Introduction to TensorFlow

Glenn Bruns
CSUMB

Many figures in this deck from Géron, Hands-on Machine Learning with Scikit-Learn and TensorFlow

Learning outcomes

After this lecture you should be able to:

1. List some of the features of TensorFlow
2. Explain the basic concept behind TensorFlow programs
3. List the main parts of a TensorFlow program
4. Explain the details of a simple TensorFlow program

Hey, where's the deep learning?

Patience, grasshopper, we will get to deep learning with TensorFlow soon.

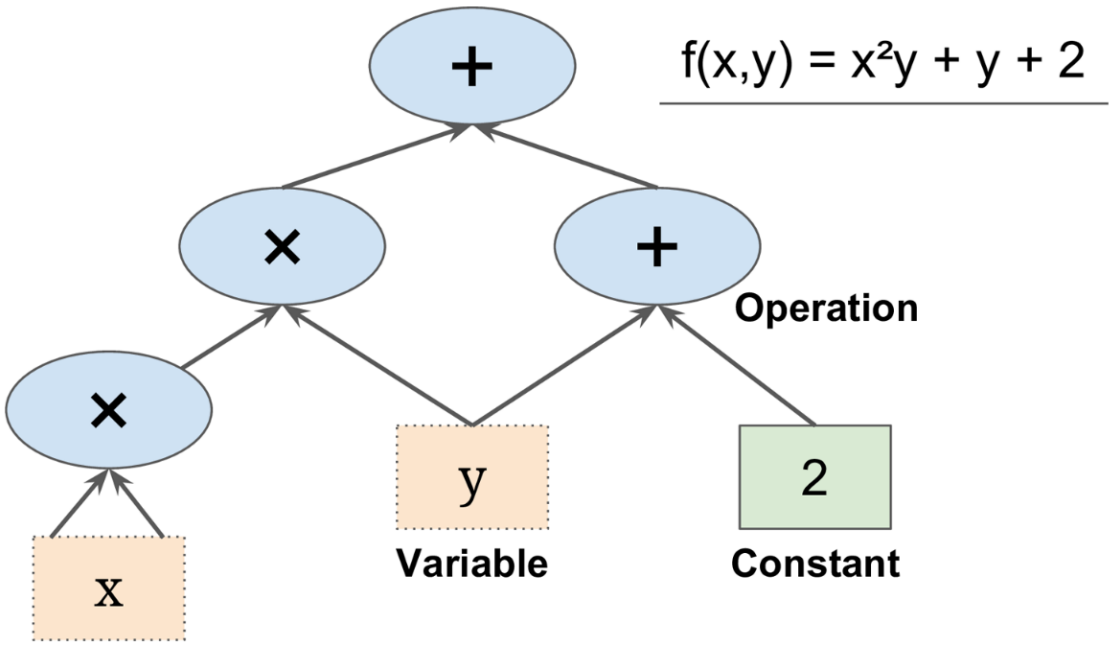
We're talking about TensorFlow first because it is a general-purpose library for numerical computing.

What is TensorFlow?

It's an open source library for numerical computation.

Principle:

1. define a computation graph
2. execute the graph efficiently with C++ code



You can even break the graph into pieces and run them on separate CPUs or GPUs.

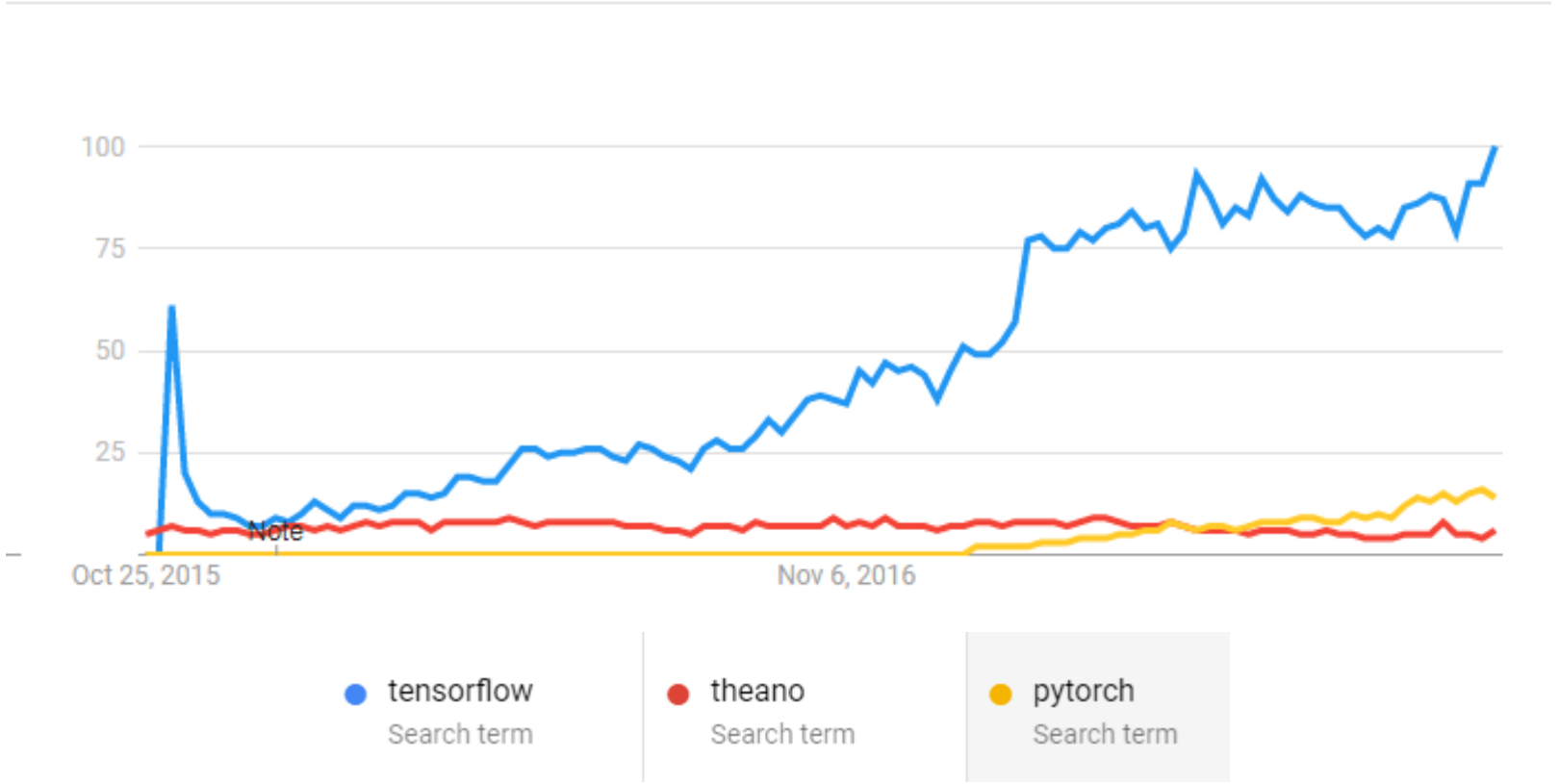
Some features of TensorFlow

TensorFlow was released by Google in Nov. 2015

- Runs on Windows, macOS, and Linux
- Highly efficient C++ implementation
- Provides a very flexible Python API
- Comes with a visualization tool called TensorBoard
- Higher-level APIs have been built on top of it

TensorFlow popularity

Google Trends data:

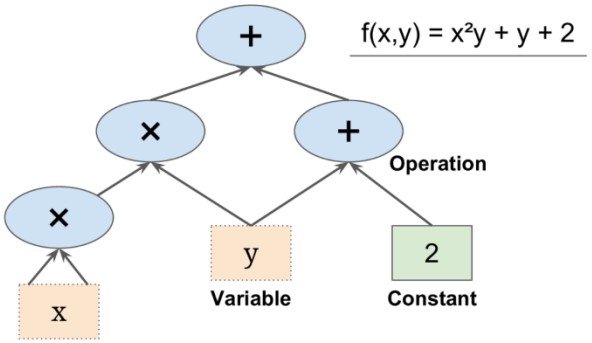


Some other deep learning libraries: Caffe, H2O, MXNet, Deeplearning4j

A simple TensorFlow program

Construction phase: build a computation graph

```
x = tf.Variable(3, name="x")
y = tf.Variable(4, name="y")
f = x*x*y + y + 2
```



Execution phase: initialize and evaluate variables

```
sess = tf.Session()
sess.run(x.initializer)
sess.run(y.initializer)
result = sess.run(f)
print(result)
```

evaluation is done within a TensorFlow session

output:

```
42
```

Alternatives for session handling

1

```
sess = tf.Session()
sess.run(x.initializer)
sess.run(y.initializer)
result = sess.run(f)
print(result)
```

a pain to write sess.run a lot

2

```
with tf.Session() as sess:
    x.initializer.run()
    y.initializer.run()
    result = f.eval()
```

sess is default session within block; session is automatically closed at end

3

```
init = tf.global_variables_initializer()
with tf.Session() as sess:
    init.run()
    result = f.eval()
```

initialize all variables at once; first statement is just a node

Interactive sessions

third alternative from last slide

```
init = tf.global_variables_initializer()
with tf.Session() as sess:
    init.run()
    result = f.eval()
```

interactive session

```
sess = tf.InteractiveSession()
init.run()
result = f.eval()
print(result)
```

sess automatically becomes default session

handy in Jupyter notebooks or Spyder

need to close session manually

Resetting a graph

When working in Jupyter notebooks or Spyder, you often rerun commands.

When nodes are created, they're added to the default graph automatically.

To avoid graphs with duplicate nodes, reset the graph:

```
tf.reset_default_graph()
```

Our text explains how to manage multiple graphs.

Evaluating a node

```
w = tf.constant(3)
x = w + 2
y = x + 5
z = x * 3

with tf.Session() as sess:
    print(y.eval())  # 10
    print(z.eval())  # 15
```

When z.eval() is run:

- the graph is examined for dependencies
- here, z depends on x, and x depends on w
- so w is evaluated, then x, then z

Across graph runs:

- node values are dropped
- except variable values

Linear Regression with TensorFlow

```
import numpy as np
from sklearn.datasets import fetch_california_h

housing = fetch_california_housing()
m, n = housing.data.shape
housing_data_plus_bias = np.c_[np.ones((m, 1)), housing.data]

X = tf.constant(housing_data_plus_bias, dtype=t
y = tf.constant(housing.target.reshape(-1, 1), dtype=tf.float32,
name="y")
XT = tf.transpose(X)
theta = tf.matmul(tf.matmul(tf.matrix_inverse(tf.matmul(XT, X)), XT), y)

with tf.Session() as sess:
    theta_value = theta.eval()
```

Where are the graph constr. and execution phases?

Is tf.transpose a TF or a scipy operation?

Where does variable X get its TF name?

We are computing the normal equation: this is not deep learning

Summary

- TensorFlow is Google's general-purpose library for computing on graphs
- It is targeted at deep learning
- A graph can be run across multiple CPUs or GPUs
- In TensorFlow programs:
  - you first construct a graph
  - you then execute the graph

Bonus content: what is a tensor?

- A tensor is an n-dimensional array – it generalizes vectors and matrices
  - scalar = 0-dimensional array
  - vector = 1-dimensional array
  - matrix = 2-dimensional array
- In TensorFlow the number of dimensions is called the 'rank'
- The number of elements in each dimension is the 'shape'

See tensorflow.org/programmers_guide/tensors