Introduction to TensorFlow

Logistic Regression with TensorFlow

郭耀仁

大綱

- 取得資料
- Benchmark
- 建構 TensorFlow 計算圖形
- 訓練
- 隨堂練習

取得資料

簡單、作為測試目的即可

Scikit-Learn Breast Cancer 資料集

In [1]:
```
from sklearn.datasets import load_breast_cancer

breast_cancer = load_breast_cancer()
print(breast_cancer.feature_names)
print(breast_cancer.DESCR)
```

```
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
```

- fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

- class:
  - WDBC-Malignant
  - WDBC-Benign

:Summary Statistics:

	Min	Max
radius (mean):	6.981	28.11
texture (mean):	9.71	39.28
perimeter (mean):	43.79	188.5
area (mean):	143.5	2501.0
smoothness (mean):	0.053	0.163
compactness (mean):	0.019	0.345
concavity (mean):	0.0	0.427
concave points (mean):	0.0	0.201
symmetry (mean):	0.106	0.304
fractal dimension (mean):	0.05	0.097
radius (standard error):	0.112	2.873
texture (standard error):	0.36	4.885
perimeter (standard error):	0.757	21.98
area (standard error):	6.802	542.2
smoothness (standard error):	0.002	0.031
compactness (standard error):	0.002	0.135
concavity (standard error):	0.0	0.396
concave points (standard error):	0.0	0.053
symmetry (standard error):	0.008	0.079
fractal dimension (standard error):	0.001	0.03
radius (worst):	7.93	36.04
texture (worst):	12.02	49.54

```
perimeter (worst):                    50.41  251.2
area (worst):                         185.2  4254.0
smoothness (worst):                   0.071  0.223
compactness (worst):                  0.027  1.058
concavity (worst):                    0.0    1.252
concave points (worst):               0.0    0.291
symmetry (worst):                     0.156  0.664
fractal dimension (worst):            0.055  0.208
====================================== ====== ======

:Missing Attribute Values: None

:Class Distribution: 212 - Malignant, 357 - Benign

:Creator:  Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

:Donor: Nick Street

:Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
https://goo.gl/U2Uwz2

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass.  They describe
characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
```

[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

.. topic:: References

- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.
- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.
- W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171.

In [2]:
```
X_arr = breast_cancer.data[:, [3]]
y_arr = breast_cancer.target
print(X_arr.shape)
print(y_arr.shape)
```

```
(569, 1)
(569,)
```

In [3]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_arr, y_arr, test_size=0.3, r
andom_state=123)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(398, 1)
(171, 1)
(398,)
(171,)
```

Benchmark

以 Scikit-Learn 的 LogisticRegression 作對照

In [4]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

clf = LogisticRegression(solver="liblinear")
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
```

In [5]:

```
print(clf.coef_)
print(clf.intercept_)
print(acc)
```

```
[[-0.00774717]]
[5.52628648]
0.8654970760233918
```

Benchmark 完整程式碼

In [6]:
```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

breast_cancer = load_breast_cancer()
X_arr = breast_cancer.data[:, [3]]
y_arr = breast_cancer.target
X_train, X_test, y_train, y_test = train_test_split(X_arr, y_arr, test_size=0.3, r
andom_state=123)
clf = LogisticRegression(solver="liblinear")
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
```

建構 TensorFlow 計算圖形

準備 Placeholders 供訓練時輸入 X_train、y_train

In [7]:
```
import tensorflow as tf

X_train_shape = X_train.shape
y_train_shape = y_train.shape
X = tf.placeholder(tf.float32, X_train_shape)
y = tf.placeholder(tf.float32, y_train_shape)
```

準備變數供訓練時尋找最適係數（Weights）與殘差項（Bias）

In [8]:
```
W_shape = (X_train_shape[1], 1)
b_shape = (1,)
W = tf.Variable(tf.random_normal(W_shape))
b = tf.Variable(tf.random_normal(b_shape))
with tf.Session() as sess:
  sess.run(W.initializer)
  sess.run(b.initializer)
  print(sess.run(W))
  print(sess.run(b))
```

```
[[1.4551902]]
[0.10155638]
```

檢查 X、W 與 b 的外觀，寫下 y_pred 的公式

In [9]:
```
print(X.shape)
print(W.shape)
print(b.shape)
y_logit = tf.squeeze(tf.matmul(X, W) + b)
y_one_prob = tf.sigmoid(y_logit) # Sigmoid
y_pred = tf.round(y_one_prob)    # Step Function
print(y_pred)
```

```
(398, 1)
(1, 1)
(1,)
Tensor("Round:0", shape=(398,), dtype=float32)
```

tf.squeeze() 做了什麼事?

In [10]:
```
a = tf.ones((398, 1), dtype=tf.int32)
b = tf.squeeze(a)
print(a.shape)
print(b.shape)
```

```
(398, 1)
(398,)
```

寫下成本函數的公式

In [11]:
```
entropy = tf.nn.sigmoid_cross_entropy_with_logits(logits=y_logit, labels=y)
loss = tf.reduce_sum(entropy)
```

宣告 Optimizer 與學習速率

In [12]:
```
# Adam as in adaptive moment estimation, 與基礎 Gradient Descent 不同的地方是 learnin
g rate 會隨著梯度最自適應（adaptive）調整
learning_rate = 0.01
opt = tf.train.AdamOptimizer(learning_rate)
optimizer = opt.minimize(loss)
```

建構 TensorFlow 計算圖形完整程式碼

In [13]:

```
import tensorflow as tf

tf.reset_default_graph()
X_train_shape = X_train.shape
y_train_shape = y_train.shape
W_shape = (X_train_shape[1], 1)
b_shape = (1,)
learning_rate = 0.001

# placeholders
with tf.name_scope("placeholders"):
  X = tf.placeholder(tf.float32, X_train_shape)
  y = tf.placeholder(tf.float32, y_train_shape)
# weights
with tf.name_scope("weights"):
  W = tf.Variable(tf.random_normal(W_shape))
  b = tf.Variable(tf.random_normal(b_shape))
# prediction
with tf.name_scope("prediction"):
  y_logit = tf.squeeze(tf.matmul(X, W) + b)
  y_one_prob = tf.sigmoid(y_logit) # Sigmoid
  y_pred = tf.round(y_one_prob)    # Step Function
# loss
with tf.name_scope("loss"):
  entropy = tf.nn.sigmoid_cross_entropy_with_logits(logits=y_logit, labels=y)
  loss = tf.reduce_sum(entropy)
# optimizer
with tf.name_scope("optimizer"):
  opt = tf.train.AdamOptimizer(learning_rate)
  optimizer = opt.minimize(loss)
```

訓練

In [14]:

```
n_steps = 1000
file_writer_path = "./graphs/logistic-regression"

with tf.Session() as sess:
  sess.run(W.initializer)
  sess.run(b.initializer)
  train_writer = tf.summary.FileWriter(file_writer_path, tf.get_default_graph())
  for i in range(n_steps):
    feed_dict = {
      X: X_train,
      y: y_train
    }
    _, loss_ = sess.run([optimizer, loss], feed_dict=feed_dict)
    if i % 100 == 0:
      print("step {}, loss: {}".format(i, loss_))
  w_final, b_final = sess.run([W, b])
```

```
---------------------------------------------------------------------------
FailedPreconditionError                   Traceback (most recent call last)
~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _do_call(self, fn, *args)
   1333     try:
-> 1334       return fn(*args)
   1335     except errors.OpError as e:


~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _run_fn(feed_dict, fetch_list, target_list, options, run_meta
data)
   1318       return self._call_tf_sessionrun(
-> 1319           options, feed_dict, fetch_list, target_list, run_metadata)
   1320


~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _call_tf_sessionrun(self, options, feed_dict, fetch_list, tar
get_list, run_metadata)
   1406         self._session, options, feed_dict, fetch_list, target_list,
```

```
-> 1407         run_metadata)
   1408
```

```
FailedPreconditionError: Attempting to use uninitialized value optimizer/beta1
_power
         [[{{node optimizer/beta1_power/read}} = Identity[T=DT_FLOAT, _class=
["loc:@optimizer/Adam/Assign_1"], _device="/job:localhost/replica:0/task:0/dev
ice:CPU:0"](optimizer/beta1_power)]]

During handling of the above exception, another exception occurred:
```

```
FailedPreconditionError                   Traceback (most recent call last)
<ipython-input-14-67a595c9083a> in <module>
     11       y: y_train
     12     }
---> 13     _, loss_ = sess.run([optimizer, loss], feed_dict=feed_dict)
     14     if i % 100 == 0:
     15       print("step {}, loss: {}".format(i, loss_))

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in run(self, fetches, feed_dict, options, run_metadata)
    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)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _run(self, handle, fetches, feed_dict, options, 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 = []

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py in _do_run(self, handle, target_list, fetch_list, feed_dict, opt
```

```
ions, 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)

~/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/clie
nt/session.py 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 optimizer/beta1
_power
         [[node optimizer/beta1_power/read (defined at <ipython-input-13-9b730
68dda26>:30)  = Identity[T=DT_FLOAT, _class=["loc:@optimizer/Adam/Assign_1"],
 _device="/job:localhost/replica:0/task:0/device:CPU:0"](optimizer/beta1_powe
r)]]

Caused by op 'optimizer/beta1_power/read', defined at:
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/runpy.py", li
ne 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/runpy.py", li
ne 85, in _run_code
    exec(code, run_globals)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel_launcher.py", line 16, in <module>
    app.launch_new_instance()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/traitlets/config/application.py", line 658, in launch_instance
    app.start()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/kernelapp.py", line 505, in start
    self.io_loop.start()
```

```
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/platform/asyncio.py", line 132, in start
    self.asyncio_loop.run_forever()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/asyncio/base_
events.py", line 438, in run_forever
    self._run_once()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/asyncio/base_
events.py", line 1451, in _run_once
    handle._run()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/asyncio/event
s.py", line 145, in _run
    self._callback(*self._args)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/ioloop.py", line 758, in _run_callback
    ret = callback()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/stack_context.py", line 300, in null_wrapper
    return fn(*args, **kwargs)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 1233, in inner
    self.run()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 1147, in run
    yielded = self.gen.send(value)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/kernelbase.py", line 370, in dispatch_queue
    yield self.process_one()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 346, in wrapper
    runner = Runner(result, future, yielded)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 1080, in __init__
    self.run()
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 1147, in run
    yielded = self.gen.send(value)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/kernelbase.py", line 357, in process_one
```

```
    yield gen.maybe_future(dispatch(*args))
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 326, in wrapper
    yielded = next(result)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/kernelbase.py", line 267, in dispatch_shell
    yield gen.maybe_future(handler(stream, idents, msg))
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 326, in wrapper
    yielded = next(result)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/kernelbase.py", line 534, in execute_request
    user_expressions, allow_stdin,
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tornado/gen.py", line 326, in wrapper
    yielded = next(result)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/ipkernel.py", line 294, in do_execute
    res = shell.run_cell(code, store_history=store_history, silent=silent)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/ipykernel/zmqshell.py", line 536, in run_cell
    return super(ZMQInteractiveShell, self).run_cell(*args, **kwargs)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/IPython/core/interactiveshell.py", line 2819, in run_cell
    raw_cell, store_history, silent, shell_futures)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/IPython/core/interactiveshell.py", line 2845, in _run_cell
    return runner(coro)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/IPython/core/async_helpers.py", line 67, in _pseudo_sync_runner
    coro.send(None)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/IPython/core/interactiveshell.py", line 3020, in run_cell_async
    interactivity=interactivity, compiler=compiler, result=result)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/IPython/core/interactiveshell.py", line 3185, in run_ast_nodes
    if (yield from self.run_code(code, result)):
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
```

```
s/IPython/core/interactiveshell.py", line 3267, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "<ipython-input-13-9b73068dda26>", line 30, in <module>
    optimizer = opt.minimize(loss)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/training/optimizer.py", line 410, in minimize
    name=name)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/training/optimizer.py", line 593, in apply_gradients
    self._create_slots(var_list)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/training/adam.py", line 128, in _create_slots
    colocate_with=first_var)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/training/optimizer.py", line 814, in _create_non_slot_vari
able
    v = variable_scope.variable(initial_value, name=name, trainable=False)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/variables.py", line 183, in __call__
    return cls._variable_v1_call(*args, **kwargs)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/variables.py", line 146, in _variable_v1_call
    aggregation=aggregation)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/variables.py", line 125, in <lambda>
    previous_getter = lambda **kwargs: default_variable_creator(None, **kwarg
s)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/variable_scope.py", line 2444, in default_variable_cre
ator
    expected_shape=expected_shape, import_scope=import_scope)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/variables.py", line 187, in __call__
    return super(VariableMetaclass, cls).__call__(*args, **kwargs)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/variables.py", line 1329, in __init__
    constraint=constraint)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
```

```
s/tensorflow/python/ops/variables.py", line 1491, in _init_from_args
    self._snapshot = array_ops.identity(self._variable, name="read")
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/array_ops.py", line 81, in identity
    return gen_array_ops.identity(input, name=name)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/ops/gen_array_ops.py", line 3454, in identity
    "Identity", input=input, name=name)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/framework/op_def_library.py", line 787, in _apply_op_helpe
r
    op_def=op_def)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/util/deprecation.py", line 488, in new_func
    return func(*args, **kwargs)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/framework/ops.py", line 3274, in create_op
    op_def=op_def)
  File "/Users/kuoyaojen/anaconda3/envs/tensorflow/lib/python3.6/site-package
s/tensorflow/python/framework/ops.py", line 1770, in __init__
    self._traceback = tf_stack.extract_stack()

FailedPreconditionError (see above for traceback): Attempting to use uninitial
ized value optimizer/beta1_power
         [[node optimizer/beta1_power/read (defined at <ipython-input-13-9b730
68dda26>:30)  = Identity[T=DT_FLOAT, _class=["loc:@optimizer/Adam/Assign_1"],
 _device="/job:localhost/replica:0/task:0/device:CPU:0"](optimizer/beta1_powe
r)]]
```

發生了什麼事情?

記得 `FailedPreconditionError` 在什麼時候會出現嗎?

檢查 Graph 上面有哪些 Variables

In [15]:
```
for v in tf.global_variables():
  print(v)
```

```
<tf.Variable 'weights/Variable:0' shape=(1, 1) dtype=float32_ref>
<tf.Variable 'weights/Variable_1:0' shape=(1,) dtype=float32_ref>
<tf.Variable 'optimizer/beta1_power:0' shape=() dtype=float32_ref>
<tf.Variable 'optimizer/beta2_power:0' shape=() dtype=float32_ref>
<tf.Variable 'weights/Variable/Adam:0' shape=(1, 1) dtype=float32_ref>
<tf.Variable 'weights/Variable/Adam_1:0' shape=(1, 1) dtype=float32_ref>
<tf.Variable 'weights/Variable_1/Adam:0' shape=(1,) dtype=float32_ref>
<tf.Variable 'weights/Variable_1/Adam_1:0' shape=(1,) dtype=float32_ref>
```

因為我們使用的 AdamOptimizer 中有宣告變數張量

初始化所有的變數張量！

In [16]:

```
n_steps = 15000
file_writer_path = "./graphs/logistic-regression"
loss_history = []

with tf.Session() as sess:
  sess.run(tf.global_variables_initializer()) # 初始化所有的變數張量!
  train_writer = tf.summary.FileWriter(file_writer_path, tf.get_default_graph())
  for i in range(n_steps):
    feed_dict = {
      X: X_train,
      y: y_train
    }
    _, loss_ = sess.run([optimizer, loss], feed_dict=feed_dict)
    loss_history.append(loss_)
    if i % 500 == 0:
      print("step {}, loss: {}".format(i, loss_))
  w_final, b_final = sess.run([W, b])
```

```
step 0, loss: 177660.65625
step 500, loss: 118222.859375
step 1000, loss: 58788.55859375
step 1500, loss: 235.24574279785156
step 2000, loss: 168.48817443847656
step 2500, loss: 165.7267608642578

step 3000, loss: 162.44384765625
step 3500, loss: 158.67784118652344
step 4000, loss: 154.50735473632812
step 4500, loss: 150.05201721191406
step 5000, loss: 145.462158203125
step 5500, loss: 140.89878845214844
step 6000, loss: 136.51068115234375
step 6500, loss: 132.41595458984375
step 7000, loss: 128.69366455078125
step 7500, loss: 125.38479614257812
step 8000, loss: 122.49922943115234
step 8500, loss: 120.02470397949219
```

In [17]:
```
import matplotlib.pyplot as plt

plt.plot(range(n_steps), loss_history)
plt.title("Loss Summary")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.show()
```

```
<Figure size 640x480 with 1 Axes>
```

In [18]:

```
import numpy as np

def sigmoid(x):
  ans = 1 / (1 + np.exp(-x))
  return(ans)
y_logit = (np.dot(X_test, w_final) + b_final[0]).ravel()
y_one_prob = sigmoid(y_logit)
y_pred = np.round(y_one_prob)
acc = accuracy_score(y_test, y_pred)
```

In [19]:
```
print(w_final)
print(b_final)
print(acc)
```

```
[[-0.01125621]]
[7.857372]
0.8771929824561403
```

隨堂練習

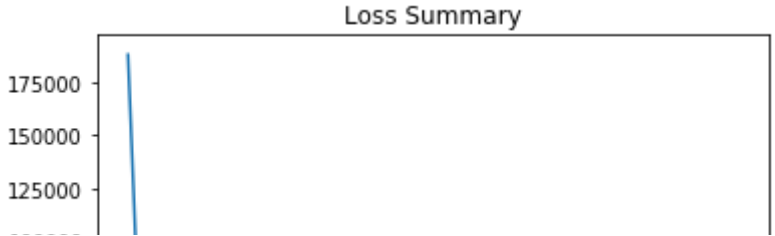
以 breast_cancer 建立一個 Logistic Regression Classifier：class ~ mean area + area error + worst area

In [20]:
```
X_arr = breast_cancer.data[:, [3, 13, 23]]
y_arr = breast_cancer.target
print(X_arr.shape)
print(y_arr.shape)
```

```
(569, 3)
(569,)
```

In [24]:
```
import matplotlib.pyplot as plt

plt.plot(range(n_steps), loss_history)
plt.title("Loss Summary")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.show()
```



In [26]:
```
print(w_final)
print(b_final)
print(acc)
```

```
[[ 0.019638  ]
 [-0.03741419]
 [-0.02478192]]
[9.2107115]
0.9239766081871345
```