

# Assignment 3

(Deep learning Methods and Applications)

2016025678 이강민

## 1. Code Description

```
1  # Check Point
2
3  import tensorflow as tf
4  from tensorflow.examples.tutorials.mnist import input_data
5
6  mnist = input_data.read_data_sets("./mnist/data/", one_hot = True)
7
8  X = tf.placeholder(tf.float32, [None, 784])
9  Y = tf.placeholder(tf.float32, [None, 10])
10
11 W1 = tf.Variable(tf.random_uniform([784, 256], -1., 1.))
12 b1 = tf.Variable(tf.random_uniform([256], -1., 1.))
13 L1 = tf.nn.sigmoid(tf.matmul(X, W1) + b1)
14
15 W2 = tf.Variable(tf.random_uniform([256, 256], -1., 1.))
16 b2 = tf.Variable(tf.random_uniform([256], -1., 1.))
17 L2 = tf.nn.sigmoid(tf.matmul(L1, W2) + b2)
18
19 W3 = tf.Variable(tf.random_uniform([256, 10], -1., 1.))
20 b3 = tf.Variable(tf.random_uniform([10], -1., 1.))
21 logits = tf.matmul(L2, W3) + b3
22 hypothesis = tf.nn.softmax(logits)
23 cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(labels=Y, logits=logits))
24 opt = tf.train.GradientDescentOptimizer(learning_rate = 0.1).minimize(cost)
25
26 batch_size = 100
27
28 ckpt_path = "model/"
```

- line 11~21: neural network의 각 layer를 설정해줍니다.
- line 22: logits을 softmax 함수에 적용하여 label을 확률적으로 나타냅니다.
- line 23: cost 함수를 cross entropy 함수로 설정합니다.
- line 24: optimizer는 gradient descent optimizer를 사용합니다.
- line 26: batch size는 100으로 합니다.
- line 28: check point를 저장하는 경로를 설정합니다.

```

30 with tf.Session() as sess:
31     sess.run(tf.global_variables_initializer())
32     saver = tf.train.Saver()
33     for epoch in range(15):
34         avg_cost = 0
35         total_batch = int(mnist.train.num_examples/batch_size)
36         for i in range(total_batch):
37             batch_xs, batch_ys = mnist.train.next_batch(batch_size)
38             c, _ = sess.run([cost, opt], feed_dict = {X: batch_xs, Y: batch_ys})
39             avg_cost += c / total_batch
40             print('Epoch:', '%d' % (epoch+1), 'cost =', '{:.9f}'.format(avg_cost))
41             is_correct = tf.equal(tf.argmax(hypothesis, 1), tf.argmax(Y, 1))
42             accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))
43             print("Accuracy", sess.run(accuracy, feed_dict = {X: mnist.test.images, Y: mnist.test.labels}))
44
45         saver.save(sess, ckpt_path)
46         saver.restore(sess, ckpt_path)
47
48     print("Accuracy", sess.run(accuracy, feed_dict = {X: mnist.test.images, Y: mnist.test.labels}))
49

```

- line 30~31: 학습을 시작하기 전에 그래프를 생성하고 모든 변수를 초기화시켜줍니다.
- line 32: check point를 저장하거나 불러오는 객체를 생성합니다.
- line 33~40: 모델을 학습시킵니다.
- line 41~43: 모델의 accuracy를 구합니다.
- line 45: 학습이 끝난 모델을 설정한 경로에 check point로 저장합니다.
- line 46: 설정한 경로에서 check point를 불러옵니다.

## 2. Result

```
Epoch: 1 cost = 0.894053180
Epoch: 2 cost = 0.437356474
Epoch: 3 cost = 0.355751146
Epoch: 4 cost = 0.309509455
Epoch: 5 cost = 0.278148182
Epoch: 6 cost = 0.254582238
Epoch: 7 cost = 0.235984411
Epoch: 8 cost = 0.220103393
Epoch: 9 cost = 0.207502571
Epoch: 10 cost = 0.196151071
Epoch: 11 cost = 0.185626195
Epoch: 12 cost = 0.176710856
Epoch: 13 cost = 0.168788150
Epoch: 14 cost = 0.161212167
Epoch: 15 cost = 0.154534612
```

Accuracy 0.9444

WARNING:tensorflow:From C:\Users\kolok\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\python\ttraining\saver.py:1266: checkpoint\_exists (from tensorflow.python.training.checkpoint\_management) is deprecated and will be removed in a future version.

Instructions for updating:

Use standard file APIs to check for files with this prefix.

INFO:tensorflow:Restoring parameters from model/

Accuracy 0.9444

		Name ▾	Last Modified	File size
..			몇 초 전	
checkpoint			1분 전	59 B

- check point가 잘 저장되어있는 것을 확인할 수 있었습니다. 또한, 저장되어있는 check point를 불러와 기존의 모델인 것을 확인할 수 있었습니다.