

5-1. 텐서플로에서 실행하는 선형 회귀 모델

1. 환경 준비

```
import numpy as np
import matplotlib.pyplot as plt

# 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

2. 데이터 준비

```
x = np.array([2, 4, 6, 8])
y = np.array([81, 93, 91, 97])
```

3. 모델 실행

```
model = Sequential()

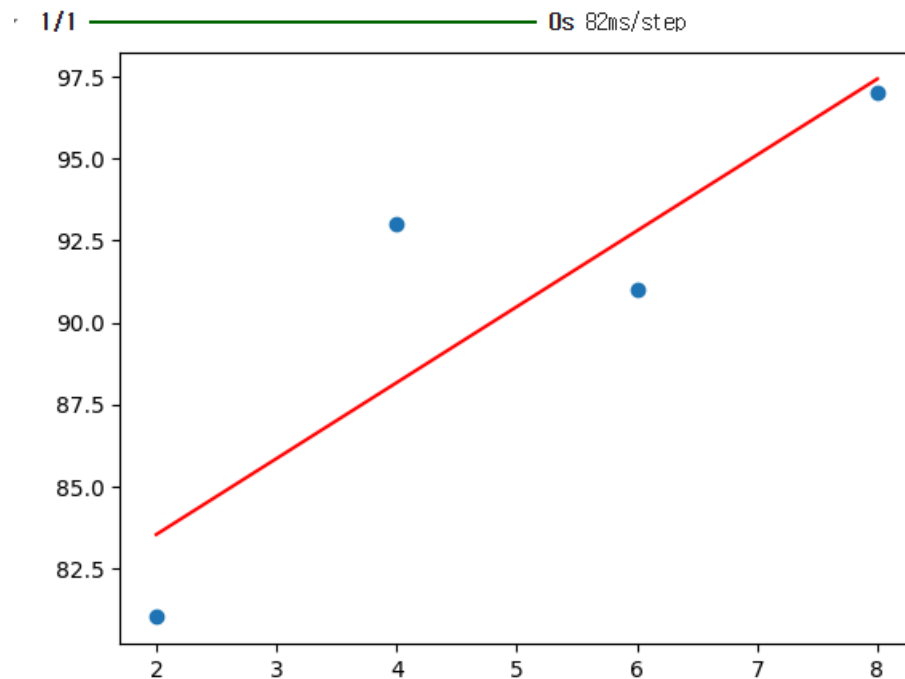
# 출력 값, 입력 변수, 분석 방법에 맞게끔 모델을 설정합니다.
model.add(Dense(1, input_dim=1, activation='linear'))

# 오차 수정을 위해 경사 하강법(sgd)을, 오차의 정도를 판단하기 위해 평균 제곱 오차(mse)를 사용합니다.
model.compile(optimizer='sgd', loss='mse')

# 오차를 최소화하는 과정을 2000번 반복합니다.
model.fit(x, y, epochs=2000)
```

4. 그래프로 확인

```
plt.scatter(x, y)
plt.plot(x, model.predict(x), 'r') # 예측 결과를 그래프로 나타냅니다.
plt.show()
```



5. 모델 테스트

```
] import tensorflow as tf

# 임의의 시간을 집어넣어 점수를 예측하는 모델을 테스트해 보겠습니다.

hour = 7
#prediction = model.predict([hour]) #텐서플로 2.9.0 까지만 가능

input_data = tf.constant([[hour]]) # 텐서플로 버전업으로 인한 수정
prediction = model.predict(input_data)[0][0] # 텐서플로 버전업으로 인한 수정

print("%.f시간을 공부할 경우의 예상 점수는 %.02f점입니다" % (hour, prediction))
```

➤ 7시간을 공부할 경우의 예상 점수는 95.12점입니다

5-2 텐서플로에서 실행하는 다중 선형 회귀 모델

1. 환경 준비

```
] import numpy as np
import matplotlib.pyplot as plt

# 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

2. 데이터 준비

```
] x = np.array([[2, 0], [4, 4], [6, 2], [8, 3]])
y = np.array([81, 93, 91, 97])
```

3. 모델 실행

```
] model = Sequential()

# 입력 변수가 2개(학습 시간, 과외 시간)이므로 input_dim에 2를 입력합니다.
model.add(Dense(1, input_dim=2, activation='linear'))
model.compile(optimizer='sgd', loss='mse')

model.fit(x, y, epochs=2000)
```

4. 모델 테스트

```
import tensorflow as tf

# 임의의 학습 시간과 과외 시간을 집어넣어 점수를 예측하는 모델을 테스트해 보겠습니다.

hour = 7
private_class = 4
#prediction = model.predict([[hour, private_class]]) #텐서플로 2.9.0 까지만 가능

input_data = tf.constant([[hour, private_class]]) # 텐서플로 버전업으로 인한 수정
prediction = model.predict(input_data)[0][0] # 텐서플로 버전업으로 인한 수정

print("%.f시간을 공부하고 %.f시간의 과외를 받을 경우, 예상 점수는 %.02f점입니다" % (hour, private_class, prediction))
```

· 7시간을 공부하고 4시간의 과외를 받을 경우, 예상 점수는 97.53점입니다

5. 텐서플로에서 실행하는 로지스틱 회귀 모델

1. 환경 준비

```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

2. 데이터 준비

```
x = np.array([2, 4, 6, 8, 10, 12, 14])
y = np.array([0, 0, 0, 1, 1, 1, 1])
```

3. 모델 실행

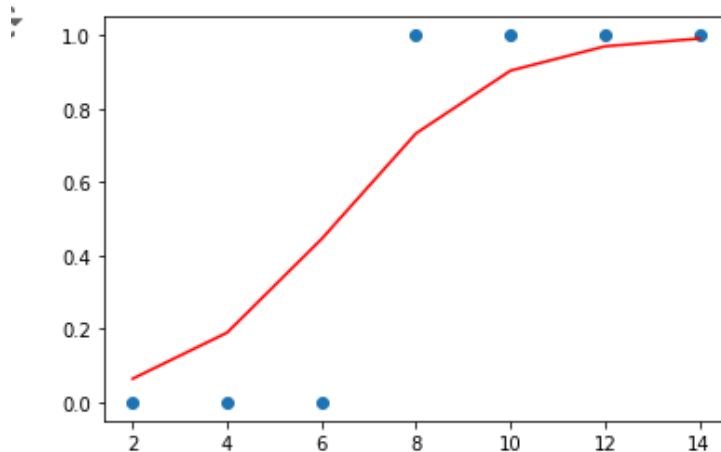
```
model = Sequential()
model.add(Dense(1, input_dim=1, activation='sigmoid'))

# 교차 엔트로피 오차 함수를 이용하기 위하여 'binary_crossentropy'로 설정합니다.
model.compile(optimizer='sgd', loss='binary_crossentropy')
model.fit(x, y, epochs=5000, verbose=0) ) #verbose=1 진행 화면 출력, verbose=0 진행 화면 출력 안함.
```

4. 모델 테스트

```
] # 그래프로 확인해 봅니다.
```

```
plt.scatter(x, y)
plt.plot(x, model.predict(x), 'r')
plt.show()
```



```
) import tensorflow as tf
```

```
# 임의의 학습 시간을 집어넣어 합격 예상 확률을 예측해 보겠습니다.
```

```
hour = 7
```

```
#prediction = model.predict([hour]) #텐서플로 2.9.0 까지만 가능
```

```
input_data = tf.constant([[hour]])
```

```
prediction = model.predict(input_data)[0][0]
```

```
print("%.f시간을 공부할 경우, 합격 예상 확률은 %.01f%%입니다" % (hour, prediction * 100))
```

➤ 7시간을 공부할 경우, 합격 예상 확률은 59.7%입니다