Working_With_Callbacks_Assignement

October 23, 2020

1 Assignment 20

1.1 Import

[1]:
```
%load_ext tensorboard
```

[2]:
```
import tensorflow as tf
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.layers import Input, Dense, Activation, Softmax
from tensorflow.keras.models import Sequential
from tensorflow.keras.metrics import Precision, Recall, AUC
from tensorflow.keras.initializers import RandomUniform, HeUniform, GlorotNormal
from tensorflow.keras.callbacks import Callback, LearningRateScheduler,␣
↪EarlyStopping, TerminateOnNaN, ReduceLROnPlateau
import datetime
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import roc_auc_score, f1_score, accuracy_score
%notebook matplotlib
```

1.2 Loading Data

[3]:
```
df = pd.read_csv('drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/data.csv')
df.head()
```

[3]:

	f1	f2	label
0	0.450564	1.074305	0.0
1	0.085632	0.967682	0.0
2	0.117326	0.971521	1.0

```
3  0.982179 -0.380408    0.0
4 -0.720352  0.955850    0.0
```

1.3 Splitting Data

[5]:
```
x = df[['f1', 'f2']].values
y = df[['label']].values
x_train, x_test, y_train, y_test = train_test_split(x, y, stratify=y,␣
 ↪test_size=0.2)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(16000, 2)
(4000, 2)
(16000, 1)
(4000, 1)
```

1.4 Normalizing the data

[6]:
```
std = StandardScaler()
x_train = std.fit_transform(x_train)
x_test  = std.transform(x_test)
```

1.5 Defining Callback Functions

[7]:
```
# Call back 1
class MetricCallback(Callback):

    # Creating History for keeping track of values
    def on_train_begin(self, logs={}):
        self.history = {}


    def on_epoch_end(self, epoch, logs={}):


        print('\n','='*50)
        print('Ending Epoch', epoch)
        self.history[epoch] = {}
        y_pred = np.asarray(self.model.predict(x_test)).round()
        y_true = np.asarray(y_test)
        self.history[epoch]['val_f1'] = f1_score(y_true, y_pred)
        self.history[epoch]['val_auc'] = roc_auc_score(y_true, y_pred)
```

```
        # Checking the point 5
        if epoch %3 ==0:
            self.model.optimizer.lr = self.model.optimizer.lr.numpy() * 0.95

        # Point 3
        print('\nF1 Score', self.history[epoch]['val_f1'])
        print('\nAUC', self.history[epoch]['val_auc'])

    # Point 6
    def on_batch_end(self, batch, logs={}):
        logs = logs or {}
        loss = logs.get('loss')
        weights = np.array(self.model.get_weights())
        for ele in weights:
            if ele is not None and (np.isnan(ele).any() or np.isinf(ele).any()):
                print('Batch %d: Invalid weights, terminating training' %␣
 ↪(batch))
                self.model.stop_training = True


custom_callback = MetricCallback()
```

[8]:
```
# Model Saving Callback
filepath_model_1="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/model_1/weights_{epoch:02d}_{accuracy:.4f}.hdf5"
model_1_checkpoint = ModelCheckpoint(
                            filepath=filepath_model_1,
                            monitor='val_accuracy',
                            save_frequency="epoch",
                            mode='max',
                            verbose=1)

filepath_model_2="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/model_2/weights_{epoch:02d}_{accuracy:.4f}.hdf5"
model_2_checkpoint = ModelCheckpoint(
                            filepath=filepath_model_2,
                            monitor='val_accuracy',
                            save_frequency="epoch",
                            mode='max',
                            verbose=1
                                )

filepath_model_3="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/model_3/weights_{epoch:02d}_{accuracy:.4f}.hdf5"
```

```
model_3_checkpoint = ModelCheckpoint(
                            filepath=filepath_model_3,
                            monitor='val_accuracy',
                            save_frequency="epoch",
                            mode='max',
                            verbose=1)

filepath_model_4="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/model_4/weights_{epoch:02d}_{accuracy:.4f}.hdf5"
model_4_checkpoint = ModelCheckpoint(
                            filepath=filepath_model_4,
                            monitor='val_accuracy',
                            save_frequency="epoch",
                            mode='max',
                            verbose=1)

# TesorBoard Callback
# Model Log Directory Cleaning

log_dir_model_1="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/logs/model_1/" + datetime.datetime.now().
 ↪strftime("%Y%m%d-%H%M%S")
tensorboard_callback_model_1 = tf.keras.callbacks.TensorBoard(
    log_dir=log_dir_model_1,
    histogram_freq=1,
    write_graph=True,
    write_grads=True
    )



log_dir_model_2="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/logs/model_2/" + datetime.datetime.now().
 ↪strftime("%Y%m%d-%H%M%S")
tensorboard_callback_model_2 = tf.keras.callbacks.TensorBoard(
    log_dir=log_dir_model_2,
    histogram_freq=1,
    write_graph=True,
    write_grads=True
    )


log_dir_model_3="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/logs/model_3/" + datetime.datetime.now().
 ↪strftime("%Y%m%d-%H%M%S")
tensorboard_callback_model_3 = tf.keras.callbacks.TensorBoard(
    log_dir=log_dir_model_3,
```

```
    histogram_freq=1,
    write_graph=True,
    write_grads=True
    )


log_dir_model_4="/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
 ↪assignment20/logs/model_4/" + datetime.datetime.now().
 ↪strftime("%Y%m%d-%H%M%S")
tensorboard_callback_model_4 = tf.keras.callbacks.TensorBoard(
    log_dir=log_dir_model_4,
    histogram_freq=1,
    write_graph=True,
    write_grads=True
    )

# Early Stopping
early_stopping_callback = EarlyStopping(monitor='val_accuracy', patience=2,␣
 ↪verbose=1)

nan_loss = TerminateOnNaN()
reduce_lr = ReduceLROnPlateau(
    monitor='val_accuracy', factor=0.1, patience=1, verbose=1
)
```

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

1.6 Model Skeleton Preperation

[9]:
```
def model_creation(initializer, activation):
    tf.keras.backend.clear_session()
    model = Sequential([
    Input(shape=(2,)),
    Dense(1024, activation=activation, kernel_initializer=initializer),
    Dense(1024, activation=activation, kernel_initializer=initializer),
    Dense(1024, activation=activation, kernel_initializer=initializer),
    Dense(1024, activation=activation, kernel_initializer=initializer),
    Dense(1024, activation=activation, kernel_initializer=initializer),
    Dense(1, activation='sigmoid'),
    ])
```

```
    return model
```

1.7 Variables

[10]:
```
epoch = 30
batch = 512
```

1.8 Model1

1.8.1 Older Logs Cleaning

[11]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/logs/model_1/"
```

[12]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/model_1/"
```

1.8.2 Model Variables

[13]:
```
learning_rate = 0.0009
momentum = 0.09
```

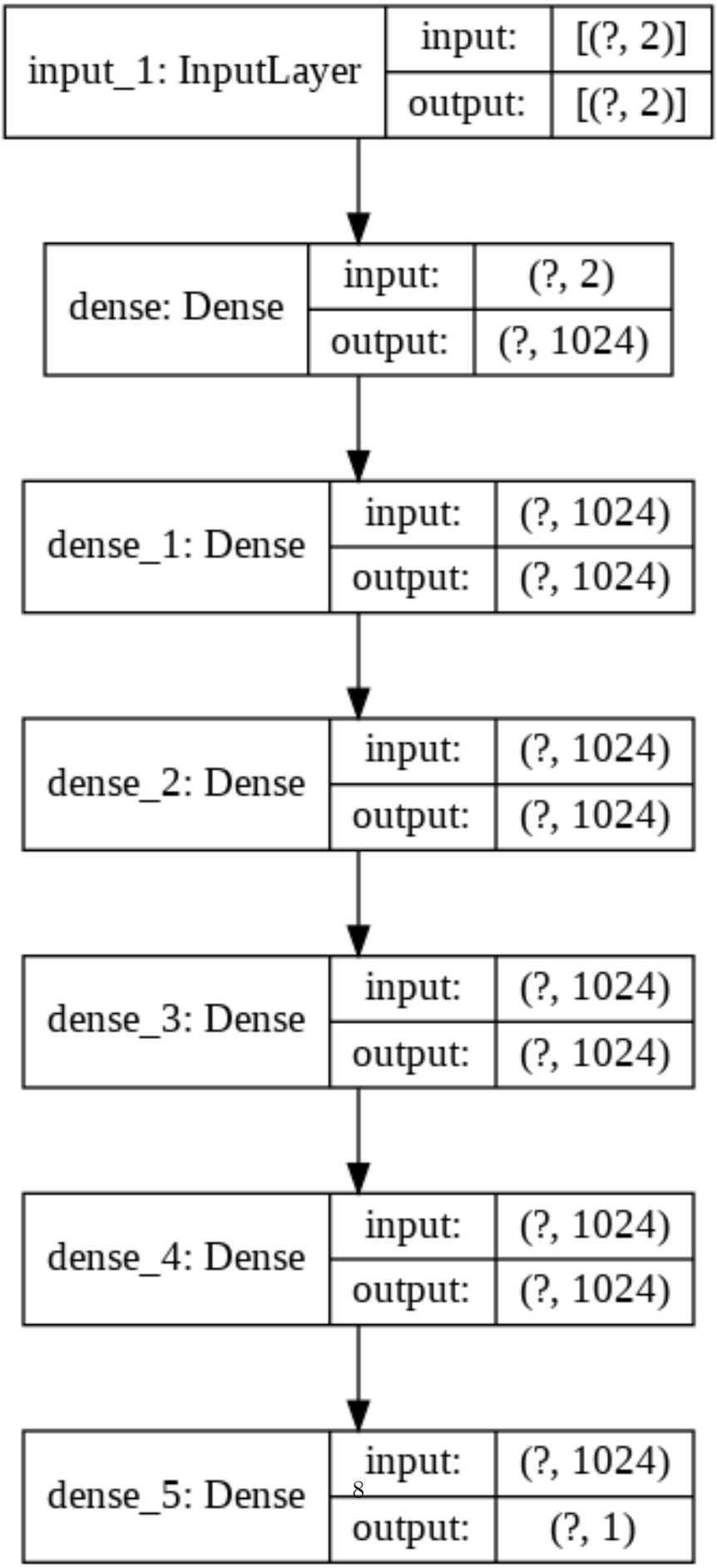
1.8.3 Model Creation

[14]:
```
model_1 = model_creation(initializer=RandomUniform(0,1), activation='tanh')
model_1.summary()
model_1.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=learning_rate,␣
↪momentum=momentum),
    loss=tf.keras.losses.BinaryCrossentropy(),
    metrics='accuracy')
tf.keras.utils.plot_model(model_1,to_file='/tmp/model_1.png', show_shapes=True)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1024)              3072
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              1049600
_________________________________________________________________
```

```
dense_5 (Dense)              (None, 1)                 1025
=================================================================
Total params: 4,202,497
Trainable params: 4,202,497
Non-trainable params: 0
_________________________________________________________________
```

[14]:


input_1: InputLayer
input: [(?, 2)]
output: [(?, 2)]
dense: Dense
input: (?, 2)
output: (?, 1024)
dense_1: Dense
input: (?, 1024)
output: (?, 1024)
dense_2: Dense
input: (?, 1024)
output: (?, 1024)
dense_3: Dense
input: (?, 1024)
output: (?, 1024)
dense_4: Dense
input: (?, 1024)
output: (?, 1024)
dense_5: Dense
input: (?, 1024)
output: (?, 1)

1.8.4 Fitting Model

[15]:
```
# Model Fitting
model_1.fit(
    x_train,
    y_train,
    epochs=epoch,
    validation_data=(x_test, y_test),
    batch_size=batch,
    callbacks=[
               reduce_lr,
               nan_loss,
               custom_callback,
               early_stopping_callback,
               tensorboard_callback_model_1,
               model_1_checkpoint,
    ]

)
```

```
Epoch 1/30
 1/32 […] - ETA: 0s - loss: 0.7195 - accuracy:
0.5332WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from
tensorflow.python.eager.profiler) is deprecated and will be removed after
2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
 2/32 [>…] - ETA: 1s - loss: 0.7299 - accuracy:
0.5020WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared
to the batch time (batch time: 0.0115s vs `on_train_batch_end` time: 0.0680s).
Check your callbacks.
31/32 [============================>.] - ETA: 0s - loss: 0.6967 - accuracy:
0.5024
 ================================================
Ending Epoch 0

F1 Score 0.5017421602787456

AUC 0.49950000000000006

Epoch 00001: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_1/weights_01_0.5024.hdf5
32/32 [==============================] - 2s 48ms/step - loss: 0.6967 - accuracy:
0.5024 - val_loss: 0.6936 - val_accuracy: 0.4995
```

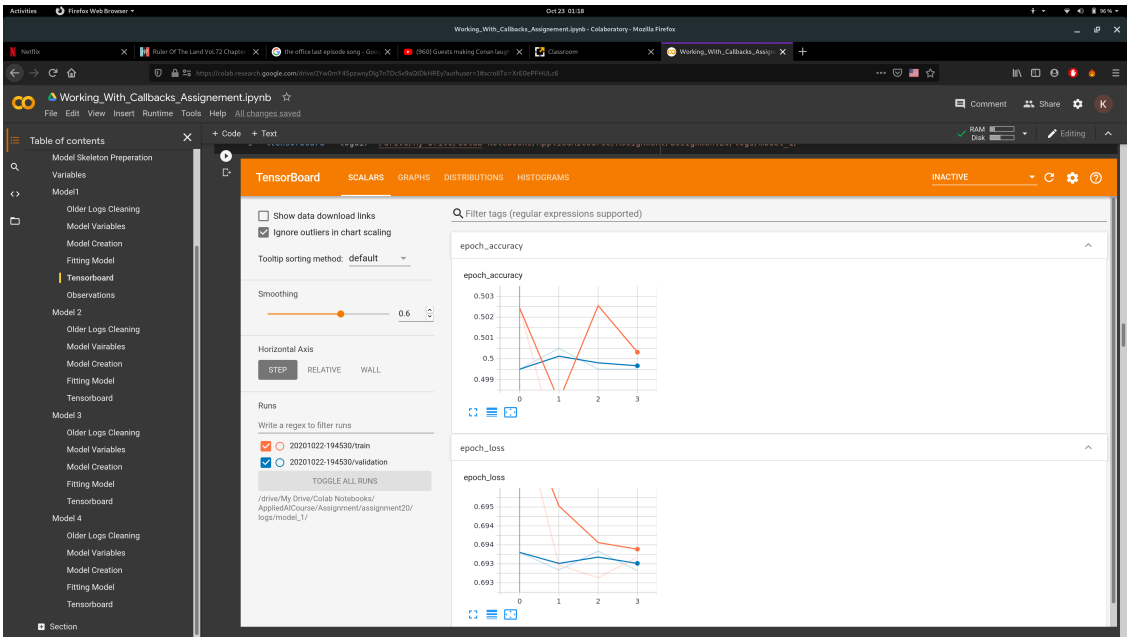
```
Epoch 2/30
31/32 [============================>.] - ETA: 0s - loss: 0.6934 - accuracy:
0.4951
 ================================================
Ending Epoch 1

F1 Score 0.4982420894023104

AUC 0.5005000000000001

Epoch 00002: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_1/weights_02_0.4953.hdf5
32/32 [==============================] - 1s 38ms/step - loss: 0.6934 - accuracy:
0.4953 - val_loss: 0.6933 - val_accuracy: 0.5005
Epoch 3/30
32/32 [==============================] - ETA: 0s - loss: 0.6931 - accuracy:
0.5069
Epoch 00003: ReduceLROnPlateau reducing learning rate to 8.549999911338091e-05.

 ================================================
Ending Epoch 2

F1 Score 0.5017421602787456

AUC 0.49950000000000006

Epoch 00003: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_1/weights_03_0.5069.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6931 - accuracy:
0.5069 - val_loss: 0.6937 - val_accuracy: 0.4995
Epoch 4/30
32/32 [==============================] - ETA: 0s - loss: 0.6935 - accuracy:
0.4977
Epoch 00004: ReduceLROnPlateau reducing learning rate to 8.549999620299786e-06.

 ================================================
Ending Epoch 3

F1 Score 0.5017421602787456

AUC 0.49950000000000006

Epoch 00004: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_1/weights_04_0.4977.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6935 - accuracy:
0.4977 - val_loss: 0.6933 - val_accuracy: 0.4995
Epoch 00004: early stopping
```

[15]: <tensorflow.python.keras.callbacks.History at 0x7f5bf0469320>

1.8.5 Tensorboard

[16]:
```
%tensorboard --logdir "/drive/My Drive/Colab Notebooks/AppliedAICourse/
↪Assignment/assignment20/logs/model_1/"
```

<IPython.core.display.Javascript object>



1.8.6 Observations

- F1 Score 0.501
- AUC 0.499
- Not a very good model due to random uniform intitalization and tanh activation

1.9 Model 2

1.9.1 Older Logs Cleaning

[17]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/logs/model_2/"
```

[18]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/model_2/"
```

1.9.2 Model Vairables

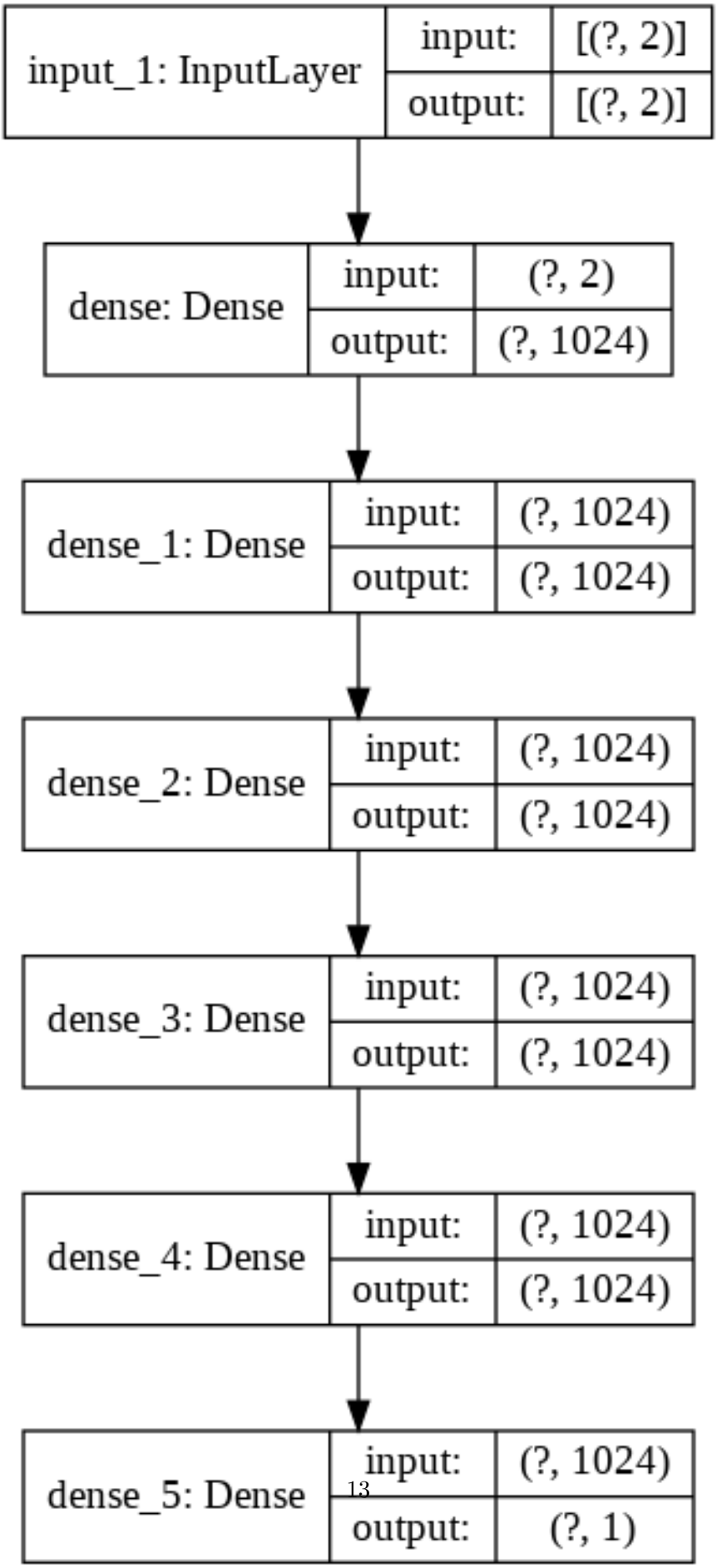
[19]:
```
learning_rate = 0.01
momentum = 0.09
```

1.9.3 Model Creation

[20]:
```
model_2 = model_creation(initializer=RandomUniform(0,1), activation='relu')
model_2.summary()
model_2.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=learning_rate,␣
 ↪momentum=momentum),
    loss=tf.keras.losses.BinaryCrossentropy(),
    metrics=[
            'accuracy'
            ]
            )
tf.keras.utils.plot_model(model_2,to_file='/tmp/model_2.png', show_shapes=True)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1024)              3072
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 1025
=================================================================
Total params: 4,202,497
Trainable params: 4,202,497
Non-trainable params: 0
_________________________________________________________________
```

[20]:


input_1: InputLayer
input: [(?, 2)]
output: [(?, 2)]
dense: Dense
input: (?, 2)
output: (?, 1024)
dense_1: Dense
input: (?, 1024)
output: (?, 1024)
dense_2: Dense
input: (?, 1024)
output: (?, 1024)
dense_3: Dense
input: (?, 1024)
output: (?, 1024)
dense_4: Dense
input: (?, 1024)
output: (?, 1024)
dense_5: Dense
input: (?, 1024)
output: (?, 1)

1.9.4 Fitting Model

[21]:
```
# Model Fitting
model_2.fit(
    x_train,
    y_train,
    epochs=epoch,
    validation_data=(x_test, y_test),
    batch_size=batch,
    callbacks=[
               reduce_lr,
               nan_loss,
               custom_callback,
               early_stopping_callback,
               tensorboard_callback_model_2,
               model_2_checkpoint,
    ]
)
```

```
Epoch 1/30
 2/32 [>…] - ETA: 1s - loss: 666334789632.0000 -
accuracy: 0.5186WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow
compared to the batch time (batch time: 0.0069s vs `on_train_batch_end` time:
0.0692s). Check your callbacks.
30/32 [===========================>..] - ETA: 0s - loss: 44422320128.0000 -
accuracy: 0.4993
 ================================================
Ending Epoch 0

F1 Score 0.6666666666666666

AUC 0.5

Epoch 00001: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_2/weights_01_0.5010.hdf5
32/32 [==============================] - 1s 43ms/step - loss: 42645426176.0000 -
accuracy: 0.5010 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 2/30
32/32 [==============================] - ETA: 0s - loss: 0.6931 - accuracy:
0.5000
Epoch 00002: ReduceLROnPlateau reducing learning rate to 0.0009499999694526196.

 ================================================
Ending Epoch 1
```

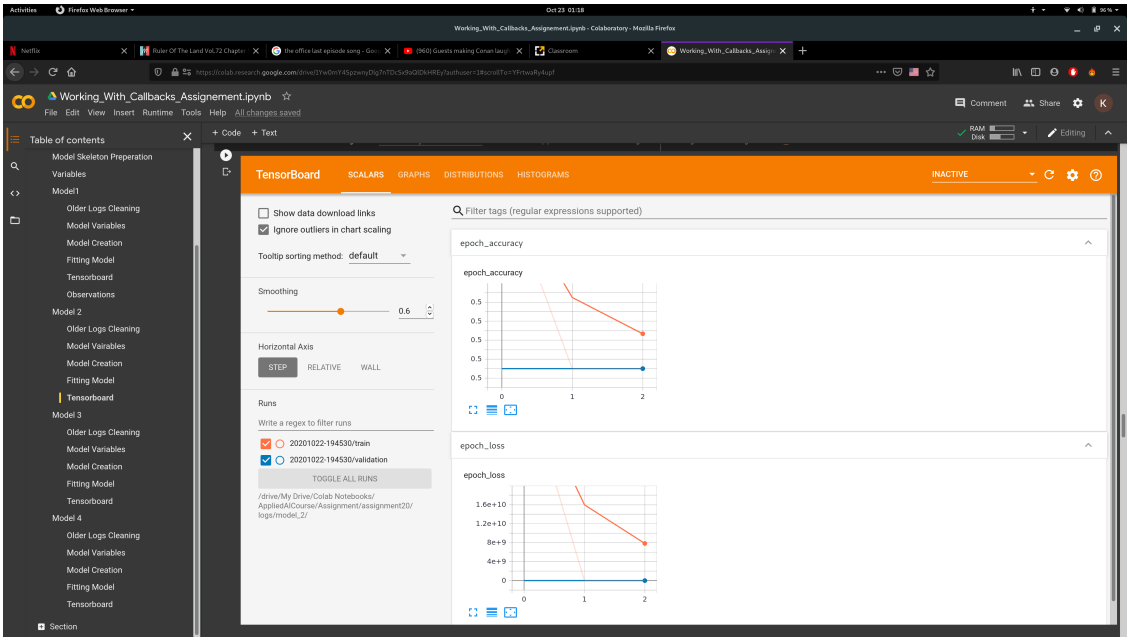
```
F1 Score 0.6666666666666666

AUC 0.5

Epoch 00002: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_2/weights_02_0.5000.hdf5
32/32 [==============================] - 1s 36ms/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 3/30
32/32 [==============================] - ETA: 0s - loss: 0.6931 - accuracy:
0.5000
Epoch 00003: ReduceLROnPlateau reducing learning rate to 9.499999578110874e-05.

 ================================================
Ending Epoch 2

F1 Score 0.6666666666666666

AUC 0.5

Epoch 00003: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_2/weights_03_0.5000.hdf5
32/32 [==============================] - 1s 35ms/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 00003: early stopping
```

[21]:
```
<tensorflow.python.keras.callbacks.History at 0x7f5bf0267240>
```

1.9.5 Tensorboard

[22]:
```
%tensorboard --logdir "/drive/My Drive/Colab Notebooks/AppliedAICourse/
↪Assignment/assignment20/logs/model_2/"
```

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



1.9.6 Observation

- F1 Score: 0.667
- AUC Score: 0.5
- Slightly better model from model1 with change in activation to relu

1.10 Model 3

1.10.1 Older Logs Cleaning

[23]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/logs/model_3/"
```

[24]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/model_3/"
```

1.10.2 Model Variables

[25]:
```
learning_rate = 0.09
momentum = 0.09
```

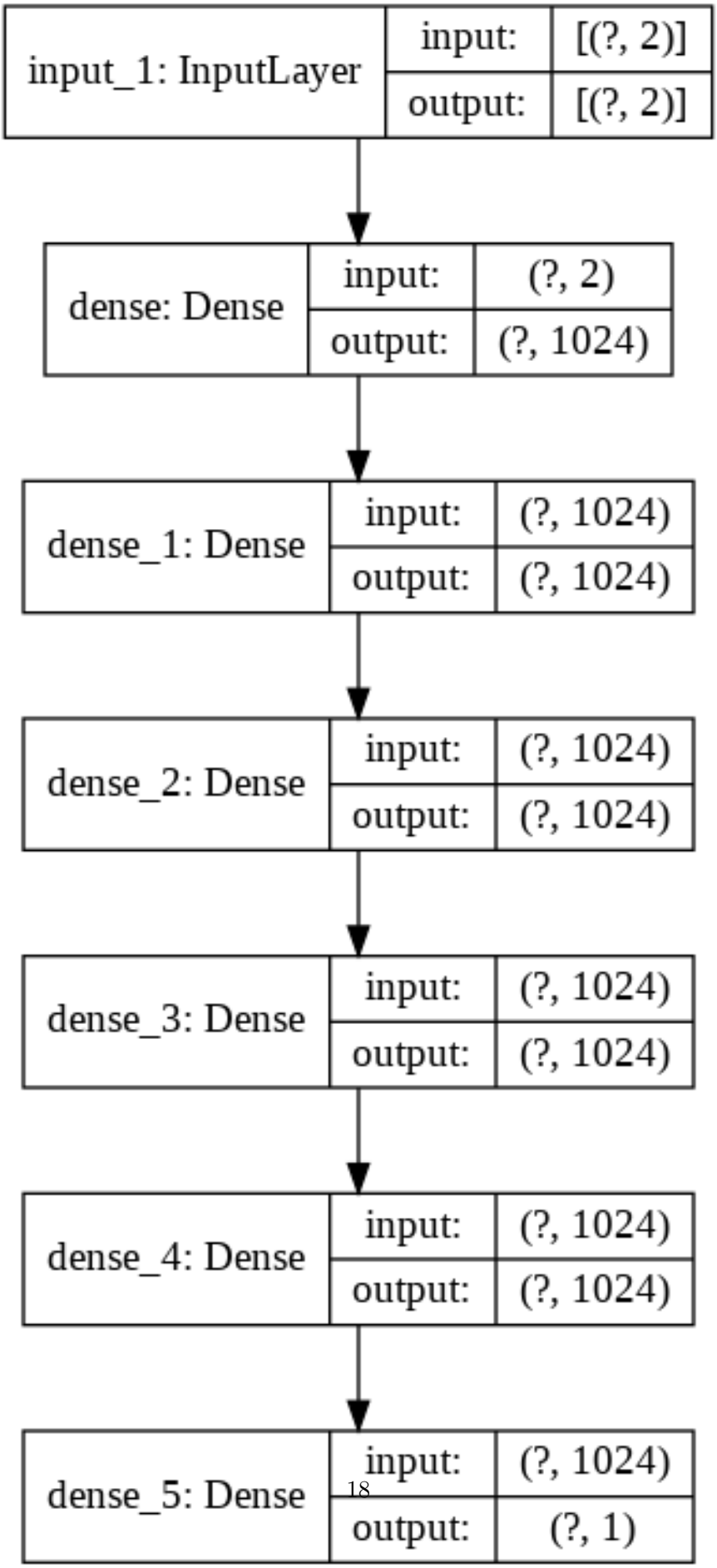
1.10.3 Model Creation

[26]:
```
model_3 = model_creation(initializer=HeUniform(), activation='relu')
model_3.summary()
model_3.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=learning_rate,␣
↪momentum=momentum),
    loss=tf.keras.losses.BinaryCrossentropy(),
```

```
    metrics=[
            'accuracy'
            ]
            )
tf.keras.utils.plot_model(model_2,to_file='/tmp/model_3.png', show_shapes=True)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1024)              3072
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 1025
=================================================================
Total params: 4,202,497
Trainable params: 4,202,497
Non-trainable params: 0
_________________________________________________________________
```

[26]:


input_1: InputLayer
input: [(?, 2)]
output: [(?, 2)]
dense: Dense
input: (?, 2)
output: (?, 1024)
dense_1: Dense
input: (?, 1024)
output: (?, 1024)
dense_2: Dense
input: (?, 1024)
output: (?, 1024)
dense_3: Dense
input: (?, 1024)
output: (?, 1024)
dense_4: Dense
input: (?, 1024)
output: (?, 1024)
dense_5: Dense
input: (?, 1024)
output: (?, 1)

1.10.4 Fitting Model

[27]:
```
# Model Fitting
model_3.fit(
    x_train,
    y_train,
    epochs=epoch,
    validation_data=(x_test, y_test),
    batch_size=batch,
    callbacks=[
                reduce_lr,
                nan_loss,
                custom_callback,
                early_stopping_callback,
                tensorboard_callback_model_3,
                model_3_checkpoint,
    ]

)
```

```
Epoch 1/30
 2/32 [>…] - ETA: 1s - loss: 18.3233 - accuracy:
0.4941WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared
to the batch time (batch time: 0.0065s vs `on_train_batch_end` time: 0.0768s).
Check your callbacks.
30/32 [===========================>..] - ETA: 0s - loss: 1.9734 - accuracy:
0.5057
 ================================================
Ending Epoch 0

F1 Score 0.002994011976047905

AUC 0.5005000000000001

Epoch 00001: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_01_0.5092.hdf5
32/32 [==============================] - 1s 45ms/step - loss: 1.9215 - accuracy:
0.5092 - val_loss: 0.6949 - val_accuracy: 0.5005
Epoch 2/30
30/32 [===========================>..] - ETA: 0s - loss: 0.6798 - accuracy:
0.5490
 ================================================
Ending Epoch 1

F1 Score 0.06913106096975516
```

```
AUC 0.51525

Epoch 00002: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_02_0.5505.hdf5
32/32 [==============================] - 1s 38ms/step - loss: 0.6794 - accuracy:
0.5505 - val_loss: 0.6830 - val_accuracy: 0.5153
Epoch 3/30
30/32 [===========================>..] - ETA: 0s - loss: 0.6722 - accuracy:
0.5840
 ================================================
Ending Epoch 2

F1 Score 0.5662555759643139

AUC 0.5867499999999999

Epoch 00003: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_03_0.5841.hdf5
32/32 [==============================] - 1s 38ms/step - loss: 0.6721 - accuracy:
0.5841 - val_loss: 0.6741 - val_accuracy: 0.5867
Epoch 4/30
30/32 [===========================>..] - ETA: 0s - loss: 0.6645 - accuracy:
0.6155
Epoch 00004: ReduceLROnPlateau reducing learning rate to 0.008550000190734864.

 ================================================
Ending Epoch 3

F1 Score 0.19675010979358806

AUC 0.5427500000000001

Epoch 00004: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_04_0.6170.hdf5
32/32 [==============================] - 1s 40ms/step - loss: 0.6641 - accuracy:
0.6170 - val_loss: 0.6714 - val_accuracy: 0.5428
Epoch 5/30
31/32 [============================>.] - ETA: 0s - loss: 0.6593 - accuracy:
0.6035
 ================================================
Ending Epoch 4

F1 Score 0.5014846585285384

AUC 0.62225

Epoch 00005: saving model to /drive/My Drive/Colab
```

```
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_05_0.6033.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6594 - accuracy:
0.6033 - val_loss: 0.6622 - val_accuracy: 0.6223
Epoch 6/30
31/32 [============================>.] - ETA: 0s - loss: 0.6570 - accuracy:
0.6368
 ================================================
Ending Epoch 5

F1 Score 0.5243664717348928

AUC 0.634

Epoch 00006: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_06_0.6366.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6571 - accuracy:
0.6366 - val_loss: 0.6613 - val_accuracy: 0.6340
Epoch 7/30
31/32 [============================>.] - ETA: 0s - loss: 0.6560 - accuracy:
0.6400
Epoch 00007: ReduceLROnPlateau reducing learning rate to 0.0008122500032186509.

 ================================================
Ending Epoch 6

F1 Score 0.5346848273677542

AUC 0.63275

Epoch 00007: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_07_0.6392.hdf5
32/32 [==============================] - 1s 38ms/step - loss: 0.6563 - accuracy:
0.6392 - val_loss: 0.6605 - val_accuracy: 0.6327
Epoch 8/30
30/32 [===========================>..] - ETA: 0s - loss: 0.6560 - accuracy:
0.6490
Epoch 00008: ReduceLROnPlateau reducing learning rate to 7.716375403106214e-05.

 ================================================
Ending Epoch 7

F1 Score 0.529073482428115

AUC 0.6315

Epoch 00008: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_3/weights_08_0.6498.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6558 - accuracy:
```

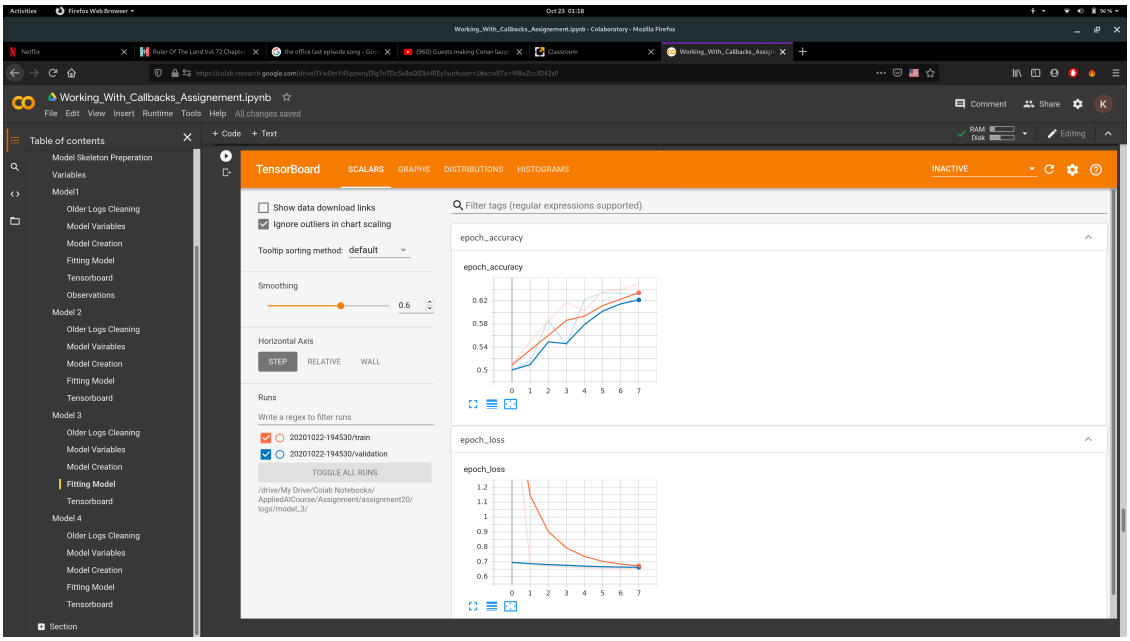
```
0.6498 - val_loss: 0.6605 - val_accuracy: 0.6315
Epoch 00008: early stopping
```

[27]:
```
<tensorflow.python.keras.callbacks.History at 0x7f5b927e3438>
```

1.10.5 Tensorboard

[28]:
```
%tensorboard --logdir "/drive/My Drive/Colab Notebooks/AppliedAICourse/
↪Assignment/assignment20/logs/model_3/"
```

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



1.10.6 Observations

- F1 Score 0.529
- AUC 0.6315
- with the change in initialization and activation better model from model 1

1.11 Model 4

1.11.1 Older Logs Cleaning

[29]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/logs/model_4/"
```

[30]:
```
!rm -rf "/drive/My Drive/Colab Notebooks/AppliedAICourse/Assignment/
↪assignment20/model_4/"
```

1.11.2 Model Variables

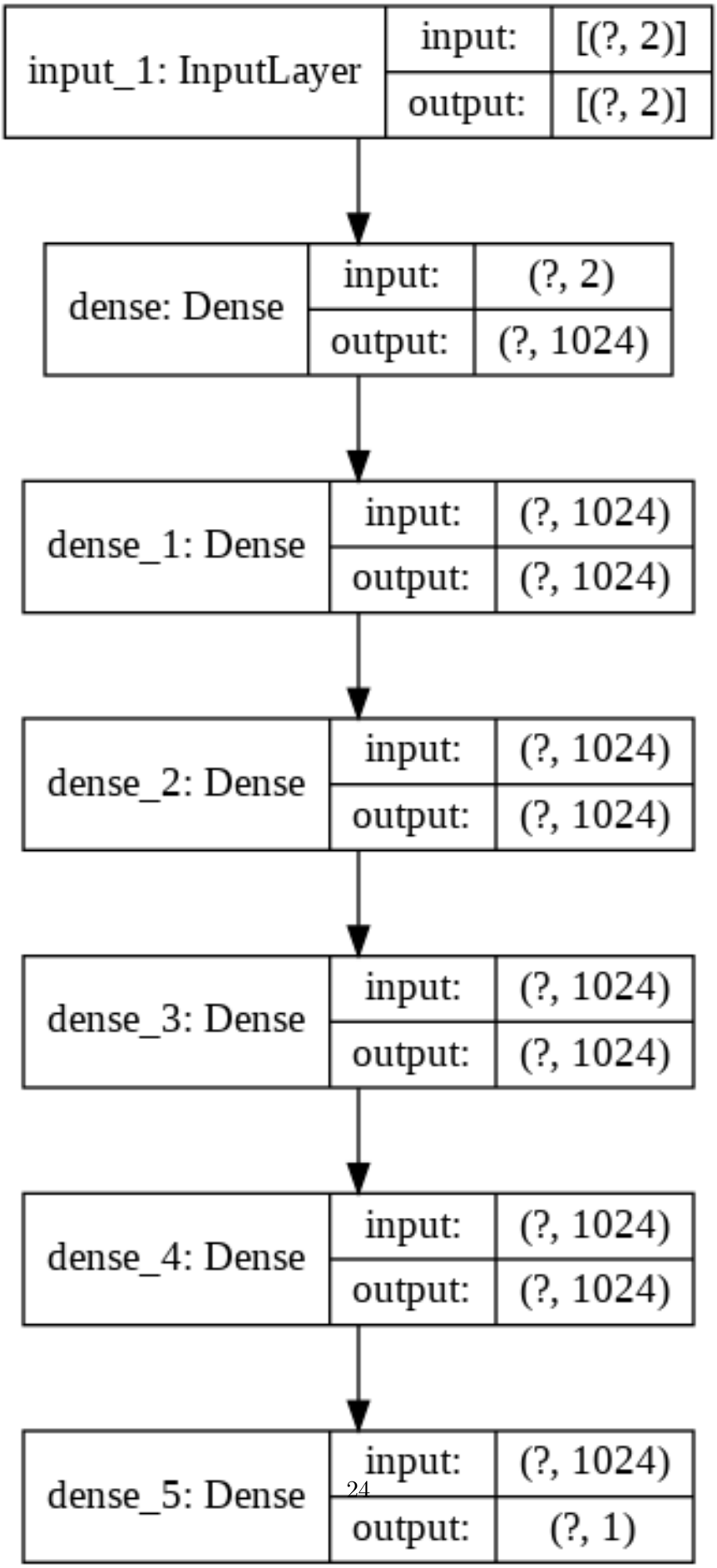
[31]:
```
learning_rate = 0.0001
```

1.11.3 Model Creation

[32]:
```
model_4 = model_creation(initializer=GlorotNormal(), activation='relu')
model_4.summary()
optimizer = tf.keras.optimizers.Adam(learning_rate)
model_4.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.BinaryCrossentropy(),
    metrics=[
             'accuracy'
            ]
            )
tf.keras.utils.plot_model(model_4,to_file='/tmp/model_4.png', show_shapes=True)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1024)              3072
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 1025
=================================================================
Total params: 4,202,497
Trainable params: 4,202,497
Non-trainable params: 0
_________________________________________________________________
```

[32]:


input_1: InputLayer
input: [(?, 2)]
output: [(?, 2)]
dense: Dense
input: (?, 2)
output: (?, 1024)
dense_1: Dense
input: (?, 1024)
output: (?, 1024)
dense_2: Dense
input: (?, 1024)
output: (?, 1024)
dense_3: Dense
input: (?, 1024)
output: (?, 1024)
dense_4: Dense
input: (?, 1024)
output: (?, 1024)
dense_5: Dense
input: (?, 1024)
output: (?, 1)

1.11.4 Fitting Model

[33]:
```
# Model Fitting
model_4.fit(
    x_train,
    y_train,
    epochs=epoch,
    validation_data=(x_test, y_test),
    batch_size=batch,
    callbacks=[
                reduce_lr,
                nan_loss,
                custom_callback,
                early_stopping_callback,
                tensorboard_callback_model_4,
                model_4_checkpoint,
    ]

)
```

```
Epoch 1/30
 2/32 [>…] - ETA: 1s - loss: 0.6900 - accuracy:
0.5195WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared
to the batch time (batch time: 0.0083s vs `on_train_batch_end` time: 0.0729s).
Check your callbacks.
31/32 [============================>.] - ETA: 0s - loss: 0.6862 - accuracy:
0.5018
 ================================================
Ending Epoch 0

F1 Score 0.02759980285855101

AUC 0.50675

Epoch 00001: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_4/weights_01_0.5014.hdf5
32/32 [==============================] - 2s 49ms/step - loss: 0.6862 - accuracy:
0.5014 - val_loss: 0.6822 - val_accuracy: 0.5067
Epoch 2/30
32/32 [==============================] - ETA: 0s - loss: 0.6694 - accuracy:
0.5781
 ================================================
Ending Epoch 1

F1 Score 0.3063864187550525
```

```
AUC 0.5710000000000001

Epoch 00002: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_4/weights_02_0.5781.hdf5
32/32 [==============================] - 1s 41ms/step - loss: 0.6694 - accuracy:
0.5781 - val_loss: 0.6631 - val_accuracy: 0.5710
Epoch 3/30
30/32 [===========================>..] - ETA: 0s - loss: 0.6311 - accuracy:
0.6559
 ================================================
Ending Epoch 2

F1 Score 0.6148648648648649

AUC 0.658

Epoch 00003: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_4/weights_03_0.6578.hdf5
32/32 [==============================] - 1s 41ms/step - loss: 0.6287 - accuracy:
0.6578 - val_loss: 0.6230 - val_accuracy: 0.6580
Epoch 4/30
30/32 [===========================>..] - ETA: 0s - loss: 0.6059 - accuracy:
0.6692
 ================================================
Ending Epoch 3

F1 Score 0.6303395399780942

AUC 0.6625

Epoch 00004: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_4/weights_04_0.6693.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6060 - accuracy:
0.6693 - val_loss: 0.6166 - val_accuracy: 0.6625
Epoch 5/30
31/32 [============================>.] - ETA: 0s - loss: 0.6007 - accuracy:
0.6740
Epoch 00005: ReduceLROnPlateau reducing learning rate to 9.02499959920533e-06.

 ================================================
Ending Epoch 4

F1 Score 0.6220624827204866

AUC 0.65825

Epoch 00005: saving model to /drive/My Drive/Colab
```

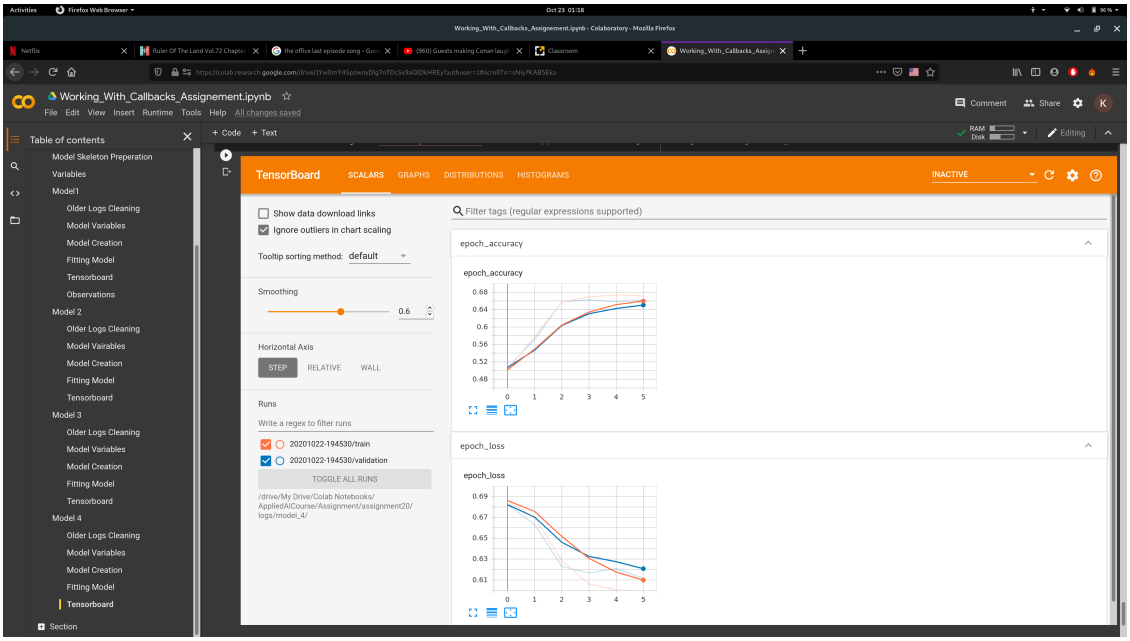
```
Notebooks/AppliedAICourse/Assignment/assignment20/model_4/weights_05_0.6740.hdf5
32/32 [==============================] - 1s 39ms/step - loss: 0.6005 - accuracy:
0.6740 - val_loss: 0.6209 - val_accuracy: 0.6582
Epoch 6/30
30/32 [===========================>..] - ETA: 0s - loss: 0.5988 - accuracy:
0.6718
Epoch 00006: ReduceLROnPlateau reducing learning rate to 9.024999599205331e-07.

 ================================================
Ending Epoch 5

F1 Score 0.6481288981288982

AUC 0.6615

Epoch 00006: saving model to /drive/My Drive/Colab
Notebooks/AppliedAICourse/Assignment/assignment20/model_4/weights_06_0.6714.hdf5
32/32 [==============================] - 1s 40ms/step - loss: 0.5992 - accuracy:
0.6714 - val_loss: 0.6115 - val_accuracy: 0.6615
Epoch 00006: early stopping
```

[33]:
```
<tensorflow.python.keras.callbacks.History at 0x7f5b92e20860>
```

1.11.5 Tensorboard

[34]:
```
%tensorboard --logdir "/drive/My Drive/Colab Notebooks/AppliedAICourse/
↪Assignment/assignment20/logs/model_4/"
```

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



1.11.6 Observations

- F1 Score 0.648
- AUC 0.661
- Best model among all due to better intialization of weights and adam optimizer