Explanation of Gradient Descent Algorithm in Linear Regression Implementation

In my linear regression implementation file, the gradient descent algorithm in the 'fit_linear_regression' function may seem as if it was pulled out of thin air.

For that reason, I have decided to include this document explaining the mathematics behind it, without going into too much detail and working.

The cost function (squared error) of the univariate linear regression algorithm is:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Where $h_\theta(x) = \theta_0 + \theta_1 x$.

The gradient descent algorithm is defined by:

Repeat until convergence {

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$

(Simultaneously update j = 0 and j = 1)

}

After calculating the required partial derivatives and subbing into the algorithm, the gradient descent for univariate linear regression is given by:

Repeat until convergence {

$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$

$\theta_1 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$

}

Which is vectorized in the gradient descent algorithm part of my implementation.