Welcome to TensorFlow!

Agenda

Welcome

Overview of TensorFlow

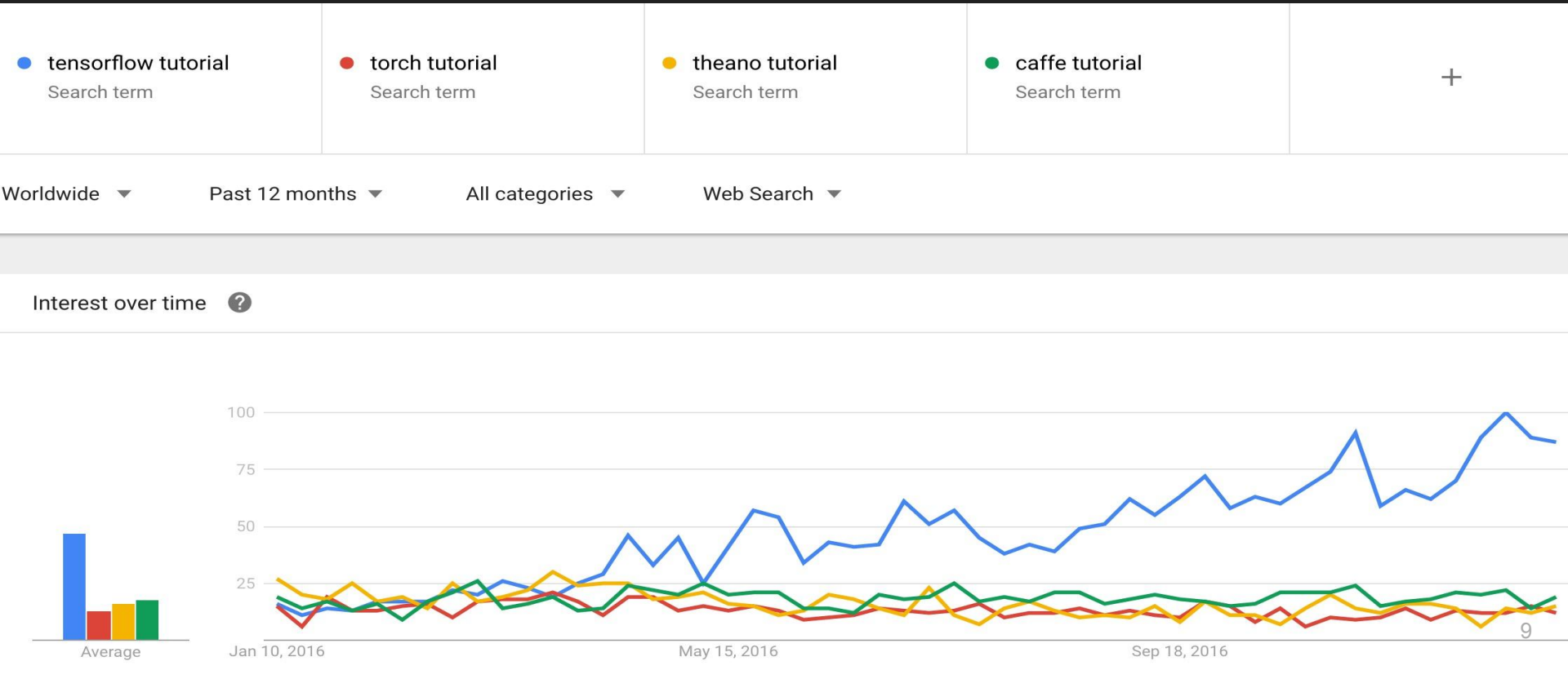
Graphs and Sessions

What's TensorFlow™?

- Open source software library for numerical computation using data flow graphs
- Originally developed by Google Brain Team to conduct machine learning and deep neural networks research
- General enough to be applicable in a wide variety of other domains as well

TensorFlow provides an extensive suite of functions and classes that allow users to build various models from scratch.

Why TensorFlow?



Why TensorFlow?

- Python API
- Portability: deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API
- Flexibility: from Raspberry Pi, Android, Windows, iOS, Linux to server farms
- Visualization (TensorBoard is da bomb)
- Checkpoints (for managing experiments)
- Auto-differentiation *autodiff* (no more taking derivatives by hand. Yay)
- Large community (> 10,000 commits and > 3000 TF-related repos in 1 year)
- Awesome projects already using TensorFlow

Companies using Tensorflow

- Google
- OpenAI
- DeepMind
- Snapchat
- Uber
- Airbus
- eBay
- Dropbox
- A bunch of startups

Books

- TensorFlow for Machine Intelligence (TFFMI)
- Hands-On Machine Learning with Scikit-Learn and TensorFlow. Chapter 9: Up and running with TensorFlow
- Fundamentals of Deep Learning. Chapter 3: Implementing Neural Networks in TensorFlow (FODL)

TensorFlow is being constantly updated so books might become outdated fast

Check tensorflow.org directly

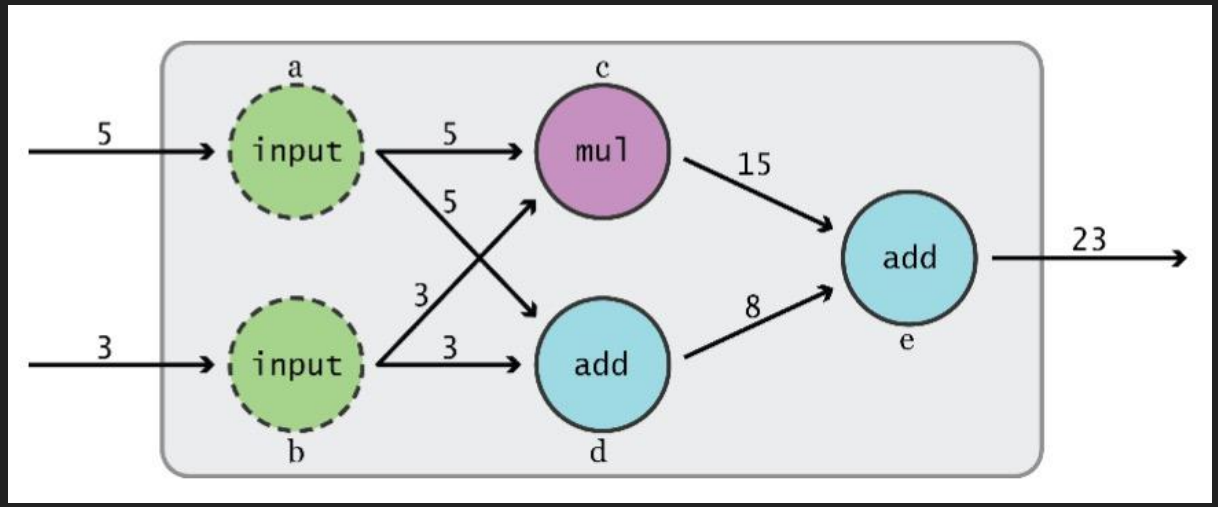
Getting Started

```
import tensorflow as tf
```

Graphs and Sessions

Data Flow Graphs

TensorFlow separates definition of computations from their execution

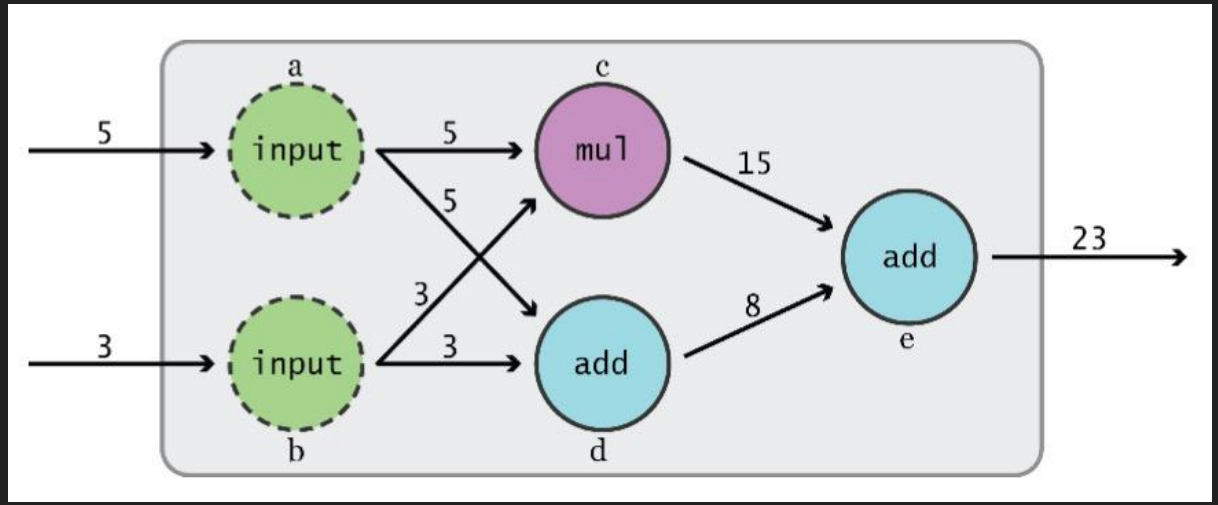


Graph by TFFMI

Data Flow Graphs

Phase 1: assemble a graph

Phase 2: use a session to execute operations in the graph.



Graph by TFFMI

What’s a tensor?

What's a tensor?

An n-dimensional array

0-d tensor: scalar (number)

1-d tensor: vector

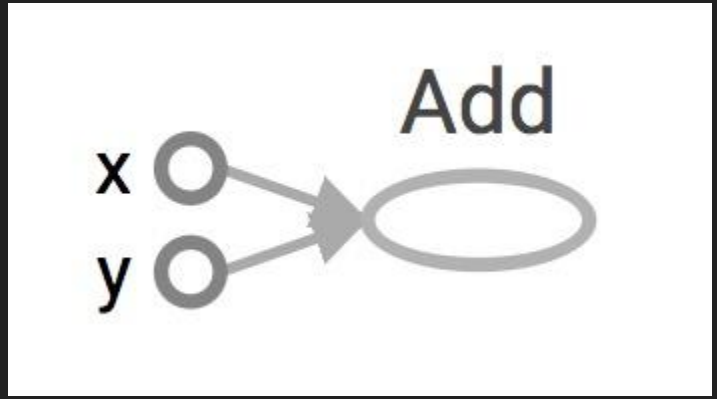
2-d tensor: matrix

and so on

Data Flow Graphs

```
import tensorflow as tf

a = tf.add(3, 5)
```

Visualized by TensorBoard



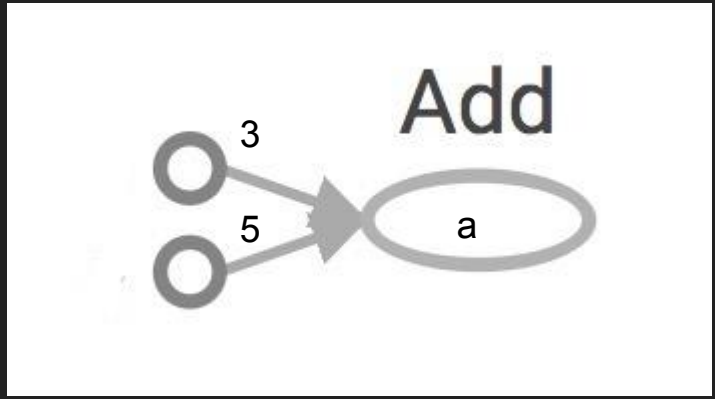
Why x, y?

TF automatically names the nodes when you don't explicitly name them.
x = 3
y = 5

Data Flow Graphs

```
import tensorflow as tf

a = tf.add(3, 5)
```

Interpreted?

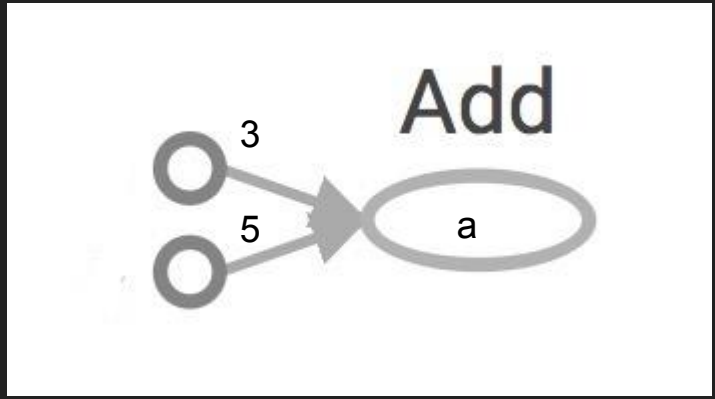


Nodes: operators, variables, and constants
Edges: tensors

Data Flow Graphs

```
import tensorflow as tf

a = tf.add(3, 5)
```

Interpreted?

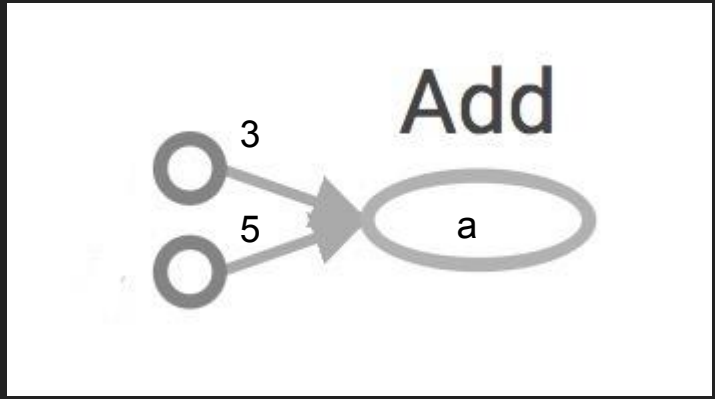


Nodes: operators, variables, and constants
Edges: tensors

Tensors are data.
Data Flow -> Tensor Flow

Data Flow Graphs

```
import tensorflow as tf

a = tf.add(3, 5)

print  a
```




```
>> Tensor("Add:0", shape=(), dtype=int32)
```

(Not 8)

How to get the value of a?

Create a **session**, assign it to variable sess so we can call it later

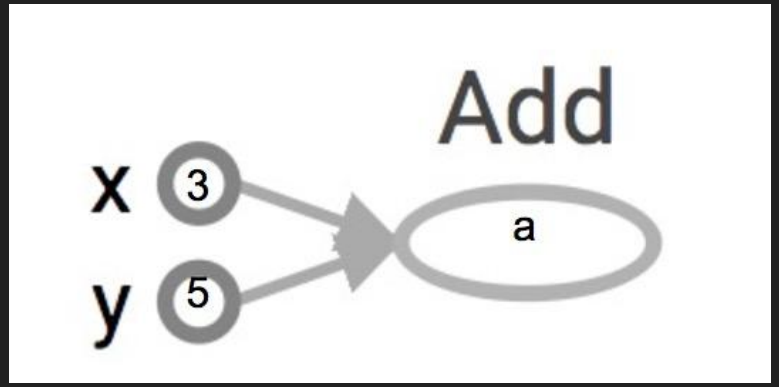
Within the session, evaluate the graph to fetch the value of a

How to get the value of a?

Create a **session**, assign it to variable sess so we can call it later

Within the session, evaluate the graph to fetch the value of a

```
import tensorflow as tf

a = tf.add(3, 5)

sess = tf.Session()

print sess.run(a)

sess.close()
```



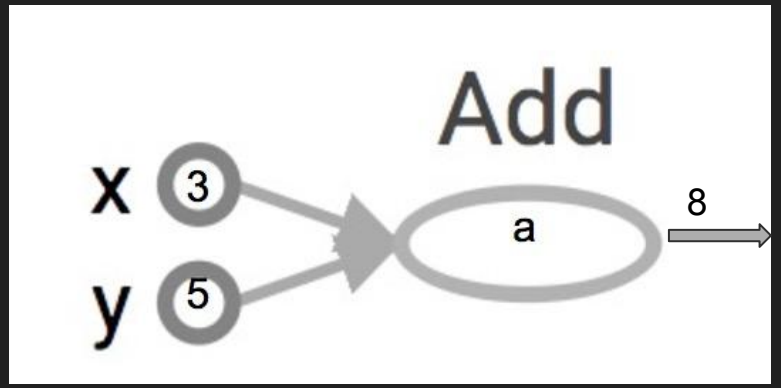
The session will look at the graph, trying to think: hmm, how can I get the value of a, then it computes all the nodes that leads to a.

How to get the value of a?

Create a **session**, assign it to variable sess so we can call it later

Within the session, evaluate the graph to fetch the value of a

```
import tensorflow as tf

a = tf.add(3, 5)

sess = tf.Session()

print sess.run(a)        >> 8

sess.close()
```



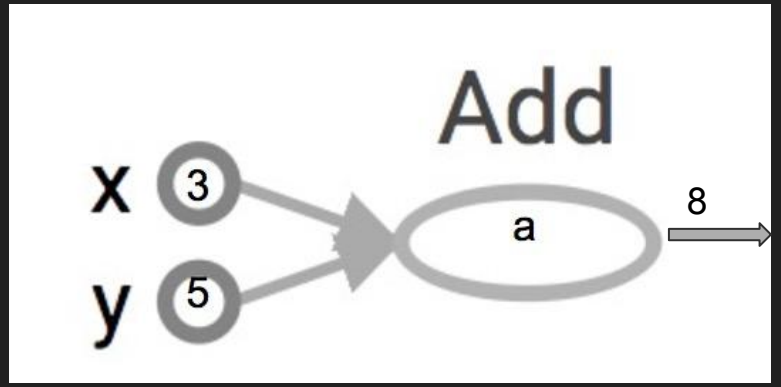
The session will look at the graph, trying to think: hmm, how can I get the value of a, then it computes all the nodes that leads to a.

How to get the value of a?

Create a **session**, assign it to variable sess so we can call it later

Within the session, evaluate the graph to fetch the value of a

```
import tensorflow as tf

a = tf.add(3, 5)

sess = tf.Session()

with tf.Session() as sess:

      print sess.run(a)

sess.close()
```



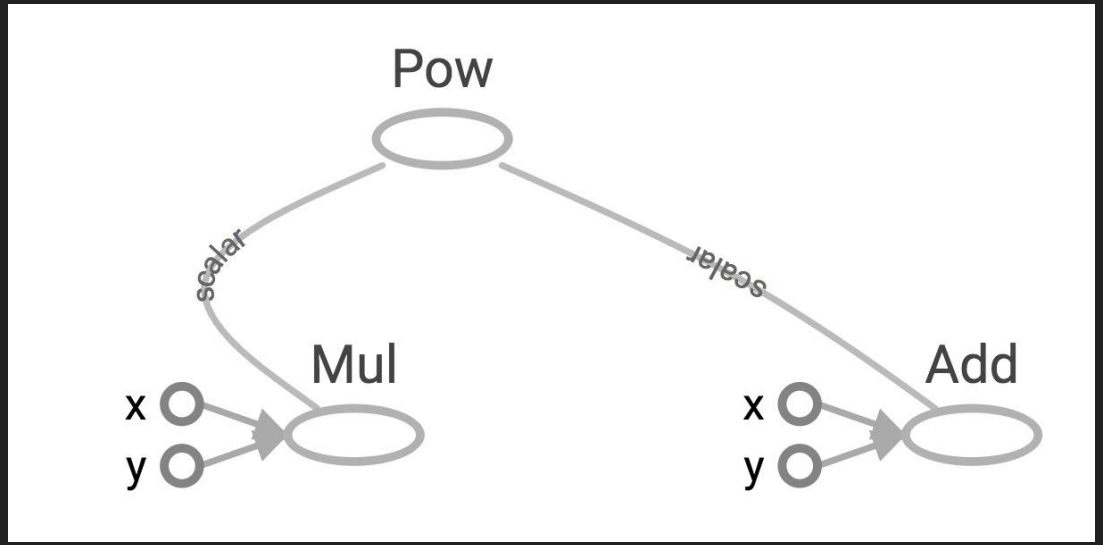
tf.Session()

A Session object encapsulates the environment in which Operation objects are executed, and Tensor objects are evaluated.

More graphs

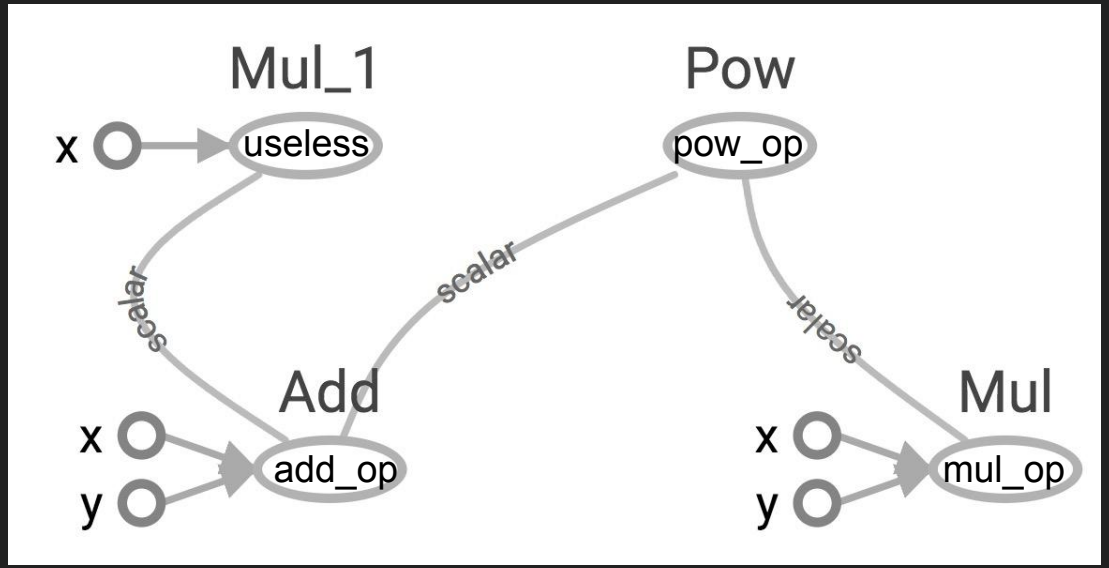
Visualized by TensorBoard

```
x = 2

y = 3

op1 = tf.add(x, y)

op2 = tf.mul(x, y)

op3 = tf.pow(op2, op1)

with tf.Session() as sess:

      op3 = sess.run(op3)
```



Subgraphs

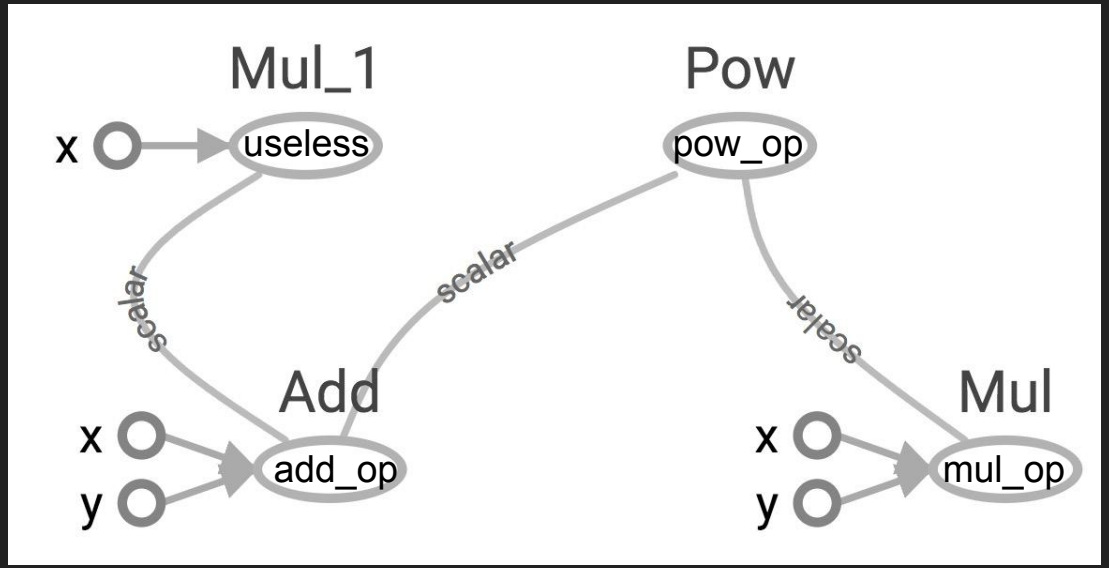
```
x = 2

y = 3

add_op = tf.add(x, y)

mul_op = tf.mul(x, y)

useless = tf.mul(x, add_op)

pow_op = tf.pow(add_op, mul_op)

with tf.Session() as sess:

      z = sess.run(pow_op)
```



Because we only want the value of pow_op and pow_op doesn't depend on useless, session won't compute value of useless
→ save computation

Subgraphs

```
x = 2

y = 3

add_op = tf.add(x, y)

mul_op = tf.mul(x, y)

useless = tf.mul(x, add_op)

pow_op = tf.pow(add_op, mul_op)

with tf.Session() as sess:

     z, not_useless = sess.run([op3, useless])
```



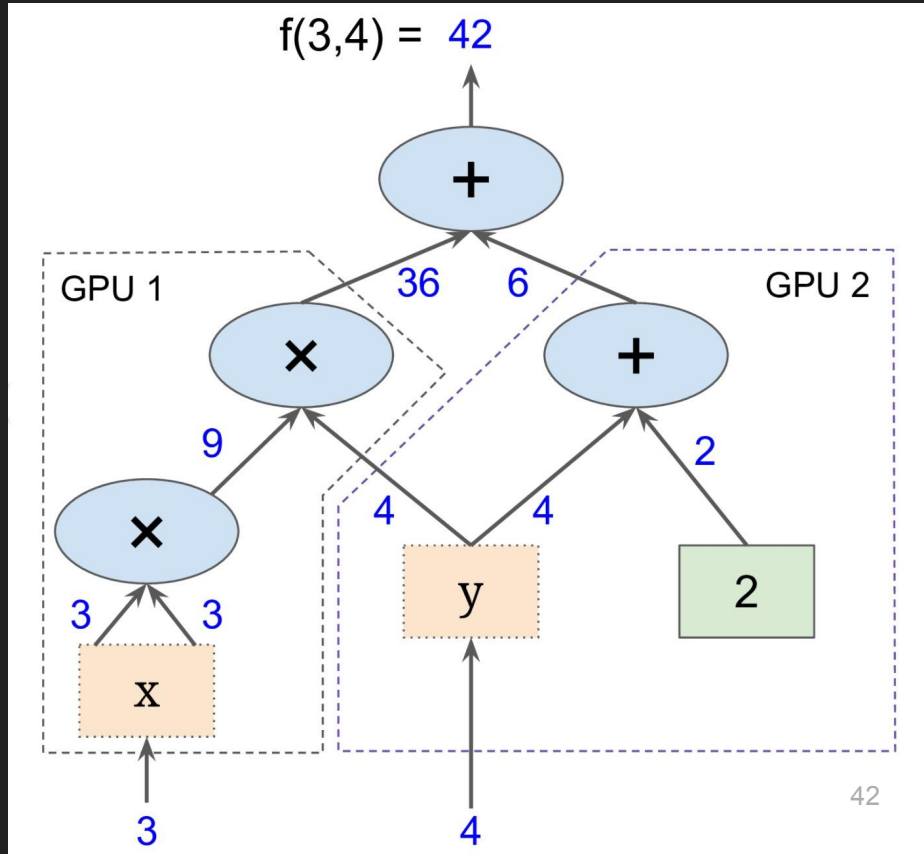

```
tf.Session.run(fetches, feed_dict=None,
options=None, run_metadata=None)
```

pass all variables whose values you want to a list in fetches

Subgraphs

Possible to break graphs into several chunks and run them parallelly across multiple CPUs, GPUs, or devices

Example: AlexNet



Graph from the book "Hands-On Machine Learning with Scikit-Learn and TensorFlow"

Distributed Computation

To put part of a graph on a specific CPU or GPU:

```
# Creates a graph.
with tf.device('/gpu:2'):
  a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], name='a')
  b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], name='b')
  c = tf.matmul(a, b)

# Creates a session with log_device_placement set to True.
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))

# Runs the op.
print sess.run(c)
```

Not covering distributed version of TensorFlow in this module