Epoch – Learning Phase

Machine Learning

Topic Report

Ishaan Jain
CO21BTECH11006

Linear Regression:

Linear Regression is estimating the output of some data by some previous data assuming it to be linear. It is a supervised learning algorithm and one of the simplest ML algorithms.

We will start with a single variable data and later generalize it.

Assume we have m training examples $(x,\ y)$, the hypothesis will be a straight line.

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

Here, θ is a parameter which is a vector $[\theta_0\ \theta_1]^T$. We have to train θ such that the line $h(\theta)$ is the best fit line. To check if the line is best fit, we have to reduce the error between the predicted value and the true value on the training data. This error function is called the cost function, donated by $J(\theta)$.

$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2$, where $(x^i,\ y^i)$ is the i^{th} example.

We must minimize this cost function to make our hypothesis function the best fit straight line. This optimization problem is solved by **Gradient Descent.**

Gradient Descent: $\theta \leftarrow \theta - \alpha\,\frac{\partial J}{\partial \theta}$

Though this is not the correct way to represent it. We must minimize the cost function with respect to both θ_0 and θ_1. The correct way is:

$$\theta_0 \leftarrow \theta_0 - \alpha\,\frac{\partial J}{\partial \theta_0}$$

$$\theta_1 \leftarrow \theta_1 - \alpha\,\frac{\partial J}{\partial \theta_1}$$

Source – Notes that I made while studying CS229

Substituting $J(\theta)$, we get

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

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

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{2}\sum_{i=1}^{m} 2\left(h_\theta(x^i) - y^i\right) .1$$

$$\frac{\partial J(\theta)}{\partial \theta_0} = \sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)$$

Similarly,

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

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

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

$$\frac{\partial J(\theta)}{\partial \theta_1} = \sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)x^i$$

With multiple iterations, we can minimize the cost function. (I have made a report on Gradient Descent as well, so not including many details here.)

This way we get the optimized parameter θ. The next step would be to find $h_\theta(x)$.

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

Here, x will be a test example and the value $h_\theta(x)$ is the predicted value at x .

For multivariate linear regression, we will have vector x in place of scalar.

$$x = [1\; x_1\; x_2 \ldots x_n]^T$$

$\therefore$ x is a vector of shape $1 \times (n+1)$

Hence, the hypothesis function will be

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

Source – Notes that I made while studying CS229

Or simply, $h_\theta(x) = \theta^T x$

Where, $\theta^T = [\theta_0\ \theta_1\ \theta_2 \cdots \theta_n]$

With the loss function,

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

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(\theta^T x^i - y^i\right)^2$$

Now, we can apply gradient descent here,

$$\theta_i \leftarrow \theta_i - \alpha\frac{\partial J}{\partial \theta_i}, \text{ for } i = \{0,\ 1,\ 2 \cdots n\} \tag{a}$$

Instead of applying for every element of θ, we can vectorize it and apply for whole θ at once. For this, we need to find the gradient of $J(\theta)$ w.r.t θ.

$$\nabla_\theta J(\theta) = \left[\frac{\partial J}{\partial \theta_0}\frac{\partial J}{\partial \theta_1}\frac{\partial J}{\partial \theta_2}\cdots\frac{\partial J}{\partial \theta_n}\right]^T \therefore \text{ a vector of shape } 1 \times (n+1)$$

As in equation (a), the shapes of vectors are compatible, we can apply gradient descent in vectorized form.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)x_j^i \tag{b}$$

where x_j^i is the j^{th} element of the i^{th} data point.

We can combine (a) and (b) as

$$\theta \leftarrow \theta - \alpha\nabla_\theta J(\theta)$$

There are some disadvantages of this approach though. The iterations in the Gradient Descent method might take time. There is an alternative algorithm for linear regression which can be done in a single step.

In order to find the minimum of Cost function, we must find a point where its gradient is 0. $i.e.$

$$\nabla_\theta J(\theta) = 0,$$

Let's define $X = \left[x^{(1)^T}\, x^{(2)^T}\, x^{(3)^T} \cdots x^{(m)^T}\right]^T$ and $Y = \left[y^{(1)}\, y^{(2)}\, y^{(3)} \cdots\, y^{(m)}\right]^T$.

We see that $X\theta = \left[\left(x^{(1)^T}\theta\right)\left(x^{(2)^T}\theta\right)\left(x^{(3)^T}\theta\right)\cdots\left(x^{(m)^T}\theta\right)\right]^T$

Or simply, $X\theta = \left[h_\theta\left(x^{(1)}\right) h_\theta\left(x^{(2)}\right) h_\theta\left(x^{(3)}\right)\cdots\, h_\theta\left(x^{(m)}\right)\right]^T$

The cost function is the sum of squares of error, so it should be the sum of squares of $X\theta - Y$. In matrices, the square is given by product of transpose of a vector by itself. $i.e.$

$$J(\theta) = \frac{1}{2}(X\theta - Y)^T(X\theta - Y)$$

$$J(\theta) = \frac{1}{2}((X\theta)^T - Y^T)(X\theta - Y)$$

$$J(\theta) = \frac{1}{2}(\theta^T X^T X\theta - 2Y^T X\theta + Y^T Y)$$

We just have to do $\nabla_\theta J(\theta) = 0$, on further solving, we find that,

$$\nabla_\theta J(\theta) = 0 \Rightarrow \frac{1}{2}(2X^T X\theta - 2X^T Y) = 0$$

$$X^T X\theta - X^T Y = 0$$

$$\Rightarrow X^T X\theta = X^T Y$$

$$\Rightarrow \theta = (X^T X)^{-1} X^T Y$$

The predicted value of the data sample is found by the hypothesis function.

An example of linear regression problem that I did is here.

Source – Notes that I made while studying CS229