Outline

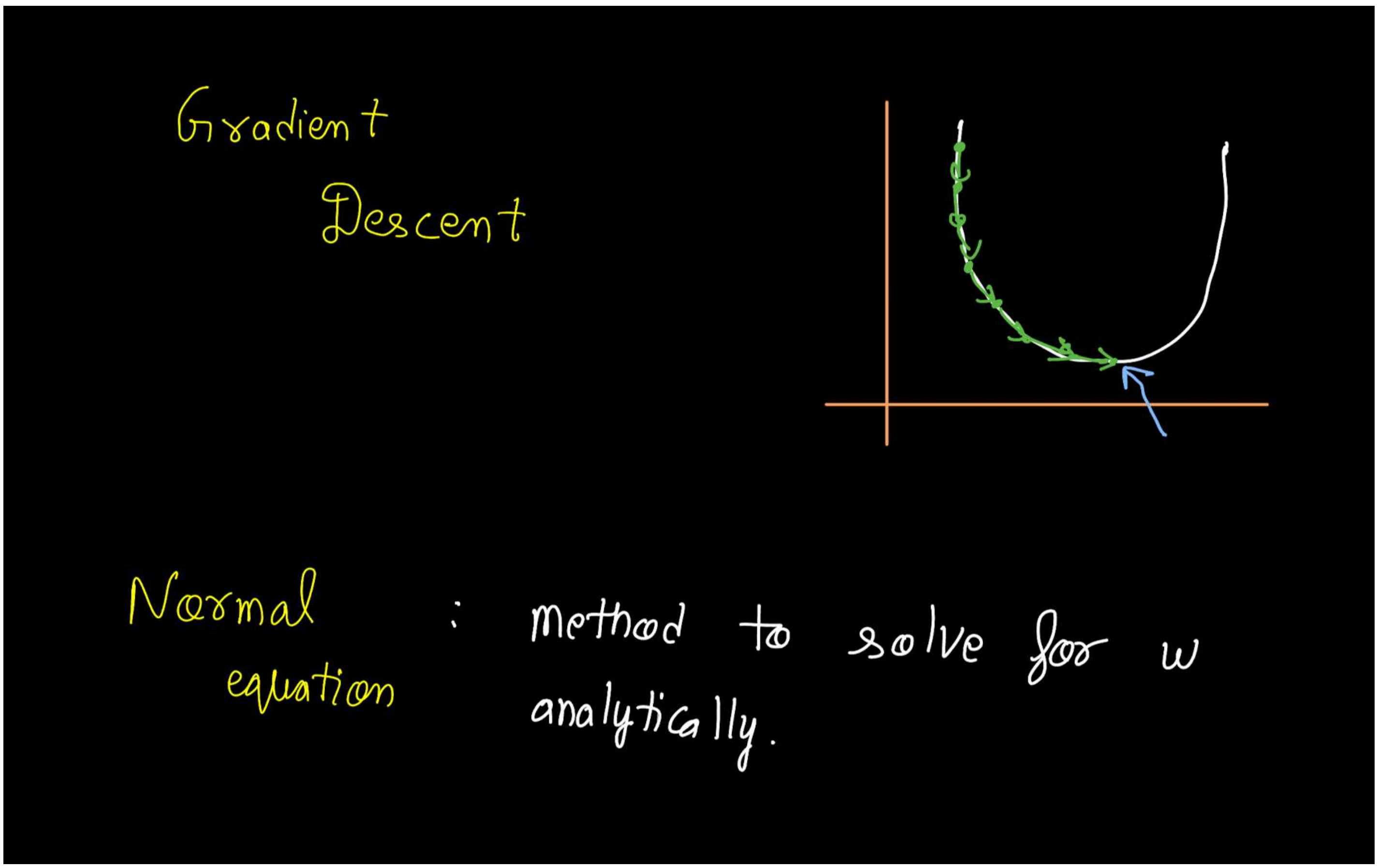
- Normal Equation/ Closed Form solution

Closed form solution:

Derivation: https://www.youtube.com/watch?v=g8qF61P741w

reference video: https://www.youtube.com/watch?v=B-Ks01zR4HY&ab_channel=ArtificialIntelligence-AllinOne

blog: https://towardsdatascience.com/normal-equation-in-python-the-closed-form-solution-for-linear-regression-13df33f9ad71



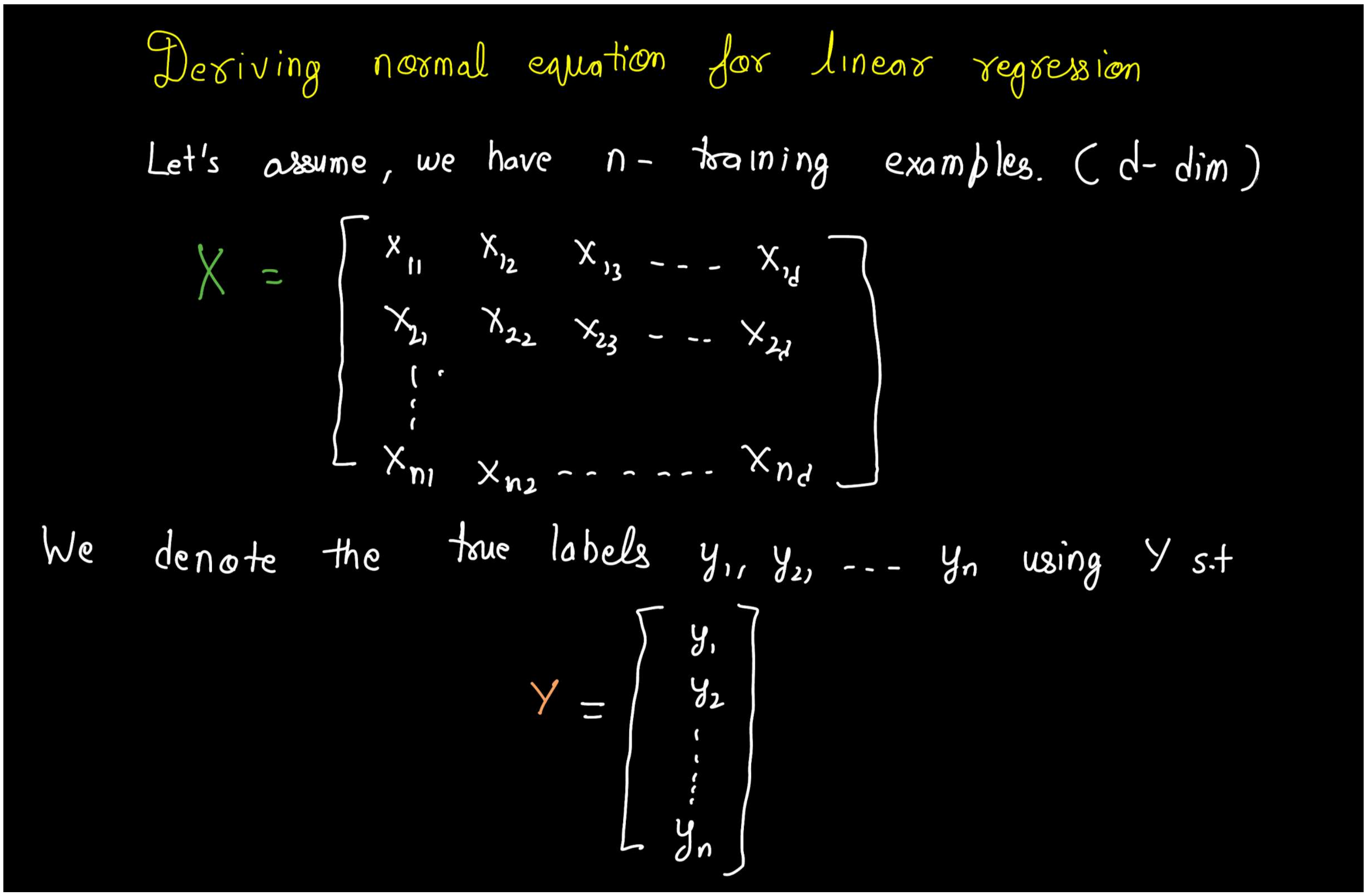
Gradient Descent is an iterative algorithm meaning that you need to take multiple steps to get to the Global optimum

Is there any other way to solve Linear Regression?

Yes. Using Normal Equation

It solves for the optimal values of the parameter `w` in one step without needing to use an iterative algorithm and this algorithm is called the Normal Equation. It works only for Linear Regression and not any other algorithm.

- Normal Equation is the Closed-form solution
- It means that we can obtain the optimal parameters by just using a formula that includes a few matrix multiplications and inversions.



Objective: We want to find weights $w_1, w_2, \ldots, w_d$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

For simplicity, assume $w_0 = 0$ (passes through origin)

Error (e_i) = $y_i - \hat{y}_i$

In matrix form,

$$\text{Error} = \begin{bmatrix} y_1 - (w_1 x_{11} + w_2 x_{12} + \cdots + w_d x_{1d}) \\ y_2 - (w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d}) \\ \vdots \\ y_n - (w_1 x_{n1} + w_2 x_{n2} + \cdots + w_d x_{nd}) \end{bmatrix} \begin{matrix} \to e_1 \\ \to e_2 \\ \vdots \\ \leftarrow e_n \end{matrix}$$

$$= Y - Xw$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1d} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2d} \\ \vdots & & & & \\ x_{n1} & x_{n2} & \cdots & \cdots & x_{nd} \end{bmatrix} \qquad W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

$$XW = \begin{bmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 + \cdots + x_{1d}w_d \\ x_{21}w_1 + x_{22}w_2 + \cdots + x_{2d}w_d \\ \vdots \\ x_{n1}w_1 + x_{n2}w_2 + \cdots + x_{nd}w_d \end{bmatrix}$$

Now,

$$\text{Error} = Y - XW$$

We know, Loss is sum of squared errors

$$\boxed{\mathcal{L} = \sum_{i=1}^{n} e_i^2}$$

$$= e_1^2 + e_2^2 + \cdots + e_n^2$$

Now,

$$\mathcal{L} = (y - Xw)^T (y - Xw)$$

$$\because [e_1 \; e_2 \; e_3 \; \dots \; e_n] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix} = e_1^2 + e_2^2 + \dots e_n^2$$

Multiplying the terms,

$$= y^T y - y^T X w - w^T X^T y + w^T X^T X w$$

$$(A+B)^T = A^T + B^T$$
$$(AB)^T = B^T A^T$$

We want to minimize loss & in order to minimize loss, we take its derivative & equate to 0.

$$\nabla_w \mathcal{L} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial w_1} \\ \frac{\partial \mathcal{L}}{\partial w_2} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial w_d} \end{pmatrix} = 0 \Rightarrow \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}$$

We know,

$$\nabla_w (w^T a) = \nabla_w (a^T w) = a$$

where $w^T a = a^T w = w_1 a_1 + w_2 a_2 + \cdots w_d a_d$

$$\begin{pmatrix} \frac{\partial (w^T a)}{\partial w_1} \\ \vdots \\ \frac{\partial (w^T a)}{\partial w_d} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{pmatrix} = a$$

$$\mathcal{L} = y^T y - y^T X w - w^T X^T y + w^T X^T X w$$

① As $y^T y$ is constant term, its derivative will be zero.

② $y^T X w$ can be written as $(X y^T)^T w$ $\quad (\because (AB)^T = B^T A^T)$

Now, $\nabla_w (w^T A) = \nabla_w (A^T w) = A$

Using this $\nabla_w (y^T X w) = \nabla_w (X^T y)^T w = X^T y$

③ Similarly, $\nabla_w (w^T X^T y) = X^T y$

$$\left(\because \nabla_w (w^T A) = \nabla_w (A^T w) = A\right)$$

4) To calculate the derivative of $w^T x^T x w$,
we'll use product rule.

$$\left[\, d(u.v) = v\,d(u) + u.\,d(v) \,\right]$$

$$\nabla_w w^T \overbrace{x^T x w}^{\text{assume constant}} = x^T x w \qquad (\because \nabla_w w^T a = a)$$

$$\nabla_w \overbrace{w^T x^T x}^{\text{assume constant}} w = \nabla_w (x^T x w)^T w \qquad (\because \nabla_w a^T w = a)$$

$$= x^T x w$$

$$\nabla_w \mathcal{L} = 0 - X^T Y - X^T Y + X^T X w + X^T X W$$

$$= -2X^T Y + 2X^T X w.$$

$$\nabla_w \mathcal{L} = -2X^T Y + 2X^T X w.$$

Equating to zero

$$\Rightarrow -2X^T Y + 2X^T X W = 0$$

$$\Rightarrow \cancel{2} X^T X W = \cancel{2} X^T Y$$

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

$$\boxed{W = (X^T X)^{-1} X^T Y} \qquad \left(\because \text{if } AB = C,\ B = A^{-1} C\right)$$

By equating the values of X and Y, we can compute the values of W

Comparing Normal equation to GD

- There is no need to for feature scaling in Normal Equation
- GD works well for any number of dimensions.
- But, We need to calculate $(X^T X)^{-1}$ in normal equation