



전자공학과 2020142001 곽종근



5주차 다차원 데이터 경사하강법 실습과제3

제출일: 2024.04.15.

실습 과제 3. 코드

```
1 from mpl_toolkits.mplot3d import Axes3D
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import pandas as pd
5
6 # 데이터 불러오기
7 fold_dir = "C:\\Users\\user\\OneDrive - 한국공학대학교\\바탕 화면\\3학년 1학기\\머신러닝실습\\Machine-Learning\\6주차\\lin_regression_data_02.csv"
8 temp_data = pd.read_csv(fold_dir)
9 temp_data = temp_data.to_numpy()
10
11 # 데이터 분리
12 x_0 = temp_data[:, 0]
13 x_1 = temp_data[:, 1]
14 y = temp_data[:, 2].reshape(-1, 1) #여기서 리쉐입은 50,1이라는 형식을 만들어주려고 한거지 트랜스포즈가 아님.
15
16 # 더미 데이터 추가
17 dummy_data = np.ones((len(temp_data), 1))
18
19 # 기존 x 데이터와 더미 데이터를 수직으로 결합하여 새로운 배열 생성
20 x_with_dummy = np.hstack((temp_data[:, :2], dummy_data)) #여기는 알고보니 트랜스포즈 왜? 과제에 보면 m은 차원 수 , n은 인덱스
21
22 # 초기 가중치 랜덤 설정
23 w_ = np.random.rand(3, 1) * 6
24
25 # 경사 하강법 함수 정의
26 def gradient_descent(X, y, w, alpha, rp):
27     w0_history = [] # w0 변화 저장
28     w1_history = [] # w1 변화 저장
29     w2_history = [] # w2 변화 저장
30     mse_history = [] # MSE 변화 저장
31
32     for i in range(rp):
33
34         y_hat = np.dot(X, w) #x는 기존 데이터+ 더미, w는 랜덤 3개 , 이게 알고보니까 트랜스포즈다, 1xN이 나오도록 수정
35         error = y_hat - y
36         mse = np.mean(error ** 2)
37         w -= alpha * np.dot(X.T, error) / len(y) # 경사 하강법 업데이트
38
39         # w0, w1, w2, MSE 값을 저장
40         w0_history.append(w[0][0])
41         w1_history.append(w[1][0])
42         w2_history.append(w[2][0])
43         mse_history.append(mse)
44
45     return w0_history, w1_history, w2_history, mse_history
46
47 # 경사 하강법 실행
48 w0_history, w1_history, w2_history, mse_history = gradient_descent(x_with_dummy, y, w_, 0.1, 100)
49
50
51
```

실습 과제 3. 코드

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

# w 그래프
ax1 = fig.add_subplot(121)
ax1.plot(w0_history, label='w0')
ax1.plot(w1_history, label='w1')
ax1.plot(w2_history, 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_history, 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')

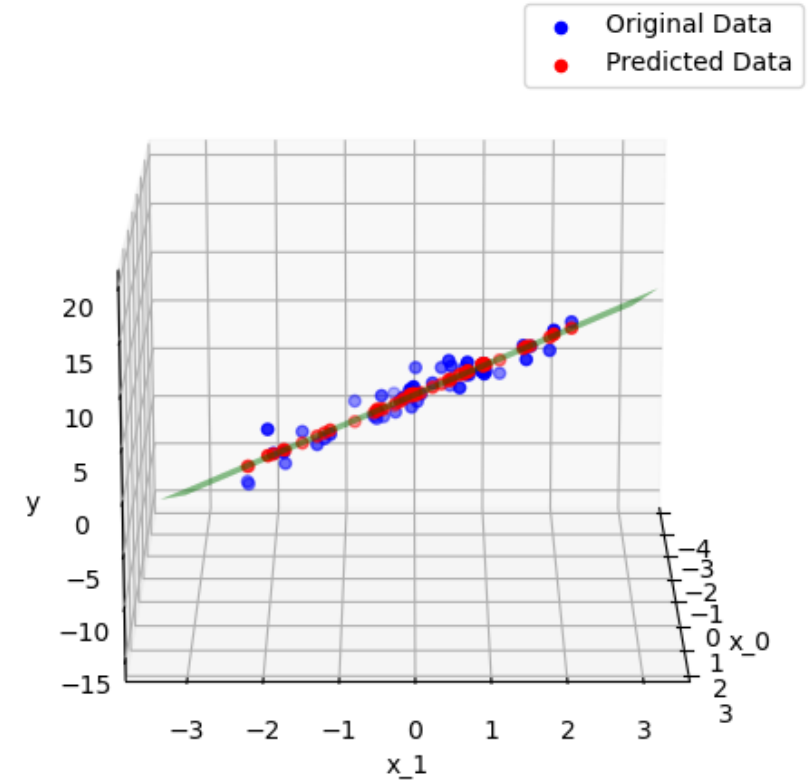
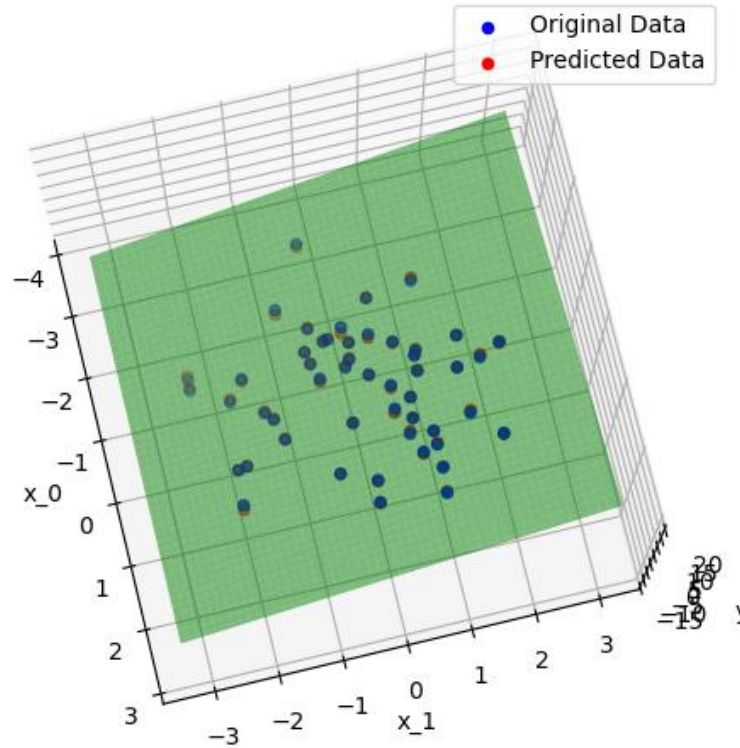
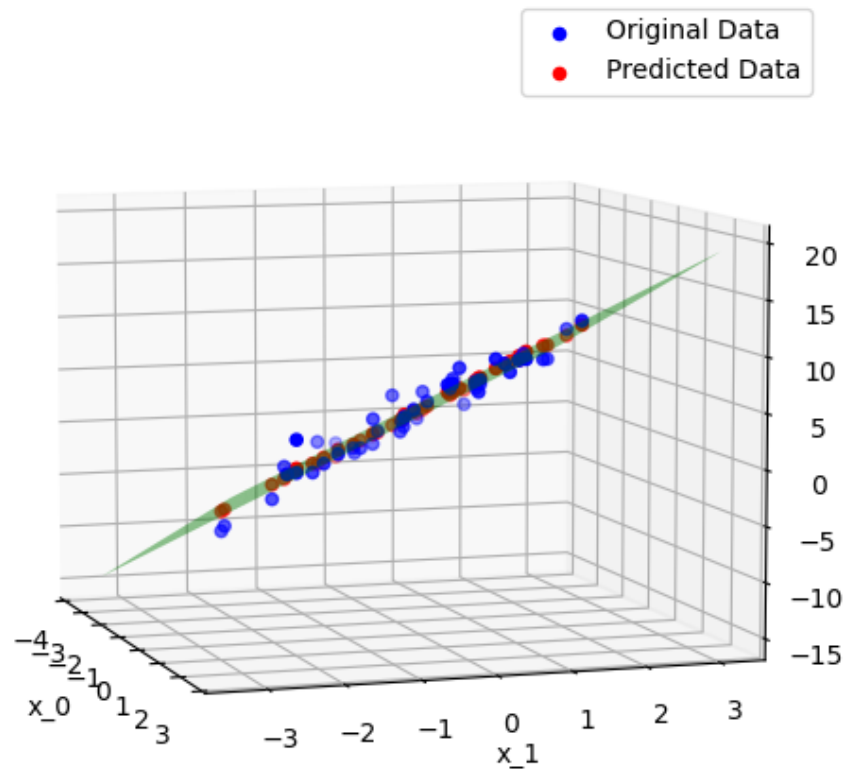
# 예측된 y^ 값을 계산
y_hat_surface = np.dot(x_with_dummy, w_).reshape(x_0.shape)

# 예측값 점으로 표시
ax.scatter(x_0, x_1, y_hat_surface, c='red', label='Predicted Data')

# 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))
y_hat_surface = w_[0][0] * x0_v + w_[1][0] * x1_v + w_[2][0]
ax.plot_surface(x0_v, x1_v, y_hat_surface, alpha=0.5, color='green', label='Weight Plane')

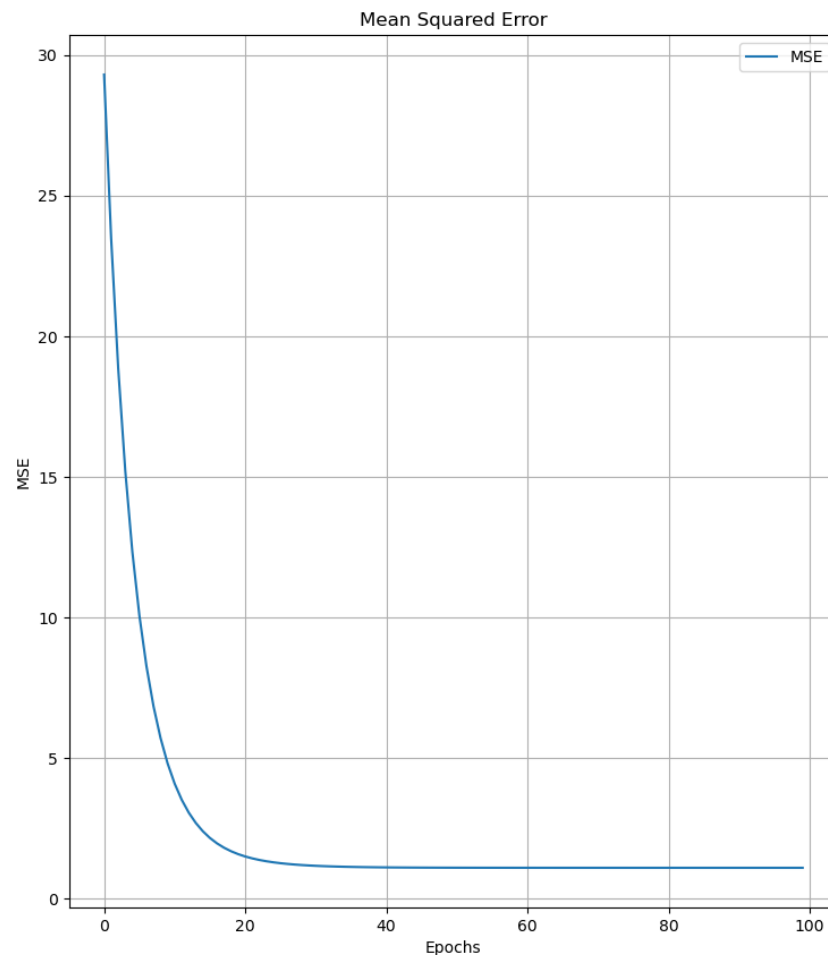
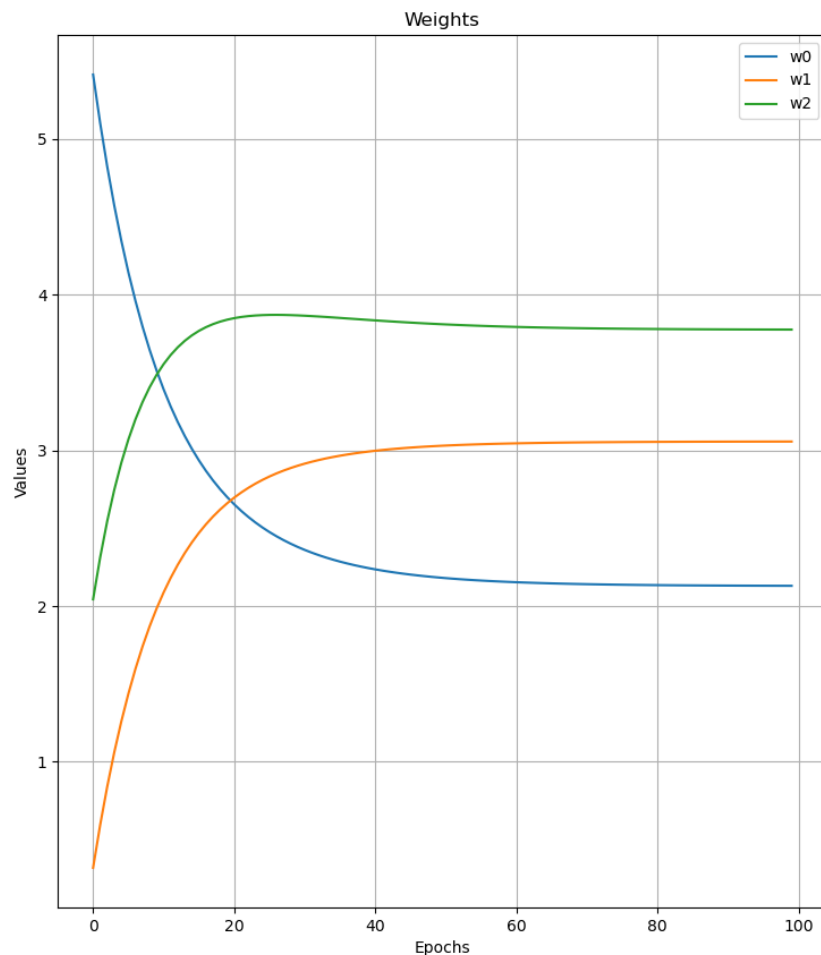
# 축 레이블 설정
ax.set_xlabel('x_0')
ax.set_ylabel('x_1')
ax.set_zlabel('y')
ax.legend(['Original Data', 'Predicted Data'])
plt.show()
```

실습 과제 3. 코드



이 값을보면 위에서 바라봤을 때 예측값과 실제 값은 똑같은 위치에 있음을 알 수 있지만 측면에서 보면 그 값들은 optimal solution을 이용한 평면 위에 존재함을 알 수 있다.

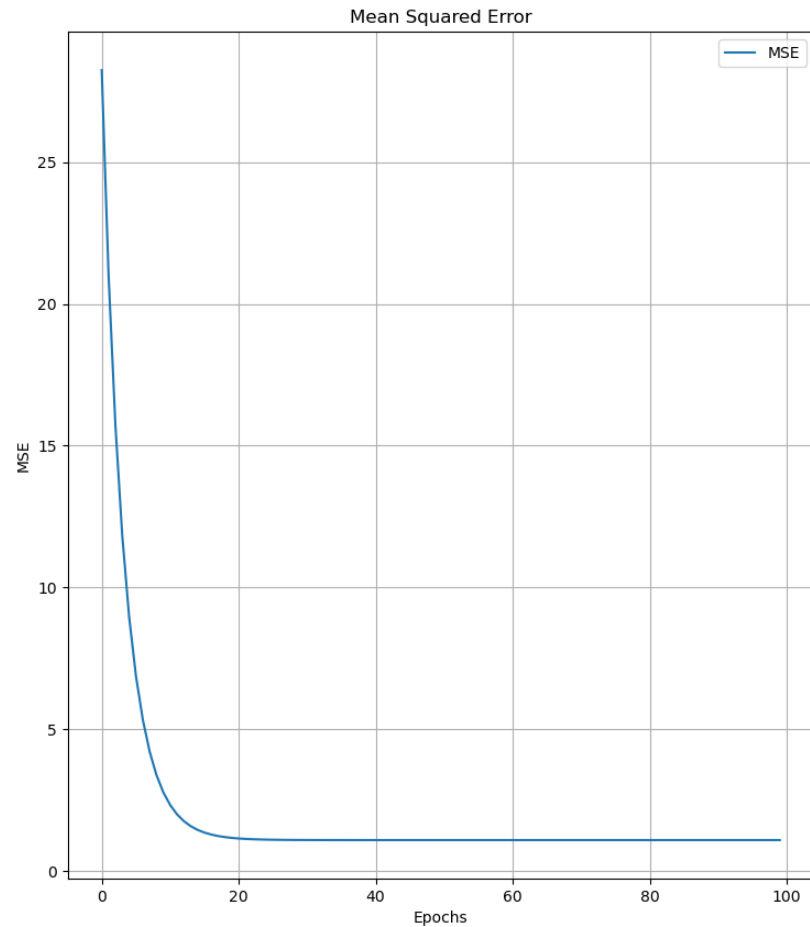
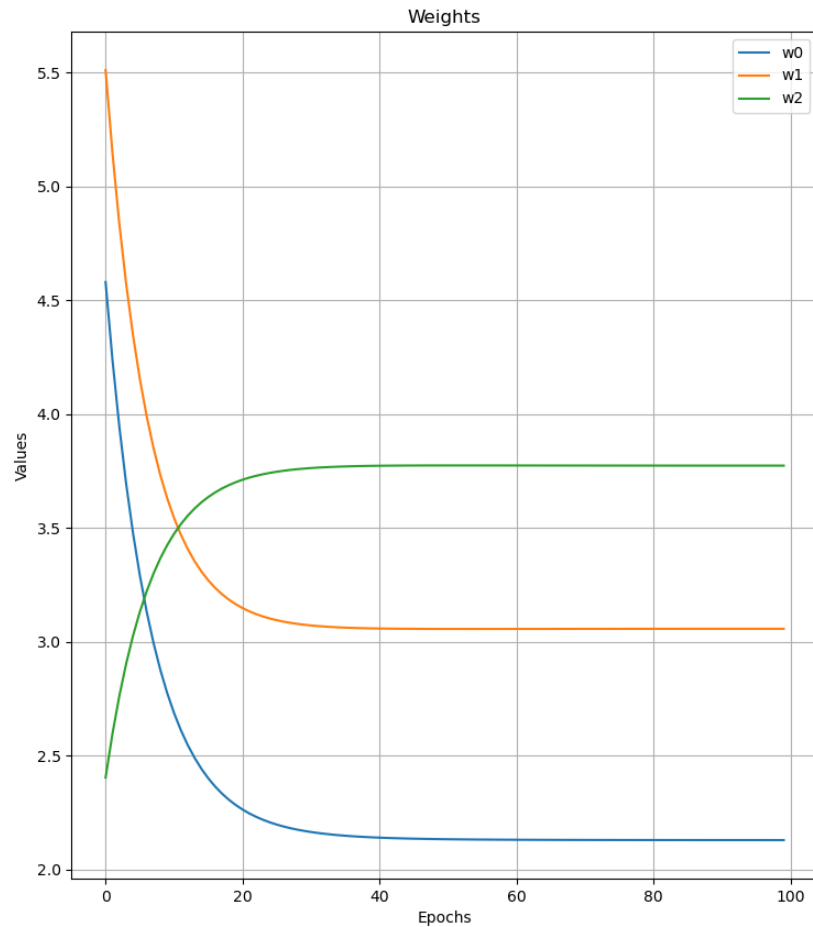
실습 과제 3. 코드



Name	Type	Size	Value
ax	mplot3d.axes3d.Axes3D	1	Axes3D object of
ax1	axes._axes.Axes	1	Axes object of
ax2	axes._axes.Axes	1	Axes object of
dummy_data	Array of float64	(50, 1)	[[1.] [1.]
fig	figure.Figure	1	Figure object of
fold_dir	str	101	C:\Users\user\
mse_history	list	100	[29.3118329812
temp_data	Array of float64	(50, 3)	[[-0.121163 [0.223035 -
w0_history	list	100	[5.41182623756
w1_history	list	100	[0.31977475547
w2_history	list	100	[2.04351230178
w_	Array of float64	(3, 1)	[[2.13004723] [3.05613967]
x0_v	Array of float64	(100, 100)	[[-3.77267 2 ...
x1_v	Array of float64	(100, 100)	[[-3.33724 -3.33 ...
x_0	Array of float64	(50,)	[-0.121163 0.]
x_1	Array of float64	(50,)	[0.497399 -0.]
x_with_dummy	Array of float64	(50, 3)	[[-0.121163 [0.223035 -
y	Array of float64	(50, 1)	[[6.28955] [3.94392]

코드를 보면 w값 3개가 랜덤으로 설정되면 이 w값의 그래프가 a값에 따라 그래프 모양이 전혀 달라짐을 알 수 있고, mse그래프의 곡률이 달라지는 것을 볼 수 있다. 알파는 0.1, 반복수는 100이다.

실습 과제 3. 코드



ax	mplot3d.axes3d.Axes3D	1	Axes3D object
ax1	axes._axes.Axes	1	Axes object of
ax2	axes._axes.Axes	1	Axes object of
dummy_data	Array of float64	(50, 1)	[[1.] [1.]
fig	figure.Figure	1	Figure object
fold_dir	str	101	C:\Users\user\
mse_history	list	100	[28.2429819886
temp_data	Array of float64	(50, 3)	[[-0.121163 [0.223035
w0_history	list	100	[4.58005658997
w1_history	list	100	[5.51137315836
w2_history	list	100	[2.40316014547
w_	Array of float64	(3, 1)	[[2.12857738] [3.05681657]
x0_v	Array of float64	(100, 100)	[[-3.77267 2 ...
x1_v	Array of float64	(100, 100)	[[-3.33724 -3.33 ...
x_0	Array of float64	(50,)	[-0.121163 0.
x_1	Array of float64	(50,)	[0.497399 -6
x_with_dummy	Array of float64	(50, 3)	[[-0.121163 [0.223035
y	Array of float64	(50, 1)	[[6.28955] [3.94392]
...

코드를 보면 w값 3개가 랜덤으로 설정되면 이 w값의 그래프가 a값에 따라 그래프 모양이 전혀 달라짐을 알 수 있고, mse그래프의 곡률이 달라지는 것을 볼 수 있다. 알파는 0.1, 반복수는 100이다.