D:₩myprojects₩classify_num₩calculate_average.py

```
# 0~10 사이 정수 10개 평균내기
# conda install tensorflow-gpu keras pillow pydot matplotlib opencv
import random as rd
from tensorflow.keras.models import Sequential, save_model, load_model
from tensorflow.keras.layers import Dense
import sys

numOfIntegers = 10
X = []; Y = []; TEST = [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]]; Z = [[5.5]]

def trainNetwork() :
    for i in range(10000) :
        print(i,end=' ')
        tenRandomNumbers = []
        sum = 0.0
        for num in range(numOfIntegers) :
            rnd_num = rd.randint(0,10) # random integer number between 0 and 100
            sum += rnd_num
            tenRandomNumbers.append(rnd_num)
        correctAnswer = sum / numOfIntegers
        #print(tenRandomNumbers, "---", correctAnswer)
        X.append(tenRandomNumbers)
        Y.append(correctAnswer)
    print("학습데이터 X: ",X)
    print("학습데이터 Y: ",Y)
    # define the keras model
    model = Sequential()
    model.add(Dense(10, input_shape=(10,), activation='linear'))
    model.add(Dense(20, activation='linear'))
    model.add(Dense(20, activation='linear'))
    model.add(Dense(1, activation='linear'))
    # compile the keras model
    #opt = SGD(lr=0.01, momentum=0.9)
    model.compile(loss='mean_squared_error', metrics=['accuracy'])

    print(model.summary())

    # fit the keras model on the dataset
    model.fit(X, Y, epochs=1000, batch_size=10)

    save_model(model, 'average_model.tf')  # tensorflow 버전 2.x 용 형식으로 저장

    # evaluate the keras model
    _, accuracy = model.evaluate(X, Y)
    print('Training Set Accuracy: %.2f' % (accuracy*100))


def testNetwork() :
    model = load_model('average_model.tf')

    predictions = model.predict(TEST)
    print("결과: ", predictions, '\n')

    inputNumberString = input ("0~10 사이의 정수 10개 입력: ")
    inputStringList = inputNumberString.split(' ')
    inputNumberList = list(map(int, inputStringList))   # Convert to Int
```

```
    predictions = model.predict([inputNumberList])
    print("TEST 결과: ", predictions)


if __name__ == '__main__' :
    if len(sys.argv) > 1 and sys.argv[1] == 'train' :
        trainNetwork()
    elif len(sys.argv) > 1 and sys.argv[1] == 'test' :
        testNetwork()
    else :
        print('Usage: python calculate_average.py train/test')
```