Linear Regression

Ugenteraan Manogaran

January 26, 2019

Suppose the inputs are :

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & . & . & . & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & . & . & . & x_2^{(m)} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_n^{(1)} & x_n^{(2)} & . & . & . & x_n^{(m)} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y^{(1)} & y^{(2)} & . & . & . & y^{(m)} \end{bmatrix} \tag{1}$$

The goal is to find a function **h**, such that **h** approximates **Y** given **X**.

To approximate **Y** as a linear function of **X** ,

$$\mathbf{h}_\theta(\mathbf{x}^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + ... + \theta_n x_n^{(i)} \tag{2}$$

or

$$\mathbf{h}_\theta(\mathbf{x}^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)} \tag{3}$$

or, simply,

$$\mathbf{h}_\theta(\mathbf{X}) = \begin{bmatrix} \theta_0 & \theta_1 & . & . & . & \theta_n \end{bmatrix} \begin{bmatrix} x_0^{(1)} & x_0^{(2)} & . & . & . & x_0^{(m)} \\ x_1^{(1)} & x_1^{(2)} & . & . & . & x_1^{(m)} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_n^{(1)} & x_n^{(2)} & . & . & . & x_n^{(m)} \end{bmatrix} \tag{4}$$

where $x_0 = 1$, $\theta_i \in \mathbb{R}$ and i = 1 , . . . , m

θ_i's are initialized with random values at first. Hence, $\mathbf{h}_\theta(\mathbf{X})$ most likely would not be close to **Y** at all.

Cost function is defined as :

$$\mathbf{J}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\mathbf{h}_\theta(\mathbf{x}^{(i)}) - \mathbf{y}^{(i)})^2 \tag{5}$$

Repeatedly perform the following update :

$$\theta_j := \theta_j - \sigma(\frac{\partial}{\partial\theta_j}\mathbf{J}(\theta)) \tag{6}$$

where j = 0 , . . . , n and $\sigma \in \mathbb{R}^+$. Usually σ will be between 0 and 1.

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\mathbf{J}(\theta) &= \frac{\partial}{\partial(\theta_j)}\frac{1}{2m}\sum_{i=1}^{m}(\mathbf{h}_\theta(x^{(i)} - y^{(i)})^2 \\
&= 2.\frac{1}{2m}\sum_{i=1}^{m}(\mathbf{h}_\theta(x^{(i)} - y^{(i)})\frac{\partial}{\partial(\theta_j)}(\mathbf{h}_\theta(x^{(i)}) - y^{(i)}) \\
&= \frac{1}{m}\sum_{i=1}^{m}(\mathbf{h}_\theta(x^{(i)} - y^{(i)})\frac{\partial}{\partial(\theta_j)}(\sum_{j=0}^{n}\theta_j x_j^{(i)} - y^{(i)}) \\
&= \frac{1}{m}\sum_{i=1}^{m}(\mathbf{h}_\theta(x^{(i)} - y^{(i)})x_j^{(i)}
\end{aligned}$$

References

[1] Andrew Ng, *CS229 Lecture notes.*

[2] David Meyer, *Notes on MSE Gradients for Neural Networks.* dmm@1-4-5.net,uoregon.edu,...