neural network layers. The Functional model is used if a more complex graph is desired. Keras is the de facto API for building neural network architectures with TensorFlow.

From here on, the code examples in this book will use the Sequential API, Functional API, and Model subclassing methods for building neural network architectures with Keras. In doing this, the reader can play around with the various examples as samples to get a feel of how they work.

TensorBoard

TensorBoard is an interactive visualization tool that comes bundled with TensorFlow. The goal of TensorBoard is to gain a visual insight into how the computational graph is constructed and executed. This information provides greater visibility for understanding, optimizing, and debugging deep learning models.

TensorBoard has a variety of visualization dashboard, such as

- Scalar dashboard: This dashboard captures metrics that change with time, such as the loss of a model or other model evaluation metrics such as accuracy, precision, recall, f1, and so on.
- Histogram dashboard: This dashboard shows the histogram distribution for a Tensor as it has changed over time.
- Distribution dashboard: This dashboard is similar to the histogram dashboard. However, it displays the histogram as a distribution.
- Graph explorer: This dashboard gives a graphical overview of the TensorFlow computational graph and how information flows from one node to the other. This dashboard provides invaluable insights into the network architecture.
- Image dashboard: This dashboard displays images saved using the method **tf.summary.image**.
- Audio dashboard: This dashboard provides audio clips saved using the method **tf.summary.audio**.

- Embedding projector: The dashboard makes it easy to visualize high-dimensional datasets after they have been transformed using **Embeddings**. The visualization uses principal component analysis (PCA) and another technique called t-distributed Stochastic Neighbor Embedding (t-SNE). Embedding is a technique for capturing the latent variables in a high-dimensional dataset by converting the data units into real numbers that capture their relationship. This technique is broadly similar to how PCA reduces data dimensionality. Embeddings are also useful for converting sparse matrices (matrices made up of mostly zeros) into a dense representation.
- Text dashboard: This dashboard is for displaying textual information.

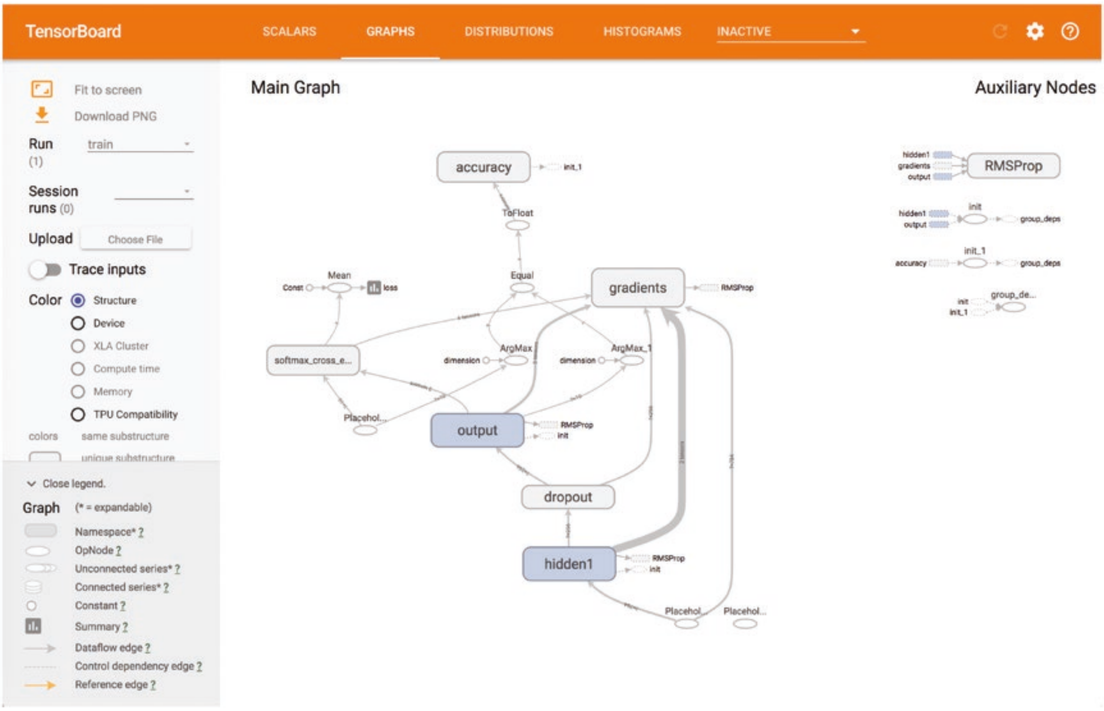

Figure 30-8. *TensorBoard*

Features in TensorFlow 2.0

TensorFlow 2.0 comes with new features for building machine learning models. Some of these new features include

- A more pythonic feel to model design and debugging with eager execution as the de facto execution mode.
- Eager execution enables instant evaluation of TensorFlow operations. This is opposed to previous versions of Tensorflow where we first construct a computational graph and then execute it in a session.
- Using tf.function to transform a Python method into high-performance TensorFlow graphs.
- Using Keras as the core high-level API for model design.
- Using FeatureColumns to parse data as input into Keras models.
- The ease of training on distributed architectures and devices.

To install and work with TensorFlow 2.0 on Google Colab, run

```
!pip install -q tensorflow==2.0.0-beta0
```

The GCP Deep Learning VM has images with TensorFlow 2.0 pre-configured.

A Simple TensorFlow Program

Let's start by building a simple TF program. Here, we will build a graph to find the roots of the quadratic expression $x^2 + 3x - 4 = 0$.

```
# import tensorflow
import tensorflow as tf

# Quadratic expression: x**2 + 3x - 4 = 0.
a = tf.constant(1.0)
b = tf.constant(3.0)
c = tf.constant(-4.0)
```