

시즌 1 - 딥러닝의 기본 - ML lab

노트북: 모두를 위한 머신러닝
만든 날짜: 2018-12-27 오후 2:24
작성자: gpfl104@gmail.com
태그: #모두를 위한, .ML lab

수정한 날짜: 2018-12-27 오후 4:46

ML lab 01

```
hello = tf.constant("Hello, TensorFlow!") # constant 노드를 만듦
sess = tf.Session() # 세션을 만듦
print(sess.run(hello)) # hello 노드를 실행시킴
```

- 데이터 타입

Tensor - constant, placeholder

Variable - Variable

- Tensor Ranks, Shapes, Types

Ranks - 차원

Shapes - element에 몇 개 들어있느냐

Types - data type (float32, int32 ...)

ML lab 03

```
cost = tf.reduce_mean(tf.square(hypothesis - y))
```

hypothesis: 예측값

hypothesis - y: 예측값에서 실제 데이터 뺀 값

square: 제곱하라

reduce_mean: 평균을 구하라

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^m (W x^{(i)} - y^{(i)}) x^{(i)}$$

```
# Minimize: Gradient Descent using derivative:
W -= learning_rate * derivative
learning_rate = 0.1
gradient = tf.reduce_mean((W * X - Y) * X)
descent = W - learning_rate * gradient
update = W.assign(descent)
```

descent 를 assign 해야 하는데 텐서플로우에서는 그냥 연산하면 안되고 W.assign으로 해줘야 한다.
이 값을 update에 넣으면, 이제 update를 그래프에서 실행시키게 되면 위에 동작들이 실행되게 됩니다.

```
# Launch the graph in a session.
# 그래프이기 때문에 session 만듦
```

```

sess = tf.Session()
# Initializes global variables in the graph.
# 변수 초기화
sess.run(tf.global_variables_initializer())

for step in range(21):
    # update 불러옴. X, Y 값도 던져줌.
    sess.run(update, feed_dict={X: x_data, Y: y_data})
    print(step, sess.run(cost, feed_dict={X: x_data, Y: y_data}), sess.run(W))

```

- minimize

```

# Minimize: Gradient Descent using derivative:  $W \leftarrow W - \text{learning\_rate} * \text{derivative}$ 
learning_rate = 0.1
gradient = tf.reduce_mean((W * X - Y) * X)
descent = W - learning_rate * gradient
update = W.assign(descent)

```

위는 식이 쉬워서 직접 적었다.

하지만 식이 어려워진다면?

아래처럼 텐서플로우에서는 알아서 cost 값을 미분해달라고 할 수 있다.

```

# Minimize: Gradient Descent Magic
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.1)
train = optimizer.minimize(cost)

```