Gradient Descent

Abdullah Zameek (arz268)

Q1.

Problem 1

① $$J(w) = \frac{1}{2}m \sum_{i=1}^{m} \left[y^{(i)} - f_w(x^{(i)}) \right]^2$$

$$= \frac{1}{8} \Big[(330 - w_0 - 1600w_1 - 1770w_2 - 3w_3)^2 + (369 - w_0 - 2400w_1 - 2740w_2 - 3w_3)^2 + (232 - w_0 - 1416w_1 - 1634w_2 - 2w_3)^2 + (540 - w_0 - 3000w_1 - 3412w_2 - 4w_3)^2 \Big]$$

$$= \frac{590485}{8} - \frac{4721w_3}{4} + \frac{19w_3^2}{4} - \frac{1471w_0}{4} + 3 w_0 w_3$$

$$+ \frac{w_0^2}{2} - 840528w_1 + 6708 w_3 w_1 + 2104 w_0 w_1$$

$$+ 2415632 w_1^2 - 954182 w_2 + \frac{15223 w_2 w_3}{2}$$

$$\cancel{+ 15223 w_3 w} + 2389 w_0 w_2 + 5489436 w_1 w_2$$

$$+ 3119025 w_2^2$$

2. $w_0 = 36.775$, $w_1 = 84053$, $w_2 = 95418$, $w_3 = 118.025$

Q2.

Problem 2

A. Feature Normalization $\quad S = \{x_1 \ldots x_m\}$

a) Mean $\Rightarrow \mu = \sum_{i=1}^{m} x_i / m \quad \mu \Rightarrow$ mean

~~b)~~ For n numbers, $\mu = \frac{\sum x}{n}$

b) Standard Deviation $\Rightarrow \sigma = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2}$

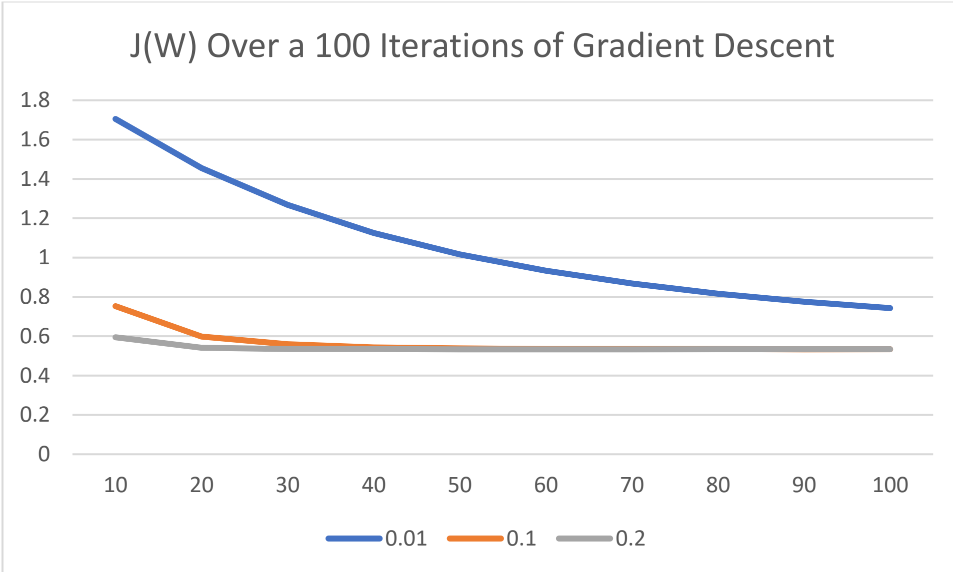
B.

a) Loss function :- $J(w) = \frac{1}{2}m \sum_{i=1}^{m} \left[y^{(i)} - f_w\left(x^{(i)}\right) \right]^2$

where $f_w(x) = W_0 + W_1 x_1 + \ldots W_n x_n$

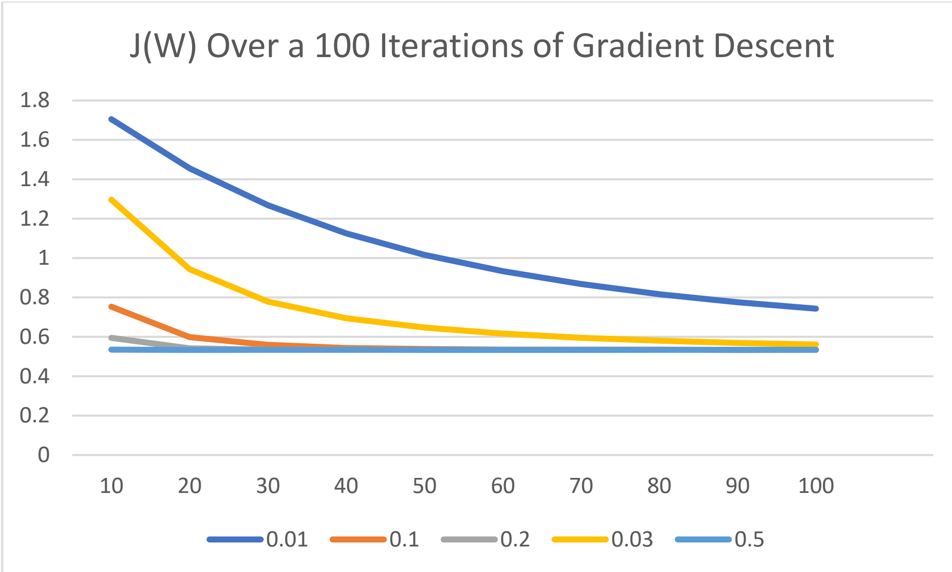
B.

The following graph was obtained for learning rates of 0.01, 0.1, and 0.2 iterations.



	0.01	0.1	0.2
10	1.705103	0.75306	0.594852
20	1.455809	0.598991	0.542142
30	1.268001	0.55879	0.535238
40	1.12561	0.543818	0.534269
50	1.016851	0.537947	0.534132
60	0.933075	0.535628	0.534113
70	0.867924	0.534711	0.53411
80	0.81672	0.534348	0.53411
90	0.776016	0.534204	0.53411
100	0.743264	0.534147	0.53411

The following graph was obtained for learning rates of 0.03 and 0.5



	0.03	0.5
10	1.296449	0.535202
20	0.942947	0.534113
30	0.779125	0.53411
40	0.695167	0.53411
50	0.647016	0.53411
60	0.616441	0.53411
70	0.595498	0.53411
80	0.580435	0.53411
90	0.569291	0.53411
100	0.560917	0.53411

Since it can be clearly seen that the graph for a learning rate of 0.5 converges faster than the rest, a learning rate of 0.5 is optimal for this setting.

C.

```
The predicted price for x= [2650,4] is  423554.11924019444
```

The predicted price for the given value is approx. 423,554

D.

```
C:\Users\Abdullah Zameek\Desktop\Machine Learning\Assignments\Gradient Descent>python gradientDescent.py
step:  1 loss function is currently:  0.6376399924081524
step:  2 loss function is currently:  0.5510882575872711
step:  3 loss function is currently:  0.5366219705048508
```

The output of the loss function seems to converge faster in the stochastic gradient descent algorithm, compared to the regular gradient descent algorithm in much fewer steps for the same learning rate of 0.05. (3 versus 100 steps)

```
step:  10 loss function is currently:  1.0460080869878303
step:  20 loss function is currently:  0.7473442958365262
step:  30 loss function is currently:  0.6466742654896991
step:  40 loss function is currently:  0.6009343054623913
step:  50 loss function is currently:  0.5756218548843677
step:  60 loss function is currently:  0.5603117996306402
step:  70 loss function is currently:  0.5507397998000471
step:  80 loss function is currently:  0.5446853769684881
step:  90 loss function is currently:  0.5408401521477201
step:  100 loss function is currently:  0.5383943298923366
```

The output for gradient decent with upto 100 steps seems to be similar to the output for the SGD, except for the fact that the SGD algorithm reached that value much faster.

Problem 3.

Problem 3

1. $$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(f_w(x^{(i)} - y^{(i)}\right)^2 \cdot \cancel{x_j^{(i)}} + \lambda \sum_{j=1}^{n} w_j^2\right]$$

$$= \frac{1}{2m}\left[2\sum_{i=1}^{m}\left(f_w(x^{(i)} - y^{(i)}\right) \cdot x_j^{(i)} + \lambda \sum_{j=1}^{n} 2\right]$$

$$= \frac{1}{m}\left[\sum_{i=1}^{m}\left(f_w(x^{(i)} - y^{(i)}\right) \cdot x_j^{(i)} + \lambda n\right]$$

$$W_j = W_j - \frac{\alpha}{m}\left[\sum_{i=1}^{m}\left(f_w(x^{(i)} - y^{(i)}\right) x_j^{(i)} + \lambda n\right]$$

2. If $\forall x_1, x_2, \forall c \in [0,1]$

$$f(cx_1 + (1-c)x_2) \le cf(x_1) + (1-c)f(x_2)$$

If the above is the case, then f is convex

a) $W = \{w_1, w_2\}$ $\quad f_{reg}(w) = \lambda(w_1^2 + w_2^2)$

Prove $f_{reg}(w)$ is convex

(A) $f(cw + (1-c)w') = \lambda((cw_1 + (1-c)w_1')^2 + (cw_2 + (1-c)w_2')^2)$

(B) $cf(w) + (1-c)f(w') = c\lambda(w_1^2 + w_2^2) + (1-c)\lambda(w_1'^2 + w_2'^2)$

Prove (B) $\ge$ (A)

Expanding (A) gives

$= \lambda(c^2w_1^2 + (1-c)^2 w_1'^2 + 2c(1-c)w_1 w_R + c^2 w_2^2 + (1-c)^2 w_2'^2$

$+ 2c(1-c)w_2 w_2')$

$= \lambda(c^2(w_1^2 + w_2^2) + (1-c)^2(w_1'^2 + w_2'^2) + 2c(1-c)(w_1'w_1 + w_2 w_2'))$

$= \lambda^2(w_1^2 + w_2^2) + \lambda(1-c)^2(w_1'^2 + w_2'^2) + 2\lambda c(1-c)(w_1'w_1 + w_2'w_2)$

ProMate

Given that $c \in [0, 1]$

$$W_1^2 + W_2^2 \geq c\left(w_1^2 + W_2^2\right)$$

$$\therefore c\left(W_1^2 + W_2^2\right) \geq c^2\left(W_1^2 + W_2^2\right) - Ⓒ$$

But $(1-c)$ is also $\in [0, 1]$

This implies $\Rightarrow (1-c)\left(w_1'^2 + W_2'^2\right) \geq (1-c)^2\left(w_1^2 + W_2^2\right)$

Ⓓ

Multiply both sides of Ⓒ & Ⓓ by λ gives us the part of expanded expression

$$\lambda c\left(W_1^2 + W_2^2\right) + \lambda(1-c)\left(W_1^2 + W_2'^2\right) \geq \lambda c^2\left(w_1^2 + W_2^2\right) + \lambda(1-c)^2\left(W_1'^2 + W_2'^2\right)$$

But $2\lambda c(1-c)\left(w_1'w_1 + W_2'W_2\right) > 0$ Ⓔ

$\therefore$ 4. Ⓐ $-$ Ⓔ $\leq$ Ⓑ

expanded A

thus, R.H.S of inequality $\geq$ L.H.S of inequality

b) Definition :- $f(cx_1 + (1-c)x_2) \leq cf(x_1) + (1-c)f(x_2)$

Let $F = f_1 + f_2$

$F(cx_1 + (1-c)x_2) = f_1(cx_1 + (1-c)x_2) + f_2(cx_1 + (1-c)x_2)$

(A)

(A) $\leq cf_1(x_1) + (1-c)f_1(x_2) + cf_2(x_1) + (1-c)f_2(x_2)$

$= c(f_1(x_1) + f_2(x_1)) + (1-c)(f_1(x_2) + f_2(x_2))$

$= c(F(x_1)) + (1-c)F(x_2)$

Since F also follows through from the definition, then $f_1 + f_2$ is also convex

c)

We know from a) that freg is a convex function. We also know that the sum of 2 convex functions is also a convex function. It was given that the existing loss function was also a convex function. This implies that the L2 regularization is also a convex function