Machine learning:

Arthur Samuel → " The field of study that gives computers the ability to learn without being explicitly programmed.

Tom Mitchell → "A computer program said to learn from experience E with respect to some class of Tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Supervised learning: "right answer"
correct output ← dataset → Actual

→ Regression: Predict continuous valued output
→ classification: Discrete valued output (0 or 1)
(0,1,2,3)

Unsupervised Learning: clustering

in dataset evry points are leveled same.

→ Social network analysis

→ Astronomical data analysis

→ Market segmentation

→ organized computing clusters

→ Gene grouping

→ We can derive structure by clustering the data based on relationships among the variables in the data.

Cocktail party problem:

Two microphones in a room with two speakers. The recording of that two microphone can be distinguish two different output by this algo.

```
[W,s,v] = svd((repmat(sum(x.*x,1),size(x,1),
               1).*x)*x');
```

Non-Clustering: The 'CPA' allows us to find structure in a chaotic environment (ie, identifying individual voices and music from a mesh of sounds)

Model Presentation:

$(x^{(i)}, y^{(i)}) \rightarrow$ ith dataset

Training Set

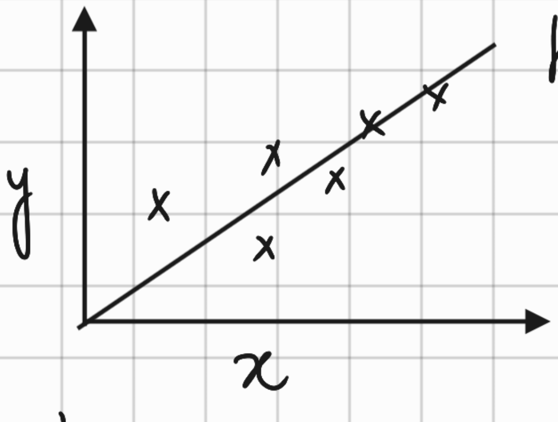
$\downarrow$

Learning Algorithm

$\downarrow$

Size of house (x) $\rightarrow$ h $\longrightarrow$ Estimated price

hypothesis (estimated value of y)



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

$\downarrow$

$h(x) = \theta_0 + \theta_1 x \rightarrow$ Linear function

Linear regression with one variable

$\rightarrow$ Univariate linear regression

Cost Function: $m = \#$ training example

$$\underset{\theta_0\ \theta_1}{\boxed{\text{minimize}}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

find sunch $\theta_0\ \theta_1$ so that

should be minimized.

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

$\hookrightarrow J(\theta) = \dfrac{sum((X \times \theta - y).\wedge 2)}{2m}$

Vect. //

Cost function

Squared error function

If $\theta_0 = 0$ $\quad J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)}\right)^2$

In our model $\rightarrow$

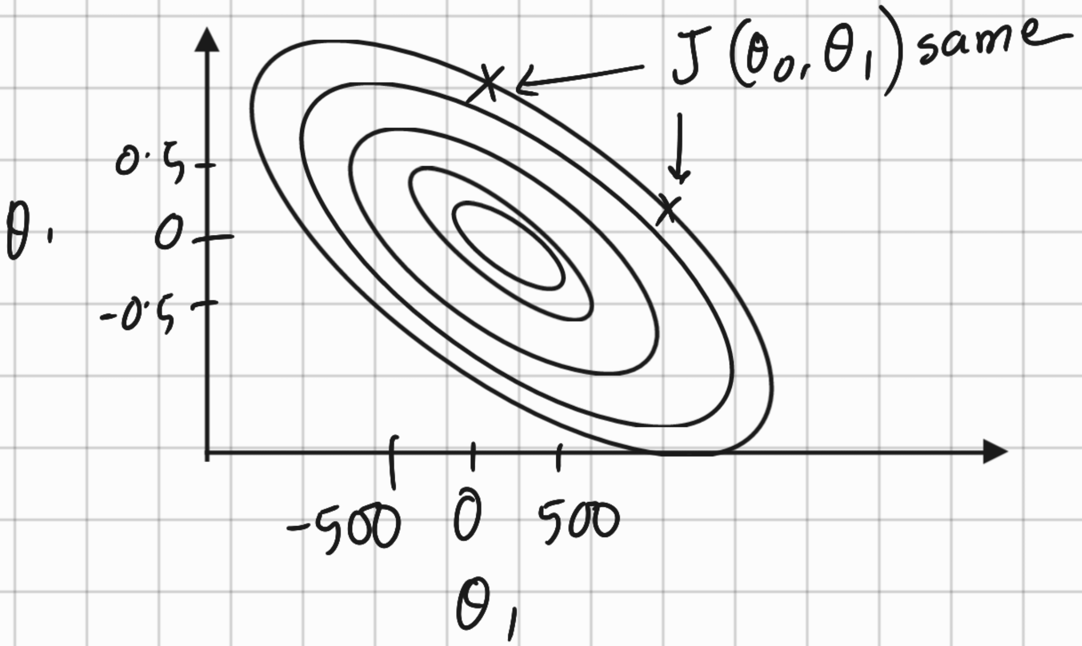
hypothesis : $h_\theta(x) = \theta_0 + \theta_1 x$

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 : $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Contour plot or figure instead of 3d graph



Gradient descent:

Have some function: $J(\theta_0 \dots \theta_n)$

Want: $\min\limits_{\theta_0 .. \theta_n} J(\theta_0 \dots \theta_n)$

Outline:

→ Start with some θ_0, θ_1

→ Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

→ Untill we hopefully endup with a minimum

$:=$ assignment

Algo:

repeat untill convergence {

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

}

α ↑ learning rate/step length

↳ Simultaneously update θ_0 & θ_1

$\alpha \uparrow$ aggresive step

$\alpha \downarrow$ baby step

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

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

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

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

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

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

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

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

repeat untill convergence {

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

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

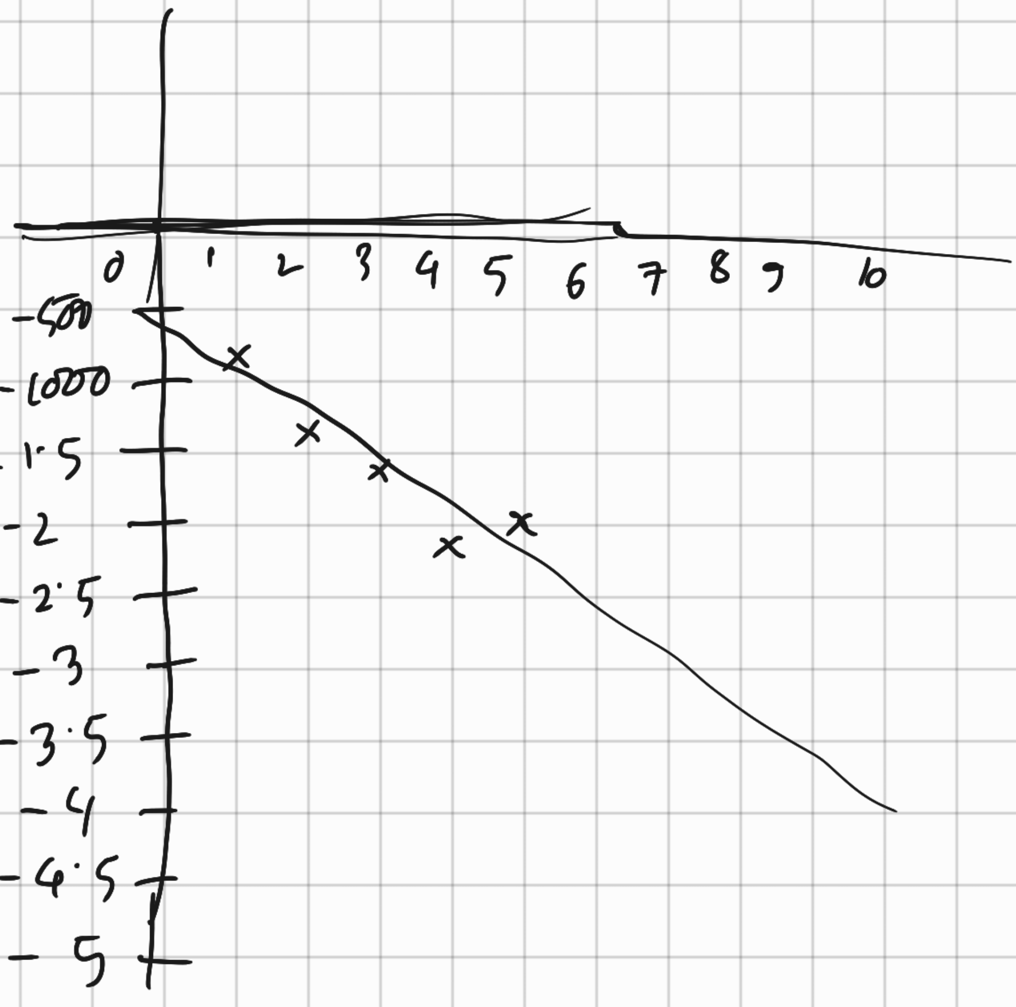
} update θ_0 & θ_1 simultaneously.

$J(\theta_0, \theta_1)$ is always a convex / Bowl-shape func.

↳ It has only one global / local optimum.

x	y	h
3	2	3
1	2	1
0	1	0
4	3	4

0, 0.5



Linear Algebra Revision

prediction = datamatrix $\times$ parameters

House Sizes:

2104
1416
1534
852

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

$$\text{data matrix} = \begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix}$$

$$\text{parameter} = \begin{bmatrix} -40 \\ 0.25 \end{bmatrix}$$

$$\text{prediction} = \begin{bmatrix} h(2104) \\ h(1416) \\ h(1534) \\ h(852) \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ -5 \\ 4 \end{bmatrix}^T \times \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 & -5 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$