Introduction to TensorFlow

Getting Started

郭耀仁

大綱

- 張量
- 運算
- TensorBoard
- 隨堂練習

張量（tensor）

- 純量（scalar）：零維張量
- 向量（vector）：一維張量
- 矩陣（matrix）：二維張量
- 張量（tensor）：n 維張量，n >= 3

In [1]:

```
import numpy as np
import tensorflow as tf

tf_scalar = tf.constant(87)
tf_vector = tf.constant([8, 7])
tf_matrix = tf.constant([
    [8, 7],
    [7, 8]
])
arr = np.arange(24).reshape(2, 3, 4)
tf_tensor = tf.constant(arr)

print(tf_scalar)
print(tf_vector)
print(tf_matrix)
print(tf_tensor)
```

```
Tensor("Const:0", shape=(), dtype=int32)
Tensor("Const_1:0", shape=(2,), dtype=int32)
Tensor("Const_2:0", shape=(2, 2), dtype=int32)
Tensor("Const_3:0", shape=(2, 3, 4), dtype=int64)
```

為什麼看不到這些張量的內容?

必須建立一個 `tf.Session()` 並且讓這些張量在其中執行才可以

In [2]:
```
with tf.Session() as sess:
    print(sess.run(tf_scalar))
    print(sess.run(tf_vector))
    print(sess.run(tf_matrix))
    print(sess.run(tf_tensor))
```

```
87
[8 7]
[[8 7]
 [7 8]]
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
```

基礎運算

加法

In [3]:
```
x = tf.constant(8, name="x")
y = tf.constant(7, name="y")
print(x)
print(y)
tf_add = tf.add(x, y)
print(tf_add)
```

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

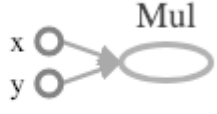
In [4]:
```
# 建立一個 Session 獲得解答
with tf.Session() as sess:
    ans = sess.run(tf_add)
    print(ans)
```

```
15
```

乗法

In [5]:
```
tf_multiply = tf.multiply(x, y)
with tf.Session() as sess:
    ans = sess.run(tf_multiply)
    print(ans)
```

```
56
```



減法

In [6]:
```
tf_subtract = tf.subtract(x, y)
with tf.Session() as sess:
    ans = sess.run(tf_subtract)
    print(ans)
```

```
1
```

除法

In [7]:
```
# 用 with 建立 Session
tf_div = tf.math.divide(x, y)
with tf.Session() as sess:
    ans = sess.run(tf_div)
    print(ans)
```

```
1.1428571428571428
```

TensorFlow 的運算方法不是唯一解，簡單點可以使用 Python 3 的運算符號

In [8]:

```
x = tf.constant(11, name="x")
y = tf.constant(6, name="y")

tf_add = x + y
tf_multiply = x * y
tf_subtract = x - y
tf_div = x / y
tf_pow = x**2
tf_mod = x % y
tf_ans = x // y
tf_ops = [tf_add, tf_multiply, tf_subtract, tf_div, tf_pow, tf_mod, tf_ans]

with tf.Session() as sess:
    for tf_op in tf_ops:
        ans = sess.run(tf_op)
        print(ans)
```

```
17
66
5
1.8333333333333333
121
5
1
```

tf.reduce_sum()

In [3]:
```
import numpy as np
import tensorflow as tf

arr = np.arange(24).reshape(2, 3, 4)
tf_tensor = tf.constant(arr)

with tf.Session() as sess:
    print(sess.run(tf_tensor))
```

```
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
```

In [4]:

```
reduced_sum_tensor = tf.reduce_sum(tf_tensor)

with tf.Session() as sess:
    ans = sess.run(reduced_sum_tensor)
    print(ans)
```

```
276
```

In [5]:
```
reduced_sum_tensor = tf.reduce_sum(tf_tensor, 0)

with tf.Session() as sess:
    ans = sess.run(reduced_sum_tensor)
    print(ans)
```

```
[[12 14 16 18]
 [20 22 24 26]
 [28 30 32 34]]
```

In [6]:

```
reduced_sum_tensor = tf.reduce_sum(tf_tensor, 1)

with tf.Session() as sess:
    ans = sess.run(reduced_sum_tensor)
    print(ans)
```

```
[[12 15 18 21]
 [48 51 54 57]]
```

In [7]:

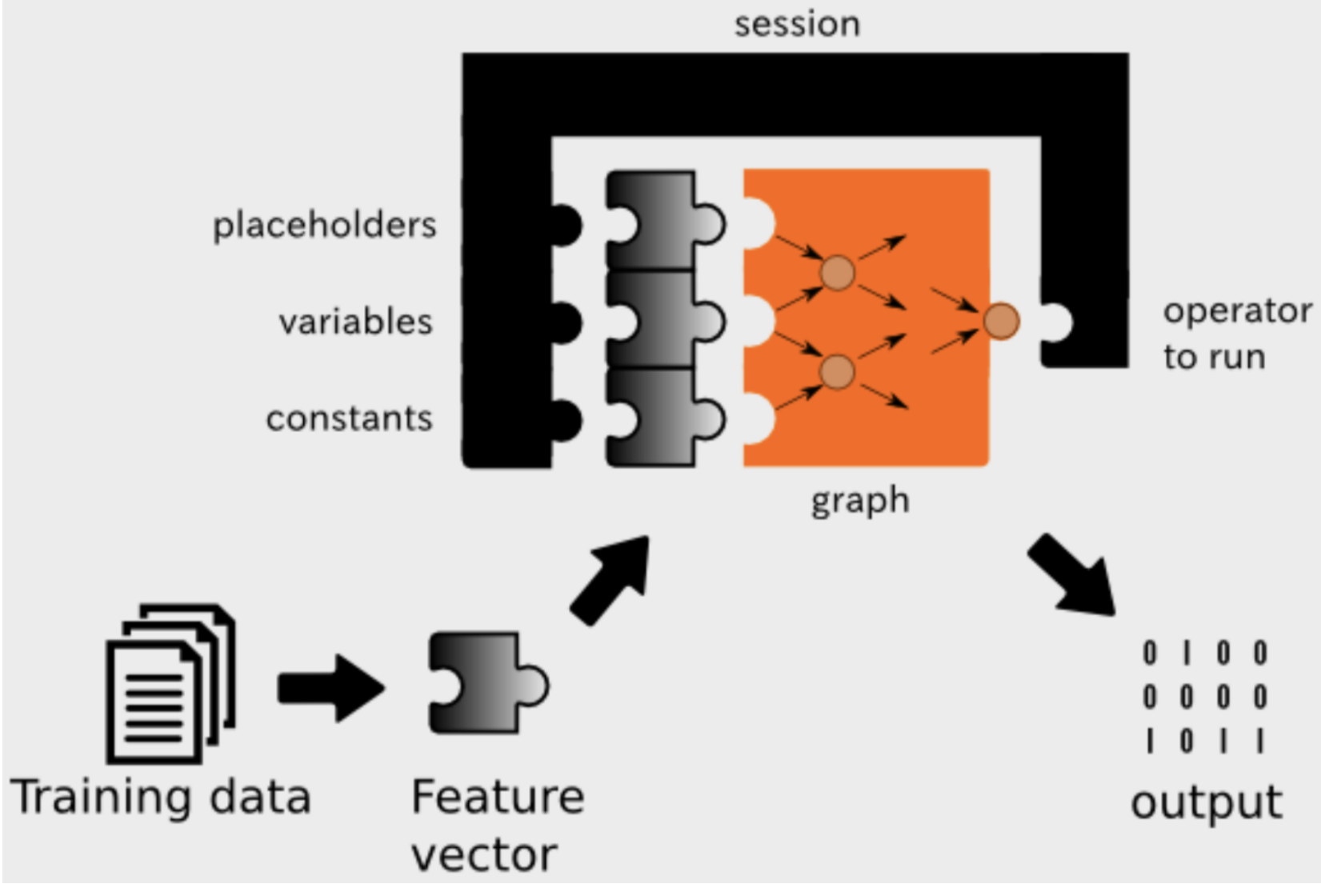
```
reduced_sum_tensor = tf.reduce_sum(tf_tensor, 2)

with tf.Session() as sess:
    ans = sess.run(reduced_sum_tensor)
    print(ans)
```

```
[[ 6 22 38]
 [54 70 86]]
```

Session 是 TensorFlow 的重要觀念

- 運算必須在 Session 中執行
- Session 接受三種類型：
    - Constant(We've used so far)
    - Placeholder(TBD)
    - Variable(TBD)

更具象化的 Session



如果您很不喜歡 session

Well...you are not alone.

> *TensorFlow 2.0 Alpha is available. Eager execution by default.*

Eager execution 是什麼?

- 直接返回具體值
- 無需建構 graph 稍後執行

該如何使用 eager execution?

- 升級到 TensorFlow 2.0 Alpha
- 在程式碼的最上方呼叫 `tf.enable_eager_execution()`

升級到 TensorFlow 2.0 Alpha

https://colab.research.google.com/drive/1sqpF7lLugnhalWGjSjtWI0iwztkZCVa0
(https://colab.research.google.com/drive/1sqpF7lLugnhalWGjSjtWI0iwztkZCVa0)

在程式碼的最上方呼叫 `tf.enable_eager_execution()`

https://colab.research.google.com/drive/1KnWI9f4DnVNgu5t6neJZWPFbzfaGuzP-
(https://colab.research.google.com/drive/1KnWI9f4DnVNgu5t6neJZWPFbzfaGuzP-)

Eager execution 與 Session 的使用目前只能二選一

TensorBoard

如何將 Graph 視覺化

建立一個叫做 graphs 的資料夾

```
# run in command line
mkdir graphs
```

撰寫一個相加的運算

In [9]:
```
import tensorflow as tf

x = tf.constant(8, name="x")
y = tf.constant(7, name="y")
tf_add = tf.add(x, y)

with tf.Session() as sess:
    writer = tf.summary.FileWriter('./graphs/tf_add', sess.graph)
    print(sess.run(tf_add))
writer.close()
```

```
15
```

啟動 TensorBoard

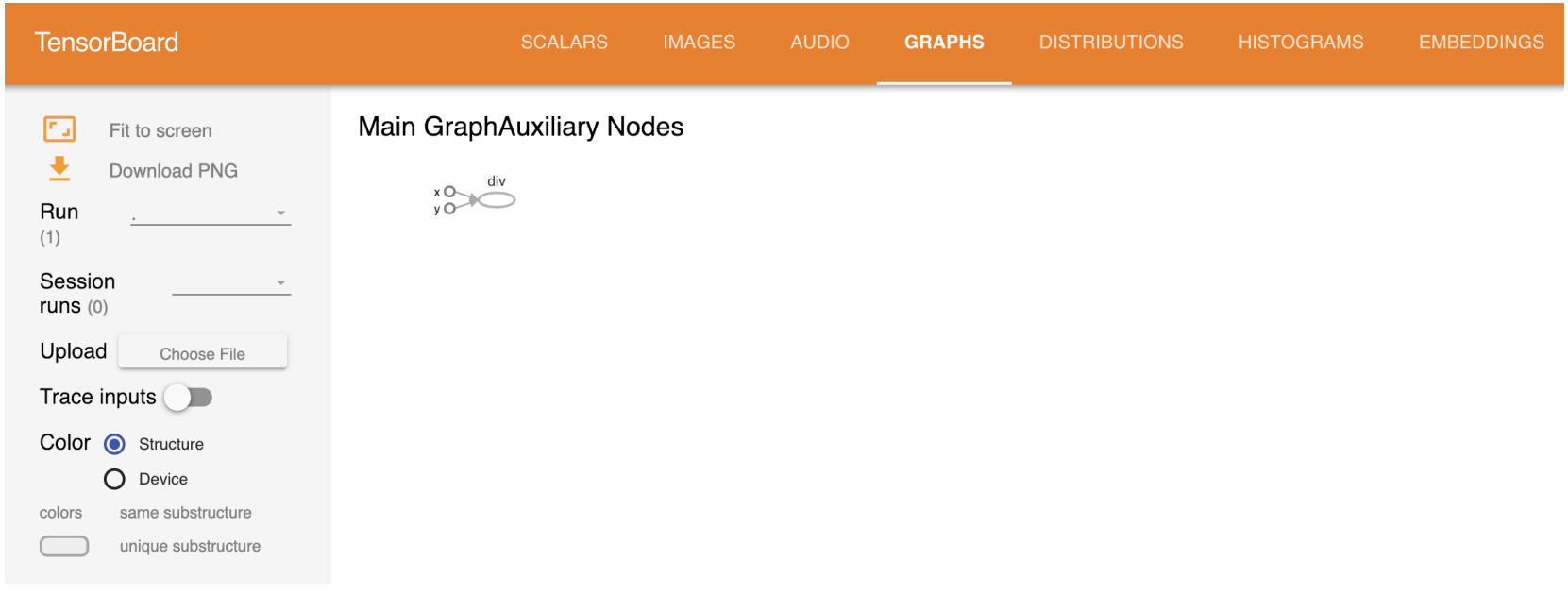
```
# run in command line
tensorboard --logdir=./graphs/tf_add --host=127.0.0.1
```

```
## Starting TensorBoard b'41' on port 6006
## (You can navigate to http://127.0.0.1:6006)
```

打開 TensorBoard

- 打開瀏覽器前往 http://127.0.0.1:6006 (http://127.0.0.1:6006)
- 點選 GRAPHS 頁籤

We are all set!



TensorBoard 可以在筆記本中使用

Using TensorBoard in Notebooks
(https://www.tensorflow.org/tensorboard/r2/tensorboard_in_notebooks)

前置作業

- `!pip install tensorflow==2.0.0-alpha0`
- `!pip install -q tf-nightly-2.0-preview`
- `%load_ext tensorboard.notebook`

用官方的 Google Colab 試試看

https://colab.research.google.com/drive/1Te8B-Fj3XbhmpGY4pcj4FVPnOwYbb-6g
(https://colab.research.google.com/drive/1Te8B-Fj3XbhmpGY4pcj4FVPnOwYbb-6g)

However, 2.0-alpha is still buggy...

In [10]:
```
# e.g. We just want to try our "hello world" script...
import tensorflow as tf

hello = tf.constant('Hello, TensorFlow!')
with tf.Session() as sess:
    print(sess.run(hello))
```

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

隨堂練習

使用 TensorFlow 實作攝氏溫度轉換華氏溫度的計算

- 將現在台北市的攝氏溫度轉換為華氏溫度
- 啟動 TensorBoard 服務觀察這個運算

$$F = \frac{9}{5}C + 32$$