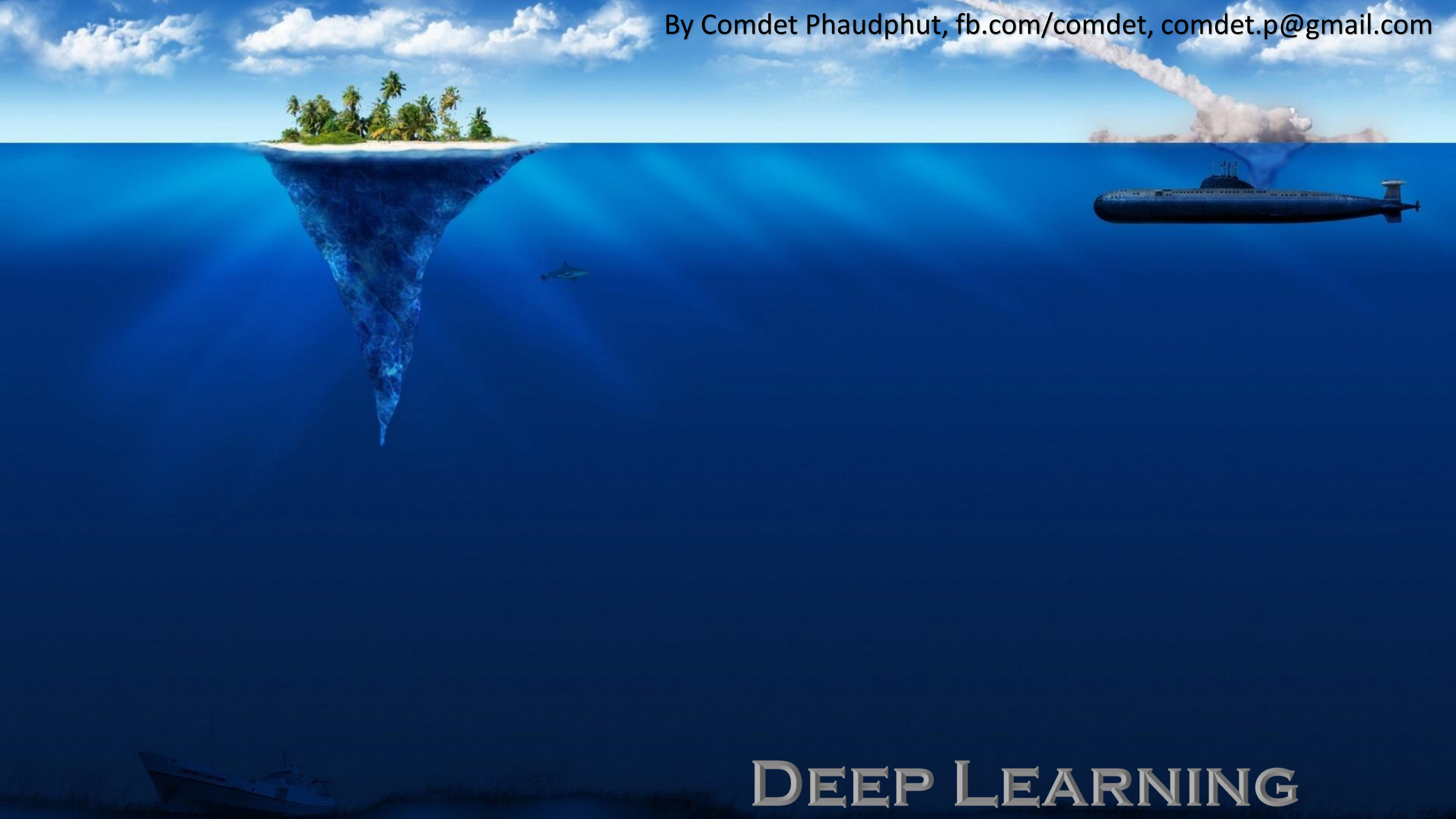
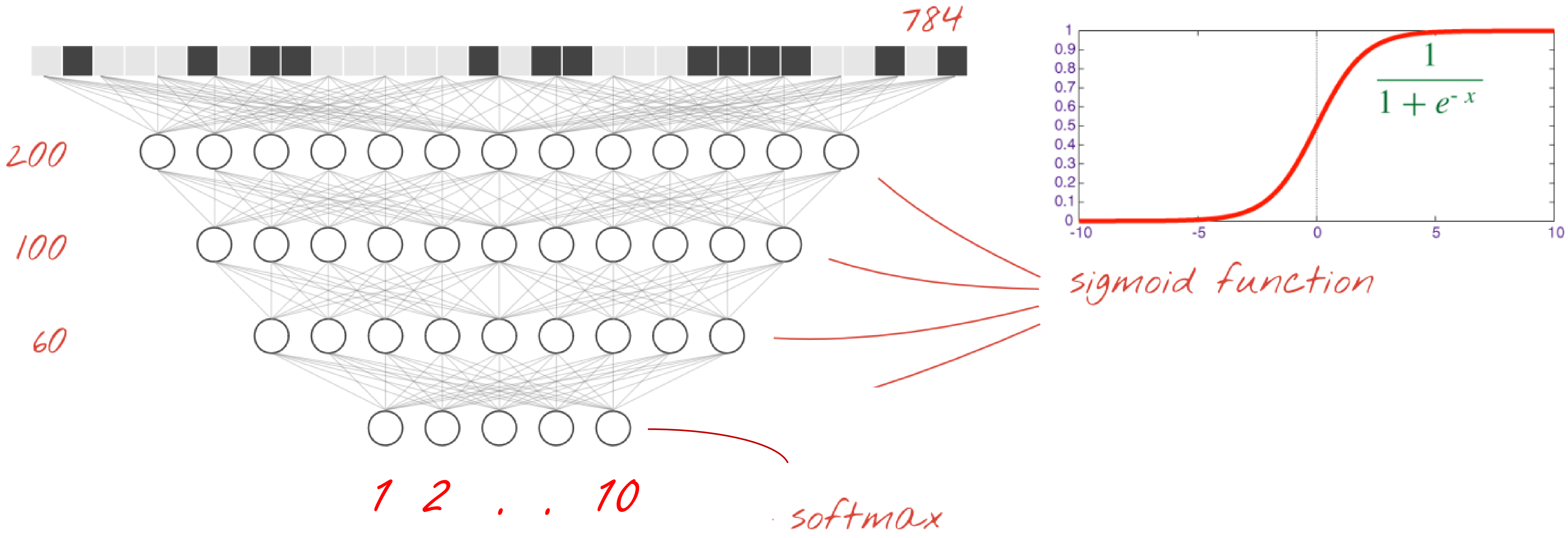


By Comdet Phaudphut, fb.com/comdet, comdet.p@gmail.com

Deep Learning

Review Prev Lab

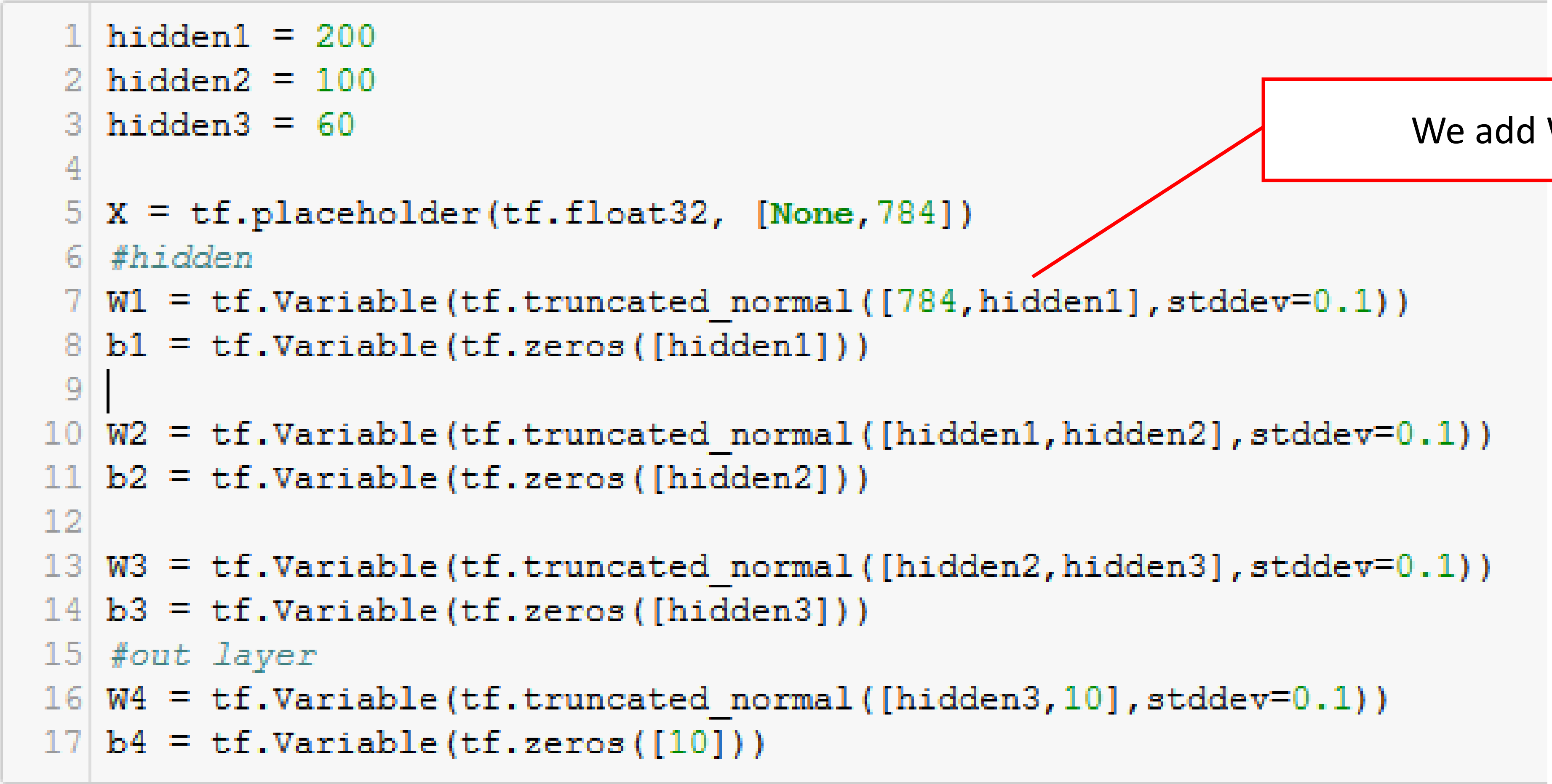
- We add more layer to classify images that called “hidden layer”
- We use sigmoid as activation function to reduce output between 0-1
- We use summary to logging output while training
- Every 100 step we sneaked test output and write to summary too.
- We gain more accuracy, resulted around 97-98%
- But ?
  - Can we gain more accuracy by add more layer?
  - By theoretical, more layer more complication, more degree of freedom



We add 3 hidden layers!

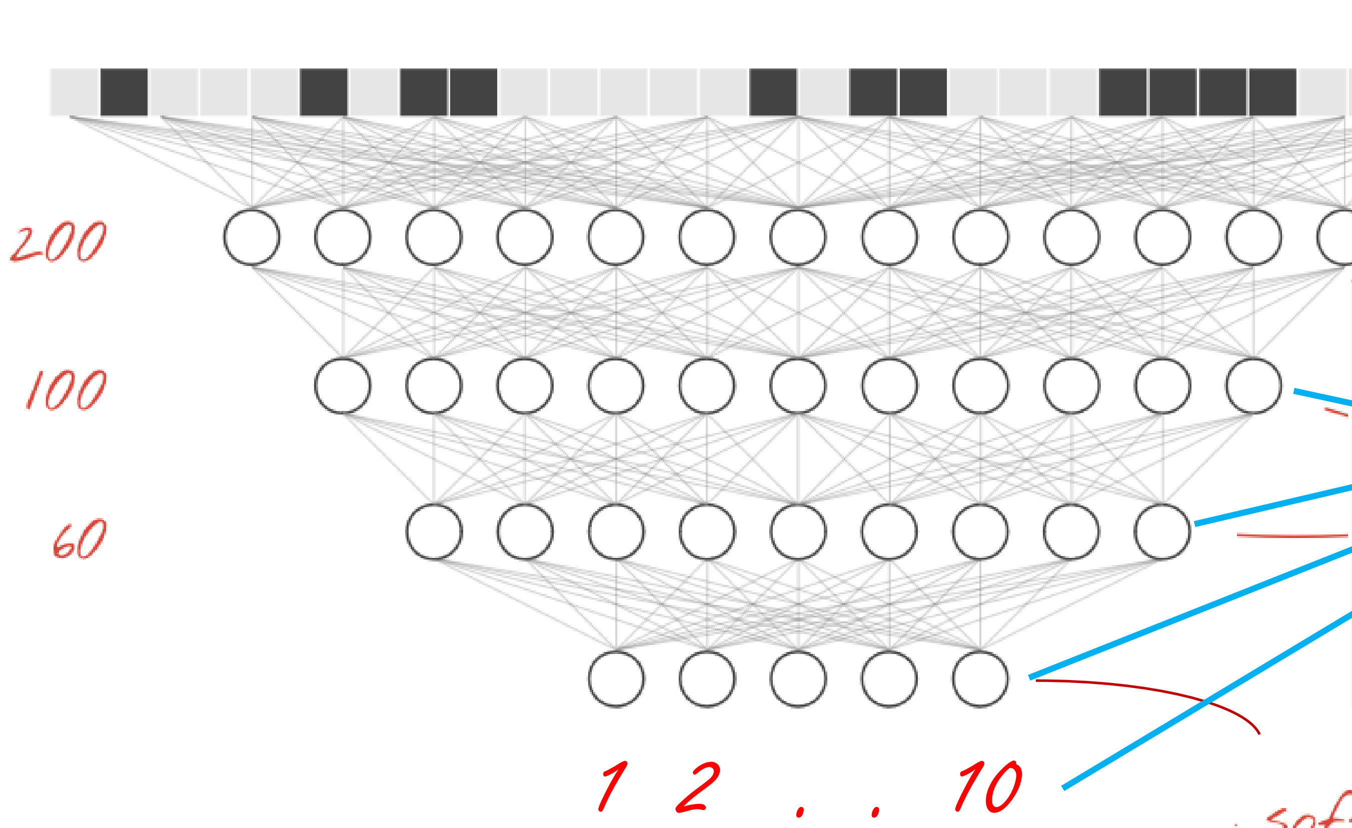
What happen?

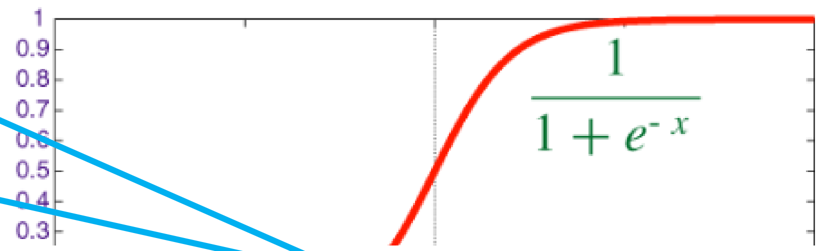
- Theoretical, we know how can we gain more accuracy by adding more layer.
- So we add 200 -> 100 -> 60 hidden layer
- And now, the hidden layer more then one we can called **“DEEP” Neural Network** ,ah! … finally.
- Let take a look closer!

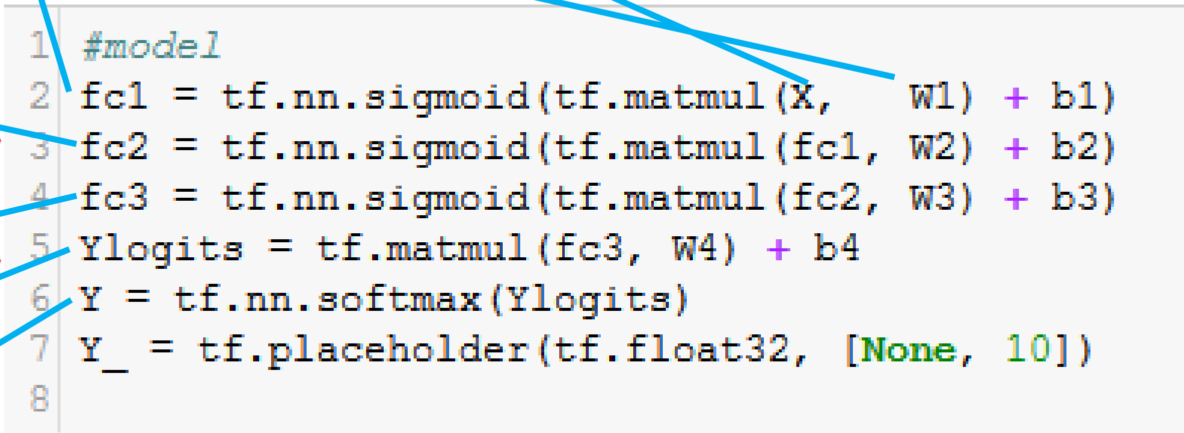


```python
hidden1 = 200
hidden2 = 100
hidden3 = 60

X = tf.placeholder(tf.float32, [None,784])
#hidden
W1 = tf.Variable(tf.truncated_normal([784,hidden1],stddev=0.1))
b1 = tf.Variable(tf.zeros([hidden1]))

W2 = tf.Variable(tf.truncated_normal([hidden1,hidden2],stddev=0.1))
b2 = tf.Variable(tf.zeros([hidden2]))

W3 = tf.Variable(tf.truncated_normal([hidden2,hidden3],stddev=0.1))
b3 = tf.Variable(tf.zeros([hidden3]))
#out layer
W4 = tf.Variable(tf.truncated_normal([hidden3,10],stddev=0.1))
b4 = tf.Variable(tf.zeros([10]))
```

We add W1,b1 … W4,b4





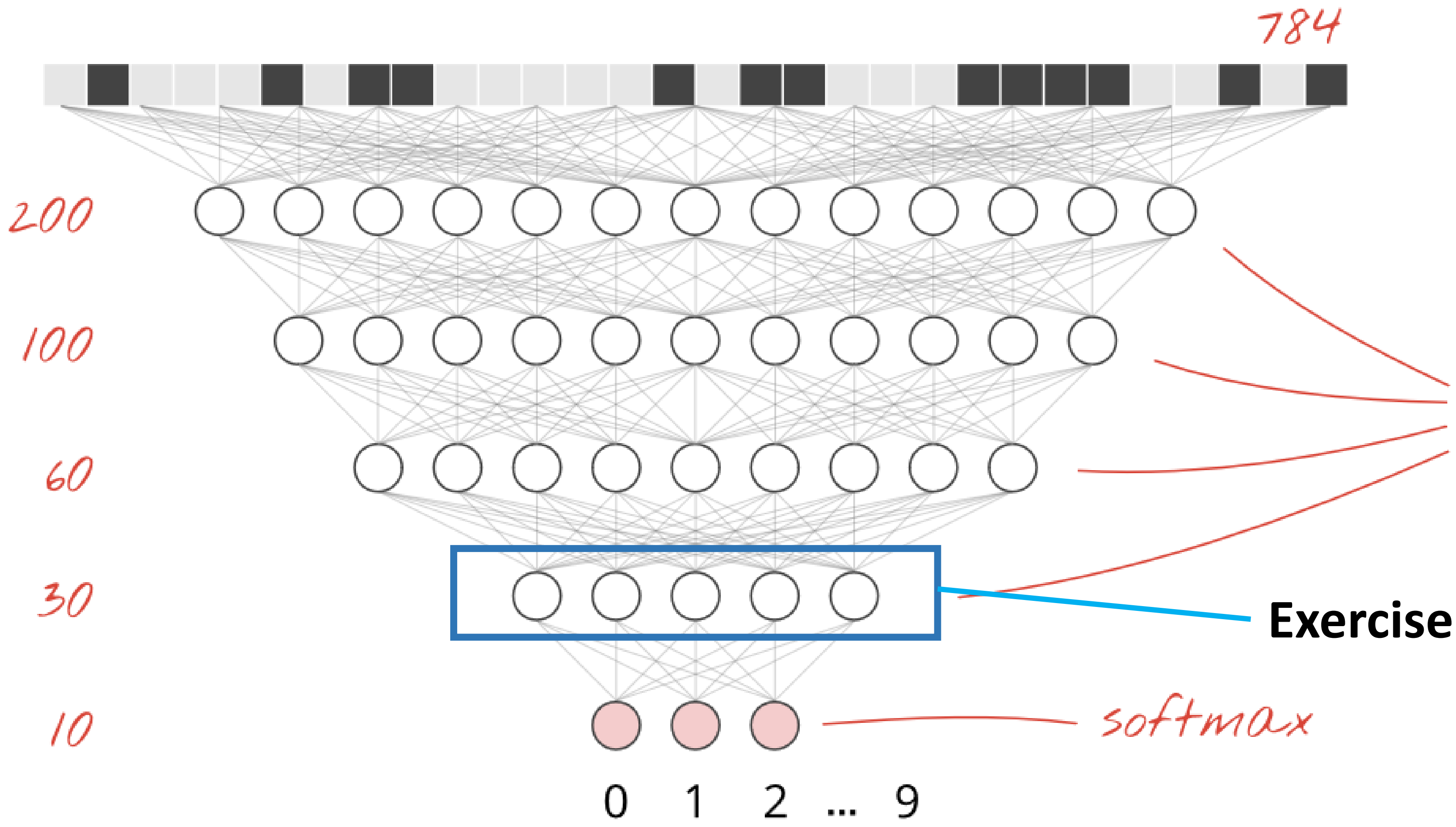


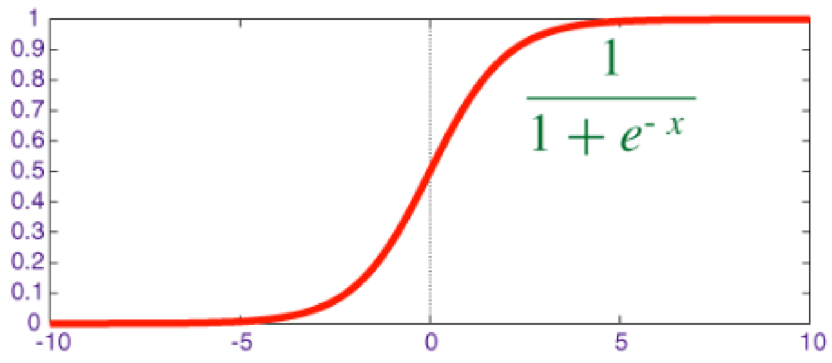
```
#model
fc1 = tf.nn.sigmoid(tf.matmul(X,    W1) + b1)
fc2 = tf.nn.sigmoid(tf.matmul(fc1, W2) + b2)
fc3 = tf.nn.sigmoid(tf.matmul(fc2, W3) + b3)
Ylogits = tf.matmul(fc3, W4) + b4
Y = tf.nn.softmax(Ylogits)
Y_ = tf.placeholder(tf.float32, [None, 10])
```

Run & open tensorboard

```
tensorboard –logdir=“logs”
```

and see what happen!





sigmoid function

Exercise : Add another 30 hidden layer