Copyright by Pierian Data Inc.
Created by Jose Marcial Portilla.

Tensorboard

Let's explore the built in data visualization capabilities that come with Tensorboard.

Full official tutorial available here: https://www.tensorflow.org/tensorboard/get_started

Data

In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
df = pd.read_csv('03 Kera Classification.csv')
```

Train Test Split

In [3]:
```python
X = df.drop('benign_0__mal_1',axis=1).values
y = df['benign_0__mal_1'].values
```

In [4]:
```python
from sklearn.model_selection import train_test_split
```

In [5]:
```python
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.25,random_state=101)
```

Scaling Data

In [6]:
```python
from sklearn.preprocessing import MinMaxScaler
```

In [7]:
```python
scaler = MinMaxScaler()
```

In [8]:
```python
scaler.fit(X_train)
```

Out[8]:
```
MinMaxScaler()
```

In [9]:
```python
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Creating the Model

In [10]:
```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation,Dropout
```

In [11]:
```python
from tensorflow.keras.callbacks import EarlyStopping,TensorBoard
```

In [12]:
```python
early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=25)
```

In [13]:
```python
pwd
```

Out[13]:
```
'C:\\Users\\Yasin\\Desktop\\Machine Excercise\\New folder\\Deep Learning and Neural Network'
```

Creating the Tensorboard Callback

TensorBoard is a visualization tool provided with TensorFlow.

This callback logs events for TensorBoard, including:

- Metrics summary plots
- Training graph visualization
- Activation histograms
- Sampled profiling

If you have installed TensorFlow with pip, you should be able to launch TensorBoard from the command line:

```
tensorboard --logdir=path_to_your_logs
```

You can find more information about TensorBoard here.

```
Arguments:
    log_dir: the path of the directory where to save the log files to be
      parsed by TensorBoard.
    histogram_freq: frequency (in epochs) at which to compute activation and
      weight histograms for the layers of the model. If set to 0, histograms
      won't be computed. Validation data (or split) must be specified for
      histogram visualizations.
    write_graph: whether to visualize the graph in TensorBoard. The log file
      can become quite large when write_graph is set to True.
    write_images: whether to write model weights to visualize as image in
      TensorBoard.
    update_freq: `'batch'` or `'epoch'` or integer. When using `'batch'`,
      writes the losses and metrics to TensorBoard after each batch. The same
      applies for `'epoch'`. If using an integer, let's say `1000`, the
      callback will write the metrics and losses to TensorBoard every 1000
      samples. Note that writing too frequently to TensorBoard can slow down
      your training.
    profile_batch: Profile the batch to sample compute characteristics. By
      default, it will profile the second batch. Set profile_batch=0 to
      disable profiling. Must run in TensorFlow eager mode.
    embeddings_freq: frequency (in epochs) at which embedding layers will
      be visualized. If set to 0, embeddings won't be visualized.
```

In [14]:
```python
from datetime import datetime
```

In [15]:
```python
datetime.now().strftime("%Y-%m-%d--%H%M")
```

Out[15]:
```
'2021-09-20--1846'
```

In [16]:
```python
# WINDOWS: Use "logs\\fit"
# MACOS/LINUX: Use "logs\fit"

log_directory = 'logs\\fit'

# OPTIONAL: ADD A TIMESTAMP FOR UNIQUE FOLDER
# timestamp = datetime.now().strftime("%Y-%m-%d--%H%M")
# log_directory = log_directory + '\\' + timestamp


board = TensorBoard(log_dir=log_directory,histogram_freq=1,
    write_graph=True,
    write_images=True,
    update_freq='epoch',
    profile_batch=2,
    embeddings_freq=1)
```

Now create the model layers:

In [17]:
```python
model = Sequential()
model.add(Dense(units=30,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(units=15,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(units=1,activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam')
```

Train the Model

In [18]:
```python
model.fit(x=X_train,
          y=y_train,
          epochs=600,
          validation_data=(X_test, y_test), verbose=1,
          callbacks=[early_stop,board]
          )
```

```
Epoch 1/600
 1/14 [=>............................] - ETA: 0s - loss: 0.7480WARNING:tensorflow:From C:\Users\Yasin\anaconda3\lib\site-packages\tensorflow\python\ops\summar
y_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.
 2/14 [===>..........................] - ETA: 3s - loss: 0.8036WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared to the batch time (ba
tch time: 0.0350s vs `on_train_batch_end` time: 0.4991s). Check your callbacks.
14/14 [==============================] - 1s 98ms/step - loss: 0.7412 - val_loss: 0.6886
Epoch 2/600
14/14 [==============================] - 0s 26ms/step - loss: 0.6822 - val_loss: 0.6621
Epoch 3/600
14/14 [==============================] - 0s 26ms/step - loss: 0.6791 - val_loss: 0.6441
Epoch 4/600
14/14 [==============================] - 0s 23ms/step - loss: 0.6405 - val_loss: 0.6227
Epoch 5/600
14/14 [==============================] - 0s 24ms/step - loss: 0.6325 - val_loss: 0.6006
Epoch 6/600
14/14 [==============================] - 0s 20ms/step - loss: 0.6045 - val_loss: 0.5683
Epoch 7/600
14/14 [==============================] - 0s 21ms/step - loss: 0.5800 - val_loss: 0.5366
Epoch 8/600
14/14 [==============================] - 0s 22ms/step - loss: 0.5666 - val_loss: 0.5114
Epoch 9/600
14/14 [==============================] - 0s 23ms/step - loss: 0.5538 - val_loss: 0.4860
Epoch 10/600
14/14 [==============================] - 0s 24ms/step - loss: 0.5227 - val_loss: 0.4564
Epoch 11/600
14/14 [==============================] - 0s 19ms/step - loss: 0.5255 - val_loss: 0.4340
Epoch 12/600
14/14 [==============================] - 0s 20ms/step - loss: 0.4899 - val_loss: 0.4141
Epoch 13/600
14/14 [==============================] - 0s 26ms/step - loss: 0.4763 - val_loss: 0.3910
Epoch 14/600
14/14 [==============================] - 0s 21ms/step - loss: 0.4380 - val_loss: 0.3699
Epoch 15/600
14/14 [==============================] - 0s 23ms/step - loss: 0.4219 - val_loss: 0.3505
Epoch 16/600
14/14 [==============================] - 0s 20ms/step - loss: 0.4121 - val_loss: 0.3308
Epoch 17/600
14/14 [==============================] - 0s 23ms/step - loss: 0.3850 - val_loss: 0.3162
Epoch 18/600
14/14 [==============================] - 0s 27ms/step - loss: 0.3619 - val_loss: 0.2977
Epoch 19/600
14/14 [==============================] - 0s 26ms/step - loss: 0.3614 - val_loss: 0.2812
Epoch 20/600
14/14 [==============================] - 0s 22ms/step - loss: 0.3750 - val_loss: 0.2689
Epoch 21/600
14/14 [==============================] - 0s 23ms/step - loss: 0.3531 - val_loss: 0.2569
Epoch 22/600
14/14 [==============================] - 0s 22ms/step - loss: 0.3345 - val_loss: 0.2512
Epoch 23/600
14/14 [==============================] - 0s 23ms/step - loss: 0.3085 - val_loss: 0.2369
Epoch 24/600
14/14 [==============================] - 0s 24ms/step - loss: 0.3182 - val_loss: 0.2291
Epoch 25/600
14/14 [==============================] - 0s 28ms/step - loss: 0.2944 - val_loss: 0.2244
Epoch 26/600
14/14 [==============================] - 0s 29ms/step - loss: 0.3110 - val_loss: 0.2126
Epoch 27/600
14/14 [==============================] - 0s 29ms/step - loss: 0.3029 - val_loss: 0.2031
Epoch 28/600
14/14 [==============================] - 0s 21ms/step - loss: 0.2818 - val_loss: 0.1994
Epoch 29/600
14/14 [==============================] - 0s 21ms/step - loss: 0.2746 - val_loss: 0.1948
Epoch 30/600
14/14 [==============================] - 0s 24ms/step - loss: 0.2764 - val_loss: 0.1838
Epoch 31/600
14/14 [==============================] - 0s 22ms/step - loss: 0.2409 - val_loss: 0.1787
Epoch 32/600
14/14 [==============================] - 0s 19ms/step - loss: 0.2367 - val_loss: 0.1730
Epoch 33/600
14/14 [==============================] - 0s 19ms/step - loss: 0.2654 - val_loss: 0.1724
Epoch 34/600
14/14 [==============================] - 0s 20ms/step - loss: 0.2487 - val_loss: 0.1686
Epoch 35/600
14/14 [==============================] - 0s 20ms/step - loss: 0.2662 - val_loss: 0.1646
Epoch 36/600
14/14 [==============================] - 0s 26ms/step - loss: 0.2284 - val_loss: 0.1643
Epoch 37/600
14/14 [==============================] - 0s 23ms/step - loss: 0.2476 - val_loss: 0.1604
Epoch 38/600
14/14 [==============================] - 0s 21ms/step - loss: 0.2079 - val_loss: 0.1538
Epoch 39/600
14/14 [==============================] - 0s 24ms/step - loss: 0.2514 - val_loss: 0.1561
Epoch 40/600
14/14 [==============================] - 0s 27ms/step - loss: 0.2026 - val_loss: 0.1522
Epoch 41/600
14/14 [==============================] - 0s 20ms/step - loss: 0.2201 - val_loss: 0.1518
Epoch 42/600
14/14 [==============================] - 0s 26ms/step - loss: 0.2115 - val_loss: 0.1449
Epoch 43/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1741 - val_loss: 0.1424
Epoch 44/600
14/14 [==============================] - 0s 25ms/step - loss: 0.2182 - val_loss: 0.1396
Epoch 45/600
14/14 [==============================] - 0s 22ms/step - loss: 0.1773 - val_loss: 0.1375
Epoch 46/600
14/14 [==============================] - 0s 28ms/step - loss: 0.1979 - val_loss: 0.1490
Epoch 47/600
14/14 [==============================] - 0s 25ms/step - loss: 0.2142 - val_loss: 0.1379
Epoch 48/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1838 - val_loss: 0.1344
Epoch 49/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1863 - val_loss: 0.1329
Epoch 50/600
14/14 [==============================] - 0s 22ms/step - loss: 0.1827 - val_loss: 0.1345
Epoch 51/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1739 - val_loss: 0.1296
Epoch 52/600
14/14 [==============================] - 0s 26ms/step - loss: 0.1666 - val_loss: 0.1278
Epoch 53/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1704 - val_loss: 0.1325
Epoch 54/600
14/14 [==============================] - 0s 30ms/step - loss: 0.1965 - val_loss: 0.1247
Epoch 55/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1539 - val_loss: 0.1245
Epoch 56/600
14/14 [==============================] - 0s 27ms/step - loss: 0.1551 - val_loss: 0.1231
Epoch 57/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1394 - val_loss: 0.1215
Epoch 58/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1511 - val_loss: 0.1267
Epoch 59/600
14/14 [==============================] - 0s 26ms/step - loss: 0.1669 - val_loss: 0.1283
Epoch 60/600
14/14 [==============================] - 0s 30ms/step - loss: 0.1501 - val_loss: 0.1185
Epoch 61/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1751 - val_loss: 0.1197
Epoch 62/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1287 - val_loss: 0.1207
Epoch 63/600
14/14 [==============================] - 0s 19ms/step - loss: 0.1548 - val_loss: 0.1153
Epoch 64/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1449 - val_loss: 0.1177
Epoch 65/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1782 - val_loss: 0.1147
Epoch 66/600
14/14 [==============================] - 0s 22ms/step - loss: 0.1456 - val_loss: 0.1165
Epoch 67/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1142 - val_loss: 0.1143
Epoch 68/600
14/14 [==============================] - 0s 28ms/step - loss: 0.1300 - val_loss: 0.1206
Epoch 69/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1413 - val_loss: 0.1169
Epoch 70/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1335 - val_loss: 0.1154
Epoch 71/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1230 - val_loss: 0.1089
Epoch 72/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1362 - val_loss: 0.1177
Epoch 73/600
14/14 [==============================] - 0s 26ms/step - loss: 0.1401 - val_loss: 0.1155
Epoch 74/600
```

```
14/14 [==============================] - 0s 21ms/step - loss: 0.1228 - val_loss: 0.1142
Epoch 75/600
14/14 [==============================] - 0s 31ms/step - loss: 0.1465 - val_loss: 0.1107
Epoch 76/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1186 - val_loss: 0.1128
Epoch 77/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1254 - val_loss: 0.1246
Epoch 78/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1245 - val_loss: 0.1080
Epoch 79/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1323 - val_loss: 0.1119
Epoch 80/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1111 - val_loss: 0.1135
Epoch 81/600
14/14 [==============================] - 0s 26ms/step - loss: 0.1370 - val_loss: 0.1108
Epoch 82/600
14/14 [==============================] - 0s 22ms/step - loss: 0.1240 - val_loss: 0.1187
Epoch 83/600
14/14 [==============================] - 0s 19ms/step - loss: 0.1297 - val_loss: 0.1133
Epoch 84/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1307 - val_loss: 0.1127
Epoch 85/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1300 - val_loss: 0.1110
Epoch 86/600
14/14 [==============================] - 0s 27ms/step - loss: 0.1292 - val_loss: 0.1192
Epoch 87/600
14/14 [==============================] - 0s 28ms/step - loss: 0.1060 - val_loss: 0.1113
Epoch 88/600
14/14 [==============================] - 0s 22ms/step - loss: 0.1045 - val_loss: 0.1146
Epoch 89/600
14/14 [==============================] - 0s 30ms/step - loss: 0.1211 - val_loss: 0.1196
Epoch 90/600
14/14 [==============================] - 0s 32ms/step - loss: 0.0903 - val_loss: 0.1116
Epoch 91/600
14/14 [==============================] - 0s 20ms/step - loss: 0.1044 - val_loss: 0.1114
Epoch 92/600
14/14 [==============================] - 0s 24ms/step - loss: 0.0937 - val_loss: 0.1123
Epoch 93/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1057 - val_loss: 0.1272
Epoch 94/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1276 - val_loss: 0.1156
Epoch 95/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1245 - val_loss: 0.1107
Epoch 96/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1288 - val_loss: 0.1128
Epoch 97/600
14/14 [==============================] - 0s 30ms/step - loss: 0.1183 - val_loss: 0.1145
Epoch 98/600
14/14 [==============================] - 0s 26ms/step - loss: 0.0883 - val_loss: 0.1156
Epoch 99/600
14/14 [==============================] - 0s 23ms/step - loss: 0.0929 - val_loss: 0.1136
Epoch 100/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1205 - val_loss: 0.1076
Epoch 101/600
14/14 [==============================] - 0s 25ms/step - loss: 0.0977 - val_loss: 0.1114
Epoch 102/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1093 - val_loss: 0.1177
Epoch 103/600
14/14 [==============================] - 0s 21ms/step - loss: 0.0946 - val_loss: 0.1204
Epoch 104/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1078 - val_loss: 0.1187
Epoch 105/600
14/14 [==============================] - 0s 28ms/step - loss: 0.0972 - val_loss: 0.1121
Epoch 106/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1092 - val_loss: 0.1162
Epoch 107/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1031 - val_loss: 0.1107
Epoch 108/600
14/14 [==============================] - 0s 23ms/step - loss: 0.1066 - val_loss: 0.1078
Epoch 109/600
14/14 [==============================] - 0s 25ms/step - loss: 0.0955 - val_loss: 0.1239
Epoch 110/600
14/14 [==============================] - 0s 22ms/step - loss: 0.1062 - val_loss: 0.1155
Epoch 111/600
14/14 [==============================] - 0s 29ms/step - loss: 0.1209 - val_loss: 0.1207
Epoch 112/600
14/14 [==============================] - 0s 24ms/step - loss: 0.1055 - val_loss: 0.1173
Epoch 113/600
14/14 [==============================] - 0s 28ms/step - loss: 0.1023 - val_loss: 0.1098
Epoch 114/600
14/14 [==============================] - 0s 23ms/step - loss: 0.0798 - val_loss: 0.1092
Epoch 115/600
14/14 [==============================] - 0s 25ms/step - loss: 0.1065 - val_loss: 0.1156
Epoch 116/600
14/14 [==============================] - 0s 22ms/step - loss: 0.0773 - val_loss: 0.1171
Epoch 117/600
14/14 [==============================] - 0s 30ms/step - loss: 0.1068 - val_loss: 0.1142
Epoch 118/600
14/14 [==============================] - 0s 21ms/step - loss: 0.1100 - val_loss: 0.1087
Epoch 119/600
14/14 [==============================] - 0s 21ms/step - loss: 0.0906 - val_loss: 0.1205
Epoch 120/600
14/14 [==============================] - 0s 21ms/step - loss: 0.0910 - val_loss: 0.1130
Epoch 121/600
14/14 [==============================] - 0s 26ms/step - loss: 0.0816 - val_loss: 0.1167
Epoch 122/600
14/14 [==============================] - 0s 24ms/step - loss: 0.0932 - val_loss: 0.1154
Epoch 123/600
14/14 [==============================] - 0s 20ms/step - loss: 0.0762 - val_loss: 0.1153
Epoch 124/600
14/14 [==============================] - 0s 21ms/step - loss: 0.0789 - val_loss: 0.1112
Epoch 125/600
14/14 [==============================] - 0s 24ms/step - loss: 0.0776 - val_loss: 0.1141
Epoch 00125: early stopping
```

Out[18]:

```
<tensorflow.python.keras.callbacks.History at 0x269fb2e1d90>
```

Running Tensorboard

Running through the Command Line

Watch video to see how to run Tensorboard through a command line call.

Tensorboard will run locally in your browser at http://localhost:6006/

In [19]:

```
print(log_directory)
```

```
logs\fit
```

In [20]:

```
pwd
```

Out[20]:

```
'C:\\Users\\Yasin\\Desktop\\Machine Excercise\\New folder\\Deep Learning and Neural Network'
```

Use cd at your command line to change directory to the file path reported back by pwd or your current .py file location.

Then run this code at your command line or terminal

In [ ]:

```
tensorboard --logdir logs\fit
```