(B.TECH) Semester-VII AY 2023-24
DL Lab Assignment No. 03

Student Name: Rohit Saini_____ **PRN No.:1032200897**

Date: 06-10-2023 **Faculty: Prof. Anita Gunjal**

Problem Statement: To implement the Graph in TensorBoard.

Objectives:

1. To implement simple addition or multiplication operation using tensor and it's the graph representation with TensorBoard.

Theory: (describe the following)

- TensorBoard and visualization Analysis

TensorBoard is a powerful visualization tool and analysis platform developed by Google as part of the TensorFlow ecosystem. It is primarily used for monitoring and visualizing various aspects of machine learning experiments and deep learning models. TensorBoard provides a user-friendly interface to help researchers and developers gain insights into their models, track training progress, and make informed decisions about model improvements. Here's a breakdown of what TensorBoard offers:

- **Visualization of Scalars:** TensorBoard allows you to track and visualize scalar values over time, such as loss, accuracy, or custom metrics. You can view these metrics as interactive plots to understand how your model's performance changes during training.

- **Graph Visualization:** One of TensorBoard's most essential features is the ability to visualize the computational graph of your deep learning model. This graph provides a detailed overview of the model's architecture, making it easier to understand and debug complex networks.

- **Histograms and Distributions:** You can inspect the distributions of weights and biases in your neural network layers using histograms. This is valuable for identifying issues like vanishing or exploding gradients during training.

- **Embeddings:** TensorBoard allows you to visualize high-dimensional data using dimensionality reduction techniques like Principal Component Analysis (PCA) or t-SNE. This can be helpful when working with embeddings, such as word embeddings or feature embeddings in neural networks.

- **Image Summaries**: For computer vision tasks, TensorBoard can display image summaries, helping you visualize input images, ground truth labels, and model predictions side by side.

Operations to be performed:

1. Implement few examples using TensorFlow.
2. Perform a simple operation using TensorBoard

Program code:

In [1]:
```
#Lab 3
import tensorflow as tf
%load_ext tensorboard
#load tensorBoard extension

with tf.compat.v1.Session() as session:
    a = tf.constant(4,name="input_A")
    b = tf.constant(6,name="input_b")
    c = tf.add(a,b,name="Result")
    writer = tf.compat.v1.summary.FileWriter("./logs",session.graph)
    print(session.run(c))
    session.close()
```

```
10
```

In [2]:
```
#Q1 Expression: (a+b)*(c-d)*(e/f)
import tensorflow as tf
%reload_ext tensorboard
%load_ext tensorboard
#load tensorBoard extension

with tf.compat.v1.Session() as session:
    a = tf.constant(4.0,name="input_A")
    b = tf.constant(6.1,name="input_b")
    ans_1 = tf.add(a,b,name="input_1")
    c = tf.constant(4.2,name="input_c")
    d = tf.constant(6.1,name="input_d")
    ans_2 = tf.subtract(c,d,name="input_2")
    e = tf.constant(18.0,name="input_e")
    f = tf.constant(6.0,name="input_f")
    ans_3 = tf.divide(c,d,name="input_3")
    ans_4 = tf.multiply(ans_1,ans_2,name="product_1")
    final = tf.multiply(ans_4,ans_3,name="product_final")
    writer = tf.compat.v1.summary.FileWriter("./logs",session.graph)
    print(session.run(final))
    session.close()
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
-13.212788
```

In [3]:
```
#Q2: (x1-x2)*(x3+x4+x5)
import tensorflow as tf
%reload_ext tensorboard
%load_ext tensorboard
#load tensorBoard extension

with tf.compat.v1.Session() as session:
    x1 = tf.constant(4.0,name="input_x1")
    x2 = tf.constant(6.1,name="input_x2")
    ans_1 = tf.add(x1,x2,name="input_1")
    x3 = tf.constant(4.2,name="input_x3")
    x4 = tf.constant(6.1,name="input_x4")
    x5 = tf.constant(18.0,name="input_x5")
    ans_3 = tf.add(x3,x4,name="input_2")
```

```
    ans_4 = tf.add(x5,ans_3,name="input_2")
    final = tf.multiply(ans_4,ans_1,name="product_final")
    writer = tf.compat.v1.summary.FileWriter("./logs",session.graph)
    print(session.run(final))
    session.close()
```

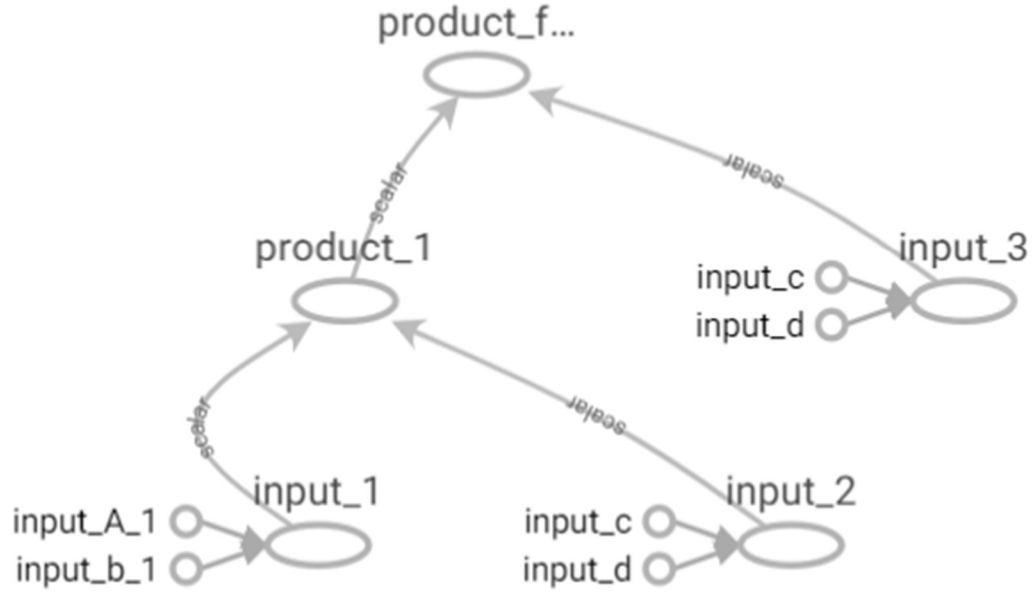
```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
285.83002
```

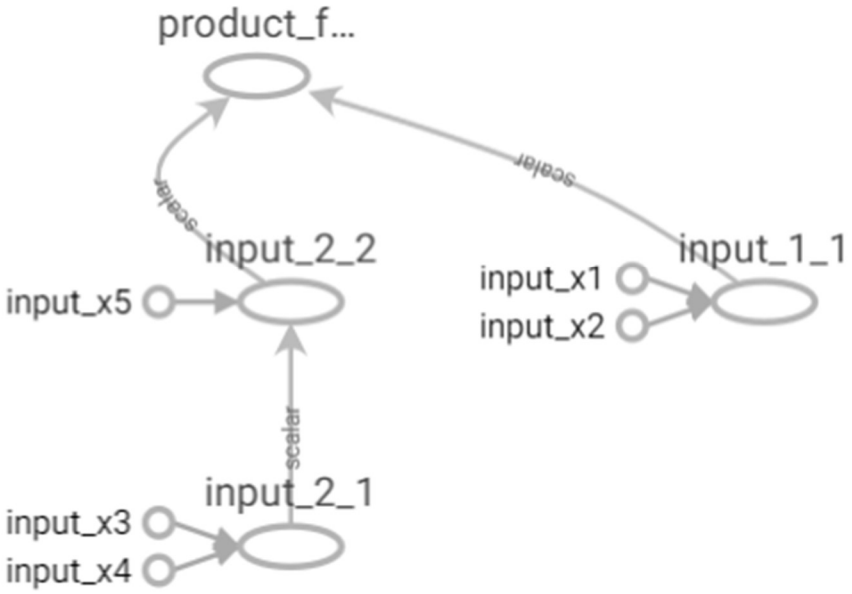
In [4]:

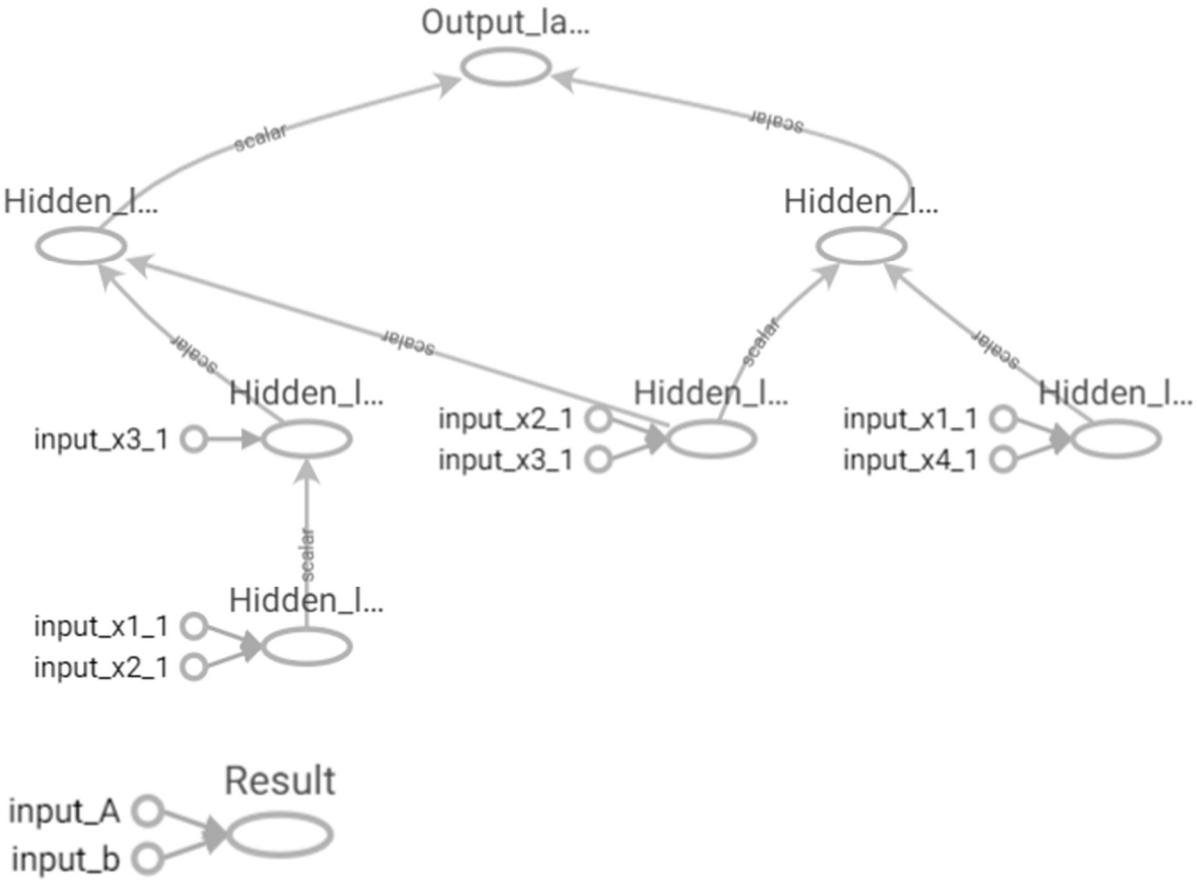
```
# rm -rf logs/
# to delete logs
import tensorflow as tf
%reload_ext tensorboard
%load_ext tensorboard
#load tensorBoard extension

with tf.compat.v1.Session() as session:
    x1 = tf.constant(4.0,name="input_x1")
    x2 = tf.constant(6.1,name="input_x2")
    x3 = tf.constant(4.2,name="input_x3")
    x4 = tf.constant(6.1,name="input_x4")
    ans_1 = tf.add(x1,x2,name="Hidden_layer_1")
    ans_1_1 = tf.add(ans_1,x3,name="Hidden_layer_1")
    ans_2 = tf.add(x2,x3,name="Hidden_layer_1_2")
    ans_3 = tf.add(x1,x4,name="Hidden_layer_1_3")
    ans_4 = tf.add(ans_1_1,ans_2,name="Hidden_layer_2_1")
    ans_5 = tf.add(ans_2,ans_3,name="Hidden_layer_2_2")
    final  = tf.add(ans_4,ans_5,name="Output_layer")
    writer = tf.compat.v1.summary.FileWriter("./logs",session.graph)
    print(session.run(final))
    session.close()
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
45.0
```

Output: **(paste output screen & graphs plotted)**







FAQs:

1) Mention the advantages of analyzing the operations using TensorBoard.
2) Why there's a need for the two phases in Classic Tensorflow?
3) Which are the three main methods of getting data into a TensorFlow program?
4) State the significance of Event logger.
5) Explain the terms with the help of examples:
    a. Constant
    b. Variable
    c. Placeholder
    d. Tensor
    e. Session

Conclusion:

The graph representation with TensorBoard was performed successfully.

Page No.
Date

DL Lab 3

FAQ

Q1. Mention the advantages of analyzing the operations using TensorBoard.

Ans

1. Graph visualization :- TensorBoard allows you to visualize the computational graph of your Tensorflow model. This helps you understand the model's architecture, identify issues such as redundant operations, and debug the graph.
2. Performance Profiling :- It provides insights into the time taken by different operations, helping you optimize your model for efficiency.
3. Real-time Metrics :- TensorBoard provides real time metrics & summaries during traning, such as loss, accuracy, and custom metrics.

Q2. Why there is a need for the two phases is Classic Tensorflow.

Ans.

1. Build Phase : Also known as Construction Phase, In this phase, you define the computational graph that represents your ML model. However, no actual computations take place during this phase;

2. Run Phase : Also known as Execution Phase. Once the computational graph is constructed, you can execute it within a Tensorflow session.

The two phase approach provides several advantages, including the ability to optimize and distribute computations across multiple devices and servers, which is crucial for dl on large datasets.

Page No.
Date

Q3. Which are the three main methods of getting data into a Tensorflow program.

Ans

1. tf.data API : It provides a high-level API for building efficient data pipelines.
2. tf.Keras.Utils.get_file(): This function allows you to download and cache datasets from the internet.
3. tf.data.experimental.CsvDataset(): If you have data in CSV format, you can use this function to create a dataset from CSV files.

Q4 State the significance of Event logger.

Ans. The event logger in TensorBoard plays a crucial role in tracking and visualizing the progress of training and other computation.

(1) Logging summaries.
(2) Real-time Monitoring
(3) Historical Analysis.
(4) Debugging and Optimization.

12/9/2023