6703 - val_loss: 0.5998 - val_micro_f1: 0.6763

End epoch 18 of training; got log keys: {'loss': 0.603130042552948, 'micro_f1':
0.6666910648345947, 'val_loss': 0.5998003482818604, 'val_micro_f1':
0.6763060688972473}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.004796254448592663

roc-auc_train: 0.7388 - roc-auc_val: 0.7425

 ################################################################################
################################################################################
#########################################

Epoch 20/20
51/52 [============================>.] - ETA: 0s - loss: 0.6046 - micro_f1:
0.6601
Stop testing; got log keys: {'loss': 0.6044525504112244, 'micro_f1':
0.6942376494407654}
52/52 [==============================] - 1s 27ms/step - loss: 0.6046 - micro_f1:
0.6602 - val_loss: 0.6045 - val_micro_f1: 0.6942

End epoch 19 of training; got log keys: {'loss': 0.6037642955780029, 'micro_f1':
0.6636781096458435, 'val_loss': 0.6044525504112244, 'val_micro_f1':
0.6942376494407654}
```

```
 ####################################################################################
######################
 Best Weights Saved :
 ####################################################################################
######################
roc-auc_train: 0.739 - roc-auc_val: 0.7426

 ####################################################################################
####################################################################################
###########################################

Stop training; got log keys: ['loss', 'micro_f1', 'val_loss', 'val_micro_f1']
125/125 [==============================] - 2s 14ms/step - loss: 0.6030 -
micro_f1: 0.6936

Test results - Loss: 0.6029993891716003 - micro_f1: 0.6936361789703369%
```



```
[ ]: from sklearn.metrics import classification_report, accuracy_score

     ypred = model1.predict(X_test)

     ypred = [1 if i>=0.5 else 0 for i in ypred.ravel()]
```

```
print("accuracy_score is ", accuracy_score(Y_test, ypred), "\n")

print(classification_report(Y_test, ypred))
```

```
accuracy_score is  0.67625

              precision    recall  f1-score   support

           0       0.71      0.61      0.65      2000
           1       0.65      0.75      0.70      2000

    accuracy                           0.68      4000
   macro avg       0.68      0.68      0.67      4000
weighted avg       0.68      0.68      0.67      4000
```

0.4.2 Model 2

[ ]:
```
from datetime import datetime
import os

logdir = "logs2/scalars/" + datetime.now().strftime("%Y%m%d-%H%M%S")

file_writer = tf.summary.create_file_writer(logdir + "/metrics")
file_writer.set_as_default()

tensorboard_callback_2 = tf.keras.callbacks.TensorBoard(log_dir=logdir,␣
 ↪histogram_freq=1, write_graph=True, write_grads=True)

%tensorboard --logdir logs2
```

[ ]:
```
model2 = get_model('relu', 'RandomUniform')

history_relu_1 = model2.fit(X_train, Y_train.ravel(), epochs=20, batch_size=250,
                         verbose=1, validation_split=0.2,
                         callbacks=[CustomCallback(), roc,␣
 ↪tensorboard_callback_2])

# Test the model after training
test_results = model2.evaluate(X_test, Y_test, verbose=1)

print(f'\nTest results - Loss: {test_results[0]} - micro_f1:␣
 ↪{test_results[1]}%')
```

```
Feature shape: (24,)
Epoch 1/20
 6/52 [==>…] - ETA: 3s - loss: 0.6918 - micro_f1:
0.0169    WARNING:tensorflow:Callback method `on_train_batch_end` is slow
```

```
compared to the batch time (batch time: 0.0227s vs `on_train_batch_end` time:
0.0300s). Check your callbacks.
50/52 [===========================>..] - ETA: 0s - loss: 0.6914 - micro_f1:
0.2337
Stop testing; got log keys: {'loss': 0.6886957287788391, 'micro_f1':
0.526188313961029}
52/52 [==============================] - 2s 33ms/step - loss: 0.6914 - micro_f1:
0.2379 - val_loss: 0.6887 - val_micro_f1: 0.5262

End epoch 0 of training; got log keys: {'loss': 0.6907801628112793, 'micro_f1':
0.30893611907958984, 'val_loss': 0.6886957287788391, 'val_micro_f1':
0.526188313961029}
roc-auc_train: 0.6873 - roc-auc_val: 0.6876

 ################################################################################
################################################################################
#########################################

Epoch 2/20
51/52 [============================>.] - ETA: 0s - loss: 0.6881 - micro_f1:
0.5939
Stop testing; got log keys: {'loss': 0.6826425790786743, 'micro_f1':
0.5204936265945435}
52/52 [==============================] - 1s 26ms/step - loss: 0.6880 - micro_f1:
0.5943 - val_loss: 0.6826 - val_micro_f1: 0.5205

End epoch 1 of training; got log keys: {'loss': 0.6865386366844177, 'micro_f1':
0.6049554944038391, 'val_loss': 0.6826425790786743, 'val_micro_f1':
0.5204936265945435}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.009999999776482582

roc-auc_train: 0.6738 - roc-auc_val: 0.6767

 ################################################################################
################################################################################
#########################################

Epoch 3/20
51/52 [============================>.] - ETA: 0s - loss: 0.6811 - micro_f1:
0.4482
Stop testing; got log keys: {'loss': 0.67177414894104, 'micro_f1':
0.5699312686920166}
52/52 [==============================] - 1s 25ms/step - loss: 0.6811 - micro_f1:
0.4506 - val_loss: 0.6718 - val_micro_f1: 0.5699

End epoch 2 of training; got log keys: {'loss': 0.6795642971992493, 'micro_f1':
```

```
0.5113114714622498, 'val_loss': 0.67177414894104, 'val_micro_f1':
0.5699312686920166}

 custom learning rate by every 3rd epoch 0.008999999612569809

roc-auc_train: 0.6874 - roc-auc_val: 0.6899

 ################################################################################
################################################################################
#########################################

Epoch 4/20
51/52 [============================>.] - ETA: 0s - loss: 0.6709 - micro_f1:
0.6033
Stop testing; got log keys: {'loss': 0.6548492908477783, 'micro_f1':
0.6412073969841003}
52/52 [==============================] - 1s 25ms/step - loss: 0.6707 - micro_f1:
0.6036 - val_loss: 0.6548 - val_micro_f1: 0.6412

End epoch 3 of training; got log keys: {'loss': 0.6672090291976929, 'micro_f1':
0.6101357936859131, 'val_loss': 0.6548492908477783, 'val_micro_f1':
0.6412073969841003}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7184 - roc-auc_val: 0.7201

 ################################################################################
################################################################################
#########################################

Epoch 5/20
51/52 [============================>.] - ETA: 0s - loss: 0.6541 - micro_f1:
0.6229
Stop testing; got log keys: {'loss': 0.63225257396698, 'micro_f1':
0.6212125420570374}
52/52 [==============================] - 1s 26ms/step - loss: 0.6539 - micro_f1:
0.6228 - val_loss: 0.6323 - val_micro_f1: 0.6212

End epoch 4 of training; got log keys: {'loss': 0.6477193236351013, 'micro_f1':
0.6215258240699768, 'val_loss': 0.63225257396698, 'val_micro_f1':
0.6212125420570374}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.008549999445676804
```

```
roc-auc_train: 0.7289 - roc-auc_val: 0.733

 ###############################################################################
################################################################################
########################################

Epoch 6/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6333 - micro_f1:
0.6426
Stop testing; got log keys: {'loss': 0.6182087659835815, 'micro_f1':
0.6597614288330078}
52/52 [==============================] - 1s 26ms/step - loss: 0.6331 - micro_f1:
0.6429 - val_loss: 0.6182 - val_micro_f1: 0.6598

End epoch 5 of training; got log keys: {'loss': 0.6297779679298401, 'micro_f1':
0.6482215523719788, 'val_loss': 0.6182087659835815, 'val_micro_f1':
0.6597614288330078}

 custom learning rate by every 3rd epoch 0.007694999687373638

roc-auc_train: 0.73 - roc-auc_val: 0.7349

 ###############################################################################
################################################################################
########################################

Epoch 7/20
52/52 [==============================] - ETA: 0s - loss: 0.6218 - micro_f1:
0.6710
Stop testing; got log keys: {'loss': 0.6110150814056396, 'micro_f1':
0.6699119806289673}
52/52 [==============================] - 1s 25ms/step - loss: 0.6217 - micro_f1:
0.6709 - val_loss: 0.6110 - val_micro_f1: 0.6699

End epoch 6 of training; got log keys: {'loss': 0.6214596629142761, 'micro_f1':
0.6664491891860962, 'val_loss': 0.6110150814056396, 'val_micro_f1':
0.6699119806289673}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7355 - roc-auc_val: 0.7397

 ###############################################################################
################################################################################
```

```
#########################################

Epoch 8/20
51/52 [============================>.] - ETA: 0s - loss: 0.6139 - micro_f1:
0.6661
Stop testing; got log keys: {'loss': 0.6080367565155029, 'micro_f1':
0.6849905848503113}
52/52 [==============================] - 1s 25ms/step - loss: 0.6140 - micro_f1:
0.6661 - val_loss: 0.6080 - val_micro_f1: 0.6850

End epoch 7 of training; got log keys: {'loss': 0.6161516904830933, 'micro_f1':
0.6668707728385925, 'val_loss': 0.6080367565155029, 'val_micro_f1':
0.6849905848503113}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7368 - roc-auc_val: 0.7406

 ###############################################################################
################################################################################
#########################################

Epoch 9/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6166 - micro_f1:
0.6788
Stop testing; got log keys: {'loss': 0.6043804287910461, 'micro_f1':
0.6684283018112183}
52/52 [==============================] - 1s 26ms/step - loss: 0.6164 - micro_f1:
0.6782 - val_loss: 0.6044 - val_micro_f1: 0.6684

End epoch 8 of training; got log keys: {'loss': 0.6127070784568787, 'micro_f1':
0.6657087802886963, 'val_loss': 0.6043804287910461, 'val_micro_f1':
0.6684283018112183}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.0073102498427033424

roc-auc_train: 0.7383 - roc-auc_val: 0.7423

 ###############################################################################
################################################################################
#########################################

Epoch 10/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6081 - micro_f1:
```

```
0.6783
Stop testing; got log keys: {'loss': 0.6057645678520203, 'micro_f1':
0.6817827224731445}
52/52 [==============================] - 1s 26ms/step - loss: 0.6082 - micro_f1:
0.6781 - val_loss: 0.6058 - val_micro_f1: 0.6818

End epoch 9 of training; got log keys: {'loss': 0.6109685301780701, 'micro_f1':
0.6748530864715576, 'val_loss': 0.6057645678520203, 'val_micro_f1':
0.6817827224731445}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7375 - roc-auc_val: 0.7403

 ################################################################################
################################################################################
#########################################

Epoch 11/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6093 - micro_f1:
0.6630
Stop testing; got log keys: {'loss': 0.6027776598930359, 'micro_f1':
0.6727354526519775}
52/52 [==============================] - 1s 26ms/step - loss: 0.6093 - micro_f1:
0.6635 - val_loss: 0.6028 - val_micro_f1: 0.6727

End epoch 10 of training; got log keys: {'loss': 0.6083833575248718, 'micro_f1':
0.671779990196228, 'val_loss': 0.6027776598930359, 'val_micro_f1':
0.6727354526519775}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.006579224951565266

roc-auc_train: 0.7383 - roc-auc_val: 0.7411

 ################################################################################
################################################################################
#########################################

Epoch 12/20
51/52 [============================>.] - ETA: 0s - loss: 0.6124 - micro_f1:
0.6632
Stop testing; got log keys: {'loss': 0.6031521558761597, 'micro_f1':
0.6613718271255493}
52/52 [==============================] - 1s 27ms/step - loss: 0.6122 - micro_f1:
```

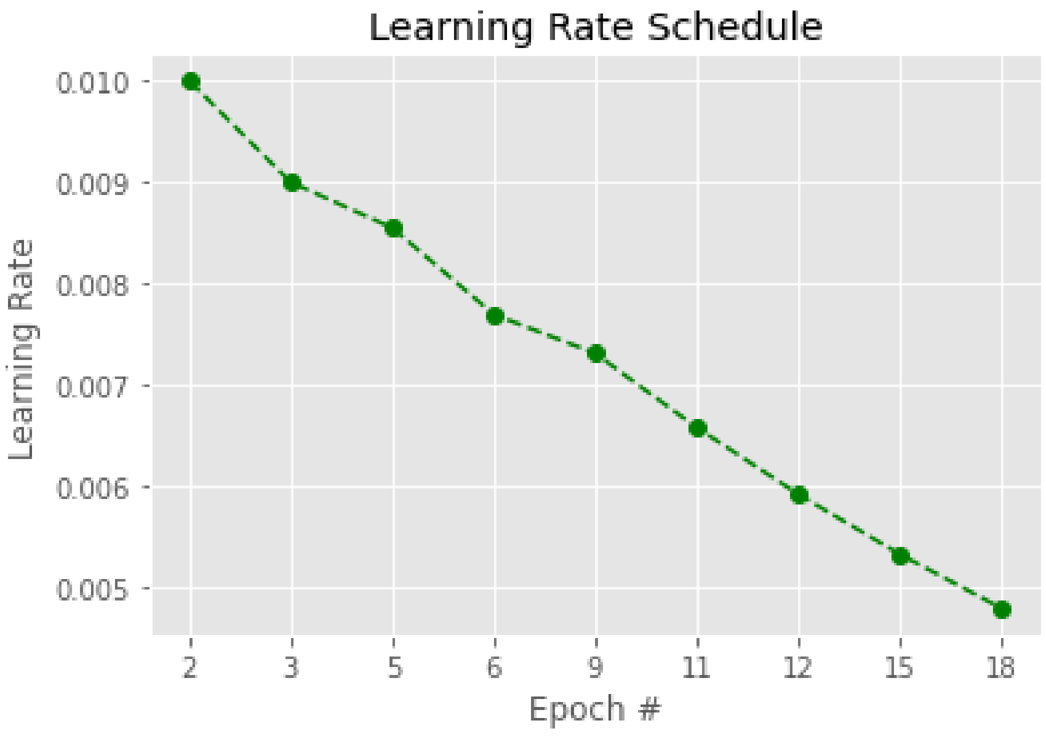
```
0.6637 - val_loss: 0.6032 - val_micro_f1: 0.6614

End epoch 11 of training; got log keys: {'loss': 0.6073871850967407, 'micro_f1':
0.675451934337616, 'val_loss': 0.6031521558761597, 'val_micro_f1':
0.6613718271255493}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.005921302363276482

roc-auc_train: 0.7385 - roc-auc_val: 0.7428

 ###############################################################################
################################################################################
#########################################

Epoch 13/20
51/52 [============================>.] - ETA: 0s - loss: 0.6101 - micro_f1:
0.6734
Stop testing; got log keys: {'loss': 0.6019361019134521, 'micro_f1':
0.6805298924446106}
52/52 [==============================] - 1s 27ms/step - loss: 0.6100 - micro_f1:
0.6734 - val_loss: 0.6019 - val_micro_f1: 0.6805

End epoch 12 of training; got log keys: {'loss': 0.6068596243858337, 'micro_f1':
0.6738134026527405, 'val_loss': 0.6019361019134521, 'val_micro_f1':
0.6805298924446106}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7392 - roc-auc_val: 0.7421

 ###############################################################################
################################################################################
#########################################

Epoch 14/20
51/52 [============================>.] - ETA: 0s - loss: 0.6064 - micro_f1:
0.6759
Stop testing; got log keys: {'loss': 0.6043311953544617, 'micro_f1':
0.68594890832901}
52/52 [==============================] - 1s 27ms/step - loss: 0.6064 - micro_f1:
0.6758 - val_loss: 0.6043 - val_micro_f1: 0.6859

End epoch 13 of training; got log keys: {'loss': 0.6066240072250366, 'micro_f1':
0.6738137602806091, 'val_loss': 0.6043311953544617, 'val_micro_f1':
```

```
0.68594890832901}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7377 - roc-auc_val: 0.7415

 ###############################################################################
################################################################################
#########################################

Epoch 15/20
52/52 [==============================] - ETA: 0s - loss: 0.6012 - micro_f1:
0.6682
Stop testing; got log keys: {'loss': 0.6020737290382385, 'micro_f1':
0.6750860214233398}
52/52 [==============================] - 1s 27ms/step - loss: 0.6013 - micro_f1:
0.6683 - val_loss: 0.6021 - val_micro_f1: 0.6751

End epoch 14 of training; got log keys: {'loss': 0.6055112481117249, 'micro_f1':
0.673953652381897, 'val_loss': 0.6020737290382385, 'val_micro_f1':
0.6750860214233398}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.0053291721269488335

roc-auc_train: 0.7392 - roc-auc_val: 0.7426

 ###############################################################################
################################################################################
#########################################

Epoch 16/20
51/52 [============================>.] - ETA: 0s - loss: 0.6050 - micro_f1:
0.6696
Stop testing; got log keys: {'loss': 0.6012682914733887, 'micro_f1':
0.6754887700080872}
52/52 [==============================] - 1s 26ms/step - loss: 0.6050 - micro_f1:
0.6697 - val_loss: 0.6013 - val_micro_f1: 0.6755

End epoch 15 of training; got log keys: {'loss': 0.605307936668396, 'micro_f1':
0.6704759001731873, 'val_loss': 0.6012682914733887, 'val_micro_f1':
0.6754887700080872}

 ###############################################################################
#####################
```

```
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7375 - roc-auc_val: 0.742

 ###############################################################################
################################################################################
#########################################

Epoch 17/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6085 - micro_f1:
0.6759
Stop testing; got log keys: {'loss': 0.6020805835723877, 'micro_f1':
0.6857197284698486}
52/52 [==============================] - 1s 27ms/step - loss: 0.6084 - micro_f1:
0.6756 - val_loss: 0.6021 - val_micro_f1: 0.6857

End epoch 16 of training; got log keys: {'loss': 0.6064726114273071, 'micro_f1':
0.6702831983566284, 'val_loss': 0.6020805835723877, 'val_micro_f1':
0.6857197284698486}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.739 - roc-auc_val: 0.7421

 ###############################################################################
################################################################################
#########################################

Epoch 18/20
51/52 [============================>.] - ETA: 0s - loss: 0.6054 - micro_f1:
0.6847
Stop testing; got log keys: {'loss': 0.6004654765129089, 'micro_f1':
0.6722474694252014}
52/52 [==============================] - 1s 27ms/step - loss: 0.6054 - micro_f1:
0.6843 - val_loss: 0.6005 - val_micro_f1: 0.6722

End epoch 17 of training; got log keys: {'loss': 0.6050372123718262, 'micro_f1':
0.6733013987541199, 'val_loss': 0.6004654765129089, 'val_micro_f1':
0.6722474694252014}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.00479625491425395

roc-auc_train: 0.7386 - roc-auc_val: 0.7423
```

```
 ###############################################################################
################################################################################
#########################################

Epoch 19/20
52/52 [==============================] - ETA: 0s - loss: 0.6020 - micro_f1:
0.6650
Stop testing; got log keys: {'loss': 0.6030206680297852, 'micro_f1':
0.6762480139732361}
52/52 [==============================] - 1s 27ms/step - loss: 0.6020 - micro_f1:
0.6651 - val_loss: 0.6030 - val_micro_f1: 0.6762

End epoch 18 of training; got log keys: {'loss': 0.6038675904273987, 'micro_f1':
0.6689395904541016, 'val_loss': 0.6030206680297852, 'val_micro_f1':
0.6762480139732361}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7387 - roc-auc_val: 0.7416

 ###############################################################################
################################################################################
#########################################

Epoch 20/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6047 - micro_f1:
0.6711
Stop testing; got log keys: {'loss': 0.6042959094047546, 'micro_f1':
0.6849602460861206}
52/52 [==============================] - 1s 26ms/step - loss: 0.6046 - micro_f1:
0.6711 - val_loss: 0.6043 - val_micro_f1: 0.6850

End epoch 19 of training; got log keys: {'loss': 0.6037721037864685, 'micro_f1':
0.6713864803314209, 'val_loss': 0.6042959094047546, 'val_micro_f1':
0.6849602460861206}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7379 - roc-auc_val: 0.7402

 ###############################################################################
```

```
################################################################################
#########################################

Stop training; got log keys: ['loss', 'micro_f1', 'val_loss', 'val_micro_f1']
125/125 [==============================] - 2s 13ms/step - loss: 0.6030 -
micro_f1: 0.6839

Test results - Loss: 0.6029933094978333 - micro_f1: 0.6838955879211426%
```



[ ]:
```python
from sklearn.metrics import classification_report, accuracy_score

ypred = model2.predict(X_test)

ypred = [1 if i>=0.5 else 0 for i in ypred.ravel()]

print("accuracy_score is ", accuracy_score(Y_test, ypred), "\n")

print("micro-f1 is ", test_results[1], "\n")

print(classification_report(Y_test, ypred))
```

```
accuracy_score is  0.675

micro-f1 is  0.6838955879211426
```

```
              precision    recall  f1-score   support

           0       0.69      0.63      0.66      2000
           1       0.66      0.72      0.69      2000

    accuracy                           0.68      4000
   macro avg       0.68      0.68      0.67      4000
weighted avg       0.68      0.68      0.67      4000
```

0.4.3 Model 3

```
[ ]: from datetime import datetime
     import os

     logdir = "logs3/scalars/" + datetime.now().strftime("%Y%m%d-%H%M%S")

     file_writer = tf.summary.create_file_writer(logdir + "/metrics")
     file_writer.set_as_default()

     tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir,␣
      ↪histogram_freq=1, write_graph=True, write_grads=True)

     %tensorboard --logdir logs3
```

```
[ ]: model3 = get_model('relu', 'he_uniform')

     history_relu_2 = model3.fit(X_train, Y_train.ravel(), epochs=20, batch_size=250,
                                 verbose=1, validation_split=0.2,
                                 callbacks=[CustomCallback(), roc,␣
      ↪tensorboard_callback])

     # Test the model after training
     test_results = model3.evaluate(X_test, Y_test, verbose=1)

     print(f'\nTest results - Loss: {test_results[0]} - micro_f1:␣
      ↪{test_results[1]}%')
```

```
Feature shape: (24,)
Epoch 1/20
 6/52 [==>…] - ETA: 3s - loss: 0.7633 - micro_f1:
0.6699WARNING:tensorflow:Callback method `on_train_batch_end` is slow compared
to the batch time (batch time: 0.0234s vs `on_train_batch_end` time: 0.0290s).
Check your callbacks.
50/52 [===========================>..] - ETA: 0s - loss: 0.7176 - micro_f1:
0.4156
Stop testing; got log keys: {'loss': 0.6772550344467163, 'micro_f1':
```

```
0.5948480367660522}
52/52 [==============================] - 2s 33ms/step - loss: 0.7164 - micro_f1:
0.4118 - val_loss: 0.6773 - val_micro_f1: 0.5948

End epoch 0 of training; got log keys: {'loss': 0.6961822509765625, 'micro_f1':
0.35076263546943665, 'val_loss': 0.6772550344467163, 'val_micro_f1':
0.5948480367660522}
roc-auc_train: 0.6083 - roc-auc_val: 0.6112

 ################################################################################
################################################################################
########################################

Epoch 2/20
52/52 [==============================] - ETA: 0s - loss: 0.6761 - micro_f1:
0.6000
Stop testing; got log keys: {'loss': 0.6557791233062744, 'micro_f1':
0.5846095681190491}
52/52 [==============================] - 1s 26ms/step - loss: 0.6760 - micro_f1:
0.6002 - val_loss: 0.6558 - val_micro_f1: 0.5846

End epoch 1 of training; got log keys: {'loss': 0.6709296703338623, 'micro_f1':
0.6084076166152954, 'val_loss': 0.6557791233062744, 'val_micro_f1':
0.5846095681190491}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.009999999776482582

roc-auc_train: 0.6809 - roc-auc_val: 0.6843

 ################################################################################
################################################################################
########################################

Epoch 3/20
51/52 [============================>.] - ETA: 0s - loss: 0.6537 - micro_f1:
0.6003
Stop testing; got log keys: {'loss': 0.6325070858001709, 'micro_f1':
0.6376380324363708}
52/52 [==============================] - 1s 26ms/step - loss: 0.6535 - micro_f1:
0.6011 - val_loss: 0.6325 - val_micro_f1: 0.6376

End epoch 2 of training; got log keys: {'loss': 0.6483854055404663, 'micro_f1':
0.6204692125320435, 'val_loss': 0.6325070858001709, 'val_micro_f1':
0.6376380324363708}

 custom learning rate by every 3rd epoch 0.008999999612569809
```

```
roc-auc_train: 0.7201 - roc-auc_val: 0.7222

 ###############################################################################
################################################################################
#########################################

Epoch 4/20
51/52 [============================>.] - ETA: 0s - loss: 0.6361 - micro_f1:
0.6314
Stop testing; got log keys: {'loss': 0.6203228235244751, 'micro_f1':
0.6476358771324158}
52/52 [==============================] - 1s 25ms/step - loss: 0.6359 - micro_f1:
0.6321 - val_loss: 0.6203 - val_micro_f1: 0.6476

End epoch 3 of training; got log keys: {'loss': 0.6310450434684753, 'micro_f1':
0.6516300439834595, 'val_loss': 0.6203228235244751, 'val_micro_f1':
0.6476358771324158}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7323 - roc-auc_val: 0.7345

 ###############################################################################
################################################################################
#########################################

Epoch 5/20
52/52 [==============================] - ETA: 0s - loss: 0.6237 - micro_f1:
0.6557
Stop testing; got log keys: {'loss': 0.6141676306724548, 'micro_f1':
0.687527060508728}
52/52 [==============================] - 1s 25ms/step - loss: 0.6236 - micro_f1:
0.6560 - val_loss: 0.6142 - val_micro_f1: 0.6875

End epoch 4 of training; got log keys: {'loss': 0.6193337440490723, 'micro_f1':
0.6711465120315552, 'val_loss': 0.6141676306724548, 'val_micro_f1':
0.687527060508728}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7342 - roc-auc_val: 0.7369
```

```
 ###############################################################################
################################################################################
#########################################

Epoch 6/20
51/52 [============================>.] - ETA: 0s - loss: 0.6163 - micro_f1:
0.6705
Stop testing; got log keys: {'loss': 0.6089470386505127, 'micro_f1':
0.6942929625511169}
52/52 [==============================] - 1s 26ms/step - loss: 0.6163 - micro_f1:
0.6706 - val_loss: 0.6089 - val_micro_f1: 0.6943

End epoch 5 of training; got log keys: {'loss': 0.6166874170303345, 'micro_f1':
0.6736488342285156, 'val_loss': 0.6089470386505127, 'val_micro_f1':
0.6942929625511169}

 custom learning rate by every 3rd epoch 0.008549999445676804

roc-auc_train: 0.7369 - roc-auc_val: 0.7382

 ###############################################################################
################################################################################
#########################################

Epoch 7/20
51/52 [============================>.] - ETA: 0s - loss: 0.6114 - micro_f1:
0.6844
Stop testing; got log keys: {'loss': 0.6048816442489624, 'micro_f1':
0.6755568385124207}
52/52 [==============================] - 1s 26ms/step - loss: 0.6114 - micro_f1:
0.6843 - val_loss: 0.6049 - val_micro_f1: 0.6756

End epoch 6 of training; got log keys: {'loss': 0.6119400262832642, 'micro_f1':
0.6821113228797913, 'val_loss': 0.6048816442489624, 'val_micro_f1':
0.6755568385124207}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.008122499100863934

roc-auc_train: 0.7365 - roc-auc_val: 0.7379

 ###############################################################################
################################################################################
#########################################

Epoch 8/20
52/52 [==============================] - ETA: 0s - loss: 0.6047 - micro_f1:
0.6829
```

```
Stop testing; got log keys: {'loss': 0.6036360263824463, 'micro_f1':
0.6911481618881226}
52/52 [==============================] - 1s 26ms/step - loss: 0.6047 - micro_f1:
0.6828 - val_loss: 0.6036 - val_micro_f1: 0.6911

End epoch 7 of training; got log keys: {'loss': 0.6090585589408875, 'micro_f1':
0.6786856055259705, 'val_loss': 0.6036360263824463, 'val_micro_f1':
0.6911481618881226}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7382 - roc-auc_val: 0.7389

 ###############################################################################
################################################################################
#########################################

Epoch 9/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6082 - micro_f1:
0.6701
Stop testing; got log keys: {'loss': 0.6100297570228577, 'micro_f1':
0.6945899724960327}
52/52 [==============================] - 1s 26ms/step - loss: 0.6081 - micro_f1:
0.6705 - val_loss: 0.6100 - val_micro_f1: 0.6946

End epoch 8 of training; got log keys: {'loss': 0.6065118908882141, 'micro_f1':
0.6772119998931885, 'val_loss': 0.6100297570228577, 'val_micro_f1':
0.6945899724960327}

 custom learning rate by every 3rd epoch 0.007310249377042055

roc-auc_train: 0.7349 - roc-auc_val: 0.7348

 ###############################################################################
################################################################################
#########################################

Epoch 10/20
51/52 [============================>.] - ETA: 0s - loss: 0.6088 - micro_f1:
0.6821
Stop testing; got log keys: {'loss': 0.6053360104560852, 'micro_f1':
0.6856521964073181}
52/52 [==============================] - 1s 26ms/step - loss: 0.6087 - micro_f1:
0.6819 - val_loss: 0.6053 - val_micro_f1: 0.6857
```

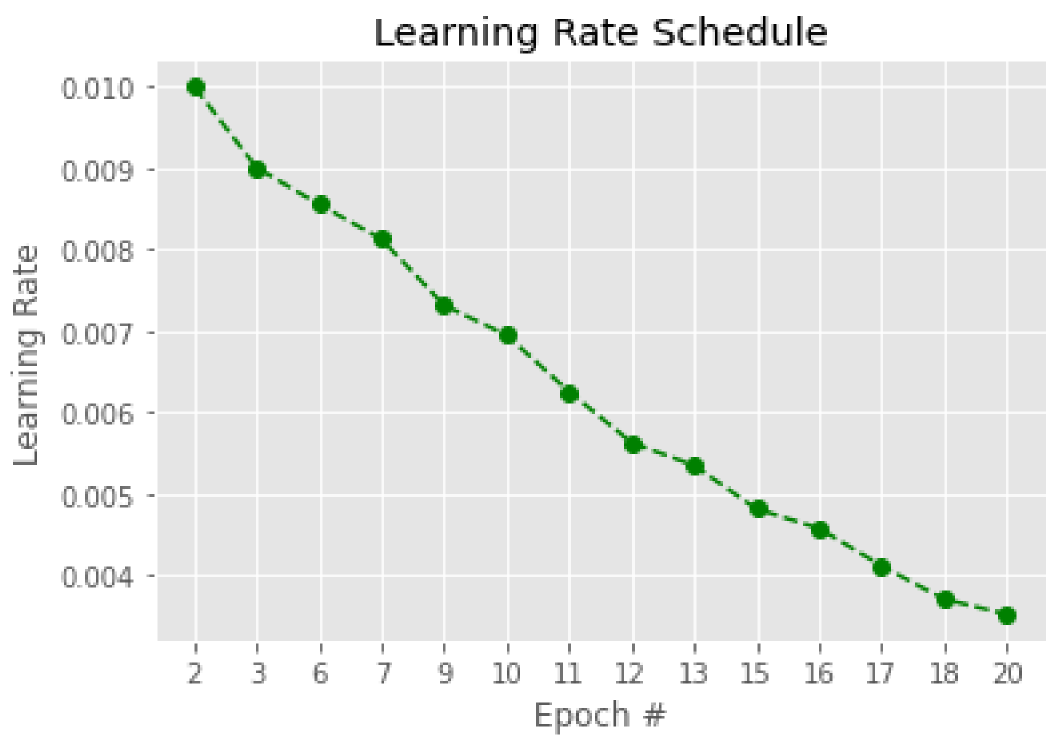
```
End epoch 9 of training; got log keys: {'loss': 0.606156587600708, 'micro_f1':
0.6763495802879333, 'val_loss': 0.6053360104560852, 'val_micro_f1':
0.6856521964073181}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.006944736931473017

roc-auc_train: 0.7361 - roc-auc_val: 0.7374

 ################################################################################
################################################################################
#########################################

Epoch 11/20
51/52 [============================>.] - ETA: 0s - loss: 0.6022 - micro_f1:
0.6787
Stop testing; got log keys: {'loss': 0.6017211079597473, 'micro_f1':
0.6804577708244324}
52/52 [==============================] - 1s 26ms/step - loss: 0.6023 - micro_f1:
0.6786 - val_loss: 0.6017 - val_micro_f1: 0.6805

End epoch 10 of training; got log keys: {'loss': 0.6047673225402832, 'micro_f1':
0.6758349537849426, 'val_loss': 0.6017211079597473, 'val_micro_f1':
0.6804577708244324}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.006250263191759586

roc-auc_train: 0.7389 - roc-auc_val: 0.7397

 ################################################################################
################################################################################
#########################################

Epoch 12/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6064 - micro_f1:
0.6687
Stop testing; got log keys: {'loss': 0.6019937992095947, 'micro_f1':
0.6838706135749817}
52/52 [==============================] - 1s 26ms/step - loss: 0.6063 - micro_f1:
0.6690 - val_loss: 0.6020 - val_micro_f1: 0.6839

End epoch 11 of training; got log keys: {'loss': 0.6047848463058472, 'micro_f1':
0.6743170022964478, 'val_loss': 0.6019937992095947, 'val_micro_f1':
0.6838706135749817}

 custom learning rate by every 3rd epoch 0.00562523677945137
```

```
roc-auc_train: 0.7388 - roc-auc_val: 0.7393

 ###############################################################################
################################################################################
#########################################

Epoch 13/20
52/52 [==============================] - ETA: 0s - loss: 0.5971 - micro_f1:
0.6868
Stop testing; got log keys: {'loss': 0.6012009382247925, 'micro_f1':
0.6575754880905151}
52/52 [==============================] - 1s 27ms/step - loss: 0.5973 - micro_f1:
0.6867 - val_loss: 0.6012 - val_micro_f1: 0.6576

End epoch 12 of training; got log keys: {'loss': 0.6041357517242432, 'micro_f1':
0.6812124252319336, 'val_loss': 0.6012009382247925, 'val_micro_f1':
0.6575754880905151}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.005343975033611059

roc-auc_train: 0.7397 - roc-auc_val: 0.7413

 ###############################################################################
################################################################################
#########################################

Epoch 14/20
52/52 [==============================] - ETA: 0s - loss: 0.6052 - micro_f1:
0.6569
Stop testing; got log keys: {'loss': 0.6008774042129517, 'micro_f1':
0.677869439125061}
52/52 [==============================] - 1s 27ms/step - loss: 0.6052 - micro_f1:
0.6571 - val_loss: 0.6009 - val_micro_f1: 0.6779

End epoch 13 of training; got log keys: {'loss': 0.6030940413475037, 'micro_f1':
0.6681166887283325, 'val_loss': 0.6008774042129517, 'val_micro_f1':
0.677869439125061}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7394 - roc-auc_val: 0.7402

 ###############################################################################
################################################################################
```

```
#########################################

Epoch 15/20
51/52 [===========================>.] - ETA: 0s - loss: 0.6036 - micro_f1:
0.6827
Stop testing; got log keys: {'loss': 0.6006501913070679, 'micro_f1':
0.678145706653595}
52/52 [==============================] - 1s 27ms/step - loss: 0.6035 - micro_f1:
0.6824 - val_loss: 0.6007 - val_micro_f1: 0.6781

End epoch 14 of training; got log keys: {'loss': 0.6025674343109131, 'micro_f1':
0.6760504245758057, 'val_loss': 0.6006501913070679, 'val_micro_f1':
0.678145706653595}

 custom learning rate by every 3rd epoch 0.0048095774836838245

roc-auc_train: 0.7393 - roc-auc_val: 0.7404

 ################################################################################
################################################################################
#########################################

Epoch 16/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6040 - micro_f1:
0.6719
Stop testing; got log keys: {'loss': 0.6004400849342346, 'micro_f1':
0.6704272031784058}
52/52 [==============================] - 1s 26ms/step - loss: 0.6039 - micro_f1:
0.6721 - val_loss: 0.6004 - val_micro_f1: 0.6704

End epoch 15 of training; got log keys: {'loss': 0.6018631458282471, 'micro_f1':
0.6761796474456787, 'val_loss': 0.6004400849342346, 'val_micro_f1':
0.6704272031784058}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.004569098819047213

roc-auc_train: 0.7398 - roc-auc_val: 0.7405

 ################################################################################
################################################################################
#########################################

Epoch 17/20
51/52 [===========================>.] - ETA: 0s - loss: 0.6012 - micro_f1:
0.6865
Stop testing; got log keys: {'loss': 0.6015841364860535, 'micro_f1':
0.6488962173461914}
```

```
52/52 [==============================] - 1s 25ms/step - loss: 0.6012 - micro_f1:
0.6861 - val_loss: 0.6016 - val_micro_f1: 0.6489

End epoch 16 of training; got log keys: {'loss': 0.6016396880149841, 'micro_f1':
0.6771554946899414, 'val_loss': 0.6015841364860535, 'val_micro_f1':
0.6488962173461914}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.004112188704311848

roc-auc_train: 0.7392 - roc-auc_val: 0.7414

 ################################################################################
################################################################################
#########################################

Epoch 18/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6035 - micro_f1:
0.6717
Stop testing; got log keys: {'loss': 0.6003637909889221, 'micro_f1':
0.6604048609733582}
52/52 [==============================] - 1s 27ms/step - loss: 0.6034 - micro_f1:
0.6718 - val_loss: 0.6004 - val_micro_f1: 0.6604

End epoch 17 of training; got log keys: {'loss': 0.6017613410949707, 'micro_f1':
0.6733213663101196, 'val_loss': 0.6003637909889221, 'val_micro_f1':
0.6604048609733582}

 custom learning rate by every 3rd epoch 0.0037009697407484055

roc-auc_train: 0.7401 - roc-auc_val: 0.7412

 ################################################################################
################################################################################
#########################################

Epoch 19/20
51/52 [============================>.] - ETA: 0s - loss: 0.5931 - micro_f1:
0.6781
Stop testing; got log keys: {'loss': 0.5991052389144897, 'micro_f1':
0.6744877696037292}
52/52 [==============================] - 1s 27ms/step - loss: 0.5934 - micro_f1:
0.6780 - val_loss: 0.5991 - val_micro_f1: 0.6745

End epoch 18 of training; got log keys: {'loss': 0.6011693477630615, 'micro_f1':
0.6752077341079712, 'val_loss': 0.5991052389144897, 'val_micro_f1':
0.6744877696037292}
```

```
 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7397 - roc-auc_val: 0.7411

 ###############################################################################
################################################################################
#########################################

Epoch 20/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6072 - micro_f1:
0.6712
Stop testing; got log keys: {'loss': 0.6003721952438354, 'micro_f1':
0.673444390296936}
52/52 [==============================] - 1s 27ms/step - loss: 0.6069 - micro_f1:
0.6714 - val_loss: 0.6004 - val_micro_f1: 0.6734

End epoch 19 of training; got log keys: {'loss': 0.6005962491035461, 'micro_f1':
0.674835205078125, 'val_loss': 0.6003721952438354, 'val_micro_f1':
0.673444390296936}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.003515921300277114

roc-auc_train: 0.7397 - roc-auc_val: 0.7407

 ###############################################################################
################################################################################
#########################################

Stop training; got log keys: ['loss', 'micro_f1', 'val_loss', 'val_micro_f1']
125/125 [==============================] - 2s 14ms/step - loss: 0.6009 -
micro_f1: 0.6693

Test results - Loss: 0.6009096503257751 - micro_f1: 0.6693171858787537%
```



[ ]:
```
from sklearn.metrics import classification_report, accuracy_score

ypred = model3.predict(X_test)

ypred = [1 if i>=0.5 else 0 for i in ypred.ravel()]

print("accuracy_score is ", accuracy_score(Y_test, ypred), "\n")

print("micro -f1 is ", test_results[1], "\n")

print(classification_report(Y_test, ypred))
```

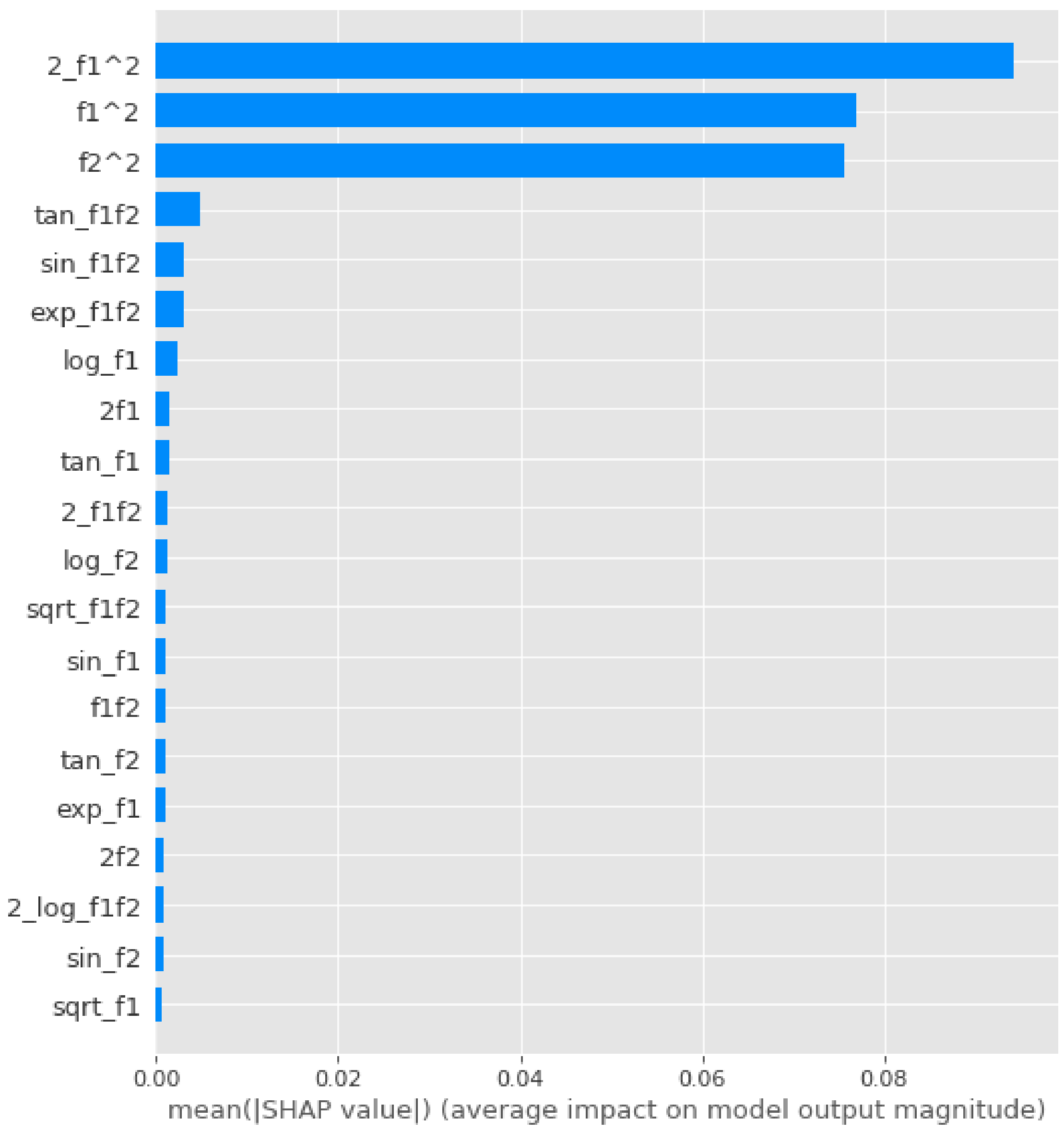
```
accuracy_score is  0.6755

micro -f1 is  0.6693171858787537

              precision    recall  f1-score   support

           0       0.67      0.68      0.68      2000
           1       0.68      0.67      0.67      2000

    accuracy                           0.68      4000
   macro avg       0.68      0.68      0.68      4000
weighted avg       0.68      0.68      0.68      4000
```

```
[ ]: !pip install shap
```

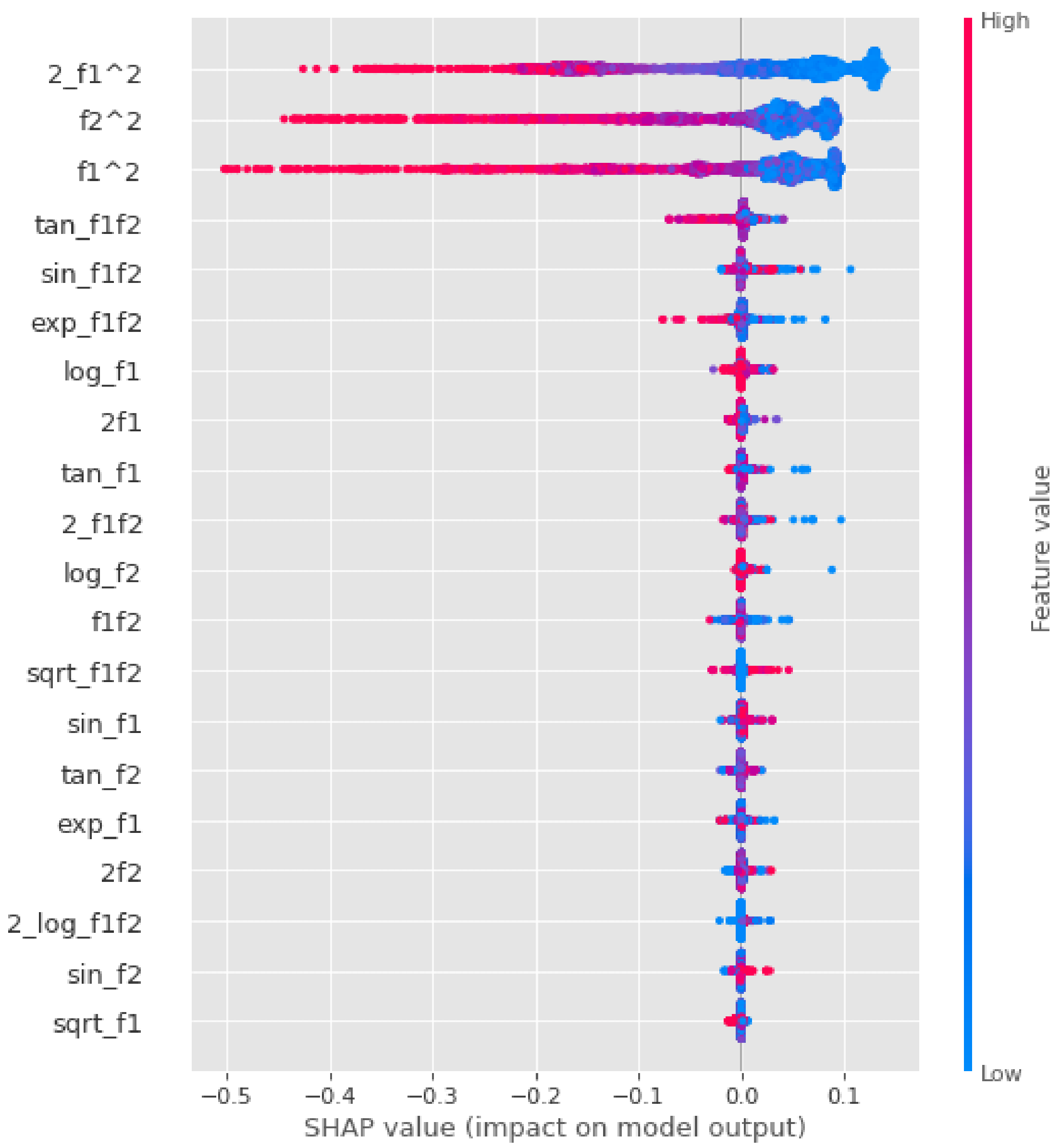
```
[ ]: from sklearn.ensemble import RandomForestRegressor

     model = RandomForestRegressor(max_depth=6, random_state=0, n_estimators=10)
     model.fit(X_train, Y_train)
```

```
[ ]: RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                           max_depth=6, max_features='auto', max_leaf_nodes=None,
                           max_samples=None, min_impurity_decrease=0.0,
                           min_impurity_split=None, min_samples_leaf=1,
                           min_samples_split=2, min_weight_fraction_leaf=0.0,
                           n_estimators=10, n_jobs=None, oob_score=False,
                           random_state=0, verbose=0, warm_start=False)
```

```
[ ]: import shap

     shap_values = shap.TreeExplainer(model).shap_values(X_train)

     shap.summary_plot(shap_values, X_train, plot_type="bar")
```



[ ]:
```
import matplotlib.pyplot as plt

shap_values = shap.TreeExplainer(model).shap_values(X_test)

f = plt.figure()
shap.summary_plot(shap_values, X_test)
f.savefig("/summary_plot1.png", bbox_inches='tight', dpi=600)
```



1 Model 4

```
[ ]: !pip install keras-tuner
```

```
[38]: from tensorflow import keras
      from tensorflow.keras import layers
      from kerastuner.tuners import RandomSearch


      def build_hypermodel(hp):
```

```
  input_shape = (X_train.shape[1],)

  model = Sequential()

  model.add(Dense(units=hp.Int(name='units_1',
            min_value=128, max_value=256, step=32),
            input_shape=input_shape,
            activation = hp.Choice(name='a_1',
                                   values=['relu','tanh','selu','swish'])))

  model.add(Dense(hp.Int('units_2', 32, 64, 16),
            activation = hp.Choice(name='a_2',
                                   values=['relu','selu','swish'])))

  model.add(Dense(hp.Int('units_3', 16, 32, 8),
            activation = hp.Choice(name='a_3',
                                   values=['relu','swish'])))

  model.add(Dense(hp.Int('units_4', 8, 16, 4),
            activation = hp.Choice(name='a_4',
                                   values=['relu','swish'])))

  model.add(Dense(hp.Int('units_5', 2, 4, 1),
            activation = hp.Choice(name='a_5',
                                   values=['relu','swish'])))

  model.add(Dense(num_classes, activation='sigmoid'))

  sgd = SGD(lr=0.01, momentum=0.9)

  model.compile(loss='binary_crossentropy', optimizer=sgd,
                metrics=['accuracy'], run_eagerly=True)

  return model
```

[7]:
```
tuner = RandomSearch(
    build_hypermodel,
    objective='val_accuracy',
    max_trials=50,
    executions_per_trial=3,
    directory='my_dir2',
    project_name='HyperModel')

stop_early = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)
```

[9]:
```
tuner.search_space_summary()
```

```
Search space summary
Default search space size: 10
units_1 (Int)
{'default': None, 'conditions': [], 'min_value': 128, 'max_value': 256, 'step':
32, 'sampling': None}
a_1 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'tanh', 'selu',
'swish'], 'ordered': False}
units_2 (Int)
{'default': None, 'conditions': [], 'min_value': 32, 'max_value': 64, 'step':
16, 'sampling': None}
a_2 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'selu', 'swish'],
'ordered': False}
units_3 (Int)
{'default': None, 'conditions': [], 'min_value': 16, 'max_value': 32, 'step': 8,
'sampling': None}
a_3 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'swish'], 'ordered':
False}
units_4 (Int)
{'default': None, 'conditions': [], 'min_value': 8, 'max_value': 16, 'step': 4,
'sampling': None}
a_4 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'swish'], 'ordered':
False}
units_5 (Int)
{'default': None, 'conditions': [], 'min_value': 2, 'max_value': 4, 'step': 1,
'sampling': None}
a_5 (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'swish'], 'ordered':
False}
```

[10]:
```
tuner.search(X_train, Y_train,
             steps_per_epoch=128,
             epochs=20,
             validation_data=(X_test, Y_test), callbacks=[stop_early])
```

```
Trial 50 Complete [00h 02m 20s]
val_accuracy: 0.6789166529973348

Best val_accuracy So Far: 0.6799166798591614
Total elapsed time: 01h 49m 47s
INFO:tensorflow:Oracle triggered exit
```

[24]:
```
best_model = tuner.get_best_models(num_models=1)[0]
```

[27]:
```
from sklearn.metrics import classification_report, accuracy_score

ypred = best_model.predict(X_test)

ypred = [1 if i>=0.5 else 0 for i in ypred.ravel()]

print("accuracy_score is ", accuracy_score(Y_test, ypred), "\n")

print(classification_report(Y_test, ypred))
```

```
accuracy_score is  0.68075

              precision    recall  f1-score   support

           0       0.68      0.69      0.68      2000
           1       0.68      0.67      0.68      2000

    accuracy                           0.68      4000
   macro avg       0.68      0.68      0.68      4000
weighted avg       0.68      0.68      0.68      4000
```

[23]:
```
tuner.results_summary()
```

```
Results summary
Results in my_dir2/HyperModel
Showing 10 best trials
Objective(name='val_accuracy', direction='max')
Trial summary
Hyperparameters:
units_1: 128
a_1: relu
units_2: 32
a_2: selu
units_3: 24
a_3: swish
units_4: 16
a_4: swish
units_5: 2
a_5: swish
Score: 0.6799166798591614
Trial summary
Hyperparameters:
units_1: 160
a_1: swish
units_2: 64
a_2: relu
units_3: 32
```

```
a_3: swish
units_4: 12
a_4: relu
units_5: 3
a_5: relu
Score: 0.6796666582425436
Trial summary
Hyperparameters:
units_1: 256
a_1: relu
units_2: 32
a_2: swish
units_3: 16
a_3: relu
units_4: 12
a_4: swish
units_5: 4
a_5: relu
Score: 0.6795000036557516
Trial summary
Hyperparameters:
units_1: 224
a_1: tanh
units_2: 64
a_2: swish
units_3: 32
a_3: relu
units_4: 16
a_4: relu
units_5: 3
a_5: swish
Score: 0.6795000036557516
Trial summary
Hyperparameters:
units_1: 160
a_1: selu
units_2: 48
a_2: relu
units_3: 24
a_3: swish
units_4: 12
a_4: swish
units_5: 4
a_5: swish
Score: 0.6794166763623556
Trial summary
Hyperparameters:
units_1: 192
```

```
a_1: tanh
units_2: 64
a_2: selu
units_3: 24
a_3: relu
units_4: 8
a_4: swish
units_5: 3
a_5: relu
Score: 0.6794166763623556
Trial summary
Hyperparameters:
units_1: 192
a_1: tanh
units_2: 48
a_2: swish
units_3: 24
a_3: relu
units_4: 12
a_4: relu
units_5: 3
a_5: relu
Score: 0.6794166564941406
Trial summary
Hyperparameters:
units_1: 224
a_1: relu
units_2: 64
a_2: selu
units_3: 24
a_3: relu
units_4: 12
a_4: relu
units_5: 4
a_5: relu
Score: 0.6792500019073486
Trial summary
Hyperparameters:
units_1: 128
a_1: tanh
units_2: 48
a_2: selu
units_3: 24
a_3: swish
units_4: 16
a_4: swish
units_5: 3
a_5: swish
```

```
Score: 0.6790833274523417
Trial summary
Hyperparameters:
units_1: 128
a_1: tanh
units_2: 32
a_2: relu
units_3: 24
a_3: swish
units_4: 16
a_4: relu
units_5: 2
a_5: relu
Score: 0.6790000001589457
```

[ ]:
```
from datetime import datetime
import os

logdir = "logs4/scalars/" + datetime.now().strftime("%Y%m%d-%H%M%S")

file_writer = tf.summary.create_file_writer(logdir + "/metrics")
file_writer.set_as_default()

tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir,␣
 ↪histogram_freq=1, write_graph=True, write_grads=True)

%tensorboard --logdir logs4
```

[44]:
```
def get_model():

  input_shape = (X_train.shape[1],)

  model = Sequential()

  model.add(Dense(128,
            input_shape=input_shape,
            activation = 'relu'))

  model.add(Dense(32,
            activation = 'selu'))

  model.add(Dense(24,
            activation = 'swish'))

  model.add(Dense(16,
            activation = 'swish'))
  model.add(Dense(2,
            activation = 'swish'))

  model.add(Dense(num_classes, activation='sigmoid'))

  sgd = SGD(lr=0.01, momentum=0.9)

  model.compile(loss='binary_crossentropy', optimizer=sgd,
                metrics=[micro_f1], run_eagerly=True)

  return model
```

[45]:
```
model4 = get_model()

history_relu_2 = model4.fit(X_train, Y_train.ravel(), epochs=20, batch_size=250,
                          verbose=1, validation_split=0.2,
                          callbacks=[CustomCallback(), roc,␣
 ↪tensorboard_callback])

# Test the model after training
test_results = model4.evaluate(X_test, Y_test, verbose=1)

print(f'\nTest results - Loss: {test_results[0]} - micro_f1:␣
 ↪{test_results[1]}%')
```

```
Epoch 1/20
51/52 [============================>.] - ETA: 0s - loss: 0.6857 - micro_f1:
0.3659
Stop testing; got log keys: {'loss': 0.6591938734054565, 'micro_f1':
0.5882662534713745}
52/52 [==============================] - 1s 24ms/step - loss: 0.6855 - micro_f1:
0.3690 - val_loss: 0.6592 - val_micro_f1: 0.5883

End epoch 0 of training; got log keys: {'loss': 0.680046021938324, 'micro_f1':
0.4479726552963257, 'val_loss': 0.6591938734054565, 'val_micro_f1':
0.5882662534713745}
roc-auc_train: 0.6673 - roc-auc_val: 0.6728

 ################################################################################
################################################################################
#########################################

Epoch 2/20
52/52 [==============================] - ETA: 0s - loss: 0.6585 - micro_f1:
0.6135
Stop testing; got log keys: {'loss': 0.6286314129829407, 'micro_f1':
0.6568498015403748}
52/52 [==============================] - 1s 22ms/step - loss: 0.6584 - micro_f1:
```

```
0.6138 - val_loss: 0.6286 - val_micro_f1: 0.6568

End epoch 1 of training; got log keys: {'loss': 0.6509668827056885, 'micro_f1':
0.6282608509063721, 'val_loss': 0.6286314129829407, 'val_micro_f1':
0.6568498015403748}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.721 - roc-auc_val: 0.7252

 ################################################################################
################################################################################
#########################################

Epoch 3/20
52/52 [==============================] - ETA: 0s - loss: 0.6303 - micro_f1:
0.6651
Stop testing; got log keys: {'loss': 0.6232697367668152, 'micro_f1':
0.6844494938850403}
52/52 [==============================] - 1s 22ms/step - loss: 0.6303 - micro_f1:
0.6652 - val_loss: 0.6233 - val_micro_f1: 0.6844

End epoch 2 of training; got log keys: {'loss': 0.6306607127189636, 'micro_f1':
0.67015141248703, 'val_loss': 0.6232697367668152, 'val_micro_f1':
0.6844494938850403}

 custom learning rate by every 3rd epoch 0.009999999776482582

roc-auc_train: 0.7173 - roc-auc_val: 0.7199

 ################################################################################
################################################################################
#########################################

Epoch 4/20
52/52 [==============================] - ETA: 0s - loss: 0.6224 - micro_f1:
0.6865
Stop testing; got log keys: {'loss': 0.6091272234916687, 'micro_f1':
0.6877511143684387}
52/52 [==============================] - 1s 21ms/step - loss: 0.6224 - micro_f1:
0.6864 - val_loss: 0.6091 - val_micro_f1: 0.6878

End epoch 3 of training; got log keys: {'loss': 0.6199597120285034, 'micro_f1':
0.6813885569572449, 'val_loss': 0.6091272234916687, 'val_micro_f1':
0.6877511143684387}
```

```
 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7348 - roc-auc_val: 0.738

 ################################################################################
################################################################################
#########################################

Epoch 5/20
52/52 [==============================] - ETA: 0s - loss: 0.6237 - micro_f1:
0.6756
Stop testing; got log keys: {'loss': 0.6105524897575378, 'micro_f1':
0.6631410121917725}
52/52 [==============================] - 1s 22ms/step - loss: 0.6236 - micro_f1:
0.6757 - val_loss: 0.6106 - val_micro_f1: 0.6631

End epoch 4 of training; got log keys: {'loss': 0.6177284717559814, 'micro_f1':
0.6819562911987305, 'val_loss': 0.6105524897575378, 'val_micro_f1':
0.6631410121917725}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.009499999694526196

roc-auc_train: 0.7354 - roc-auc_val: 0.7399

 ################################################################################
################################################################################
#########################################

Epoch 6/20
52/52 [==============================] - ETA: 0s - loss: 0.6149 - micro_f1:
0.6782
Stop testing; got log keys: {'loss': 0.6061949729919434, 'micro_f1':
0.7038408517837524}
52/52 [==============================] - 1s 23ms/step - loss: 0.6149 - micro_f1:
0.6782 - val_loss: 0.6062 - val_micro_f1: 0.7038

End epoch 5 of training; got log keys: {'loss': 0.6147578358650208, 'micro_f1':
0.6797739863395691, 'val_loss': 0.6061949729919434, 'val_micro_f1':
0.7038408517837524}

 custom learning rate by every 3rd epoch 0.008549999445676804

roc-auc_train: 0.738 - roc-auc_val: 0.7409
```

```
 ################################################################################
################################################################################
#########################################

Epoch 7/20
51/52 [============================>.] - ETA: 0s - loss: 0.6074 - micro_f1:
0.6850
Stop testing; got log keys: {'loss': 0.6053068041801453, 'micro_f1':
0.6933410167694092}
52/52 [==============================] - 1s 24ms/step - loss: 0.6076 - micro_f1:
0.6849 - val_loss: 0.6053 - val_micro_f1: 0.6933

End epoch 6 of training; got log keys: {'loss': 0.6114994287490845, 'micro_f1':
0.6824136972427368, 'val_loss': 0.6053068041801453, 'val_micro_f1':
0.6933410167694092}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.008122499100863934

roc-auc_train: 0.7379 - roc-auc_val: 0.7402

 ################################################################################
################################################################################
#########################################

Epoch 8/20
52/52 [==============================] - ETA: 0s - loss: 0.6138 - micro_f1:
0.6840
Stop testing; got log keys: {'loss': 0.6096577644348145, 'micro_f1':
0.6974152326583862}
52/52 [==============================] - 1s 24ms/step - loss: 0.6138 - micro_f1:
0.6841 - val_loss: 0.6097 - val_micro_f1: 0.6974

End epoch 7 of training; got log keys: {'loss': 0.609895646572113, 'micro_f1':
0.6842409372329712, 'val_loss': 0.6096577644348145, 'val_micro_f1':
0.6974152326583862}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7368 - roc-auc_val: 0.7389

 ################################################################################
################################################################################
#########################################
```

```
Epoch 9/20
52/52 [==============================] - ETA: 0s - loss: 0.6110 - micro_f1:
0.6855
Stop testing; got log keys: {'loss': 0.6018050312995911, 'micro_f1':
0.6826603412628174}
52/52 [==============================] - 1s 23ms/step - loss: 0.6110 - micro_f1:
0.6855 - val_loss: 0.6018 - val_micro_f1: 0.6827

End epoch 8 of training; got log keys: {'loss': 0.6100642085075378, 'micro_f1':
0.6832600235939026, 'val_loss': 0.6018050312995911, 'val_micro_f1':
0.6826603412628174}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.007310249377042055

roc-auc_train: 0.7387 - roc-auc_val: 0.7403

 ################################################################################
################################################################################
#########################################

Epoch 10/20
50/52 [===========================>..] - ETA: 0s - loss: 0.6090 - micro_f1:
0.6709
Stop testing; got log keys: {'loss': 0.6004201769828796, 'micro_f1':
0.6847881078720093}
52/52 [==============================] - 1s 25ms/step - loss: 0.6089 - micro_f1:
0.6714 - val_loss: 0.6004 - val_micro_f1: 0.6848

End epoch 9 of training; got log keys: {'loss': 0.6075038313865662, 'micro_f1':
0.6804603338241577, 'val_loss': 0.6004201769828796, 'val_micro_f1':
0.6847881078720093}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7394 - roc-auc_val: 0.7415

 ################################################################################
################################################################################
#########################################

Epoch 11/20
51/52 [============================>.] - ETA: 0s - loss: 0.6117 - micro_f1:
0.6777
```

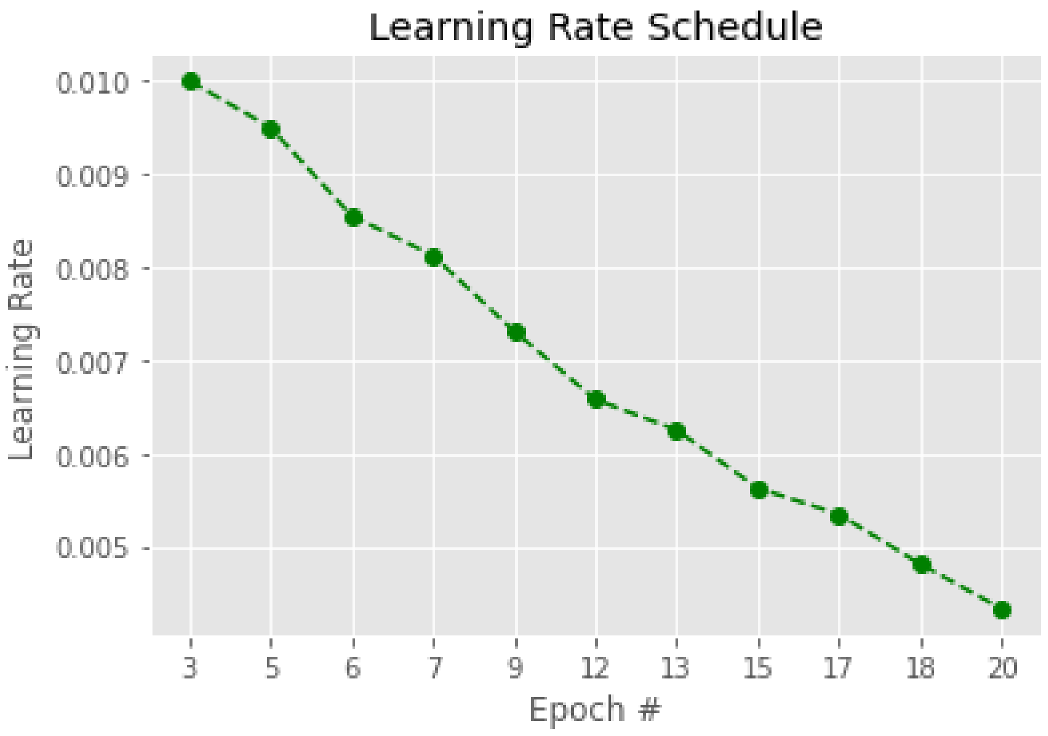
```
Stop testing; got log keys: {'loss': 0.6013289093971252, 'micro_f1':
0.6882352232933044}
52/52 [==============================] - 1s 24ms/step - loss: 0.6115 - micro_f1:
0.6777 - val_loss: 0.6013 - val_micro_f1: 0.6882

End epoch 10 of training; got log keys: {'loss': 0.6068454384803772, 'micro_f1':
0.6787027716636658, 'val_loss': 0.6013289093971252, 'val_micro_f1':
0.6882352232933044}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7393 - roc-auc_val: 0.7411

 ################################################################################
################################################################################
#########################################

Epoch 12/20
52/52 [==============================] - ETA: 0s - loss: 0.6064 - micro_f1:
0.6845
Stop testing; got log keys: {'loss': 0.601469099521637, 'micro_f1':
0.6919034719467163}
52/52 [==============================] - 1s 23ms/step - loss: 0.6064 - micro_f1:
0.6845 - val_loss: 0.6015 - val_micro_f1: 0.6919

End epoch 11 of training; got log keys: {'loss': 0.6076664328575134, 'micro_f1':
0.6818433403968811, 'val_loss': 0.601469099521637, 'val_micro_f1':
0.6919034719467163}

 custom learning rate by every 3rd epoch 0.006579224485903978

roc-auc_train: 0.7393 - roc-auc_val: 0.7408

 ################################################################################
################################################################################
#########################################

Epoch 13/20
51/52 [============================>.] - ETA: 0s - loss: 0.6018 - micro_f1:
0.6848
Stop testing; got log keys: {'loss': 0.6046627163887024, 'micro_f1':
0.6712588667869568}
52/52 [==============================] - 1s 25ms/step - loss: 0.6020 - micro_f1:
0.6846 - val_loss: 0.6047 - val_micro_f1: 0.6713
```

```
End epoch 12 of training; got log keys: {'loss': 0.6051682829856873, 'micro_f1':
0.6801440119743347, 'val_loss': 0.6046627163887024, 'val_micro_f1':
0.6712588667869568}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.006250263191759586

roc-auc_train: 0.7375 - roc-auc_val: 0.7389

 ################################################################################
################################################################################
#########################################

Epoch 14/20
51/52 [============================>.] - ETA: 0s - loss: 0.6053 - micro_f1:
0.6735
Stop testing; got log keys: {'loss': 0.6005087494850159, 'micro_f1':
0.6775618195533752}
52/52 [==============================] - 1s 24ms/step - loss: 0.6053 - micro_f1:
0.6736 - val_loss: 0.6005 - val_micro_f1: 0.6776

End epoch 13 of training; got log keys: {'loss': 0.6052651405334473, 'micro_f1':
0.6748425364494324, 'val_loss': 0.6005087494850159, 'val_micro_f1':
0.6775618195533752}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7396 - roc-auc_val: 0.7415

 ################################################################################
################################################################################
#########################################

Epoch 15/20
52/52 [==============================] - ETA: 0s - loss: 0.6023 - micro_f1:
0.6824
Stop testing; got log keys: {'loss': 0.6017211675643921, 'micro_f1':
0.6870965361595154}
52/52 [==============================] - 1s 26ms/step - loss: 0.6024 - micro_f1:
0.6823 - val_loss: 0.6017 - val_micro_f1: 0.6871

End epoch 14 of training; got log keys: {'loss': 0.6048648357391357, 'micro_f1':
0.6780858635902405, 'val_loss': 0.6017211675643921, 'val_micro_f1':
0.6870965361595154}
```

```
 custom learning rate by every 3rd epoch 0.00562523677945137

roc-auc_train: 0.7384 - roc-auc_val: 0.7406

 ###############################################################################
################################################################################
#########################################

Epoch 16/20
52/52 [==============================] - ETA: 0s - loss: 0.6010 - micro_f1:
0.6777
Stop testing; got log keys: {'loss': 0.6001630425453186, 'micro_f1':
0.6891850829124451}
52/52 [==============================] - 1s 24ms/step - loss: 0.6011 - micro_f1:
0.6777 - val_loss: 0.6002 - val_micro_f1: 0.6892

End epoch 15 of training; got log keys: {'loss': 0.6046549677848816, 'micro_f1':
0.6788519620895386, 'val_loss': 0.6001630425453186, 'val_micro_f1':
0.6891850829124451}

 ###############################################################################
#####################
 Best Weights Saved :
 ###############################################################################
#####################
roc-auc_train: 0.7393 - roc-auc_val: 0.7413

 ###############################################################################
################################################################################
#########################################

Epoch 17/20
52/52 [==============================] - ETA: 0s - loss: 0.6046 - micro_f1:
0.6831
Stop testing; got log keys: {'loss': 0.599748969078064, 'micro_f1':
0.6812810897827148}
52/52 [==============================] - 1s 23ms/step - loss: 0.6046 - micro_f1:
0.6830 - val_loss: 0.5997 - val_micro_f1: 0.6813

End epoch 16 of training; got log keys: {'loss': 0.6040599346160889, 'micro_f1':
0.6783866882324219, 'val_loss': 0.599748969078064, 'val_micro_f1':
0.6812810897827148}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.005343975033611059

roc-auc_train: 0.7395 - roc-auc_val: 0.742
```

```
 ################################################################################
################################################################################
#########################################

Epoch 18/20
52/52 [==============================] - ETA: 0s - loss: 0.6049 - micro_f1:
0.6696
Stop testing; got log keys: {'loss': 0.6005180478096008, 'micro_f1':
0.6776215434074402}
52/52 [==============================] - 1s 23ms/step - loss: 0.6049 - micro_f1:
0.6697 - val_loss: 0.6005 - val_micro_f1: 0.6776

End epoch 17 of training; got log keys: {'loss': 0.6039154529571533, 'micro_f1':
0.6780399084091187, 'val_loss': 0.6005180478096008, 'val_micro_f1':
0.6776215434074402}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.0048095774836838245

roc-auc_train: 0.7397 - roc-auc_val: 0.7411

 ################################################################################
################################################################################
#########################################

Epoch 19/20
51/52 [============================>.] - ETA: 0s - loss: 0.6034 - micro_f1:
0.6745
Stop testing; got log keys: {'loss': 0.6037914156913757, 'micro_f1':
0.686241865158081}
52/52 [==============================] - 1s 24ms/step - loss: 0.6034 - micro_f1:
0.6746 - val_loss: 0.6038 - val_micro_f1: 0.6862

End epoch 18 of training; got log keys: {'loss': 0.6032600998878479, 'micro_f1':
0.6778269410133362, 'val_loss': 0.6037914156913757, 'val_micro_f1':
0.686241865158081}

 ################################################################################
#####################
 Best Weights Saved :
 ################################################################################
#####################
roc-auc_train: 0.7375 - roc-auc_val: 0.7382

 ################################################################################
################################################################################
#########################################
```

```
Epoch 20/20
51/52 [============================>.] - ETA: 0s - loss: 0.6013 - micro_f1:
0.6807
Stop testing; got log keys: {'loss': 0.5985977649688721, 'micro_f1':
0.6815659999847412}
52/52 [==============================] - 1s 22ms/step - loss: 0.6014 - micro_f1:
0.6805 - val_loss: 0.5986 - val_micro_f1: 0.6816

End epoch 19 of training; got log keys: {'loss': 0.6034585237503052, 'micro_f1':
0.6749088764190674, 'val_loss': 0.5985977649688721, 'val_micro_f1':
0.6815659999847412}

 custom learning rate by val_accracy if prev_val_acc > next_val_acc epoch drop
10% : 0.004328619688749313

roc-auc_train: 0.7402 - roc-auc_val: 0.742

 ################################################################################
################################################################################
#########################################

Stop training; got log keys: ['loss', 'micro_f1', 'val_loss', 'val_micro_f1']
125/125 [==============================] - 1s 10ms/step - loss: 0.6006 -
micro_f1: 0.6766

Test results - Loss: 0.6006020307540894 - micro_f1: 0.6765919923782349%
```



#Result

```
[48]: from prettytable import PrettyTable

      # Specify the Column Names while initializing the Table
      myTable = PrettyTable(["Model", "micro-F1", "AUC", 'Accuracy', 'Activation',␣
       ↪'Weight_Initializer'])

      # Add rows
      myTable.add_row(["1", "67 %", "74 %", "68 %", 'tanh', 'RandomUniform'])
      myTable.add_row(["2", "68 %", "74 %", "68 %", 'relu', 'RandomUniform'])
      myTable.add_row(["3", "67 %", "74 %", "68 %", 'relu', 'he_uniform'])
      myTable.add_row(["4", "68 %", "74 %", "68 %", 'relu + selu + swish' ,␣
       ↪'glorot_uniform'])

      print(myTable)
```

```
+-------+----------+------+----------+---------------------+--------------------
+
| Model | micro-F1 | AUC  | Accuracy |      Activation     | Weight_Initializer
|
+-------+----------+------+----------+---------------------+--------------------
+
|   1   |   67 %   | 74 % |   68 %   |         tanh        |   RandomUniform
|
|   2   |   68 %   | 74 % |   68 %   |         relu        |   RandomUniform
|
|   3   |   67 %   | 74 % |   68 %   |         relu        |     he_uniform
|
|   4   |   68 %   | 74 % |   68 %   | relu + selu + swish |   glorot_uniform
|
+-------+----------+------+----------+---------------------+--------------------
+
```