

Spring 2023

SWCON253: Machine Learning

Lecture 01

Introduction to Machine Learning

Jinwoo Choi
Assistant Professor
CSE, Kyung Hee University



ToC

1. ML Definitions
2. ML Types
3. AI & ML
4. Datasets
5. Toy Example

1. ML Definitions

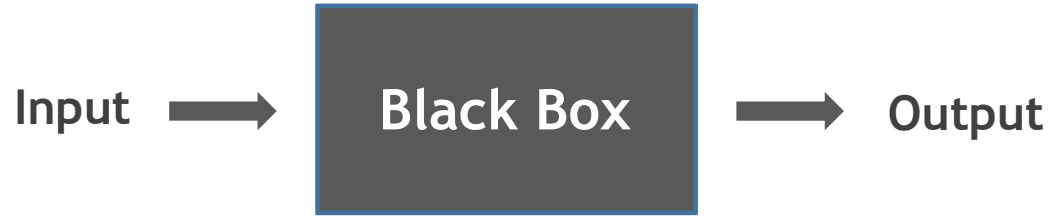
"A breakthrough in machine learning would be worth ten Microsofts"

-- **Bill Gates, Microsoft Co-founder**





Image source: <https://www.gatesnotes.com/Books>

Learn from Data: *Black Box*



◆ Black Box를 어떻게 만들 것인가?

- 
- 
- 

기계학습 정의 – by Samuel

■ 학습이란? <표준국어대사전>

“경험의 결과로 나타나는, 비교적 지속적인 행동의 변화나 그 잠재력의 변화. 또는 지식을 습득하는 과정[국립국어원2017]”

■ 기계 학습이란?

- 인공지능 초창기 사무엘의 정의



“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed”

— Arthur L. Samuel, AI pioneer, 1959

(This is likely not an original quote but a paraphrased version of Samuel's sentence "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.")

Arthur L. Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210-229.

Image Source: <https://history-computer.com/ModernComputer/thinkers/images/Arthur-Samuel1.jpg>

기계학습 정의 – by Mitchell

- 현대적 정의

“A computer program is 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 .”

— Tom Mitchell, Professor at Carnegie Mellon University

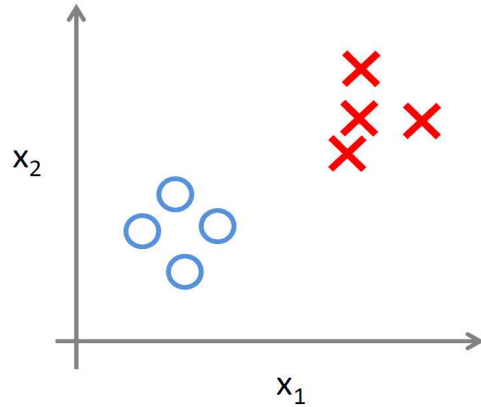
Handwriting Recognition Example:



- Task T : ?
- Performance measure P : ?
- Training experience E : ?

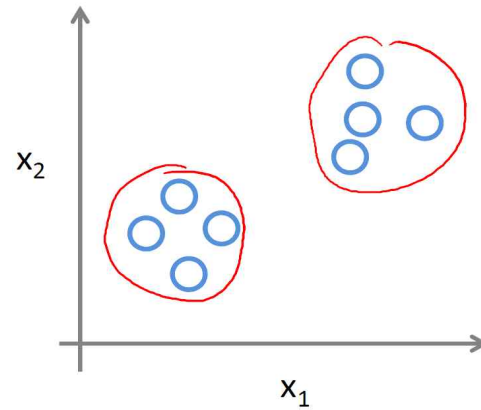
2. ML Types

Supervised Learning



classification

Unsupervised Learning

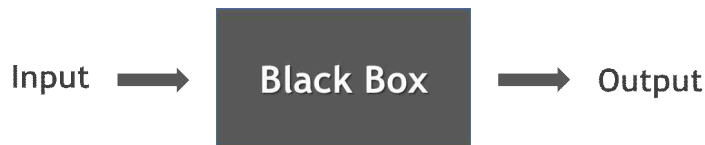


clustering

기계학습의 유형

◆ 지도 학습 (Supervised Learning)

- 특징 벡터(input) \mathbf{x} 와 목표값(output) y 가 모두 주어진 상황
- 회귀 문제 (Regression)
- 분류 문제 (Classification)



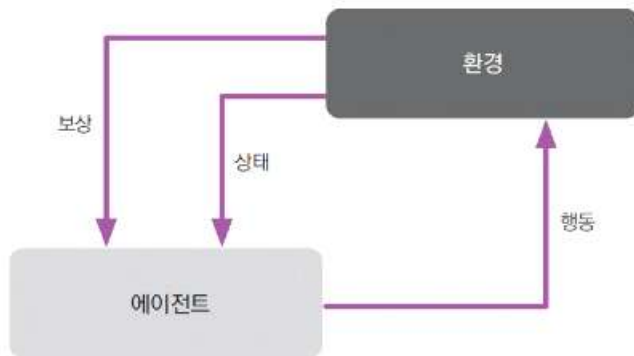
◆ 비지도 학습 (Unsupervised Learning)

- 특징 벡터 \mathbf{x} 는 주어지는데 목표값 y 가 주어지지 않는 상황
- 군집화(Clustering)
- 밀도 추정(Density Estimation)
- 차원 축소(Dimensionality Reduction)

◆ 준지도 학습 (Semi-Supervised Learning)

◆ 강화 학습 (Reinforcement Learning)

- 행동(action)을 수행하고 얻은 전체 보상(reward)을 최대화하는 일련의 행동을 학습



Supervised Learning (지도학습)

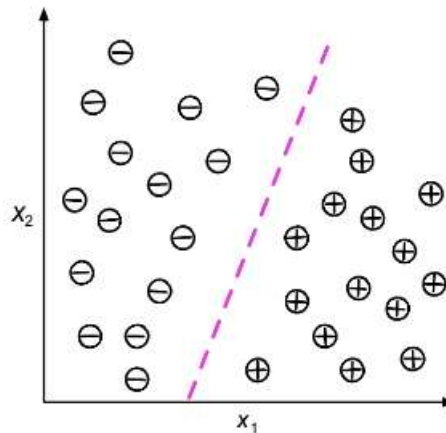
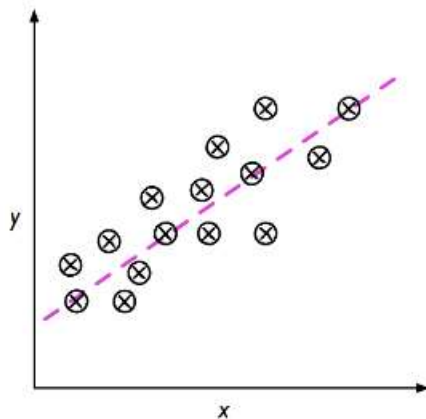
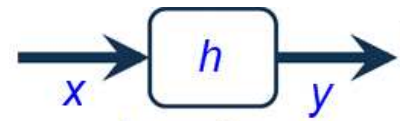
◆ Predict output for given input

- **Labeled data**: we already know what our **correct output** should look like.

Thus, we can give direct feedback based on the correct prediction results.

Categorized into:

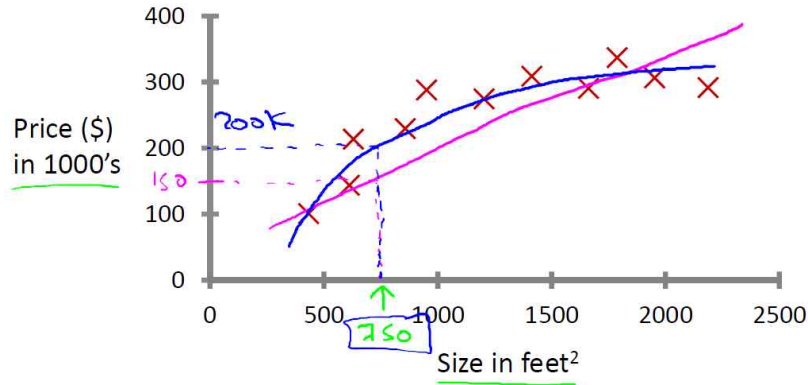
- ★ **Regression** problem: predict results within a **continuous output** ($y = f(x)$).
- ★ **Classification** problem: predict results in a **discrete output** (categories).



Supervised Learning – Examples

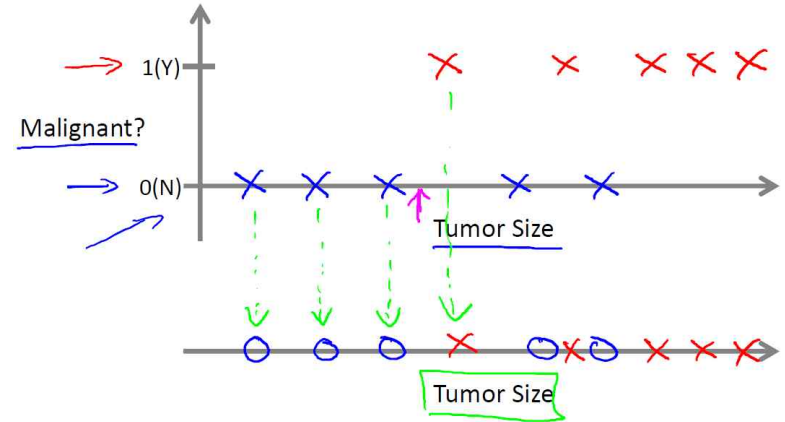
Ex.1) Housing price prediction

- regression: price



Ex.2) Breast cancer

- classification: malignant or benign



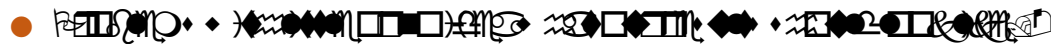
Ex.3) Spam mail filtering

- classification: spam or not

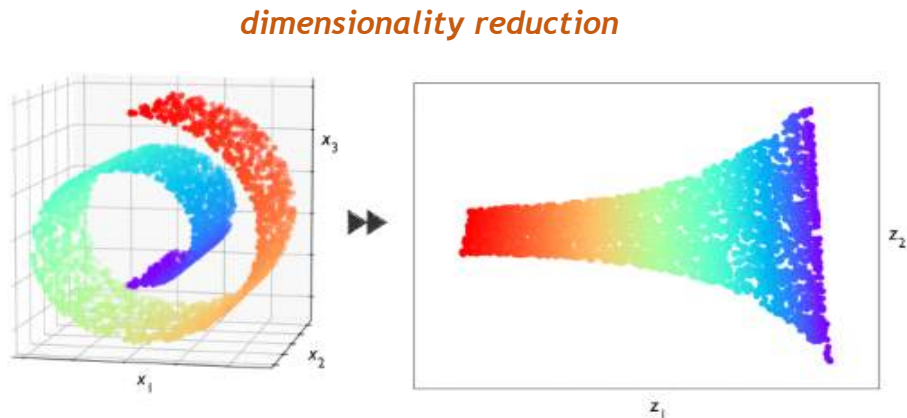
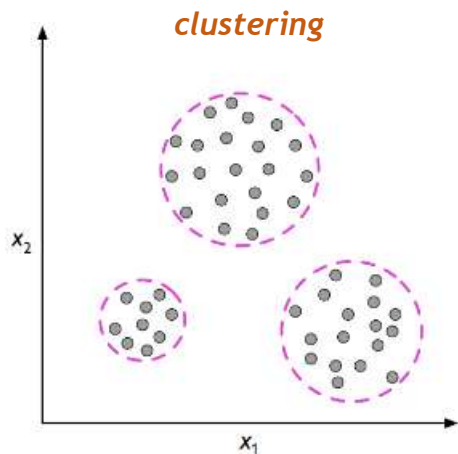
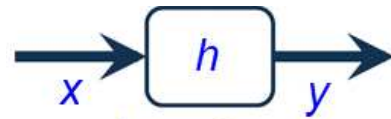


Unsupervised Learning (비지도학습)

◆ Find hidden structure in data



- We can derive structure from data by **clustering** the data based on relationships among the variables in the data.

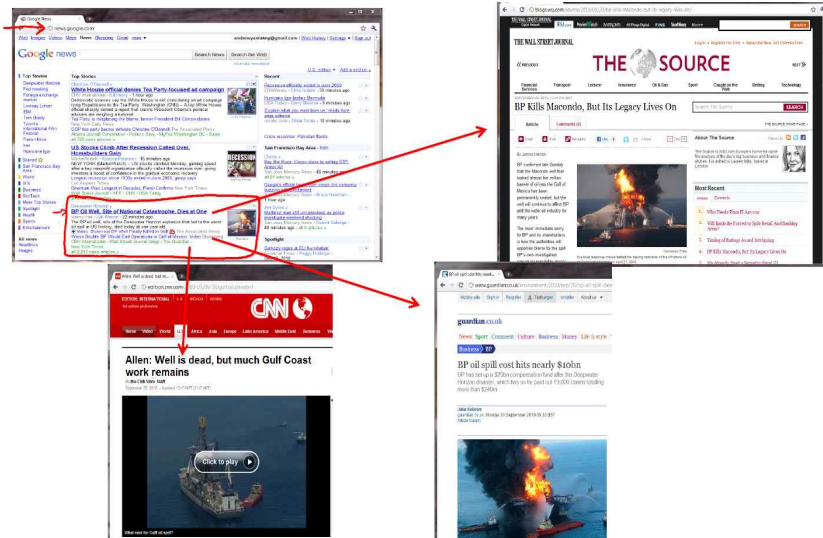


Manifold Hypothesis: Natural Data in High Dimensional Spaces Concentrates Close to Lower Dimensional Manifolds

Unsupervised Learning – Examples

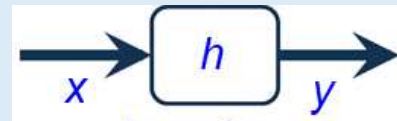
Ex.1) Google news


- Clustering: Given a set of news articles found on the web, group them into set of articles about the same story.



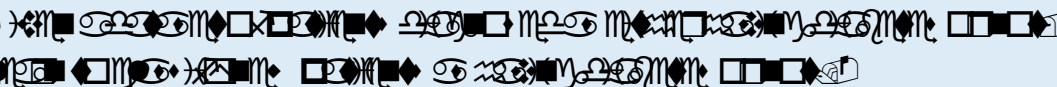
Quiz

◆ Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

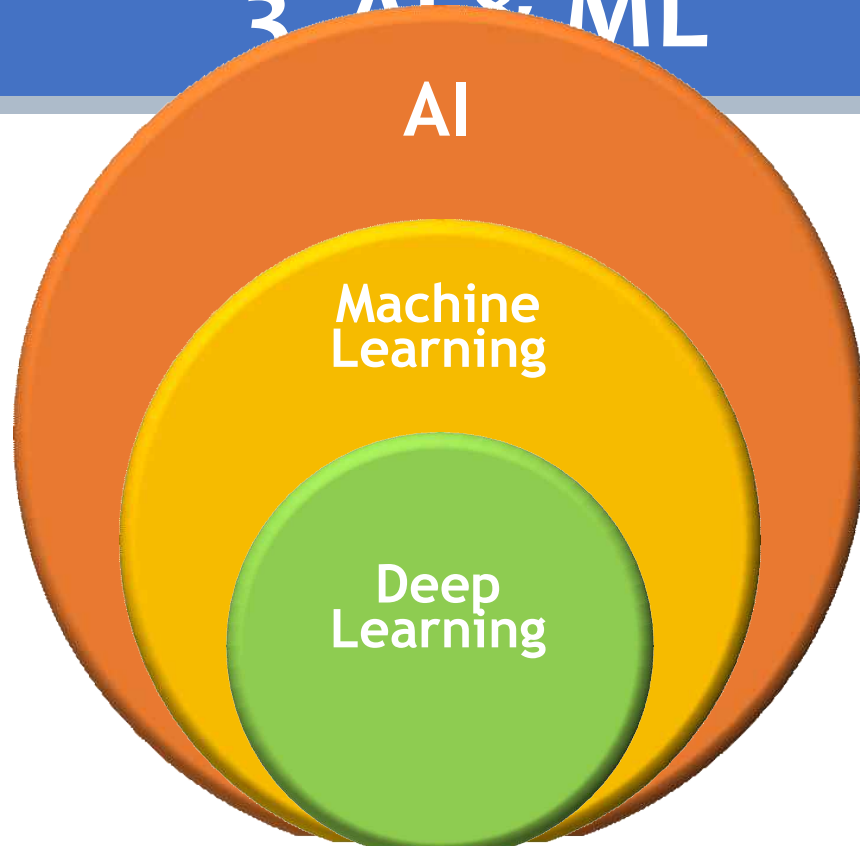


- Given email labeled as spam/not spam, learn a spam filter.
-  articles about the same story.

Given a database of customer data, automatically discover market segments and group customers into different market segments.

- 

3 AI & ML



AI (인공지능)

■ 인공지능의 탄생

- 컴퓨터의 뛰어난 능력
 - 사람이 어려워하는 일을 아주 쉽게 함
 - $80932.46789076 * 0.39001324$ 와 같은 곱셈을 고속으로 수행(현재는 초당 수십억개)
 - 복잡한 함수의 미분과 적분 적절
- 컴퓨터에 대한 기대감 (컴퓨터의 능력 과신)
 - 사람이 쉽게 하는 일, 예를 들어 고양이/개 구별하는 일도 잘 하지 않을까
 - 1950년대에 인공지능이라는 분야 등장



AI (cont'd)

■ 초창기는 지식기반 방식이 주류

- 예) “구멍이 2개이고 중간 부분이 홀쭉하며, 맨 위와 아래가 둥근 모양이라면 8이다”

■ 큰 깨달음

- 지식기반의 한계
- 단추를 “가운데 구멍이 몇 개 있는 물체”라고 규정하면 많은 오류 발생

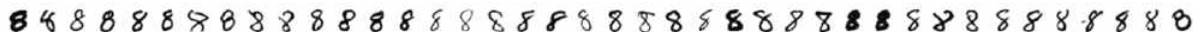


그림 1-2 인식 시스템이 대처해야 하는 심한 변화 양상(8과 단추라는 패턴을 어떻게 기술할 것인가?)

- 사람은 변화가 심한 장면을 아주 쉽게 인식하지만, 왜 그렇게 인식하는지 서술하지는 못함

ML (기계학습)

■ 인공지능의 주도권 전환

- 지식기반 → 기계 학습
- 기계 학습: 데이터 중심 접근방식

"지식기반에서 데이터중심으로"



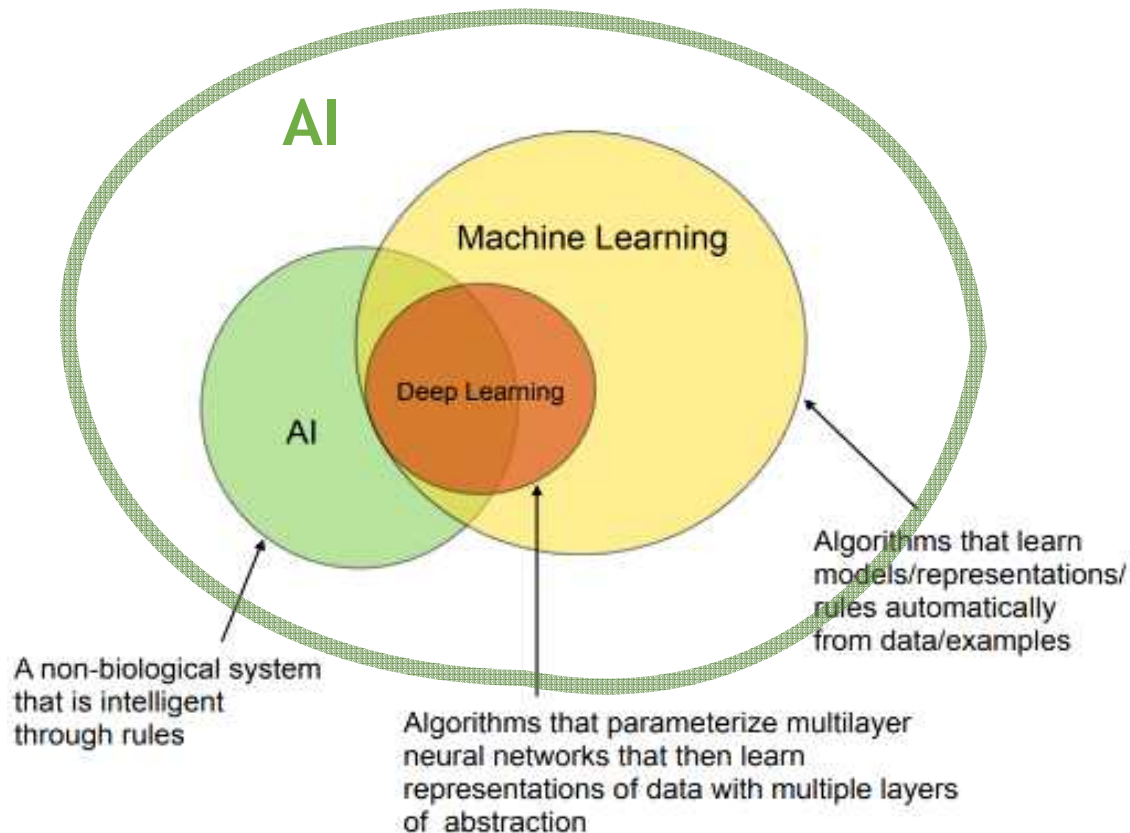
그림 1-3 기계 학습으로 만든 최첨단 인공지능 제품들

- 기계 학습은 인공지능 실현에 핵심 기술
- 기계 학습 알고리즘과 응용의 다양화
- 서로 다른 알고리즘과 응용의 융합
- 딥러닝이 기계 학습의 주류
- 표현 학습이 중요해짐

■ 리뷰 논문

- [LeCun2015, Jordan2015, Jones2014]

Where is Neural Network..?



[참고] AI와 ML의 간략한 역사 (1/3)

- 1843 에이다 “... 해석엔진은 꽤 복잡한 곡을 작곡할 수도 있다.”라는 논문 발표[Ada1843]
- 1950 인공지능 여부를 판별하는 튜링 테스트[Turing1950]
- 1956 최초의 인공지능 학술대회인 다트머스 콘퍼런스 개최. ‘인공지능’ 용어 탄생[McCarthy1955]
- 1958 로젠블랫이 퍼셉트론 제안[Rosenblatt1958]
 인공지능 언어 Lisp 탄생
- 1959 사무엘이 기계 학습을 이용한 체커 게임 프로그램 개발[Samuel1959]
- 1969 민스키가 퍼셉트론의 과대포장 지적. 신경망 내리막길 시작[Minsky1969]
 제1회 IJCA(International Joint Conference on Artificial Intelligence) 개최
- 1972 인공지능 언어 Prolog 탄생
- 1973 Lighthill 보고서로 인해 인공지능 내리막길, 인공지능 겨울AI winter 시작
- 1974 웨어보스가 오류 역전파 알고리즘을 기계 학습에 도입[Werbos1974]
- 1975경 의료진단 전문가 시스템 Mycin – 인공지능에 대한 관심 부활
- 1979 「IEEE Transactions on Pattern Analysis and Machine Intelligence」 저널 발간
- 1980 제1회 ICML(International Conference on Machine Learning) 개최
 후쿠시마가 NeoCognitron 제안[Fukushima1980]
- 1986 「Machine Learning」 저널 발간
 「Parallel Distributed Processing」 출간
 다층 퍼셉트론으로 신경망 부활



[참고] AI와 ML의 간략한 역사 (2/3)

- 1987 Lisp 머신의 시장 붕괴로 제2의 인공지능 겨울
 UCI 리포지토리 서비스 시작
 NIPS(Neural Information Processing Systems) 컨퍼런스 시작
- 1989 「Neural Computation」 저널 발간
- 1993 R 언어 탄생
- 1997 IBM 딥블루가 세계 체스 챔피언인 카스파로프 이김
 LSTM(Long short-term memory) 개발됨
- 1998경 SVM이 MNIST 인식 성능에서 신경망 추월
- 1998 르쿤이 CNN의 실용적인 학습 알고리즘 제안[LeCun1998]
 『Neural Networks: Tricks of the Trade』 출간
- 1999 NVIDIA사에서 GPU 공개
- 2000 「Journal of Machine Learning Research」 저널 발간
 OpenCV 최초 공개
- 2004 제1회 그랜드 챌린지(자율 주행)
- 2006 층별학습 탄생[Hinton2006a]
- 2007경 딥러닝이 MNIST 인식 성능에서 SVM 추월



[참고] AI와 ML의 간략한 역사 (3/3)

2007 GPU 프로그래밍 라이브러리인 CUDA 공개

어번 챌린지(도심 자율 주행)

Scikit-learn 라이브러리 최초 공개

2009 Theano 서비스 시작

2010 ImageNet 탄생

제1회 ILSVRC 대회

2011 IBM 왓슨이 제퍼디 우승자 꺾음

2012 MNIST에 대해 0.23% 오류율 달성

AlexNet 발표 (3회 ILSVRC 우승)

2013 제1회 ICLR(International Conference on Learning Representations) 개최

2014 Caffe 서비스 시작

2015 TensorFlow 서비스 시작

OpenAI 창립

2016 알파고와 이세돌의 바둑 대회에서 알파고 승리[Silver2016]

『Deep Learning』 출간

2017 알파고 제로[Silver2017]



4. Datasets

1. Public Datasets
2. Sparsity of Datasets
3. Data Visualization

Public Datasets

◆ The Three Famous Datasets

- Iris (<https://archive.ics.uci.edu/ml/datasets/iris>)
- MNIST (<http://yann.lecun.com/exdb/mnist/>)
- ImageNet (<http://www.image-net.org/>)

◆ UC Irvine: UCI ML Repository

- <http://archive.ics.uci.edu/ml/index.html>

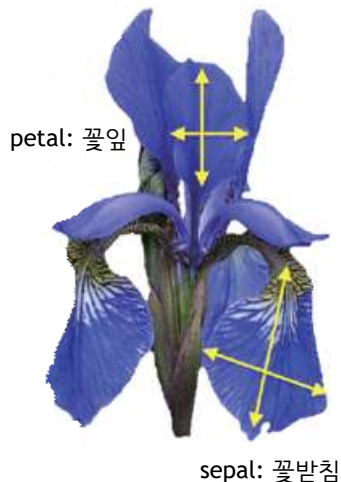
◆ Wikipedia: List of datasets for machine-learning research

- https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research



Iris Dataset

- Iris 데이터베이스는 통계학자인 피셔 교수가 1936년에 캐나다 동부 해안의 가스페 반도에 서식하는 3종의 붓꽃(*setosa*, *versicolor*, *virginica*)을 50송이씩 채취하여 만들었다[Fisher1936]. 150개 샘플 각각에 대해 꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비를 측정하여 기록하였다. 따라서 4차원 특징 공간이 형성되며 목꽃값은 3종을 숫자로 표시함으로써 1, 2, 3 값 중의 하나이다. <http://archive.ics.uci.edu/ml/datasets/iris>에 접속하여 내려받을 수 있다.



Setosa



Versicolor



Virginica



특징 (feature) 종 (label)

Sepal length ♦	Sepal width ♦	Petal length ♦	Petal width ♦	Species ♦
5.2	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
7.0	3.2	4.7	1.4	<i>I. versicolor</i>
6.4	3.2	4.5	1.5	<i>I. versicolor</i>
6.9	3.1	4.9	1.5	<i>I. versicolor</i>
5.5	2.3	4.0	1.3	<i>I. versicolor</i>
6.3	3.3	6.0	2.5	<i>I. virginica</i>
5.8	2.7	5.1	1.9	<i>I. virginica</i>
7.1	3.0	5.9	2.1	<i>I. virginica</i>
6.3	2.9	5.6	1.8	<i>I. virginica</i>

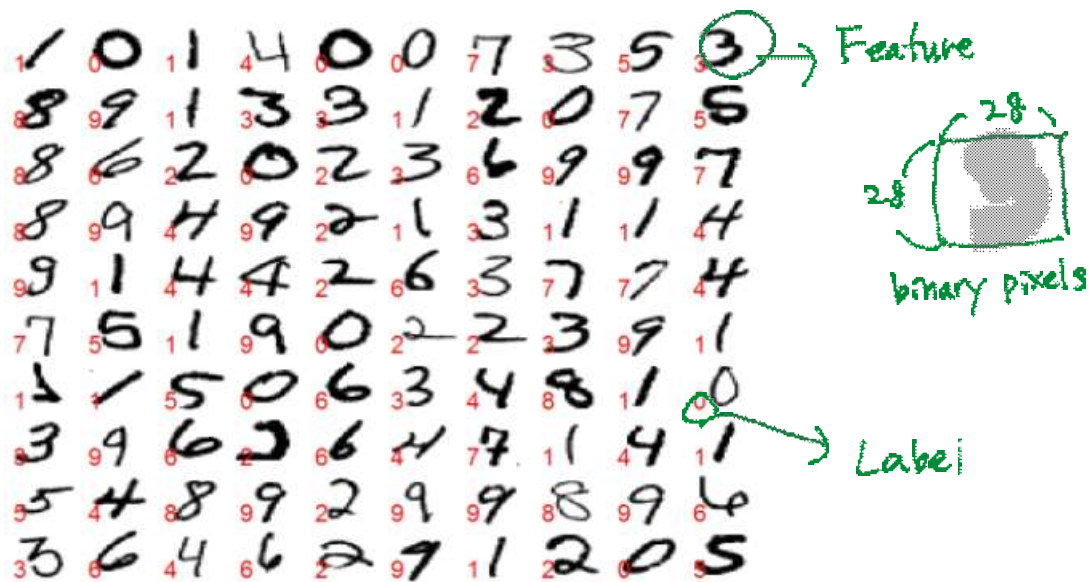
$$X \in \mathbb{R}^{150 \times 4}$$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

MNIST Dataset

- MNIST 데이터베이스는 미국표준국(NIST)에서 수집한 필기 숫자 데이터베이스로, 훈련집합 60,000자, 테스트집합 10,000자를 제공한다. <http://yann.lecun.com/exdb/mnist>에 접속하면 무료로 내려받을 수 있으며, 1988년부터 시작한 인식률 경쟁 기록도 볼 수 있다. 2017년 8월 기준으로는 [Ciresan2012] 논문이 0.23%의 오류율로 최고 자리를 차지하고 있다. 테스트집합에 있는 10,000개 샘플에서 단지 23개만 틀린 것이다.

- 훈련집합
- 테스트집합



ImageNet Dataset

- ImageNet 데이터베이스는 정보검색 분야에서 만든 WordNet의 단어 계층 분류를 그대로 따랐고, 부류마다 수백에서 수천 개의 영상을 수집하였다[Deng2009]. 총 21,841개 부류에 대해 총 14,197,122개의 영상을 보유하고 있다. 그중에서 1,000개 부류를 뽑아 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)라는 영상인식 경진대회를 2010년부터 매년 개최하고 있다. 대회 결과에 대한 자세한 내용은 4.4절을 참조하라. <http://image-net.org>에서 내려받을 수 있다.

• 부류 (class)



(a) 'swing' 부류

그림 4-20 ImageNet의 예제 영상



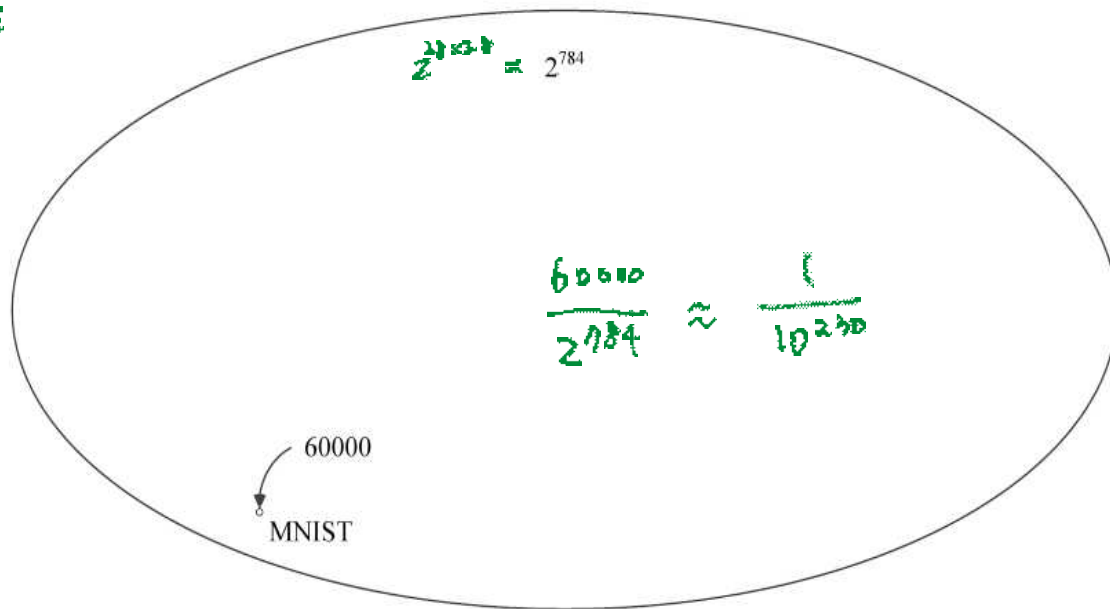
(b) 'Great white shark' 부류

↑ label

Cf.) Sparsity of Datasets

◆ 특징 공간의 크기와 데이터셋의 크기

- 데이터셋의 크기(샘플 개수)는 통상 특징 공간의 크기에 비해 매우 작다 (매우 희소하다).
- 예) MNIST: 28×28 흑백 비트맵이라면 서로 다른 총 샘플 수는 2^{784} 가지이지만, MNIST는 고작 6만 개 샘플



Cf.) Data Visualization

◆ 4차원 이상의 초공간은 한꺼번에 가시화 불가능

그림에 있는 ●=setosa, ●=versicolor, ●=virginica

◆ 여러 가지 가시화 기법

- 예) 2개씩 조합하여 여러 개 그래프로 그림

Sepal length ◊	Sepal width ◊	Petal length ◊	Petal width ◊	Species ◊
5.2	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
7.0	3.2	4.7	1.4	<i>I. versicolor</i>
6.4	3.2	4.5	1.5	<i>I. versicolor</i>
6.9	3.1	4.9	1.5	<i>I. versicolor</i>
5.5	2.3	4.0	1.3	<i>I. versicolor</i>
6.3	3.3	6.0	2.5	<i>I. virginica</i>
5.8	2.7	5.1	1.9	<i>I. virginica</i>
7.1	3.0	5.9	2.1	<i>I. virginica</i>
6.3	2.9	5.6	1.8	<i>I. virginica</i>

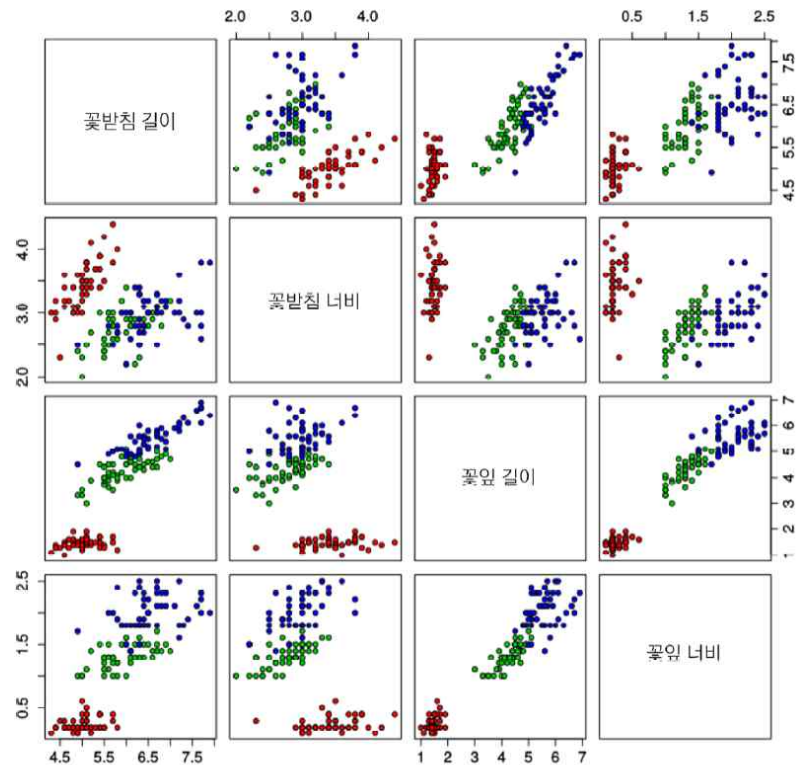


그림 1-10 고차원 특징 공간의 가시화

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

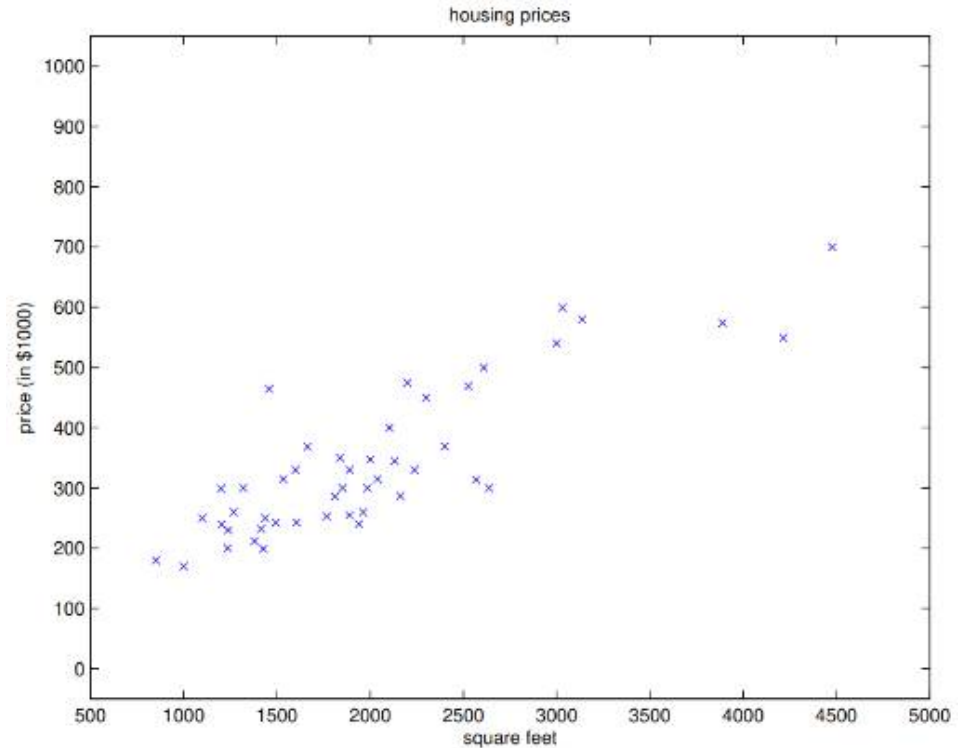


1. Problem Definition

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

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

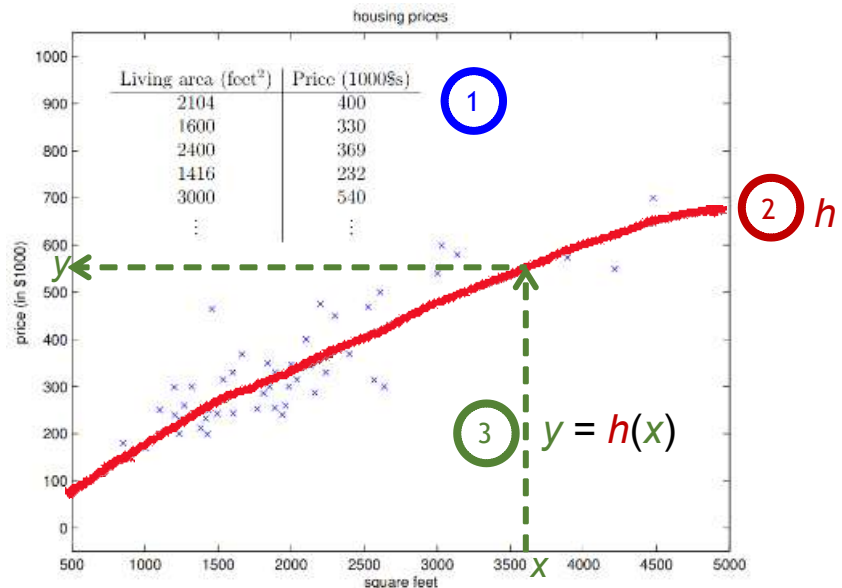
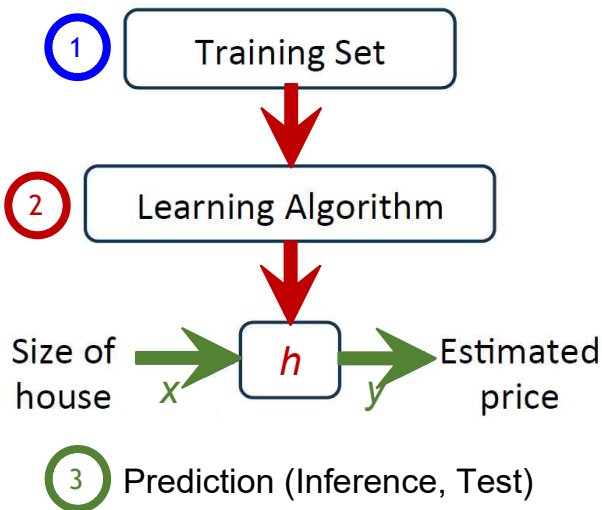
- ◆ Given data like this,
how can we *learn to predict*
the prices of other houses?



1. Problem Definition (cont'd)

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

- ~~randomly chosen~~ **'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.

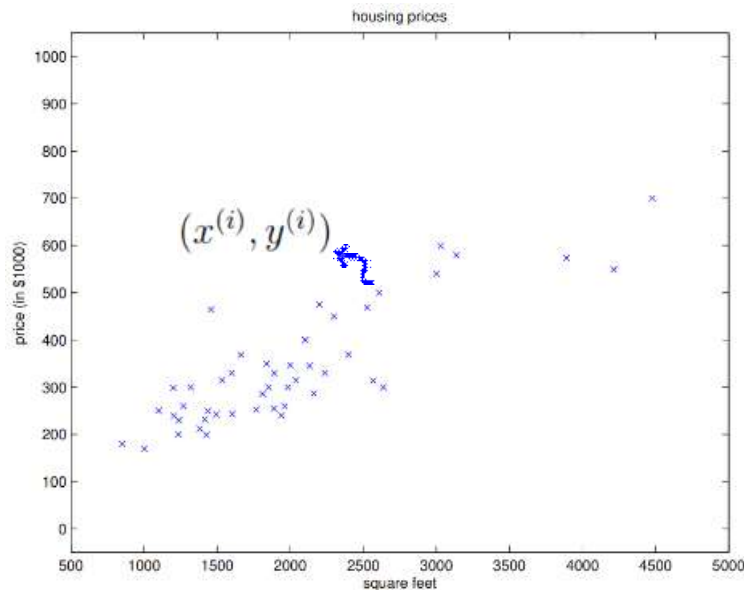


2. Data Representation

◆ Training Set

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

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



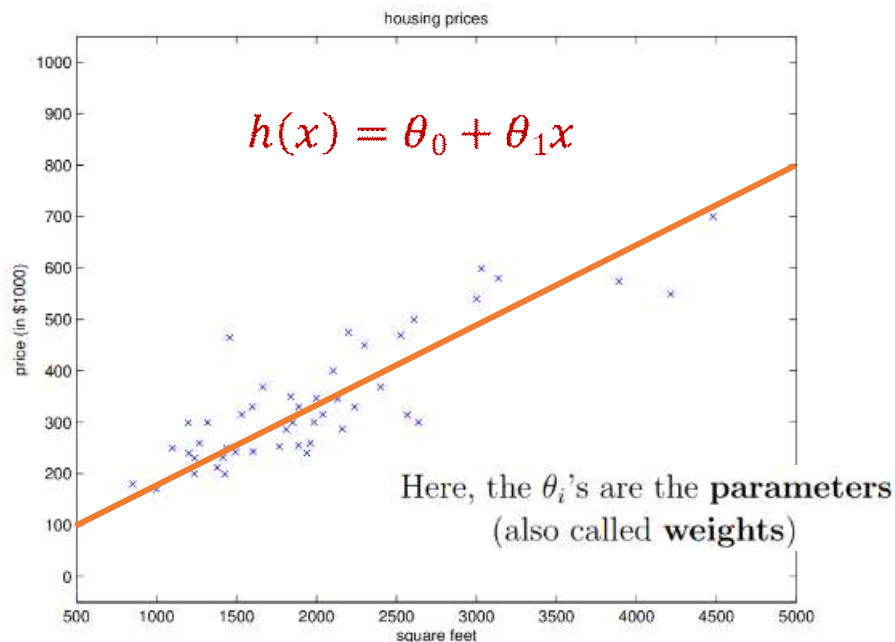
Design Matrix : $X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix}$

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

3. Model Representation

◆ Linear Model (Linear Hypothesis)

- We call this problem '*linear regression*'



- Vector Notation for Linear Model

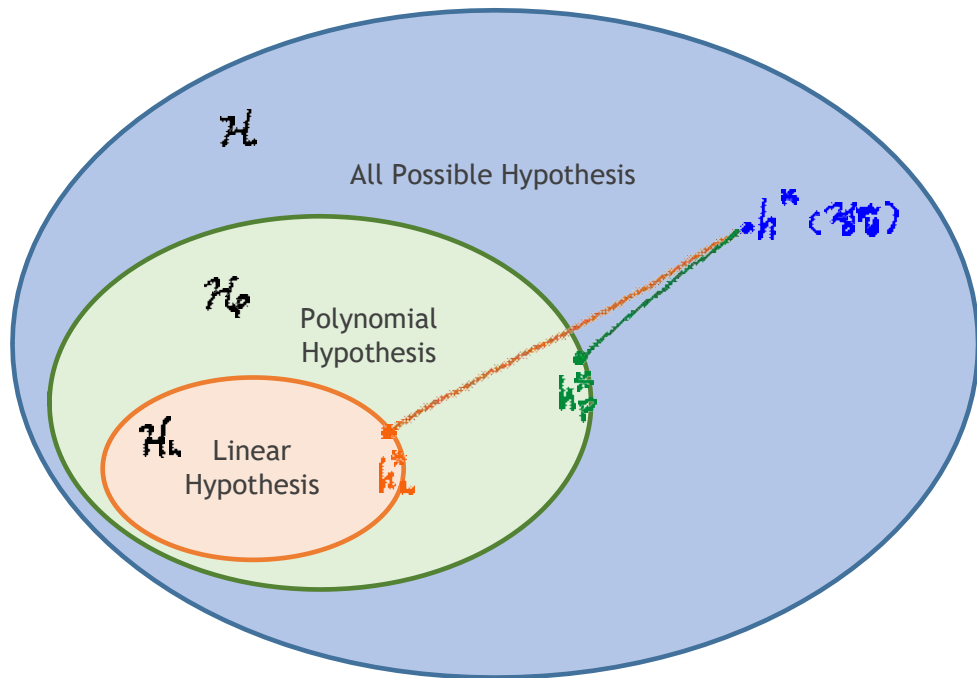
$$\text{Let } \underline{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}, \underline{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}.$$

$$\begin{aligned} \text{Then } y &= h_{\theta}(\underline{x}) \\ &= \underline{\theta}^T \underline{x} = \underline{x}^T \underline{\theta} \end{aligned}$$

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

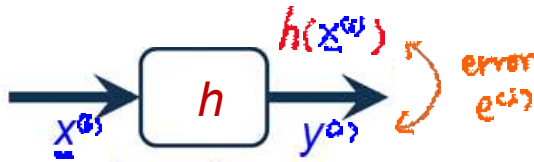
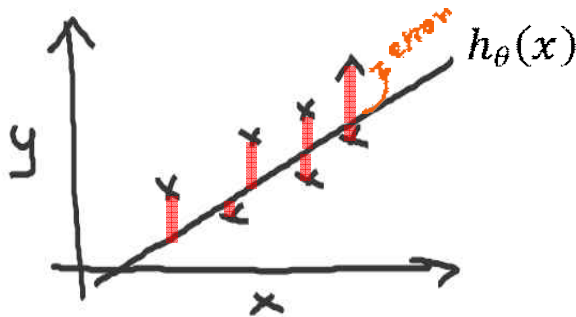
- $h \in \mathcal{H}_L$: $h(x) = \theta_0 + \theta_1 x$
- $h \in \mathcal{H}_P$: $h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
- $h \in \mathcal{H}$: $h(x) = \theta_0 + \cos(x) + e^{x^2}$

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):

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

Vector form of the MSE:

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

$$\begin{aligned} \text{Then } J &= \frac{1}{2} \|\underline{e}\|_2^2 = \frac{1}{2} \underline{e}^T \underline{e} \\ &= \frac{1}{2} \|\hat{\underline{y}} - \underline{y}\|_2^2 = \frac{1}{2} (\hat{\underline{y}} - \underline{y})^T (\hat{\underline{y}} - \underline{y}) \end{aligned}$$

4. Cost Function – Iterative Minimization Example

- 훈련집합

$$\mathbb{X} = \{x_1 = 2.0, x_2 = 4.0, x_3 = 6.0, x_4 = 8.0\},$$

$$\mathbb{Y} = \{y_1 = 3.0, y_2 = 4.0, y_3 = 5.0, y_4 = 6.0\}$$

- 초기 매개변수 $\theta_1 = (0.1, 4.0)^T$ 라 가정 $\rightarrow J(\theta_1) = 0.4$

$$x_1, y_1 \rightarrow (f_{\theta_1}(2.0) - 3.0)^2 = ((0.1 * 2.0 + 4.0) - 3.0)^2 = 1.44$$

$$x_2, y_2 \rightarrow (f_{\theta_1}(4.0) - 4.0)^2 = ((0.1 * 4.0 + 4.0) - 4.0)^2 = 0.16$$

$$x_3, y_3 \rightarrow (f_{\theta_1}(6.0) - 5.0)^2 = ((0.1 * 6.0 + 4.0) - 5.0)^2 = 0.16$$

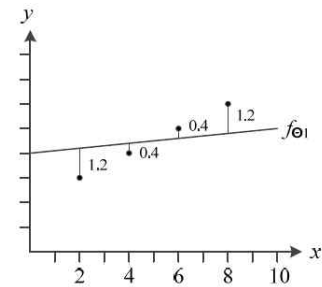
$$x_4, y_4 \rightarrow (f_{\theta_1}(8.0) - 6.0)^2 = ((0.1 * 8.0 + 4.0) - 6.0)^2 = 1.44$$

$$\theta = (\theta_1, \theta_0)^T$$

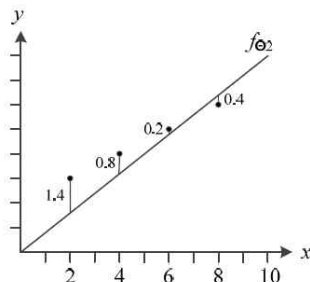
↑ ↑
1.44 0.16

- θ_1 을 개선하여 $\theta_2 = (0.8, 0.0)^T$ 가 되었다고 가정 $\rightarrow J(\theta_2) = 0.35$

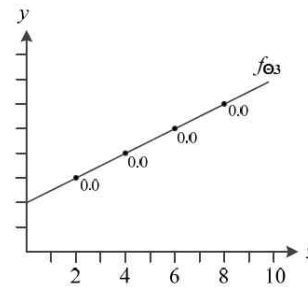
- θ_2 를 개선하여 $\theta_3 = (0.5, 2.0)^T$ 가 되었다고 가정 $\rightarrow J(\theta_3) = 0.0$



(a) 초기 매개변수 θ_1




(b) θ_1 을 개선하여 θ_2 가 됨

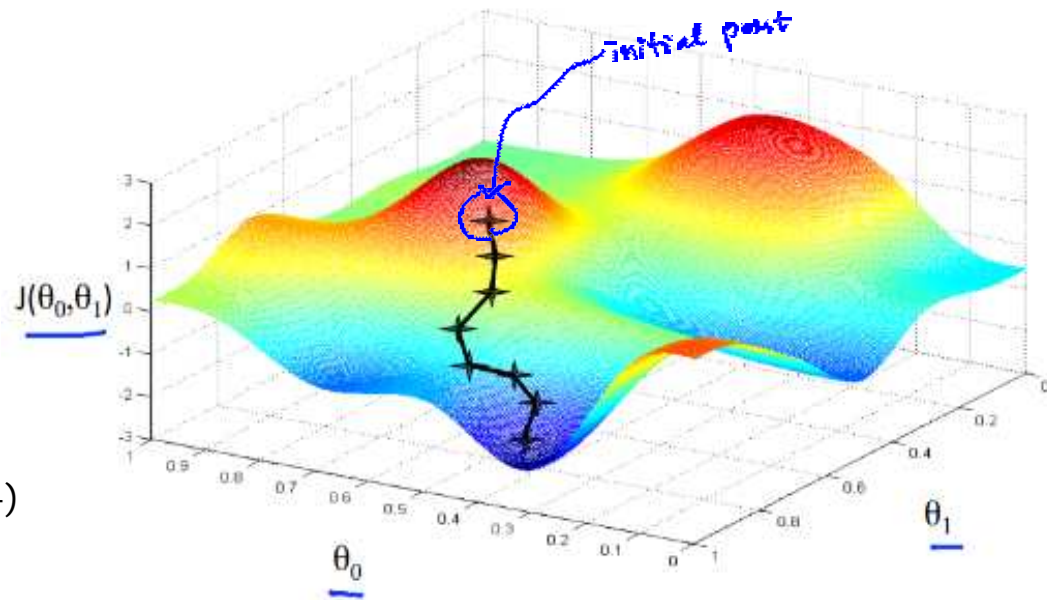


(c) θ_2 를 개선하여 최적의 θ_3 을 찾음

5. Optimization (최적화)

◆ Iterative Methods (반복법)

- Gradient Descent (GD)
- 
- Newton's Method



◆ Analytic Methods (해석적 방법)

- Normal Equation (Linear Model의 경우)

◆ *We will study them later...*

6. Extensions to Multiple Features

◆ Training Set

$$\underline{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$$

	Living area (feet ²)	#bedrooms	Price (1000\$)
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	2104 $x_1^{(1)}$	3 $x_2^{(1)}$	400 $y^{(1)}$
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	1600 $x_1^{(2)}$	3 $x_2^{(2)}$	330 $y^{(2)}$
\vdots	2400 \vdots	3 \vdots	369 \vdots
\vdots	1416 \vdots	2 \vdots	232 \vdots
\vdots	3000 \vdots	4 \vdots	540 \vdots
$\begin{bmatrix} 1 \\ n \end{bmatrix}$	$x_1^{(n)}$	$x_2^{(n)}$	$y^{(n)}$



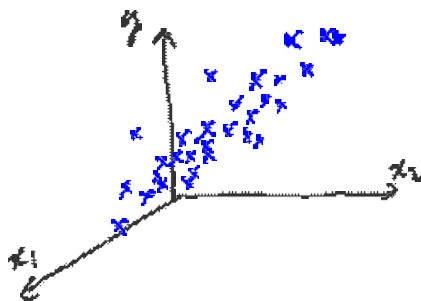
◆ Data Representation

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

- Design matrix:

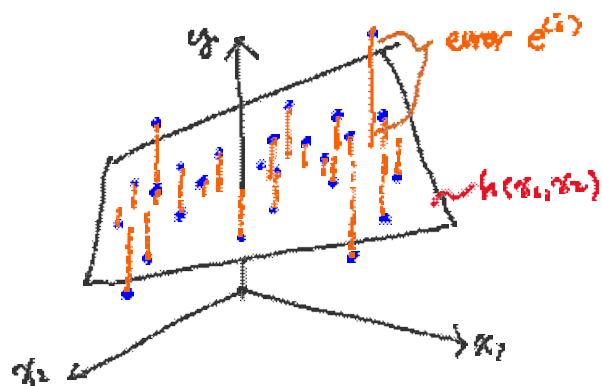
$$X = \begin{bmatrix} \underline{x}_1^{(1)} & \underline{x}_2^{(1)} \\ \underline{x}_1^{(2)} & \underline{x}_2^{(2)} \\ \vdots & \vdots \\ \underline{x}_1^{(n)} & \underline{x}_2^{(n)} \end{bmatrix} = \begin{bmatrix} -\underline{x}^{(1)T} - \\ -\underline{x}^{(2)T} - \\ \vdots - \\ -\underline{x}^{(n)T} - \end{bmatrix}$$

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



6. Extensions to Multiple Features (cont'd)

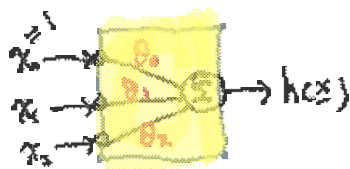
◆ Model Representation (Linear Model)



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

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

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



◆ Cost Function (MSE Cost)

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (e^{(i)})^2$$

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

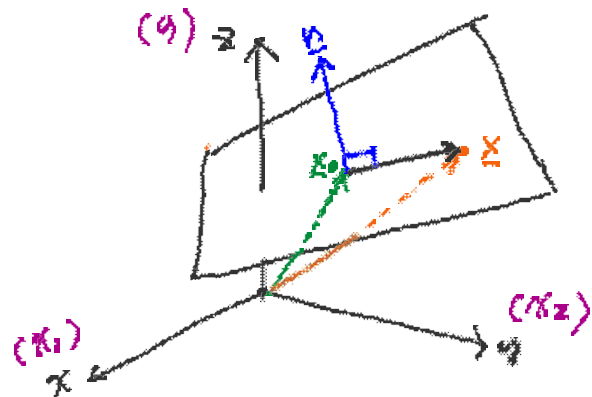
$$\text{Let } \underline{e} = \begin{bmatrix} e^{(1)} \\ \vdots \\ e^{(m)} \end{bmatrix}$$

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

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

[Remind] 3차원 공간에서 평면의 방정식

◆ 점 \underline{x}_0 를 지나는 벡터 \underline{n} 에 수직인 평면의 방정식은 ?



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

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

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

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

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

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

$$\text{or } ax + by + cz = d$$

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

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

* Hyper-plane : n 차원 공간상에서 $(n-1)$ 차원 부분공간을 만듦
(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)

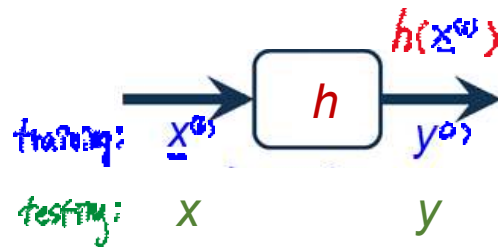
- what we want to predict
- synonymous to correct output, ground truth, label (in classification)

◆ 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

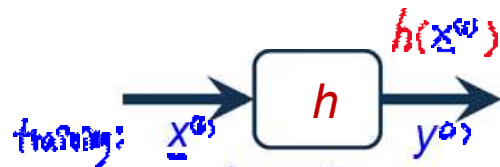
- a collection of training examples



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.

