5주차 실습과제 코드

2020142001 곽종근
제출일: 2024.04.15.

실습과제 3.

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# 데이터 불러오기
fold_dir = "C:\\Users\\user\\OneDrive - 한국공학대학교\\바탕 화면\\3학년
1학기\\머신러닝실습\\Machine-Learning\\6주차\\lin_regression_data_02.csv"
temp_data = pd.read_csv(fold_dir)
temp_data = temp_data.to_numpy()

# 데이터 분리
x_0 = temp_data[:, 0]
x_1 = temp_data[:, 1]
y = temp_data[:, 2].reshape(-1, 1) #여기서 리쉐입은 50,1이라는 형식을 만들어주려고 한거지 트렌스포즈가 아님.

# 더미 데이터 추가
dummy_data = np.ones((len(temp_data), 1))

# 기존 x 데이터와 더미 데이터를 수직으로 결합하여 새로운 배열 생성
x_with_dummy = np.hstack((temp_data[:, :2], dummy_data))  #여기는 알고보니 트렌스포즈 왜? 교제에 보면
M은 차원 수 , n은 인덱스

#결과적으로 이 코드에서 구한 것들은 모든 값들의 트렌스포즈라고 볼 수 있다.

# 초기 가중치 랜덤 설정
w_ = np.random.rand(3, 1) * 6

# 경사 하강법 함수 정의
def gradient_descent(X, y, w, alpha, rp):
    w0_change = []  # w0 변화 저장
    w1_change = []  # w1 변화 저장
    w2_change = []  # w2 변화 저장
    mse_change = []  # MSE 변화 저장

    for i in range(rp):

        y_hat = np.dot(X, w)   #x는 기존 데이터+ 더미, w는 랜덤 3개 , 이게 알고보니까 트렌스포즈다, 1xN이
나오도록 수정
        error = y_hat - y
```

```
        mse = np.mean(error ** 2)
        w -= alpha * np.dot(X.T, error) / len(y)  # 경사 하강법 업데이트하는 부분

        # w0, w1, w2, MSE 값을 저장하는데 리스트로 저장하고, 각 0번째w, 1번째w, 2번째w를 append해서 저장한다.
        w0_change.append(w[0][0])
        w1_change.append(w[1][0])
        w2_change.append(w[2][0])
        mse_change.append(mse)

    return w0_change, w1_change, w2_change, mse_change

# 경사 하강법 실행
w0_change, w1_change, w2_change, mse_change = gradient_descent(x_with_dummy, y, w_, 0.1, 100)

# 그래프 그리기
fig = plt.figure(figsize=(20, 10))

# w 그래프
ax1 = fig.add_subplot(121)
ax1.plot(w0_change, label='w0')
ax1.plot(w1_change, label='w1')
ax1.plot(w2_change, label='w2')
ax1.grid()
ax1.legend()
ax1.set_xlabel('Epochs')
ax1.set_ylabel('Values')




ax1.set_title('Weights')

# mse 그래프
ax2 = fig.add_subplot(122)
ax2.plot(mse_change, label='MSE')
ax2.legend()
ax2.set_xlabel('Epochs')
ax2.set_ylabel('MSE')
ax2.set_title('Mean Squared Error')
ax2.grid()

# 3차원 그래프
fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x_0, x_1, y, c='blue', label='Original Data')
```

```
# 예측된 y^ 값을 계산
y_hat_surface = np.dot(x_with_dummy, w_).reshape(x_0.shape)

# 예측값 점으로 표시
ax.scatter(x_0, x_1, y_hat_surface, c='red', label='Predicted Data')   #위에서 구한 y^에 대한 값을 점으로 표시

# 가중치 평면 그리기
x0_v, x1_v = np.meshgrid(np.linspace(x_0.min() - 1, x_0.max() + 1, 100), np.linspace(x_1.min() - 1,
x_1.max() + 1, 100)) #각 x0,x1을 meshgrid하여 2차원으로 선언하고
y_hat_surface = w_[0][0] * x0_v + w_[1][0] * x1_v + w_[2][0]  #위에서 선언한 것들을 이용해 y를 선언
ax.plot_surface(x0_v, x1_v, y_hat_surface, alpha=0.5, color='green', label='Weight Plane') #최종적으로 각 축
3개의 범위를 선언하여주었다.


# 축 레이블 설정
ax.set_xlabel('x_0')
ax.set_ylabel('x_1')
ax.set_zlabel('y')
ax.legend(['Original Data', 'Predicted Data']) #그냥 legend라고 선언하니 값이 계속 뜨지 않아서
https://blog.naver.com/inna1225/222299957057 참조
plt.show()
```