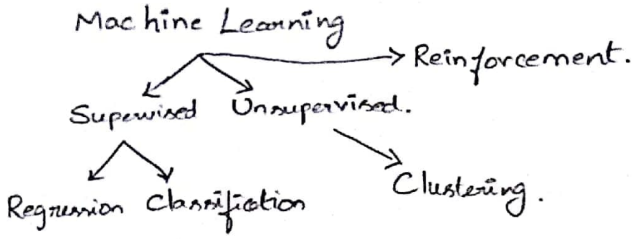
1.



2. Univariate Linear Regression.

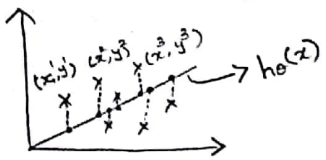
You'll be given a data with one column of X's and one column of Y's. These X's are called "features" and Y's are called "Actual output". We need to find a function which can map these X's to Y's.

X	Y
x^1	y^1
x^2	y^2
x^3	y^3

→ m samples.

$x \rightarrow \boxed{h_\theta(x)} \rightarrow y$ This $h_\theta(x)$ is called hypothesis function

The data along with the hypotheses looks like this when plotted:



All x^i refer to the feature
All y^i refer to the actual output
All "." refer to the predicted output ($h_\theta(x)^{(i)}$)

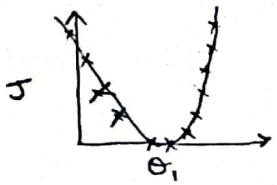
The cost function is written as the following.

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]^2$$ This is sometimes called as Mean Squared Error.

$h_\theta(x)$ for univariate L.R. looks like $h_\theta(x) = \theta_0 + \theta_1 x$.

Intuition 1 for squared cost function:

Let's keep θ_0 as zero and plot with θ_1 alone against cost function, J. for diff values of θ_1 we get the following parabolic looking graph.



Intuition 2 for squared Error function:

There is a field of mathematics called Convex optimization. It uses parabolic like shapes to optimize parameters. Squaring a function automatically makes any type of function parabolic. That is why we use MSE.

Now Let's see how Gradient Descent Works. (A form of convex optimization)

While not $(\theta_{OLD} - \theta_{NEW}) <$ a small value,

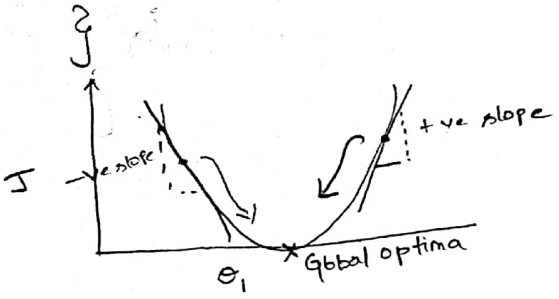
do { $\theta^j_{NEW} = \theta^j_{OLD} - \alpha \frac{d}{d\theta^j_{OLD}} J(\theta_0, \theta_1)$

Derivative:

1. Derivative can be thought of a slope of the tangent line at that point.

This has a -ve slope

This has a +ve slope



α is the learning rate.

$\alpha \gg$ overshoots, never converges.

$\alpha \ll$ Takes forever to converge.

The whole linear regression algorithm.

1. Feed X and y to the system. Split 80% of x and y for training and 20% for testing.
2. The computer asks for θ_0 and θ_1. Give a Random Value.
3. Using the given θ_0 and θ_1 it plots a hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$.
4. It checks for the mean squared Error. Once when it knows that there is room for optimization, it calls the G.D.
5. GD will run until $\theta_{OLD} - \theta_{NEW} \ll$ small value. (It will ask for the learning rate)
6. Now again the most optimal $h_\theta(x)$ is plotted with the best θ_0 and θ_1.
7. Use the test set to check the accuracy.

Multivariate

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \cdots$

Polynomial

$h_\theta(x) = \theta_0 + \theta_1 x_1^2 + \theta_2 x^3 + \theta_3 (x+2)^2$

Overfitting: When the curve fits very specific the train data, it is called overfitting. It is very similar to memorizing. When something which is not in the book is asked, the model cannot predict properly. It cannot generalize.