Week 1

Introduction: What is machine learning?

- **Tom Mitchell:**

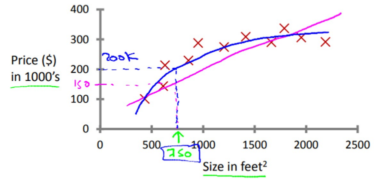
Learn from experience E ——>to do task T——>performance measure P

- **Learning algorithm:**

1. Supervised learning:teach computer how to do

    Give the algorithm a data set called "right answers" Relationship between the input and output.

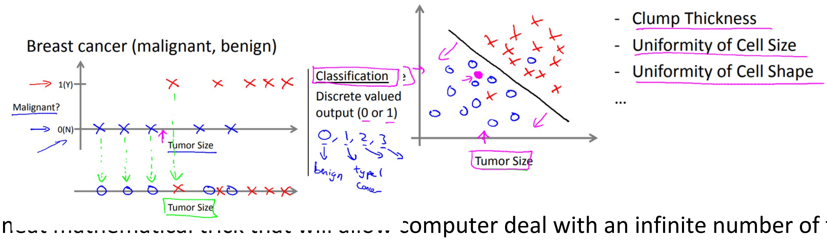
    - Regression :try to predict the sort of continuous values attribute.



Supervised Learning
"right answers" given

Regression: Predict continuous valued output (price)

    - Classification:try to predict a discrete value zero or one. Sometimes more than two possible

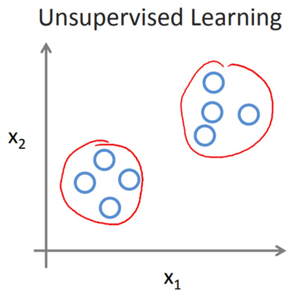


SVM:n... computer deal with an infinite number of features.

2. unsupervised learning:learn it by itself.

    Not told what to do with set and not told what each data point is. Find some structure in the dat set. Cluster. There is no right answer in advance.

    - Cluster:



    - Cocktail party algorithm: separate out two audio sources that were being added or being summed together. find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a cocktail party).

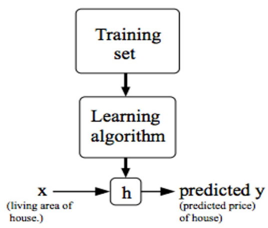
3. reinforcememt learning
4. recommender systems.

Linear Regression with One Variable：

- **Model representaiton：**

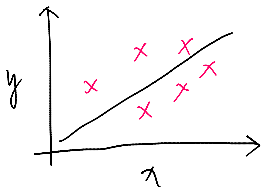
Training set of housing prices (Portland, OR)

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…



$(x^{(i)}, y^{(i)})$ called training example.

- Liner regression: one variable, x to predict y

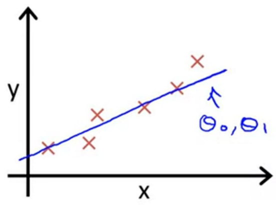


$h(x) = \theta_0 + \theta_1 x$.

parameters.

- Cost Function:

Measure accuracy of function by using Cost Function.



training example.

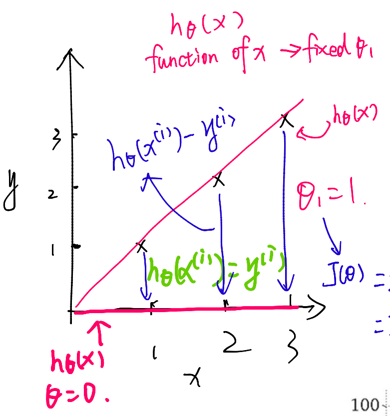
$\underset{\theta_0, \theta_1}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

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

$\underset{\theta_0, \theta_1}{\text{minimize}} \underbrace{J(\theta_0, \theta_1)}_{\text{Cost function}}$

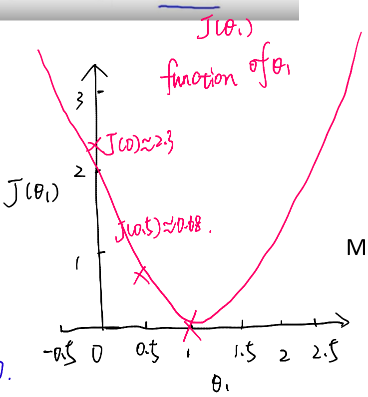
Squared error function

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



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

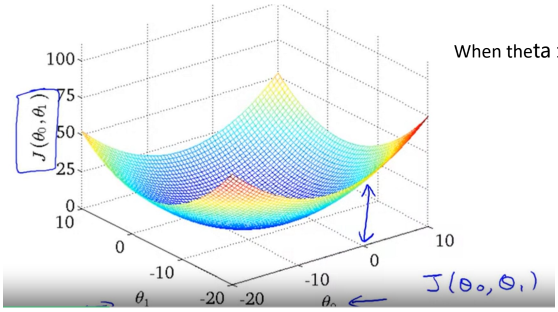
$= \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)}\right)^2 = 0$.



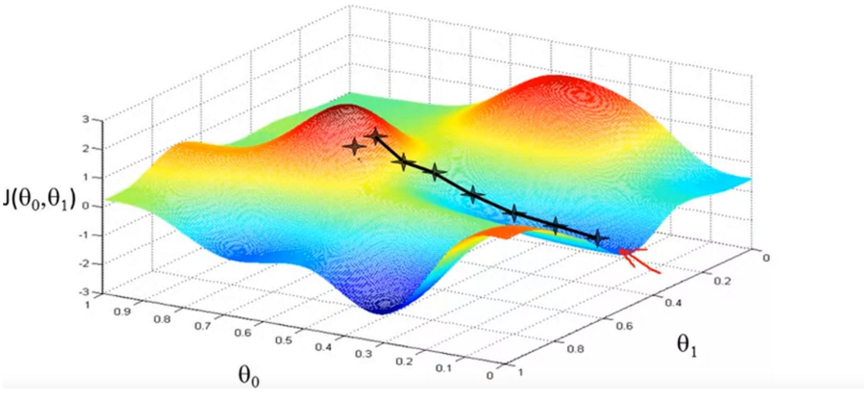
Each value of theta 1 correspond to a different hypothesis, then derive a different value of j

Minmize J

When theta 1 and theta 0 don't equal 0.



- Gradient descent:
  - 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 Sometimes is a local minimum.



Assignment | truth assertion

$a := b$ | $a=b$ just a claim

$a := a+1$

Gradient descent algorithm

repeat until convergence {

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

}

assignment.

learning rate. big correspond aggressive gradient descent like taking huge steps downhill

small ... little baby steps downhill.

Simultaneously update θ_1 and θ_j.

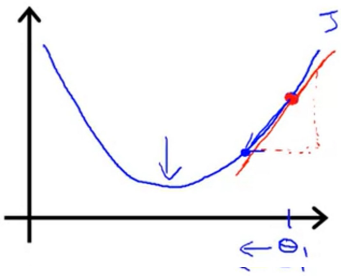
Correct: Simultaneous update

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

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

$\theta_0 := \text{temp0}$

$\theta_1 := \text{temp1}$

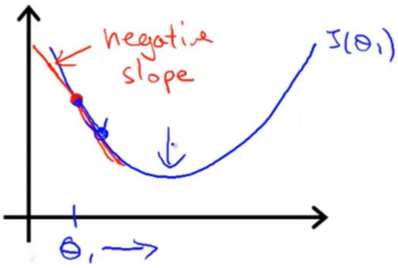


$(\theta_1 \in \mathbb{R})$

$$\theta_1 := \theta_1 - \alpha \underbrace{\boxed{\frac{d}{d\theta_1} J(\theta_1)}}_{\geq 0}$$

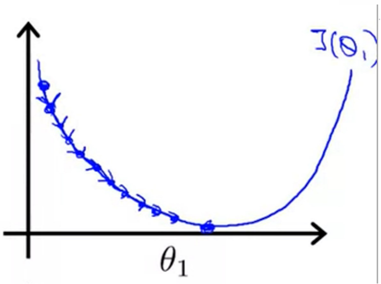
$\frac{\partial}{\partial \theta_1}$

$\theta_1 := \theta_1 - \alpha \cdot (\text{positive number})$



$$\underbrace{\frac{d}{d\theta_1} J(\theta_1)}_{\leq 0}$$

$\theta_1 := \theta_1 - \alpha\,(\text{negative number})$



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

α learning rate

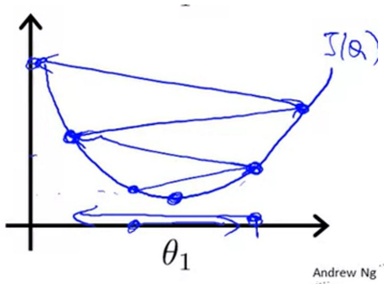
① too small → gradient descent too slow

② too big → overshoot minimum coverge / diverge.

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

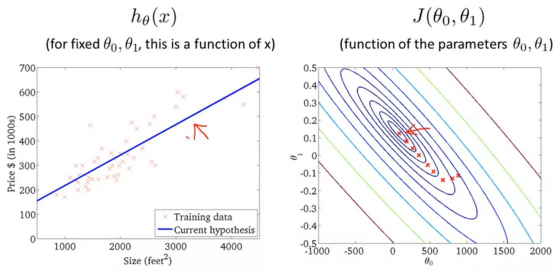
$\theta_0 \quad j=0 \quad : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}).$

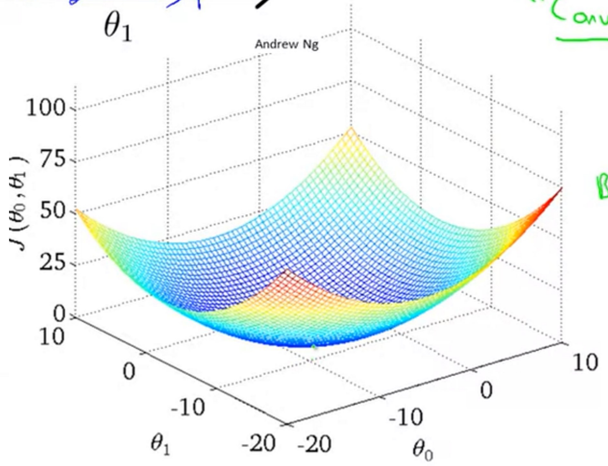
$\theta_1 \quad j=1 \quad : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left[(h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \right].$



For linear regression, the cost function is a convex function. So there is only one optimum result, no local optimum.

"Convex function"





Bowl-shaped

"Batch" Gradient Descent

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

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

- Linear Algebra Review:

	matrix: rectangular array of numbers , dimension of matrix: number of rows by number of columns

Vector: An n x 1 matrix.

$n = 4$

$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$ ← 4-dimensional vector. $\mathbb{R}^4$

$y_i = i^{th}$ element

$y_1 = 460$

$y_2 = 232$

$y_3 = 315$

1-indexed vs 0-indexed:

y[1] $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$ $y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$ y[0]

1-indexed   0-indexed

	Only two matrices that are of the same dimensions can add.

Matrix Addition

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

3 x 2 matrix   3 x 2   3 x 2

	scalar multiplication:take your elements of the matrix and multiply the real number.

Scalar Multiplication

real number

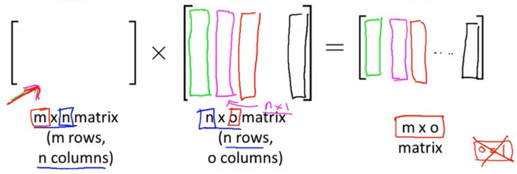
$$3 \times \underbrace{\begin{bmatrix}1 & 0\\ 2 & 5\\ 3 & 1\end{bmatrix}}_{3\times 2} = \underbrace{\begin{bmatrix}3 & 0\\ 6 & 15\\ 9 & 3\end{bmatrix}}_{3\times 2} = \begin{bmatrix}1 & 0\\ 2 & 5\\ 3 & 1\end{bmatrix} \times 3$$

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

Matrix-matrix multiplication:

Details:

$A \times B = C$



The i^{th} column of the matrix C is obtained by multiplying A with the i^{th} column of B. (for i = 1,2,...,o)

Properties: $A \times B \neq B \times A$ not commutative

$(A \times B) \times C = A \times (B \times C)$ associative

Identity matrix: it has ones along the diagonals.(denoted I or $I_{n\times n}$)

For any matrix A,

$$\underset{m\times n}{A} \cdot \underset{n\times n}{\boxed{I}} = \underset{m\times m}{\boxed{I}} \cdot \underset{m\times n}{A} = \underset{m\times n}{A}$$

Inverse and transpose:

only square matrix has inverse matrix:

Matrix inverse: square matrix (#rows = #columns) A^{-1}

If A is an m x m matrix, and if it has an inverse,

$$\rightarrow \quad A(A^{-1}) = A^{-1}A = I.$$

Matrices that don't have an inverse are "singular"or "degenerate"