Tensorflow basic

What is Tensorflow?

- Software framework for numerical computations based on dataflow graphs
- An open -source software library for numerical computation using
- data flow graphs

- Quite suitable for implementing machine learning algorithms, especially, **deep learning** algorithms.

- Executable across a variety of environments and hardware platforms
  - Android, iOS, Raspberry Pi, ..
  - Linux, macOS, Windows, …

TensorFlow Hello World!

Hello TensorFlow!

In [2]:
```
# Create a constant op
# This op is added as a node to the default graph
hello = tf.constant("Hello, TensorFlow!")

# seart a TF session
sess = tf.Session()

# run the op and get result
print(sess.run(hello))
```

```
b'Hello, TensorFlow!'
```

b'String' '**b**' indicates *Bytes literals*. http://stackoverflow.com/questions/6269765/

Hello world – Our first tensorflow example

- Import Tensorflow package
  - Import tensorflow as tf
- Constant variables and Operations
  - H = tf.constant(‘hello’)
  - W = tf.constant(‘ world’)
  - HW = tf.add(H, W) or = H + W
- Execution
  - Session
  - Session.run()

Our first tensorflow program: Hello world

In [4]:
```
import tensorflow as tf
h = tf.constant("Hello")
w = tf.constant(" world")
hw = h + w
```

In [5]:
```
with tf.Session() as sess:
    ans = sess.run(hw)

print(ans)
```

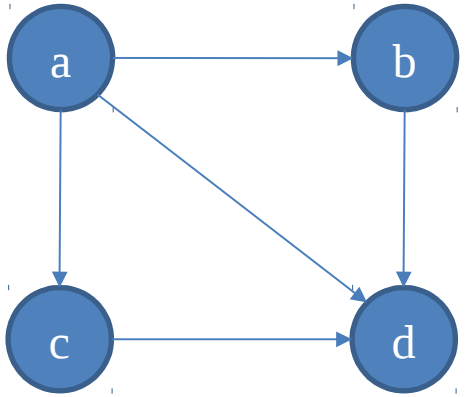
```
b'Hello world'
```

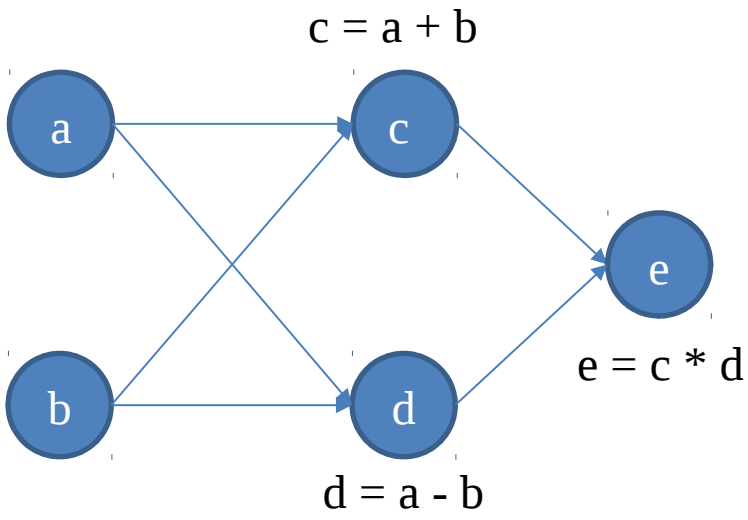
Understanding Tensorflow Basics

- Graphs
  - Nodes
  - Edges

- Computation Graphs
  - Operations = Nodes
  - Tensors = Edges

- The benefits of graph computation
  - Dependency analysis
  - Redundant elimination





Understanding Tensorflow Basics

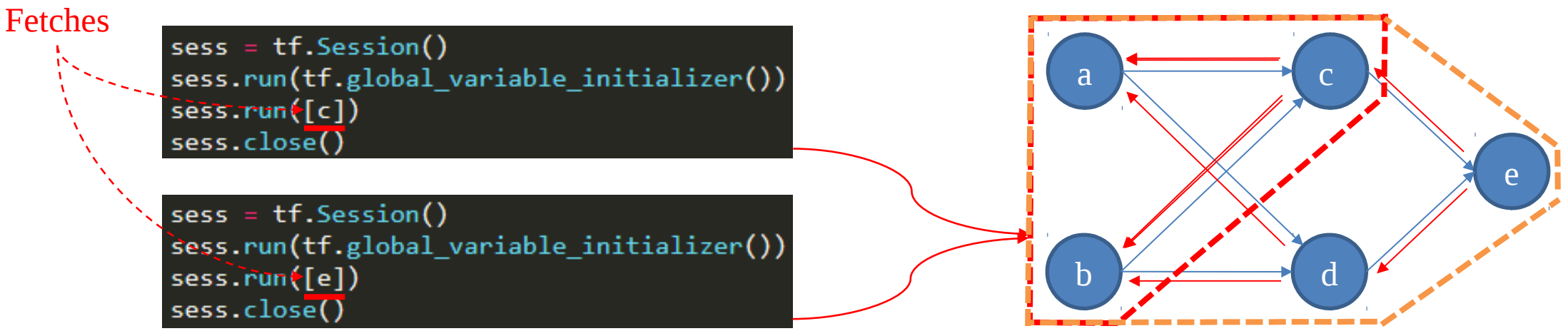
- Tensorflow workflow:
  1) constructing computation graph
  2) executing the graph

- How to create computation graph
  1) Right after we import tensorflow, a specific, empty graph is formed or
  2) g = tf.Graph() creates a new graph

- How to add nodes to graphs
  - All nodes created are automatically added to the default graph or
  - Use ‘g.as_default():’ to add nodes to new graphs

Understanding Tensorflow Basics

- However, until executing graphs, operations do not produce values
- How to execute the computation graphs
  - Sessions – execute graphs or part of graphs, resource managements
  - Fetches – Operations to execute
- From the fetches, tensorflow finds dependencies among nodes and specifies graph or part of graph to execute.



Understanding Tensorflow Basics

- Tensor
  - An object used in the Python API as a handle for the result of an operation in the graph.
  - Mathematical term for *n*-dimensi
    - [1 x 1] tensor = scalar
    - [1 x *n*] tensor = array
    - [*m* x *n*] tensor = matrix
    - [*m* x *n* x *k*] tensor = 3-dimensional a
    - …
  - Tensorflow automatically infer the shape of the data

Tensors

```
In [3]: 3 # a rank 0 tensor; this is a scalar with shape []
        [1. ,2., 3.] # a rank 1 tensor; this is a vector with shape [3]
        [[1., 2., 3.], [4., 5., 6.]] # a rank 2 tensor; a matrix with shape [2, 3]
        [[[1., 2., 3.]], [[7., 8., 9.]]] # a rank 3 tensor with shape [2, 1, 3]

Out[3]: [[[1.0, 2.0, 3.0]], [[7.0, 8.0, 9.0]]]
```

Tensor Ranks, Shapes, and Types

```
t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Rank	Math entity	Python example
0	Scalar (magnitude only)	`s = 483`
1	Vector (magnitude and direction)	`v = [1.1, 2.2, 3.3]`
2	Matrix (table of numbers)	`m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
3	3-Tensor (cube of numbers)	`t = [[[2], [4], [6]], [[8], [10], [12]], [[14], [16], [18]]]`
n	n-Tensor (you get the idea)	`....`

https://www.tensorflow.org/programmers_guide/dims_types

Tensor Ranks, Shapes, and Types

```
t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Rank	Shape	Dimension number	Example
0	[]	0-D	A 0-D tensor. A scalar.
1	[D0]	1-D	A 1-D tensor with shape [5].
2	[D0, D1]	2-D	A 2-D tensor with shape [3, 4].
3	[D0, D1, D2]	3-D	A 3-D tensor with shape [1, 4, 3].
n	[D0, D1, ... Dn-1]	n-D	A tensor with shape [D0, D1, ... Dn-1].

https://www.tensorflow.org/programmers_guide/dims_types

Tensor Ranks, Shapes, and Types

```
t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

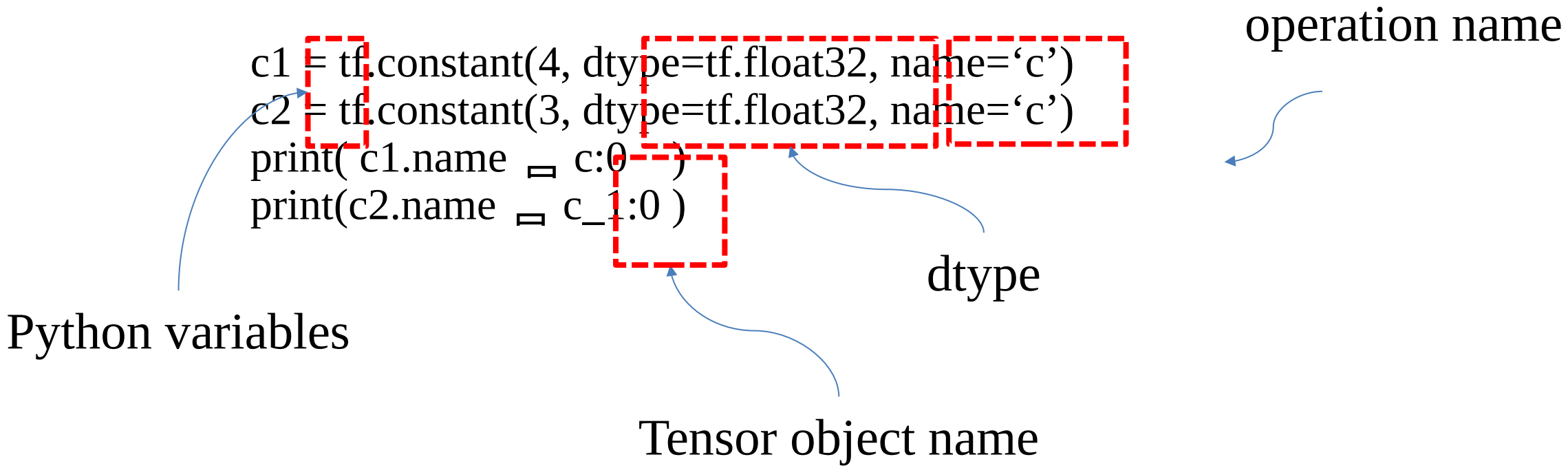
Data type	Python type	Description
DT_FLOAT	**tf.float32**	32 bits floating point.
DT_DOUBLE	**tf.float64**	64 bits floating point.
DT_INT8	**tf.int8**	8 bits signed integer.
DT_INT16	**tf.int16**	16 bits signed integer.
DT_INT32	**tf.int32**	32 bits signed integer.
DT_INT64	**tf.int64**	64 bits signed integer.

...

https://www.quora.com/When-should-I-use-tf-float32-vs-tf-float64-in-TensorFlow

Understanding Tensorflow Basics

- Name of tensor objects, data type



The name of tensor object is simply corresponding operation name concatenated with a colon, followed by the index of that tensor in the outputs of the operation that produce it.

Understanding Tensorflow Basics

- Name scope
  - Hierarchical node grouping for dealing with a large, complicate graph
  - with tf.name_scope(“scope_name”):

- Variable
  - tf.Variable(initializer = …, dtype = …, name = …, …)
  - Used for parameters such as weights, biases which are tuned for optimization
  - Should be accommodated with tf.global_variables_initializer(), which allocates the memory and sets initial values.

- Placeholders
  - Empty variables that will be filled in execution time
  - Filled with sess.run(op, feed_dict={X:x_data, Y:y_data})

Understanding Tensorflow Basics

```
import numpy as np
import tensorflow as tf

x_data = np.random.randn(5, 10)
w_data = np.random.randn(10, 1)


with tf.Graph().as_default():
    x = tf.placeholder(dtype=tf.float32, shape=(5, 10), name='x')
    w = tf.placeholder(dtype=tf.float32, shape=(10, 1), name='w')
    b = tf.fill(1., shape=(5, 1))
    xwb = tf.matmul(x, w) + b
    s = tf.reduce_max(xwb)

    with tf.Session() as sess:
        outs = sess.run(s, feed_dict={x:x_data, w:w_data})

print ("out = {}".format(outs))


-----------------------------------------------------------------
out = 3.06512
-----------------------------------------------------------------
```