

시즌 1 - 딥러닝의 기본 - ML lab 10

노트북: 모두를 위한 머신러닝
만든 날짜: 2019-01-09 오후 2:14
작성자: rr
태그: #모두를 위한, .ML lab

수정한 날짜: 2019-01-09 오후 5:18

ML Lab 10

- mnist_nn

```
24 W1 = tf.Variable(tf.random_normal([784, 256]))
25 b1 = tf.Variable(tf.random_normal([256]))
26 L1 = tf.nn.relu(tf.matmul(X, W1) + b1)
27
28 W2 = tf.Variable(tf.random_normal([256, 256]))
29 b2 = tf.Variable(tf.random_normal([256]))
30 L2 = tf.nn.relu(tf.matmul(L1, W2) + b2)
31
32 W3 = tf.Variable(tf.random_normal([256, 10]))
33 b3 = tf.Variable(tf.random_normal([10]))
34 hypothesis = tf.matmul(L2, W3) + b3
```

- mnist_nn_xavier

```
25 W1 = tf.get_variable("W1", shape=[784, 256],
26                       initializer=tf.contrib.layers.xavier_initializer())
27 b1 = tf.Variable(tf.random_normal([256]))
28 L1 = tf.nn.relu(tf.matmul(X, W1) + b1)
29
30 W2 = tf.get_variable("W2", shape=[256, 256],
31                       initializer=tf.contrib.layers.xavier_initializer())
32 b2 = tf.Variable(tf.random_normal([256]))
33 L2 = tf.nn.relu(tf.matmul(L1, W2) + b2)
34
35 W3 = tf.get_variable("W3", shape=[256, 10],
36                       initializer=tf.contrib.layers.xavier_initializer())
37 b3 = tf.Variable(tf.random_normal([10]))
38 hypothesis = tf.matmul(L2, W3) + b3
```

초기값 설정 잘 됨

레이어 늘리면 overfitting 문제 생김

- dropout

```

29 W1 = tf.get_variable("W1", shape=[784, 512],
30                       initializer=tf.contrib.layers.xavier_initializer())
31 b1 = tf.Variable(tf.random_normal([512]))
32 L1 = tf.nn.relu(tf.matmul(X, W1) + b1)
33 L1 = tf.nn.dropout(L1, keep_prob=keep_prob)
34
35 W2 = tf.get_variable("W2", shape=[512, 512],
36                       initializer=tf.contrib.layers.xavier_initializer())
37 b2 = tf.Variable(tf.random_normal([512]))
38 L2 = tf.nn.relu(tf.matmul(L1, W2) + b2)
39 L2 = tf.nn.dropout(L2, keep_prob=keep_prob)

```

overfitting 문제 해결

keep_prob는 placeholder로 만들어주고,

```
keep_prob = tf.placeholder(tf.float32)
```

학습하는 단계에서는 keep_prob를 0.7이라고 feed_dict로 만든 다음 넘겨줌

```
feed_dict = {X: batch_xs, Y: batch_ys, keep_prob: 0.7}
```

학습이 끝나게 되면 keep_prob값을 반드시 1로 만들어줌

```
print('Accuracy:', sess.run(accuracy, feed_dict={
    X: mnist.test.images, Y: mnist.test.labels, keep_prob: 1}))
```