① { Gradient descent }

Gradient descent is a first order iterative optimization problem.

This algorithm is used in Linear Regression, Logistic Regression, T-sne and all algorithms in deep learning.

5 videos

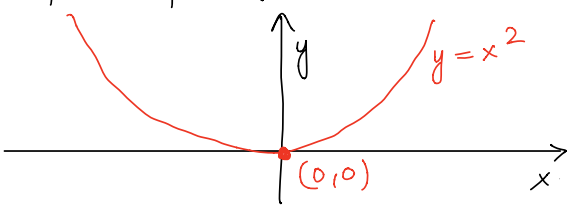
1. Gradient descent from Scratch
2. Batch GD
3. Stochastic GD
4. mini-batch GD
5. SGD Regressor { Scikit-learn implementation of stochastic GD }

② { Process of GD }

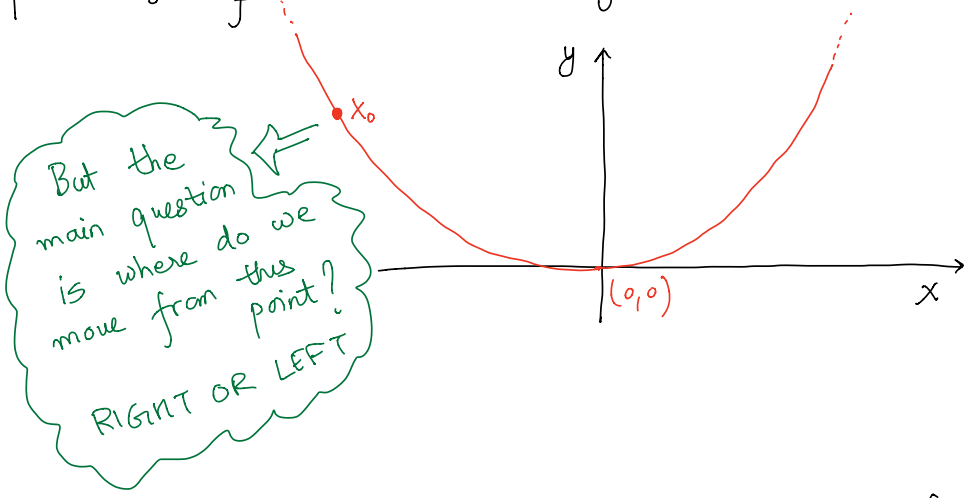
Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a function of n-variables. For simplicity, let us restrict our discussion to real valued function of one real variable only initially.

So, let $f : \mathbb{R} \to \mathbb{R}$ be a function and let us suppose it has a minima in its domain. We need to find the point of minima using iterations (epochs).

We can take example of $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2 \; \forall x \in \mathbb{R}$.



We know from our knowledge of calculus, $f(x) = x^2$ has a minima at $(0,0)$. But let's say we did not have any idea about this. In that case, we would just take any arbitrary point $x_0 \in D_f$ and start moving in the direction of minima.



But the main problem is where do we move from this initial point x_0. The answer to this is the gradient at that point. In case of 1D function, gradeent at a point is the same as derivative of function evaluated at that particular point.

Suppose $\left.\frac{df}{dx}\right|_{x=x_0} < 0$, and we know that at minima $\frac{df}{dx} = 0$

So, in order to find the new point, we need to move forward, that is we need to move in RIGHT direction. For this, we need to add something +ve in x_0, or perhaps subtract something −ve from x_0. we can use derivative at this point.

So, in case if $\left.\frac{df}{dx}\right|_{x=x_0} < 0$, the new point in our iteration x_1 is

$$\boxed{x_1 = x_0 - \left.\frac{df}{dx}\right|_{x=x_0}}$$

Now, suppose at x_0, we have $\left.\frac{df}{dx}\right|_{x=x_0} > 0$, then we need to move LEFT of x_0 on the curve. This can be done by subtracting a +ve quantity from x_0 or adding a -ve quantity in x_0. Here again, we can use derivative for this Thus,

$$\boxed{x_1 = x_0 - \left.\frac{df}{dx}\right|_{x=x_0}}$$

③ {final GD equation}

Thus, the final equation in GD is

$$x_{new} = x_{old} - \left.\frac{df}{dx}\right|_{x=x_0}$$

$$= x_{old} - (\nabla f)_{x=x_0}$$

where $\nabla f \triangleq$ gradient of $f : \mathbb{R}^n \to \mathbb{R}$.

However, there is a slight modification which is required in this equation. We need to multiply the gradient of f by the learning rate (η). The learning rate is a very crucial parameter in the GD. If the LR is too high, the algorithm might oscillate or may as well diverge from the minima. If it is too low, then the algorithm might take longer to reach the minima and thus the # of epochs will increase which is also not feasible. Thus, we need to make sure that η is tuned in such a way that the algorithm converges rapidly to minima.

$$X_{new} = X_{old} - \eta (\nabla f)_{x=x_0}$$

Here $\nabla f \triangleq \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$ ⟹ gradient vector of $f: \mathbb{R}^n \to \mathbb{R}$

In general, for most of the algorithms, we take $\eta = 0.1$.

The term $\eta (\nabla f)_{x=x_0}$ is called the **step size**.

④ {Example of GD}

Let us take the function $f: \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2 \; \forall \; x \in \mathbb{R}$. We need to find the point of its minima.

$$(\nabla f)_{x=x_0} = \left.\frac{df}{dx}\right|_{x=x_0} = (2x)_{x=x_0} = 2x_0$$

Let us take the LR $(\eta) = 0.1$, and $x_0 = -3$.

1st iteration

$$x_1 = x_0 - \eta(\nabla f)_{x=x_0}$$
$$= -3 - (0.1)[2\times(-3)] = -3 + 0.6 = -2.4$$

Thus, $x_1 = -2.4$

and $f(x_1) = -5.76$

2nd iteration

$$x_2 = x_1 - \eta(\nabla f)_{x=x_1}$$
$$= (-2.4) - (0.1)[2\times(-2.4)]$$
$$= -2.4 + \frac{1}{10}(4.8) = -2.4 + 0.48$$
$$= -1.92$$

Thus, $x_2 = -1.92$

$f(x_2) = -3.6864$

3rd iteration

$$x_3 = x_2 - \eta(\nabla f)_{x=x_2}$$
$$= -1.92 - (0.1)\times 2\times(-1.92)$$
$$= (-1.92)(1 - 0.2) = -(1.92)\times(0.8)$$

$= -1.536$

Thus, $x_3 = -1.536$

$f(x_3) = x_3^2 = -2.359296$

4th iteration

$$x_4 = x_3 - \eta (\nabla f)_{x=x_3} = x_3 - (0.1)[2x_3]$$

$$= (0.8)x_3 = 0.8 \times (-1.536)$$

$$= -1.2288$$

Thus, $x_4 = -1.2288$

$f(x_4) = (-1.2288)^2 = -1.5099$

Similarly, proceding with the further iterations, we get the minima.

⑤ { GD in Linear Regression }

In LR, we have OLS. which gives us the exact solution. However, in higher dimensions, matrix multiplication is expansive and thus, we may use iterative methods like GD.

The, Loss function in SLR is given by

$$L \triangleq \sum_{i \in \{1,\dots,n\}} (\hat{y}_i - y_i)^2$$

$$L = \sum_{i=1}^{n} (mx_i + b - y_i)^2$$

$L : \mathbb{R}^2 \to \mathbb{R}$ is a function in two variables, m and b.

$$\nabla L = \left(\frac{\partial L}{\partial m}, \frac{\partial L}{\partial b} \right)$$

$$= \left(2 \sum (mx_i + b - y_i)(x_i), \; 2 \sum (mx_i + b - y_i)(1) \right)$$

We start with (m_0, b_0) and then find subsequent points.

$$(m_1, b_1) = (m_0, b_0) - \eta \left(2 \sum (m_0 x_i + b_0 - y_i)(x_i), \; 2 \sum (m_0 x_i + b_0 - y_i) \right)$$

$$\begin{pmatrix} m_1 \\ b_1 \end{pmatrix} = \begin{pmatrix} m_0 - 2\eta \sum_i (m_0 x_i + b_0 - y_i)(x_i) \\ b_0 - 2\eta \sum_i (m_0 x_i + b_0 - y_i) \end{pmatrix}$$

We can write the general equation as:

$$\begin{pmatrix} m_{k+1} \\ b_{k+1} \end{pmatrix} = \begin{pmatrix} m_k - 2\eta \sum_i (m_k x_i + b_k - y_i)(x_i) \\ b_k - 2\eta \sum_i (m_k x_i + b_k - y_i) \end{pmatrix}$$