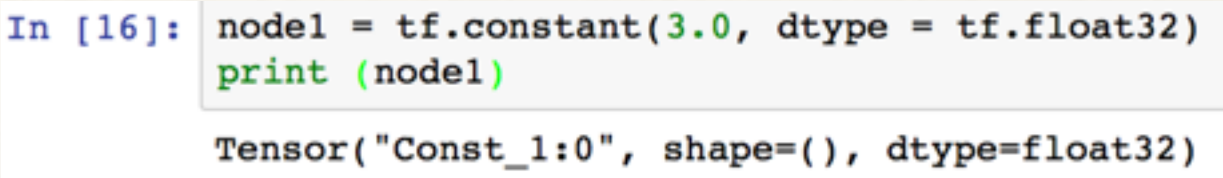
TENSORFLOW應用

Jerry Wu & NTPU Team

基本架構

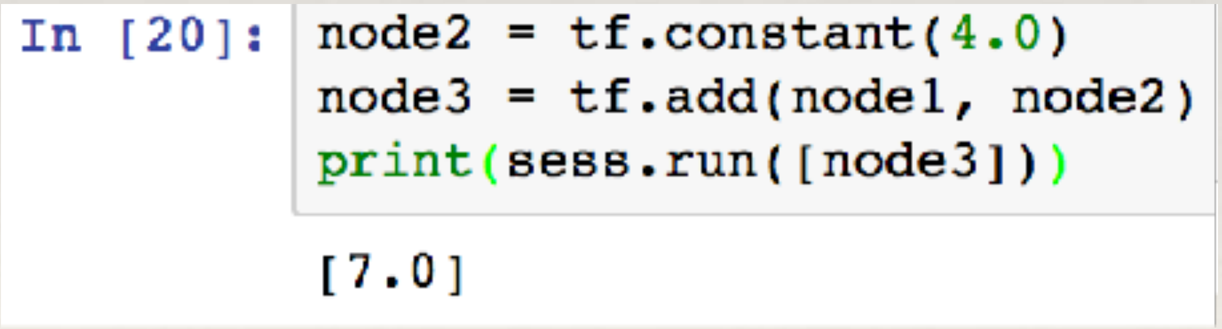


```
In [16]: node1 = tf.constant(3.0, dtype = tf.float32)
         print (node1)

         Tensor("Const_1:0", shape=(), dtype=float32)
```

- computational graph
- tensorflow 由許多運算node組成
- 每個節點接受數個tensors或沒有輸入，輸出一個tensor

SESSION

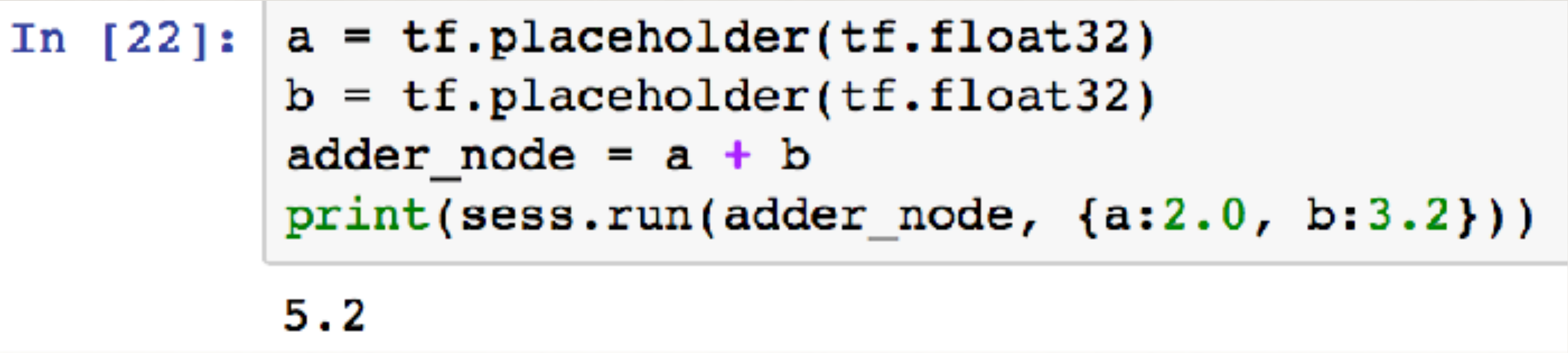
- 進行運算時，必須建立一個session在session內中執行computational graph



```
In [20]: node2 = tf.constant(4.0)
         node3 = tf.add(node1, node2)
         print(sess.run([node3]))

         [7.0]
```

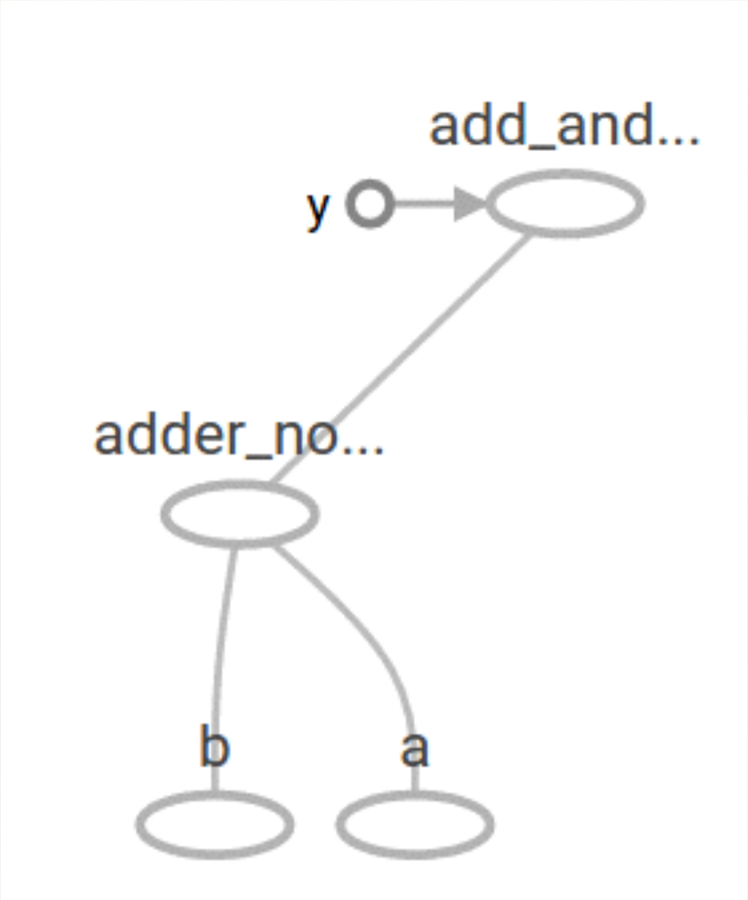
PLACEHOLDER

- 一種可以讓computational graph 保留輸入欄位的節點



```
In [22]: a = tf.placeholder(tf.float32)
         b = tf.placeholder(tf.float32)
         adder_node = a + b
         print(sess.run(adder_node, {a:2.0, b:3.2}))

         5.2
```

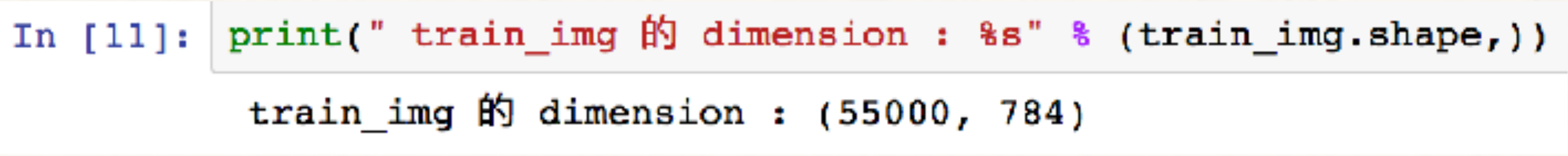
COMPUTATIONAL GRAPH



MNIST機器學習

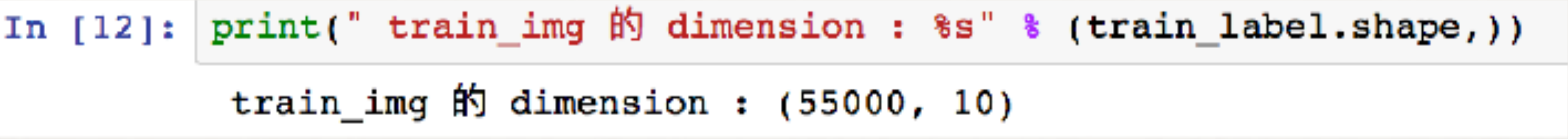
- MINIST DATA手寫數字資料集
- 用Softmax Regression模型解決問題
- Image = x, Label = y

Training Data:
Testing Data:



```
In [11]: print(" train_img 的 dimension : %s" % (train_img.shape,))

          train_img 的 dimension : (55000, 784)
```

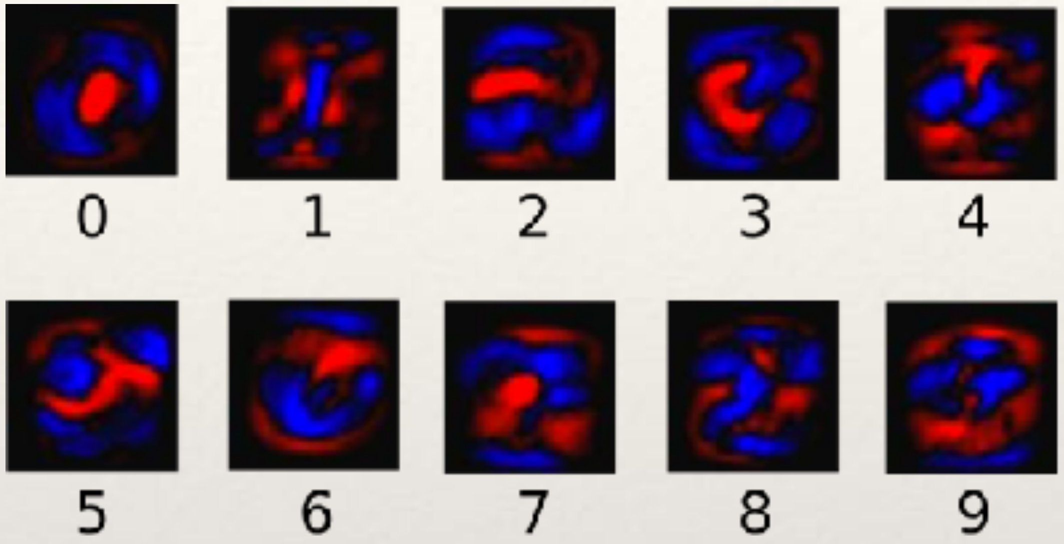
- 784每個圖片中 28*28 pixels 的 array
- 55000代表資料庫大小



```
In [12]: print(" train_img 的 dimension : %s" % (train_label.shape,))

          train_img 的 dimension : (55000, 10)
```

- 10代表1~10
- one - hot vector

- 以下為每個學習的權重，紅色代表負數權重，藍色代表正面權重

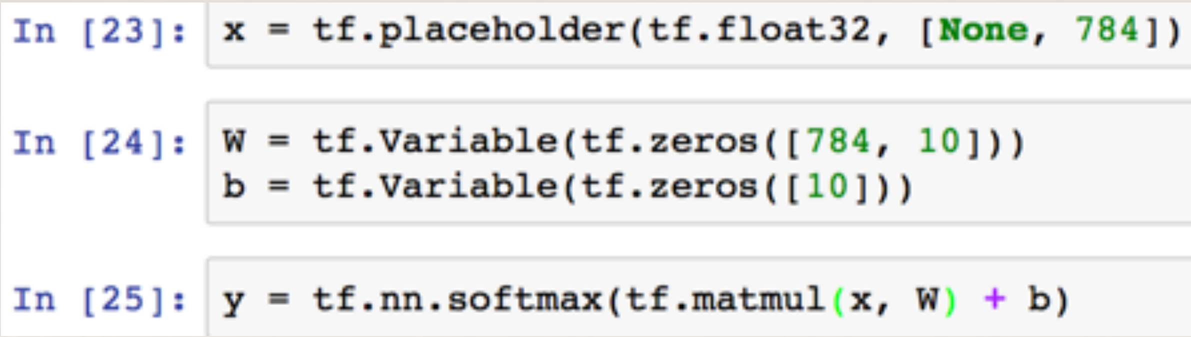


$$\text{evidence}_i = \sum_j W_{i,\ j} x_j + b_i$$

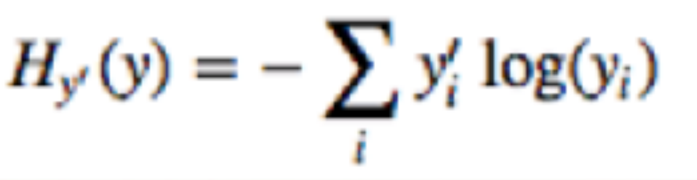
- w代表權重
- x代表輸入的圖片再乘上權重+bias(偏差值)
- j代表所有x相片中的像素j總和

$$y = \mathrm{softmax}(\mathrm{evidence})$$

- 將evidence丟入softmax函數
- softmax函數看成一個轉換成對應10個數字的機率分佈

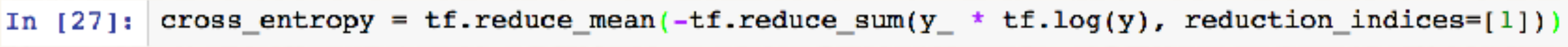


```
In [23]: x = tf.placeholder(tf.float32, [None, 784])

In [24]: W = tf.Variable(tf.zeros([784, 10]))
         b = tf.Variable(tf.zeros([10]))

In [25]: y = tf.nn.softmax(tf.matmul(x, W) + b)
```

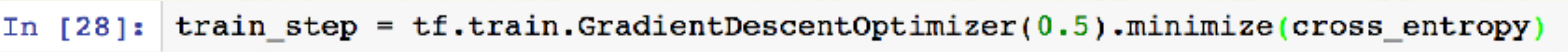


$$H_{y'}(y) = -\sum_i y'_i \log(y_i)$$

- cross-entropy
- y是預測的機率向量
- y’代表實際的結果(one-hot vector)
- 預測與真實情況差距

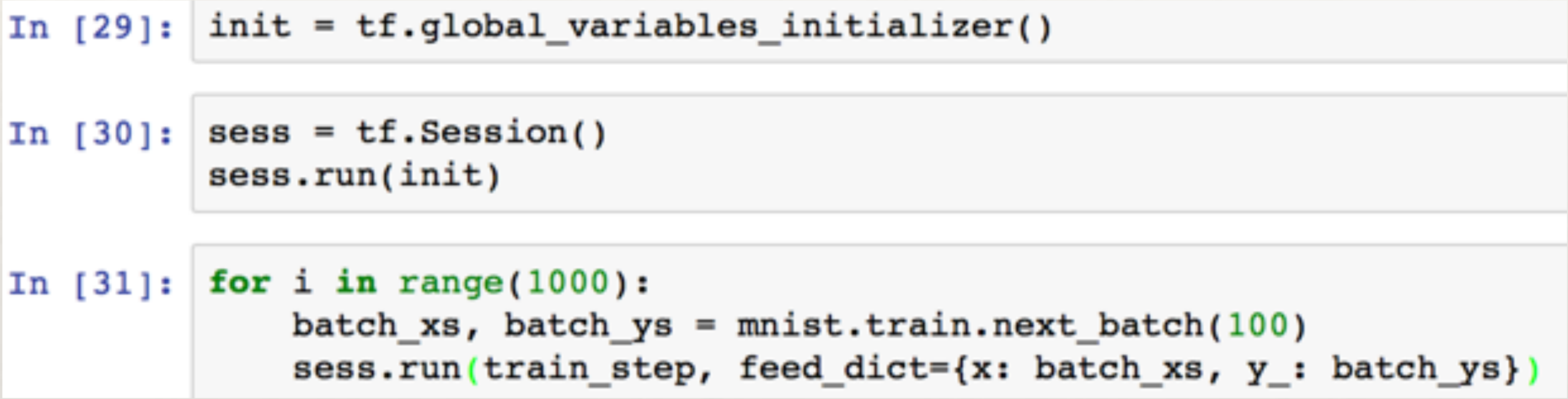


```
In [27]: cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
```



```
In [28]: train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
```

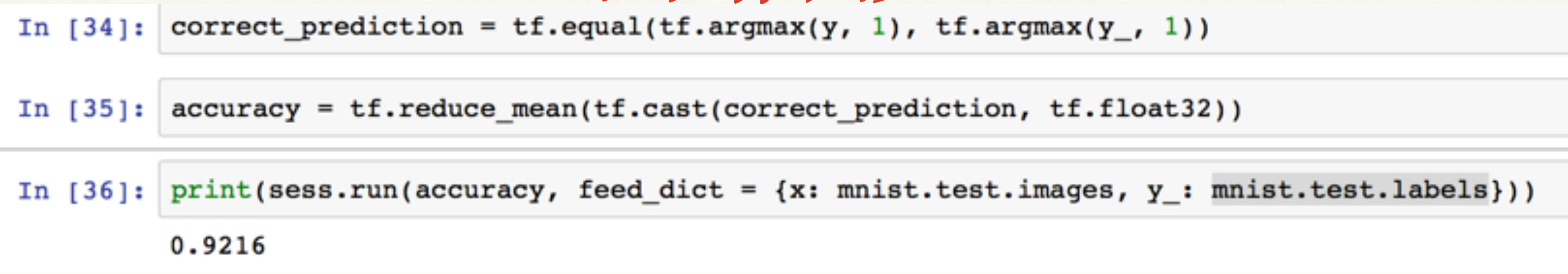
- tensorflow中提供許多最佳化演算法，來訓練模型參數



```
In [29]: init = tf.global_variables_initializer()

In [30]: sess = tf.Session()
         sess.run(init)

In [31]: for i in range(1000):
             batch_xs, batch_ys = mnist.train.next_batch(100)
             sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})
```

- 進行1000次訓練，每次隨機抓100筆數據

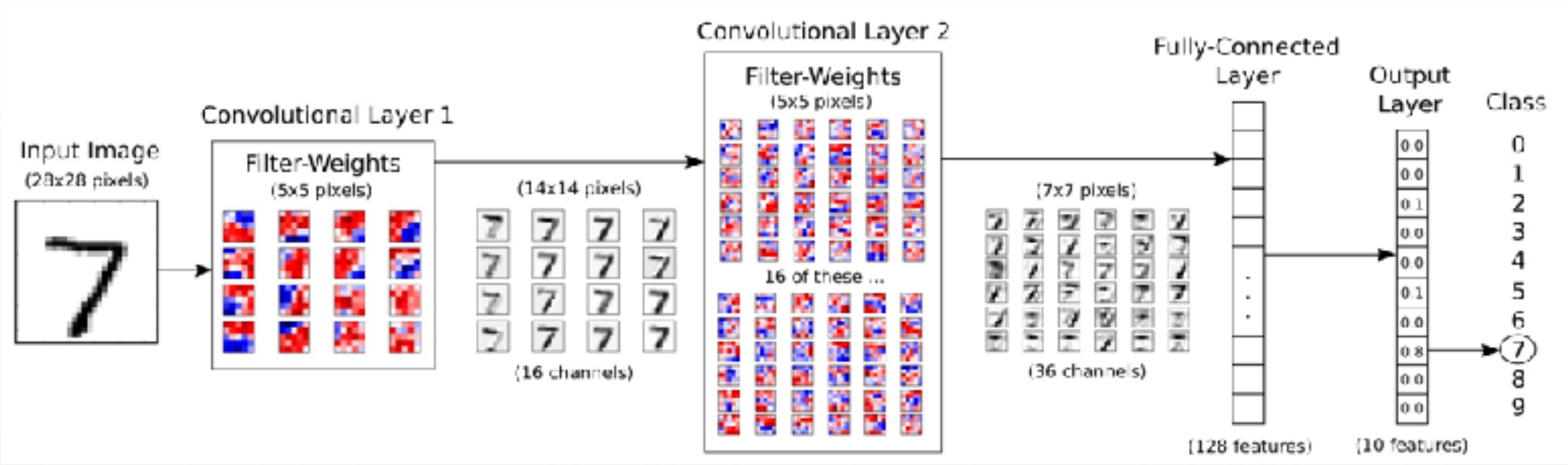
評估模型



```
In [34]: correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))

In [35]: accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

In [36]: print(sess.run(accuracy, feed_dict = {x: mnist.test.images, y_: mnist.test.labels}))

         0.9216
```

- argmax找出某一維tensor中最大值，代表模型中每一筆輸入最有可能的數字
- tf.equal判斷是否正確並回傳布林值
- 使用reduce_mean取平均值

利用CNN訓練模型


Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected
Layer
(128 features)
Output
Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9