배추가격 예측 훈련 코드

from google.colab import drive

import os

if os.path.exists('/content/gdrive')==False:

drive.mount('/content/gdrive')

print('Google Drive is mounted\n')

else:

print('Google Drive is already mounted\n')

import os

import tensorflow as tf

import numpy as np

from pandas.io.parsers import read\_csv

model = tf.global\_variables\_initializer()

if os.path.exists('/content/gdrive/My Drive/Colab Notebooks/price\_data\_yk.csv')==False:

print('Train data downloading..')

! curl 'https://raw.githubusercontent.com/unizard/2019.Spring.AI/master/price\_data\_yk.csv' -o '/content/gdrive/My Drive/Colab Notebooks/price\_data\_yk.csv'

print('Done..\n')

else:

print('File already exists \n')

# 학습데이터 로딩

data = read\_csv('/content/gdrive/My Drive/Colab Notebooks/price\_data\_yk.csv',sep=',')

xy = np.array(data,dtype=np.float32)

# 4개의 변인을 입력을 받습니다.

x\_data = xy[:, 1:-1]

# 가격 값을 입력 받습니다.

y\_data = xy[:, [-1]]

# 플레이스 홀더를 설정합니다.

X = tf.placeholder(tf.float32, shape=[None, 4])

Y = tf.placeholder(tf.float32, shape=[None, 1])

W = tf.Variable(tf.random\_normal([4, 1]), name="weight")

b = tf.Variable(tf.random\_normal([1]), name="bias")

# 가설을 설정합니다.

#

hypothesis = tf.matmul(X,W)+b

# 비용 함수를 설정합니다.

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

# 최적화 함수를 설정합니다.

optimizer = tf.train.GradientDescentOptimizer(learning\_rate=0.000005)

train = optimizer.minimize(cost)

# 세션을 생성합니다.

sess = tf.Session()

# 글로벌 변수를 초기화합니다.

sess.run(tf.global\_variables\_initializer())

# 학습을 수행합니다.

print('=========================================')

print(' 학습 시작')

print('=========================================')

for step in range(100001):

cost\_, hypo\_, \_ = sess.run([cost, hypothesis, train], feed\_dict={X: x\_data, Y: y\_data})

if step % 10000 == 0:

print("#", step, " 손실 비용: ", cost\_)

print("- 배추 가격: ", hypo\_[0])

print('=========================================')

print(' 학습 끝')

print('=========================================')

# 오늘의 날짜를 파일 이름으로 활용할 예정이었으나, 여러분의 혼란을 줄이고자 주석처리함

# 오늘의 날짜를 사용하기 위한 라이브러리 import

import datetime

# 학습 모델 저장을 위한 그래프 설정

saver = tf.train.Saver()

# 학습 모델 이름 설정

#strTLog = datetime.date.today().strftime("%Y%m%d")

path = '/content/gdrive/My Drive/Colab Notebooks/saved.cpkt'

# 학습 모델 저장

save\_path = saver.save(sess,path)

print('학습된 모델을 저장했습니다.')

배추 가격 예측 테스트 코드

from google.colab import drive

import os

if os.path.exists('/content/gdrive')==False:

drive.mount('/content/gdrive')

print('Google Drive is mounted\n')

else:

print('Google Drive is already mounted\n')

import tensorflow as tf

import numpy as np

# 플레이스 홀더를 설정합니다.

X = tf.placeholder(tf.float32, shape=[None, 4])

Y = tf.placeholder(tf.float32, shape=[None, 1])

W = tf.Variable(tf.random\_normal([4, 1]), name="weight")

b = tf.Variable(tf.random\_normal([1]), name="bias")

# 가설을 설정합니다.

hypothesis = tf.matmul(X,W)+b

# 4가지 변수를 입력 받습니다.

avg\_temp = float(input('평균 온도: '))

min\_temp = float(input('최저 온도: '))

max\_temp = float(input('최고 온도: '))

rain\_fall = float(input('강수량: '))

# 저장된 모델을 불러오는 객체를 선언합니다.

saver = tf.train.Saver()

model = tf.global\_variables\_initializer()

with tf.Session() as sess:

# 그래프 연결

sess.run(model)

# 학습 모델 로드

save\_path = '/content/gdrive/My Drive/Colab Notebooks/saved.cpkt'

saver.restore(sess, save\_path)

# 모델에 넣을 입력데이터 가공

data = ((avg\_temp, min\_temp, max\_temp, rain\_fall), (0, 0, 0, 0))

arr = np.array(data, dtype=np.float32)

x\_data = arr[0:4]

print(x\_data[0])

# 평균온도, 최대온도, 최저온도, 강수량으로 배추가격 예측을 위해 세션 실행

predict = sess.run(hypothesis, feed\_dict={X: x\_data})

print(predict[0])

#sess.close()

변수 입력 후 배추 가격예측 결과

