Introduction to TensorFlow

Constants, variables, and placeholders

郭耀仁

大綱

- 常數
- 變數
- Placeholders

常數 Constants

如何宣告張量為常數

- `tf.constant()`
- `tf.zeros()`
- `tf.ones()`
- `tf.fill()`
- `tf.range()`
- `tf.random_normal()`
- `tf.random_uniform()`

如何跟 NumPy 對應

NumPy	TensorFlow
`np.array()`	`tf.constant()`
`np.zeroes()`	`tf.zeros()`
`np.ones()`	`tf.ones()`
`np.full()`	`tf.fill()`
`np.arange()`	`tf.range()`
`np.random.normal()`	`tf.random_normal()`
`np.random.uniform()`	`tf.random_uniform()`

In [1]:

```
# np.array() vs. tf.constant()
import numpy as np
import tensorflow as tf

const_tensor = tf.constant(24)
print(np.array(24))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
24
24
```

In [2]:

```
# np.zeros() vs. tf.zeros()
import numpy as np
import tensorflow as tf

const_tensor = tf.zeros(24)
print(np.zeros(24))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

In [3]:

```
# np.ones() vs. tf.ones()
import numpy as np
import tensorflow as tf

const_tensor = tf.ones(24)
print(np.ones(24))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

In [4]:

```
# np.full() vs. tf.fill()
import numpy as np
import tensorflow as tf

const_tensor = tf.fill((24,), 24)
print(np.full(24, (24,)))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24]
[24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24]
```

In [5]:

```
# np.arange() vs. tf.range()
import numpy as np
import tensorflow as tf

const_tensor = tf.range(24)
print(np.arange(24))
with tf.Session() as sess:
  print(sess.run(const_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]
[ 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 [6]:

```
# np.random.normal() vs. tf.random_normal()
import numpy as np
import tensorflow as tf

const_tensor = tf.random_normal((24,))
print(np.random.normal(size=24))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[-1.1467378   0.83455172 -1.53485829  1.45471805 -1.19605712 -0.37423164
 -1.54544204 -0.47310866 -0.90608266 -0.32665465 -0.42425325  1.62287938
 -0.38121458 -0.14374681 -0.70762663  2.16814083 -0.86912318  1.15250448
  0.61403438 -1.42487763  0.77637587 -0.74018035  1.51315797  1.27656247]
[ 0.62538743 -1.0750122  -2.6668274  -0.5583715   0.11181478 -0.96974725
 -0.7624392   1.0945379   0.40959305  2.443584    0.2784222  -0.20217635
  1.8892621  -0.32116467 -0.12829572  0.664623    0.57437265 -0.38665575
 -0.8686913  -1.9773843   1.2281301  -1.6643158   0.5441105   0.41093013]
```

In [7]:

```
# np.random.uniform() vs. tf.random_uniform()
import numpy as np
import tensorflow as tf

const_tensor = tf.random_uniform((24,))
print(np.random.uniform(size=24))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[0.81158876 0.45773529 0.56171791 0.79739982 0.89199952 0.81644822
 0.02011246 0.02823636 0.05446537 0.00279311 0.36008668 0.9222017
 0.26422074 0.59165804 0.26407344 0.16142114 0.55624794 0.50406805
 0.03192423 0.8712371  0.45221964 0.74415067 0.47068362 0.11900758]
[0.43590772 0.584324   0.90497625 0.42483795 0.64262104 0.47081673
 0.72082984 0.0970856  0.59833574 0.06022489 0.92139125 0.8407004
 0.48723388 0.58953977 0.72041273 0.7179949  0.09972143 0.380167
 0.3185699  0.97219765 0.6379056  0.02075279 0.2693652  0.14287198]
```

如何使用 TensorFlow 處理矩陣

- `tf.reshape()`
- `tf.eye()`
- `tf.diag()`
- `tf.matrix_transpose()`
- `tf.matmul()`

如何跟 NumPy 對應

NumPy	TensorFlow
`arr.reshape()`	`tf.reshape()`
`np.eye()`	`tf.eye()`
`np.diag()`	`tf.diag()`
`np.transpose()`	`tf.matrix_transpose()`
`np.dot()`	`tf.matmul()`

In [8]:

```
# arr.reshape() vs. tf.reshape()
import numpy as np
import tensorflow as tf

const_tensor = tf.reshape(tf.range(24), (6, 4))
print(np.arange(24).reshape(6, 4))
with tf.Session() as sess:
  print(sess.run(const_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]]
[[ 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 [9]:

```
# np.eye() vs. tf.eye()
import numpy as np
import tensorflow as tf

const_tensor = tf.eye(3)
print(np.eye(3))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

In [10]:

```python
# np.diag() vs. tf.diag()
import numpy as np
import tensorflow as tf

const_tensor = tf.diag([1, 2, 3])
print(np.diag([1, 2, 3]))
with tf.Session() as sess:
  print(sess.run(const_tensor))
```

```
[[1 0 0]
 [0 2 0]
 [0 0 3]]
[[1 0 0]
 [0 2 0]
 [0 0 3]]
```

In [11]:
```
# np.transpose() vs. tf.matrix_transpose()
import numpy as np
import tensorflow as tf

const_tensor = tf.ones((2, 4))
const_tensor_t = tf.matrix_transpose(const_tensor)
print(np.ones((2, 4)))
print(np.ones((2, 4)).T)
print("\n")
with tf.Session() as sess:
  print(sess.run(const_tensor))
  print(sess.run(const_tensor_t))
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]]
[[1. 1.]
 [1. 1.]
 [1. 1.]
 [1. 1.]]


[[1. 1. 1. 1.]

 [1. 1. 1. 1.]]
[[1. 1.]
 [1. 1.]
 [1. 1.]
 [1. 1.]]
```

In [12]:

```
# np.dot() vs. tf.matmul()
import numpy as np
import tensorflow as tf

const_tensor = tf.ones((2, 2))
matrix_multiply = tf.matmul(const_tensor, const_tensor)
print(np.dot(np.ones((2, 2)), np.ones((2, 2))))
with tf.Session() as sess:
  print(sess.run(matrix_multiply))
```

```
[[2. 2.]
 [2. 2.]]
[[2. 2.]
 [2. 2.]]
```

隨堂練習

分別使用 NumPy 與 TensorFlow 常數張量計算 $u^T v$

$$u = \begin{bmatrix} 4 \\ -4 \\ -3 \end{bmatrix}$$

$$v = \begin{bmatrix} 4 \\ 2 \\ 4 \end{bmatrix}$$

隨堂練習

分別使用 NumPy 與 TensorFlow 常數張量計算 AB 與 BA

$$A = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

隨堂練習

分別使用 NumPy 與 TensorFlow 常數張量計算 AI 與 IA

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

熟悉 NumPy 的 Python 使用者來說學習 TensorFlow 有很大優勢

變數 Variables

以 tf.Variable() 宣告

In [13]:
```
import tensorflow as tf

lucky_number = tf.Variable(24)
print(lucky_number)
```

```
<tf.Variable 'Variable:0' shape=() dtype=int32_ref>
```

宣告變數張量不如常數張量那麼單純

- 宣告變數張量的初始值、類型與外觀
- 初始化變數張量

如果宣告的變數張量沒有經過初始化，將會得到錯誤

In [14]:
```
import tensorflow as tf

lucky_number = tf.Variable(24)
with tf.Session() as sess:
  print(sess.run(lucky_number))
```

```
---------------------------------------------------------------------------
FailedPreconditionError                   Traceback (most recent call last)
<ipython-input-14-fc3367f69152> in <module>()
      3 lucky_number = tf.Variable(24)
      4 with tf.Session() as sess:
----> 5   print(sess.run(lucky_number))

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/client/session.pyc in run(self, fetches, feed_dict, options, run_me
tadata)
    927     try:
    928       result = self._run(None, fetches, feed_dict, options_ptr,
--> 929                          run_metadata_ptr)
    930       if run_metadata:
    931         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/client/session.pyc in _run(self, handle, fetches, feed_dict, option
s, run_metadata)
   1150     if final_fetches or final_targets or (handle and feed_dict_tensor)
:
   1151       results = self._do_run(handle, final_targets, final_fetches,
-> 1152                              feed_dict_tensor, options, run_metadata)
   1153     else:
   1154       results = []

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/client/session.pyc in _do_run(self, handle, target_list, fetch_lis
t, feed_dict, options, run_metadata)
   1326     if handle is None:
```

```
   1327       return self._do_call(_run_fn, feeds, fetches, targets, options,
-> 1328                            run_metadata)
   1329     else:
   1330       return self._do_call(_prun_fn, handle, feeds, fetches)

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/client/session.pyc in _do_call(self, fn, *args)
   1346           pass
   1347       message = error_interpolation.interpolate(message, self._graph)
-> 1348       raise type(e)(node_def, op, message)
   1349
   1350   def _extend_graph(self):

FailedPreconditionError: Attempting to use uninitialized value Variable_1
         [[{node _retval_Variable_1_0_0}} = _Retval[T=DT_INT32, index=0, _dev
ice="/job:localhost/replica:0/task:0/device:CPU:0"](Variable_1)]]
```

該如何修正呢?

將變數張量的 `initializer` 屬性放入 Session 中執行

In [15]:
```
import tensorflow as tf

lucky_number = tf.Variable(24)
with tf.Session() as sess:
  sess.run(lucky_number.initializer)
  print(sess.run(lucky_number))
```

```
24
```

透過 .assign() 賦值

In [16]:
```
import tensorflow as tf

lucky_number = tf.Variable(24)
assign_op = lucky_number.assign(7)
with tf.Session() as sess:
  sess.run(lucky_number.initializer)
  # sess.run(assign_op)
  print(sess.run(lucky_number))
```

```
24
```

重新賦值也是一種運算（operation），必須放入 Session 執行

In [17]:

```
import tensorflow as tf

lucky_number = tf.Variable(24)
assign_op = lucky_number.assign(7)
with tf.Session() as sess:
  sess.run(lucky_number.initializer)
  sess.run(assign_op)
  print(sess.run(lucky_number))
```

```
7
```

變數張量被宣告之後，重新賦值時必須要注意類型

In [18]:
```
import tensorflow as tf

lucky_number = tf.Variable(24)
assign_op = lucky_number.assign(7.0)
with tf.Session() as sess:
  sess.run(lucky_number.initializer)
  sess.run(assign_op)
  print(sess.run(lucky_number))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-18-712e3684872e> in <module>()
      2
      3 lucky_number = tf.Variable(24)
----> 4 assign_op = lucky_number.assign(7.0)
      5 with tf.Session() as sess:
      6   sess.run(lucky_number.initializer)

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/ops/variables.pyc in assign(self, value, use_locking, name, read_va
lue)
   1716     """
   1717     assign = state_ops.assign(self._variable, value, use_locking=use_l
ocking,
-> 1718                               name=name)
   1719     if read_value:
   1720       return assign


/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/ops/state_ops.pyc in assign(ref, value, validate_shape, use_lockin
g, name)
    219     return gen_state_ops.assign(
    220         ref, value, use_locking=use_locking, name=name,
--> 221         validate_shape=validate_shape)
    222   return ref.assign(value, name=name)
    223
```

```
/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/ops/gen_state_ops.pyc in assign(ref, value, validate_shape, use_loc
king, name)
     59     _, _, _op = _op_def_lib._apply_op_helper(
     60         "Assign", ref=ref, value=value, validate_shape=validate_shape,
---> 61         use_locking=use_locking, name=name)
     62     _result = _op.outputs[:]
     63     _inputs_flat = _op.inputs

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/framework/op_def_library.pyc in _apply_op_helper(self, op_type_nam
e, name, **keywords)
    517                     "type '%s' instead." %
    518                     (dtypes.as_dtype(dtype).name, input_arg.name, op_typ
e_name,
--> 519                      repr(values), type(values).__name__))
    520           except ValueError:
    521             # What type does convert_to_tensor think it has?

TypeError: Expected int32 passed to parameter 'value' of op 'Assign', got 7.0
 of type 'float' instead.
```

變數張量被宣告之後，重新賦值時必須要注意外觀

In [19]:
```
import tensorflow as tf

lucky_numbers = tf.Variable([7, 24])
assign_op = lucky_numbers.assign(87)
with tf.Session() as sess:
  sess.run(lucky_numbers.initializer)
  sess.run(assign_op)
  print(sess.run(lucky_numbers))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-19-894731d878c5> in <module>()
      2
      3 lucky_numbers = tf.Variable([7, 24])
----> 4 assign_op = lucky_numbers.assign(87)
      5 with tf.Session() as sess:
      6   sess.run(lucky_numbers.initializer)

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/ops/variables.pyc in assign(self, value, use_locking, name, read_va
lue)
   1716     """
   1717     assign = state_ops.assign(self._variable, value, use_locking=use_l
ocking,
-> 1718                               name=name)
   1719     if read_value:
   1720       return assign


/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/ops/state_ops.pyc in assign(ref, value, validate_shape, use_lockin
g, name)
    219     return gen_state_ops.assign(
    220         ref, value, use_locking=use_locking, name=name,
--> 221         validate_shape=validate_shape)
    222   return ref.assign(value, name=name)
    223
```

```
/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/ops/gen_state_ops.pyc in assign(ref, value, validate_shape, use_loc
king, name)
     59     _, _, _op = _op_def_lib._apply_op_helper(
     60         "Assign", ref=ref, value=value, validate_shape=validate_shape,
---> 61         use_locking=use_locking, name=name)
     62     _result = _op.outputs[:]
     63     _inputs_flat = _op.inputs

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/framework/op_def_library.pyc in _apply_op_helper(self, op_type_nam
e, name, **keywords)
    785           op = g.create_op(op_type_name, inputs, output_types, name=scop
e,
    786                            input_types=input_types, attrs=attr_protos,
--> 787                            op_def=op_def)
    788       return output_structure, op_def.is_stateful, op
    789

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/util/deprecation.pyc in new_func(*args, **kwargs)
    486                 'in a future version' if date is None else ('after %s'
% date),
    487                 instructions)
--> 488       return func(*args, **kwargs)
    489     return tf_decorator.make_decorator(func, new_func, 'deprecated',
    490                                        _add_deprecated_arg_notice_to_d
ocstring(

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/framework/ops.pyc in create_op(***failed resolving arguments***)
   3272           input_types=input_types,
   3273           original_op=self._default_original_op,
-> 3274           op_def=op_def)
   3275       self._create_op_helper(ret, compute_device=compute_device)
   3276     return ret
```

```
/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/framework/ops.pyc in __init__(self, node_def, g, inputs, output_typ
es, control_inputs, input_types, original_op, op_def)
   1790           op_def, inputs, node_def.attr)
   1791       self._c_op = _create_c_op(self._graph, node_def, grouped_inputs,
-> 1792                                 control_input_ops)
   1793
   1794     # Initialize self._outputs.

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/framework/ops.pyc in _create_c_op(graph, node_def, inputs, control_
inputs)
   1629   except errors.InvalidArgumentError as e:
   1630     # Convert to ValueError for backwards compatibility.
-> 1631     raise ValueError(str(e))
   1632
   1633   return c_op

ValueError: Shapes must be equal rank, but are 1 and 0 for 'Assign_3' (op: 'As
sign') with input shapes: [2], [].
```

TensorFlow 外觀的註記與 ndarray 相同

- 零維：`()`
- 一維：`(m,)`
- 二維：`(m, n)`
- 三維：`(q, m, n)`

In [20]:

```
import numpy as np

zero_d = np.array(24)
one_d = np.arange(24)
two_d = np.arange(24).reshape(6, 4)
three_d = np.arange(24).reshape(2, 3, 4)
print(zero_d.shape)
print(one_d.shape)
print(two_d.shape)
print(three_d.shape)
```

```
()
(24,)
(6, 4)
(2, 3, 4)
```

可以使用 .get_shape() 確認 Tensor 外觀

In [21]:
```
import tensorflow as tf

zero_d = tf.Variable(24)
one_d = tf.reshape(tf.Variable(tf.range(24)), (24,))
two_d = tf.reshape(tf.Variable(tf.range(24)), (6, 4))
three_d = tf.reshape(tf.Variable(tf.range(24)), (2, 3, 4))
print(zero_d.get_shape())
print(one_d.get_shape())
print(two_d.get_shape())
print(three_d.get_shape())
```

```
()
(24,)
(6, 4)
(2, 3, 4)
```

隨堂練習

將華氏溫度轉換為攝氏溫度的計算以 TensorFlow 的變數張量改寫

In [22]:
```
import numpy as np

city_temps = np.array([59, 30, 45, 28, 47]) # Taipei, New York, London, Reykjavik, Tokyo
city_temps = (city_temps - 32) * 5 / 9
print(city_temps)
```

```
[15 -2  7 -3  8]
```

檢查點：TensorFlow 的變數名稱不變

```
<tf.Variable 'Variable_1:0' shape=(5,) dtype=int32_ref>
[59 30 45 28 47]
[15 -2  7 -3  8]
<tf.Variable 'Variable_1:0' shape=(5,) dtype=int32_ref>
```

隨堂練習

將公里轉換為英里的計算以 TensorFlow 的變數張量改寫

In [24]:
```
import numpy as np

distances = np.array([5, 10, 21.095, 42.195]) # 5k, 10k, Half Marathon, Marathon
distances = distances / 1.609344
print(distances)
```

```
[ 3.10685596  6.21371192 13.1078253  26.21875746]
```

檢查點：TensorFlow 的變數名稱不變

```
<tf.Variable 'Variable_1:0' shape=(5,) dtype=int32_ref>
[59 30 45 28 47]
[15 -2  7 -3  8]
<tf.Variable 'Variable_1:0' shape=(5,) dtype=int32_ref>
```

Placeholders

以 tf.placeholder() 宣告

Placeholder 作為 TensorFlow 模型的資料輸入口

- Feed dict 資料要以 Python dict 的格式餵入
- Fetch 是模型運算的輸出，類型是 `ndarray`

可以把 Placeholder 想成像是 `None` 或 `np.NaN`

In [26]:
```
import numpy as np

none_list = [None for _ in range(3)]
nan_arr = [np.NaN for _ in range(3)]
print("Pythonic:")
print(len(none_list))
print(none_list)
print("NumPy:")
print(len(nan_arr))
print(nan_arr)
```

```
Pythonic:
3
[None, None, None]
NumPy:
3
[nan, nan, nan]
```

In [27]:
```
import numpy as np

none_list = [None for _ in range(3)]
nan_arr = [np.NaN for _ in range(3)]
lucky_numbers = [7, 24, 34]
for i in range(3):
  none_list[i] = lucky_numbers[i]
  nan_arr[i] = lucky_numbers[i]
print(none_list)
print(nan_arr)
```

```
[7, 24, 34]
[7, 24, 34]
```

In [28]:

```
import tensorflow as tf

tf_placeholder = tf.placeholder(tf.int32, shape=(3,))
print(tf_placeholder)
print(tf_placeholder.get_shape()[0])
```

```
Tensor("Placeholder:0", shape=(3,), dtype=int32)
3
```

將資料以 dict 餵入 placeholder

語法為:

```
import tensorflow as tf

my_ph = tf.placeholder(...)
my_op = ...
feed_dict = {
  my_ph: ...
}
with tf.Session() as sess:
  fetch = sess.run(my_op, feed_dict)
```

In [29]:
```
import tensorflow as tf

tf_placeholder = tf.placeholder(tf.int32, shape=(3,))
with tf.Session() as sess:
  fetch = sess.run(tf_placeholder, {tf_placeholder: [7, 24, 34]})

print(fetch)
print(type(fetch))
```

```
[ 7 24 34]
<type 'numpy.ndarray'>
```

Placeholders 也很嚴謹

不同的外觀不能餵入

In [30]:
```
import tensorflow as tf

tf_placeholder = tf.placeholder(dtype=tf.int32, shape=(3,))
with tf.Session() as sess:
  print(sess.run(tf_placeholder, {tf_placeholder: [7, 24]}))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-30-e4672dc2e8e2> in <module>()
      3 tf_placeholder = tf.placeholder(dtype=tf.int32, shape=(3,))
      4 with tf.Session() as sess:
----> 5   print(sess.run(tf_placeholder, {tf_placeholder: [7, 24]}))

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/client/session.pyc in run(self, fetches, feed_dict, options, run_me
tadata)
    927     try:
    928       result = self._run(None, fetches, feed_dict, options_ptr,
--> 929                          run_metadata_ptr)
    930       if run_metadata:
    931         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python2.7/site-packages/tensorf
low/python/client/session.pyc in _run(self, handle, fetches, feed_dict, option
s, run_metadata)
   1126                                 'which has shape %r' %
   1127                                 (np_val.shape, subfeed_t.name,
-> 1128                                  str(subfeed_t.get_shape())))
   1129           if not self.graph.is_feedable(subfeed_t):
   1130             raise ValueError('Tensor %s may not be fed.' % subfeed_t)

ValueError: Cannot feed value of shape (2,) for Tensor u'Placeholder_2:0', whi
ch has shape '(3,)'
```

Placeholders 也很嚴謹

外觀相同、不同的資料類型則會做隱性轉換

In [31]:
```
import tensorflow as tf

tf_placeholder = tf.placeholder(dtype=tf.int32, shape=(3,))
with tf.Session() as sess:
  print(sess.run(tf_placeholder, {tf_placeholder: [7.0, 24.0, 34.0]}))
```

```
[ 7 24 34]
```

隨堂練習

以 TensorFlow 的 placeholder 張量將公里轉換為英里。

隨堂練習

以 TensorFlow 的 placeholder 張量將華氏溫度轉換為攝氏溫度。

$$C = \frac{(F - 32) \times 5}{9}$$

In [33]:
```
city_temps_f = [59, 30, 45, 28, 47] # Taipei, New York, London, Reykjavik, Tokyo
```

隨堂練習

以 TensorFlow 的 placeholder 張量計算五個球員的 BMI

$$BMI = \frac{weight_{kg}}{height_{m}^{2}}$$

In [35]:
```
# Shaq, Dirk, LeBron, MJ, Nash
player_heights = [216, 213, 203, 198, 191]
player_weights = [147, 111, 113, 98, 82]
```