빅데이터와 머신러닝 소프트웨어

텐서플로우 프로그래밍

TensorFlow

- **Symbolic graph style: define-and-run**
  - Step 1 : Define a graph, which contains model architecture, parameter specifications, optimization process, etc.
  - Step 2 : Run the graph through a session (Session.run()), a binding to a particular execution context (e.g. CPU, GPU)
    - Initialize the session
    - Feed data and fetch results

TensorFlow Programming Example

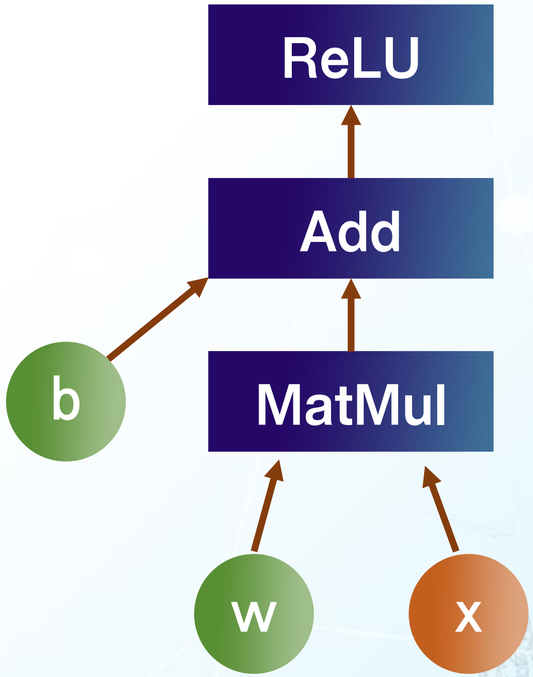
Define a graph: h = ReLU(Wx + b)

```
import tensorflow as tf

b = tf.get_variable('bias', tf.zeros((100,)))
W = tf.get_variable('weights’,
        tf.random_uniform((784, 100), -1, 1))

x = tf.placeholder(tf.float32, (None, 784))

h = tf.nn.relu(tf.matmul(x, W) + b)
```



TensorFlow Programming Example

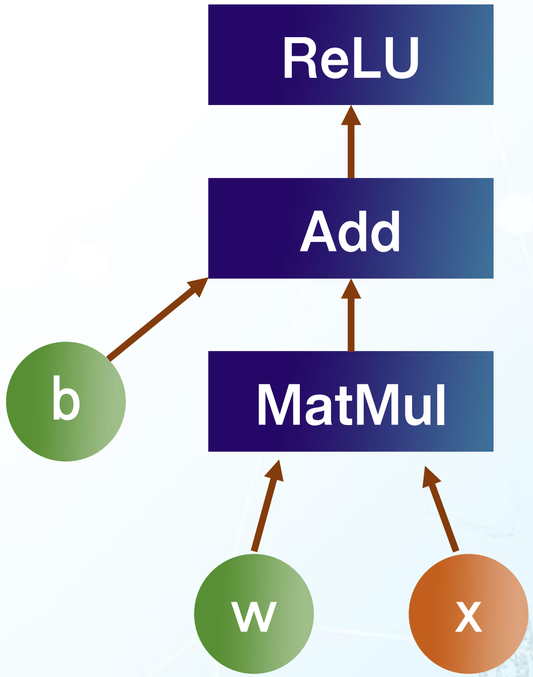
Define a graph

```
import tensorflow as tf

b = tf.get_variable('bias', tf.zeros((100,)))
W = tf.get_variable('weights’,
        tf.random_uniform((784, 100), -1, 1))

x = tf.placeholder(tf.float32, (None, 784))

h = tf.nn.relu(tf.matmul(x, W) + b)
```



TensorFlow Programming Example

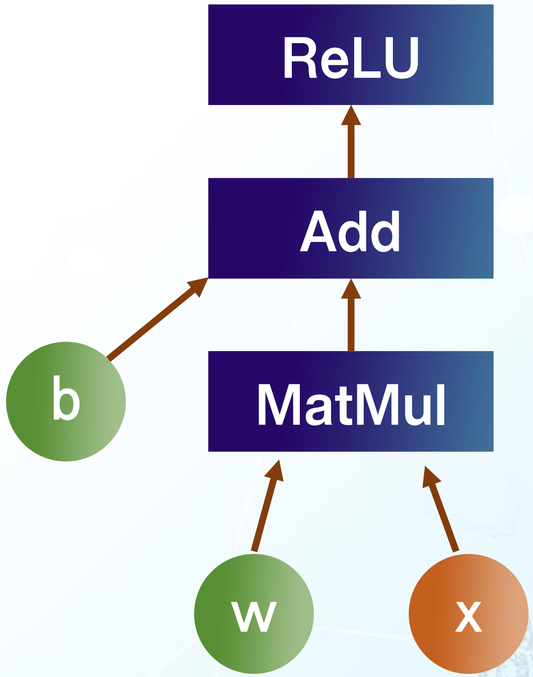
Define a graph

```
import tensorflow as tf

b = tf.get_variable('bias', tf.zeros((100,)))
W = tf.get_variable('weights',
        tf.random_uniform((784, 100), -1, 1))

x = tf.placeholder(tf.float32, (None, 784))

h = tf.nn.relu(tf.matmul(x, W) + b)
```



TensorFlow Programming Example

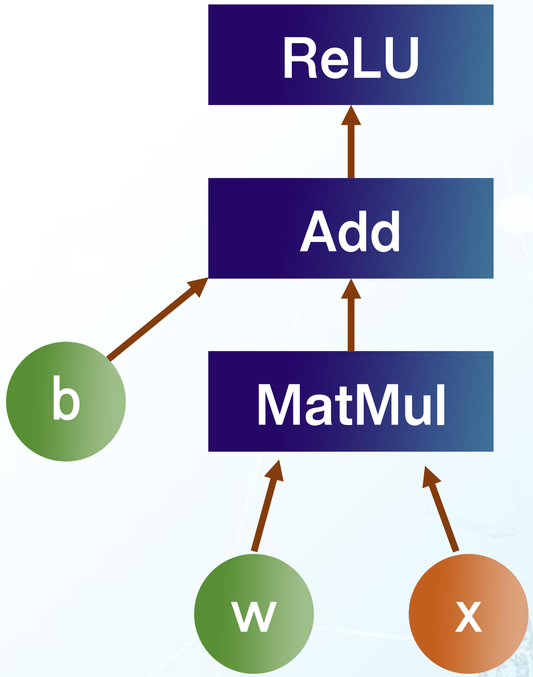
Define a graph

```
import tensorflow as tf

b = tf.get_variable('bias', tf.zeros((100,)))
W = tf.get_variable('weights’,
        tf.random_uniform((784, 100), -1, 1))

x = tf.placeholder(tf.float32, (None, 784))

h = tf.nn.relu(tf.matmul(x, W) + b)
```



TensorFlow Programming Example

- We can deploy the graph with a session

```
import tensorflow as tf
 import numpy as np

 b = tf.get_variable('bias', tf.zeros((100,)))
 W = tf. get_variable('weights’,
           tf.random_uniform((784, 100), -1, 1))
 x = tf.placeholder(tf.float32, (None, 784))
 h = tf.nn.relu(tf.matmul(x, W) + b)

 sess = tf.Session()
 sess.run(tf.initialize_all_variables())
 sess.run(h, {x: np.random.random(64, 784})
```

TensorFlow Eager Mode

- **Enabling eager execution requires two lines of code**

```
import tensorflow as tf
import tensorflow.contrib.eager as tfe
tfe.enable_eager_execution() # Call this at program start-up
```

- **Lets you write code that you can easily execute in a REPL**

```
x = [[3.]]  # No need for placeholders!
m = tf.matmul(x, x)
print(m)  # No sessions!
# tf.Tensor([[9.]], shape=(1, 1), dtype=float32)
```

TensorFlow Example: Linear Regression

```
import tensorflow as tf
import utils
DATA_FILE = "data/system_cpuutil_applatency.txt"

# Step 1: read in data from the .txt file
# data is a numpy array of shape (100000, 2), each row is a datapoint
data, n_samples = utils.read_system_cpuutil_applatency(DATA_FILE)

# Step 2: create placeholders for X (CPU util) and Y (App latency)
X = tf.placeholder(tf.float32, name='X')
Y = tf.placeholder(tf.float32, name='Y')
```

TensorFlow Example: Linear Regression

```
# Step 3: create weight and bias, initialized to 0
w = tf.get_variable('weights', initializer=tf.constant(0.0))
b = tf.get_variable('bias', initializer=tf.constant(0.0))

# Step 4: construct model to predict Y (app latency from CPU util)
Y_predicted = w * X + b

# Step 5: use the square error as the loss function
loss = tf.square(Y - Y_predicted, name='loss')

# Step 6: using gradient descent with learning rate of 0.01 to minimize
loss
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.001)
                    .minimize(loss)
```

TensorFlow Example: Linear Regression

```
with tf.Session() as sess:
    # Step 7: initialize the necessary variables, in this case, w and b
    sess.run(tf.global_variables_initializer())

    # Step 8: train the model
    for i in range(100): # run 100 epochs
        for x, y in data:
            # Session runs train_op to minimize loss
            sess.run(optimizer, feed_dict={X: x, Y:y})

    # Step 9: output the values of w and b
    w_out, b_out = sess.run([w, b])
```

TensorFlow Dataset

```
dataset = tf.data.FixedLengthRecordDataset([file1, file2, file3, ...])
iterator = dataset.make_initializable_iterator()
...
for i in range(100):
        sess.run(iterator.initializer)
        total_loss = 0
        try:
            while True:
                sess.run([optimizer])
        except tf.errors.OutOfRangeError:
            pass
```

TensorFlow Dataset

- **Shuffle, repeat, batch your data**

```
dataset = dataset.shuffle(1000)
dataset = dataset.repeat(100)
dataset = dataset.batch(128)
```

- **Map each element of your dataset to transform it in a specific way to create a new dataset**

```
dataset = dataset.map(lambda x: tf.one_hot(x, 10))
# convert each element of dataset to one_hot vector
```