10. 딥러닝 실습 예제 2 - 보스턴 집값 예측

이전 목차에선 이해를 위해 순서가 조금 뒤죽박죽이었습니다. 이번 목차에서는 딥러닝의 과정을 그대로 따라서 만들어가며 학습하도록 하겠습니다

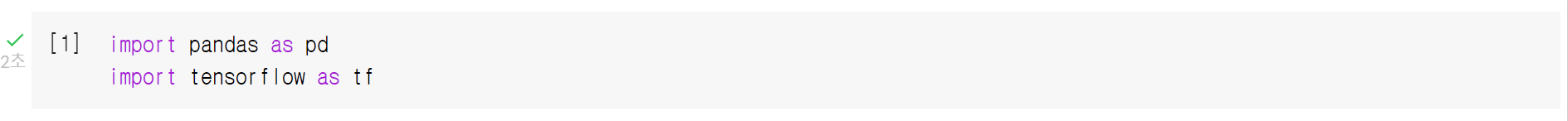
1) 데이터 준비

코랩에서 새 노트를 만들어 진행하도록 하겠습니다.

먼저 사용할 라이브러리를 불러와야 합니다.

import pandas as pd

import tensorflow as tf



이후엔 사용할 데이터를 불러와서 변수에 담아야겠군요.

Boston=pd.read\_csv('https://raw.githubusercontent.com/blackdew/tensorflow1/master/csv/boston.csv')

Boston # 제대로 담긴 것인가 확인합니다

테이블이(가) 표시된 사진

자동 생성된 설명

이젠 독립변수와 종속변수를 설정해주면 끝입니다.

input = Boston[['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis', 'rad', 'tax', 'ptratio', 'b', 'lstat']]

output = Boston[['medv']]

print(input.shape, output.shape) # 역시나 제대로 담겼는지 확인합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

input(독립변수)에는 506개의 row(행), 13개의 column(열)

output(종속변수)에는 506개의 row(행), 1개의 column(열)이 들어간 것이 맞나요? 이런 것을 한번 더 확인하는 습관이 나중에 문제가 커지지 않게 도와줍니다.

2) 모델 구조 생성

이제 모델 구조를 만들 차례입니다.

X = tf.keras.Input(shape=[13])

Y = tf.keras.layers.Dense(1)(X)

model = tf.keras.models.Model(X, Y)

model.compile(loss=’mse’)

텍스트이(가) 표시된 사진

자동 생성된 설명

어떤가요? 아까랑 굉장히 비슷하지 않나요? 바뀐 것은 X = tf.keras.Input(shape=[13]) 부분뿐입니다.

여기서 13은 입력 값 즉 독립변수의 층(종류, 개수)를 의미합니다. input에 Boston의 독립변수 13개를 할당했으니 이번엔 13이 된 것입니다.

그럼 만약 출력 값 즉 종속변수가 2가지라면 어떻게 될까요?

Y = tf.keras.layers.Dense(2)(X) 가 될 것입니다. 이제 슬슬 감이 오시나요?

3) 학습

이제 모델을 학습시킬 차례입니다.

model.fit(input, output, epochs=1000) # 일단 1000번만 학습시키겠습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

여기서 충분히 학습되었나는 어떻게 알까요? 기준이야 각자 세우기 나름이겠지만, 크게 2가지로 나누자면

1. loss의 값이 얼마나 크게 변하는가?
2. 모델이 만든 값과, 데이터의 값을 비교해서 얼마나 차이 나는가?

정도가 되겠습니다. 2번 항목을 사용하여 확인해보겠습니다.

4) 학습된 모델 검증

모델에 실제 데이터의 값을 넣어 예측을 해봅시다. 아까 1번 과정에서 입력 값인 독립변수들은 input에 저장해 놨으니 input의 0번부터 5개의 값을 넣어봅시다.

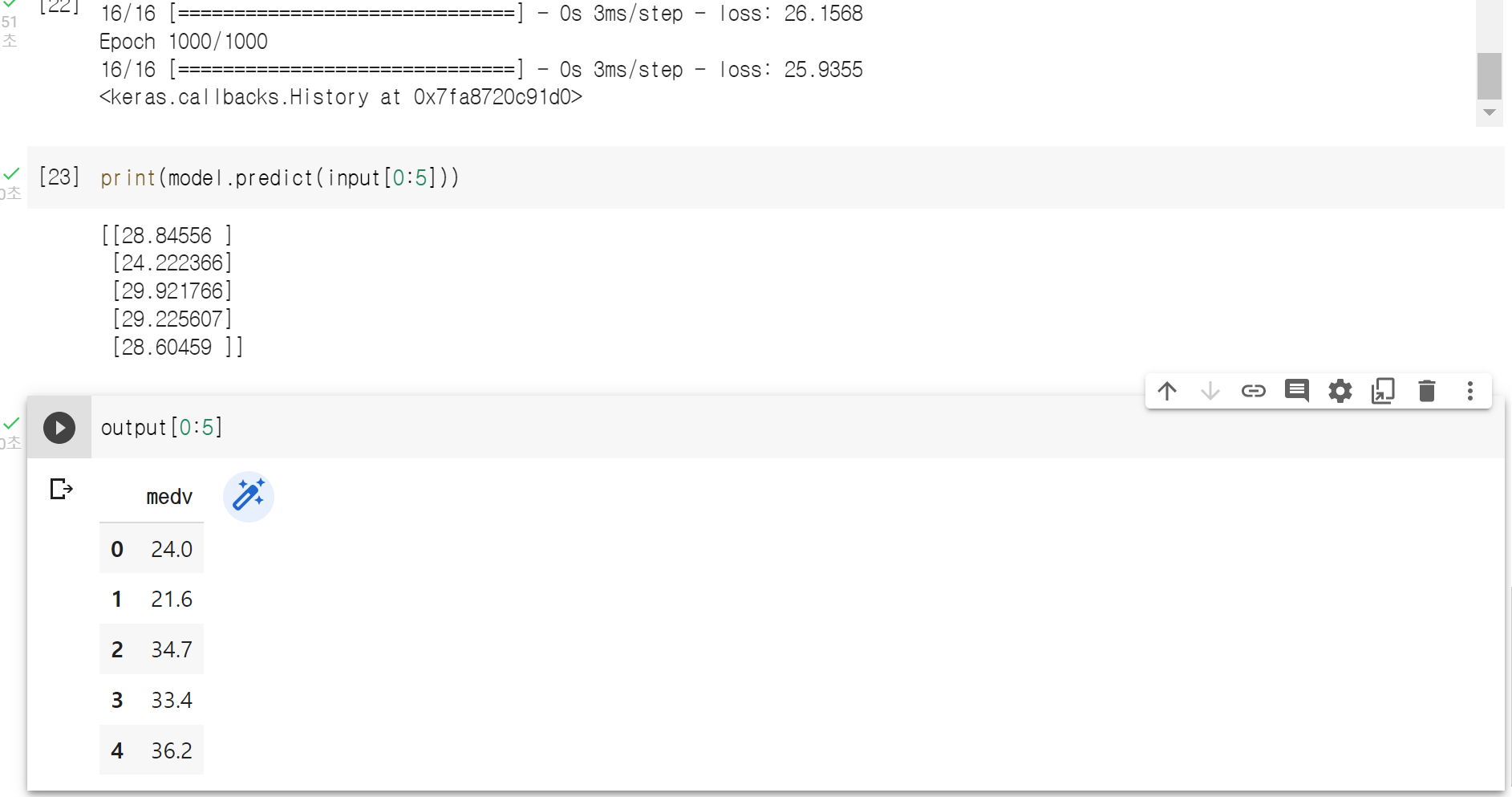
print(model.predict(input[0:5]))

테이블이(가) 표시된 사진

자동 생성된 설명

위와 같이 나왔습니다. 여러분들은 다른 값이 나올 것입니다.

그럼 이제 실제 데이터와 비교하기 위해 실제 값을 한번 볼까요? 아까 1번 과정에서 출력 값인 종속변수 medv는 output에 저장해 놨으니 output의 0번부터 5개의 값을 봅시다.



|  |  |  |
| --- | --- | --- |
|  | 모델이 예측한 값 | 실제 값 |
| 0번 | 28.84556 | 24.0 |
| 1번 | 24.222366 | 21.6 |
| 2번 | 29.921766 | 34.7 |
| 3번 | 29.225607 | 33.4 |
| 4번 | 28.60459 | 36.2 |

오차가 꽤 큰 것을 알 수 있습니다. 더 많이 학습시켜 보겠습니다. 아주 많이요.

텍스트이(가) 표시된 사진

자동 생성된 설명

총 약 45만번의 학습을 진행한 결과입니다. 별로 달라진 것이 없습니다.

우리는 이전에 인공지능은 얼마나 많은 데이터를 얼마나 오래 넣고 얼마나 많은 과정을 거치는지 보다

어떤 데이터를 넣을지, 얼마나 적정한 층과 뉴런을 설정할지를 고르는 것이 중요하다는 것을 배운 적이 있습니다. 이렇게 확인하게 되었네요.

이제 컴퓨터가 추론한 것을 한번 볼까요?

model.get\_weights()

텍스트이(가) 표시된 사진

자동 생성된 설명

이게 무슨 뜻일까요? 이것을 해석하기 위해서는 위로 돌아갈 필요가 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

여기서 crim, zn, indus, chas, … 을 입력 값으로 넣어 주었습니다. weights()의 결과로 나온 숫자들은 이 입력 값에 곱해주는 식으로 사용됩니다.

(crim \* -1.0596015e-01) + (zn \* 4.8147503e-02) + (indus \* 1.9814029e-02) + … + 36.72191

이것은 컴퓨터가 만들어낸 수식입니다. 정확하지는 않더라도 인간이 이런 공식을 쉽게 만들 수 있을까요?

5) 책을 마치며

이렇게 우리는 인공지능, 머신러닝, 딥러닝, 인공신경망에 대해 배우고 간단한 딥러닝 모델까지 직접 구현해봤습니다.

여러분들이 생각한 딥러닝은 고양이와 개 사진을 보여주면 구분하는 그런 모습이었는데 아니어서 실망하진 않으셨나요? 여러분들이 생각하신 그 딥러닝 모델들도 이미지를 숫자로 변환한 다음 학습하는 방법을 통해 거의 같은 과정을 거치게 됩니다. 즉 기본기는 완성된 상태라는 뜻이죠.

아직 여러분들은 파이썬 프로그래밍이나 여러 데이터를 다루는 것에 익숙하지 않을 수도 있기 때문에 이정도로 책은 마무리하지만, 여러분들은 딥러닝으로 따지자면 모델의 구조는 완성된 단계라고 볼 수 있습니다.

구조가 완성된 모델이 데이터를 통해 학습을 시작하듯 여러분들도 이제 경험을 통해 학습을 시작할 때입니다. 학습을 통해 여러분들만의 공식을 만들어 나가시길 바라며 이 책을 마무리하겠습니다.