$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots\ldots + \theta_n x_n$.

So the number of columns in θ would be number of features +1

$$x = \begin{bmatrix} \ldots x_1 \ldots \\ \ldots x_2 \ldots \\ \vdots \\ \ldots x_m \ldots \end{bmatrix}$$

m training examples,
n features,

But for calculating $h_\theta(x)$ we need $n+1$ columns to add the bias.

$$h_\theta(x) = \underbrace{\theta_0}_{1} + \underbrace{\theta_1 x_1 + \theta_2 x_2 + \ldots\ldots + \theta_n x_n}_{n \text{ columns in } x \text{ vector}}$$

Formula for gradient descend.

Loop {

for $i = 1$ to n, {

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}, \quad \text{(for every } j\text{)}$$

}

}

$$\theta_j := \theta_j - \alpha \cdot \frac{2}{m} \sum \left(\vec{x}_b \theta - y \right) x_j$$

vector. scaler. vector vector vector vecto

Vector Dimension

$x_b \rightarrow (m, n+1)$ (n+1 to accomodate the bias)

$\theta \rightarrow (n+1, 1)$

$y \rightarrow (m, 1)$

So to apply the vectorization we do the following.

$$x_b . \theta \rightarrow (m, n+1) \times (n+1, 1) \rightarrow (m, 1).$$

$$x_b . \theta - y \rightarrow (m, 1) - (m, 1) \rightarrow (m, 1)$$

So to be able to multiply X_b with the result we need to transpose it.

$x_b^T \rightarrow (n+1, m)$ } compatible $\rightarrow (n+1, 1)$

$x_b \theta - y \rightarrow (m, 1)$ }

$$\theta_j := \underbrace{\theta_j}_{(n+1,1)} - \underbrace{\alpha \cdot \frac{2}{m} \sum (\vec{x}_0 \theta - y) x_j}_{n+1,1}$$

$(n+1,1)$ $(n+1,1)$

Hence the dimensions are properly matched.

And it would use the parallelism to calculate the gradient.

Further optimizations:

We can stop the G.D after certain steps if the cost reduction is very low.