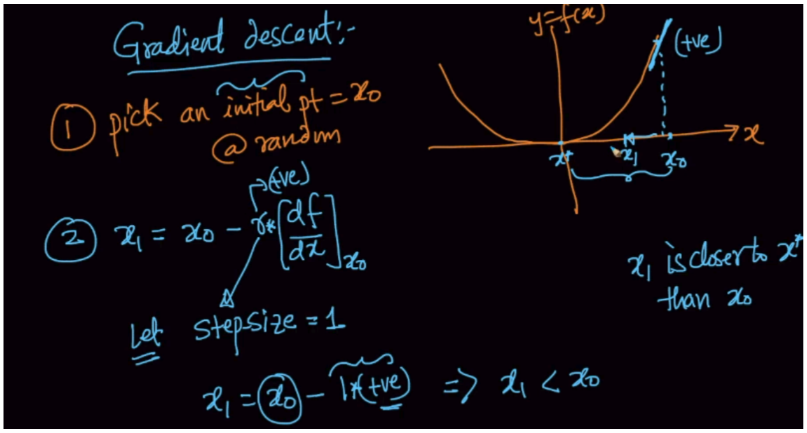
Gradient Descent (Intuition) -

Gradient descent is an iterative optimization algorithm for finding the local minimum of a function.

- Gradient Descent is a first order iterative optimization algorithm for finding the minimum of a function.
- Gradient/slope is a multivariable generalization of the derivative. The gradient is a vector valued function as opposed to the scalar derivative.
- **The objective of GD is to find the minima of a convex function by taking iterative steps of descend until the difference between its former and current state doesn't change much or reaches 0.**



Gradient descent:-

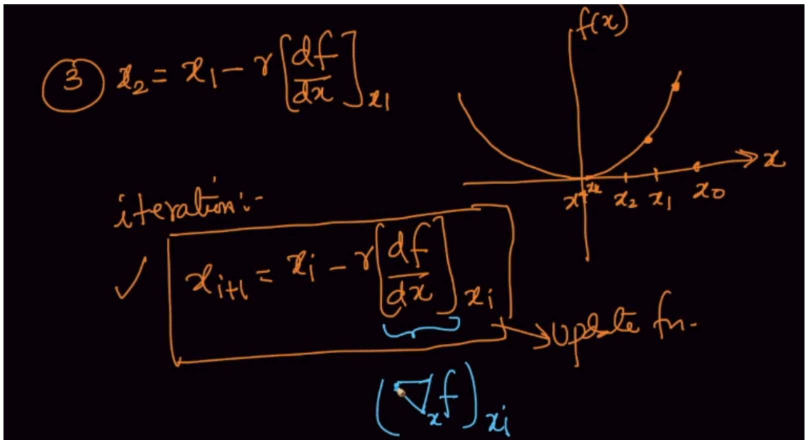
① pick an initial pt $= x_0$ @ random

② $x_1 = x_0 - \overset{(+ve)}{r *}\left[\frac{df}{dx}\right]_{x_0}$

let stepsize $= 1$

$x_1 = x_0 - 1 * (+ve) \Rightarrow x_1 < x_0$

x_1 is closer to x^* than x_0



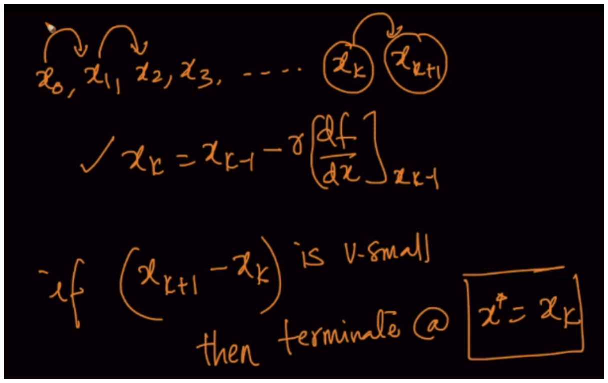
③ $x_2 = x_1 - r\left[\frac{df}{dx}\right]_{x_1}$

iteration:-

✓ $\boxed{x_{i+1} = x_i - r\left[\frac{df}{dx}\right]_{x_i}}$ → update fn.

$(\nabla_x f)_{x_i}$

If the x_0 is on the negative side, then the slope will be negative and hence the convergence to local minima will still hold true.



$x_0, x_1, x_2, x_3, \ldots\ x_k, x_{k+1}$

✓ $x_k = x_{k-1} - r\left[\frac{df}{dx}\right]_{x_{k-1}}$

if $(x_{k+1} - x_k)$ is v-small then terminate @ $\boxed{x^* = x_k}$

However, if we don’t optimise the learning rate with each iteration, we may face an oscillation problem where the convergence just oscillates on either side of the local minima.

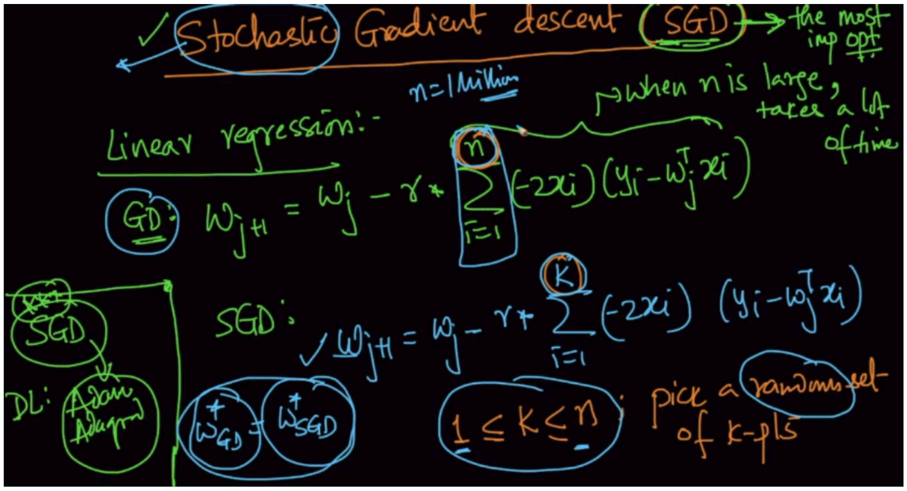
- To remove oscillating problem we minimize value of r or step-size.

Cost Function -

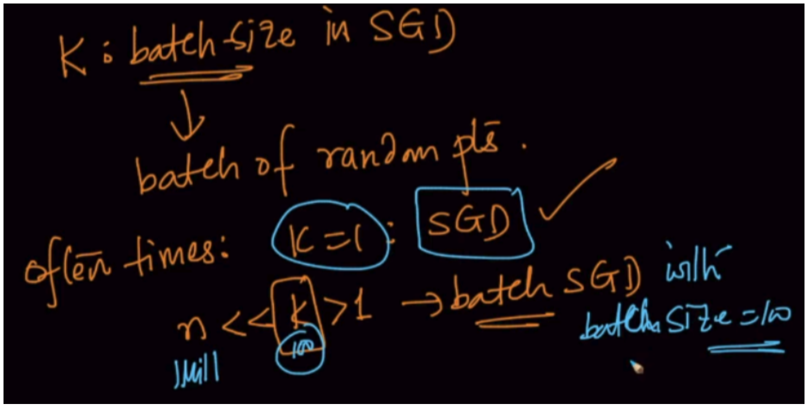
It is a function that measures the performance of a model for any given data. Cost Function quantifies the error between predicted values and expected values and presents it in the form of a single real number.

Stochastic Gradient Descent:

In SGD algorithm, instead of finding the sum of all differentiables attached with the learning rate, we randomly shuffle the training examples and use a certain k points instead of all the n points for evaluating the step.


Stochastic Gradient descent (SGD) → the most imp opt
n=1 Million
when n is large, takes a lot of time
Linear regression:-
GD: $W_{j+1} = W_j - r * \sum_{i=1}^{n} (-2x_i)(y_i - W_j^T x_i)$
SGD: $W_{j+1} = W_j - r * \sum_{i=1}^{k} (-2x_i)(y_i - W_j^T x_i)$
$1 \leq k \leq n$; pick a random set of k-pts
SGD
DL: Adam Adagrad
$W^*_{GD} = W^*_{SGD}$

- In SGD, because it’s using only one/few examples at a time, its path to the minima is noisier (more random) than that of the batch gradient. But it’s ok as we are indifferent to the path, as long as it gives us the minimum AND the shorter training time.



SGD with Example- There are a few downsides of the gradient descent algorithm.

Say we have 10,000 data points and 10 features. The sum of squared residuals consists of as many terms as there are data points, so 10000 terms in our case. We need to compute the derivative of this function with respect to each of the features, so in effect we will be doing 10000 *10*

= 100,000 computations per iteration. It is common to take 1000 iterations, in effect we have 100,000 1000 = 100000000 computations to complete the algorithm. That is pretty much an overhead and hence gradient descent is slow on huge data.

Stochastic gradient descent comes to our rescue. “Stochastic” means “random”.

Ques-Where can we potentially induce randomness in our gradient descent algorithm?

Ans- It is while selecting data points at each step to calculate the derivatives. SGD randomly picks one data point from the whole data set at each iteration to reduce the computations enormously. It is also common to sample a small number of data points instead of just one point at each step and that is called “mini-batch” gradient descent. Mini-batch tries to strike a balance between the goodness of gradient descent and speed of SGD.

Refer -

1. https://towardsdatascience.com/difference-between-batch-gradient-descent-and-stochastic-gradient-descent-1187f1291aa1 (https://towardsdatascience.com/difference-between-batch-gradient-descent-and-stochastic-gradient-descent-1187f1291aa1)
2. https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31 (https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31)
3. https://en.wikipedia.org/wiki/Stochastic_gradient_descent (https://en.wikipedia.org/wiki/Stochastic_gradient_descent)