In [2]:

```
from google.colab import drive
drive.mount('gdrive',force_remount=True)
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6
qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%
b&scope=email%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.
2Fauth%2Fdrive%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2Fww
ogleapis.com%2Fauth%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:
··········
Mounted at gdrive
```

In [0]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [0]:

```
# Importing libraries
import pandas as pd
import numpy as np

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
import tensorflow as tf
from sklearn.metrics import roc_auc_score
from sklearn.metrics import confusion_matrix, f1_score, precision_score, recall_score
import matplotlib.pyplot as plt
from tqdm import tqdm
import datetime
```

In [5]:

```
import keras
from sklearn.preprocessing import StandardScaler
from sklearn.feature_extraction.text import TfidfVectorizer
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.initializers import RandomUniform
from keras import initializers, optimizers
from keras.utils import np_utils
from keras.callbacks import ModelCheckpoint, EarlyStopping,LearningRateScheduler,TensorBoard
```

```
Using TensorFlow backend.
```

In [0]:

```
# magic function
%load_ext tensorboard
```

In [0]:

```
data=pd.read_csv('gdrive/My Drive/data/callbacks/data (1).csv')
```

In [8]:

```
data.head(3)
```

Out[8]:

	f1	f2	label

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

In [9]:

```
data['label'].value_counts()
```

Out[9]:

```
1.0    10000
0.0    10000
Name: label, dtype: int64
```

We have a balanced dataset

In [0]:

```
Y=data['label'].values
X=data[['f1','f2']]
```

In [11]:

```
# Train and test split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,stratify=Y,test_size=0.2,random_state=2)
print('train_size:',len(X_train))
print('test_size:',len(X_test))
```

```
train_size: 16000
test_size: 4000
```

In [12]:

```
X_tr=np.hstack((X_train['f1'].values.reshape(-1,1),X_train['f2'].values.reshape(-1,1)))
print(X_tr.shape)
X_te=np.hstack((X_test['f1'].values.reshape(-1,1),X_test['f2'].values.reshape(-1,1)))
print(X_te.shape)
```

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

In [13]:

```
Y_train = np_utils.to_categorical(Y_train, 2)
Y_test = np_utils.to_categorical(Y_test, 2)

print("After converting the output into a vector : ",Y_train[0])
```

```
After converting the output into a vector :  [1. 0.]
```

In [14]:

```
# Defining  model 1
keras.backend.clear_session()
model1=Sequential()
model1.add(Dense(50,activation='tanh',input_shape=(X_tr.shape[1],),kernel_initializer=initializers.
RandomUniform(minval=0, maxval=1, seed=None)))

model1.add(Dense(40,activation='tanh',kernel_initializer=initializers.RandomUniform(minval=0, maxva
l=1, seed=None)))

model1.add(Dense(30,activation='tanh',kernel_initializer=initializers.RandomUniform(minval=0, maxva
l=1, seed=None)))

model1.add(Dense(20,activation='tanh',kernel_initializer=initializers.RandomUniform(minval=0, maxva
```

```
l=1, seed=None)))

model1.add(Dense(10,activation='tanh',kernel_initializer=initializers.RandomUniform(minval=0, maxva
l=1, seed=None)))
model1.add(Dense(2,activation='softmax'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:107: The name tf.reset_default_graph is deprecated. P
lease use tf.compat.v1.reset_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:111: The name tf.placeholder_with_default is
deprecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph is deprecated. Plea
se use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is deprecated. Please us
e tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. Pleas
e use tf.random.uniform instead.
```

In [0]:

```
# Function to get 'auc' score
def au_roc(y_true, y_pred):
  try:
      return tf.py_func(roc_auc_score, (y_true, y_pred), tf.double)
  except ValueError:
    pass
```

In [0]:

```
# Function to get 'f1_score' score
import keras.backend as K
def f1_metric(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)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    recall = true_positives / (possible_positives + K.epsilon())
    f1_val = 2*(precision*recall)/(precision+recall+K.epsilon())
    return f1_val
```

In [0]:

```
# Storing the weights if the 'val_acc' is improving
callbacks = ModelCheckpoint(
        filepath='gdrive/My Drive/data/callbacks/training_1-{epoch:02d}-{val_acc:.4f}.hdf5',
        # Path where to save the model
        # The two parameters below mean that we will overwrite
        # the current checkpoint if and only if
        # the `val_acc` score has improved.
        save_best_only=True,
        monitor='val_acc',
        verbose=1)
# Class to change the learning rate
class LossHistory(tf.keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        ## on begin of training, we are creating a instance varible called history
        ## it is a dict with keys [loss, acc, val_loss, val_acc]
        self.history={'loss': [],'acc': [],'val_loss': [],'val_acc': []}

    def on_epoch_end(self, epoch, logs={}):
        ## on end of each epoch, we will get logs and update the self.history dict
        self.history['loss'].append(logs.get('loss'))
        self.history['acc'].append(logs.get('acc'))

        if logs.get('val_loss', -1) != -1:
```

```
        if logs.get('val_loss', -1) != -1:
            self.history['val_loss'].append(logs.get('val_loss'))
        if logs.get('val_acc', -1) != -1:
            self.history['val_acc'].append(logs.get('val_acc'))
        current_score = logs.get('val_acc')
        #Updating "learning rate"
        if current_score > self.history['val_acc'][epoch-1]:
            if (epoch+1)%3==0:
                #lr= self.model.optimizer.lr*0.90 # will not update
                K.set_value(self.model.optimizer.lr, 0.95 * K.get_value(self.model.optimizer.lr))#
update the learning rate,decreasing by 5%
            else:
                #lr= self.model.optimizer.lr*0.90 # will not update
                K.set_value(self.model.optimizer.lr, 0.90 * K.get_value(self.model.optimizer.lr))#
update the learning rate,decreasing by 10%
                # tf.keras.backend.set_value(self.model.optimizer.lr, scheduled_lr) # Referring the
Tensorflow Documentation

        rate=K.eval(self.model.optimizer.lr)
        print('Changed learning rate:',rate)
        # Stop training if "validation accuracy" is not improving
        if epoch>0:
            if  self.history['val_acc'][epoch-1]==self.history['val_acc'][epoch]:
               print("'validation accuracy' is not decreased in last 2 epochs {}".format(epoch))
               self.model.stop_training = True
        # Terminating the traning if "loss" is Nan
        loss = logs.get('loss')
        if loss is not None:
            if np.isnan(loss) or np.isinf(loss):
                print("Invalid loss and terminated at epoch {}".format(epoch))
                self.model.stop_training = True
# Creating object to change the learning rate
lrschedule=LossHistory()
```

In [0]:

```
# Tensorboard
log_dir="gdrive/My Drive/data/callbacks/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True,write_grads=
True)
```

In [21]:

```
# Compiling model 1
sgd_with_momemtum = optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model1.compile(optimizer=sgd_with_momemtum, loss='categorical_crossentropy', metrics=['accuracy',f
1_metric,au_roc])
history=model1.fit(X_tr,Y_train,validation_data=(X_te,Y_test),epochs=30,batch_size=300,verbose=1,ca
llbacks=[callbacks,lrschedule,tensorboard_callback])
```

```
Train on 16000 samples, validate on 4000 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1068: The name t
f.summary.histogram is deprecated. Please use tf.compat.v1.summary.histogram instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name t
f.summary.merge_all is deprecated. Please use tf.compat.v1.summary.merge_all instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name t
f.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/30
16000/16000 [==============================] - 1s 47us/step - loss: 0.6942 - acc: 0.5017 -
f1_metric: 0.5017 - au_roc: 0.5018 - val_loss: 0.6931 - val_acc: 0.5035 - val_f1_metric: 0.5035 -
val_au_roc: 0.5031
Changed learning rate: 0.01
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name t
f.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/30
16000/16000 [==============================] - 0s 28us/step - loss: 0.6933 - acc: 0.5069 -
f1_metric: 0.5069 - au_roc: 0.5078 - val_loss: 0.6932 - val_acc: 0.5000 - val_f1_metric: 0.5000 -
val_au_roc: 0.4980
Changed learning rate: 0.01
Epoch 3/30
```

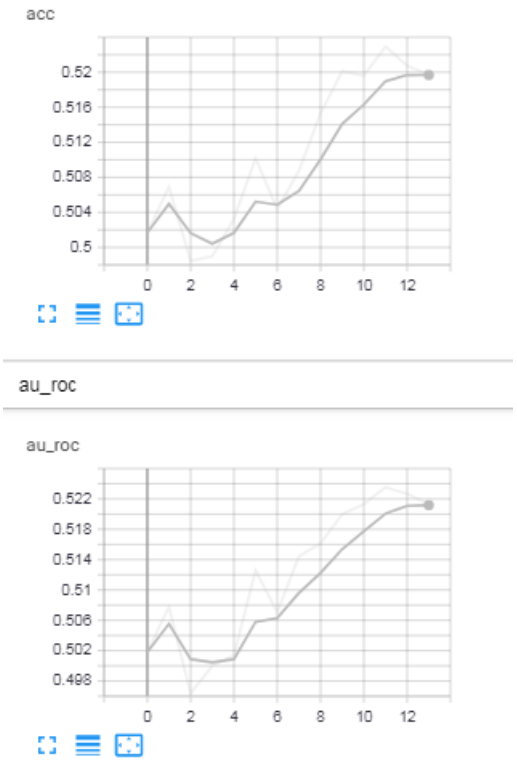
```
16000/16000 [==============================] - 0s 26us/step - loss: 0.6937 - acc: 0.4984 -
f1_metric: 0.4984 - au_roc: 0.4964 - val_loss: 0.6934 - val_acc: 0.5020 - val_f1_metric: 0.5020 -
val_au_roc: 0.5017
Changed learning rate: 0.0095
Epoch 4/30
16000/16000 [==============================] - 0s 29us/step - loss: 0.6938 - acc: 0.4990 -
f1_metric: 0.4990 - au_roc: 0.5000 - val_loss: 0.6932 - val_acc: 0.5005 - val_f1_metric: 0.5005 -
val_au_roc: 0.5009
Changed learning rate: 0.0095
Epoch 5/30
16000/16000 [==============================] - 0s 29us/step - loss: 0.6936 - acc: 0.5032 -
f1_metric: 0.5032 - au_roc: 0.5014 - val_loss: 0.6931 - val_acc: 0.5052 - val_f1_metric: 0.5052 -
val_au_roc: 0.5057
Changed learning rate: 0.008549999
Epoch 6/30
16000/16000 [==============================] - 0s 26us/step - loss: 0.6931 - acc: 0.5102 -
f1_metric: 0.5102 - au_roc: 0.5126 - val_loss: 0.6931 - val_acc: 0.5070 - val_f1_metric: 0.5070 -
val_au_roc: 0.5073
Changed learning rate: 0.008122499
Epoch 7/30
16000/16000 [==============================] - 0s 26us/step - loss: 0.6934 - acc: 0.5044 -
f1_metric: 0.5044 - au_roc: 0.5071 - val_loss: 0.6930 - val_acc: 0.5137 - val_f1_metric: 0.5137 -
val_au_roc: 0.5137
Changed learning rate: 0.0073102494
Epoch 8/30
16000/16000 [==============================] - 0s 27us/step - loss: 0.6930 - acc: 0.5087 -
f1_metric: 0.5087 - au_roc: 0.5144 - val_loss: 0.6929 - val_acc: 0.5115 - val_f1_metric: 0.5115 -
val_au_roc: 0.5111
Changed learning rate: 0.0073102494
Epoch 9/30
16000/16000 [==============================] - 0s 29us/step - loss: 0.6929 - acc: 0.5153 -
f1_metric: 0.5153 - au_roc: 0.5161 - val_loss: 0.6925 - val_acc: 0.5163 - val_f1_metric: 0.5162 -
val_au_roc: 0.5160
Changed learning rate: 0.006944737
Epoch 10/30
16000/16000 [==============================] - 0s 29us/step - loss: 0.6925 - acc: 0.5201 -
f1_metric: 0.5201 - au_roc: 0.5200 - val_loss: 0.6922 - val_acc: 0.5218 - val_f1_metric: 0.5217 -
val_au_roc: 0.5213
Changed learning rate: 0.006250263
Epoch 11/30
16000/16000 [==============================] - 0s 28us/step - loss: 0.6924 - acc: 0.5196 -
f1_metric: 0.5196 - au_roc: 0.5213 - val_loss: 0.6923 - val_acc: 0.5228 - val_f1_metric: 0.5227 -
val_au_roc: 0.5226
Changed learning rate: 0.005625237
Epoch 12/30
16000/16000 [==============================] - 0s 26us/step - loss: 0.6922 - acc: 0.5229 -
f1_metric: 0.5229 - au_roc: 0.5235 - val_loss: 0.6921 - val_acc: 0.5220 - val_f1_metric: 0.5220 -
val_au_roc: 0.5216
Changed learning rate: 0.005625237
Epoch 13/30
16000/16000 [==============================] - 0s 27us/step - loss: 0.6923 - acc: 0.5207 -
f1_metric: 0.5207 - au_roc: 0.5227 - val_loss: 0.6920 - val_acc: 0.5225 - val_f1_metric: 0.5225 -
val_au_roc: 0.5225
Changed learning rate: 0.0050627133
Epoch 14/30
16000/16000 [==============================] - 0s 27us/step - loss: 0.6923 - acc: 0.5197 -
f1_metric: 0.5197 - au_roc: 0.5213 - val_loss: 0.6921 - val_acc: 0.5225 - val_f1_metric: 0.5225 -
val_au_roc: 0.5222
Changed learning rate: 0.0050627133
'validation accuracy' is not decreased in last 2 epochs 13
```

In [23]:

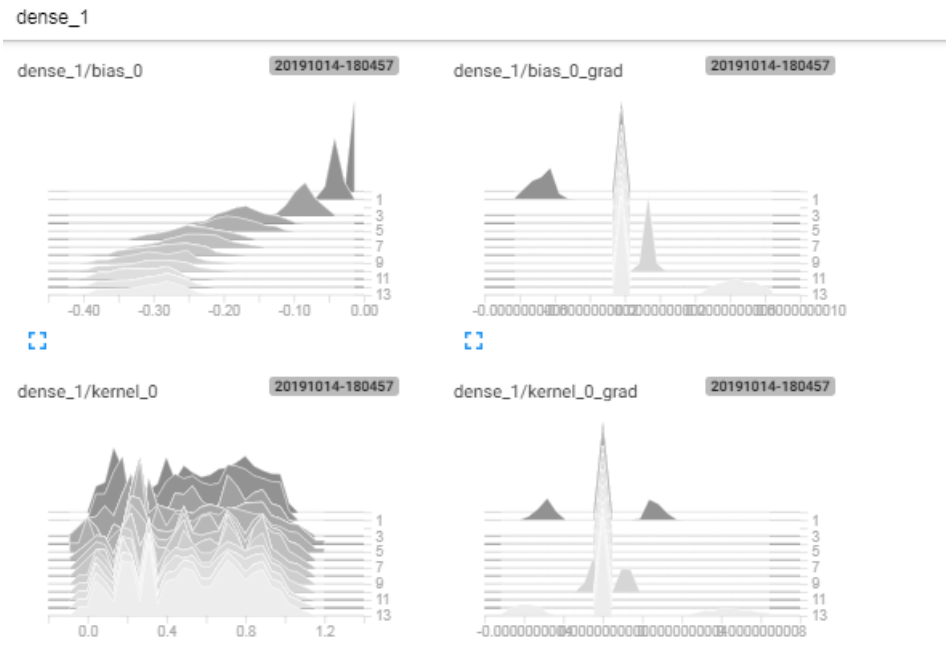
```
%tensorboard --logdir 'gdrive/My Drive/data/callbacks/'
```

```
Reusing TensorBoard on port 6006 (pid 537), started 0:00:33 ago. (Use '!kill 537' to kill it.)
```

Model 1: Scalar



Model 1: GRADIENT DISTRIBUTION



The gradient distribution helps to see the changes in the gradients over the epoch. Here the weights are changing as gradients also change over epoch. As the gradient changes, our model learns some pattern in data.

Model 2

In [0]:

```
# Defining  model 2
keras.backend.clear_session()
model2=Sequential()
```

```
model2.add(Dense(50,activation='relu',input_shape=(2,),kernel_initializer=initializers.RandomUniform(minval=0, maxval=1, seed=None)))

model2.add(Dense(40,activation='relu',kernel_initializer=initializers.RandomUniform(minval=0, maxval=1, seed=None)))

model2.add(Dense(30,activation='relu',kernel_initializer=initializers.RandomUniform(minval=0, maxval=1, seed=None)))

model2.add(Dense(20,activation='relu',kernel_initializer=initializers.RandomUniform(minval=0, maxval=1, seed=None)))

model2.add(Dense(10,activation='relu',kernel_initializer=initializers.RandomUniform(minval=0, maxval=1, seed=None)))
model2.add(Dense(2,activation='softmax'))
```

In [0]:

```
# Tensorboard
log_dir="gdrive/My Drive/data/callbacks/model 2/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True,write_grads=True)
```

In [30]:

```
# Compiling model 2
sgd_with_momemtum = optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model2.compile(optimizer=sgd_with_momemtum, loss='categorical_crossentropy', metrics=['accuracy',f1_metric,au_roc])
model2.fit(X_tr,Y_train,validation_data=(X_te,Y_test),epochs=50,batch_size=300,callbacks=[callbacks,
                                                                                          lrschedul
ensorboard_callback])
```

```
Train on 16000 samples, validate on 4000 samples
Epoch 1/50
16000/16000 [==============================] - 1s 45us/step - loss: 4.8803 - acc: 0.4956 -
f1_metric: 0.4944 - au_roc: 0.5018 - val_loss: 4.1003 - val_acc: 0.5000 - val_f1_metric: 0.5000 -
val_au_roc: 0.4921
Changed learning rate: 0.01
Epoch 2/50
16000/16000 [==============================] - 0s 31us/step - loss: 3.4521 - acc: 0.5000 -
f1_metric: 0.5000 - au_roc: 0.4720 - val_loss: 2.7066 - val_acc: 0.5000 - val_f1_metric: 0.5000 -
val_au_roc: 0.4550
Changed learning rate: 0.01
'validation accuracy' is not decreased in last 2 epochs 1
```

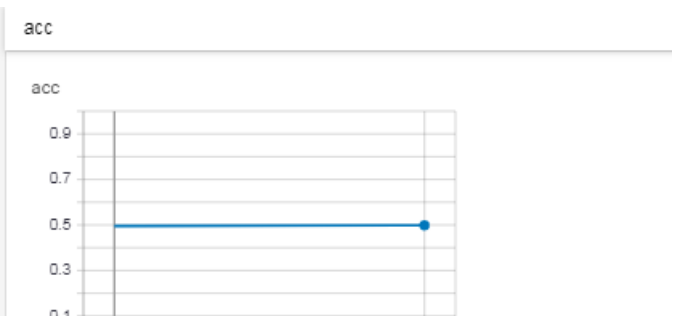
Out[30]:

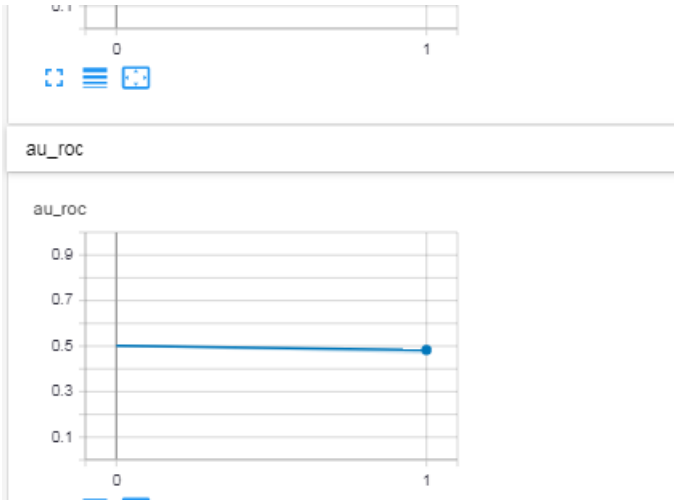
```
<keras.callbacks.History at 0x7f206d887160>
```

In [37]:

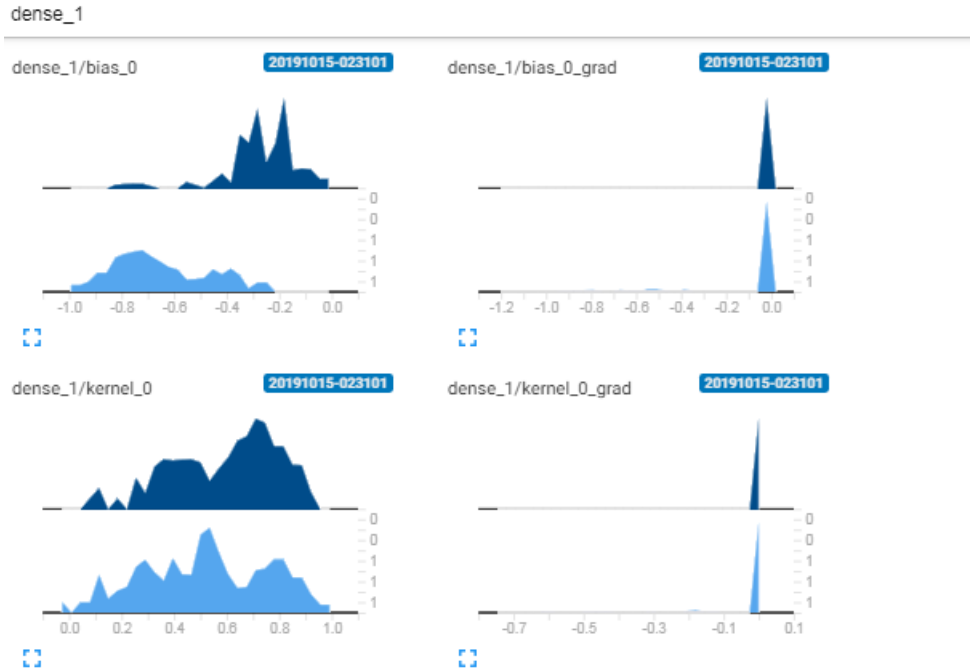
```
%tensorboard --logdir 'gdrive/My Drive/data/callbacks/model 2'
```

Model 2 : SCALAR





Model 2 : GRADIENT DISTRIBUTION



The gradient distribution helps to see the changes in the gradients over the epoch. Here the weights are not changing as gradients remain same over iterations. So we stopped training since validation accuracy did not improve for two epochs.

Model 3

In [0]:

```
# Defining  model 3

keras.backend.clear_session()
model3=Sequential()
model3.add(Dense(50,activation='relu',input_shape=(2,),kernel_initializer='he_normal'))

model3.add(Dense(40,activation='relu',kernel_initializer='he_normal'))

model3.add(Dense(30,activation='relu',kernel_initializer='he_normal'))

model3.add(Dense(20,activation='relu',kernel_initializer='he_normal'))

model3.add(Dense(10,activation='relu',kernel_initializer='he_normal'))
```

```
model3.add(Dense(2,activation='softmax'))
```

In [0]:

```
# Tensorboard
log_dir="gdrive/My Drive/data/callbacks/model 3/" + datetime.datetime.now().strftime("%Y%m%d-
%H%M%S")
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True,write_grads=
True)
```

In [36]:

```
# Compiling model 3
sgd_with_momemtum = optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model3.compile(optimizer=sgd_with_momemtum, loss='categorical_crossentropy', metrics=['accuracy',f
1_metric,au_roc])
model3.fit(X_tr,Y_train,validation_data=(X_te,Y_test),epochs=50,batch_size=300,callbacks=[lrschedul
e,tensorboard_callback])
```

```
Train on 16000 samples, validate on 4000 samples
Epoch 1/50
16000/16000 [==============================] - 1s 49us/step - loss: 0.6006 - acc: 0.6686 -
f1_metric: 0.6686 - au_roc: 0.7380 - val_loss: 0.5949 - val_acc: 0.6815 - val_f1_metric: 0.6815 -
val_au_roc: 0.7486
Changed learning rate: 0.01
Epoch 2/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.6008 - acc: 0.6696 -
f1_metric: 0.6696 - au_roc: 0.7374 - val_loss: 0.5975 - val_acc: 0.6795 - val_f1_metric: 0.6795 -
val_au_roc: 0.7499
Changed learning rate: 0.01
Epoch 3/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.6012 - acc: 0.6682 -
f1_metric: 0.6682 - au_roc: 0.7383 - val_loss: 0.5957 - val_acc: 0.6762 - val_f1_metric: 0.6762 -
val_au_roc: 0.7489
Changed learning rate: 0.01
Epoch 4/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.6012 - acc: 0.6701 -
f1_metric: 0.6701 - au_roc: 0.7381 - val_loss: 0.5944 - val_acc: 0.6810 - val_f1_metric: 0.6810 -
val_au_roc: 0.7487
Changed learning rate: 0.009
Epoch 5/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.6010 - acc: 0.6692 -
f1_metric: 0.6692 - au_roc: 0.7376 - val_loss: 0.5929 - val_acc: 0.6818 - val_f1_metric: 0.6817 -
val_au_roc: 0.7492
Changed learning rate: 0.008099999
Epoch 6/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.6001 - acc: 0.6721 -
f1_metric: 0.6721 - au_roc: 0.7388 - val_loss: 0.5936 - val_acc: 0.6790 - val_f1_metric: 0.6790 -
val_au_roc: 0.7487
Changed learning rate: 0.008099999
Epoch 7/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.6001 - acc: 0.6701 -
f1_metric: 0.6701 - au_roc: 0.7387 - val_loss: 0.5938 - val_acc: 0.6783 - val_f1_metric: 0.6782 -
val_au_roc: 0.7475
Changed learning rate: 0.008099999
Epoch 8/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.6001 - acc: 0.6699 -
f1_metric: 0.6699 - au_roc: 0.7388 - val_loss: 0.5945 - val_acc: 0.6785 - val_f1_metric: 0.6785 -
val_au_roc: 0.7493
Changed learning rate: 0.007289999
Epoch 9/50
16000/16000 [==============================] - 0s 29us/step - loss: 0.6009 - acc: 0.6681 -
f1_metric: 0.6681 - au_roc: 0.7383 - val_loss: 0.5937 - val_acc: 0.6818 - val_f1_metric: 0.6817 -
val_au_roc: 0.7485
Changed learning rate: 0.006925499
Epoch 10/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.6005 - acc: 0.6697 -
f1_metric: 0.6697 - au_roc: 0.7383 - val_loss: 0.5933 - val_acc: 0.6820 - val_f1_metric: 0.6820 -
val_au_roc: 0.7494
Changed learning rate: 0.006232949
Epoch 11/50
16000/16000 [==============================] - 0s 29us/step - loss: 0.6003 - acc: 0.6726 -
f1_metric: 0.6726 - au_roc: 0.7378 - val_loss: 0.5938 - val_acc: 0.6802 - val_f1_metric: 0.6802 -
val_au_roc: 0.7495
```

```
Changed learning rate: 0.006232949
Epoch 12/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.6005 - acc: 0.6691 -
f1_metric: 0.6691 - au_roc: 0.7388 - val_loss: 0.5938 - val_acc: 0.6852 - val_f1_metric: 0.6852 -
val_au_roc: 0.7497
Changed learning rate: 0.0059213014
Epoch 13/50
16000/16000 [==============================] - 0s 29us/step - loss: 0.6003 - acc: 0.6701 -
f1_metric: 0.6701 - au_roc: 0.7385 - val_loss: 0.5934 - val_acc: 0.6775 - val_f1_metric: 0.6775 -
val_au_roc: 0.7486
Changed learning rate: 0.0059213014
Epoch 14/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.6001 - acc: 0.6710 -
f1_metric: 0.6710 - au_roc: 0.7388 - val_loss: 0.5928 - val_acc: 0.6827 - val_f1_metric: 0.6827 -
val_au_roc: 0.7499
Changed learning rate: 0.005329171
Epoch 15/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.5998 - acc: 0.6706 -
f1_metric: 0.6706 - au_roc: 0.7389 - val_loss: 0.5928 - val_acc: 0.6815 - val_f1_metric: 0.6815 -
val_au_roc: 0.7494
Changed learning rate: 0.005329171
Epoch 16/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.5998 - acc: 0.6679 -
f1_metric: 0.6679 - au_roc: 0.7394 - val_loss: 0.5928 - val_acc: 0.6820 - val_f1_metric: 0.6820 -
val_au_roc: 0.7495
Changed learning rate: 0.004796254
Epoch 17/50
16000/16000 [==============================] - 0s 28us/step - loss: 0.6006 - acc: 0.6706 -
f1_metric: 0.6706 - au_roc: 0.7388 - val_loss: 0.5932 - val_acc: 0.6827 - val_f1_metric: 0.6827 -
val_au_roc: 0.7493
Changed learning rate: 0.0043166284
Epoch 18/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.5997 - acc: 0.6708 -
f1_metric: 0.6707 - au_roc: 0.7389 - val_loss: 0.5936 - val_acc: 0.6830 - val_f1_metric: 0.6830 -
val_au_roc: 0.7495
Changed learning rate: 0.004100797
Epoch 19/50
16000/16000 [==============================] - 0s 27us/step - loss: 0.5997 - acc: 0.6706 -
f1_metric: 0.6706 - au_roc: 0.7390 - val_loss: 0.5934 - val_acc: 0.6830 - val_f1_metric: 0.6830 -
val_au_roc: 0.7495
Changed learning rate: 0.004100797
'validation accuracy' is not decreased in last 2 epochs 18
```

Out[36]:

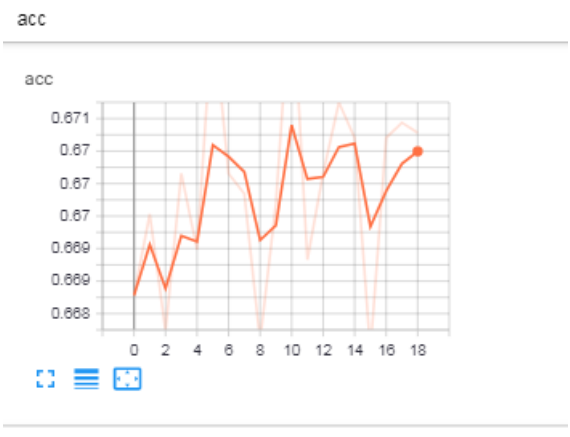
```
<keras.callbacks.History at 0x7f206b25bc18>
```

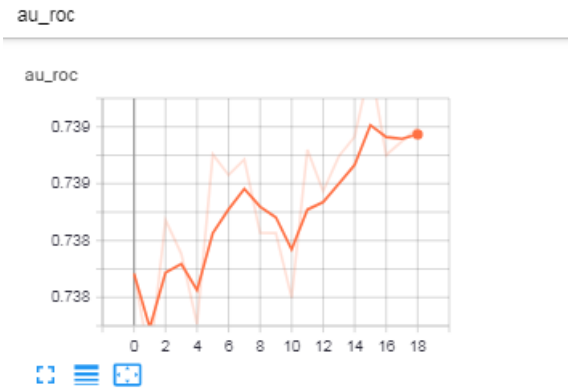
In [36]:

```
%tensorboard --logdir 'gdrive/My Drive/data/callbacks/model 3'
```

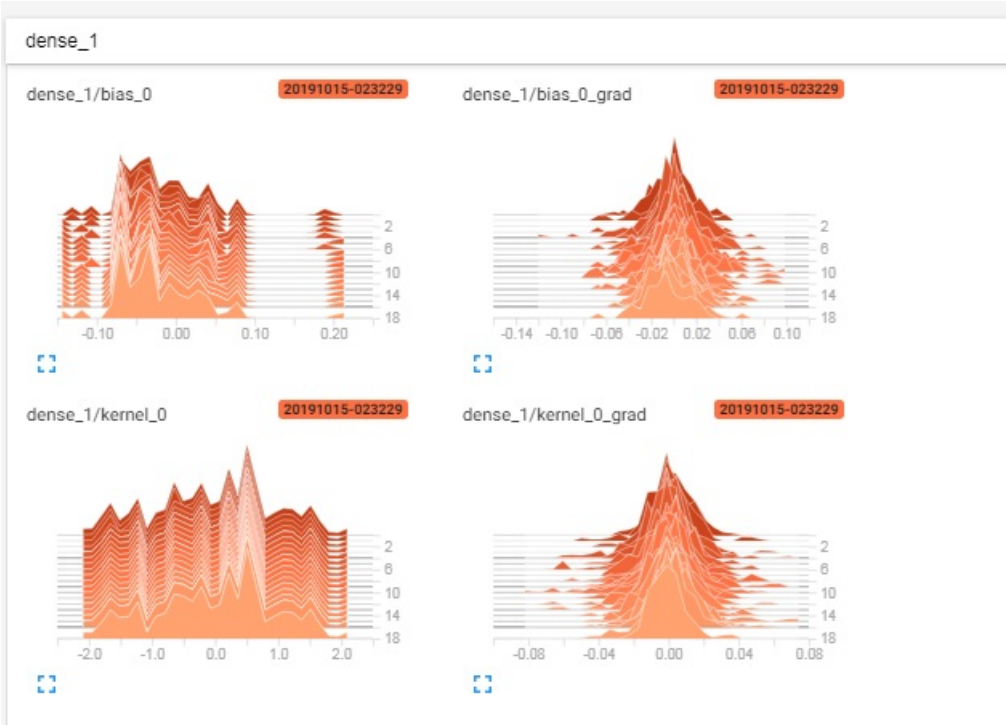
```
Reusing TensorBoard on port 6007 (pid 2719), started 0:02:33 ago. (Use '!kill 2719' to kill it.)
```

Model 3 : SCALAR





Mdoel 3: GRADIENT DISTRIBUTION



The distribution of grdients shows that , gradients and weights are changing over epoch.

Model 4

Sigmoid activation

In [0]:

```
# Defining  model 4('sigmoid activation')
keras.backend.clear_session()
model4=Sequential()
model4.add(Dense(50,activation='sigmoid',input_shape=(2,),kernel_initializer='he_normal'))

model4.add(Dense(40,activation='sigmoid',kernel_initializer='he_normal'))
model4.add(Dense(30,activation='sigmoid',kernel_initializer='he_normal'))

model4.add(Dense(20,activation='sigmoid',kernel_initializer='he_normal'))

model4.add(Dense(10,activation='sigmoid',kernel_initializer='he_normal'))
model4.add(Dense(2,activation='softmax'))
```

In [0]:

```
# Tensorboard
log_dir="gdrive/My Drive/data/callbacks/model 4/" + datetime.datetime.now().strftime("%Y%m%d-
%H%M%S")
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True,write_grads=
True)
```

In [43]:

```
# Compiling model 4('RMS prop optimizer')
rms_prop=keras.optimizers.RMSprop(lr=0.01, rho=0.9)
model4.compile(optimizer=rms_prop, loss='categorical_crossentropy', metrics=['accuracy',f1_metric,
au_roc])
#lr_schedule=LearningRateScheduler(logs_history)
history=model4.fit(X_tr,Y_train,validation_data=(X_te,Y_test),epochs=15,batch_size=300,callbacks=[l
rschedule,tensorboard_callback])
```

```
Train on 16000 samples, validate on 4000 samples
Epoch 1/15
16000/16000 [==============================] - 1s 48us/step - loss: 0.7030 - acc: 0.5031 -
f1_metric: 0.5031 - au_roc: 0.4837 - val_loss: 0.6940 - val_acc: 0.5000 - val_f1_metric: 0.5000 -
val_au_roc: 0.4994
Changed learning rate: 0.01
Epoch 2/15
16000/16000 [==============================] - 1s 35us/step - loss: 0.6946 - acc: 0.5027 -
f1_metric: 0.5027 - au_roc: 0.5602 - val_loss: 0.6932 - val_acc: 0.5000 - val_f1_metric: 0.5000 -
val_au_roc: 0.6066
Changed learning rate: 0.01
Epoch 3/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6945 - acc: 0.5039 -
f1_metric: 0.5039 - au_roc: 0.6003 - val_loss: 0.6951 - val_acc: 0.5000 - val_f1_metric: 0.5000 -
val_au_roc: 0.5856
Changed learning rate: 0.0095
Epoch 4/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6944 - acc: 0.4987 -
f1_metric: 0.4987 - au_roc: 0.6021 - val_loss: 0.6890 - val_acc: 0.5625 - val_f1_metric: 0.5625 -
val_au_roc: 0.6218
Changed learning rate: 0.008549999
Epoch 5/15
16000/16000 [==============================] - 0s 28us/step - loss: 0.6778 - acc: 0.5764 -
f1_metric: 0.5764 - au_roc: 0.6350 - val_loss: 0.6644 - val_acc: 0.5915 - val_f1_metric: 0.5915 -
val_au_roc: 0.6922
Changed learning rate: 0.0076949997
Epoch 6/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6477 - acc: 0.6222 -
f1_metric: 0.6222 - au_roc: 0.7084 - val_loss: 0.6550 - val_acc: 0.6090 - val_f1_metric: 0.6090 -
val_au_roc: 0.7344
Changed learning rate: 0.00731025
Epoch 7/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6262 - acc: 0.6513 -
f1_metric: 0.6513 - au_roc: 0.7295 - val_loss: 0.6046 - val_acc: 0.6773 - val_f1_metric: 0.6772 -
val_au_roc: 0.7416
Changed learning rate: 0.006579225
Epoch 8/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6191 - acc: 0.6553 -
f1_metric: 0.6553 - au_roc: 0.7320 - val_loss: 0.6305 - val_acc: 0.6543 - val_f1_metric: 0.6542 -
val_au_roc: 0.7433
Changed learning rate: 0.006579225
Epoch 9/15
16000/16000 [==============================] - 1s 37us/step - loss: 0.6195 - acc: 0.6589 -
f1_metric: 0.6589 - au_roc: 0.7324 - val_loss: 0.6000 - val_acc: 0.6785 - val_f1_metric: 0.6785 -
val_au_roc: 0.7452
Changed learning rate: 0.0062502637
Epoch 10/15
16000/16000 [==============================] - 0s 30us/step - loss: 0.6164 - acc: 0.6578 -
f1_metric: 0.6578 - au_roc: 0.7335 - val_loss: 0.6052 - val_acc: 0.6778 - val_f1_metric: 0.6777 -
val_au_roc: 0.7435
Changed learning rate: 0.0062502637
Epoch 11/15
16000/16000 [==============================] - 0s 28us/step - loss: 0.6175 - acc: 0.6574 -
f1_metric: 0.6574 - au_roc: 0.7334 - val_loss: 0.5982 - val_acc: 0.6822 - val_f1_metric: 0.6822 -
val_au_roc: 0.7464
Changed learning rate: 0.0056252372
Epoch 12/15
```

```
16000/16000 [==============================] - 0s 30us/step - loss: 0.6124 - acc: 0.6637 -
f1_metric: 0.6637 - au_roc: 0.7347 - val_loss: 0.6103 - val_acc: 0.6665 - val_f1_metric: 0.6665 -
val_au_roc: 0.7459
Changed learning rate: 0.0056252372
Epoch 13/15
16000/16000 [==============================] - 0s 30us/step - loss: 0.6137 - acc: 0.6646 -
f1_metric: 0.6646 - au_roc: 0.7343 - val_loss: 0.6179 - val_acc: 0.6600 - val_f1_metric: 0.6600 -
val_au_roc: 0.7428
Changed learning rate: 0.0056252372
Epoch 14/15
16000/16000 [==============================] - 0s 30us/step - loss: 0.6118 - acc: 0.6656 -
f1_metric: 0.6656 - au_roc: 0.7345 - val_loss: 0.6170 - val_acc: 0.6608 - val_f1_metric: 0.6607 -
val_au_roc: 0.7393
Changed learning rate: 0.0050627133
Epoch 15/15
16000/16000 [==============================] - 0s 28us/step - loss: 0.6135 - acc: 0.6643 -
f1_metric: 0.6642 - au_roc: 0.7338 - val_loss: 0.5978 - val_acc: 0.6825 - val_f1_metric: 0.6825 -
val_au_roc: 0.7485
Changed learning rate: 0.0048095775
```

Leaky Relu activation

In [0]:

```
# leaky relu
# Defining  model 4('leaky relu activation')
keras.backend.clear_session()
leaky_relu=keras.layers.LeakyReLU(alpha=0.3)
model4=Sequential()
model4.add(Dense(50,activation=leaky_relu,input_shape=(2,),kernel_initializer='glorot_uniform'))

model4.add(Dense(40,activation=leaky_relu,kernel_initializer='glorot_uniform'))
model4.add(Dense(30,activation=leaky_relu,kernel_initializer='glorot_uniform'))

model4.add(Dense(20,activation=leaky_relu,kernel_initializer='glorot_uniform'))

model4.add(Dense(10,activation=leaky_relu,kernel_initializer='glorot_uniform'))
model4.add(Dense(2,activation='softmax'))
```

In [0]:

```
# Tensorboard
log_dir="gdrive/My Drive/data/callbacks/model 4/" + datetime.datetime.now().strftime("%Y%m%d-
%H%M%S")
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1, write_graph=True,write_grads=
True)
```

In [29]:

```
# Compiling model 4('RMS prop optimizer')
rms_prop=keras.optimizers.RMSprop(lr=0.01, rho=0.9)
model4.compile(optimizer=rms_prop, loss='categorical_crossentropy', metrics=['accuracy',f1_metric,
au_roc])
#lr_schedule=LearningRateScheduler(logs_history)
history=model4.fit(X_tr,Y_train,validation_data=(X_te,Y_test),epochs=15,batch_size=300,callbacks=[l
rschedule,tensorboard_callback])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name t
f.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is deprecated. Please use tf.ma
th.log instead.

WARNING:tensorflow:From <ipython-input-15-116bbdf7c629>:3: py_func (from
tensorflow.python.ops.script_ops) is deprecated and will be removed in a future version.
Instructions for updating:
tf.py_func is deprecated in TF V2. Instead, there are two
    options available in V2.
    - tf.py_function takes a python function which manipulates tf eager
    tensors instead of numpy arrays. It's easy to convert a tf eager tensor to
    an ndarray (just call tensor.numpy()) but having access to eager tensors
```

```
    an ndarray (just call tensor.numpy()) but having access to eager tensors
    means `tf.py_function`s can use accelerators such as GPUs as well as
    being differentiable using a gradient tape.
    - tf.numpy_function maintains the semantics of the deprecated tf.py_func
    (it is not differentiable, and manipulates numpy arrays). It drops the
    stateful argument making all functions stateful.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow_core/python/ops/math_grad.py:1424: where (from
tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please us
e tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please use tf
.compat.v1.assign instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3005: The name tf.Session is deprecated. Please use t
f.compat.v1.Session instead.

Train on 16000 samples, validate on 4000 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:190: The name tf.get_default_session is deprecated. P
lease use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:197: The name tf.ConfigProto is deprecated. Please us
e tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:207: The name tf.global_variables is deprecated. Plea
se use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:216: The name tf.is_variable_initialized is
deprecated. Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:223: The name tf.variables_initializer is deprecated.
Please use tf.compat.v1.variables_initializer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1068: The name t
f.summary.histogram is deprecated. Please use tf.compat.v1.summary.histogram instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name t
f.summary.merge_all is deprecated. Please use tf.compat.v1.summary.merge_all instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name t
f.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/15
16000/16000 [==============================] - 1s 94us/step - loss: 0.6562 - acc: 0.6065 -
f1_metric: 0.6065 - au_roc: 0.6723 - val_loss: 0.6341 - val_acc: 0.6300 - val_f1_metric: 0.6300 -
val_au_roc: 0.7019
Changed learning rate: 0.01
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name t
f.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6280 - acc: 0.6521 -
f1_metric: 0.6521 - au_roc: 0.7207 - val_loss: 0.6036 - val_acc: 0.6700 - val_f1_metric: 0.6700 -
val_au_roc: 0.7417
Changed learning rate: 0.009
Epoch 3/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6208 - acc: 0.6571 -
f1_metric: 0.6571 - au_roc: 0.7235 - val_loss: 0.6541 - val_acc: 0.6285 - val_f1_metric: 0.6285 -
val_au_roc: 0.7278
Changed learning rate: 0.009
Epoch 4/15
16000/16000 [==============================] - 0s 31us/step - loss: 0.6187 - acc: 0.6570 -
f1_metric: 0.6570 - au_roc: 0.7270 - val_loss: 0.6082 - val_acc: 0.6702 - val_f1_metric: 0.6702 -
val_au_roc: 0.7368
Changed learning rate: 0.008099999
Epoch 5/15
```

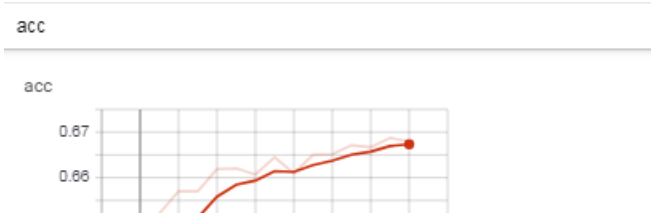
```
Epoch 5/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6142 - acc: 0.6619 -
f1_metric: 0.6619 - au_roc: 0.7289 - val_loss: 0.6371 - val_acc: 0.6365 - val_f1_metric: 0.6365 -
val_au_roc: 0.7392
Changed learning rate: 0.008099999
Epoch 6/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6119 - acc: 0.6621 -
f1_metric: 0.6621 - au_roc: 0.7296 - val_loss: 0.6180 - val_acc: 0.6598 - val_f1_metric: 0.6597 -
val_au_roc: 0.7393
Changed learning rate: 0.007694999
Epoch 7/15
16000/16000 [==============================] - 0s 30us/step - loss: 0.6125 - acc: 0.6607 -
f1_metric: 0.6607 - au_roc: 0.7313 - val_loss: 0.6112 - val_acc: 0.6612 - val_f1_metric: 0.6612 -
val_au_roc: 0.7303
Changed learning rate: 0.006925499
Epoch 8/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6081 - acc: 0.6645 -
f1_metric: 0.6645 - au_roc: 0.7333 - val_loss: 0.6380 - val_acc: 0.6540 - val_f1_metric: 0.6540 -
val_au_roc: 0.7225
Changed learning rate: 0.006925499
Epoch 9/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6116 - acc: 0.6609 -
f1_metric: 0.6609 - au_roc: 0.7298 - val_loss: 0.6180 - val_acc: 0.6543 - val_f1_metric: 0.6542 -
val_au_roc: 0.7243
Changed learning rate: 0.006579224
Epoch 10/15
16000/16000 [==============================] - 0s 31us/step - loss: 0.6082 - acc: 0.6650 -
f1_metric: 0.6650 - au_roc: 0.7314 - val_loss: 0.6099 - val_acc: 0.6708 - val_f1_metric: 0.6707 -
val_au_roc: 0.7466
Changed learning rate: 0.0059213014
Epoch 11/15
16000/16000 [==============================] - 0s 30us/step - loss: 0.6079 - acc: 0.6651 -
f1_metric: 0.6651 - au_roc: 0.7320 - val_loss: 0.6009 - val_acc: 0.6788 - val_f1_metric: 0.6787 -
val_au_roc: 0.7428
Changed learning rate: 0.005329171
Epoch 12/15
16000/16000 [==============================] - 0s 28us/step - loss: 0.6061 - acc: 0.6671 -
f1_metric: 0.6671 - au_roc: 0.7337 - val_loss: 0.6030 - val_acc: 0.6620 - val_f1_metric: 0.6620 -
val_au_roc: 0.7417
Changed learning rate: 0.005329171
Epoch 13/15
16000/16000 [==============================] - 0s 30us/step - loss: 0.6053 - acc: 0.6667 -
f1_metric: 0.6667 - au_roc: 0.7338 - val_loss: 0.5980 - val_acc: 0.6760 - val_f1_metric: 0.6760 -
val_au_roc: 0.7461
Changed learning rate: 0.004796254
Epoch 14/15
16000/16000 [==============================] - 0s 28us/step - loss: 0.6046 - acc: 0.6688 -
f1_metric: 0.6687 - au_roc: 0.7348 - val_loss: 0.5962 - val_acc: 0.6810 - val_f1_metric: 0.6810 -
val_au_roc: 0.7449
Changed learning rate: 0.0043166284
Epoch 15/15
16000/16000 [==============================] - 0s 29us/step - loss: 0.6042 - acc: 0.6679 -
f1_metric: 0.6679 - au_roc: 0.7350 - val_loss: 0.6025 - val_acc: 0.6713 - val_f1_metric: 0.6712 -
val_au_roc: 0.7413
Changed learning rate: 0.0043166284
```

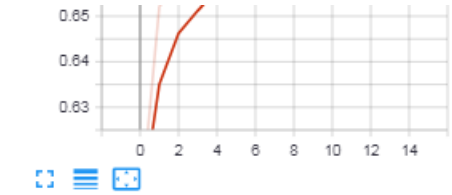
In [33]:

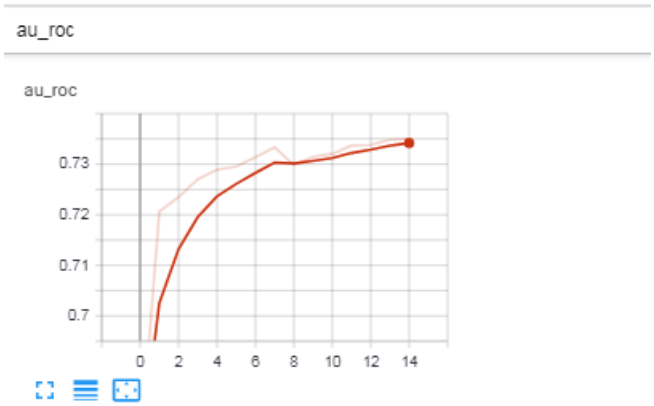
```
%tensorboard --logdir 'gdrive/My Drive/data/callbacks/model 4'
```

```
Reusing TensorBoard on port 6006 (pid 1311), started 0:01:22 ago. (Use '!kill 1311' to kill it.)
```

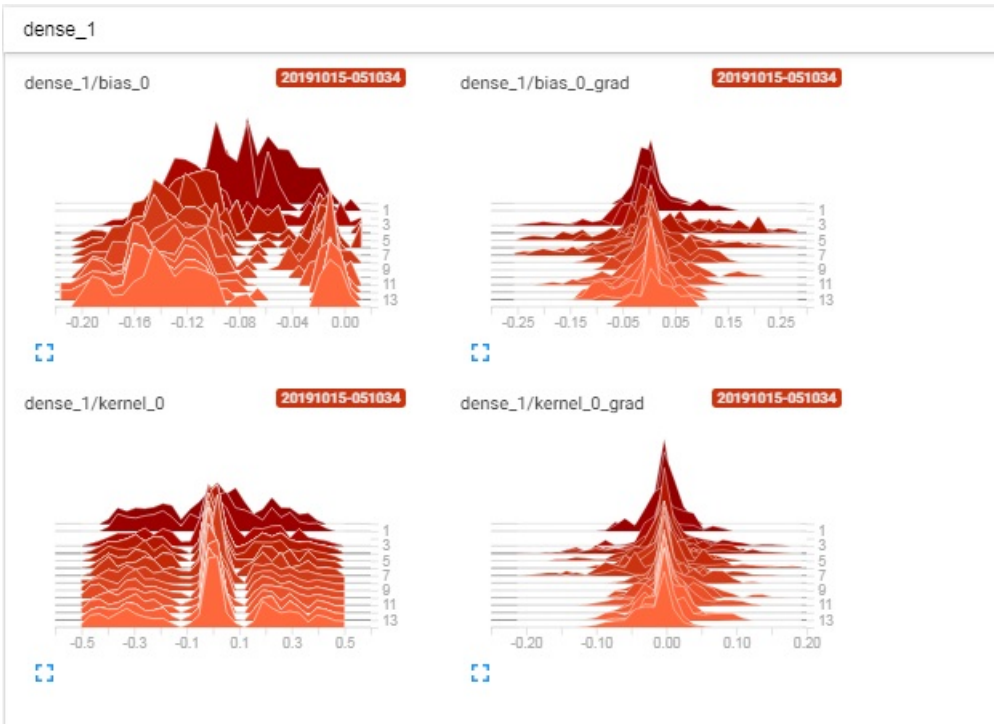
Model 4: SCALAR







Model 4: GRADIENT DISTRIBUTION



On seeing gradient distribution, we find that gradients are chaning over epoch and help in improving the validation auc. The model learns pattern in the data and predicts test data with high validation auc.

In [0]:

In [0]:

In [0]: