02 - Simple Liner Regression

201804188 권은채

import tensorflow as tf #탠서플로우를 tf로 불러온다.

import numpy as np #

tf.enable\_eager\_execution() #execution을 활성화하여 즉시 실행하도록 한다.

# Data

x\_data = [1, 2, 3, 4, 5]

y\_data = [1, 2, 3, 4, 5]

#x, y데이터에 [1,2,3,4,5]를 준다.

# W, b initialize

W = tf.Variable(2.9)

b = tf.Variable(0.5)

#처음의 w와 b의 값을 임의로 설정한다.

# W, b update

for i in range(100): #i가 100까지 변한다.

# Gradient descent

with tf.GradientTape() as tape:

#with 블록 안에서 변화하는 W, b를 tape에 기록한다.

hypothesis = W \* x\_data + b

cost = tf.reduce\_mean(tf.square(hypothesis - y\_data))

W\_grad, b\_grad = tape.gradient(cost, [W, b])

W.assign\_sub(learning\_rate \* W\_grad)

b.assign\_sub(learning\_rate \* b\_grad)

#w, b값을 업데이트한다.

if i % 10 == 0:

print("{:5}|{:10.4f}|{:10.4f}|{:10.6f}".format(i, W.numpy(), b.numpy(), cost))

#확인을 위하여 10번마다 한번씩 I, w, b, cost를 출력한다.

#결과창

# I w b cost

0| 2.4520| 0.3760| 45.660004

10| 1.1036| 0.0034| 0.206336

20| 1.0128| -0.0209| 0.001026

30| 1.0065| -0.0218| 0.000093

40| 1.0059| -0.0212| 0.000083

50| 1.0057| -0.0205| 0.000077

60| 1.0055| -0.0198| 0.000072

70| 1.0053| -0.0192| 0.000067

80| 1.0051| -0.0185| 0.000063

90| 1.0050| -0.0179| 0.000059

tf.Tensor(5.0066934, shape=(), dtype=float32)

tf.Tensor(2.4946523, shape=(), dtype=float32)

#cost함수가 최소화되고 w는 1로 가까워지고 b는 0으로 가까워진다.