



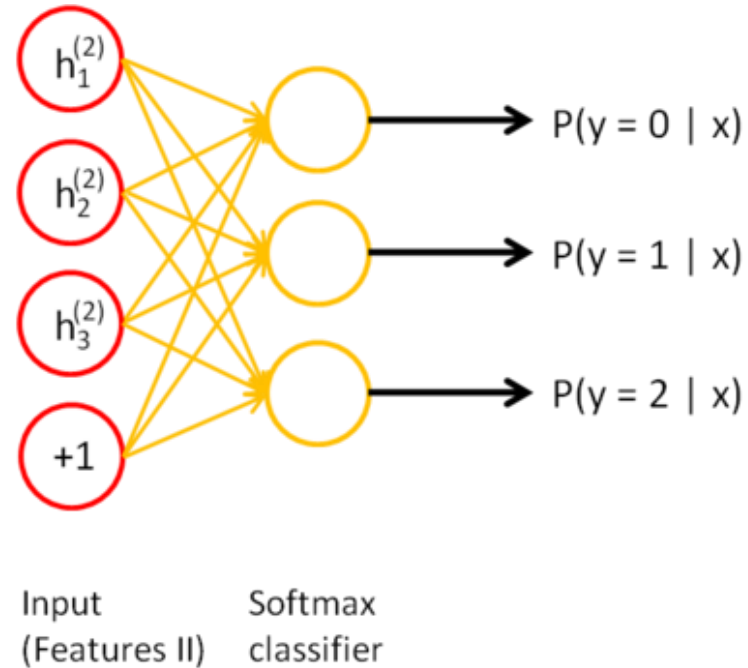
# Simple Softmax (Multi-node Perceptron)

By Comdet Phaudphut , [fb.com/comdet](https://fb.com/comdet) , [comdet.p@gmail.com](mailto:comdet.p@gmail.com)

# Review Previous Lab (Perceptron)

- We have only one node
- We have weight
- We know weighted sum , activation function, we know error
- And we know backprop
- BUT!
  - I only one output , how can we classify more output with perceptron

# Multi-node perceptron



- Same algorithm as one node but we add more node to classify more than one output

# The MNIST of handwritten digits

- Database of images and labels for handwritten digits
- The training set contains 60000 examples
- The test set 10000 examples.
- There are python handled data, more easy to use



```
10 import tensorflow as tf
11 import numpy as np
12 from tensorflow.examples.tutorials.mnist import input_data
13 mnist = input_data.read_data_sets("data/", one_hot = True)
```

ทำการอ่านข้อมูล จาก folder data, กำหนดให้  
label เป็น **one-hot** ?

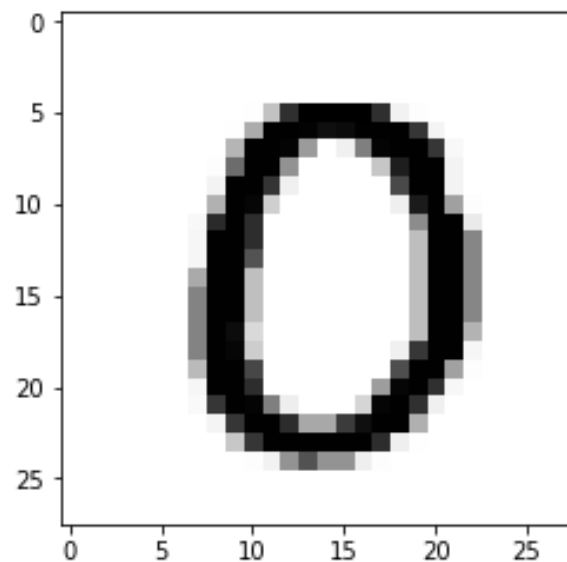
In [3]:

```
1 index = 10
2 tmp = mnist.train.images[index]
3 tmp = tmp.reshape((28,28))
4
5 plt.imshow(tmp, cmap = cm.Greys)
6 plt.show()
7 print("One-hot Label for this images = ", end=" ")
8 onehot_label = mnist.train.labels[index]
9 print(onehot_label)
10 print("Index = %d" % np.argmax(onehot_label))
```

ดึงรูปภาพที่ 100 จาก 60000 จะได้ภาพ ขนาด  
1\*784 จากนั้นทำการปรับให้อยู่ขนาด 28\*28

แสดงรูปภาพ จะปรากฏรูปด้านล่าง เลข 0

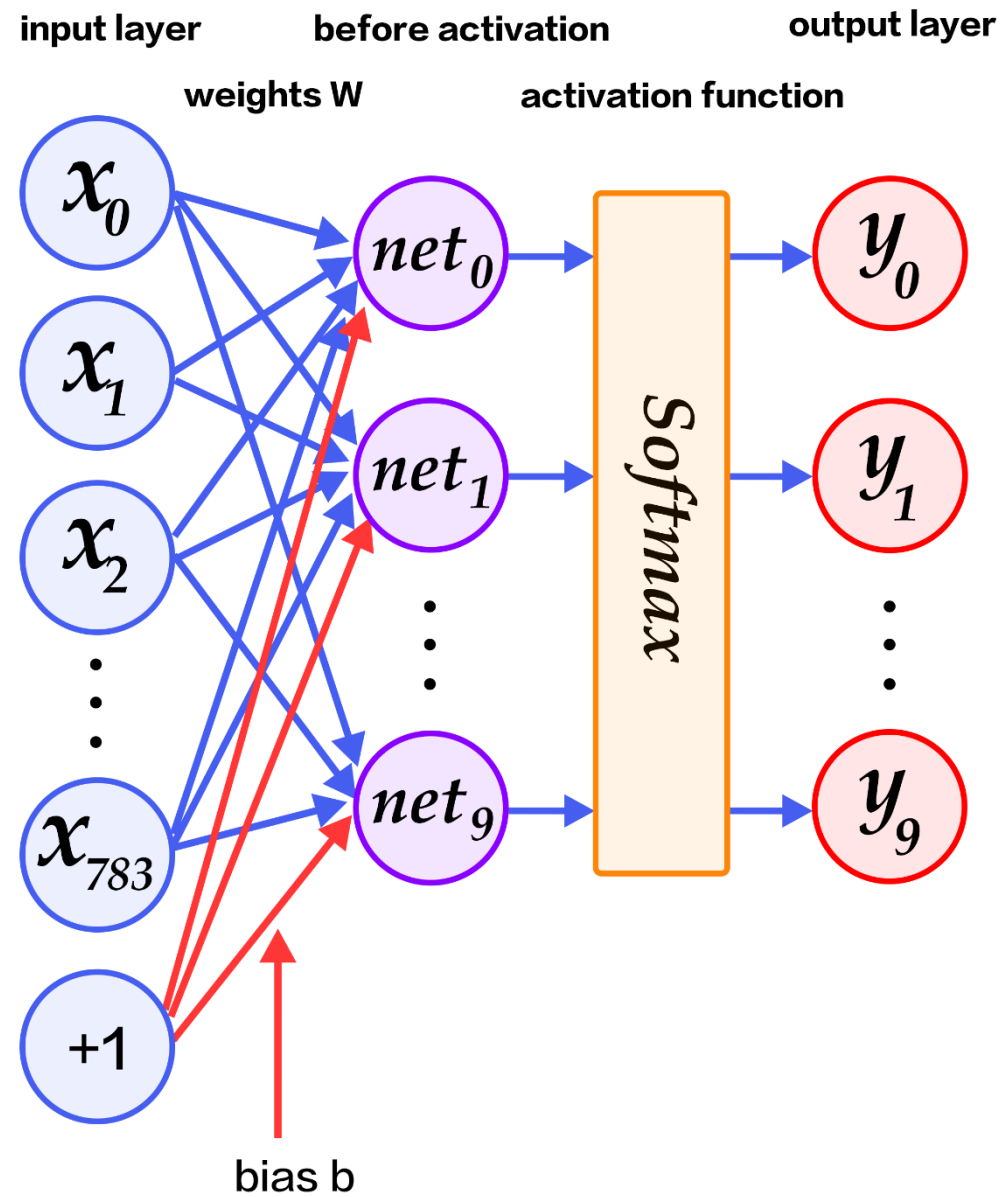
แสดง Label ในรูปแบบ One-hot



ตำแหน่งที่มีค่ามากที่สุด

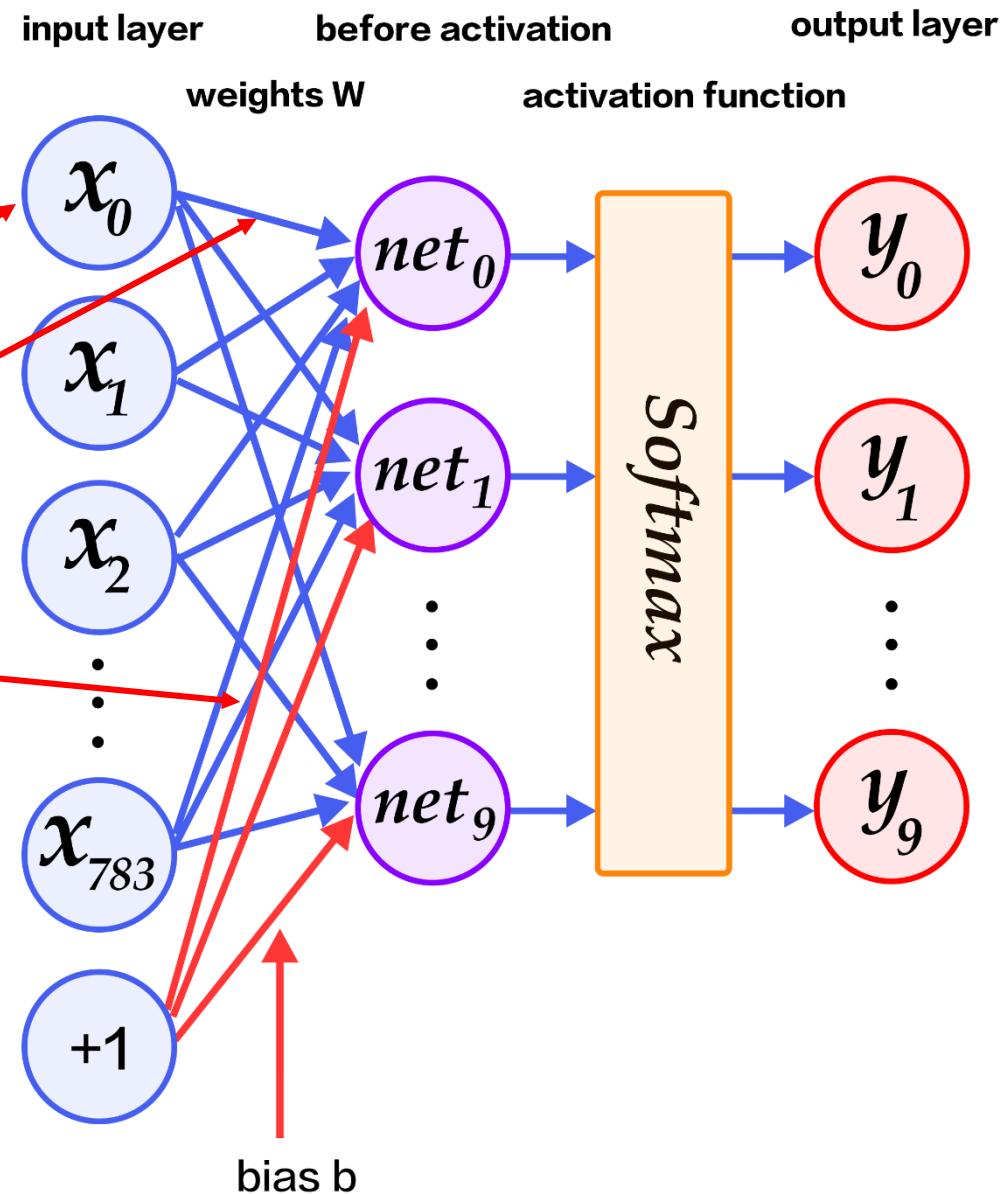
```
One-hot Label for this images = [ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
Index = 0
```





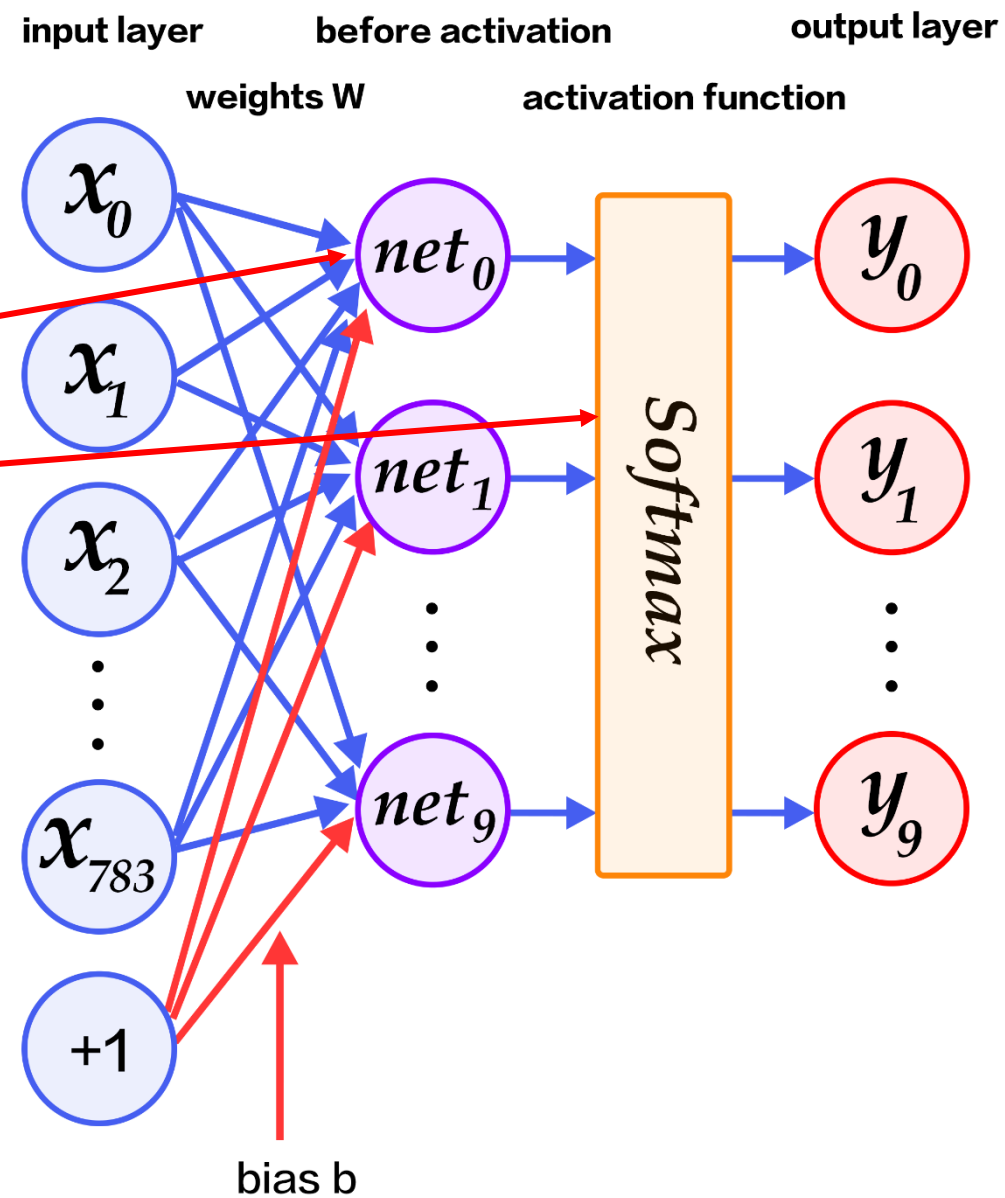
None คือไม่ได้กำหนด จำนวน Batch input

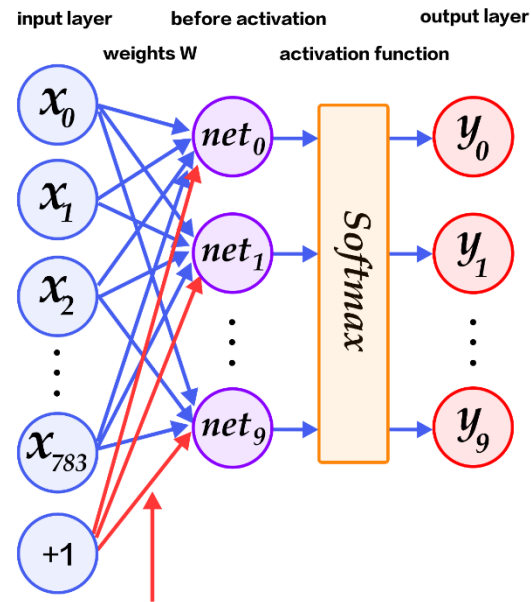
```
1 X = tf.placeholder(tf.float32, [None, 784])  
2  
3 W = tf.Variable(tf.zeros([784, 10]))  
4 b = tf.Variable(tf.zeros([10]))
```





```
1 #model
2 net = tf.matmul(X, W) + b #logits
3 Y = tf.nn.softmax(net)
```





```

1 # Define loss and optimizer
2 Y_ = tf.placeholder(tf.float32, [None, 10])
3 #loss function
4 cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=Y_, logits=net))
5
6 #Optimizer
7 optimizer = tf.train.GradientDescentOptimizer(0.05)
8 train_step = optimizer.minimize(cross_entropy)
9 #or train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
10
11 # % of correct answers found in batch
12 is_correct = tf.equal(tf.argmax(Y,1), tf.argmax(Y_,1))
13 accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))

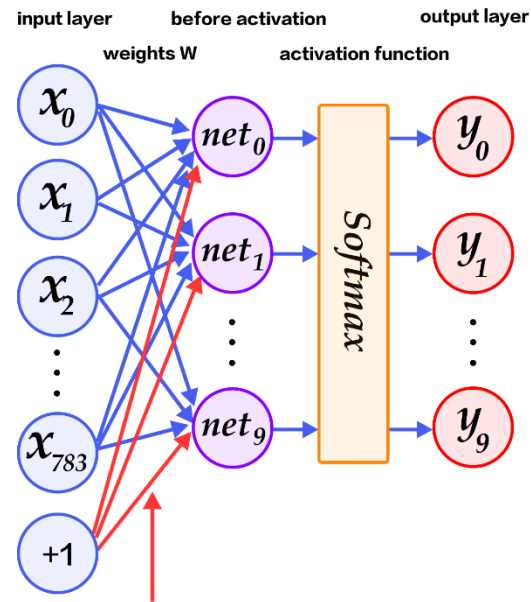
```

Placeholder สำหรับค่า expected

การหาค่า error ระหว่าง expected กับ predicted

สร้าง optimizer สำหรับ minimize error (ปรับ weight)

การหา % ความแม่นยำ



```

1 # Define loss and optimizer
2 Y_ = tf.placeholder(tf.float32, [None, 10])
3 #loss function
4 cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=Y_, logits=net))
5
6 #Optimizer
7 optimizer = tf.train.GradientDescentOptimizer(0.05)
8 train_step = optimizer.minimize(cross_entropy)
9 #or train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
10
11 # % of correct answers found in batch
12 is_correct = tf.equal(tf.argmax(Y,1), tf.argmax(Y_,1))
13 accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))

```

Placeholder สำหรับค่า expected

การหาค่า error ระหว่าง expected กับ predicted

สร้าง optimizer สำหรับ minimize error (ปรับ weight)

การหา % ความแม่นยำ

```

1 sess = tf.InteractiveSession()
2 tf.global_variables_initializer().run()
3
4 for i in range(10000):
5     #load batch of images and correct answers
6     batch_X, batch_Y = mnist.train.next_batch(100)
7     train_data={X: batch_X, Y_: batch_Y}
8     #train
9     sess.run(train_step, feed_dict=train_data)
10    if i % 100 == 0:
11        #success ?
12        a,c = sess.run([accuracy,cross_entropy],feed_dict=train_data)
13        print("Step : %d acc = %.4f loss = %.4f" % (i,a,c))
14    #--- edit
15    #success on test data?
16    test_data = {X: mnist.test.images, Y_: mnist.test.labels}
17    a,c = sess.run([accuracy, cross_entropy], feed_dict=test_data)
18    print("Test data acc = %.4f loss = %.4f" % (a,c))

```

สอนไปเรื่อย ๆ 10000 epoch

ดึงรูปและ label มาจากข้อมูล mnist ที่ละ 100

สร้าง Feeddicit ส่งข้อมูลเข้า placeholder

สอนโมเดล

ทุก 100 รอบจะส่ง training set เข้าไป  
ทดสอบ accuracy และ loss

เมื่อ train เสร็จจะทดสอบโมเดลด้วย 10000  
รูปที่ใช้ทดสอบ เพื่อหาค่าความถูกต้อง

Run the code and see what's happen!?

Try add more 10000 epoch and see what happen!?