AzureML로 시작하는 Machine Learning

Microsoft Student Partner

KAIST 정태영

Session

- Session 1 : Introduction to Introduction to ML

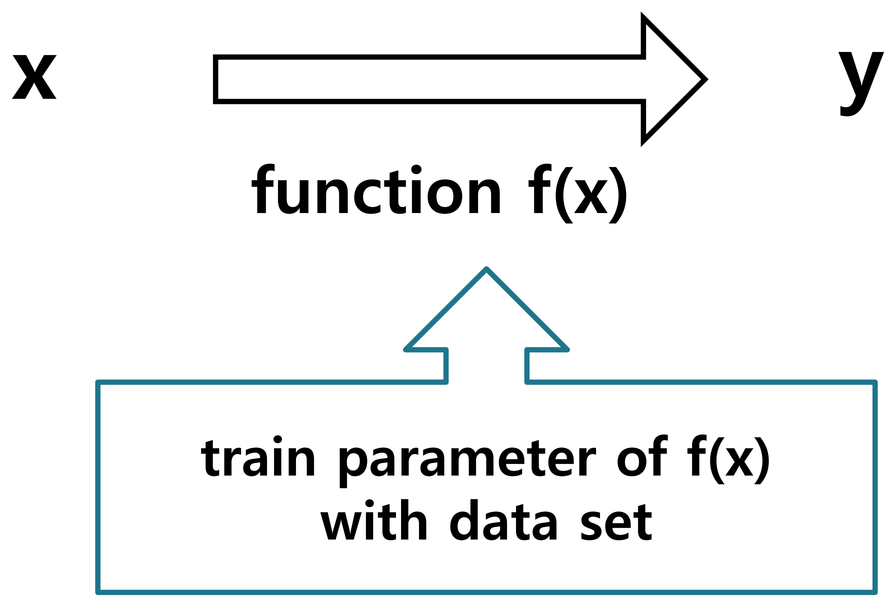
  머신러닝의 기초의 기초에 대해 배웁니다.

- Session 2 : Azure Machine Learning 맛보기

  Azure ML studio를 통해 붓꽃을 분류하는
  머신 러닝 프로그램을 만들어봅니다.

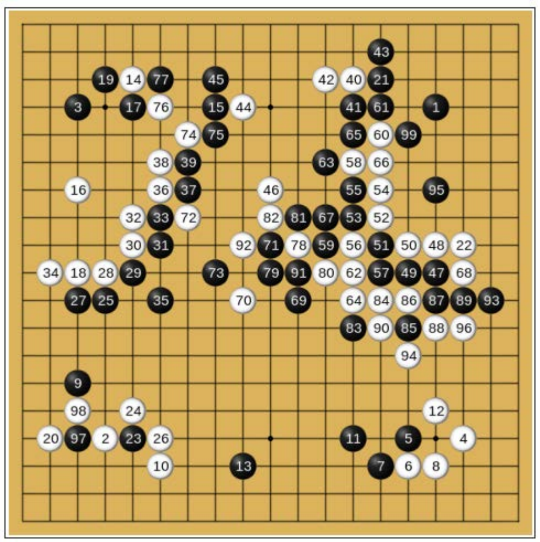
What is ML?

- 명시적으로 프로그래밍 되지 않은 것을 컴퓨터가 학습할 수 있게 하는 학문
- 데이터를 통해 학습하는 알고리즘에 관한 학문

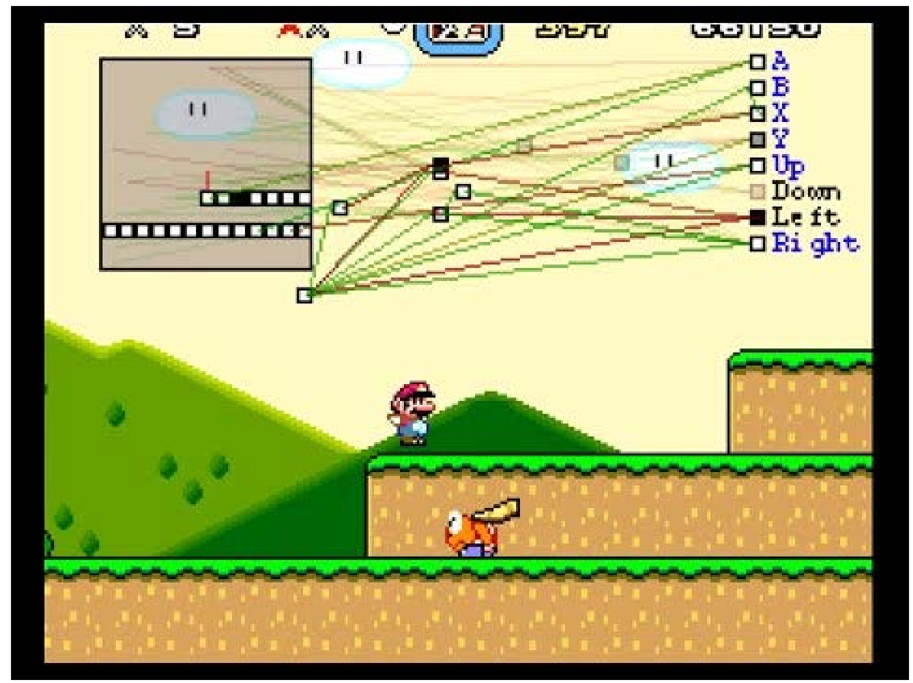


ML Applications

- 인공지능

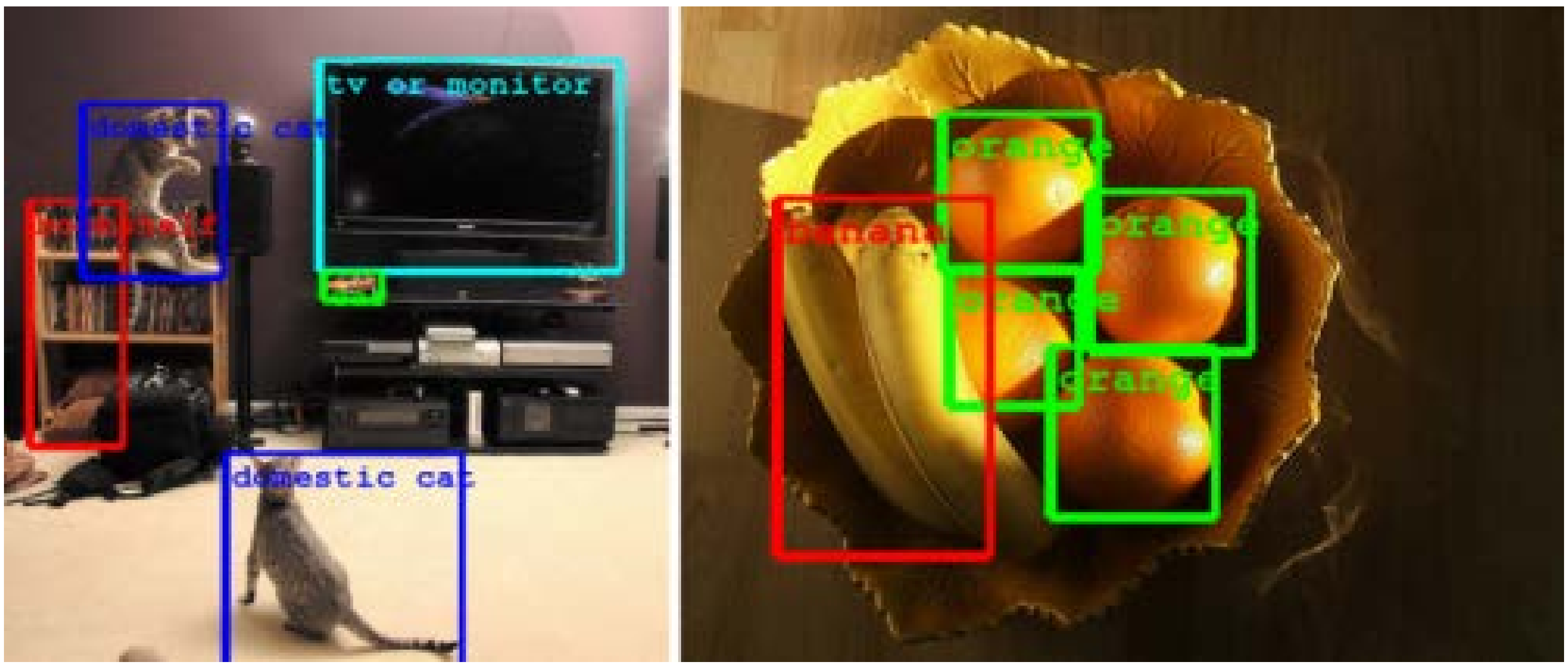


AlphaGo



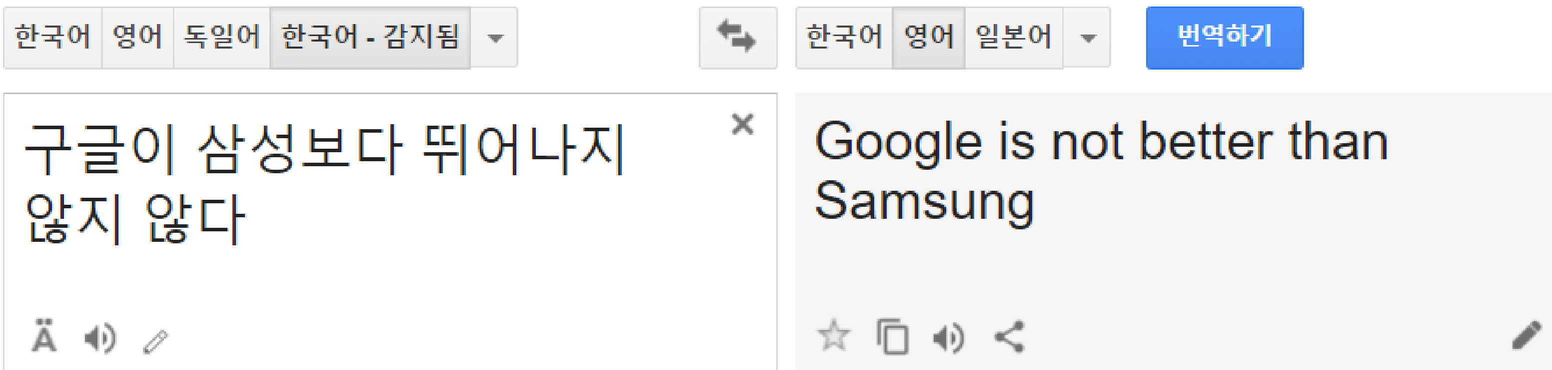
ML Applications

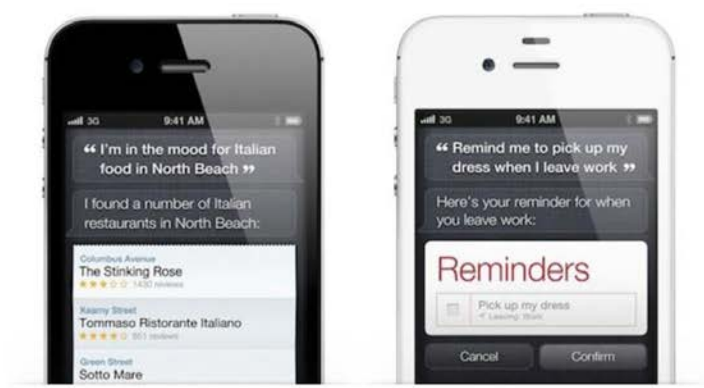
- 시각 정보 처리

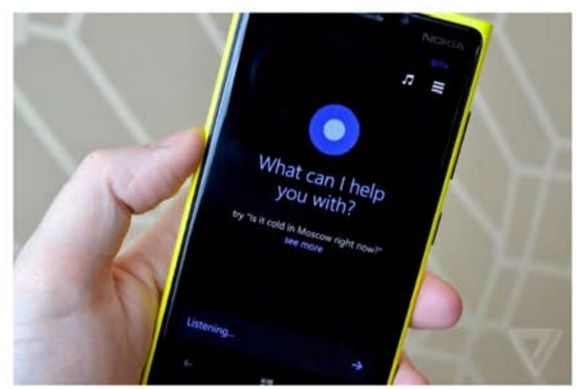


ML Applications

- 언어

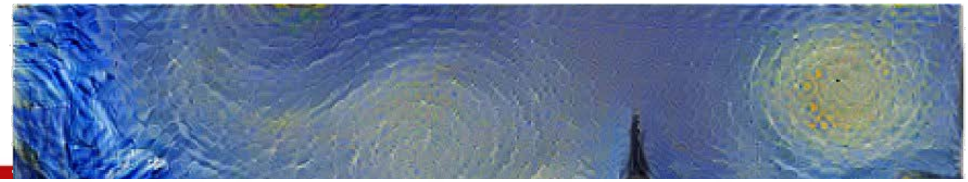


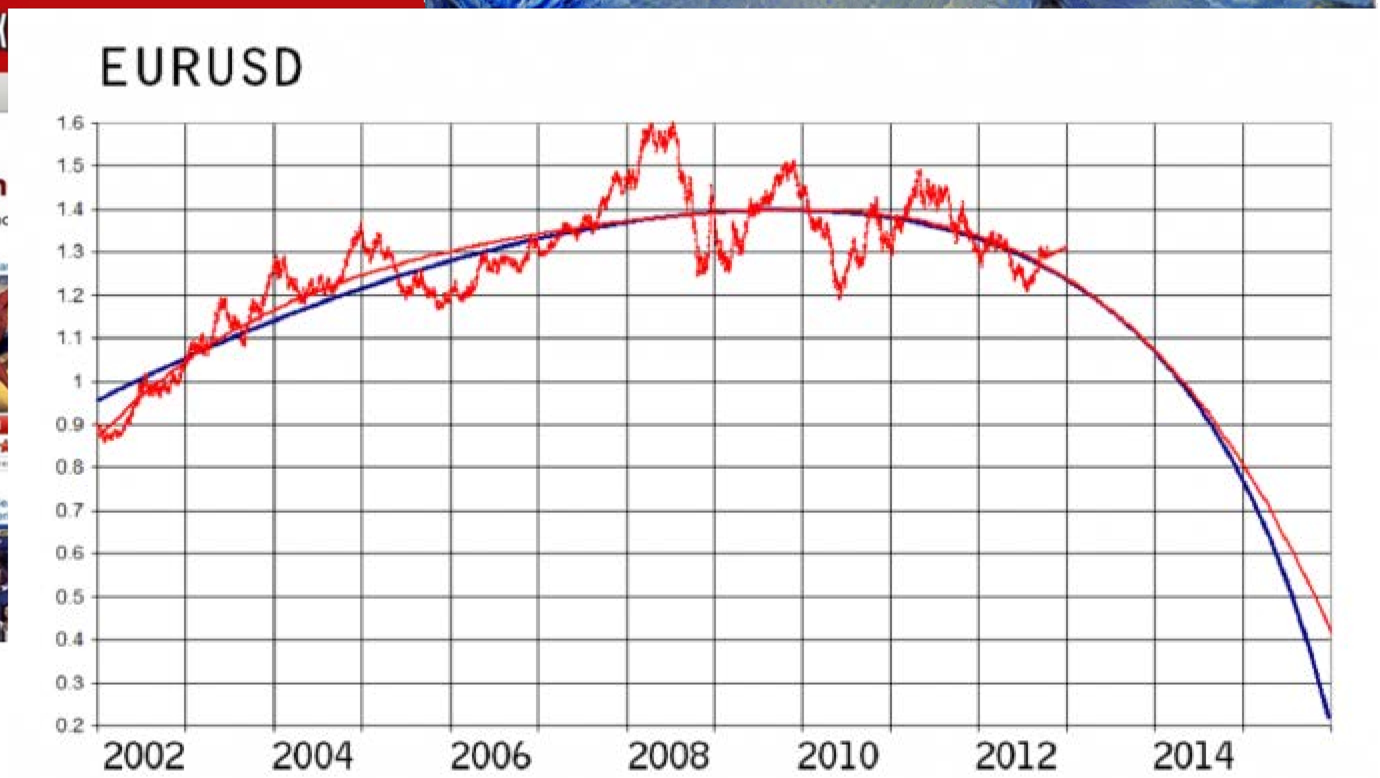




ML Applications

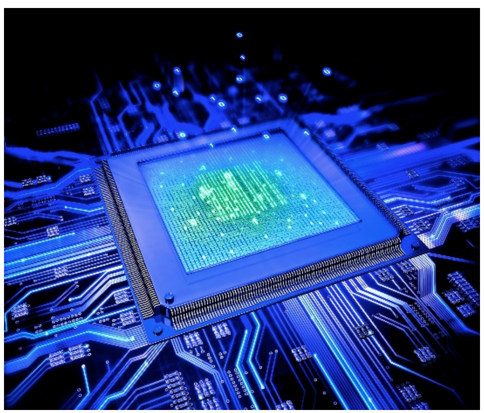
- 기타



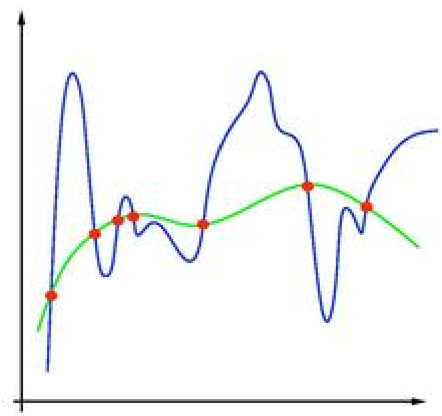


Why ML?

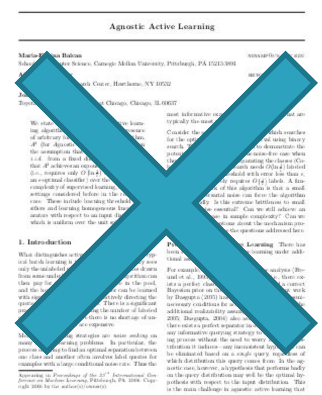
- 특히, 왜 deep learning이 떠오를까?
- 과거 Deep learning의 문제들



Computation Cost



Overfitting


Agnostic Active Learning

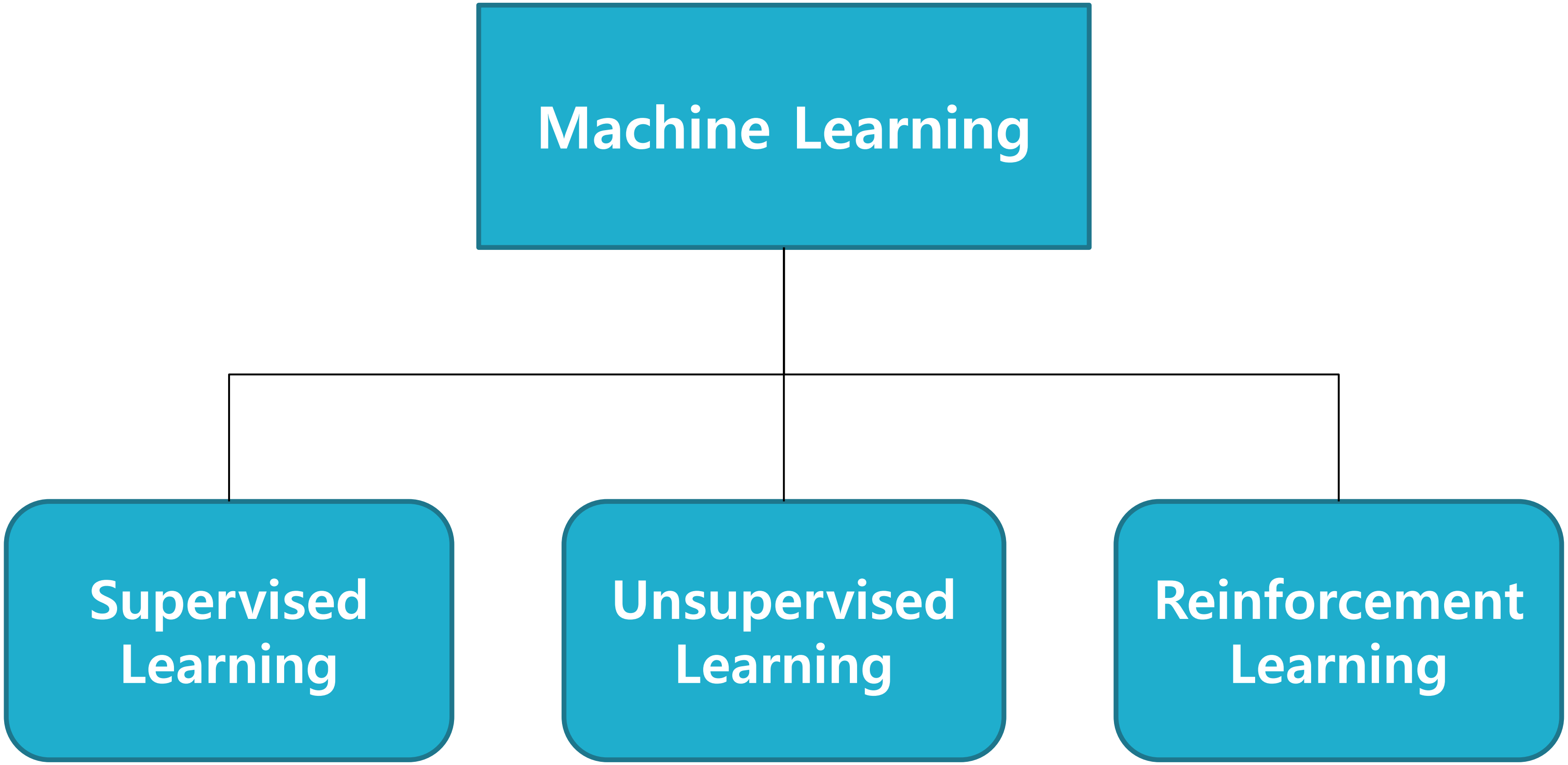
No theorical base

Why ML?

- 현재

- Overfitting : Regularization(dropout, ReLU..)
- 연산 능력 : 하드웨어의 발전, 병렬 처리
- 이론적 배경 : 실제 결과가 좋음

Category of ML



Category of ML

- 지도학습 : Label된 데이터를 통해 훈련

- 회귀분석(regression), 분류(classification)
- 집값 예측, 개/고양이 사진 판단

- 비지도학습 : Label이 없는 데이터를 통해 훈련

- Clustering, Dimensionality reduction

- 강화학습 : (상태, 행동)에 대한 보상으로 훈련

- Markov Decission Process
- 게임 AI(알파고)

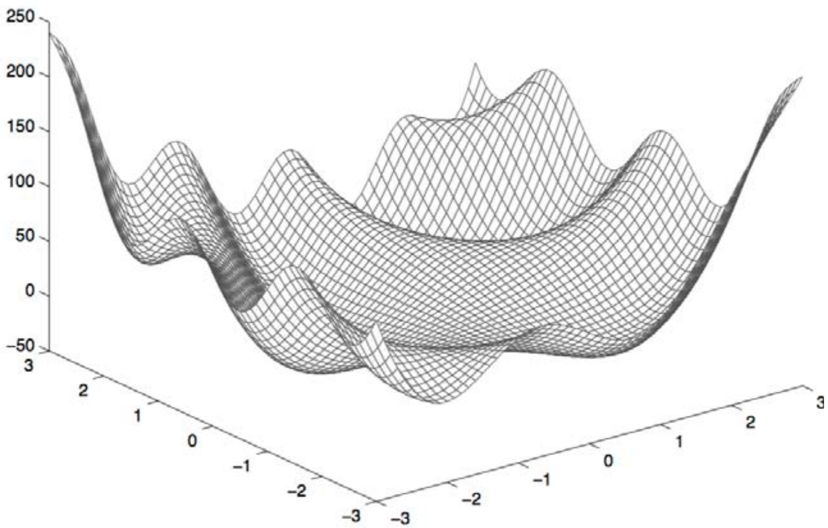
Math for ML

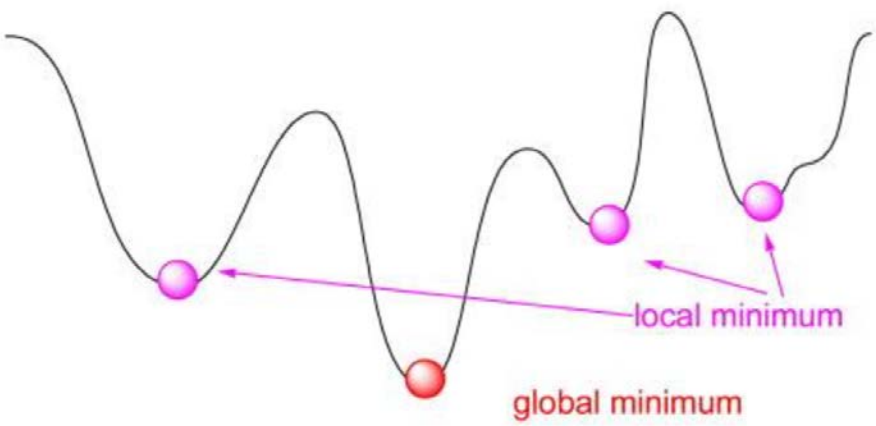
- 확률, 통계, 선형대수, 최적화 이론
- Optimization

$$\begin{aligned} & minimize\ f_0(x) \\ s.t\ \ & f_i(x) \leq 0, i = 1, 2, \dots, k \\ & h_j(x) = 0, j = 1, 2, \dots, l \end{aligned}$$

- 복잡한 그래프에서는
  최적의 해를 찾기 힘듦

Math for ML

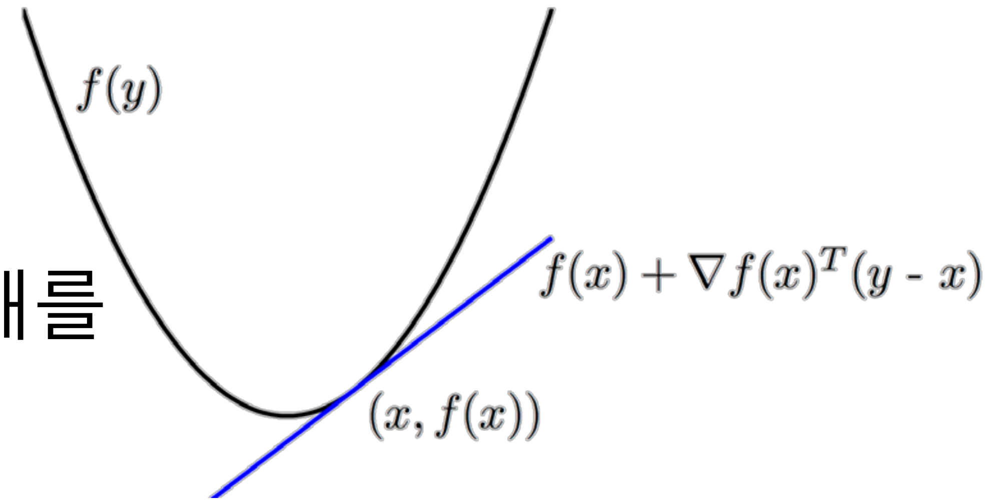

250
200
150
100
50
0
−50
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


local minimum
global minimum

Math for ML

- Convex Optimization (볼록최적화)

  볼록한 함수에서 최적해를 찾는 것은 훨씬 쉽다.



Math for ML

$$
\begin{aligned}
&minimize\ f_0(x) \\
s.t\quad & f_i(x) \leq 0, i = 1, 2, \dots, k \\
& h_j(x) = 0, j = 1, 2, \dots, l
\end{aligned}
$$

Lagrangian function with $\lambda_i \geq 0, v_j \ \in R$

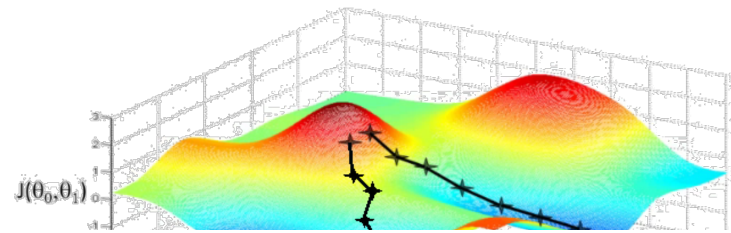
$$L(x, \lambda, v) = f_0(x) + \sum_{i=1}^{k} \lambda_i f_i(x) + \sum_{j=1}^{l} v_j h_j(x)$$

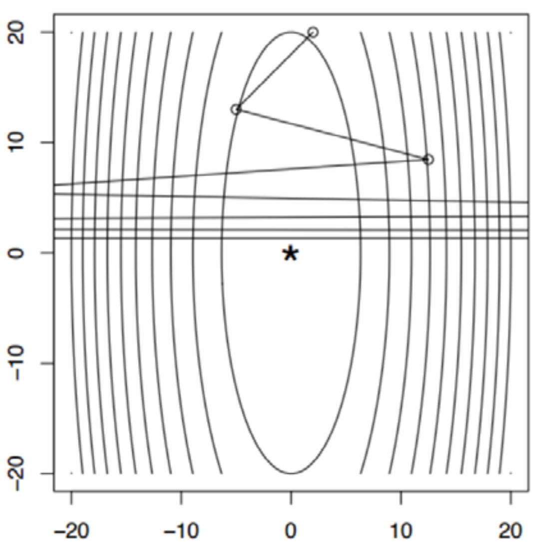
$$maximize_{\lambda \geq 0, v} \left[\inf_{\mathrm{x}} L(x, \lambda, v) \right]$$

Math for ML

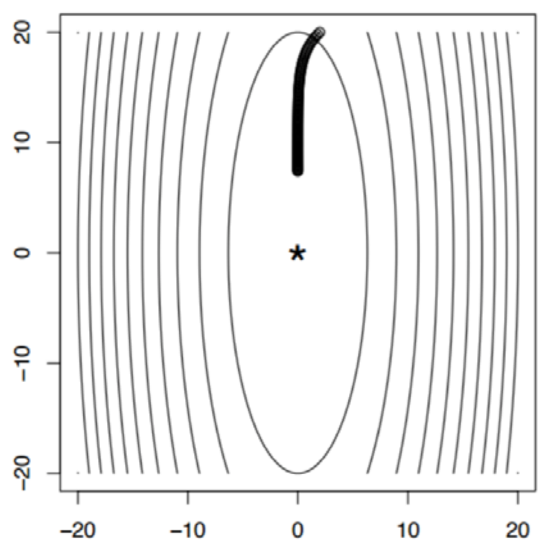
- Gradient Descent Method

  현재 기울기를 계산해
  더 좋은 방향으로 이동

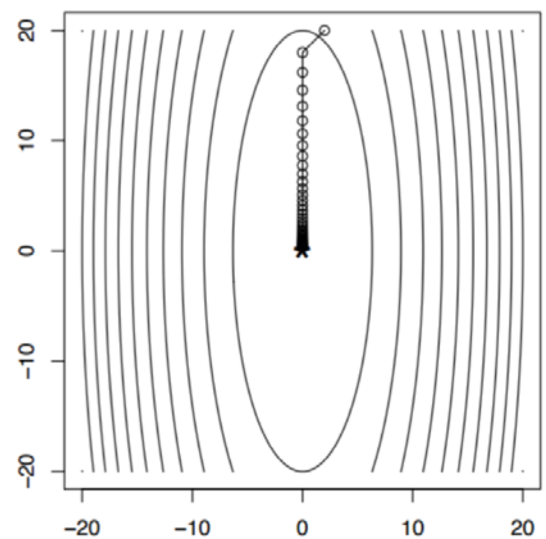




t is too large



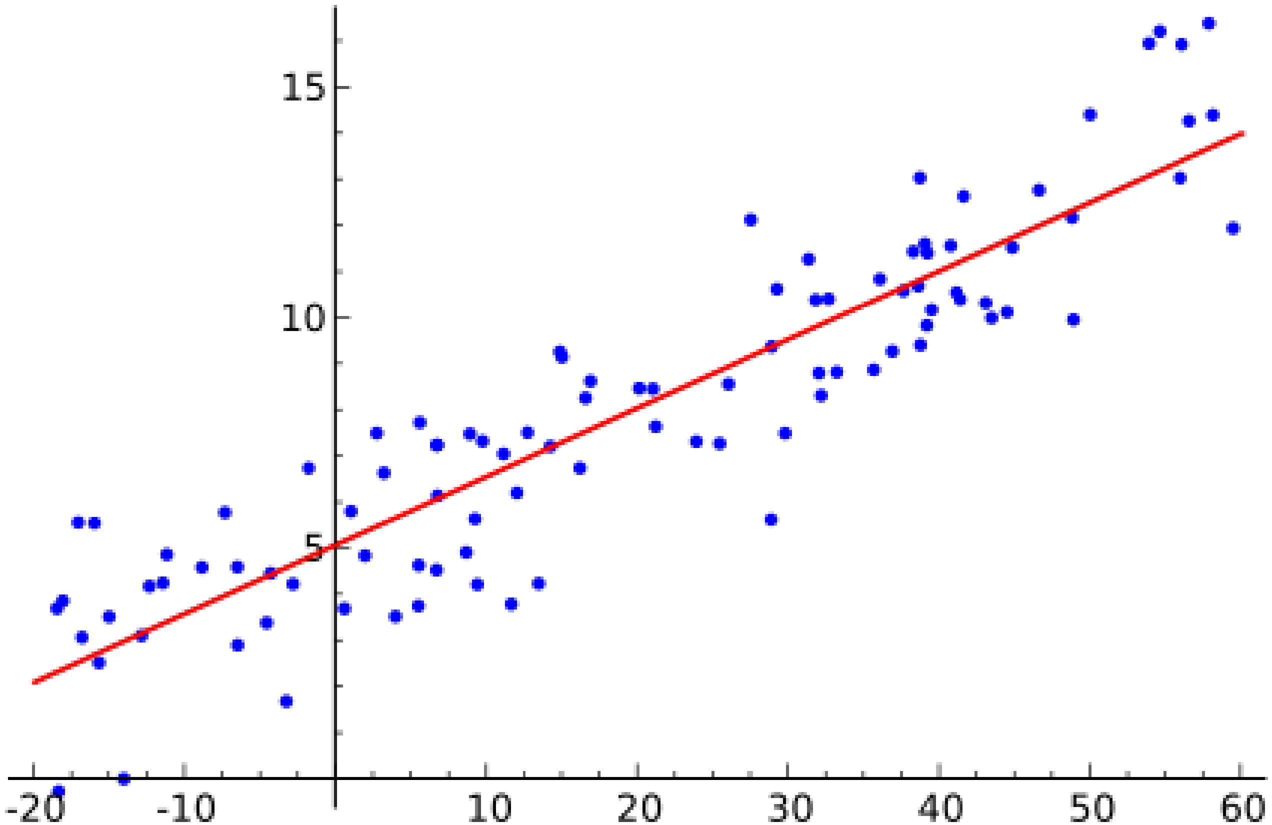
t is too small



appropriate

Regression

- 회귀분석: 주어진 데이터에 가장 적합한 함수 찾기



Regression

- 선형회귀 : 선형 함수를 찾는 회귀분석

$X \in R^{n \times p}, y \in R^n$일 때 $y = XA$에 가장 가까운 $\mathrm{A} \in R^p$

$$minimize_A \ \ \|y - XA\|_2^2$$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}$$

Regression

$$\nabla \|y - XA\|_2^2 = 0$$

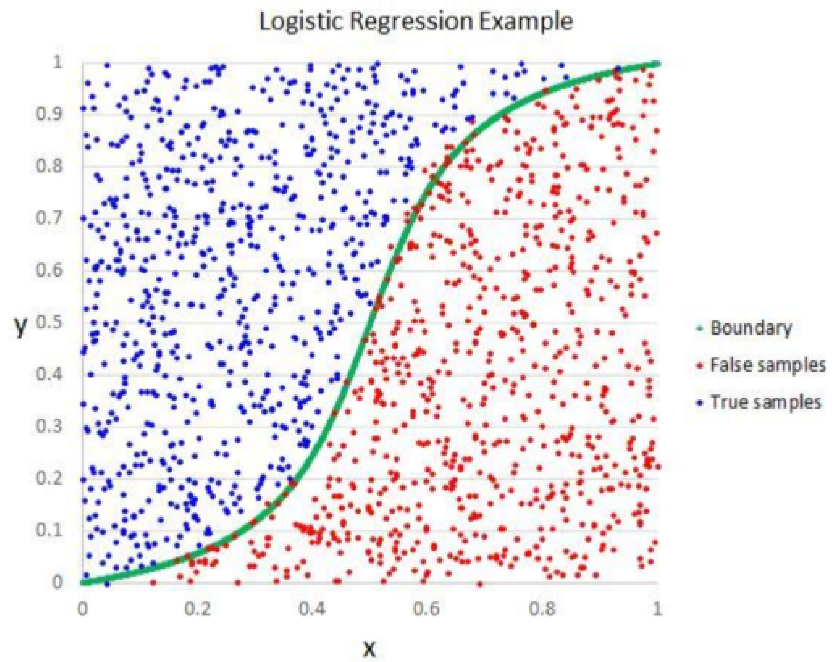
$$\Downarrow$$

$$A = (X^T X)^{-1} X^T y$$

Regression

- 로지스틱회귀 : 분류를 위한 모델

$$P(y = 1|X; A) = \frac{e^{XA}}{1 + e^{XA}}$$



Regression

- Multi-class logistic regression(softmax)

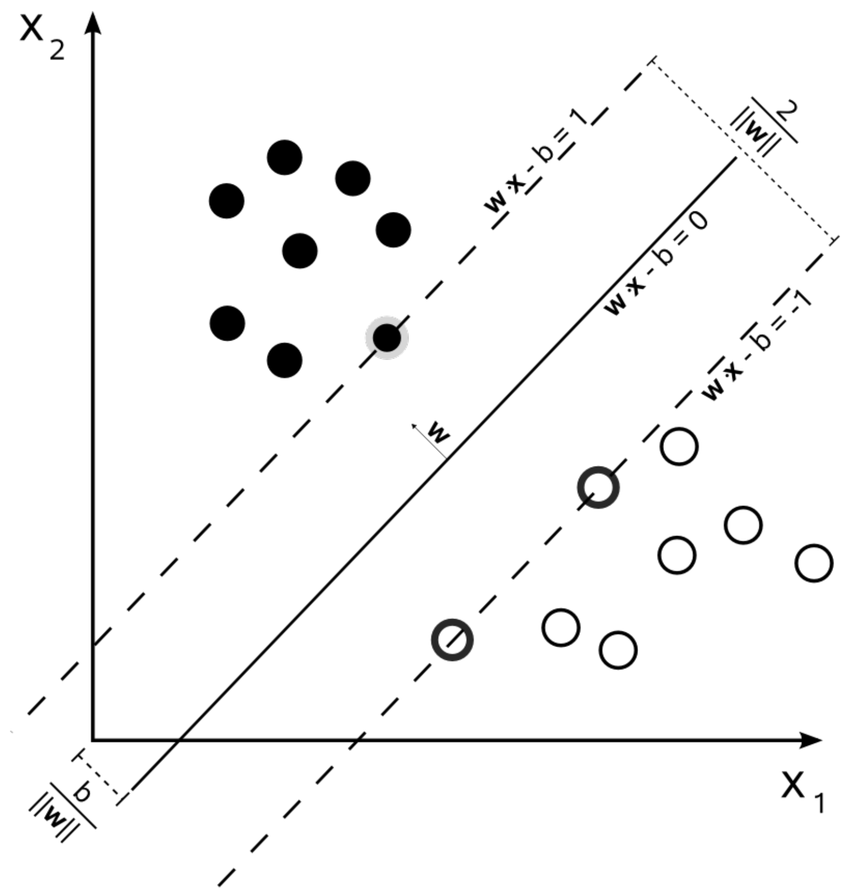
$$P(y = k|X; A_1, \ldots, A_k) = \frac{e^{XA_i}}{\sum_{i=0}^{k} e^{XA_i}}, A_0 = \vec{0}$$

- Binary class logistic regression objective

$$\log\left(\prod_{i=1}^{n} P(y_i|x_i; A)\right)$$
$$= \sum_{i=1}^{n} y_i log p_i(A) + (1 - y_i)\log(1 - p_i(A))$$

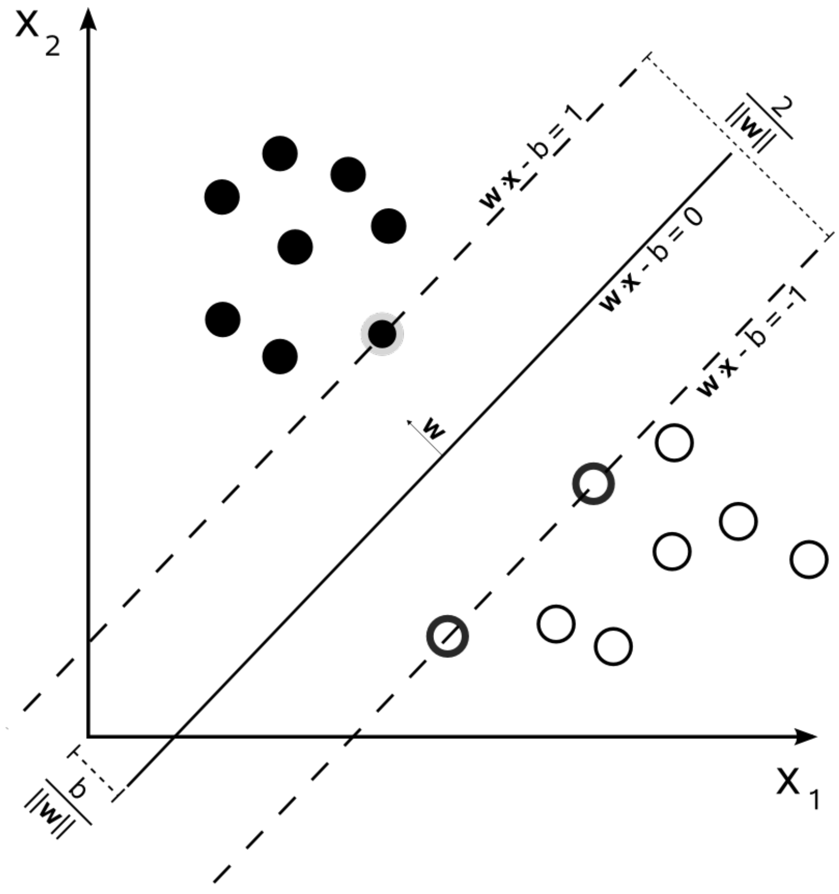
SVM

- Support Vector Machines



SVM

- Support Vector Machines

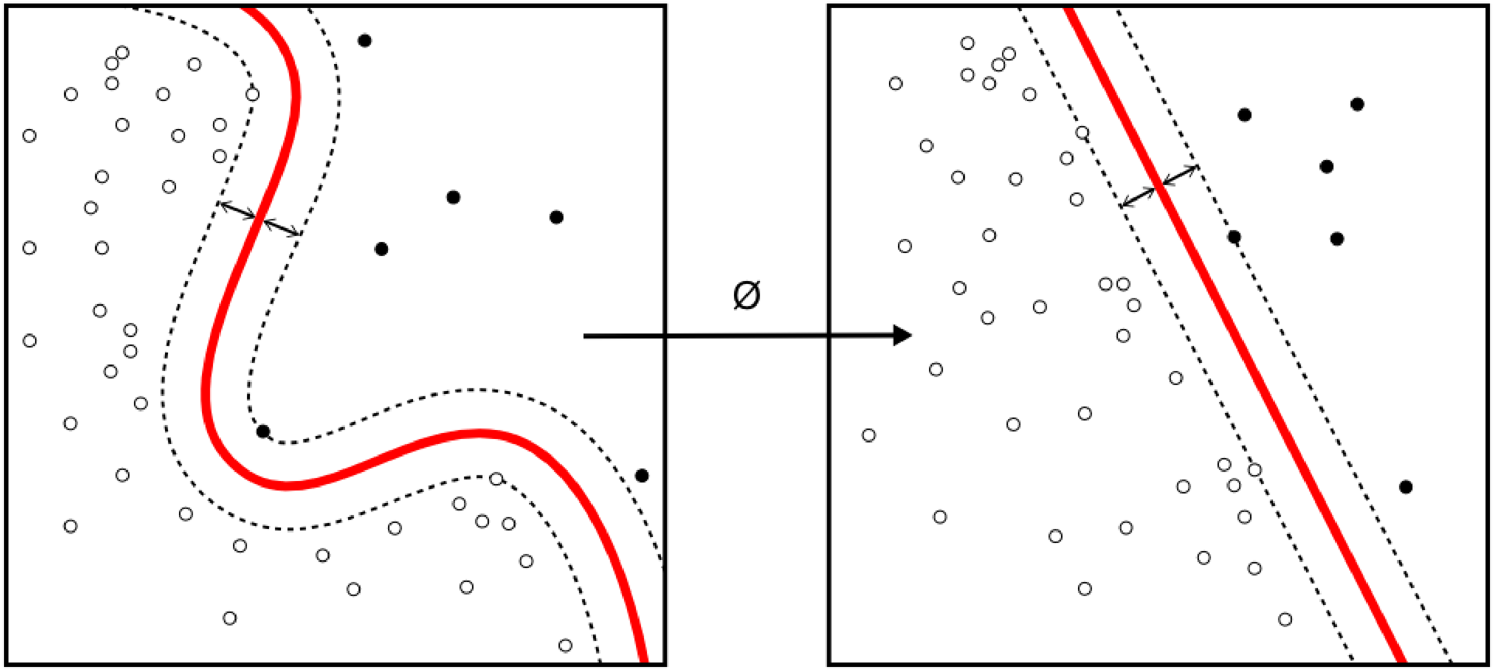


$$\max_{w,b,y_i(w*x_i+b)\geq 0} \rho$$

$$\rho = \min_{i\in[1,m]} \frac{|w * x_i + b|}{||w||}$$

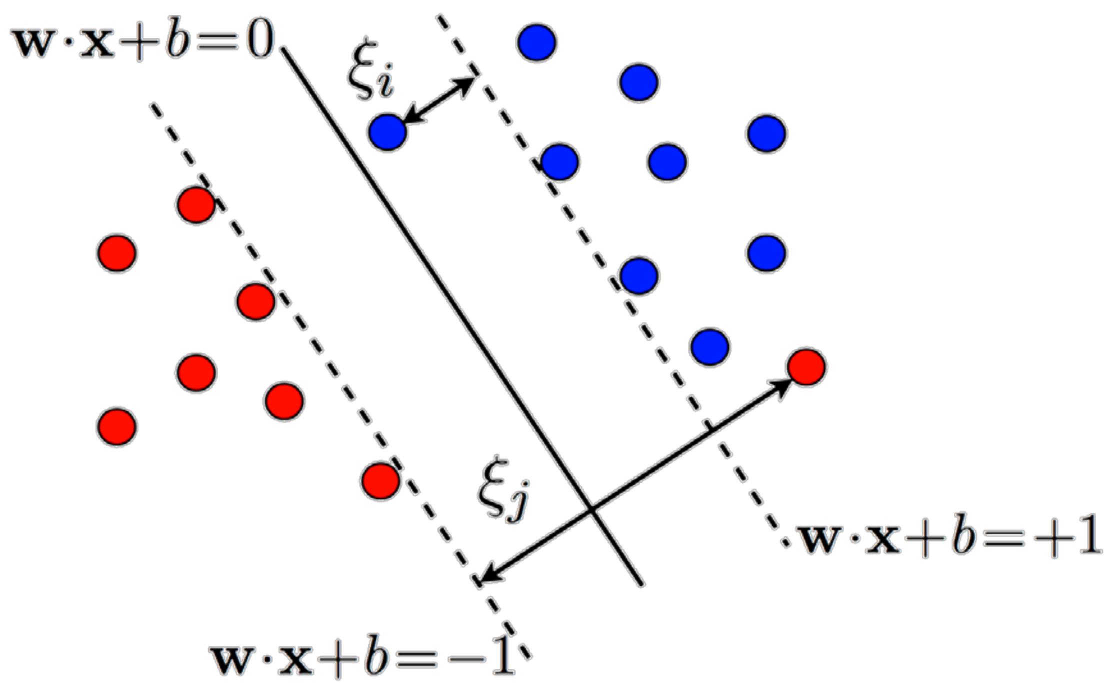
SVM

- Kernel trick



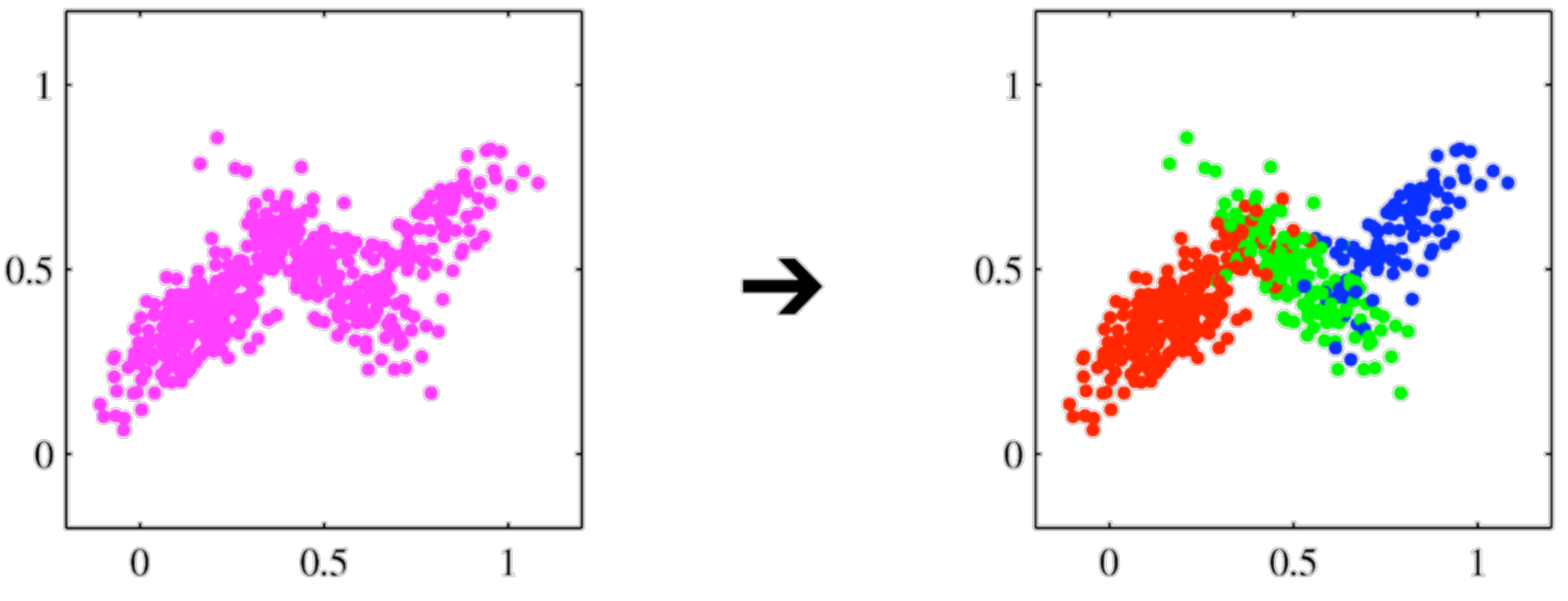
SVM

- Slack variable



Clustering

- 비슷한 데이터들을 묶는 방법



Clustering

- K-means clustering

K개의 그룹을 만들고, 각 점을 가장 가까운 그룹에 배정한다. 거리는 그룹의 평균 점과의 거리로 둔다.

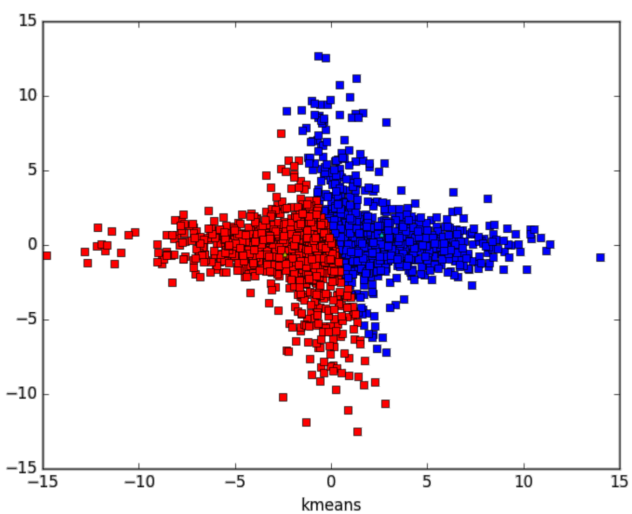
$$minimize_{r,u} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - u_k\|^2$$

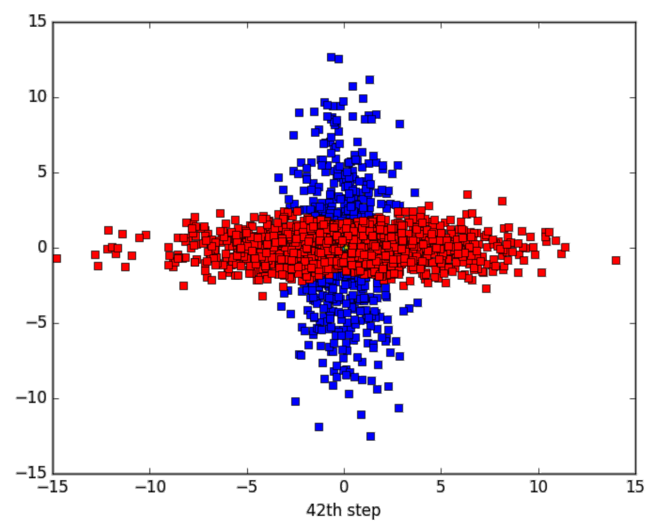
Clustering

- Gaussian Mixture Model

K개의 정규분포를 따르는 그룹으로 나눈다.

$$\ln p(X|p,u,s) = \sum_{n=1}^{N} ln\left\{\sum_{k=1}^{K} p_k N(x_n|u_k,s_k)\right\}$$

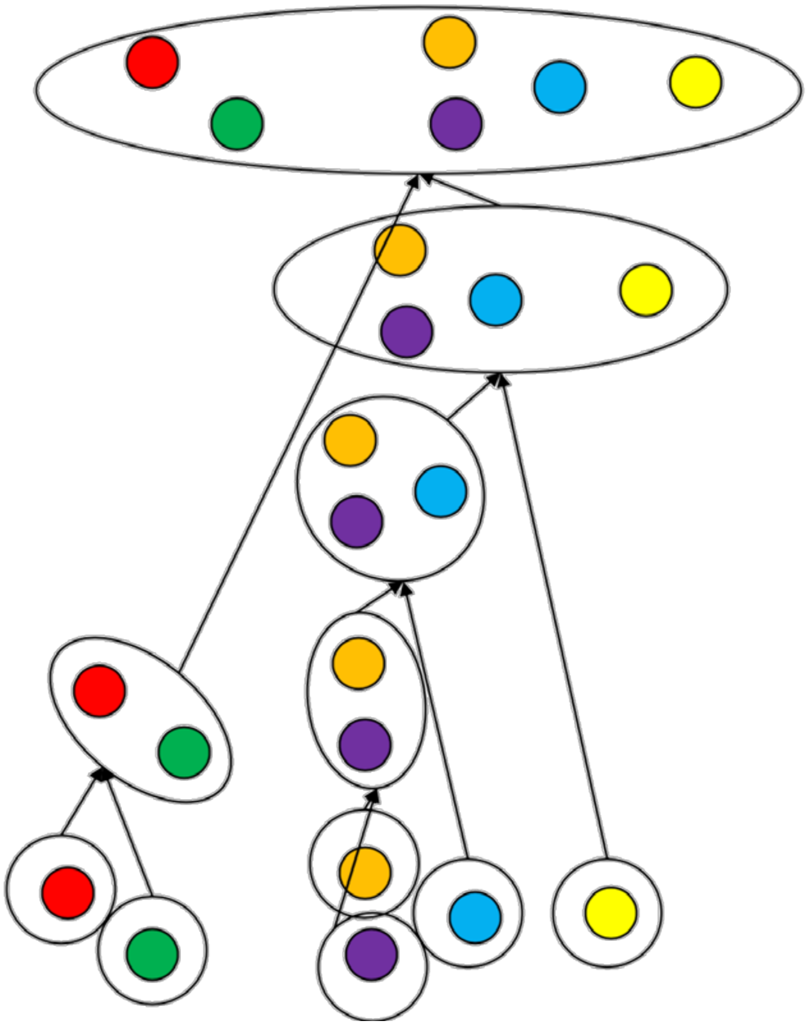




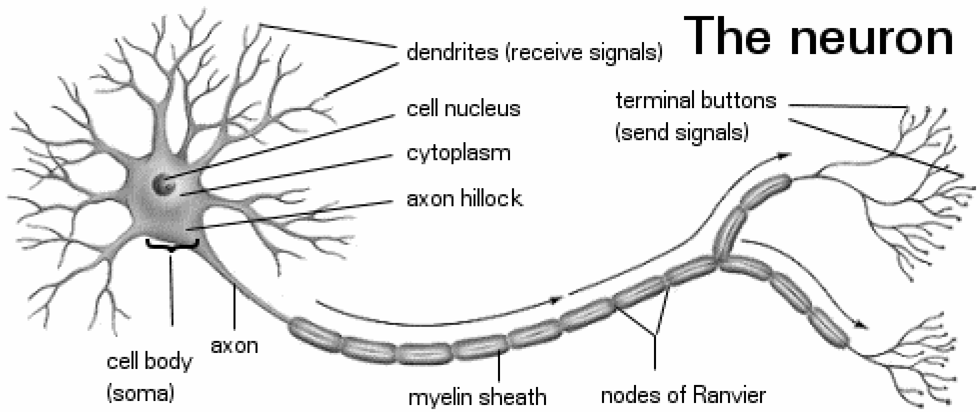
Clustering

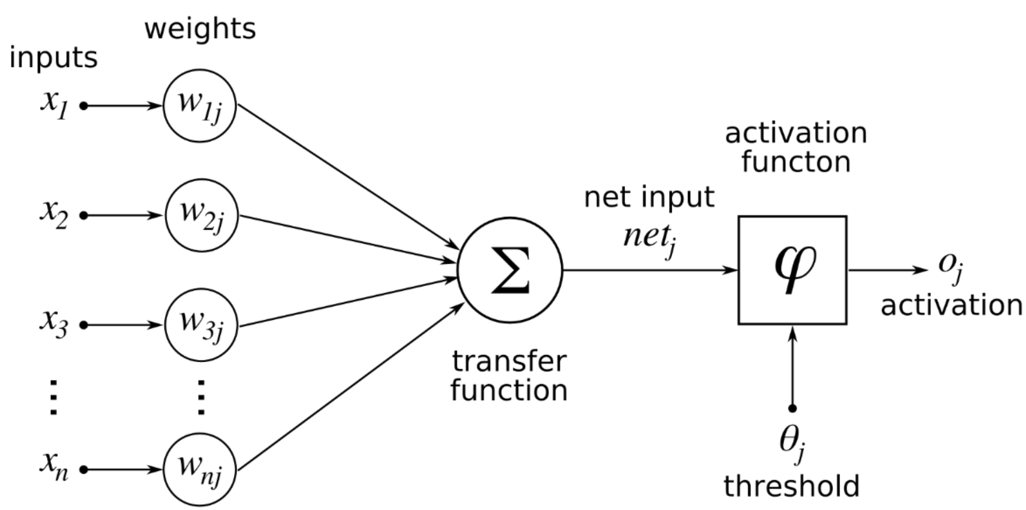
- Hierarchical Clustering

  가장 비슷한 그룹끼리
  묶어 나간다.



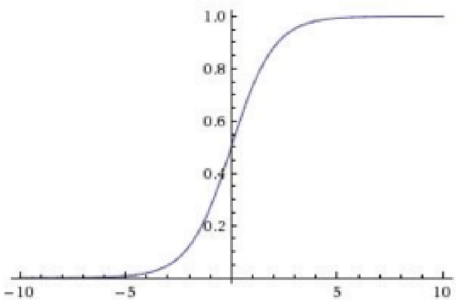
Neural Network - DNN



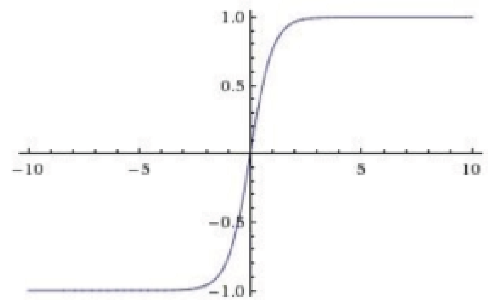


Neural Network - DNN

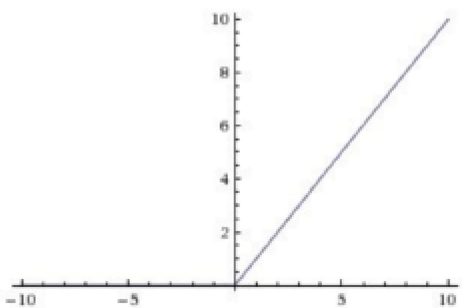
- Activation Fuctions



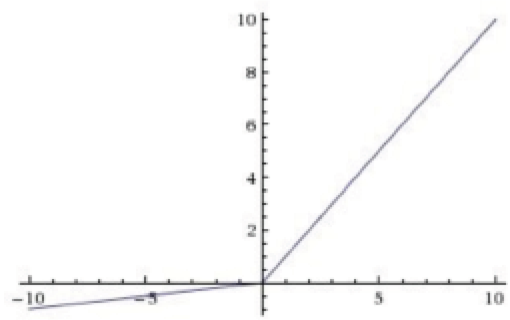
$$f(x) = \frac{1}{1 + e^{-x}}$$



$$f(x) = tanh(x)$$



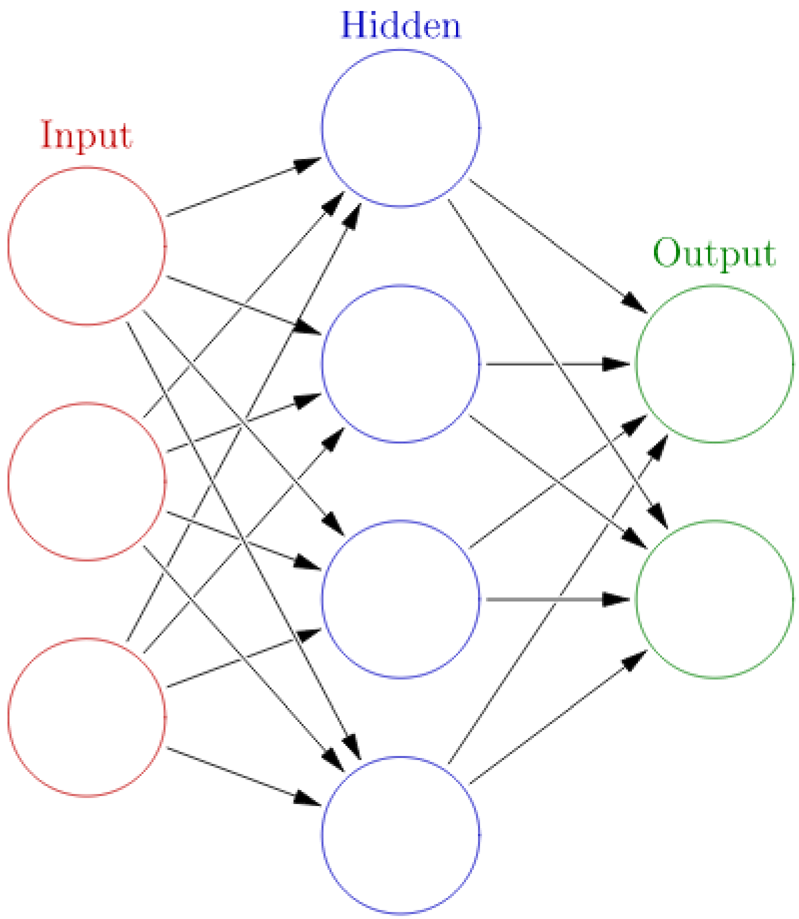
$$\max\{0, x\}$$



$$\max\{0.1x, x\}$$

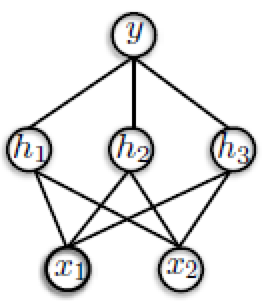
Neural Network - DNN

- 인공신경망의 구조

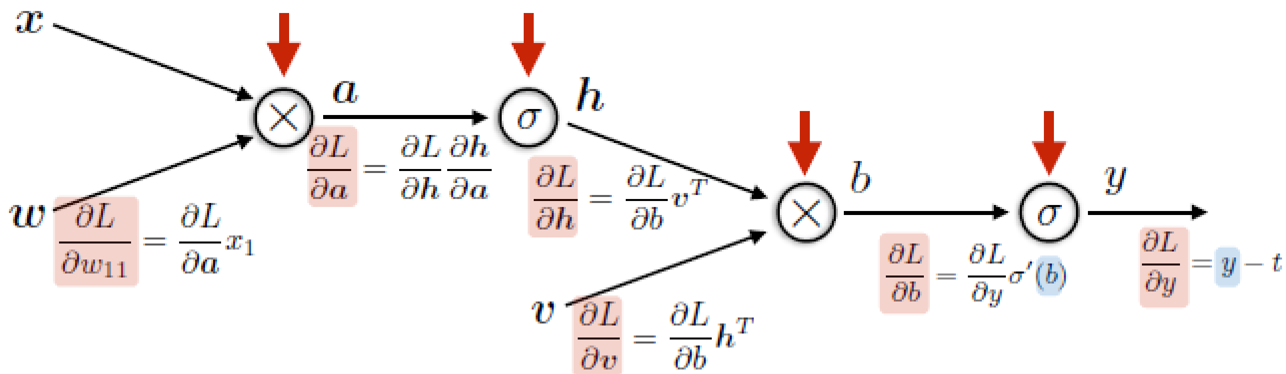


Neural Network - DNN

- 학습 : Backpropagation algorithm

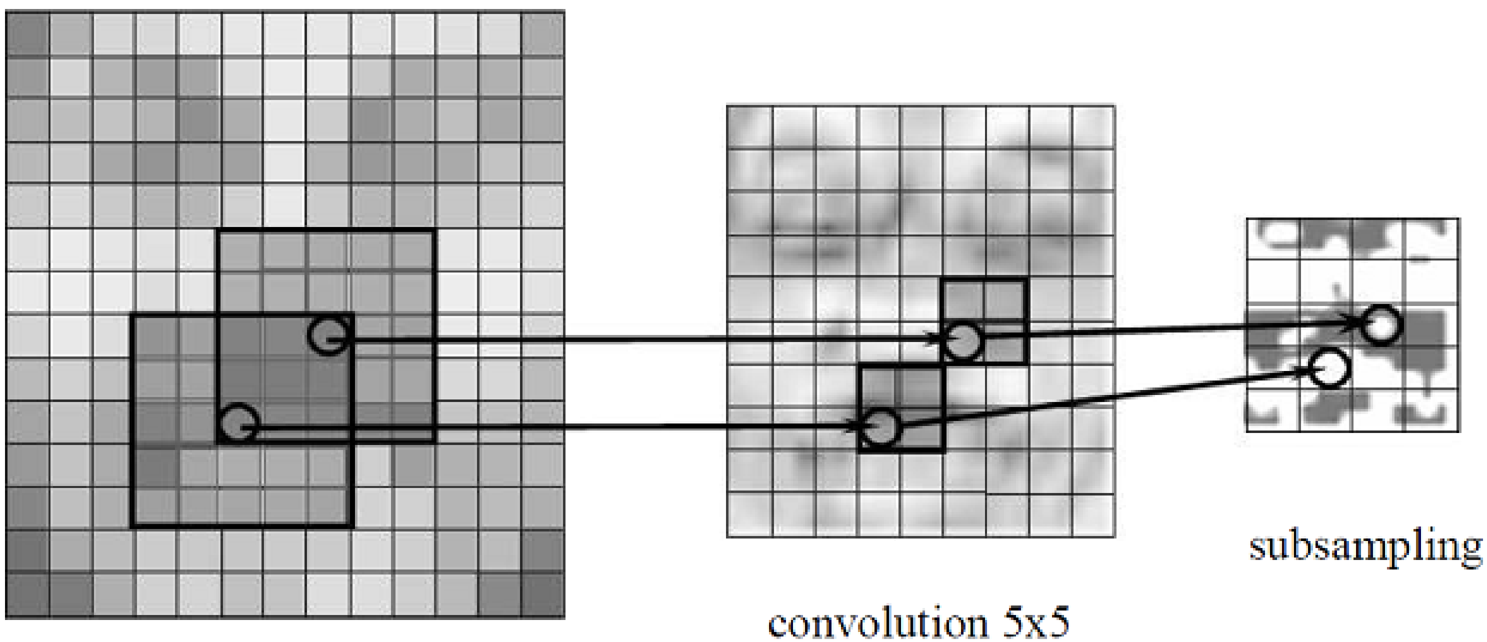


$$y = \sigma(b)$$
$$b = vh$$
$$h = \sigma(a)$$
$$a = wx$$



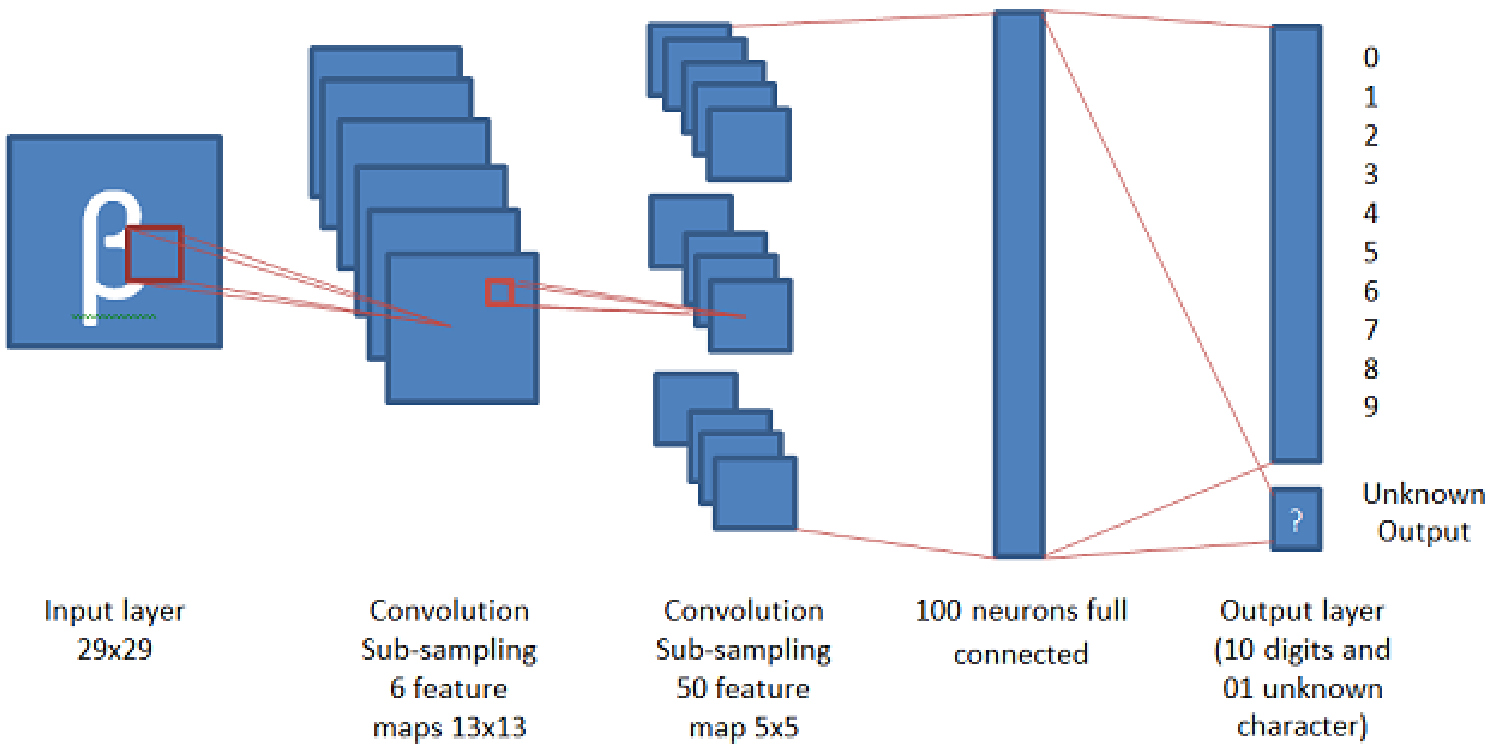
Neural Network - CNN

- Convolutional Neural Network



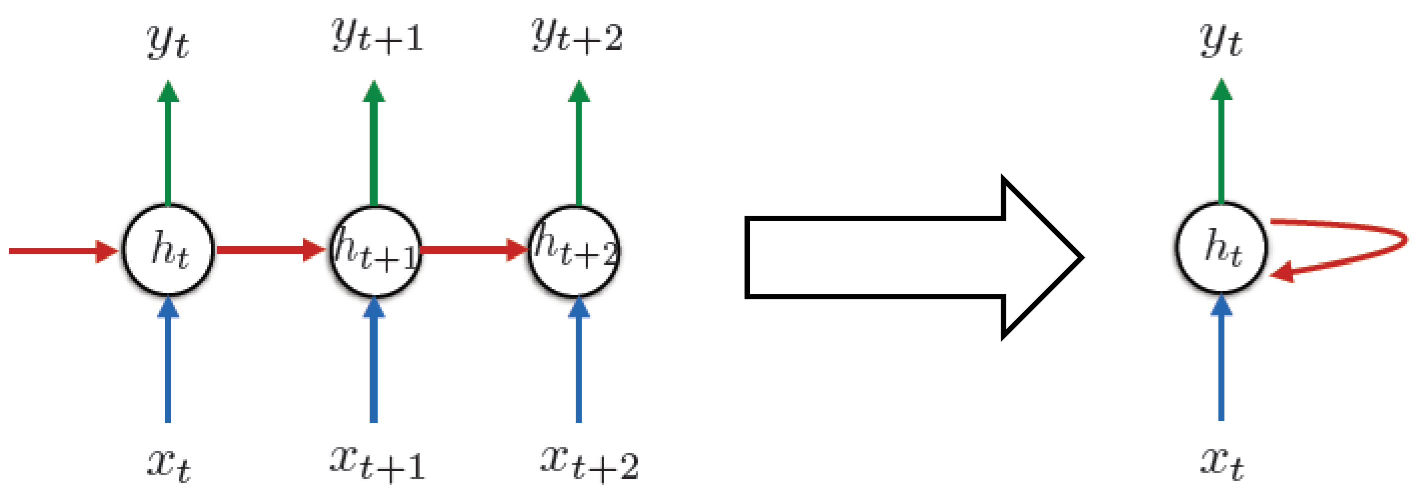
Neural Network - CNN

- Convolutional Neural Network



Neural Network - RNN

- Recurrent Neural Network
- 시간에 따라 바뀌는 데이터에 관한 NN



Neural Network - RNN

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

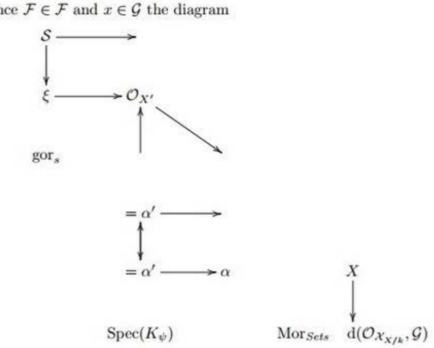
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

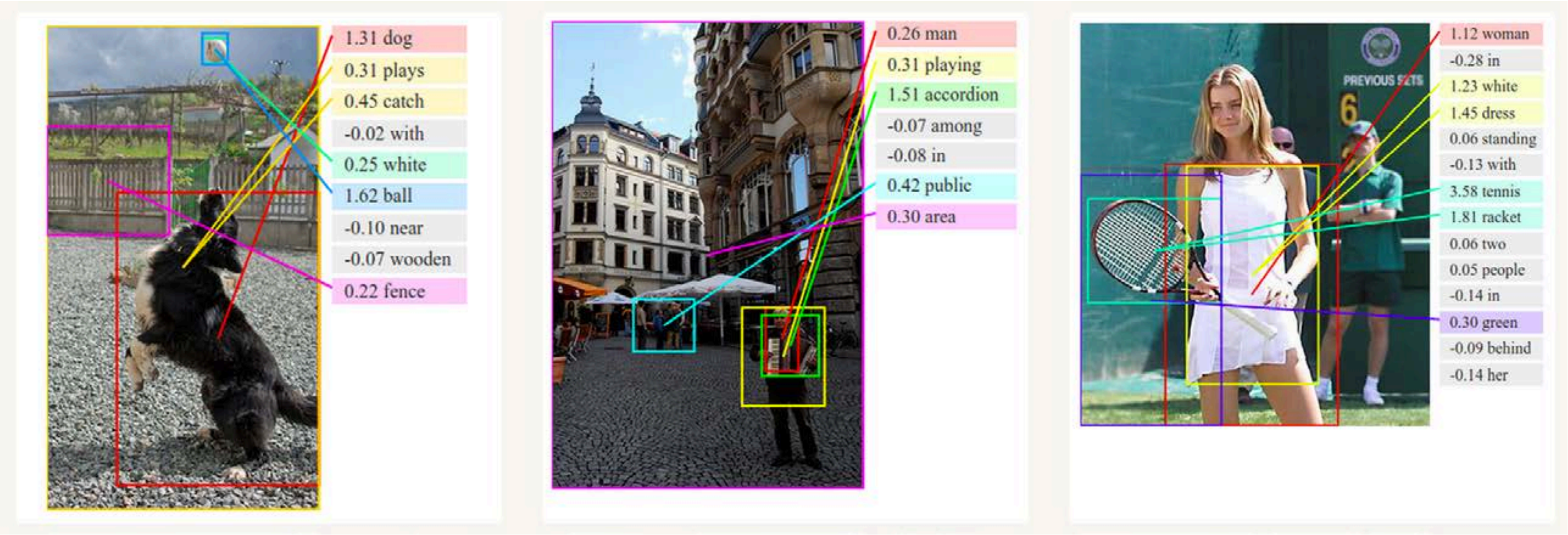
Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Neural Network - RNN



감사합니다