★ Week 1 ★

Sadguru

Page No: 1

Date: / /

Course org — Andrew Ng

① Model representation — linear regression

m → no of training examples df.shape( ) → (m,)

x's → Input variable / features

y's → output variable / target value.

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

Training Set

↓

Learning Algo

↓

size of house (x) → h → price (estimated y)

hypothesis its a fn

$$h(x) = \theta_0 + \theta_1 x$$

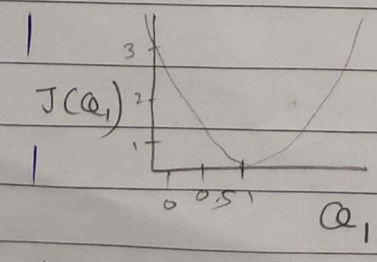
$$c + mx$$

now, minimize $J(\theta_0, \theta_1) \frac{1}{2m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$ → cost fn

$h_\theta(x)$ — for fixed θ_1, this is function of x

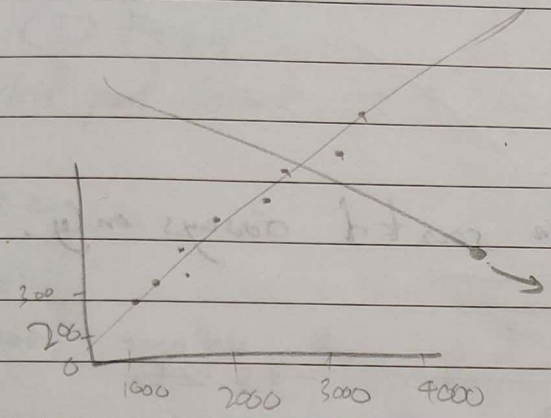
$J(\theta_1)$ — function of parameter θ_1

we considered $\theta_0 = 0$ & for each θ_1 we calculated $J(\theta_1)$

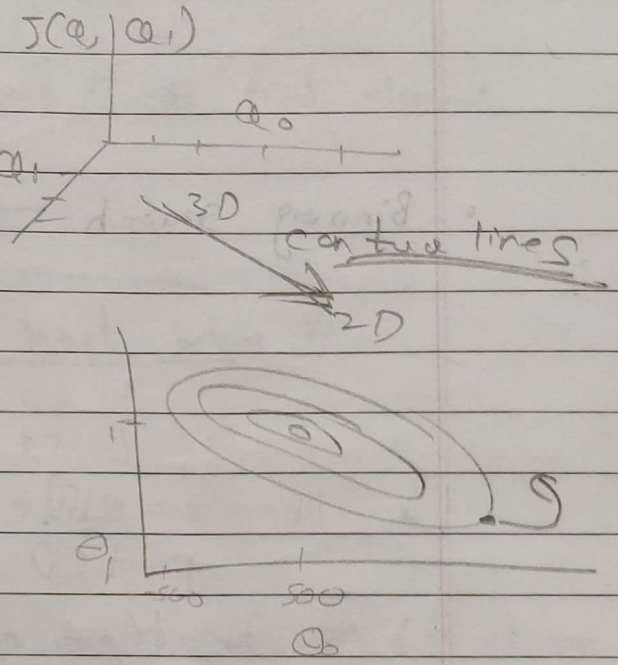


nothing is fixed here.

$h_\theta(x)$

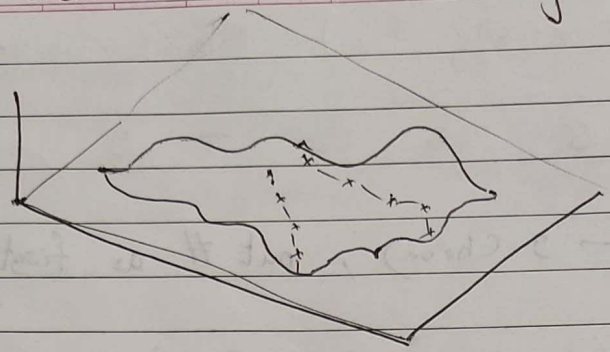


$J(\theta_0, \theta_1)$

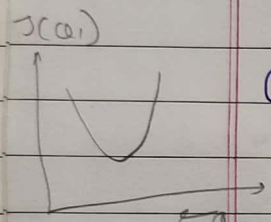


Gradient Descent

Gradient Descent Algo.



Depending on starting point, the local minima changes.



$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j=0 \text{ \& } j=1)$$

α : learning rate

$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$: derivative term.

First calculate RHS & update θ_0, θ_1 simultaneously.

Correct: simultaneous update

$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := temp0$

$\theta_1 := temp1$

Incorrect

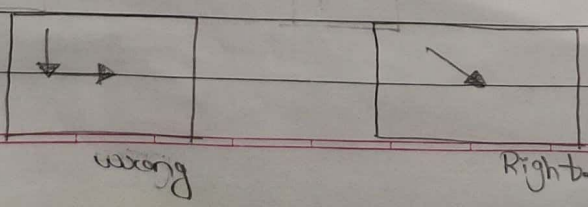
$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\theta_0 := temp0$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

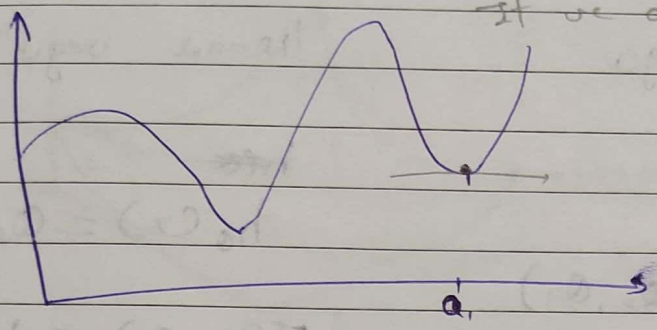
$\theta_1 := temp1$

We want to move in steepest direction in each step, thats why we want simultaneous update.



Gradient Descent Intution - - -

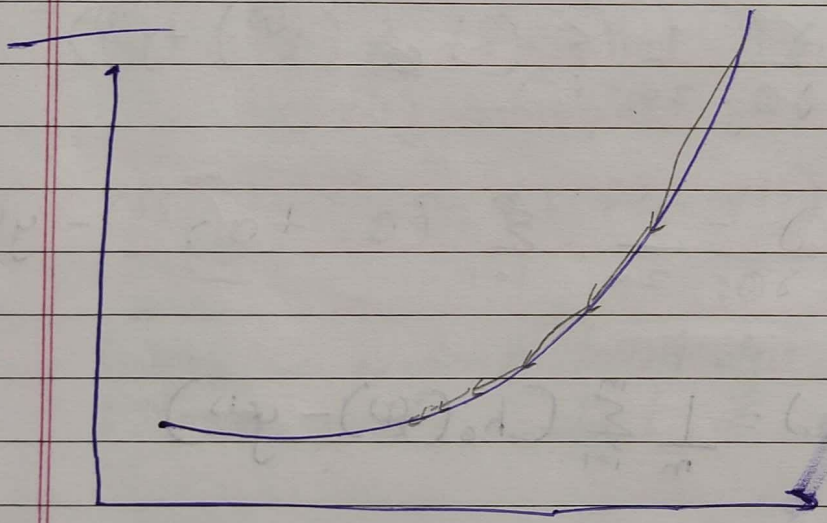
Sadguru
Page No: 4
Date: / /



$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

$\frac{d}{d\theta_1} J(\theta_1) \to 0$

no change in θ_1.



derivation term ↓'s,
thats why step ↓'s

so, don't have to chang α over time.

-- @ Gradient Descent for Linear Regression — —

Gradient Descent algo

repeat until convergence

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for $j=1$ & $j=0$) $\rightarrow \theta_0, \theta_1$

linear regression model.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

θ_0 $j=0$: $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$

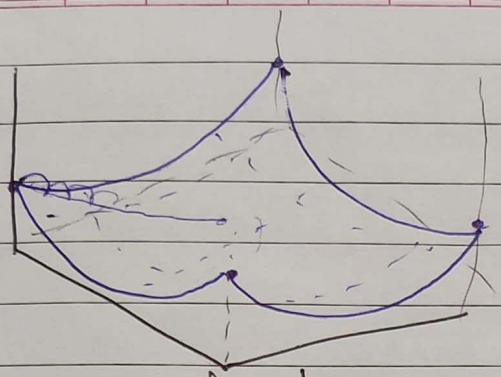
θ_1 $j=1$: $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$

Gradient descent for simple linear regression

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

update θ_1 & θ_0 simultaneously



convex function → for linear regression only one global minimum & no local minimum

"Batch" Gradient Descent

"Batch" : Each step of gradient descent uses all training examples.

$$\sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})$$

There are gradient descent algos which are not batch, looks in specific subset of training set.

★ Normal equation methods :-

Directly solves for θ_0, θ_1

Gradient Descent scales better to larger data sets than a normal equation method.

-- @ Matrices and Vectors - -

linear algebra

Sadguru

Page No: 7

Date: / /

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$A_{11} = 1$

$A_{12} = 2$

4 × 2 matrix

or

$\mathbb{R}^{4\times 2}$

Vector matrix : n × 1 matrix

$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$\mathbb{R}^4$