MACHINE LEARNING

WEEK-1

- **Machine learning**
  - Grew out of work in AI
  - New capability for computers

Examples:

- Database mining
  - large datasets from growth of automation/web
  - Eg. web click data, medical records etc.
- Application can't program by hand
  - Eg. Autonomous helicopter, handwriting recognition, most of NLP, Computer Vision
- self customization programs
  - Amazon/Netflix product recommendations
- Understand human learning (Brain, real AI)

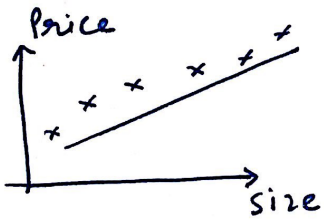
"A computer program is said to learn from experience E w.r.t some task T and some performance measure P, if its performance on T as measured by P, improves with experience E"

– Tom Mitchell (definition)

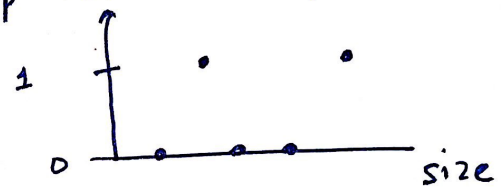
Types of Machine Learning

→ **Supervised learning**

– we are given a data set & already know what our correct o/p should look like, having the idea that there is a relationship between input & output.

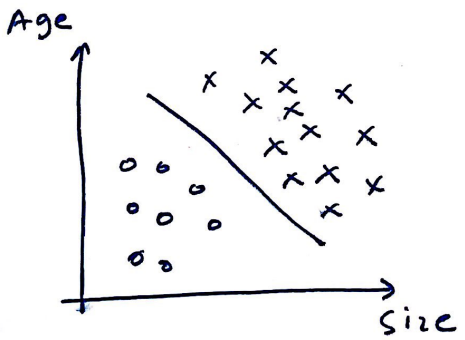


– (Regression)** continous valued variable



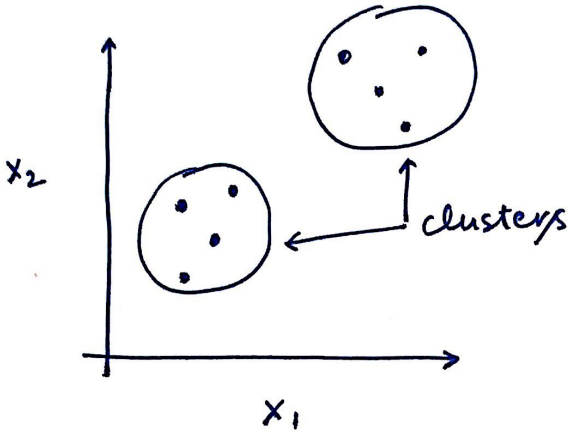
(Discrete valued o/p)

**– classification problem



→ Unsupervised Learning

Allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't know the effect of the variables.



→ Can you find structure or cluster in the data?

Eg:
- organize computing clusters
- Social network Analysis
- Market segmentation **
- Astronomical data Analysis

• LINEAR REGRESSION (1 variable) — MODEL

- Supervised learning : given the 'right answer' for each example in the data.

Training set of diff. housing prices

Size in ft^2 (x)	Price in '000's (y)
2104	460
1416	232
1534	315
852	178

(m) → no. of training examples

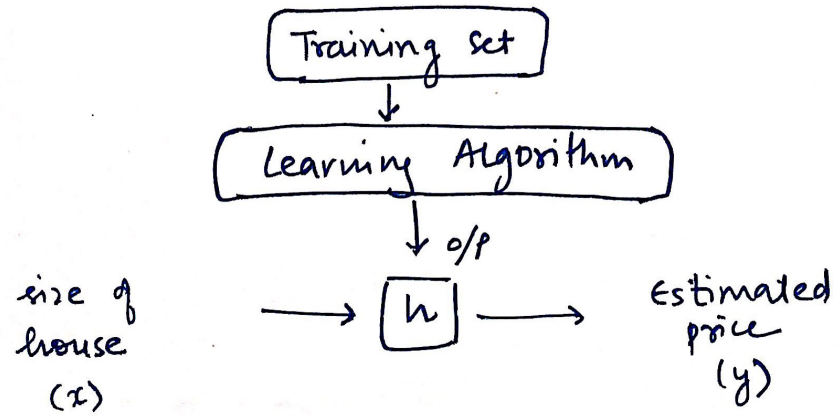
($x^{(i)}$'s) → Input Variable / features

($y^{(i)}$'s) → Output variable / 'target'

(x, y) :— one training example

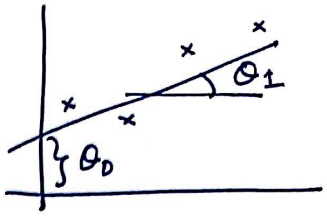
$(x^{(i)}, y^{(i)})$ → i^{th} training example

How does the algorithm work (supervised learning) ?



Hypothesis: (h) $\quad\quad$ θ_i : Parameters

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



Idea: choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y) : (minize the distance b/w data points)

$$\therefore \min_{\theta_0, \theta_1} \left[\frac{1}{2m} \sum_{i=1}^{m} \left(\underbrace{h_\theta(x^{(i)})}_{\text{hypothesized or on line or predicted value}} - \underbrace{y^{(i)}}_{\text{Actual}} \right)^2 = \underbrace{J(\theta_0, \theta_1)}_{\text{cost fn.}} \right]$$

↓

squared error function

θ_0, θ_1 → Parameters

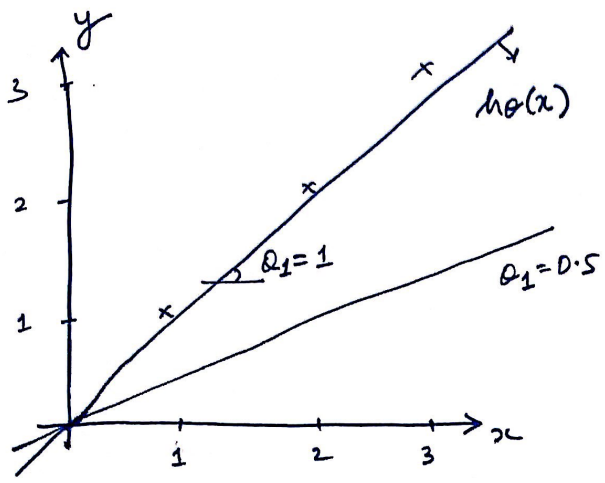
↳ slope and ~~grea~~ offset

Goal : $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Example 1 : $\theta_0 = 0$

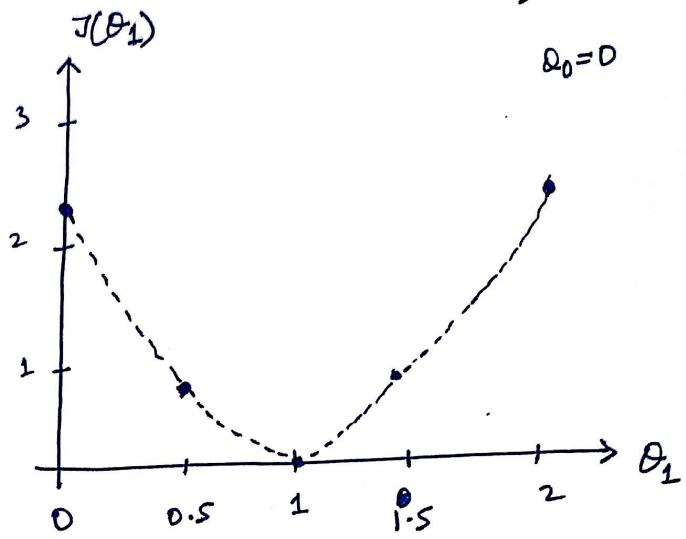
$h_\theta(x)$

hypothesis fn. is the fn. of x



$J(\theta_1)$

Cost fn. is the fn. of θ_1 ~ gradient



$$J(\theta_1) = \frac{1}{2m}(\theta^2 + 0^2 + 0^2) = 0$$

$$\Downarrow$$

$$J(\theta_1 = 1) = 0 \quad — \quad ①$$

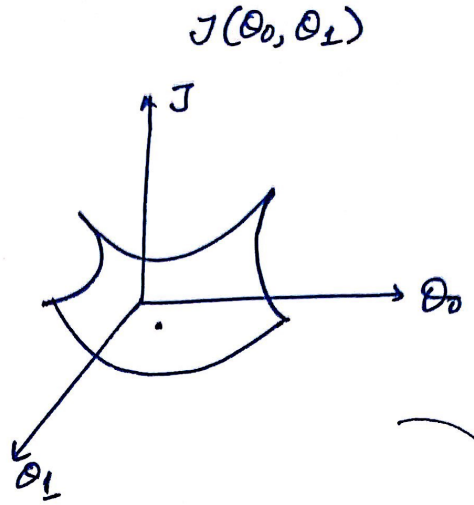
$$J(\theta_1 = 0.5) = \frac{1}{2.3}(0.5^2 + 1^2 + 1.5^2) \rightarrow J(\theta_1 = 1.5)$$

$$= 0.68 \quad — \quad ②$$

$$J(\theta_1 = 0) = 2.3 \rightarrow J(\theta_1 = 2)$$

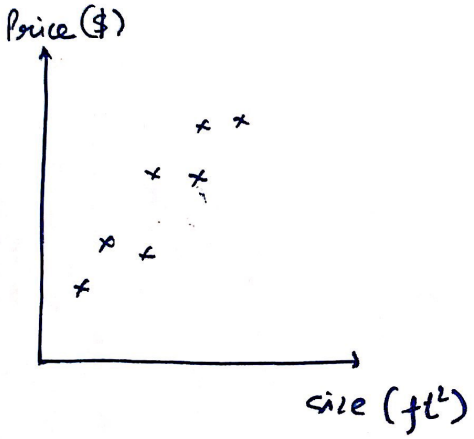
Example 2: $\theta_0 \neq 0$

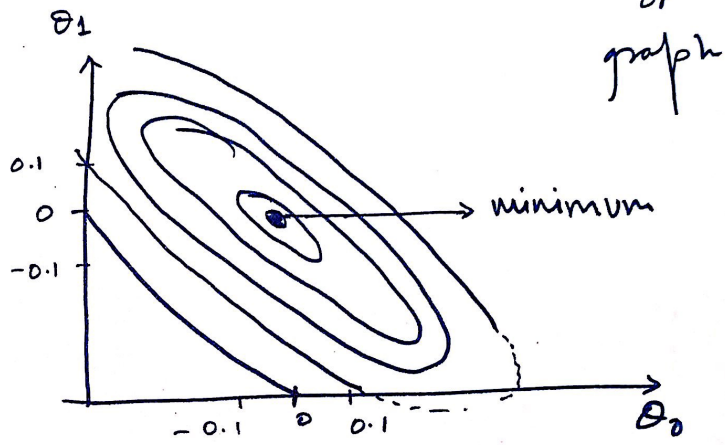
3-d graph:



contour plots or graph

$h_\theta(x)$





* We want to find/ build an algorithm that minimizes cost fm. over $(\theta_0, \theta_1, \ldots\ldots)$ parameters

Algorithm : Gradient Descent

Have some fn. $J(\theta_0, \theta_1)$ [Any fn.]

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

Outline :

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at minimum
- Goes to local minimum ***

Algorithm :

repeat until Convergence {

$$\theta_j := \theta_j - \alpha \left(\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \right) \quad \forall \; j=0, \; j=1$$

}

simultaneously update θ_0 and θ_1

learning rate → high : big step

learning rate → low : small steps

$$\text{temp } 0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

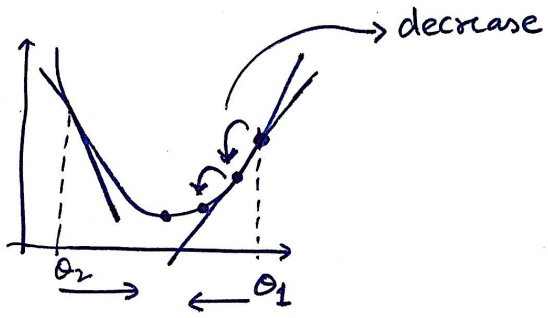
$$\text{temp } 1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 = \text{temp } 0$$

$$\theta_1 = \text{temp } 1$$

Correct simultaneous update.

Example: $\min_{\theta_1} J(\theta_1)$ $\quad \theta_1 \in \mathbb{R}$



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

slope is positive

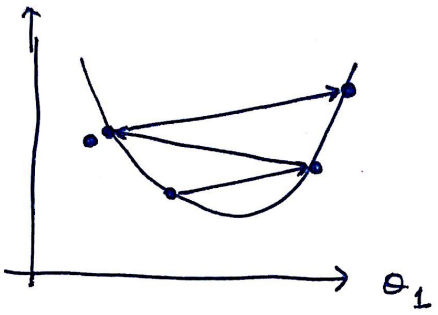
This indicate, subtract from θ_1. move left

$$\theta_2 := \theta_2 - \alpha \cdot \underbrace{\frac{d}{d\theta_2} J(\theta_2)}$$

This indicates, add to θ_2. move right

* If α is too small, gradient descent can be slow

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



• Gradient descent for linear Regression

Gradient descent Algo

repeat until convergence {

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

}

Linear Regression Model

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

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

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

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

$$\left\{ \begin{array}{l} \therefore \ \theta_0 \quad j=0 \ : \ \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right) \\ \\ \theta_1 \quad j=1 \ : \ \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right) . \ x^i \end{array} \right\} \text{derivatives}$$

The algorithm is sometimes also called

'Batch' Gradient Descent

↳ Each step of gradient descent uses all the training examples