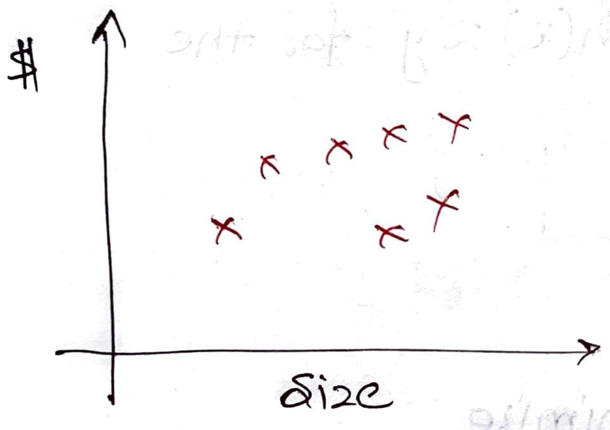
CS229: Linear Regression

└ Supervised Learning

$X \to y$

Size	# Bedrooms	Price ($)
2104	3	400
1416	2	232
1534	3	315
852	2	178
⋮	⋮	⋮



TRAINING SET

↓

LEARNING ALGORITHM

↓

$\boxed{\text{Size}}$ ⟶ Function (hypothesis) or h ⟶ $\boxed{\text{Estimated Price}}$

Q. How to represent h?

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

x_1 = Size

x_2 = # bedrooms

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

where $X_0 = 1$

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

θ = parameters

m = # training sample

X = inputs | features

y = output | target variable

$$h(x) = \sum_{j=0}^{2} \theta_j x_j \quad \text{where } X_0 = 1$$

Choose θ such that $h(x) \approx y$ for the training samples

$$\boxed{h_\theta(x) = h(x)}$$

Ordinary least Squares

$$(h_\theta(x) - y)^2 \rightarrow \text{minimize}$$

$$\min_\theta \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}))^2$$

$$\min_\theta \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}))^2$$

for easier derivative

Cost function

$$J(\theta) = \min_\theta \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

GRADIENT DESENT :

1. Start with some θ (say $\theta = \vec{0}$) [zero vector]
2. Keep changing the θ to reduce $J(\theta)$.

$$\boxed{\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta)}$$

$\rightarrow \alpha$ = learning rate

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

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

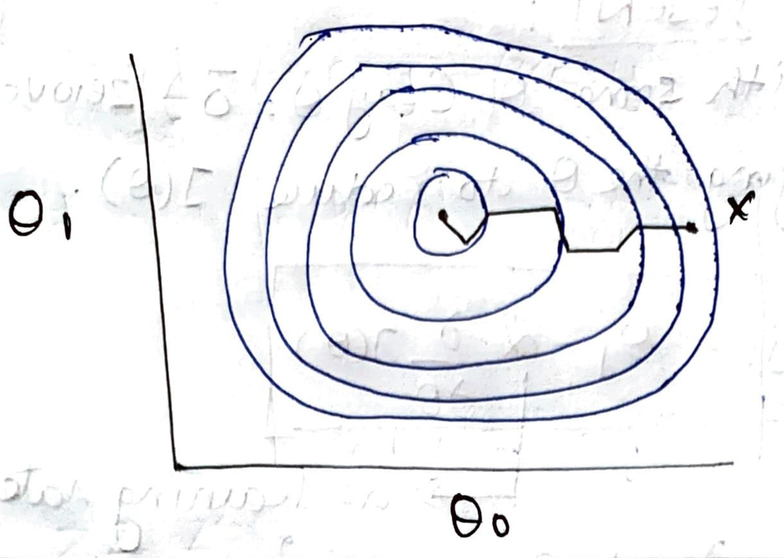
$$\Rightarrow (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \underbrace{(\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n - y)}$$

$$* \quad \frac{\partial}{\partial \theta_j} \theta_j x_j = x_j$$

$$\boxed{\theta_j := \theta_j - \alpha (h(x) - y) \cdot x_j}$$

$$\underbrace{\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}}_{\text{Rebeat until convergence}}$$

for ($j = 0, 1, \dots n$)



BATCH GRADIENT DESCENT

↳ you look at entire dataset as a batch of data

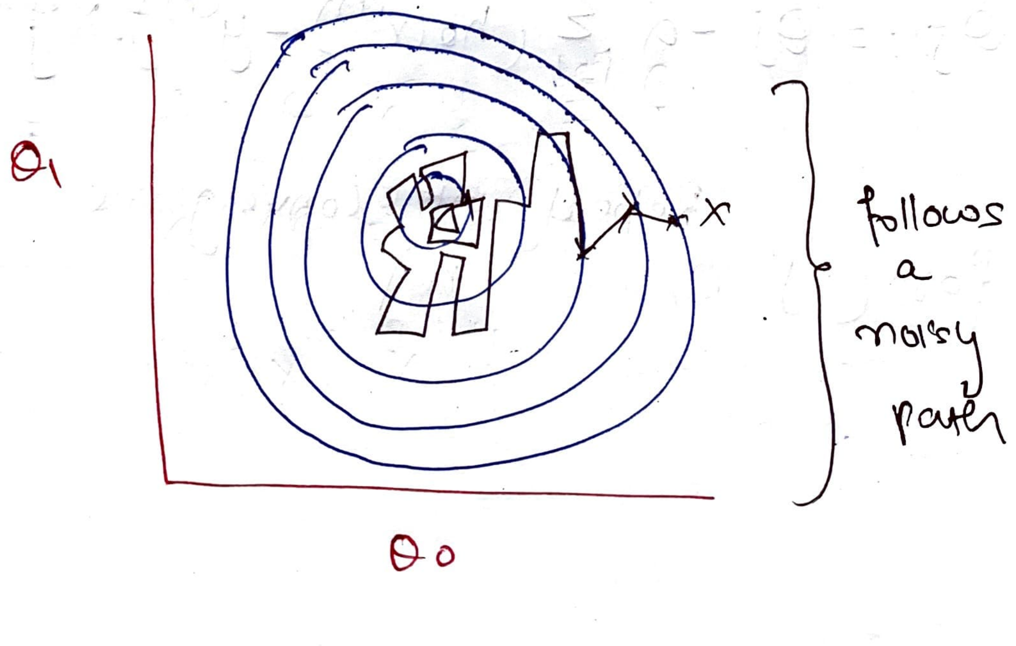
Repeat {

for j = 1 to m {

$\theta_j := \theta_j - \alpha \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)}$

STOCHASTIC GRADIENT DESCENT

↳ using single samples



NORMAL EQUATION

└ Only for linear Regression

└ Reaches Global optimum in onestep.

$\nabla_\theta J(\theta)$ — derivative

$\theta \in \mathbb{R}^{n+1}$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)}\right)^2$$

$X\theta$ [Design matrix]

$$X\theta = \begin{bmatrix} \text{---} (x^{(1)})^T \text{---} \\ \text{---} (x^{(2)})^T \text{---} \\ \vdots \\ \text{---} (x^{(m)})^T \text{---} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

$$= \begin{bmatrix} x^{(1)T}\theta \\ x^{(2)T}\theta \\ \vdots \\ x^{(m)T}\theta \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) \\ h_\theta(x^{(2)}) \\ \vdots \\ h_\theta(x^{(n+1)}) \end{bmatrix}$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$$J(\theta) = \frac{1}{2} (X\theta - y)^T (X\theta - y)$$

$$X\theta - y = \begin{bmatrix} h(x^{(1)}) - y^{(1)} \\ \vdots \\ h(x^{(m)}) - y^{(m)} \end{bmatrix}$$

$\because z^T z = z^2$

$$(X\theta - y)^T (X\theta - y)$$

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2} (X\theta - y)^T (X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta (\theta^T X^T - y^T)(X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta \left[\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y \right]$$

$$\{ (ax-b)(ax-b) = a^2x^2 - axb - bax + b^2 \}$$

$$= \frac{1}{2} \left[X^T X\theta + X^T X\theta - X^T y - X^T y \right]$$

$$= X^T X\theta - X^T y \overset{set}{=} 0$$

$$X^T X\theta = X^T y$$

$$\boxed{= (X^T X)^{-1} X^T y} \rightarrow \text{Normal equation}$$