HW0partB

September 22, 2022

[280]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

[281]:
```
dataset = pd.read_csv('D3.csv')
dataset.head()
```

[281]:

	X1	X2	X3	Y
0	0.000000	3.440000	0.440000	4.387545
1	0.040404	0.134949	0.888485	2.679650
2	0.080808	0.829899	1.336970	2.968490
3	0.121212	1.524848	1.785455	3.254065
4	0.161616	2.219798	2.233939	3.536375

[282]:
```
x1 = dataset.values[:,0]
x2 = dataset.values[:,1]
x3 = dataset.values[:,2]
y = dataset.values[:, 3]
m = len(y)

print('x1 = ', x1[: 5])
print('x2 = ', x2[: 5])
print('m = ', m)
```

```
x1 =  [0.         0.04040404 0.08080808 0.12121212 0.16161616]
x2 =  [3.44       0.1349495  0.82989899 1.52484848 2.21979798]
m =  100
```

[283]:
```
x0 = np.ones((m, 1))
x0[:5]
```

[283]:
```
array([[1.],
       [1.],
       [1.],
       [1.],
       [1.]])
```

[284]:
```
X1 = x1.reshape(m,1)
X2 = x2.reshape(m,1)
X3 = x3.reshape(m,1)

X1[:5]
```

[284]:
```
array([[0.        ],
       [0.04040404],
       [0.08080808],
       [0.12121212],
       [0.16161616]])
```

[285]:
```
x1 = np.hstack((x0, X1))
x2 = np.hstack((x0, X2))
x3 = np.hstack((x0, X3))
x4 = np.hstack((x0, X1, X2, X3))
x4[:5]
```

[285]:
```
array([[1.        , 0.        , 3.44      , 0.44      ],
       [1.        , 0.04040404, 0.1349495 , 0.88848485],
       [1.        , 0.08080808, 0.82989899, 1.3369697 ],
       [1.        , 0.12121212, 1.52484848, 1.78545454],
       [1.        , 0.16161616, 2.21979798, 2.23393939]])
```

[286]:
```
theta = np.zeros(4)
theta
```

[286]:
```
array([0., 0., 0., 0.])
```

[287]:
```
def get_loss(x4, y, theta):

    H = x4.dot(theta)
    error = np.subtract(H, y)
    sqrError = np.square(error)
    J = 1 / (2 * m) * np.sum(sqrError)

    return J
```

[288]:
```
def gradient_descent(x4, y, theta, alpha, iterations):

    cost_history = np.zeros(iterations)

    for i in range(iterations):
        H = x4.dot(theta)
        error = np.subtract(H, y)
        sum_delta = (alpha / m) * x4.transpose().dot(error);
        theta = theta - sum_delta;
        cost_history[i] = get_loss(x4, y, theta)
```

```
    return theta, cost_history
```

[315]:
```
theta = np.zeros(4)
iterations = 3000;
alpha = 0.05;
```

[316]:
```
theta, cost_history = gradient_descent(x4, y, theta, alpha, iterations)

print(theta)
print(cost_history)
```

```
[ 5.31416557 -2.00371904  0.5325636  -0.26560163]
[4.35632837 3.99754934 3.73994659 … 0.73846424 0.73846424 0.73846424]
```

[317]:
```
J_cost = get_loss(x4, y, theta)

print(J_cost)
```

```
0.7384642415684312
```

[318]:
```
h1 = theta[0] + (1)*theta[1] + (1)*theta[2] + (1)*theta[3]
h1
```

[318]:
```
3.5774084969375672
```

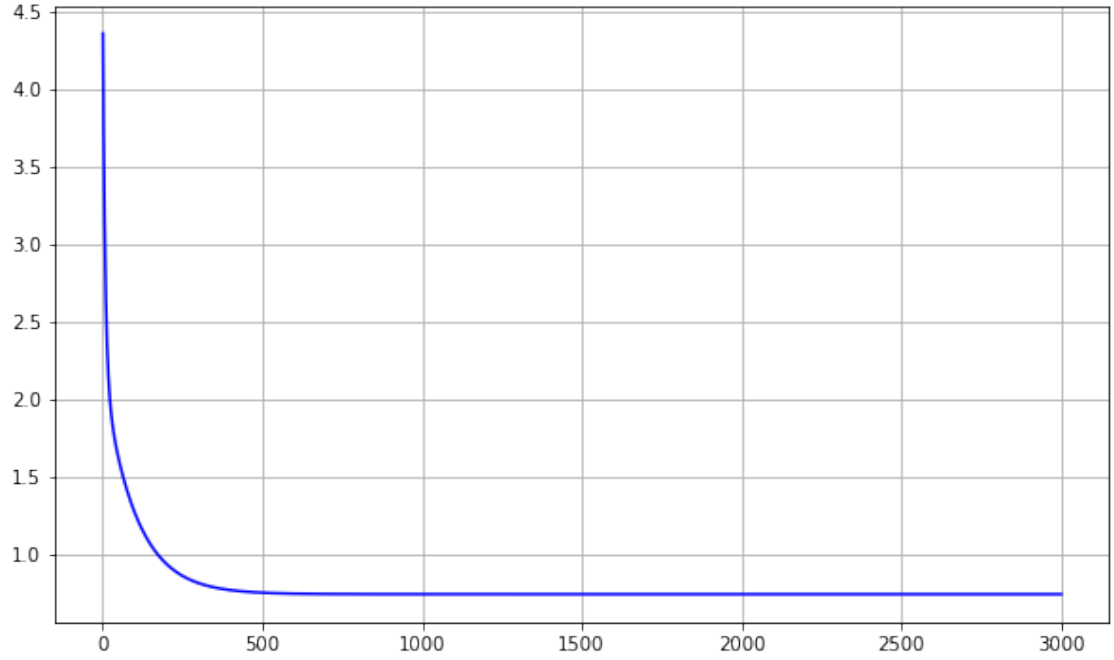
[319]:
```
h2 = theta[0] + (2)*theta[1] + (0)*theta[2] + (3)*theta[3]
h2
```

[319]:
```
0.5099225928223513
```

[320]:
```
h3 = theta[0] + (3)*theta[1] + (2)*theta[2] + (1)*theta[3]
h3
```

[320]:
```
0.1025340138616545
```

[321]:
```
plt.plot(range(1, iterations + 1), cost_history, color = 'blue')
plt.rcParams["figure.figsize"] = (10, 6)
plt.grid()
```


4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0
500
1000
1500
2000
2500
3000

[ ]: