Introduction to Tensorflow

Subtitle

What does a deep learning library do?

- High-level Deep Learning API (Keras)
  - Convolution layer, fully connected layer

- Deep learning libraries (Tensorflow, Caffe, etc)
  - 2D Convolution, Backward propagation, Optimization

- Neural network computation (cuDNN)
  - Convolution, Pooling, Normalization

- Low level computation (CUDA)
  - BLAS : basic linear algebra subroutines
  - FFT : fast Fourier transform
  - LAPACK-like : matrix factorization, equation solver, etc.

Deep Learning Libraries

- Tensorflow (by Google)

- Keras (on Tensorflow, MXNet, DL4J, or Microsoft Cognitive Toolkit)

- PyTorch (by Facebook)

- Caffe (by Berkeley Vision and Learning Center)

- MXNet (by Amazon)
- Microsoft Cognitive Toolkit

- Deeplearning4J

- Caffe2 (by NVIDIA and Facebook)

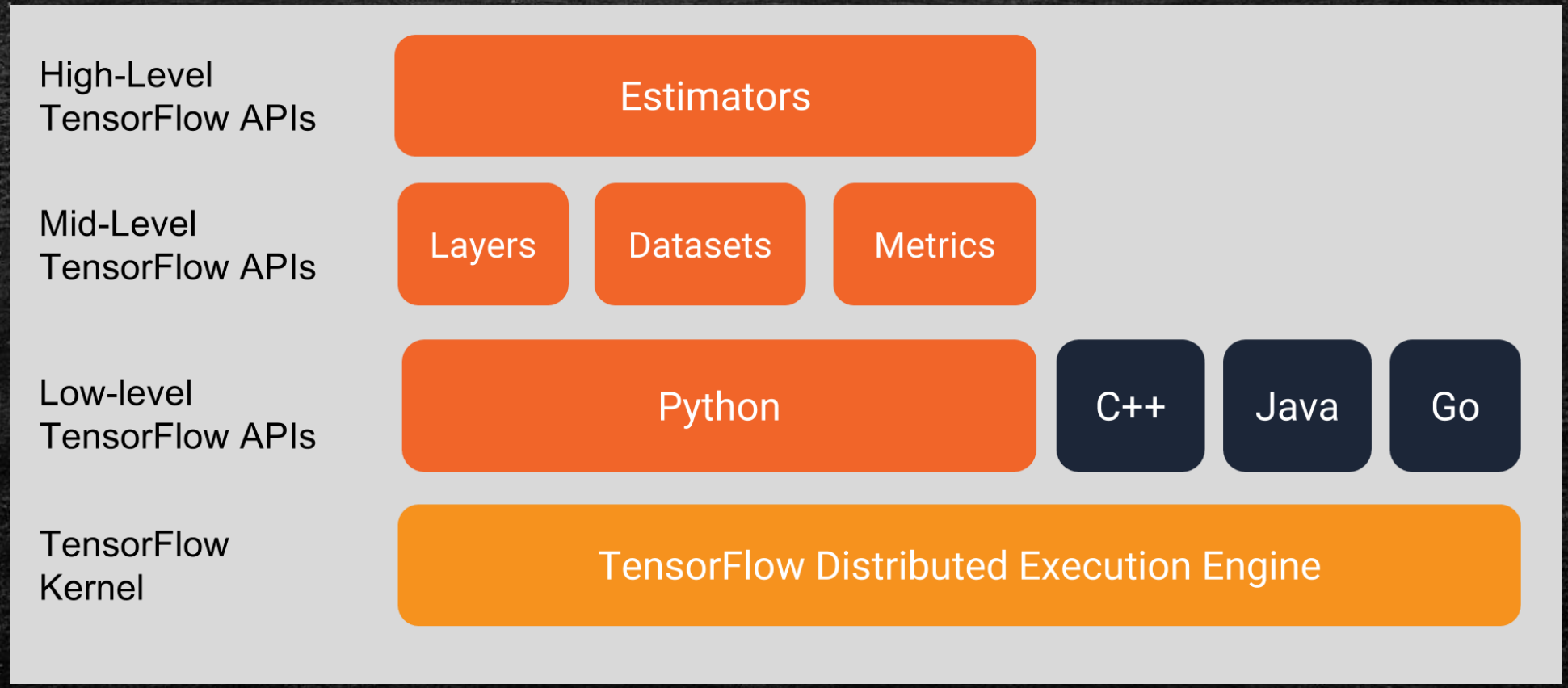
Comparison of Deep Learning Libraries

	User Community	Flexibility	Scalability	Performance	Deployment
Tensorflow	++++	+++	+++	++	++++
PyTorch	++	++++	++	++++	++
MXNet	++	++	++++	++++	+++
Caffe	+++	++	++	+++	+++
DL4J	+	+++	+++	+++	++++

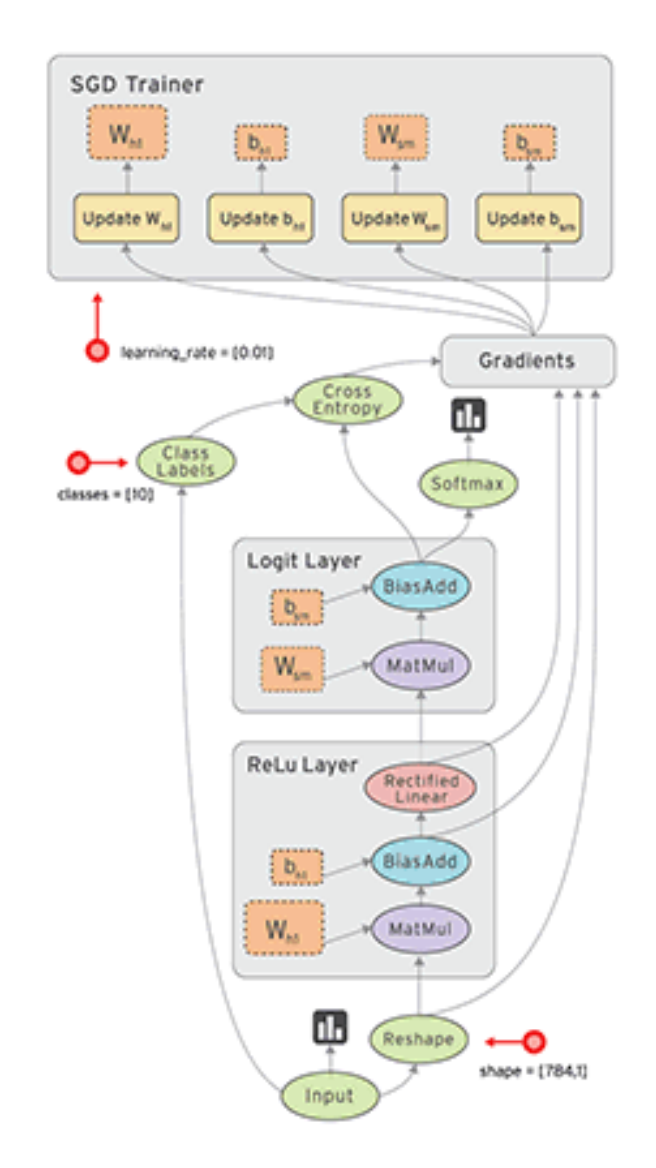
Why Tensorflow ?

- Large user community
  - >92,000 starts on Github as of March, 2018
  - >24,000 Tensorflow questions on StackOverflow
- Plenty of online learning material
- Lots of examples on Github
- Lots of pre-trained models
- Easily deployed to different devices

Tensorflow programming environment

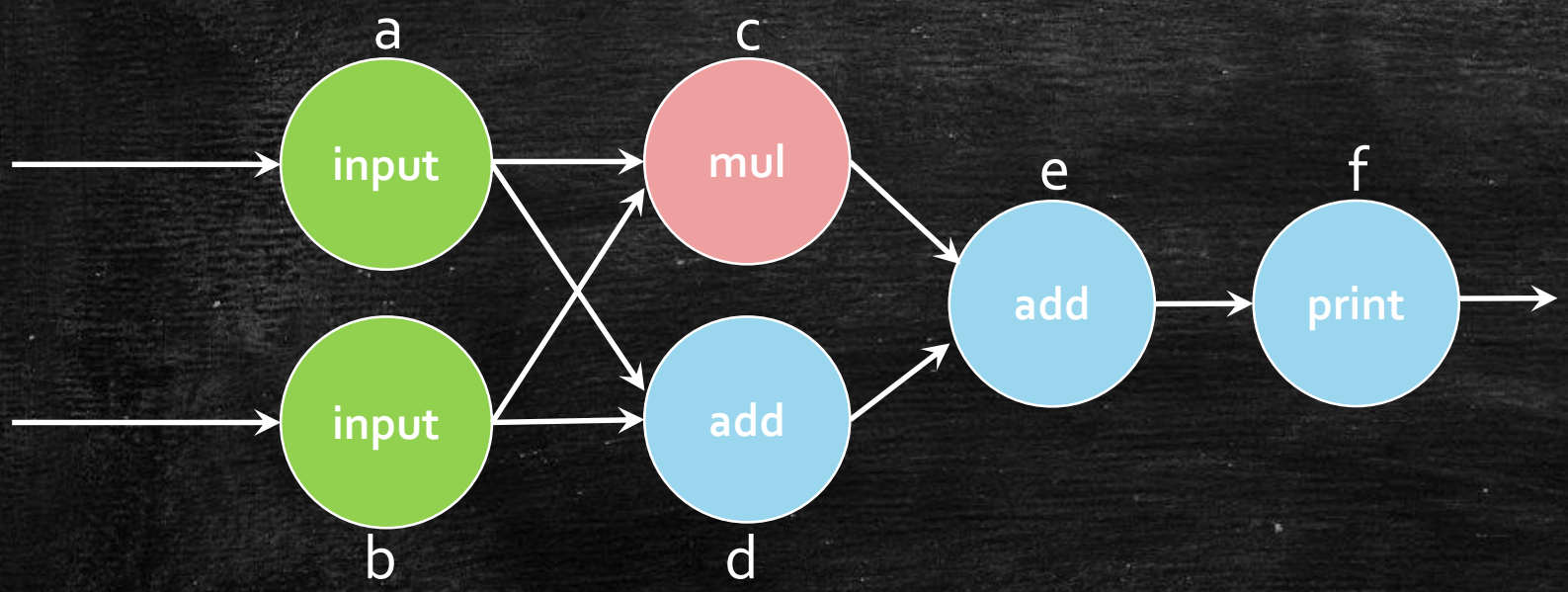


Dataflow graph in Tensorflow



- Advantages of a dataflow model
  - Parallelism
  - Distributed execution
  - Compilation
  - Portability

An example of Tensorflow Graph




```
a = tf.placeholder(tf.float32, None, name='a')
b = tf.placeholder(tf.float32, None, name='b')
c = tf.multiply(a,b)
d = tf.add(a, b)
e = tf.add(c, d)
f = tf.Print(e, [a, b, c, d, e])
```

If we run f :

```
with tf.Session() as sess:
    sess.run(f, feed_dict={a:1, b:2})
```

We'll see in *stderr*:

```
[1][2][2][3][5]
```

If we run c :

```
with tf.Session() as sess:
    print(sess.run(c, feed_dict={a:1, b:2}))
```

We'll see:

```
2
```

Tensorflow Session

- tf.Session class represents a connection between client program (typically a python program) and the tensorflow C++ runtime.
- A tf.Session object provides access to devices in the local machine and remote devices using distributed tensorflow runtime.
- Since tf.Session owns physical resources (e.g. GPU), it's typically used as a context manager

Data container in Tensorflow

- tf.placeholder
  - An empty container to receive input data when you begin to train the network
- tf.variable
  - An entity that stores persistent information during training
  - For example, weights and biases of convolution kernel