Machine Learning

Arthur Samuel:-

Field of study that gives computer the ability to learn without being explicitly programed

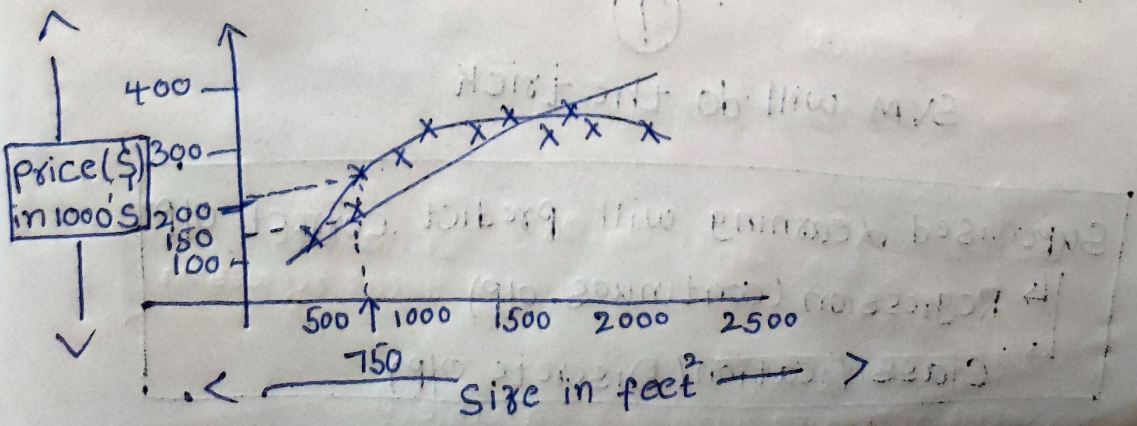
Tom Mitchell:-

A computer program is said to learn from experience E with respect to some task T & some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine Learning Algorithms

- Supervised learning
- Unsupervised learning
- Reinforcement learning, recommender systems

Supervised Learning

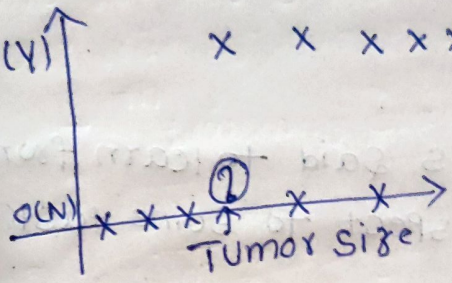


- Regression problem (Predict continuos valued o/p)

Here in this case its (Price)

- Breast Cancer (malignant, benign) [classification problem]

eg:-

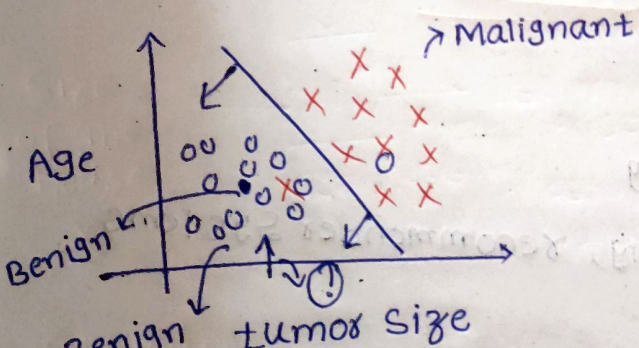


Classification
Discrete valued
o/p (0 or 1)

Here we considered "tumor size" as an attribute

So this is just for one attribute.

Considering more than 1 attribute



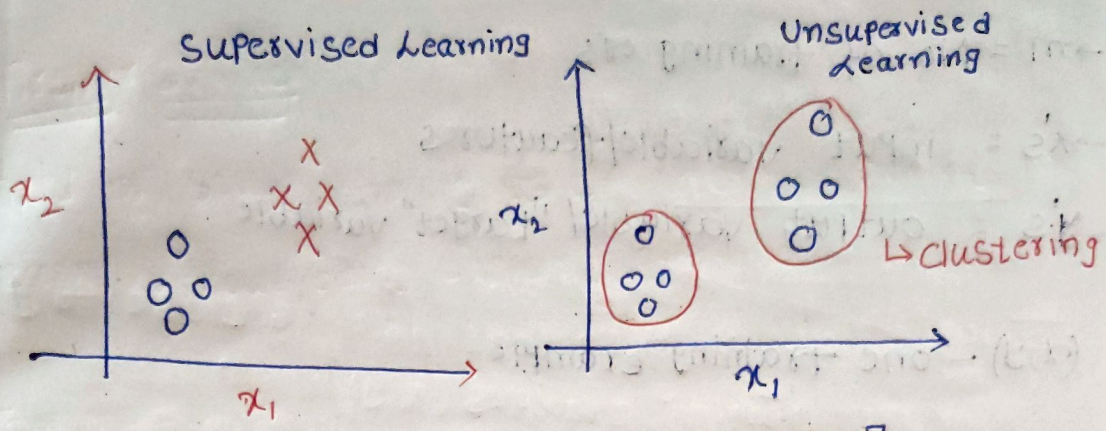
⇒ If there are infinite no of feautures

(?)

SVM will do the trick

Supervised Learning will predict correct o/p
- Regression (continuos o/p)
- Classification (Discrete o/p)

Unsupervised Learning

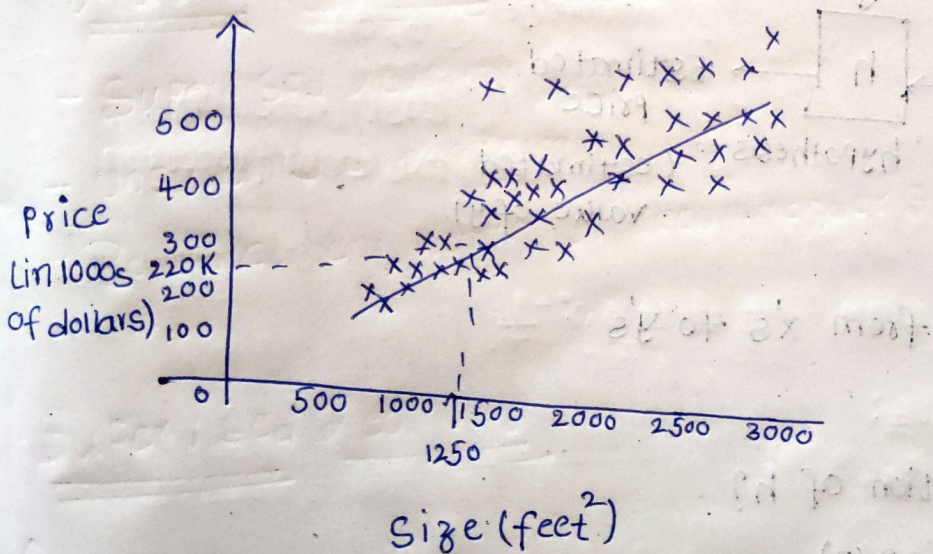


[Labels are present]

[No Labels]

↳ Given a dataset & you should find the structure b/w the data points

Model Representation



Supervised learning

Given the "right answer" for each example in the data

Regression Problem

Predict real-valued O/P

↳ continues

Notation

→ m = No of training eg's

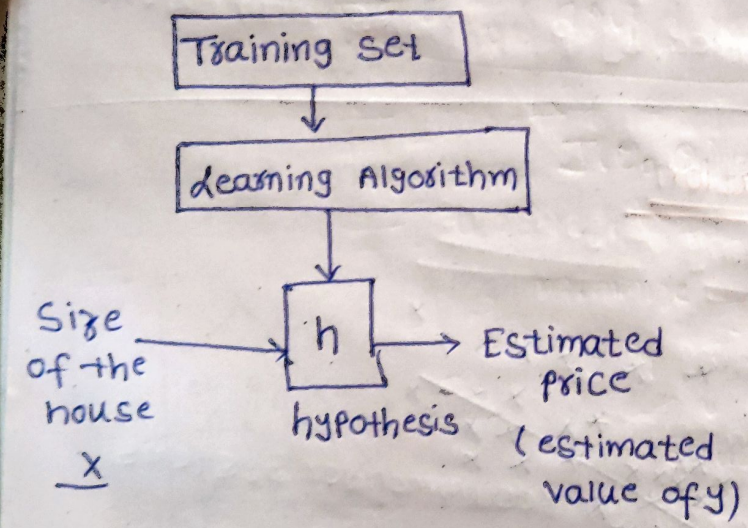
→ x's = input variable/features

→ y's = output variable/ "target" variable

(x, y) - one training example

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

How we build a model

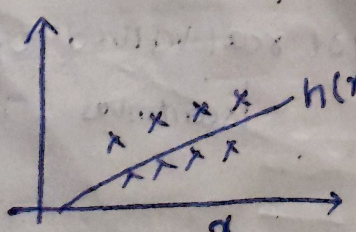


h maps from x's to y's

Representation of h?

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

shorthand : $h(x)$



Univariate linear regression

Linear regression with one variable(x)

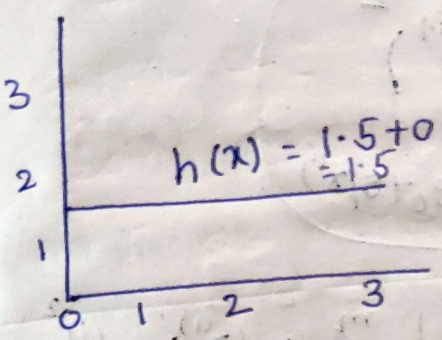
Cost function [This will decide the best fit line in our data]

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

θ_i's : Parameters

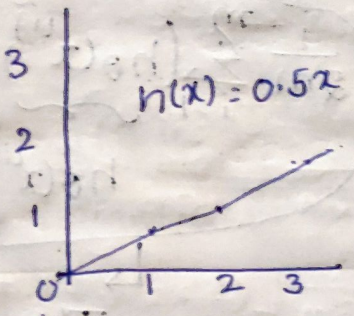
How to choose θ_i's?

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



$\theta_0 = 1.5$

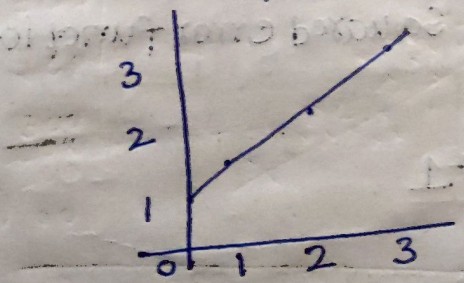
$\theta_1 = 0$



$\theta_0 = 0$

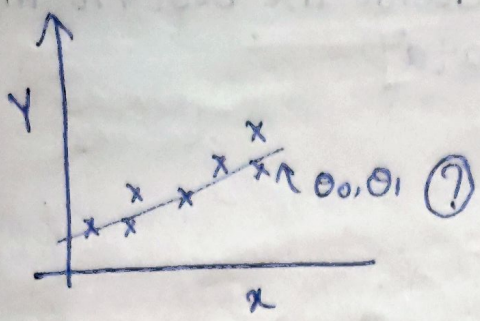
$\theta_1 = 0.5$

$h(x) = 1 + 0.5x$



$\theta_0 = 1$

$\theta_1 = 0.5$



Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y)

minimize $\theta_0 \theta_1$

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

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

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

minimize $J(\theta_0, \theta_1)$
θ_0, θ_1

↓ cost function

Squared error function

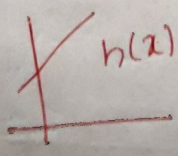
Cost function - Intution - 1

Hypothesis:

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

Parameters

θ_0, θ_1



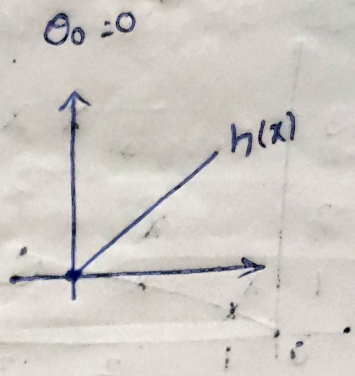
Cost function

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

Goal: minimize $J(\theta_0, \theta_1)$

Simplified form

$h_\theta(x) = \theta_1 x \quad (\theta_0 = 0)$



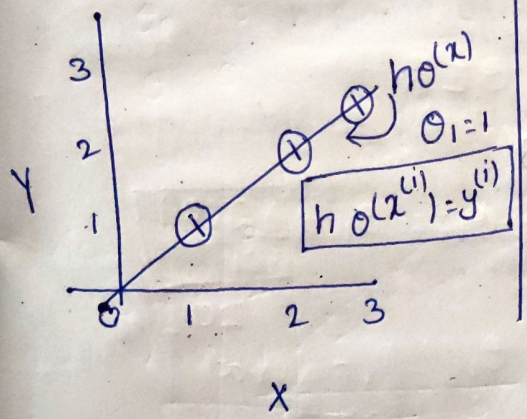
θ_1

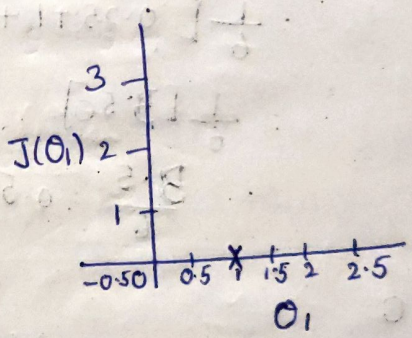
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{2m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$h_\theta(x^{(i)}) \rightarrow \theta_1 x^{(i)}$

minimize $J(\theta_1)$

$h_\theta(x)$ (for fixed θ_1, function of x)	$J(\theta_1)$ (function of Parameter θ_1)





$J(1) = 0$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}]^2$$

~~$\theta_1 = 0.5$~~

$$= \frac{1}{2m} \sum_{i=1}^{m} [\theta_1 (x^{(i)} - y^{(i)}]^2$$

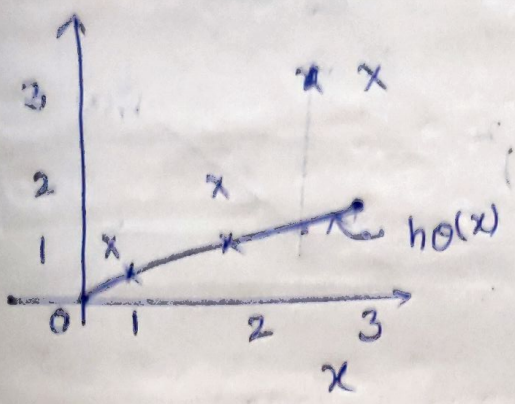
$$= \frac{1}{2m} [0^2 + 0^2 + 0^2]$$

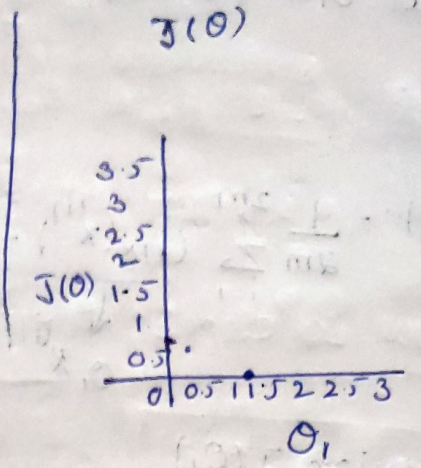
$J(\theta_1) = 0$

$J(1) = 0$

$\theta_1 = 0.5$

$h_\theta(x) = 0.5x$





$$J(0.5) = \frac{1}{2m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]^2$$

$$= \frac{1}{2m}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right]$$

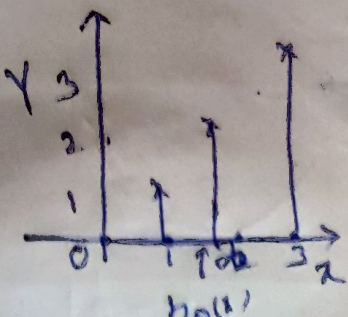
$$= \frac{1}{6}\left[0.25 + 1 + 2.25\right]$$

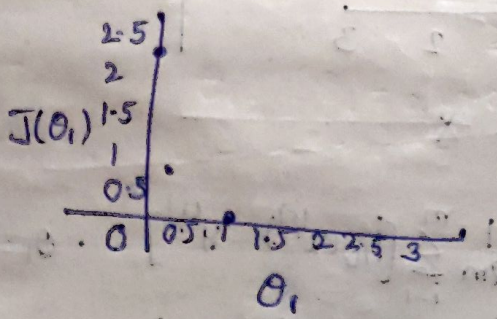
$$= \frac{1}{6}\left[3.50\right]$$

$$= \frac{3.5}{6} = 0.58$$

$\theta_1 = 0$

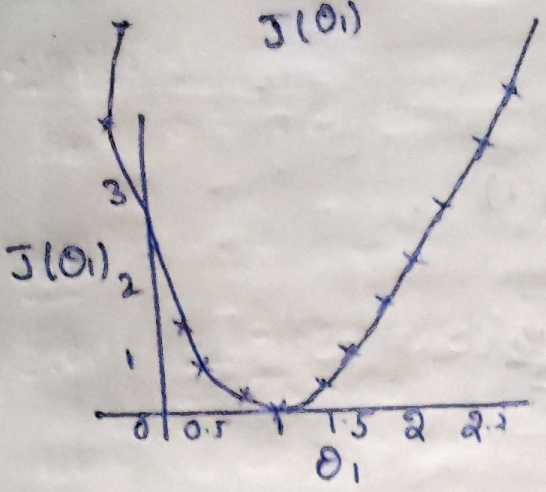
$h(0) = \theta(x) = 0$





$$\therefore J(0) = \frac{1}{6}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]^2$$

$$= \frac{1}{6}\left[1 + 4 + 9\right] = \frac{14}{6} = 2.33$$



Gradient Descent :- [Minimize the cost function & other facts as well]

$\Rightarrow J(\theta_0, \theta_1)$

want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline

Start with some θ_0, θ_1 (say $\theta_0 = 0$, $\theta_1 = 0$)

Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Gradient descent algorithm

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

Simultaneously update θ_0 & θ_1

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

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

$\theta_0 = temp0$

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

$\theta_1 := temp1$

$\theta_0 = 1$ & $\theta_1 = 2$

$\theta_j := \theta_j + \sqrt{\theta_0 \theta_1}$ (for $j = 0$ & $j = 1$)

$\theta_0 := \theta_0 + \sqrt{2}$

$\boxed{\theta_0 := 1 + \sqrt{2}}$

$\theta_1 := \theta_1 + \sqrt{\theta_0 \theta_1}$

$\boxed{\theta_1 := 2 + \sqrt{2}}$

Gradient descent algorithm

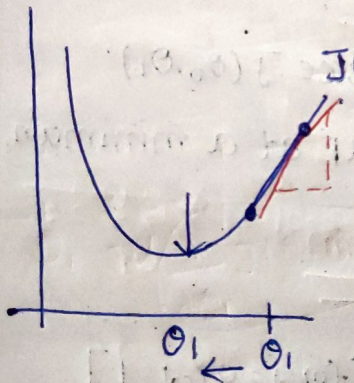
Repeat until convergence

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$ [simultaneously update θ_0 & θ_1]

α — learning rate

$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ — derivative

eg:-

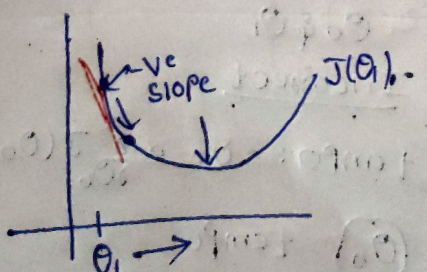


$J(\theta_1)$ $\theta_1 \in \mathbb{R}$

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

≥ 0

$\theta_1 := \theta_1 - \alpha(+ve)$

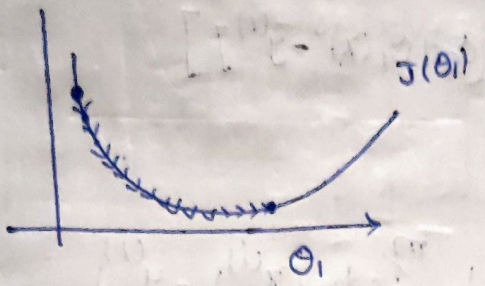


$\theta_1 := \theta_1 - \alpha(-ve)$

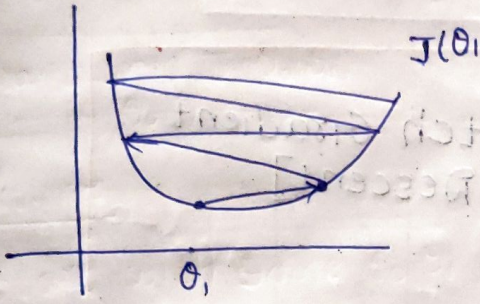
$\theta_1 := \theta_1 - \alpha(-ve)$

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

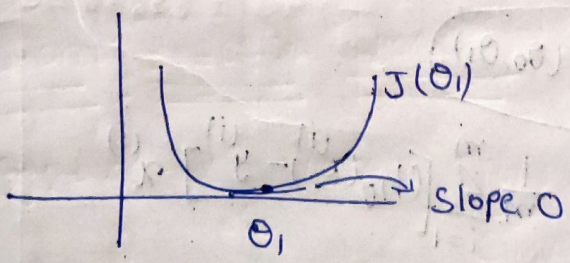
α is too small, gradient descent can be slow



α is too large gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



If θ_1 is at local minima then what happens?



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

$\therefore \theta_1 := \theta_1 - \alpha(0)$

$\theta_1 := \theta_1$ | Remains unchanged

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

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

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

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

Gradient algorithm [Batch Gradient Descent]

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

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

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

$\hookrightarrow \frac{1}{m} \sum_{i=1}^{m} \left[(h_\theta(x^{(i)}) - y^{(i)} \right] x^{(i)}$

Linear Algebra

Matrix Addition:-

$$\begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 4 & 0.5 \\ 2 & 5 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 0.5 \\ 4 & 10 \\ 3 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 4 & 0.5 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} ? \end{bmatrix} \quad \times$$

3×2 $\qquad$ 2×2

Cant perform; should be of square matrices

$$3 \times \begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 6 & 15 \\ 9 & 3 \end{bmatrix} \checkmark$$

$$\begin{bmatrix} 4 & 0 \\ 6 & 3 \end{bmatrix} / 4 = \begin{bmatrix} 1 & 0 \\ 3/2 & 3/4 \end{bmatrix} \checkmark$$

Combination of operands

$$3 \times \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} - \begin{bmatrix} 3 \\ 0 \\ 2 \end{bmatrix} / 3 = \begin{bmatrix} 3 \\ 12 \\ 6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ \frac{2}{3} \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 12 \\ \frac{31}{3} \end{bmatrix} \checkmark$$

Matrix multiplication

eg:-

$$\begin{bmatrix} 1 & 3 \\ 4 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1\times1+3\times5 \\ 4\times1+0\times5 \\ 2\times1+1\times5 \end{bmatrix}$$

$3\times2 \quad 2\times1$

↳ Res: 3×1

$$= \begin{bmatrix} 16 \\ 4 \\ 7 \end{bmatrix}_{3\times1}$$

House sizes

2104
1416
1534
852

$$h_\theta(x) = -40 + 0.25x$$

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \begin{bmatrix} -40 \\ 0.25 \end{bmatrix} = \begin{bmatrix} -40+2104\times0.25 \\ -40+1416\times0.25 \\ -40+1534\times0.25 \\ -40+852\times0.25 \end{bmatrix}$$

$\rightarrow h_\theta^{(2104)}$

$\rightarrow h_\theta(1416)$

Multivariate Linear Regression

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots \theta_n x_n$$

For convinience of notation define $x_0 = 1$

$$x_0^{(i)} = 1$$

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \& \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots \theta_n x_n$$

$$\boxed{h_\theta(x) = \theta^T x}$$

$$h_\theta(x) = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \begin{bmatrix} x_0 x_1 x_2 x_3 \cdots x_n \end{bmatrix} = \theta^T x$$

$$[\theta_0\ \theta_1 \cdots \theta_n]$$

$(1 \times n+1)$

$$h_\theta(x) = \begin{bmatrix} \theta_0 \theta_1 \theta_2 \cdots \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \theta_0 x_0 + \theta_1 x_1 + \cdots \theta_n x_n \end{bmatrix}$$

$1 \times (n+1)$ $\quad$ $(n+1) \times 1$ $\quad$ 1×1

Gradient descent for multiple features

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \cdots \theta_n$

Assume a vector θ

Cost function :-

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

Gradient descent

$$\theta_j : \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\underbrace{\theta_0, \cdots \theta_j}_{\theta})$$

~~$\theta_1 : \theta_1 - \alpha \frac{\partial}{\partial \theta_j}$~~

$$\boxed{\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}}$$

Gradient Descent in practice I - Feauture Scaling

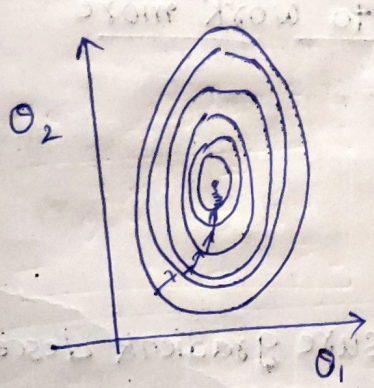
Feauture Scaling

Idea: Make sure feautures are on a similar scale

Eg x_1 = size(0-2000 feet^2)

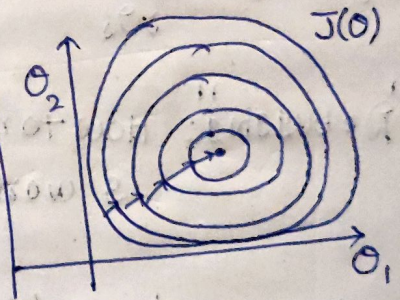
x_2 = no of bedroom (1-5)

If you implement with these then gradient descent will take huge time to reach global minima



$x_1 = \frac{\text{size(feet}^2\text{)}}{500}$ $0 \le x_1 \le 1$

$x_2 = \frac{\text{no of bedrooms}}{5}$ $0 \le x_2 \le 1$



Feauture Scaling

Get every feauture approximately a $-1 \le x_i \le 1$

$x_0 = 1$

$0 \le x_1 \le 3$ ✓

$-2 \le x_2 \le 0.5$ ✓

$-100 \le x_3 \le 100$ X

$-0.001 \le x_4 \le 0.0001$ X

-3 to 3 ✓

$-\frac{1}{3}$ to $\frac{1}{3}$ ✓

Mean Normalization [Do not apply to $x_0 = 1$]

$$x_1 = \frac{\text{Size} - \mu_i}{2000}$$

$$x_2 = \frac{\text{bedrooms} - 2}{5}$$

$x_1 \leftarrow \frac{x_1 - \mu_i}{S_1}$; μ_i avg value of x in training set

S_1 Range

$x_2 \leftarrow \frac{x_2 - \mu_i}{S_2}$

approx :- $-0.5 \leq x_1 \leq 0.5$ & $-0.5 \leq x_2 \leq 0.5$

[Should need not be exact]

How to make gradient descent to work more accurately

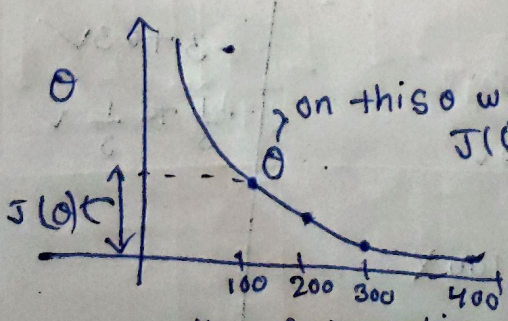
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- "Debugging": How to make sure gradient descent is working correctly

- How to choose learning rate α.

Make sure gradient descent is working correctly

Try to make the plot b/w No of iteration



on this so we will evaluate $J(\theta)$

For 100 iteration we will get the value of θ & will evaluat

J(θ) at that θ point & the height is J(θ)

The plot is showing as No of interations ↑se the value of J(θ) is ↓ing

- For sufficiently small α, J(θ) should decrease on every iteration
- But if α is too small, gradient descent can be slow to converge

Summary

- If α is too small: slow convergence
- If α is too large: J(θ) may not decrease on every iteration; may not converge

To choose α, try

... 0.001, 0.01, , 0.1, , 1, ...

Logistic Regression

- Classification
- $y \rightarrow$ Discrete o/p

$y \in \{0,1\}$

$\Rightarrow h_\theta(x) \geq 0.5$ predict $y = 1$
$h_\theta(x) < 0.5$ predict $y = 0$

- Logistic Regression $[0 \leq h_\theta(x) \leq 1]$

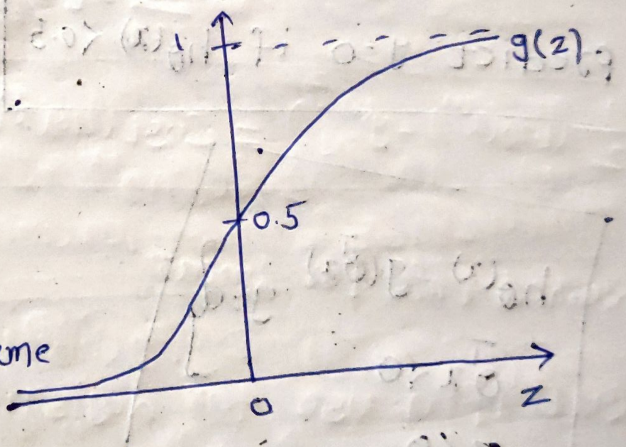
Hypothesis Representation

$$h_\theta(x) = g(\theta^T x)$$ ($\theta^T x \rightarrow z$)

$$g(z) = \frac{1}{1+e^{-z}}$$

- sigmoid fnc
- logistic func

} same



Interpretation of hypothesis o/p

$h_\theta(x)$ = estimated probability that $y=1$ on i/p x

eg:- $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorsize} \end{bmatrix}$

$y = 1$

$h_\theta(x) = 0.7$

$$h_\theta(x) = P(y=1 \mid x, \theta)$$ "probability that $y=1$, given x, parameterized by θ"

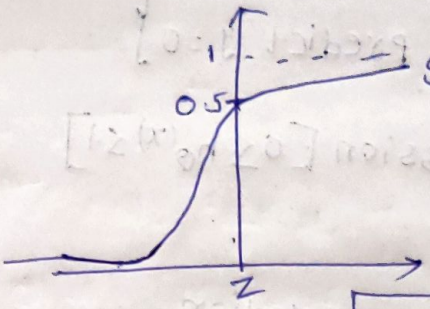
$P(y=0|x;\theta) = 1 - P(y=1|x;\theta)$

$\Rightarrow \boxed{P(y=0|x;\theta) + P(y=1|x;\theta) = 1}$

Decesion Boundary

$h_\theta(x) = g(\theta^T x)$

$g(z) = \frac{1}{1+e^{-z}}$



Predict $y=1$ if $h_\theta(x) \geq 0.5$

predict $y=0$ if $h_\theta(x) < 0.5$

$g(z) \geq 0.5$ when $z \geq 0$

~~$g(z) <$~~

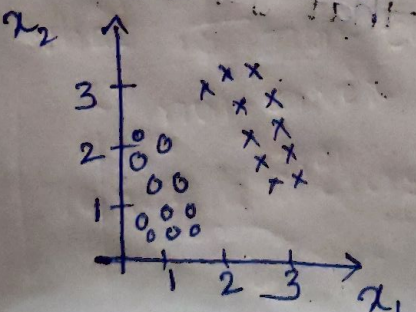
$y = 1$ when $\theta^T x \geq 0$

$h_\theta(x) = g(\theta^T x)$ $y=0$

$\theta^T x < 0$

$z < 0$

Decesion boundary illustration



$\rightarrow h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

$-3 \quad 1 \quad 1$

$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

$y = 1$?

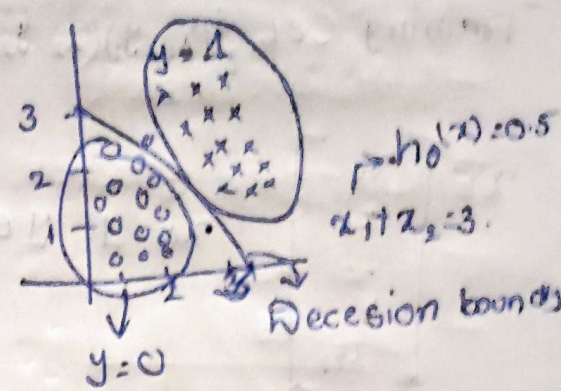
$y = 0$?

Predict $y=1$ if $z \geq 0$ i.e $\theta^T x \geq 0$ i.e

$$-3 + x_1 + x_2 \geq 0 \rightarrow y=1$$

$$-3 + x_1 + x_2 = 0$$

$$\boxed{x_1 + x_2 = 3}$$

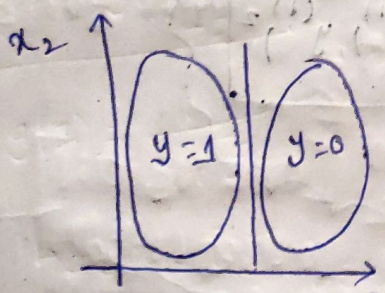


eg:- Features x_1 & x_2 & $\theta_0 = 5, \theta_1 = -1, \theta_2 = 0$, & so

$$h_\theta(x) = g(5 - x_1)$$

Decesion boundary of $h_\theta(x)$?

$\Rightarrow$



$y=1$; $5 - x_1 \geq 0$

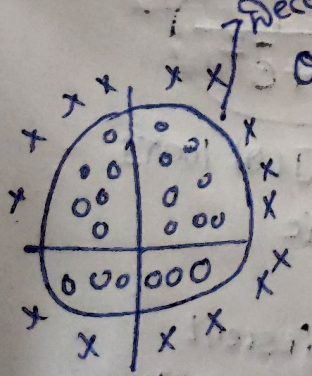
$x \leq 5$

$y=0$; $5 - x_1 < 0$

$\Rightarrow$ Non-linear decesion boundaries.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Decesion boundary



$$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$\therefore z = -1 + x_1^2 + x_2^2$

$\therefore y=1$; $x_1^2 + x_2^2 \geq 1$

$y=0$; $x_1^2 + x_2^2 < 1$

Logistic Regression cost function

Training Set $\{(x_1, y_1)(x_2, y_2), (x_3, y_3) \dots (x_m, y_m)\}$

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad x_0 = 1 \quad y \in \{0, 1\}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

How to choose θ ?

For Line Reg $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left[h_\theta(x^{(i)}) - y^{(i)} \right]^2$

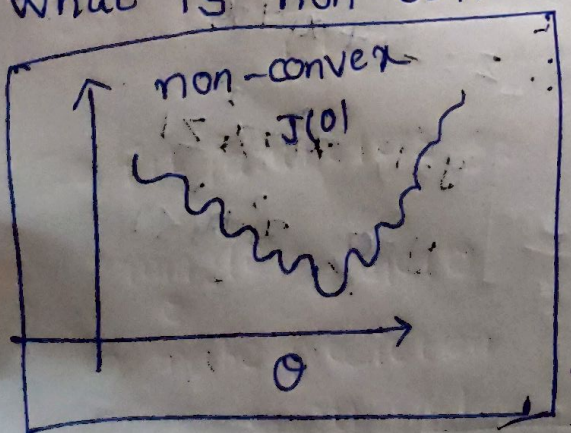
$$\frac{1}{2} \left(\left[h_\theta(x^{(i)}) - y^{(i)} \right]^2 \right) = Cost\left(h_\theta(x^{(i)}), y^{(i)}\right)^2$$

worked fine for linear

But if u use for logistic then

$\left[h_\theta(x) - y \right]^2$ will be "non-convex"

what is non-convex ?



WKT $h_\theta(x) = \frac{1}{1 + e^{-z}}$

then J(θ) looks like

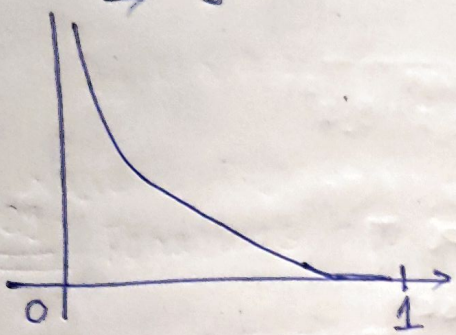
→ Gradient descent will not Converge

Logistic regression Cost function

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

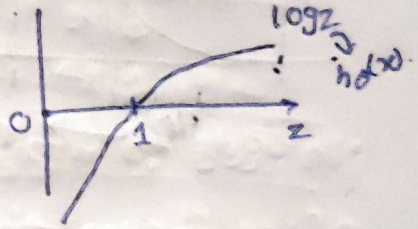
↳ Single training example

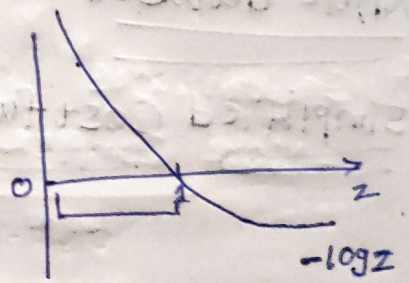
$\Rightarrow y = 1$



Fig

In general





if $y = 1$, then $h_\theta(x) = 1$

But as $h_\theta(x) \to 0$

$\text{Cost} \to \infty$

Interested b/w 0 & 1

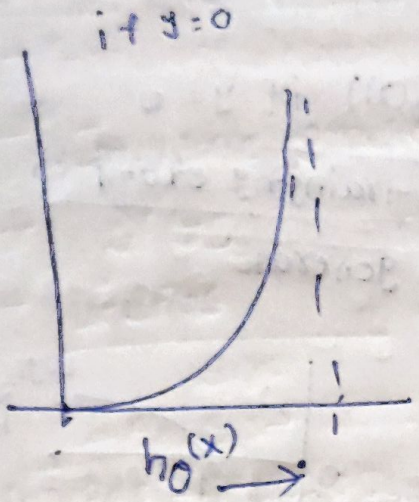
so it is as above in Fig

Captures intution that if $h_\theta(x) = 0$,

(Predict $P(y=1|x;\theta) = 0$ but $y=1$)

we'll penalize learning algorithm by a very large cost.

$\Rightarrow y=0$



whole dataset

simplified Cost function & Gradient descent

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{cost}(h_\theta(x), y)$$

$$\text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

↳ For $y=0$ or $y=1$

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

To fit parameters θ:

$\min_\theta J(\theta)$ ↗ Get θ

To make prediction

output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ $p(y=1|x;\theta)$

Giradient Descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]$$

Repat {

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

↑ simultaneously update all θ_j

}

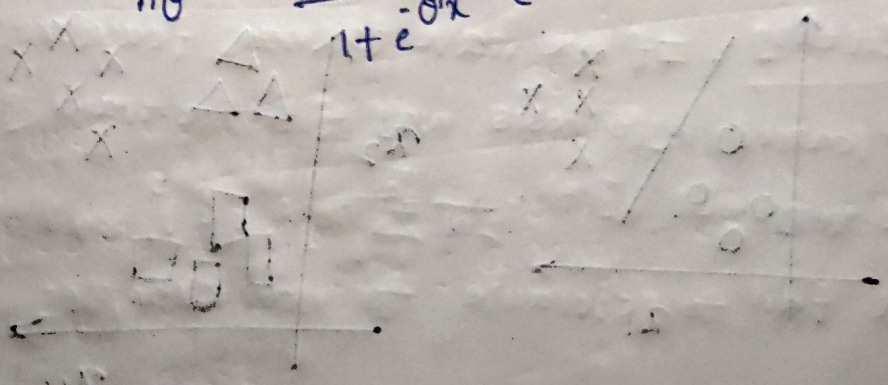
$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

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

↳ Exactly same as linear regression.

but $h_\theta(x) = \theta^T x$ [for linear regression]

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ [logistic] } Not same



Advanced optimization

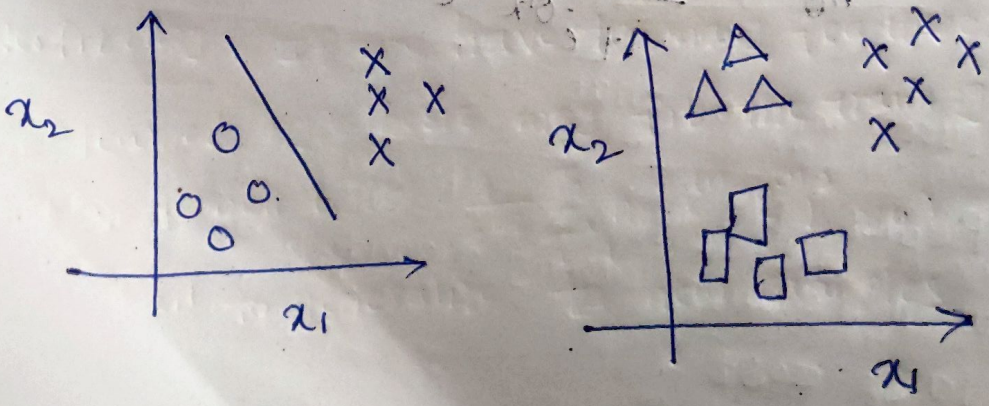
optimization algorithms

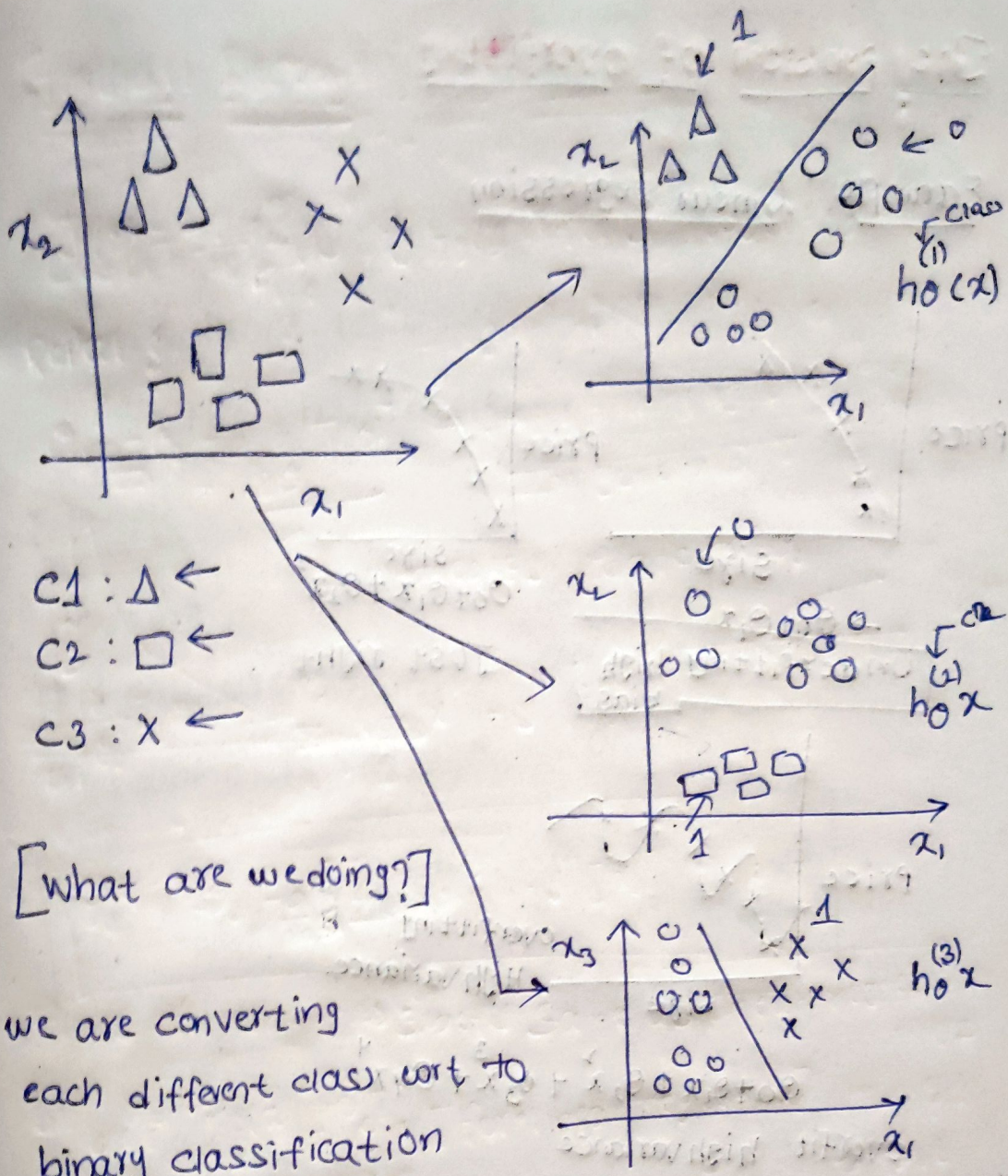
- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

↳ No need to manually pick α
Often faster than Grad. desc
→ Line searching algorithm will make the work for α

Disadvantages
- More Complex

Multiclass Classification [one Vs all]





C1 : Δ ←

C2 : □ ←

C3 : X ←

[What are we doing?]

we are converting each different class sort to binary classification problem

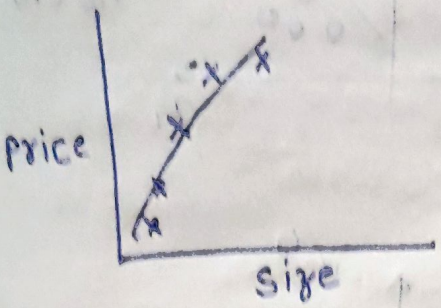
$h_\theta^{(i)}(x) = p(y = i | x; \theta)$ $(i = 1, 2, 3)$.

On a new i/p x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)} x$$

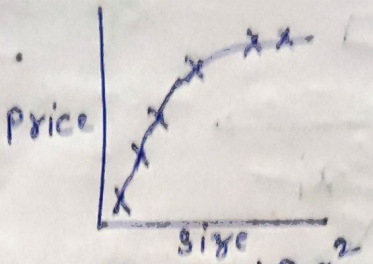
The problem of overfitting

Example: Linear regression



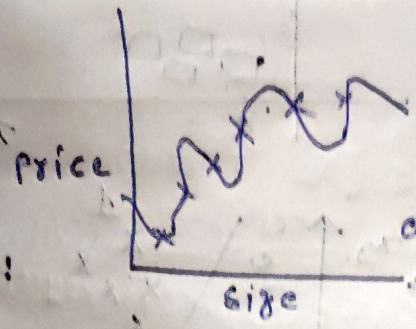
$\rightarrow \theta_0 + \theta_1 x$

underfitting high bias



$\theta_0 + \theta_1 x + \theta_2 x^2$

Just right



overfitting
high variance

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

overfit high variance

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} h_\theta(x^{(i)} - y^{(i)})^2 \approx 0$), but fail to generalize to new examples.

Regularization

- Small values for parameters $\theta_0, \theta_1, \cdots \theta_n$
- Less prone to overfitting
- "Simpler" hypothesis

Example

Housing

⇒ Features: $x_1, x_2, \cdots, x_{100}$

⇒ Parameters: $\theta_0, \theta_1, \theta_2, \cdots \theta_{100}$

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