Linear Regression

Linear regression is a basic statistical method used to understand the relationship between two variables. In simple terms, it tries to find a straight line that best fits the data points. The goal is to predict the value of one variable (the dependent variable) based on the value of another (the independent variable).

For example, if you have data about how much people study (independent variable) and their exam scores (dependent variable), linear regression can help predict a student's score based on the hours they studied.

Types of Linear Regression:

1. **Simple Linear Regression**: Involves only one independent variable and one dependent variable. It fits a straight line to the data.
2. **Multiple Linear Regression**: Involves more than one independent variable. It predicts the dependent variable based on several factors.

Regression is used when your target variable is continuous and a value needs to be predicted.

Examples of regression problems include:

- Estimating sales and price of a product
- Predicting the score of a team
- Predicting the weather
- Sales forecasting

Machine Learning Algorithms

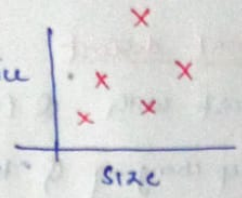
Day 1: Linear Regression {Supervised Learning Algorithms}

↳ It is used to predict the value of a variable based on the value of another variable.

- The variable you want to predict is called dependent variable.
- The variable you are using to predict the other variable's value is called Independent variable

Example Let's take house price

Size (feet)	Price ($1000s)
2104	400
1023	232
1503	315
0853	178
⋮	⋮



Training set

↓

Learning Algorithm

Size → [Hypothesis] → Price estimate

Q. How to represent 'h'?

$h(x) = \theta_0 + \theta x$

* Here we have single input-feature, but we can have multiple input features

for ex: no. of bedrooms, size

$h(x) = \theta_0 + \theta x_1 + \theta x_2$

$$h(x) = \sum_{j=0}^{2} \theta_j x_j$$

where $x_0 = 1$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \qquad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

Notations:

- θ = parameters
- m = no. of training examples
  # Rows of above table
- x = "inputs" {features}
- y = "output" {target variable}
- (x, y) ⇒ training example

$(x^{(i)}, y^{(i)})$ → i^{th} training example

n = features = {2 (size, no of bedrooms)

ex $x_1^{(1)} = 2104$

$x_1^{(2)} = 1023$

Choose θ such that $h(x)$ is close to y for training ex.

$h_\theta(x) = h(x)$ {some thing}

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\underbrace{(h_\theta(x) - y)^2}_{\text{minimizing difference}} \quad \text{minimize } \theta$$

* Gradient descent

Start with θ (say $\theta = \vec{0}$)

Keep changing θ to reduce $J(\theta)$

α → learning rate

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \qquad (j = 0, 1, 2)$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2}(h_\theta(x) - y)^2$$

chain rule

$$\Rightarrow 2 \cdot \frac{1}{2}(h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j}(h_\theta(x) - y)$$

$$= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j}(\theta_0 x_0 + \underbrace{\theta_1 x_1}_{x_1} \cdots \theta_n x_n - y)$$

if $j = 1$

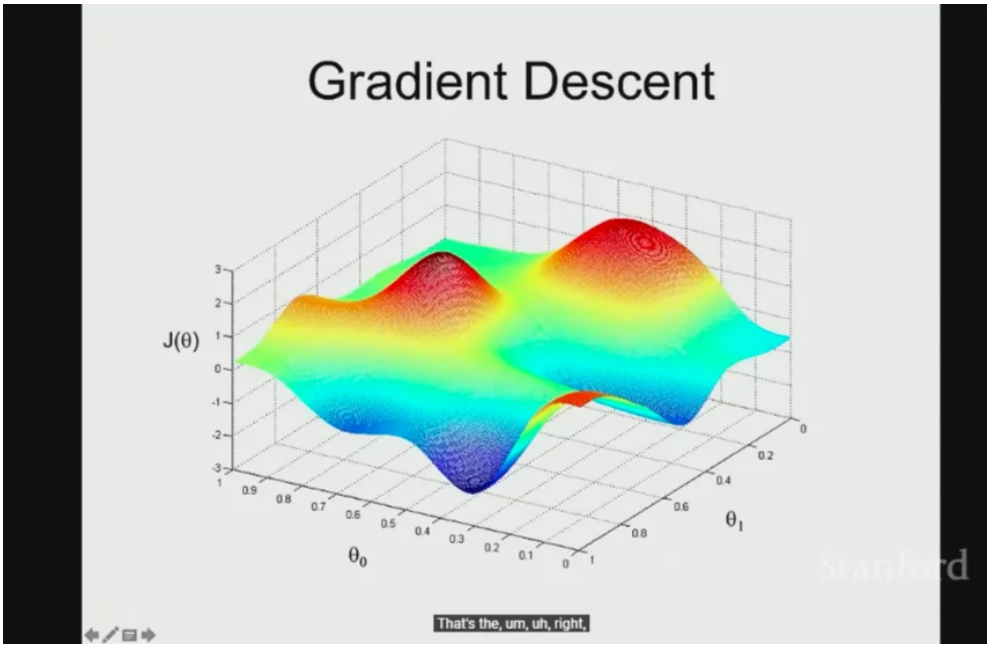
$$\Rightarrow (h_\theta(x) - y) \cdot x_j$$

others does not depend.

So partial derivative of others will be zero

Repeat until convergence

$$\theta_j = \theta_j - \alpha \underbrace{\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}}_{\frac{\partial}{\partial \theta_j} J(\theta)}$$



We are trying to take baby steps so that we can reach from highest point to lowest