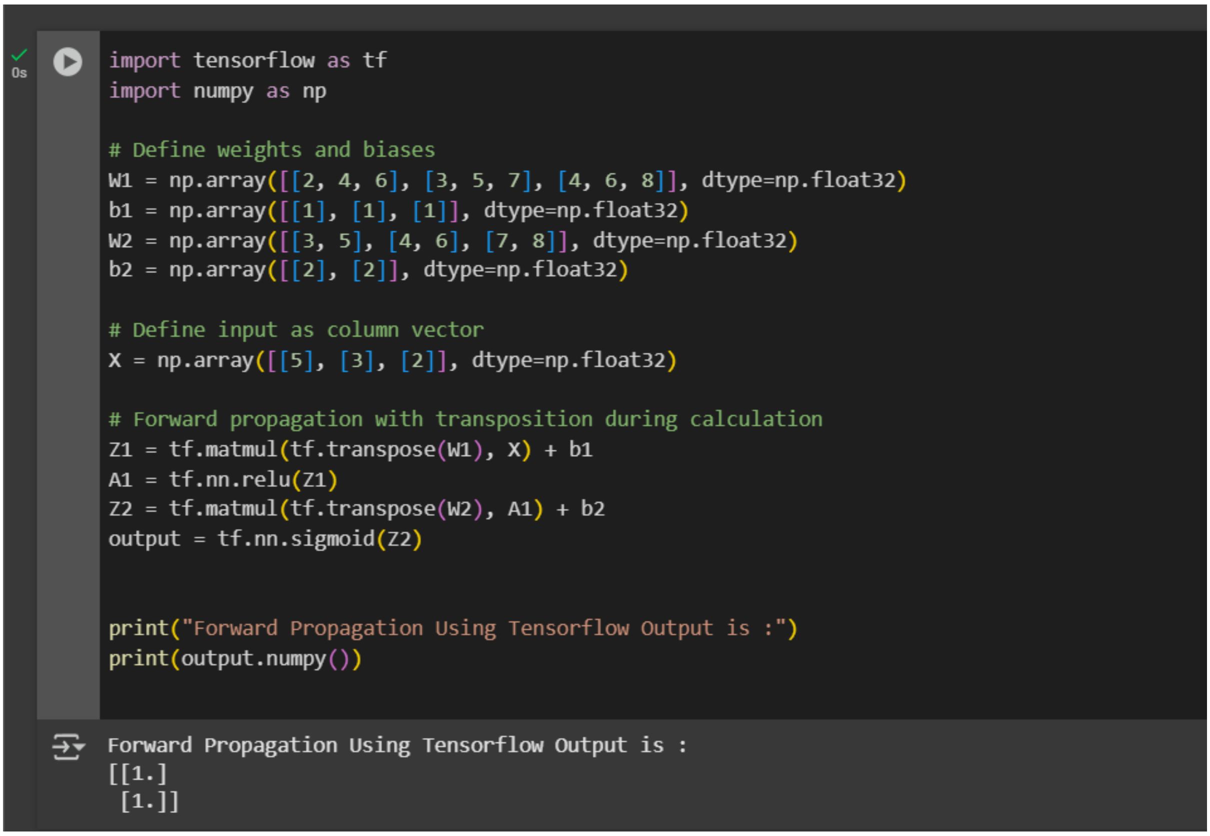
Forward Propagation in TensorFlow

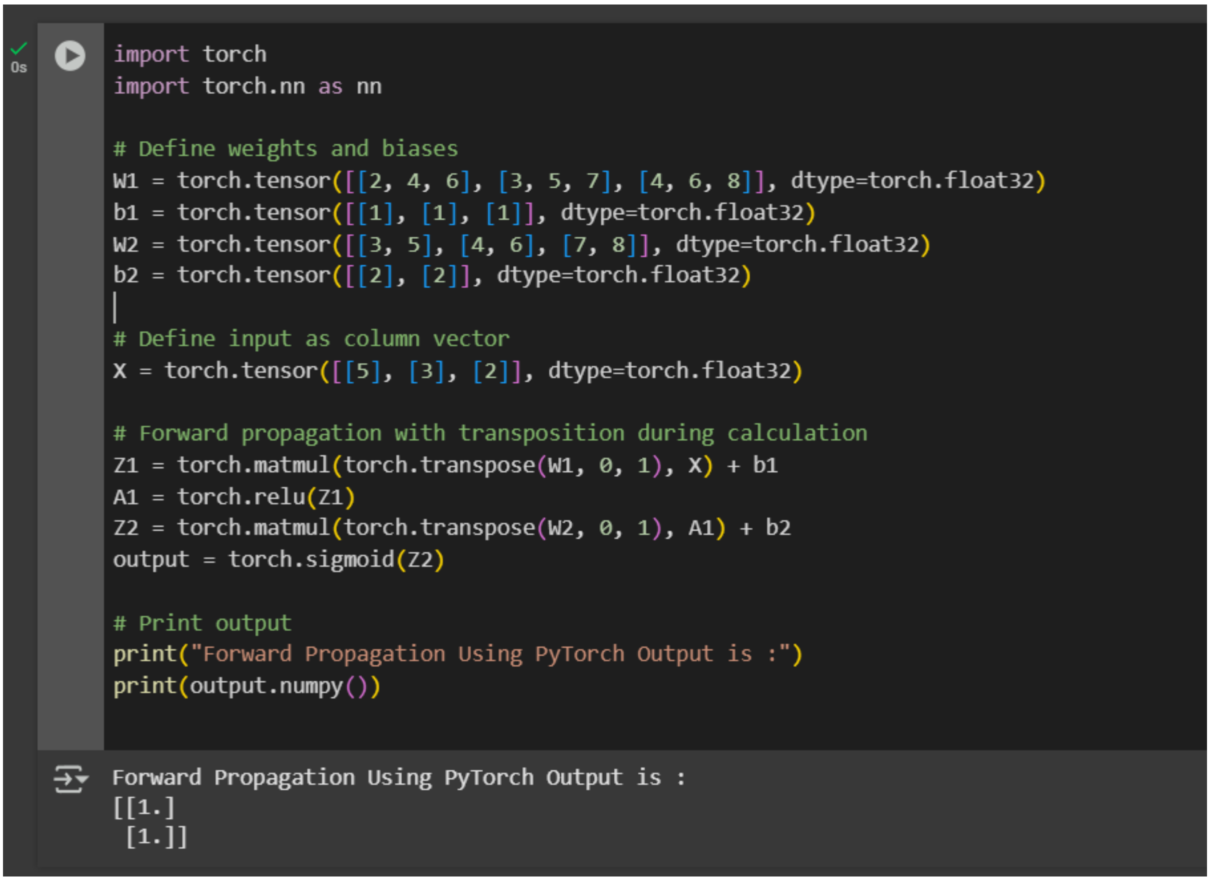


```
import tensorflow as tf
import numpy as np

# Define weights and biases
W1 = np.array([[2, 4, 6], [3, 5, 7], [4, 6, 8]], dtype=np.float32)
b1 = np.array([[1], [1], [1]], dtype=np.float32)
W2 = np.array([[3, 5], [4, 6], [7, 8]], dtype=np.float32)
b2 = np.array([[2], [2]], dtype=np.float32)

# Define input as column vector
X = np.array([[5], [3], [2]], dtype=np.float32)

# Forward propagation with transposition during calculation
Z1 = tf.matmul(tf.transpose(W1), X) + b1
A1 = tf.nn.relu(Z1)
Z2 = tf.matmul(tf.transpose(W2), A1) + b2
output = tf.nn.sigmoid(Z2)


print("Forward Propagation Using Tensorflow Output is :")
print(output.numpy())
```

```
Forward Propagation Using Tensorflow Output is :
[[1.]
 [1.]]
```

Figure 1: Forward Propagation Diagram using TensorFlow

Forward Propagation in PyTorch



```
import torch
import torch.nn as nn

# Define weights and biases
W1 = torch.tensor([[2, 4, 6], [3, 5, 7], [4, 6, 8]], dtype=torch.float32)
b1 = torch.tensor([[1], [1], [1]], dtype=torch.float32)
W2 = torch.tensor([[3, 5], [4, 6], [7, 8]], dtype=torch.float32)
b2 = torch.tensor([[2], [2]], dtype=torch.float32)

# Define input as column vector
X = torch.tensor([[5], [3], [2]], dtype=torch.float32)

# Forward propagation with transposition during calculation
Z1 = torch.matmul(torch.transpose(W1, 0, 1), X) + b1
A1 = torch.relu(Z1)
Z2 = torch.matmul(torch.transpose(W2, 0, 1), A1) + b2
output = torch.sigmoid(Z2)

# Print output
print("Forward Propagation Using PyTorch Output is :")
print(output.numpy())
```

```
Forward Propagation Using PyTorch Output is :
[[1.]
 [1.]]
```

Figure 2: Forward Propagation Diagram using PyTorch