

The Question

Hints for NPTEL MLESA assignment-3/question-7, by <http://perwad.in> on 2019/02/21 at 03:47

Consider

$$J(\mathbf{w}) = \frac{1}{10} \sum_{i=1}^5 (y^{(i)} - w_1 x^{(i)} - w_0)^2 \quad \text{where } \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

and the constants $x^{(i)}$ and $y^{(i)}$ are provided in the table below:

i	x	y
1	0.00	0.8822
2	0.25	1.2165
3	0.50	1.3171
4	0.75	1.7930
5	1.00	1.9826

Find the \mathbf{w} which minimize the $J(\mathbf{w})$.

i.e. $\underset{\mathbf{w}}{\operatorname{argmin}} J(\mathbf{w})$

Variables in the function J

The $J(\mathbf{w})$ is a multivariable function. There are only two variables in the function: w_0 and w_1 . The $x^{(i)}$ and $y^{(i)}$ are constants not variables. Let us expand the $J(\mathbf{w})$ in order to get clarified.

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{10} \sum_{i=1}^5 (y^{(i)} - w_1 x^{(i)} - w_0)^2 \\ &= \frac{1}{10} ((y^{(1)} - x^{(1)} w_1 - w_0)^2 + (y^{(2)} - x^{(2)} w_1 - w_0)^2 + \\ &\quad (y^{(3)} - x^{(3)} w_1 - w_0)^2 + (y^{(4)} - x^{(4)} w_1 - w_0)^2 + (y^{(5)} - x^{(5)} w_1 - w_0)^2) \end{aligned}$$

x and y are constants. Let us replace them with corresponding values.

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{10} ((0.8822 - 0.00w_1 - w_0)^2 + (1.2165 - 0.25w_1 - w_0)^2 + \\ &\quad (1.3171 - 0.50w_1 - w_0)^2 + (1.7930 - 0.75w_1 - w_0)^2 + (1.9826 - 1.00w_1 - w_0)^2) \\ &\approx 0.5w_0^2 + 0.5w_0w_1 - 1.43828w_0 + 0.1875w_1^2 - 0.858005w_1 + 1.11385 \end{aligned}$$

It proves that there are only two variables in the function. i.e. w_0 and w_1 . **Please cross-check** the approximation. I didn't verify the correctness of the approximation.

Derivative Rules

- Constant Rule

$$f(x) = c \quad \text{then} \quad f'(x) = 0$$

- Constant Multiple Rule

$$g(x) = c \times f(x) \quad \text{then} \quad g'(x) = c \times f'(x)$$

- Power Rule

$$f(x) = x^n \quad \text{then} \quad f'(x) = n \times x^{n-1}$$

- Sum and Difference Rule

$$h(x) = f(x) \pm g(x) \quad \text{then} \quad h'(x) = f'(x) \pm g'(x)$$

- Product Rule

$$h(x) = f(x) \times g(x) \quad \text{then} \quad h'(x) = f'(x) \times g(x) + f(x) \times g'(x)$$

- Quotient Rule

$$h(x) = \frac{f(x)}{g(x)} \quad \text{then} \quad h'(x) = \frac{f'(x) \times g(x) - f(x) \times g'(x)}{g(x)^2}$$

- Chain Rule

$$h(x) = f(g(x)) \quad \text{then} \quad h'(x) = f'(g(x)) \times g'(x)$$

Gradient of the function J

Let us see ∇J (gradient of J).

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \frac{\partial J}{\partial w_1} \end{bmatrix}$$

The partial differentiation of J with respect to w_0 and w_1 :

$$\begin{aligned} \frac{\partial J}{\partial w_0} &= \\ &= \frac{\partial}{\partial w_0} \left(\frac{1}{10} \sum_{i=1}^5 (y^{(i)} - w_1 x^{(i)} - w_0)^2 \right) \\ &= \frac{1}{10} \frac{\partial}{\partial w_0} \left(\sum_{i=1}^5 (y^{(i)} - w_1 x^{(i)} - w_0)^2 \right) && \text{Constant Multiple Rule} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(\frac{\partial}{\partial w_0} (y^{(i)} - w_1 x^{(i)} - w_0)^2 \right) && \text{Sum Rule} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(2 \times (y^{(i)} - w_1 x^{(i)} - w_0)^1 \times \frac{\partial}{\partial w_0} (y^{(i)} - w_1 x^{(i)} - w_0) \right) && \text{Chain Rule} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(2 \times (y^{(i)} - w_1 x^{(i)} - w_0) \times \left(\frac{\partial y^{(i)}}{\partial w_0} - \frac{\partial w_1 x^{(i)}}{\partial w_0} - \frac{\partial w_0}{\partial w_0} \right) \right) && \text{Sum/Diff Rule} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(2 \times (y^{(i)} - w_1 x^{(i)} - w_0) \times \left(0 - 0 - \frac{\partial w_0}{\partial w_0} \right) \right) && \text{Constant Rule} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(-2 \times (y^{(i)} - w_1 x^{(i)} - w_0) \times \frac{\partial w_0}{\partial w_0} \right) && \text{Simplification} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(-2 \times (y^{(i)} - w_1 x^{(i)} - w_0) \times 1 \times w_0^0 \right) && \text{Power Rule} \\ &= \frac{1}{10} \sum_{i=1}^5 \left(-2 \times (y^{(i)} - w_1 x^{(i)} - w_0) \right) && \text{Simplification} \\ &= \frac{-2}{10} \sum_{i=1}^5 \left(y^{(i)} - w_1 x^{(i)} - w_0 \right) && \text{Constant Rule} \\ &\quad \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \end{aligned}$$

and

Computing \mathbf{w}^{new} from \mathbf{w}^{old}

Each iteration starts with a \mathbf{w} , say \mathbf{w}^{old} , and at the end of the iteration will have the updated \mathbf{w} , say \mathbf{w}^{new} .

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \alpha * \nabla_{\mathbf{w}^{old}} J \quad \text{where } \alpha \text{ is learning rate.}$$

Let us expand it.

$$\begin{aligned} \begin{bmatrix} w_0^{new} \\ w_1^{new} \end{bmatrix} &= \begin{bmatrix} w_0^{old} \\ w_1^{old} \end{bmatrix} - \alpha * \begin{bmatrix} \frac{\partial J}{\partial w_0^{old}} \\ \frac{\partial J}{\partial w_1^{old}} \end{bmatrix} \\ &= \begin{bmatrix} w_0^{old} - \alpha * \frac{\partial J}{\partial w_0^{old}} \\ w_1^{old} - \alpha * \frac{\partial J}{\partial w_1^{old}} \end{bmatrix} \end{aligned}$$

Have a look at the w_0^{new} and w_1^{new} separately.

$$w_0^{new} = w_0^{old} - \alpha * \frac{\partial J}{\partial w_0^{old}}$$

and

$$w_1^{new} = w_1^{old} - \alpha * \frac{\partial J}{\partial w_1^{old}}$$

Python Code

```
import numpy as np

x = np.array([0, 0.25, 0.5, 0.75, 1.00])
y = np.array([0.8822, 1.2165, 1.3171, 1.7930, 1.9826])

def J(w):
    return 1.0/10 * sum([(y[i]-w[1]*x[i]-w[0])**2 for i in range(5)])

def gradientJ(w):
    return np.array([
        -2.0/10 * sum([(y[i]-w[1]*x[i]-w[0]) for i in range(5)]),      # wrt w0
        -2.0/10 * sum([(y[i]-w[1]*x[i]-w[0])*x[i] for i in range(5)]) # wrt w1
    ])

w = np.array([0, 0])
alpha = 1.0

for k in range(1, 6):
    print("%d old W:%-24s " % (k, w), end="")
    w = w - alpha * gradientJ(w)
    print(" new W:%s " % (w))
```