경희대학교
KYUNG HEE UNIVERSITY

[SWCON253] Machine Learning – Lec.03

Toy Example

(Housing Price Prediction)

Fall 2025

김 휘 용

Visual Media Lab
http://vmlab.khu.ac.kr

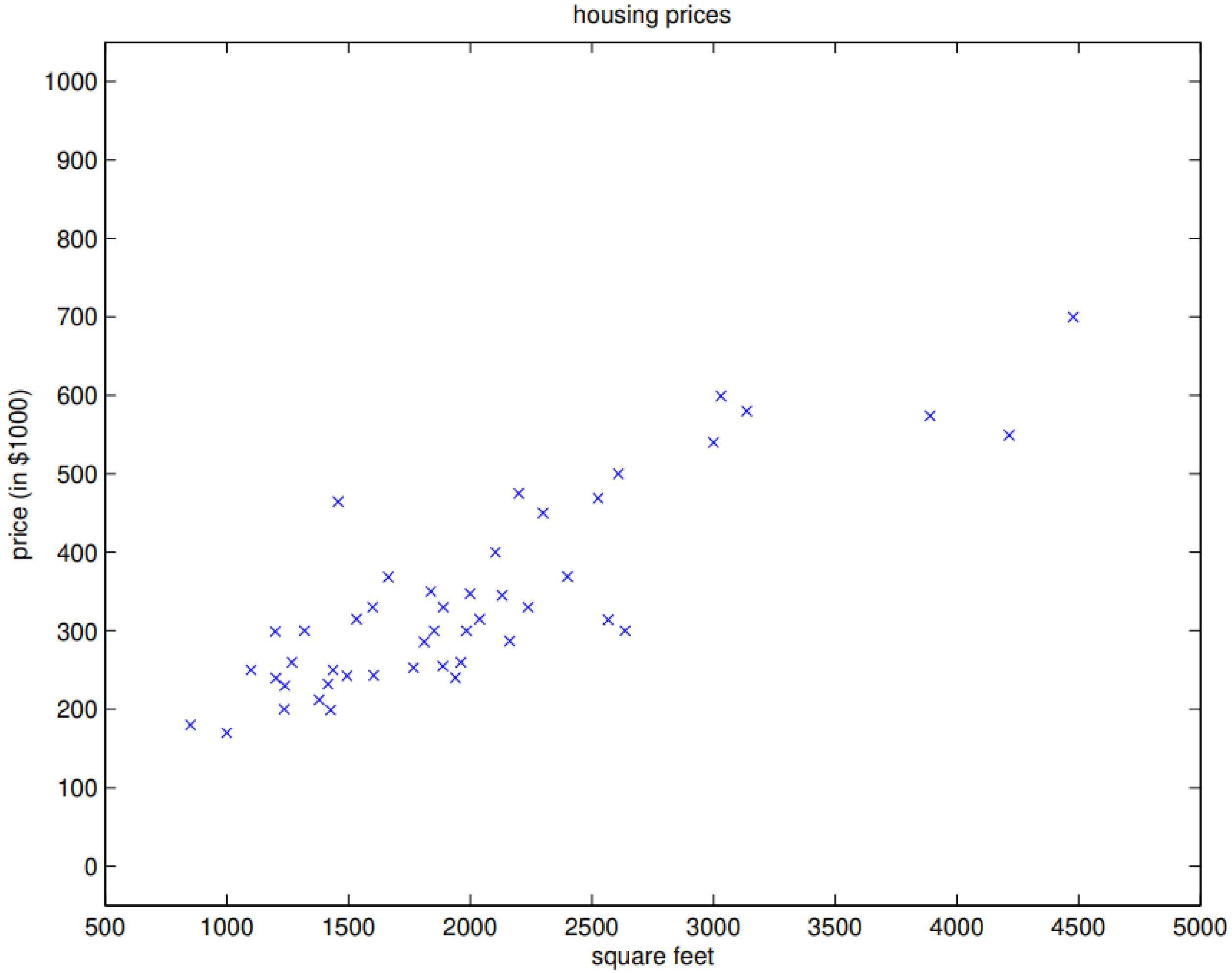
Housing Price Prediction

1. Problem Definition
2. Data Representation
3. Model Representation
4. Cost Function
5. Optimization
6. *Extension to Multiple Features*
7. Summary of Terminologies

1. Problem Definition

- Suppose we have a dataset giving the living areas and prices of 47 houses:

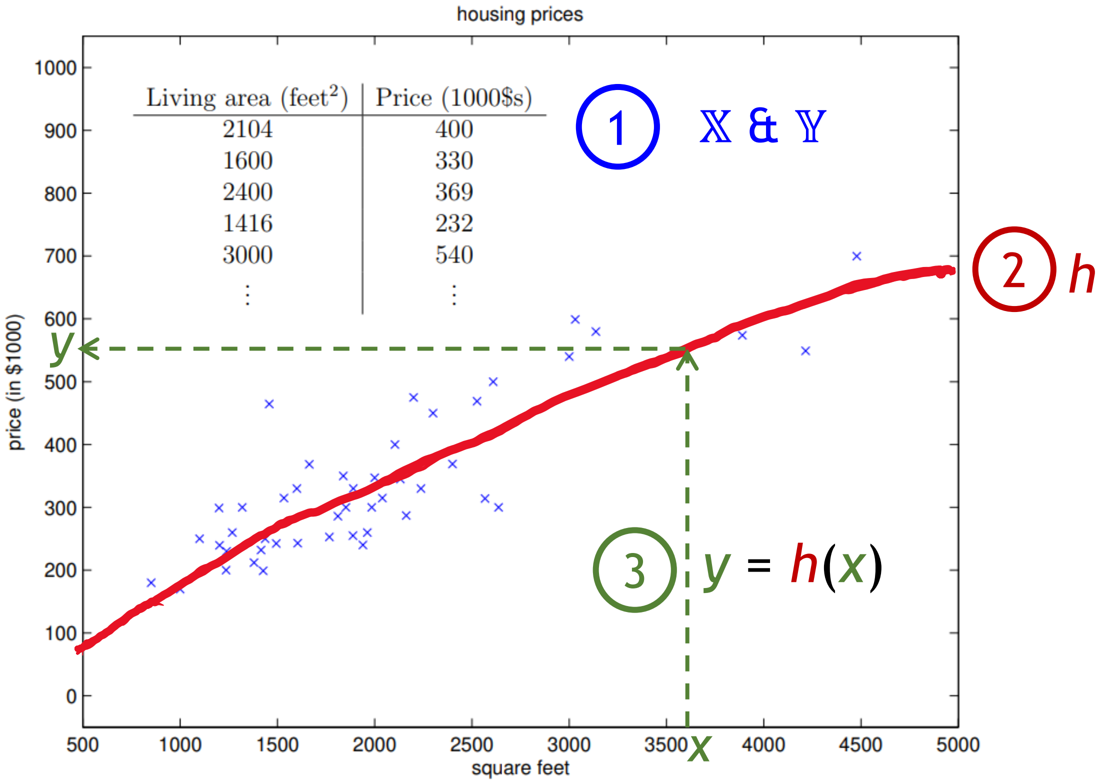
Living area ($feet^2$)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

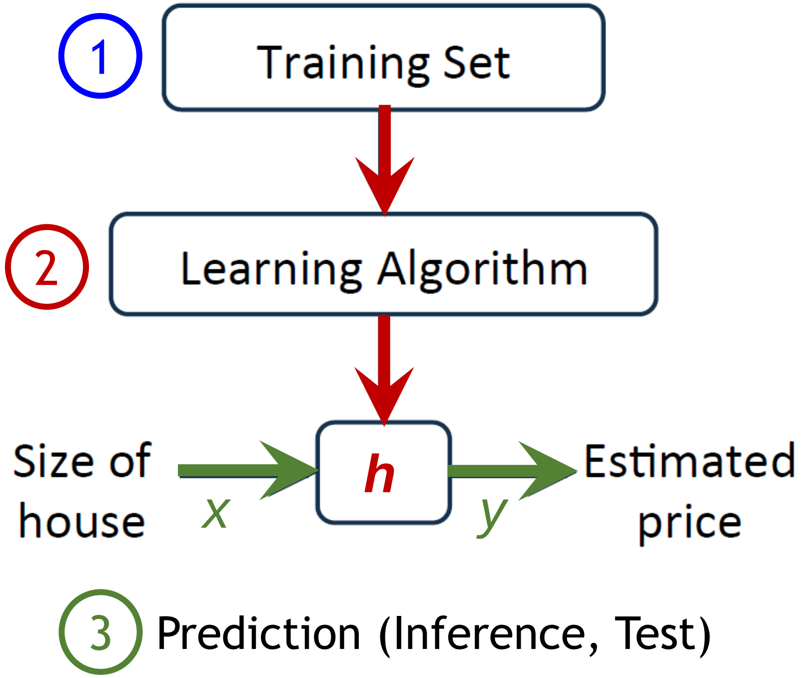


- Given data like this,
  how can we ***learn to predict***
  the prices of other houses?

1. Problem Definition (cont'd)

- **Given data like this, how can we *learn to predict* the prices of other houses?**
  - The dataset $\mathbb{X}$ & $\mathbb{Y}$ is called a '***training set***'
  - The prediction function *h* is called a '***hypothesis***' or '***model***' : $y = h(x)$
  - Note: this is a ***supervised learning*** task. More specifically, it is a ***regression*** problem.



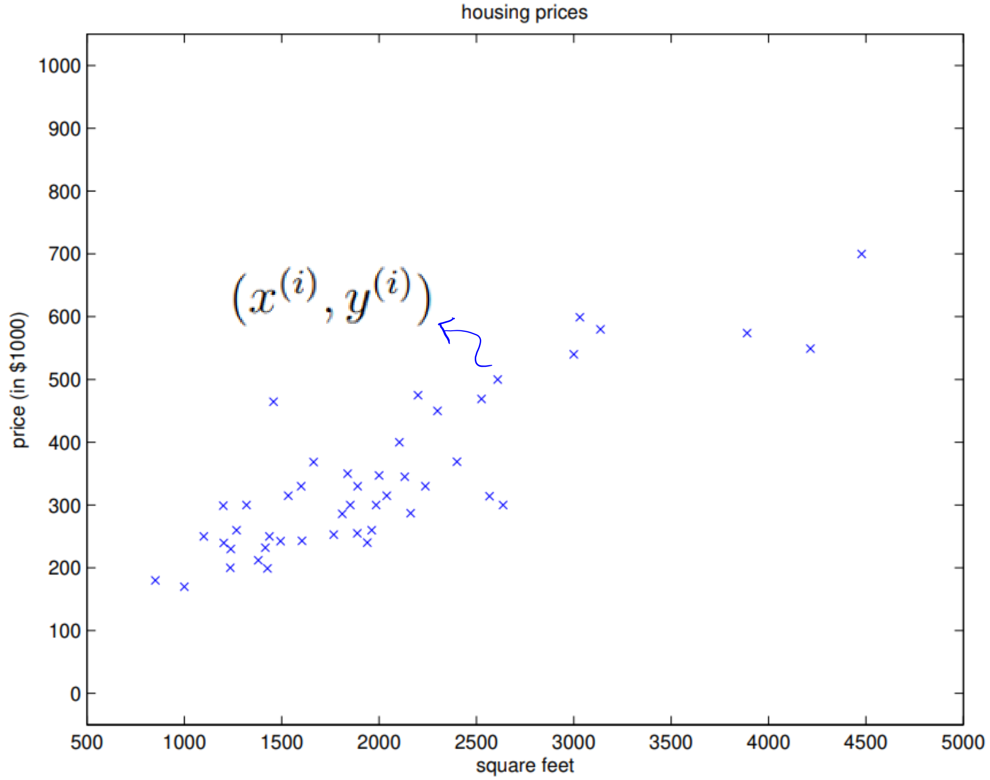


2. Data Representation

Training Set

index	$\mathbb{X}$ Living area (feet^2)	$\mathbb{Y}$ Price (1000\$s)
1	2104	400
2	1600	330
⋮	2400	369
i	$x^{(i)}$ 1416	232 $y^{(i)}$
⋮	3000	540
n	⋮	⋮

- $x^{(i)}$: "**input**" variables, also called input ***features***
- $y^{(i)}$: "**output**" or ***target*** variable
- $(x^{(i)}, y^{(i)})$: a ***training example***
- $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, n\}$: a ***training set***



Design Matrix : $X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix}$

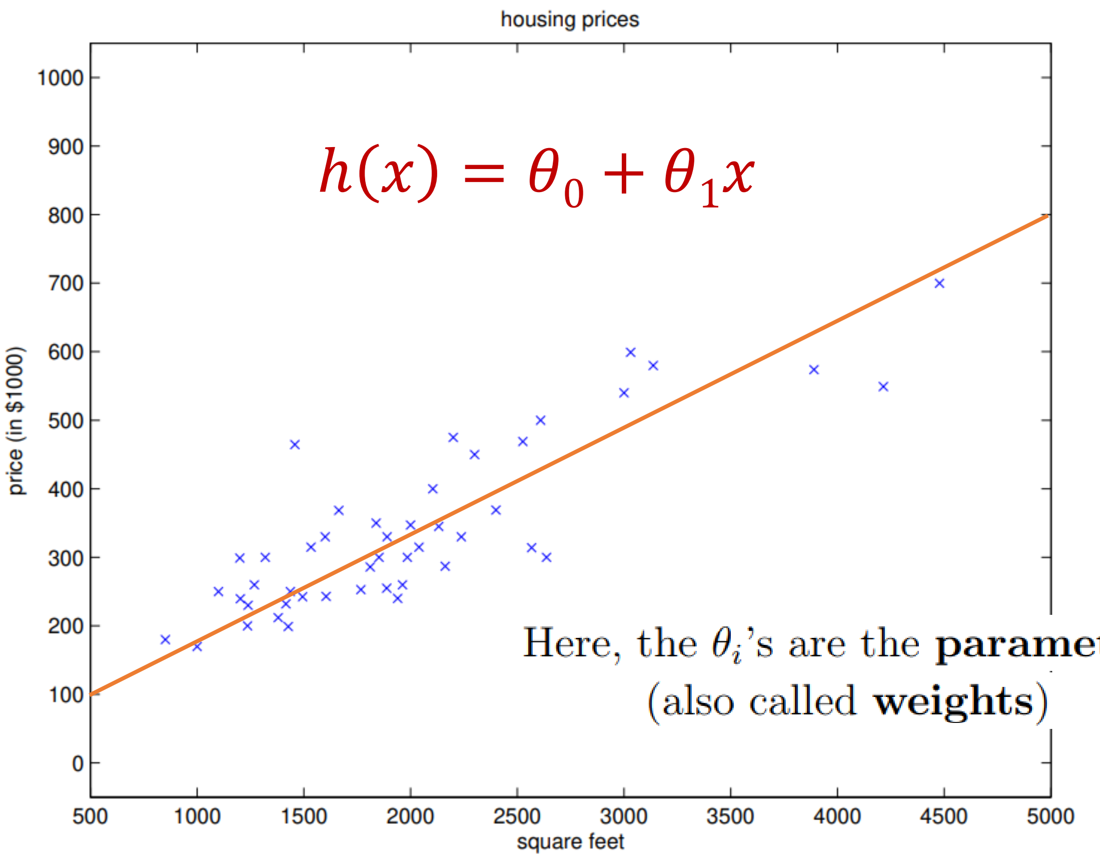
&

Target Vector $\underline{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

3. Model Representation

Linear Model (Linear Hypothesis)

- We call this problem '*linear regression*'



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

Here, the θ_i's are the **parameters** (also called **weights**)

- Vector Notation for Linear Model

Let $\underline{x} = \begin{bmatrix}1\\ x\end{bmatrix}$, $\underline{\theta} = \begin{bmatrix}\theta_0\\ \theta_1\end{bmatrix}$.

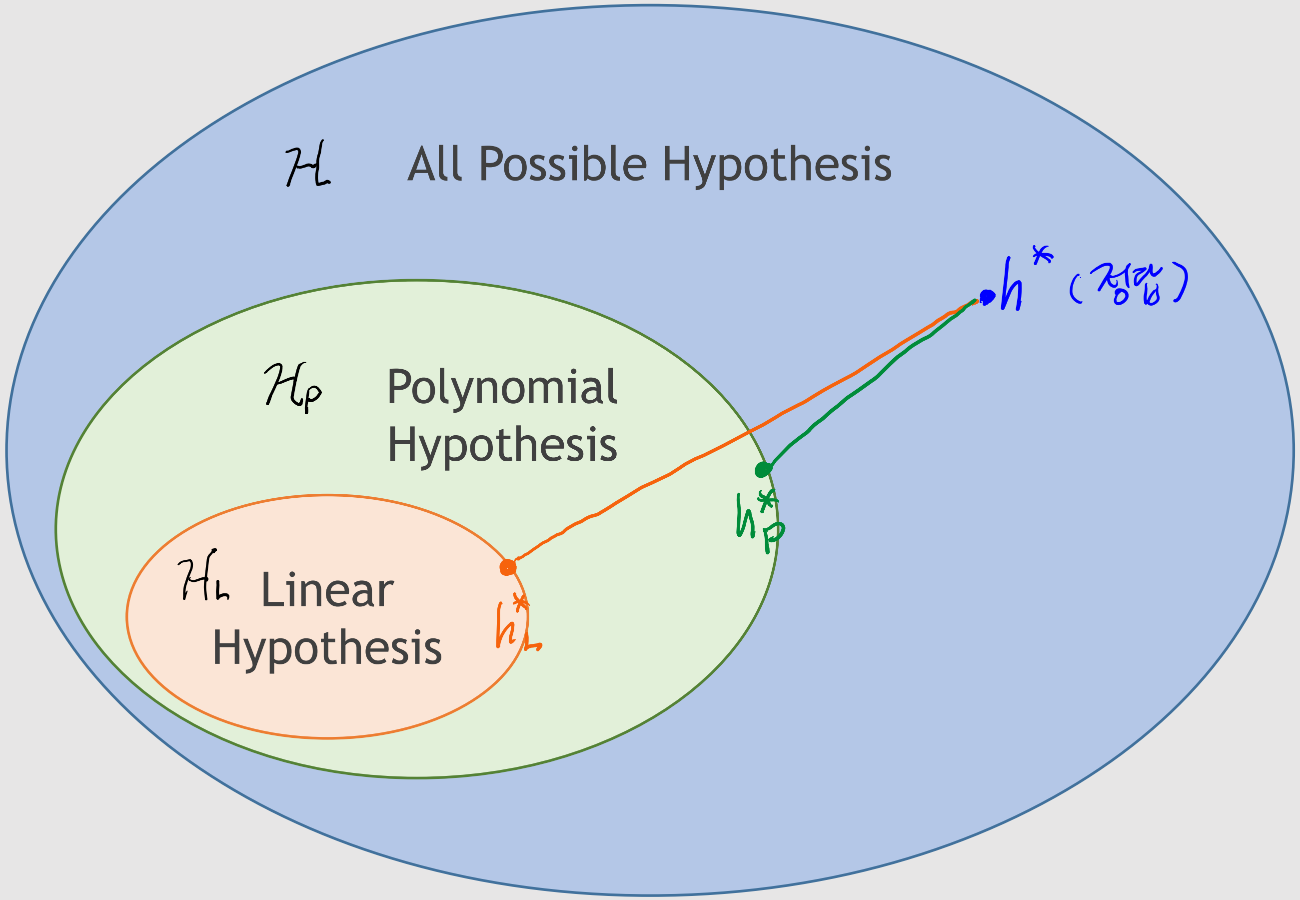
Then $y = h_{\underline{\theta}}(\underline{x})$

$= \underline{\theta}^T \underline{x} = \underline{x}^T \underline{\theta}$

3. Model Representation (cont'd)

Hypothesis Space & Model Capacity

- ML algorithm: find the best hypothesis h^* within its hypothesis space $\mathcal{H}$.
- Larger hypothesis space implies larger model capacity: (+) easier to fit (-) harder to train.



Examples

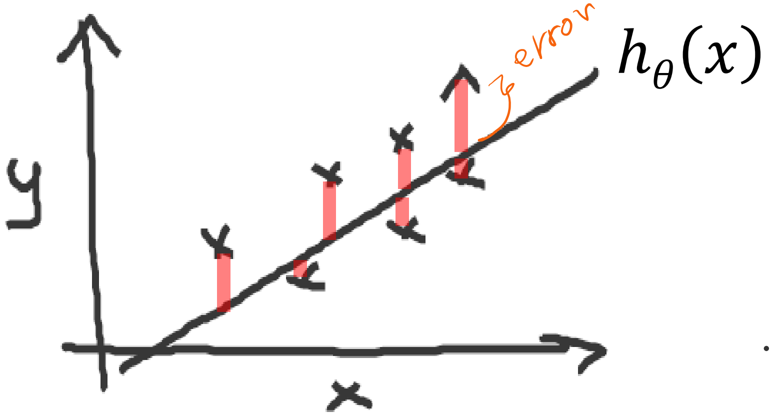
- $h \in \mathcal{H}_L$: $h(\underline{x}) = \theta_0 + \theta_1 x$
- $h \in \mathcal{H}_P$: $h(\underline{x}) = \theta_0 + \theta_1 x + \theta_2 x^2$
- $h \in \mathcal{H}$: $h(x) = \theta_0 + \cos(x) + e^{-x^2}$

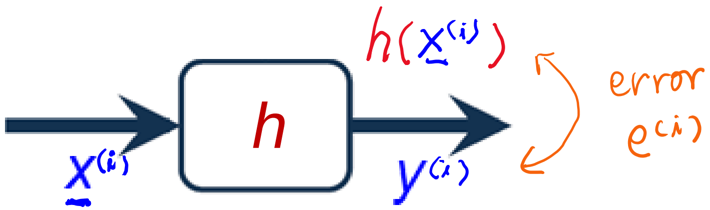
4. Cost Function (비용함수)

◆ How to choose the model parameters?

- MSE (Mean Squared Error)
- SSE (Sum of Squared Error)

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





Ex) Minimize MSE Cost (Loss):

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\underbrace{(h_\theta(x^{(i)}) - y^{(i)})^2}_{\triangleq\, e^{(i)}}$$

Vector form of the MSE:

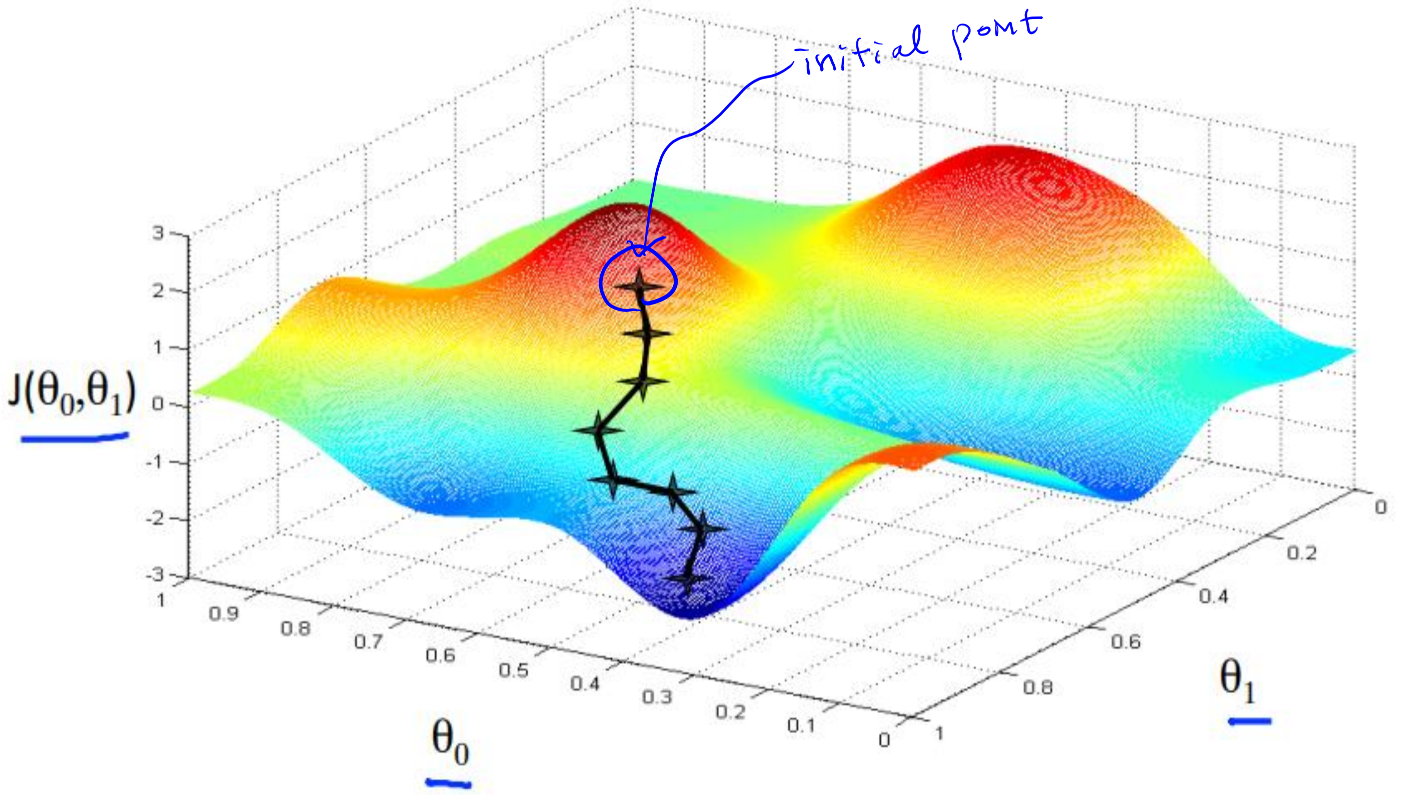
Let $\underline{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$ $\quad \hat{\underline{y}} = \begin{bmatrix} h(x^{(1)}) \\ \vdots \\ h(x^{(n)}) \end{bmatrix}$ $\quad \underline{e} = \hat{\underline{y}} - \underline{y}.$

Then $J = \frac{1}{2}\|\underline{e}\|_2^2 = \frac{1}{2}\underline{e}^T\underline{e}$

$$= \frac{1}{2}\|\hat{\underline{y}} - \underline{y}\|_2^2 = \frac{1}{2}(\hat{\underline{y}} - \underline{y})^T(\hat{\underline{y}} - \underline{y})$$

5. Optimization (최적화)

- **Iterative Methods (반복법)**
  - Gradient Descent (GD)
  - Stochastic Gradient Descent (SGD)
  - Newton's Method



- **Analytic Methods (해석적 방법)**
  - Normal Equation (Linear Model의 경우)

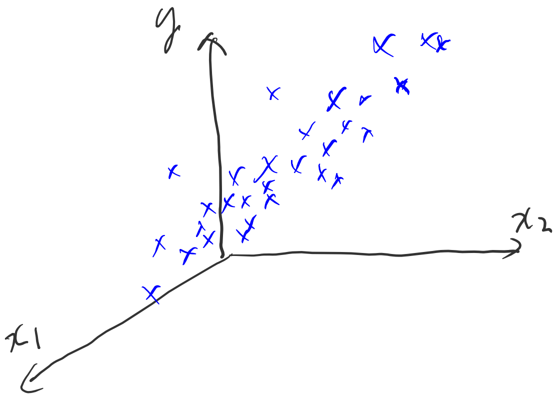
- ***We will study them later...***

6. Extensions to Multiple Features

Training Set

$\underset{\sim}{x} = [\, x_1 \quad x_2 \,]^T \quad y$

	Living area (feet^2)	#bedrooms	Price (1000\$s)
$i=1$:	2104 $x_1^{(1)}$	3 $x_2^{(1)}$	400 $y^{(1)}$
$i=2$:	1600 $x_1^{(2)}$	3 $x_2^{(2)}$	330 $y^{(2)}$
⋮	2400 ⋮	3 ⋮	369 ⋮
	1416	2	232
	3000	4	540
$i=n$:	⋮ $x_1^{(n)}$	⋮ $x_2^{(n)}$	⋮ $y^{(n)}$



Data Representation

- Feature vector: $\underset{\sim}{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

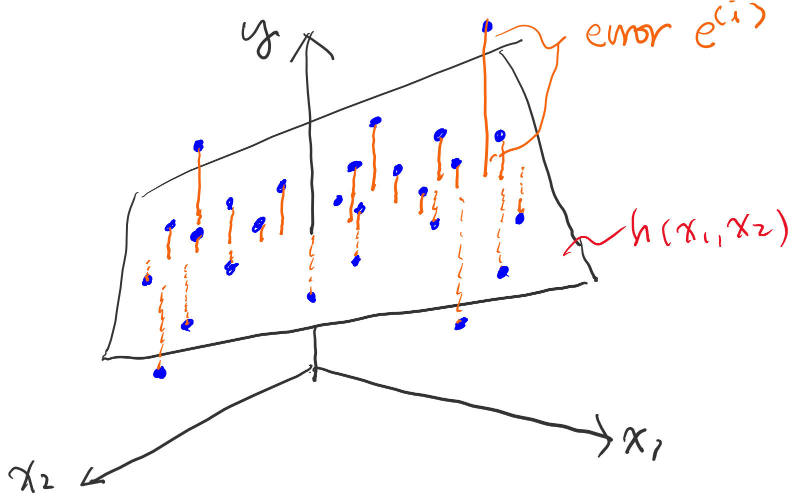
- Design matrix:

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots \\ x_1^{(n)} & x_2^{(n)} \end{bmatrix} = \begin{bmatrix} - & \underset{\sim}{x}^{(1)T} & - \\ - & \underset{\sim}{x}^{(2)T} & - \\ & \vdots & \\ - & \underset{\sim}{x}^{(n)T} & - \end{bmatrix}$$

- Target vector: $\underset{\sim}{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$

6. Extensions to Multiple Features (cont'd)

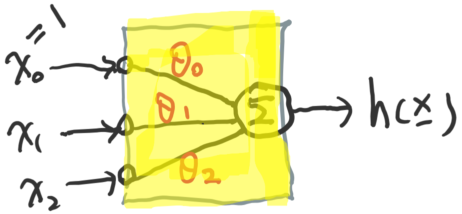
Model Representation (Linear Model)



$$h(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$\underline{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \qquad \underline{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

$$\Rightarrow \quad h(\underline{x}) = \underline{\theta}^T \underline{x} = \underline{x}^T \underline{\theta}$$



Cost Function (MSE Cost)

$$J(\underline{\theta}) = \frac{1}{2} \sum_{i=1}^{n} \left(e^{(i)}\right)^2$$

where $$e^{(i)} = \underbrace{h(\underline{x}^{(i)})}_{\triangleq \hat{y}^{(i)}} - y^{(i)}$$

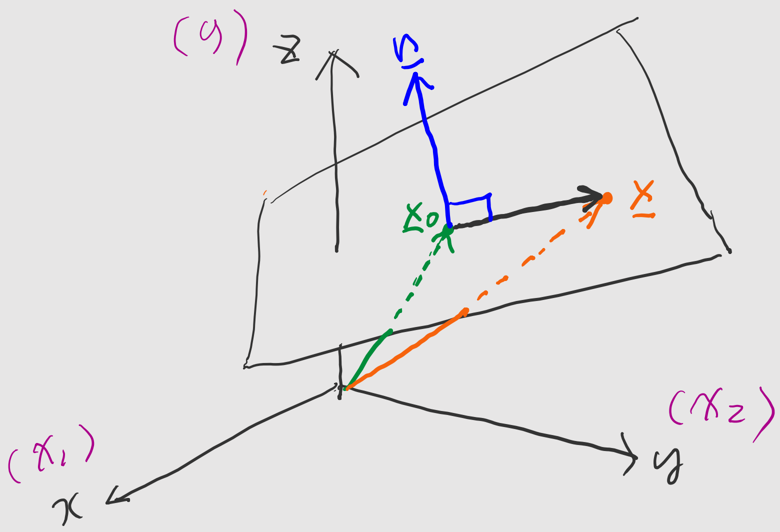
Let $$\underline{e} = \begin{bmatrix} e^{(1)} \\ \vdots \\ e^{(n)} \end{bmatrix}$$

$$J(\underline{\theta}) = \frac{1}{2} \|\underline{e}\|^2 = \frac{1}{2} \underline{e}^T \underline{e}$$

$$\left(= \frac{1}{2} (\hat{\underline{y}} - \underline{y})^T (\hat{\underline{y}} - \underline{y}) \right)$$

[Remind] 3차원 공간에서 평면의 방정식

◆ 점 $\underline{x}_0$를 지나고 벡터 $\underline{n}$이 수직인 평면의 방정식은 ?



$\underline{x}_0 = (x_0\ y_0\ z_0)^T$

$\underline{n} = (a, b, c)^T$

$\underline{x} = (x, y, z)^T$

$$(\underline{x} - \underline{x}_0) \underset{\text{(수직)}}{\perp} \underline{n}$$

$$\Rightarrow \underline{n}^T(\underline{x} - \underline{x}_0) = 0$$

$$\Rightarrow a(x - x_0) + b(y - y_0) + c(z - z_0) = 0$$

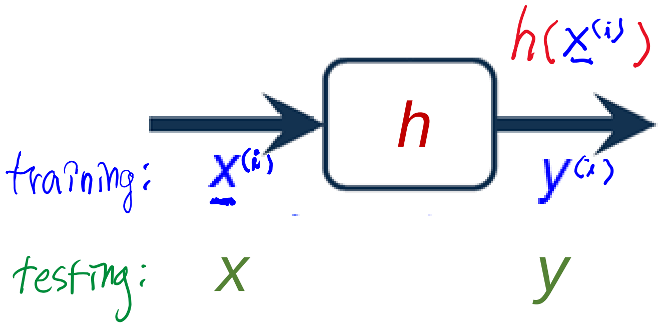
or $ax + by + cz = d$

$$\Rightarrow z = \frac{d}{c} - \frac{a}{c}x - \frac{b}{c}y$$

$$\Rightarrow y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

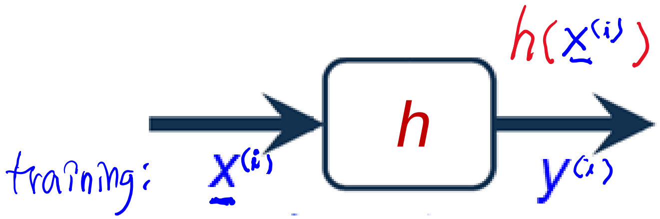
※ Hyper-plane : n차원 공간상에서 $(n-1)$차원 부분공간을 말함
(초평면)

7. Summary of Terminologies



- **Feature / Input (*x*)**
  - input to the black box, a column in the table representing the dataset
  - synonymous to variable, attribute, covariate
- **Output / Prediction (*y*)**
  - means output from the model, synonymous to response variable, dependent variable
- **Targets / Labels (true *y*)**
  - what we want to predict
  - synonymous to correct output, ground truth, label (in classification)
- **Training example (*x*, true *y*)**
  - a pair of input & target, a row in the table representing the dataset
  - synonymous to an observation, training record, training instance, training sample
- **Training set**
  - a collection of training examples

7. Summary of Terminologies (cont'd)



Hypothesis / Model (*h*)

- a certain function that we believe is similar to the true function,
  the target function that we want to model
- In the ML, the terms hypothesis and model are often used interchangeably

Classifier 분류기

- a hypothesis or discrete-valued function that is used to assign class labels to particular data points

(Machine) Learning algorithm

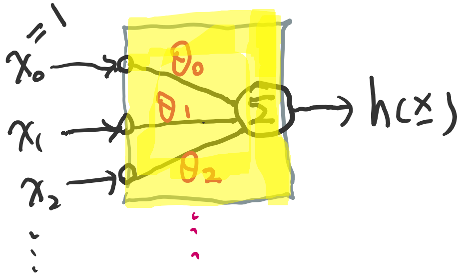
- a set of instructions that tries to model the target function using our training dataset
- a learning algorithm comes with a hypothesis space, the set of possible hypotheses it explores to model the unknown target function by formulating the final hypothesis

Loss function / Cost function (Objective function) 손실함수, 비용함수, 목적함수, 오차함수

- In some contexts the loss for a single data point,
  whereas the cost function refers to the overall loss over the entire dataset
- Sometimes also called error function or empirical risk.

Q & A

* Linear Model



$$h(x_1, x_2, \cdots, x_d) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d$$

$$h(\underset{\sim}{x}) = [\theta_0 \ \theta_1 \ \cdots \ \theta_d] \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \underset{\sim}{\theta}^T \underset{\sim}{x} = \underset{\sim}{x}^T \underset{\sim}{\theta}$$