



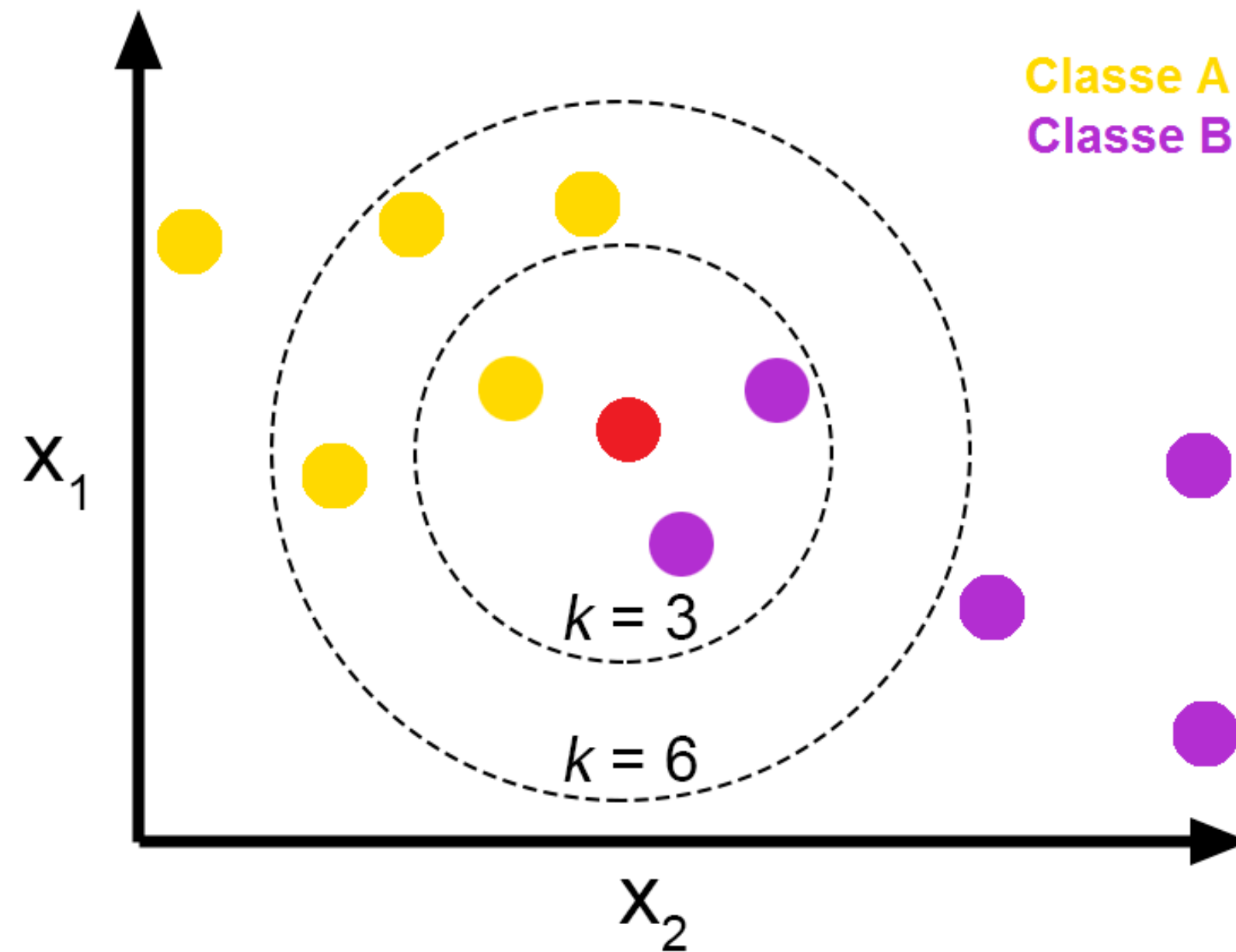
ML/DS 지식 따라가기

KNN / SVM / RNN / LSTM



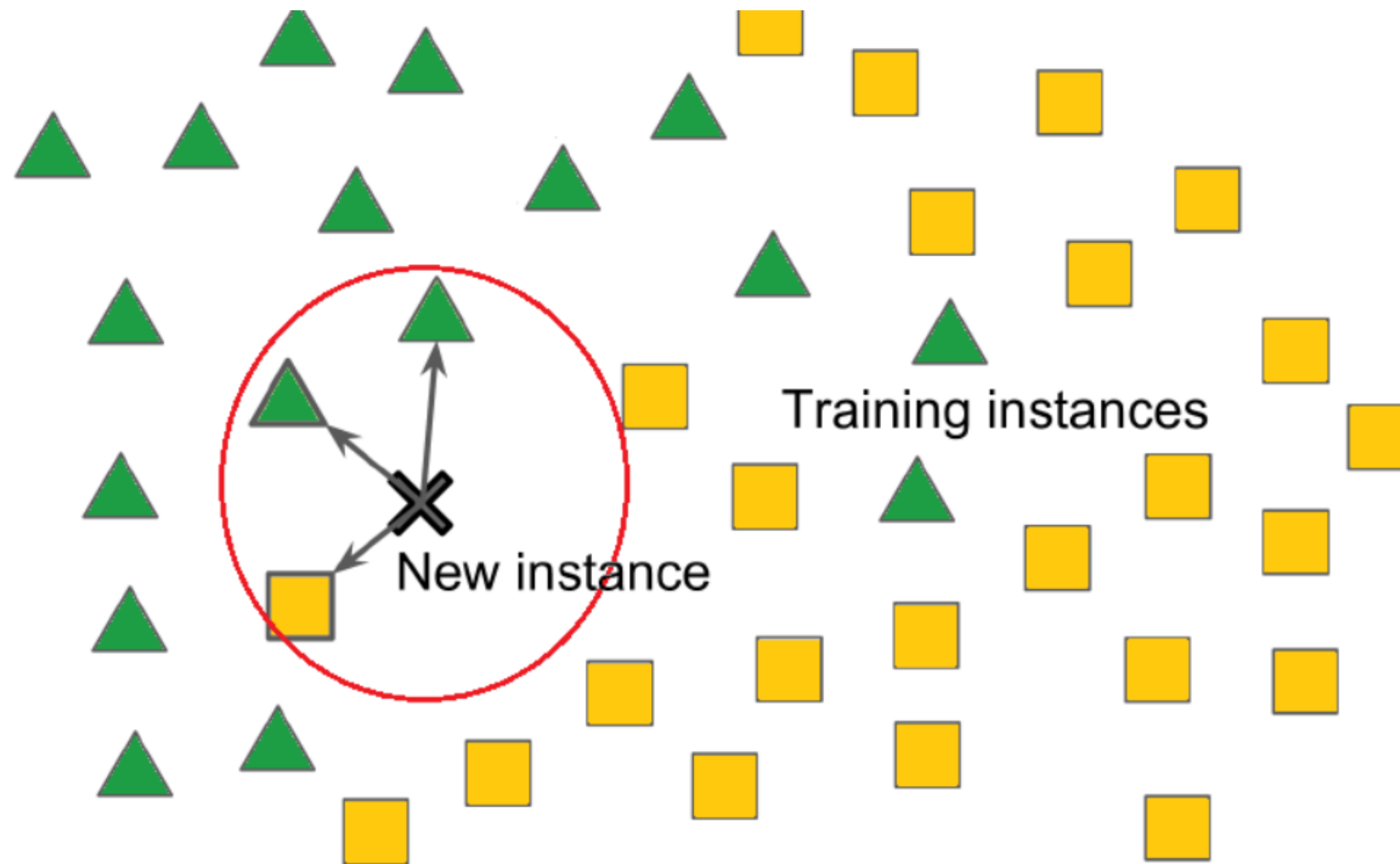
K-NN 이란?

최근접 이웃 알고리즘. 주변의 데이터를 살펴본 뒤 더 많은 데이터가 포함되어 있는 범주로 분류함.



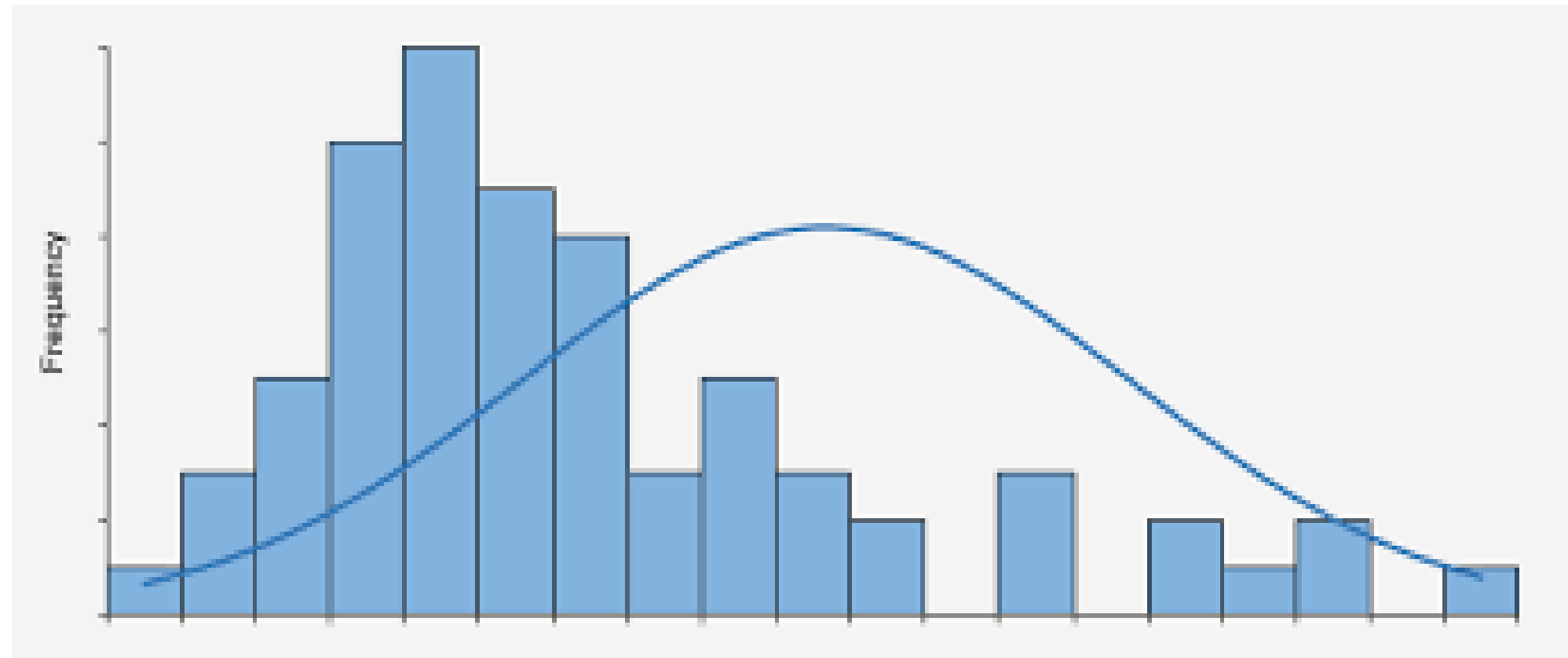
이웃

근처에 있는 데이터



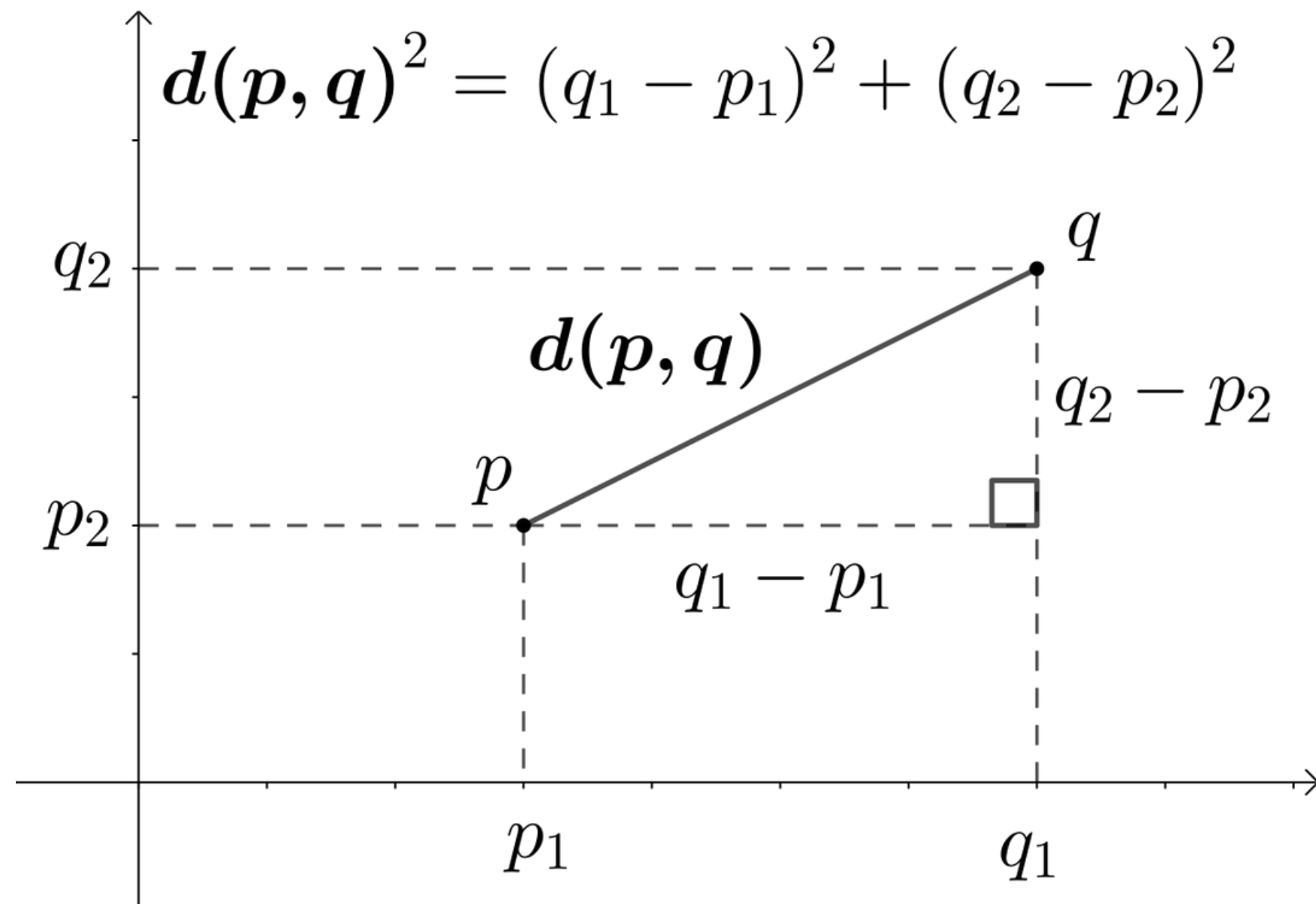
비모수 통계

모집단의 형태에 관계없이 주어진 데이터에서 직접 확률을 계산하여 통계학적 검정을 하는 분석법



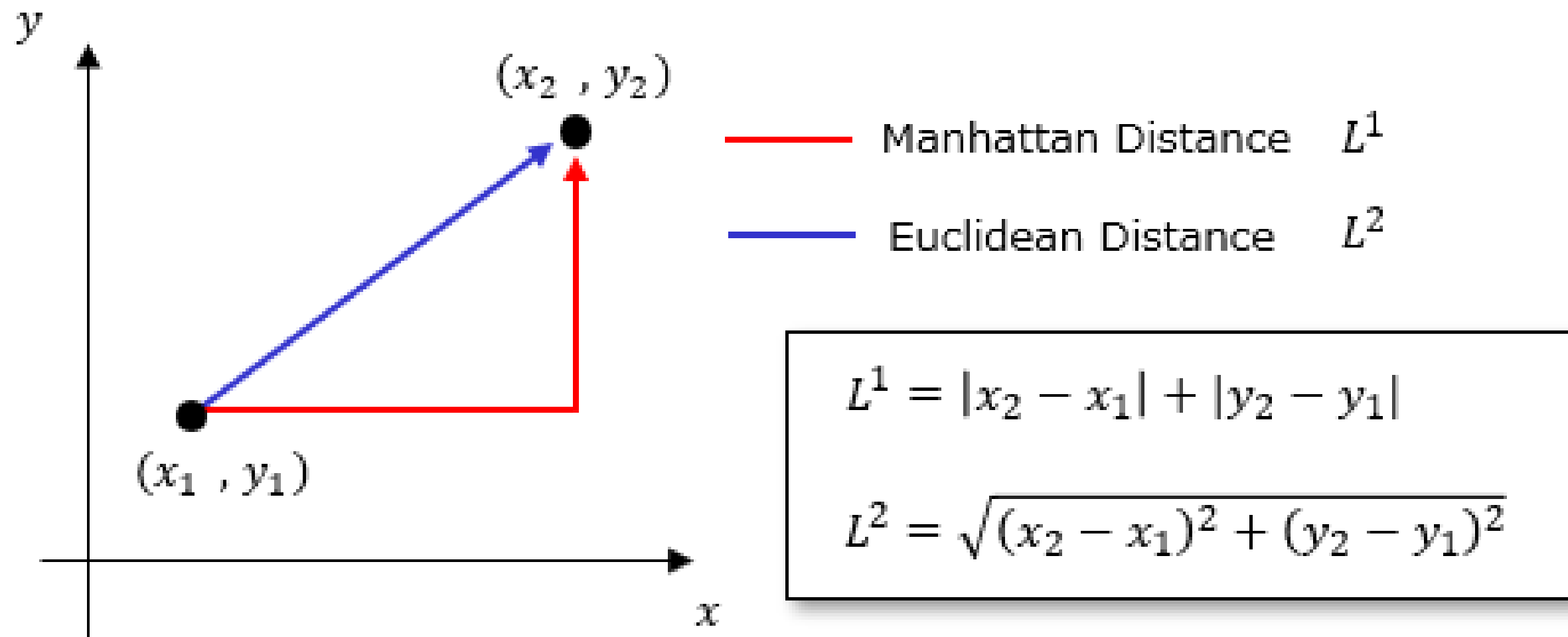
거리 측정 : EUCLIDEAN DISTANCE

두 점 간의 기하학적 거리를 측정함



거리 측정 : MANHATTAN DISTANCE

두 관찰치 사이의 거리로 각 feature간 차이의 절대값의 합



거리 측정 : MAHALANOBIS DISTANCE

평균과의 거리가 표준편차의 몇 배인지를 나타내서 이상치의 영향력을 측정함

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}.$$

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}.$$

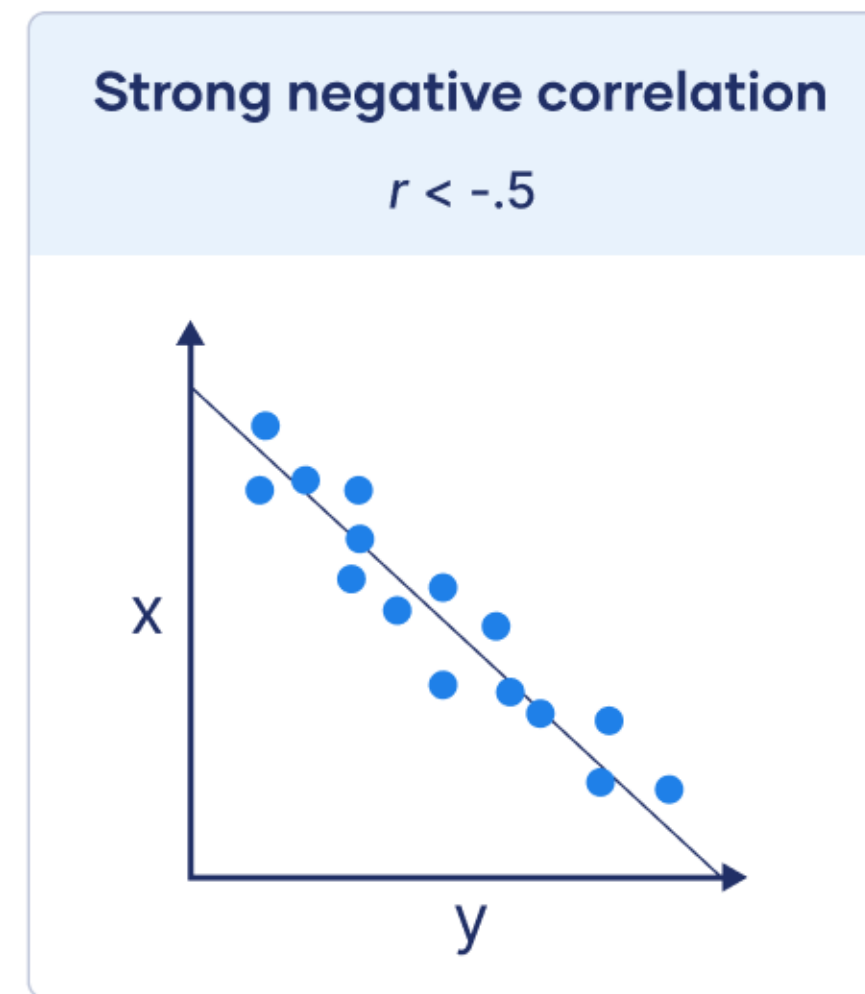
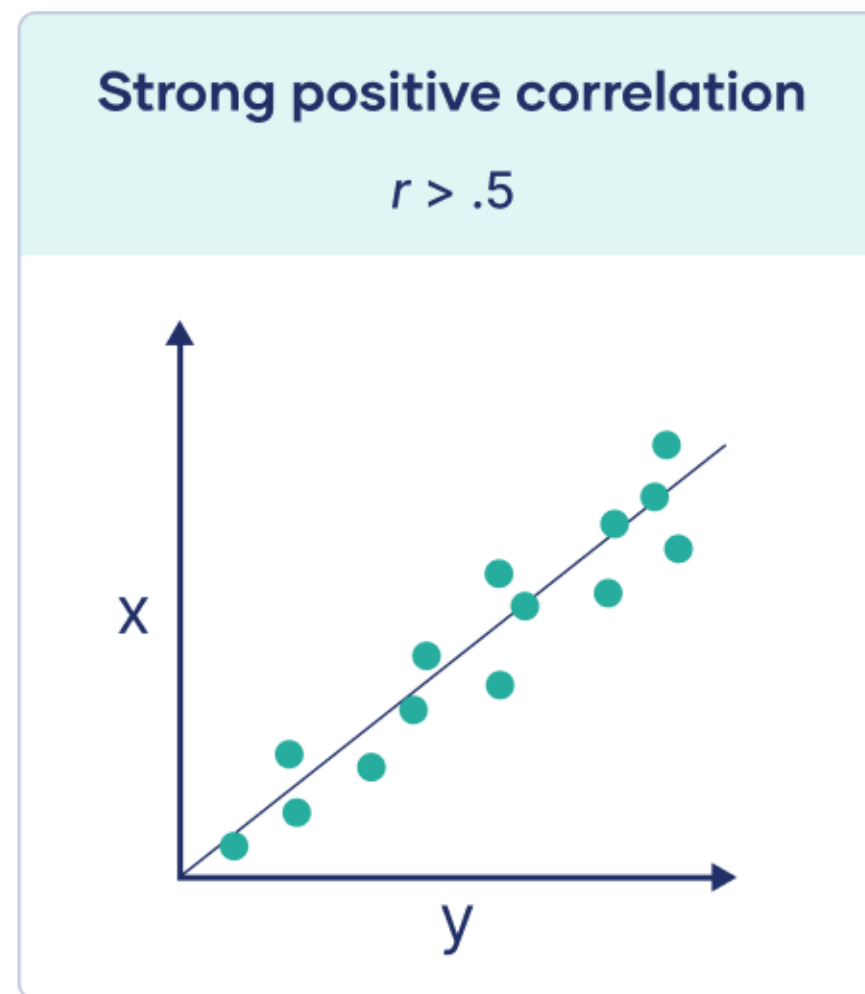
거리 측정 : COSINE SIMILARITY

두 벡터간 각도의 코사인값을 이용하여 측정된 벡터간의 유사한 정도

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

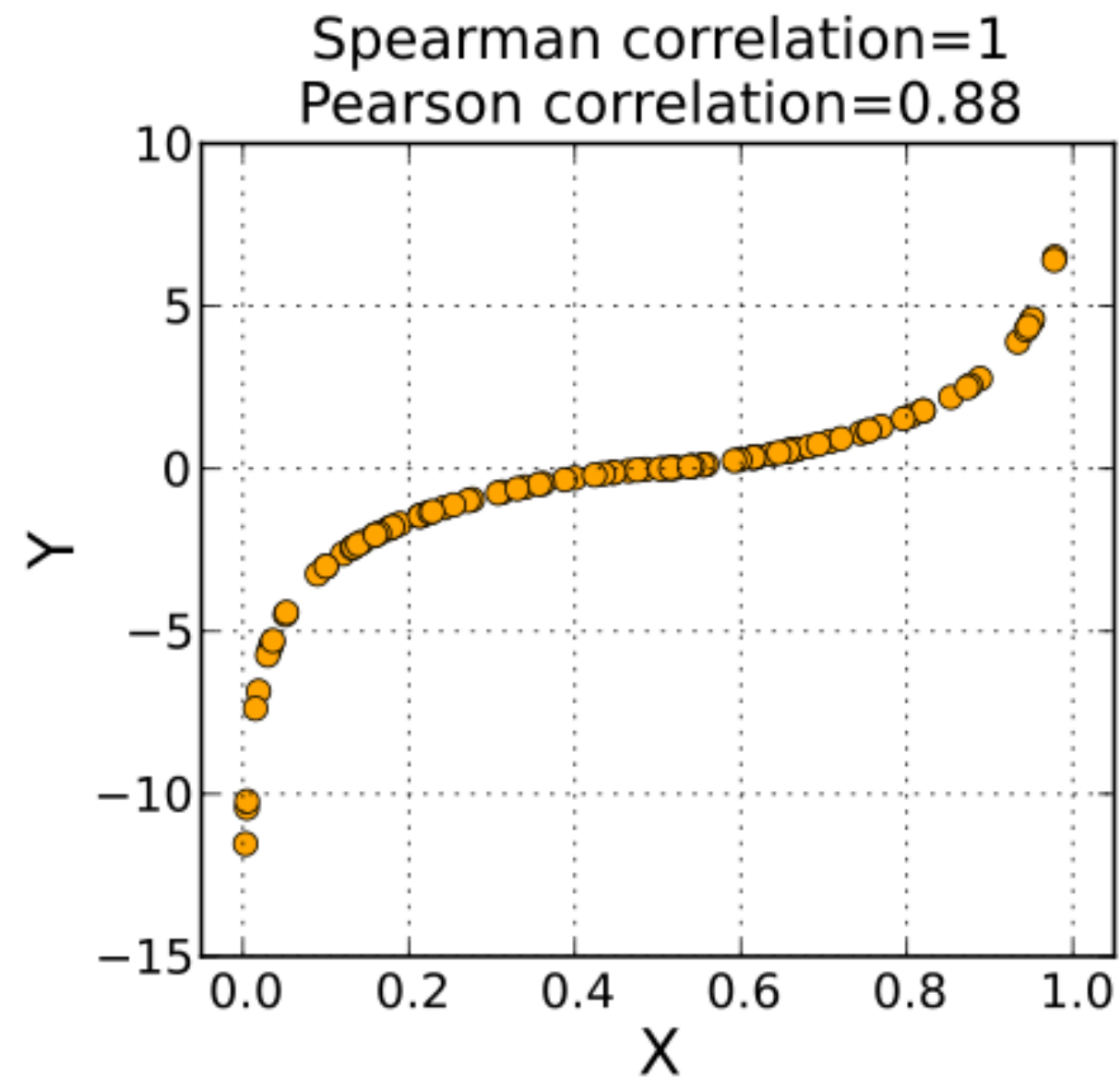
거리 측정 : CORRELATION DISTANCE

Pearson Correlation을 통해 데이터 패턴의 유사도를 측정함



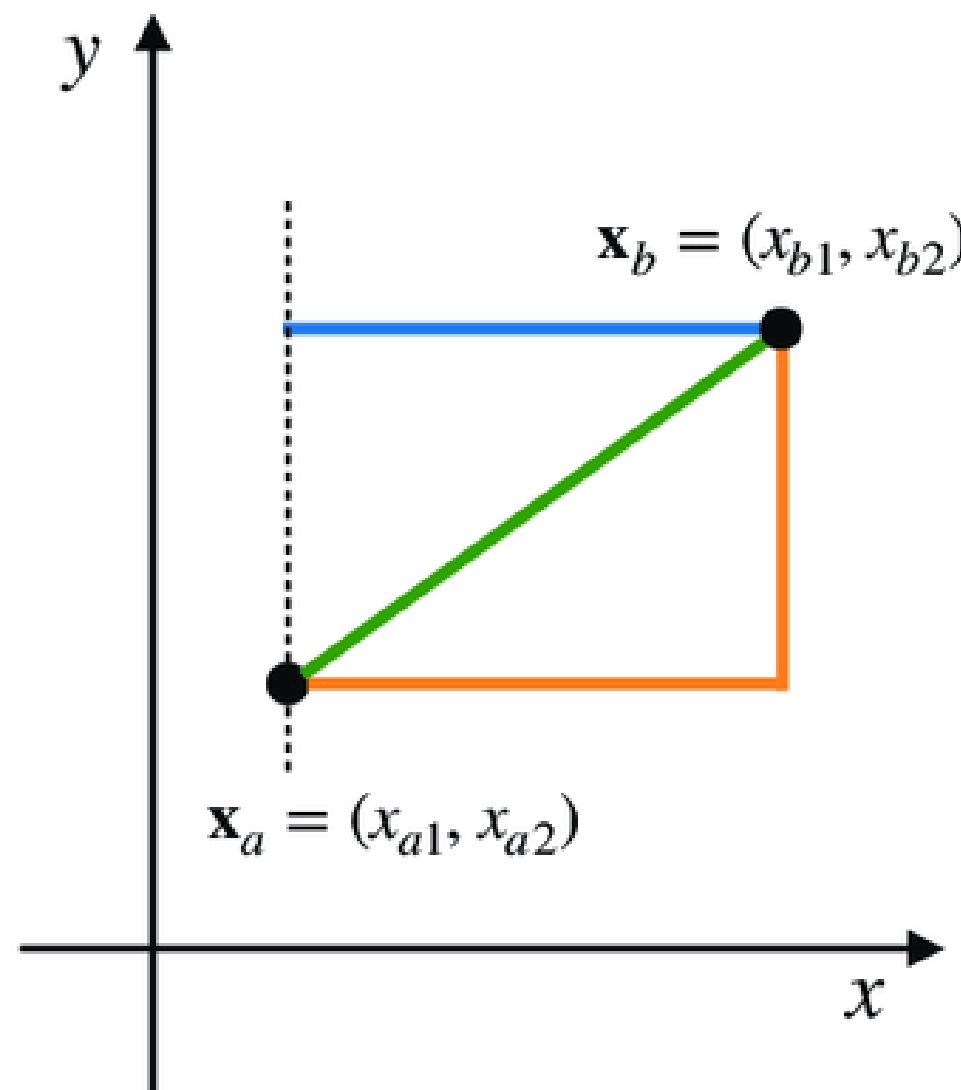
거리 측정 : SPEARMAN CORRELATION DISTANCE

두 변수의 순위 사이의 통계적 의존성을 측정함



거리 측정 : CHEBYSHEV DISTANCE

feature 간 최대 거리 차의 절대값을 통해 측정함



$p = 2$ Euclidean distance

$$\|\mathbf{x}_a - \mathbf{x}_b\|_2 = (|x_{a1} - x_{b1}|^2 + |x_{a2} - x_{b2}|^2)^{\frac{1}{2}}$$

$p = 1$ Manhattan distance

$$\|\mathbf{x}_a - \mathbf{x}_b\|_M = |x_{a1} - x_{b1}| + |x_{a2} - x_{b2}|$$

$p = \infty$ Chebyshev distance

$$\|\mathbf{x}_a - \mathbf{x}_b\|_\infty = \max\{|x_{a1} - x_{b1}|, |x_{a2} - x_{b2}|\}$$

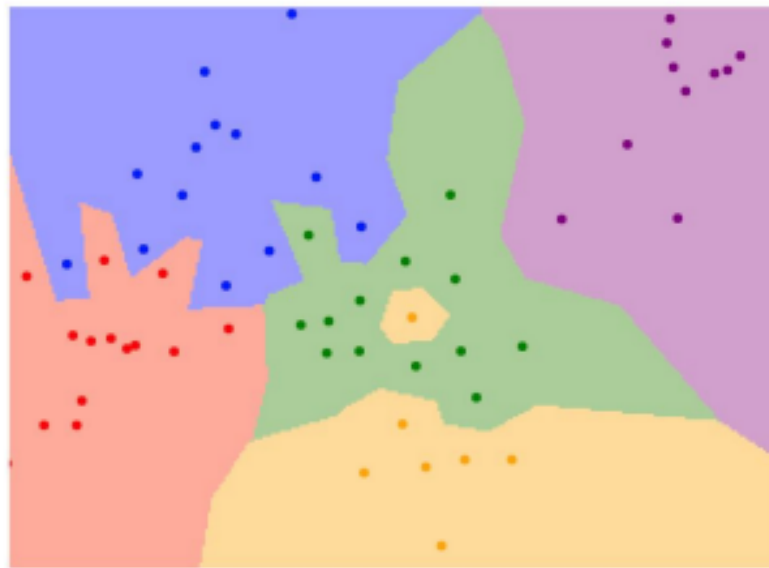
거리 측정 : MINKOWSKI DISTANCE

맨하탄 거리와 유클리드 거리 등을 일반화한 방식

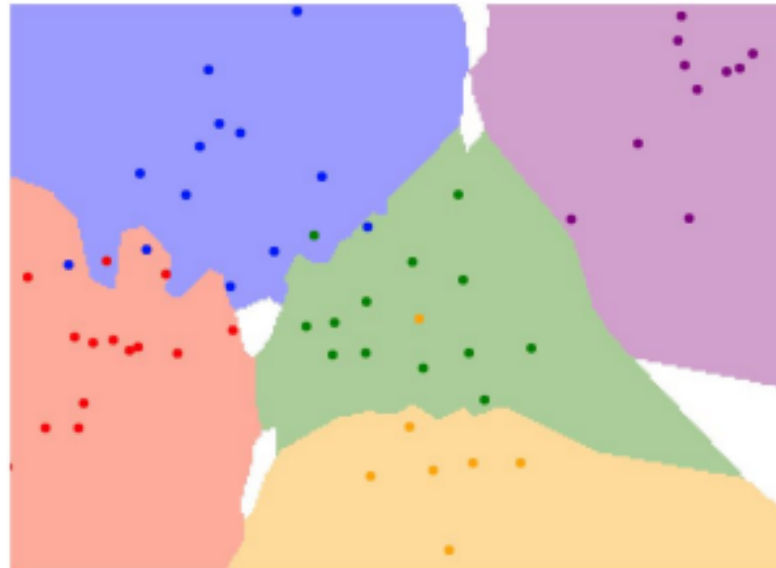
$$\left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

K값

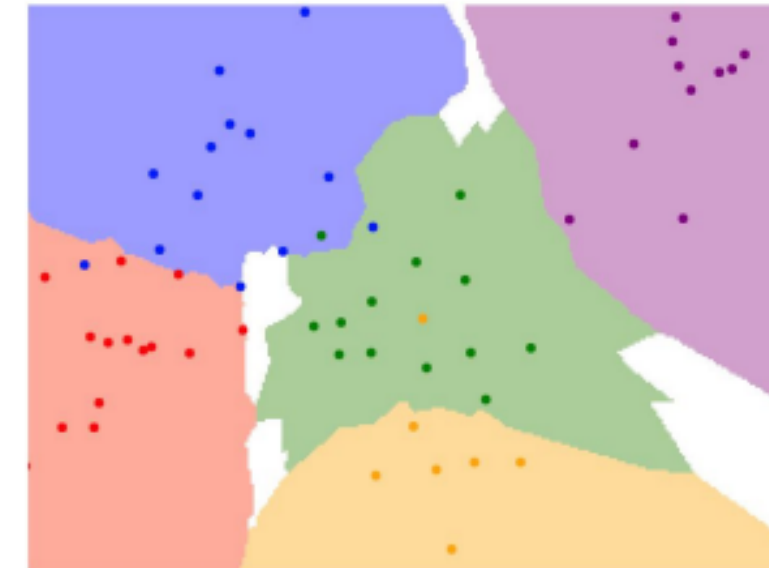
K값을 몇으로 잡냐에 따라 결과가 달라짐



K = 1



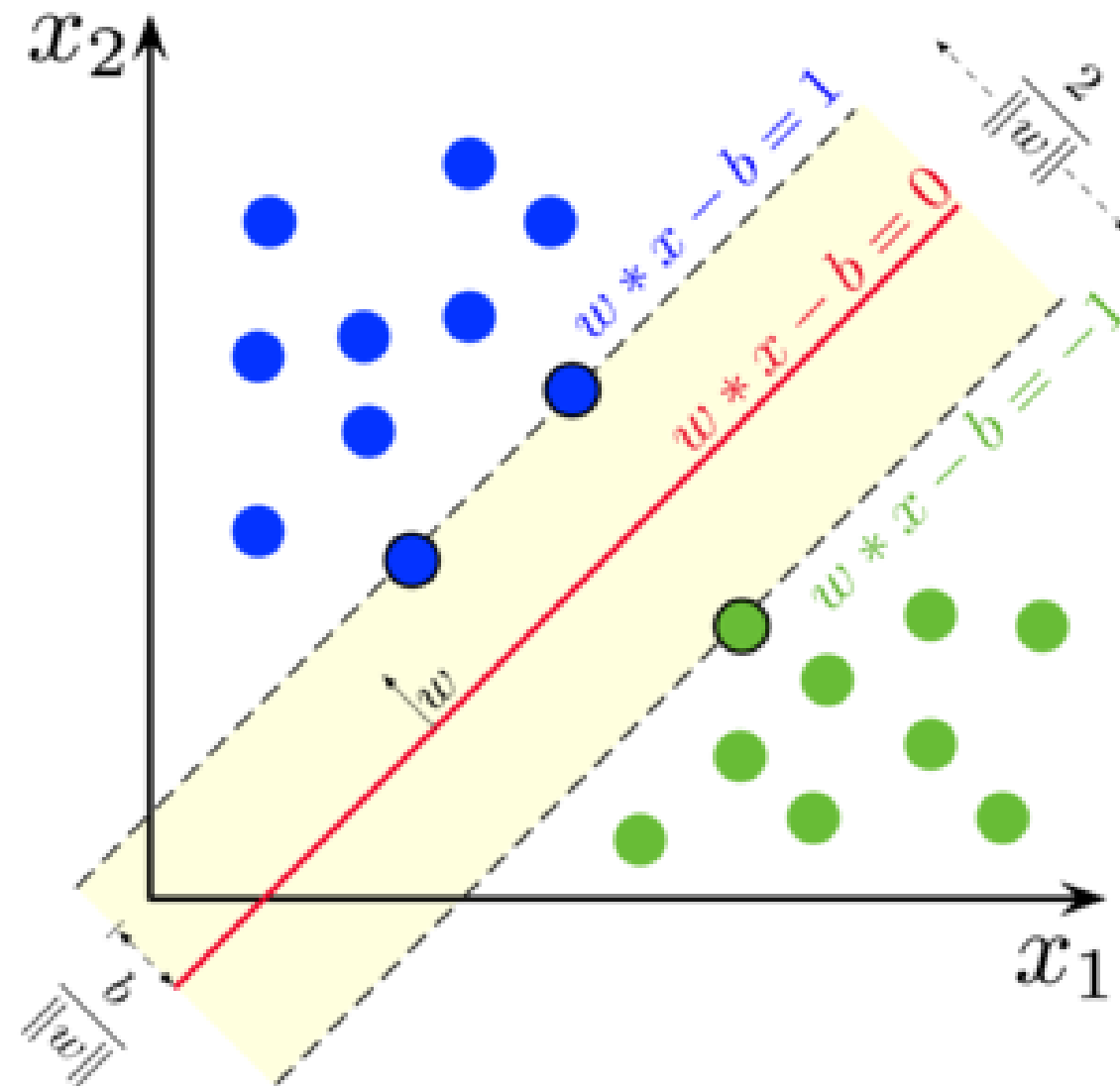
K = 3



K = 5

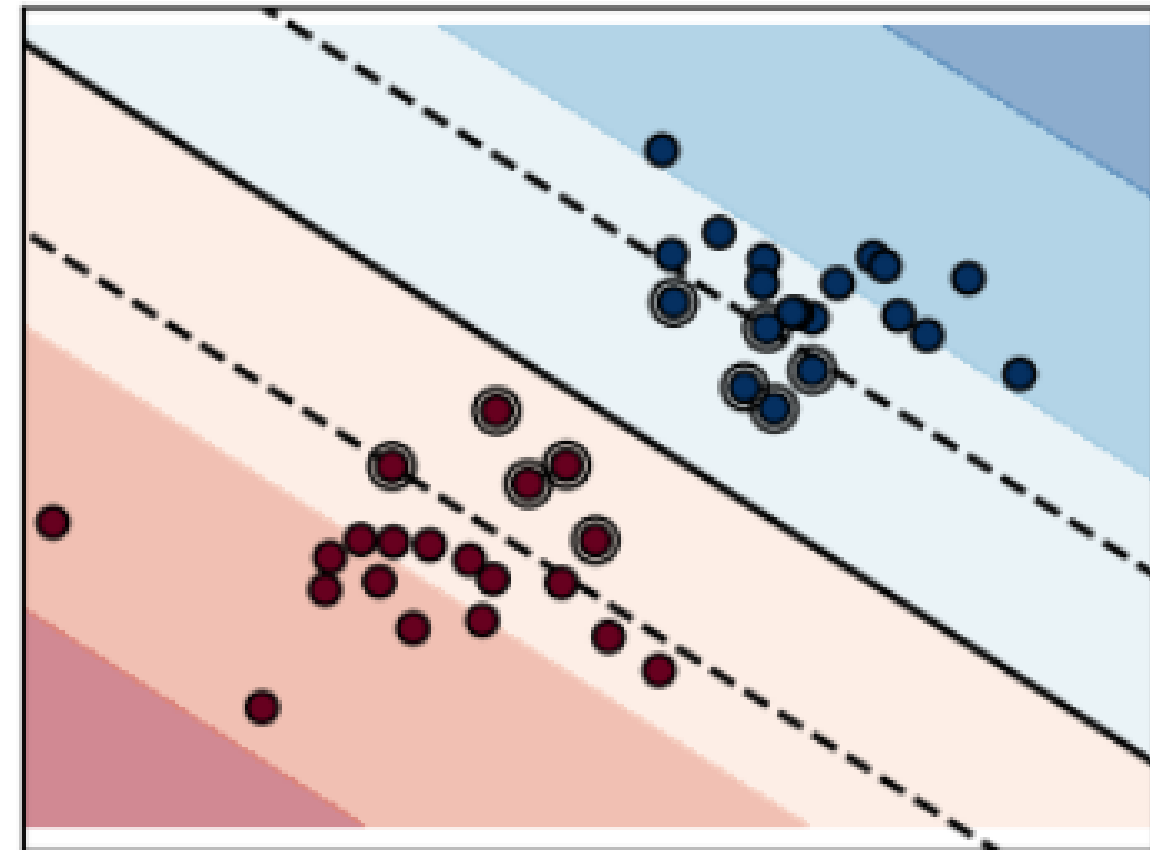
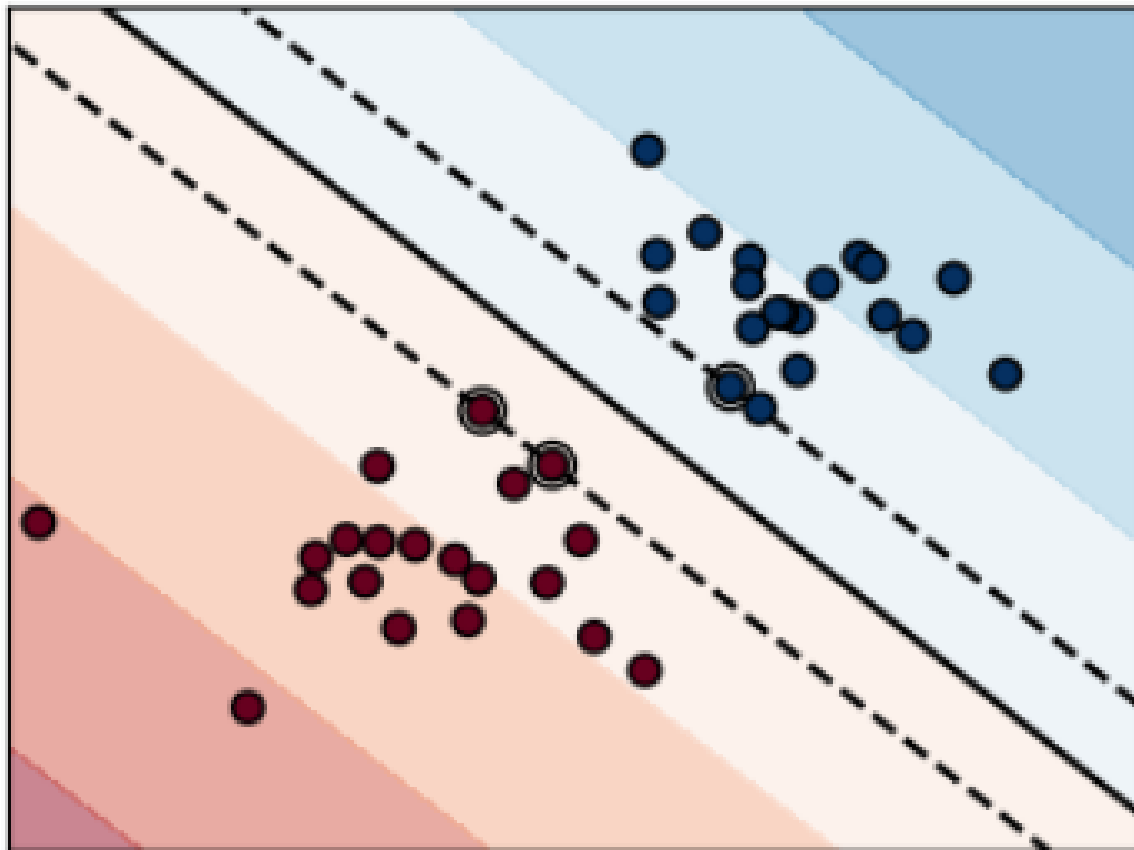
SVM 이란?

Soft Vector Machine. 최적의 분류 경계를 탐색하고, 그 기준으로 탐색함.



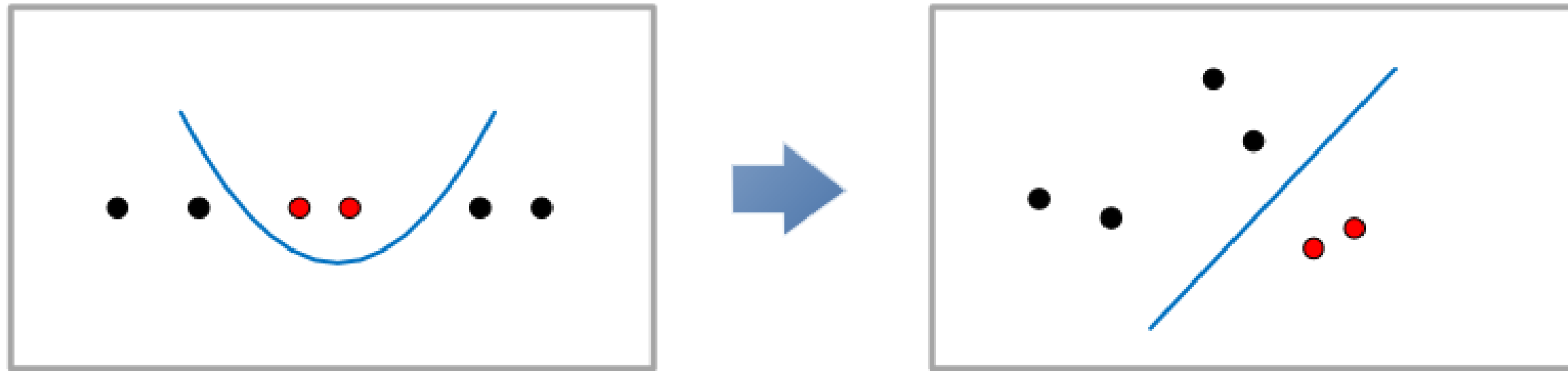
마진이란?

데이터 클래스를 구분하는 도로의 경계



커널 트릭이란?

non-linear한 데이터에 대해서 차원을 높여서 구분 가능하게 만드는 방법



RNN

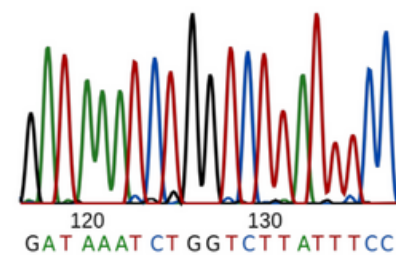
Recurrent란

- RNN에서 Recurrent는 repetitive의 의미를 가짐
 - Input 데이터는 sequential(순차적)
 - parameter는 순차적임

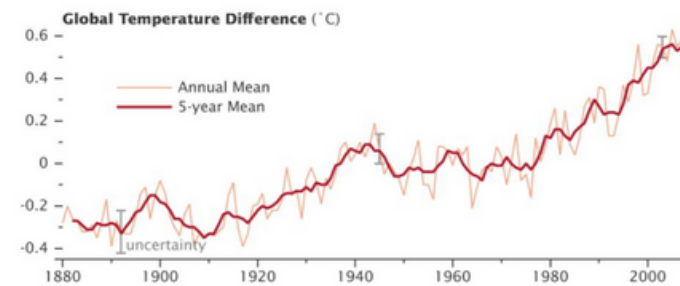
RNN

Sequential Data

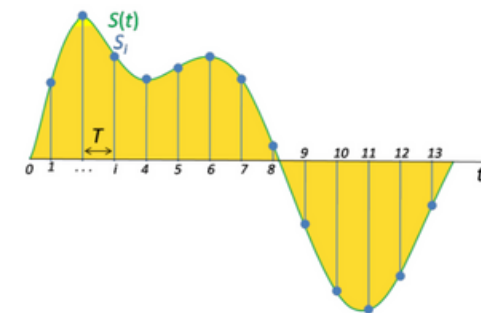
- data point : 각 시가에 따른 vector x 의 sequence
- batch data : 서로 다른 길이를 가진 sequence
- label : scalar, vector, 혹은 sequence
- 서로 다른 종류의 sequence들도 label이 될 수 있음



DNA 염기 서열
(Sequential Data)



세계 기온 변화
(Temporal Sequence)



샘플링된 소리 신호
(Time Series)

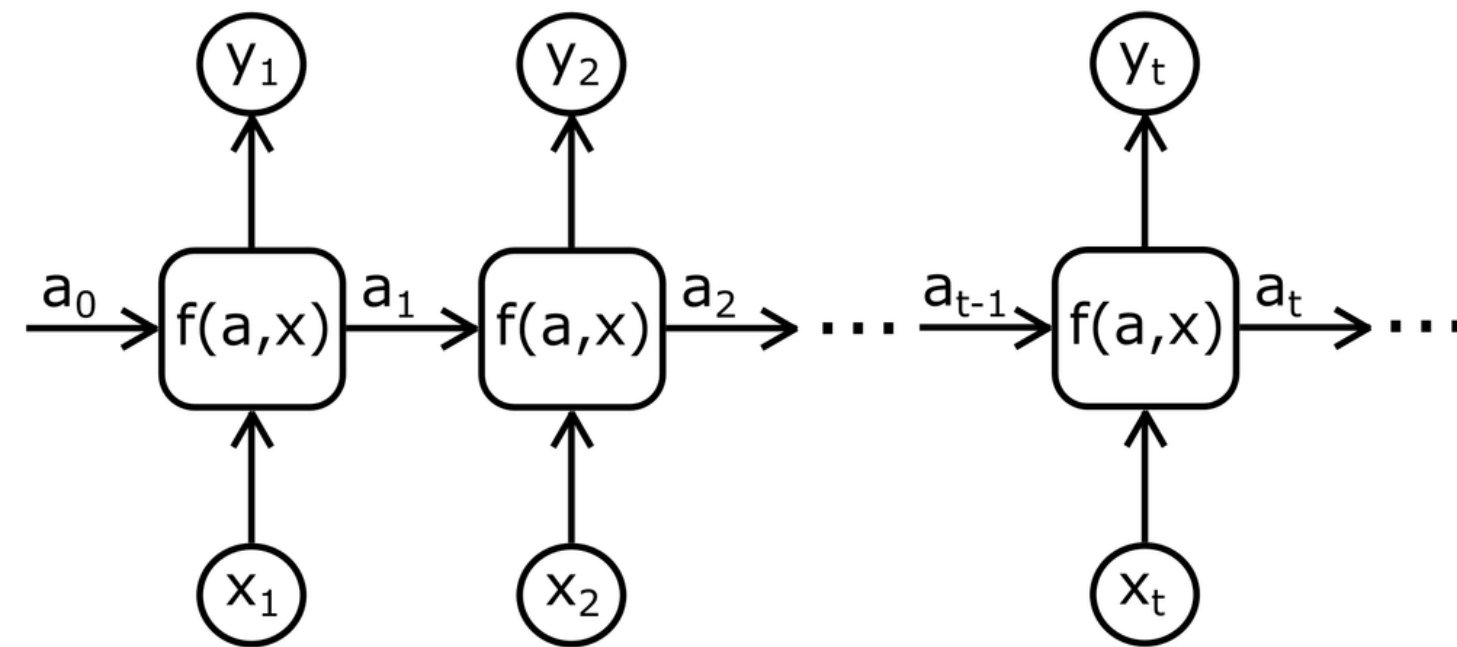
순서가 의미가 있으며, 순서가 달라질 경우 의미가 손상되는 데이터를 **순차 데이터**라고 한다.

시간적 의미가 있는 경우 **Temporal Sequence**라고 하며, 일정한 시간차라면 **Time Series**라고 한다.

RNN

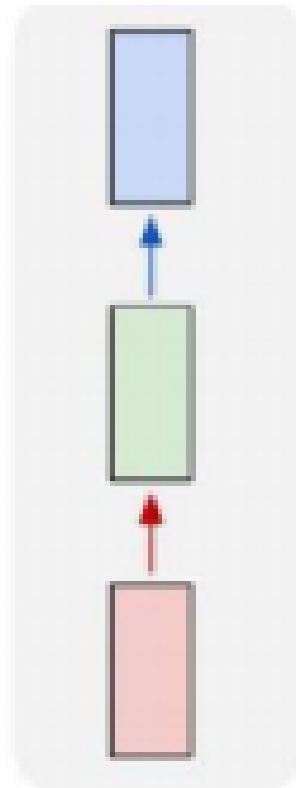
RNN이란

- 입력과 출력을 시퀀스 단위로 처리하는 모델

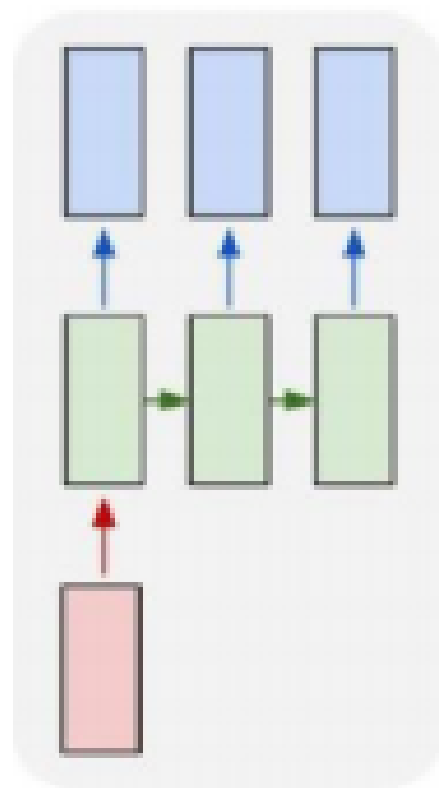


RNN

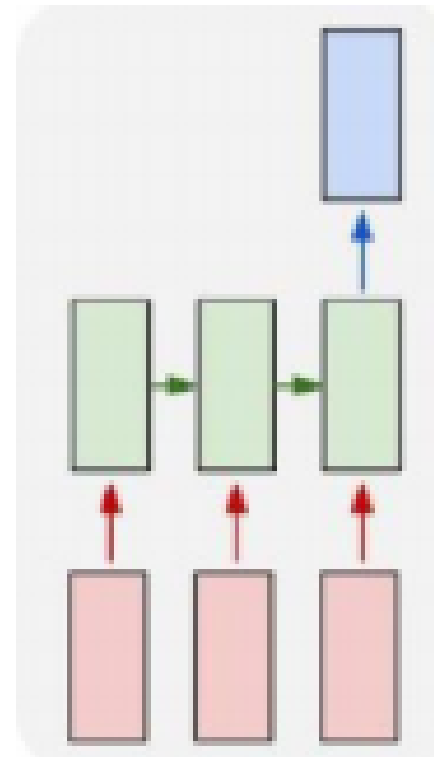
RNN 구조들



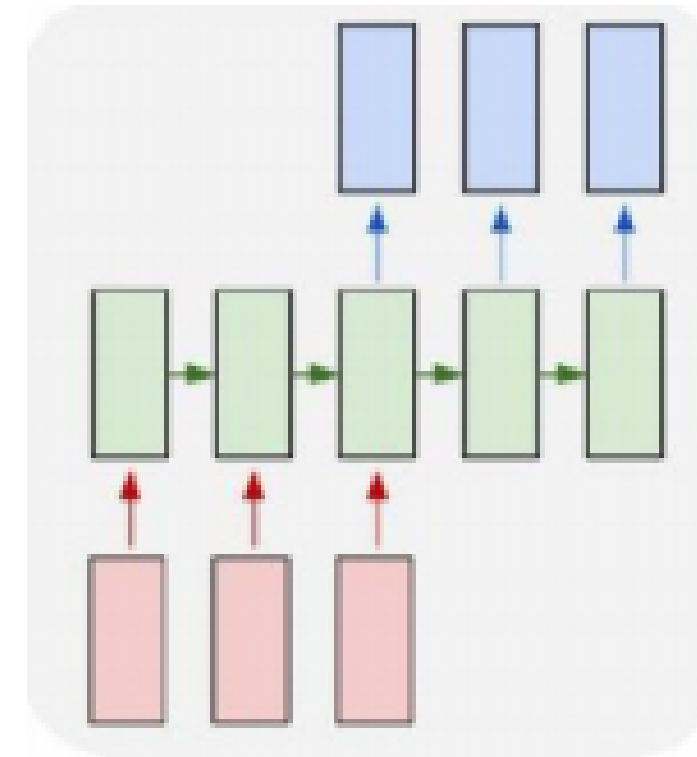
ONE TO ONE
(기본형)



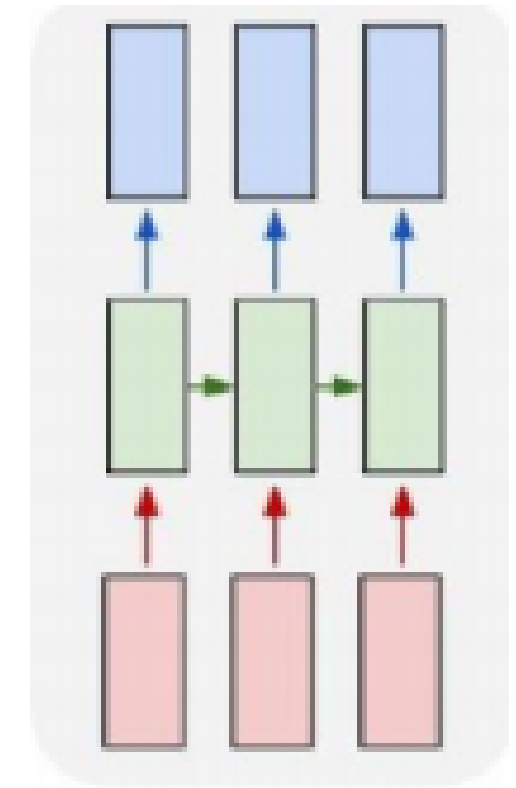
ONE TO MANY



MANY TO ONE



MANY TO MANY
1



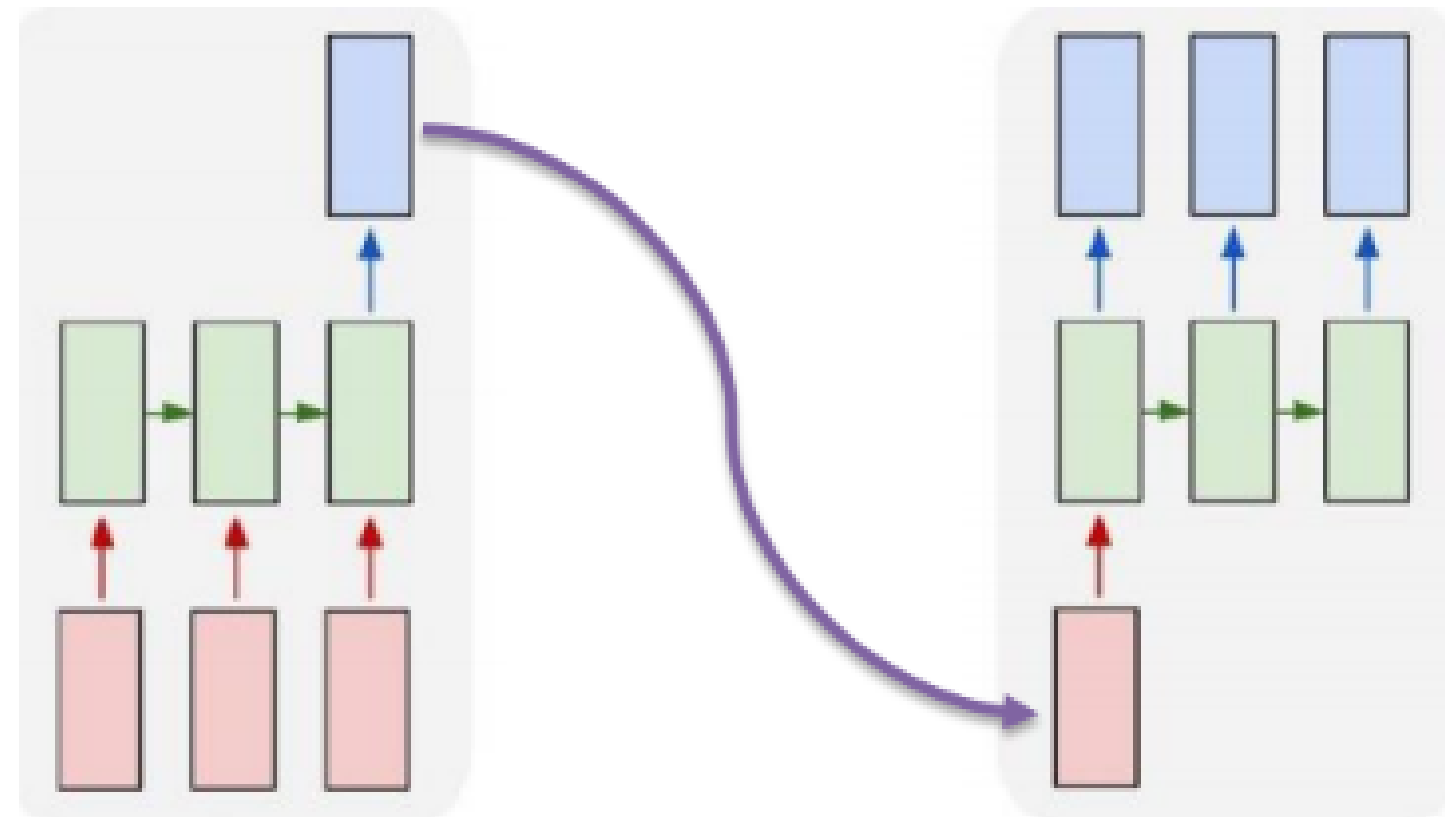
MANY TO MANY
2

RNN

seq2seq

- Many-to-One 와 One-to-Many를 합한 것
- 주로 번역기에서 사용함

인코더



디코더

RNN

RNN의 input과 output

- RNN의 INPUT은
 - sequence인 데이터
 - 이전의 hidden state
- RNN의 OUTPUT은
 - 현재의 hidden state
 - 처리된 결과물 (마지막)

RNN

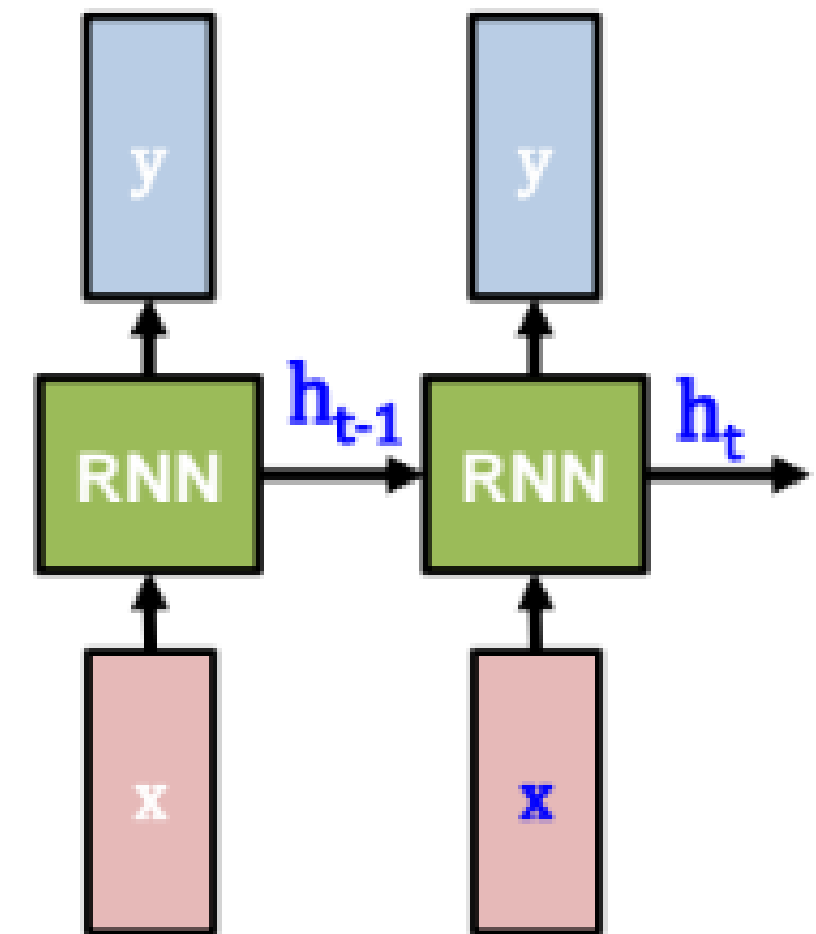
hidden state와 Memory cell

- hidden state는 다음 cell로 보낼 값
ex)

$$h_t = f_w(h_{t-1}, x_t)$$

Some function with parameters W
 New state Old state Input vector at some time step

- Memory Cell은 RNN에서 하나의 Layer을 지칭하는 말

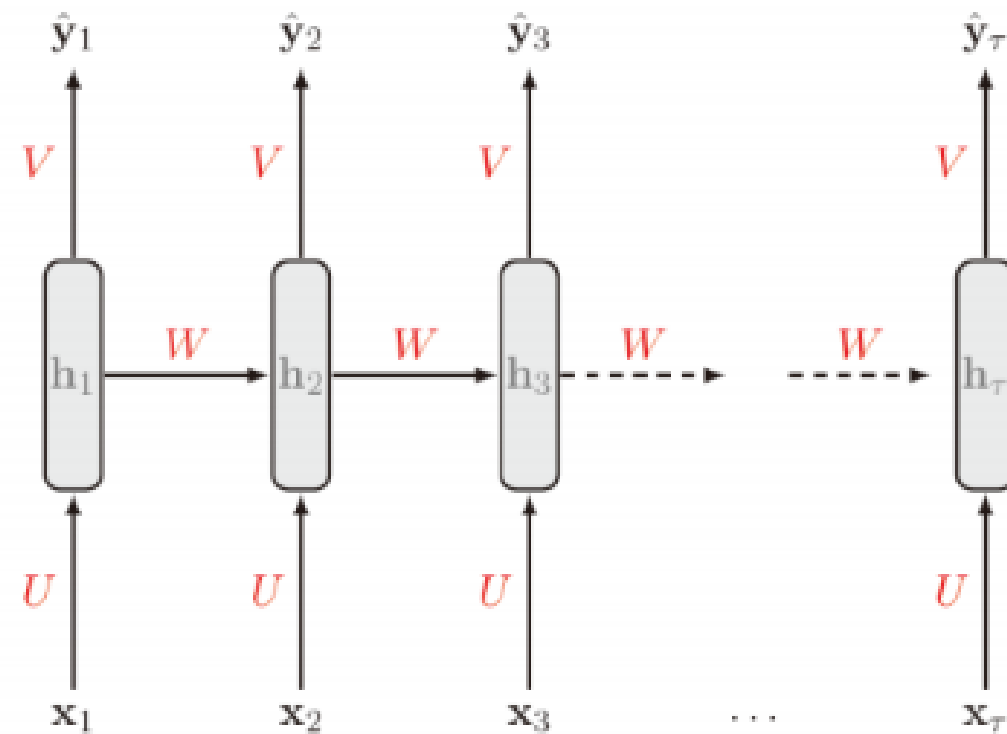


RNN

RNN의 Computational Graph

$$h_t = \psi(W h_{t-1} + U x_t) \quad \Psi(\text{psi}) : \tanh \text{ 등}$$

$$\hat{y}_t = \phi(V h_t) \quad \phi(\text{phi}) : \text{softmax 등}$$



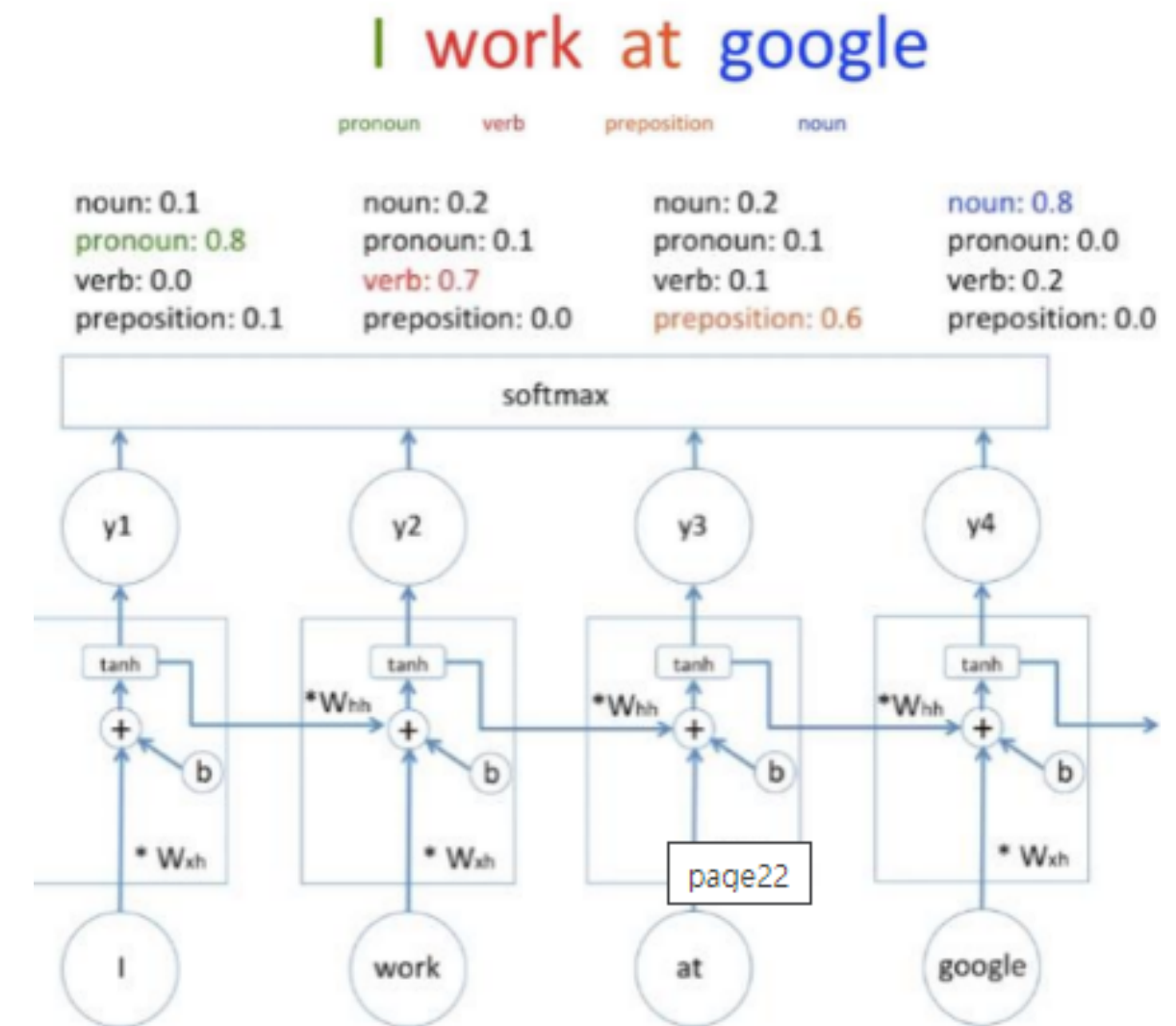
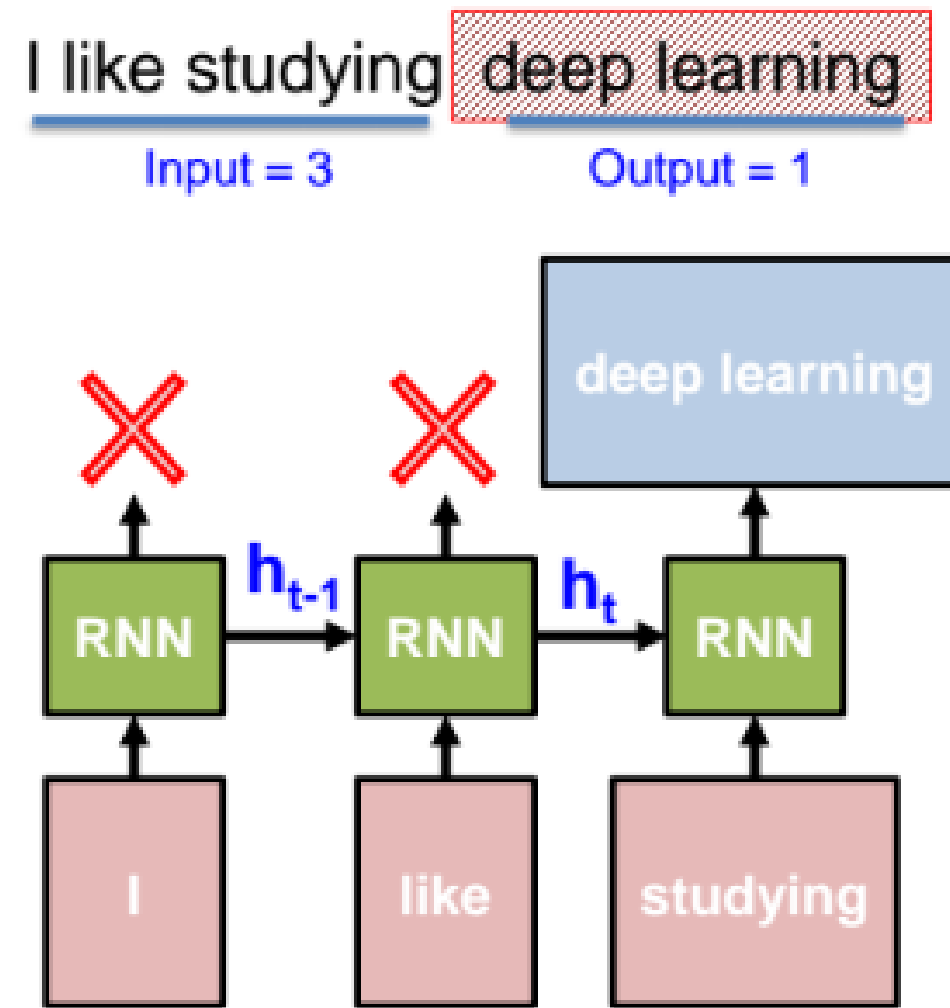
U : 파라미터인 tensor 값 = 가중치 W_{xh}

V : 파라미터인 tensor 값 = 가중치 W_{hy}

W : 파라미터인 tensor 값 = 가중치 W_{hh}

RNN

RNN 적용 예시



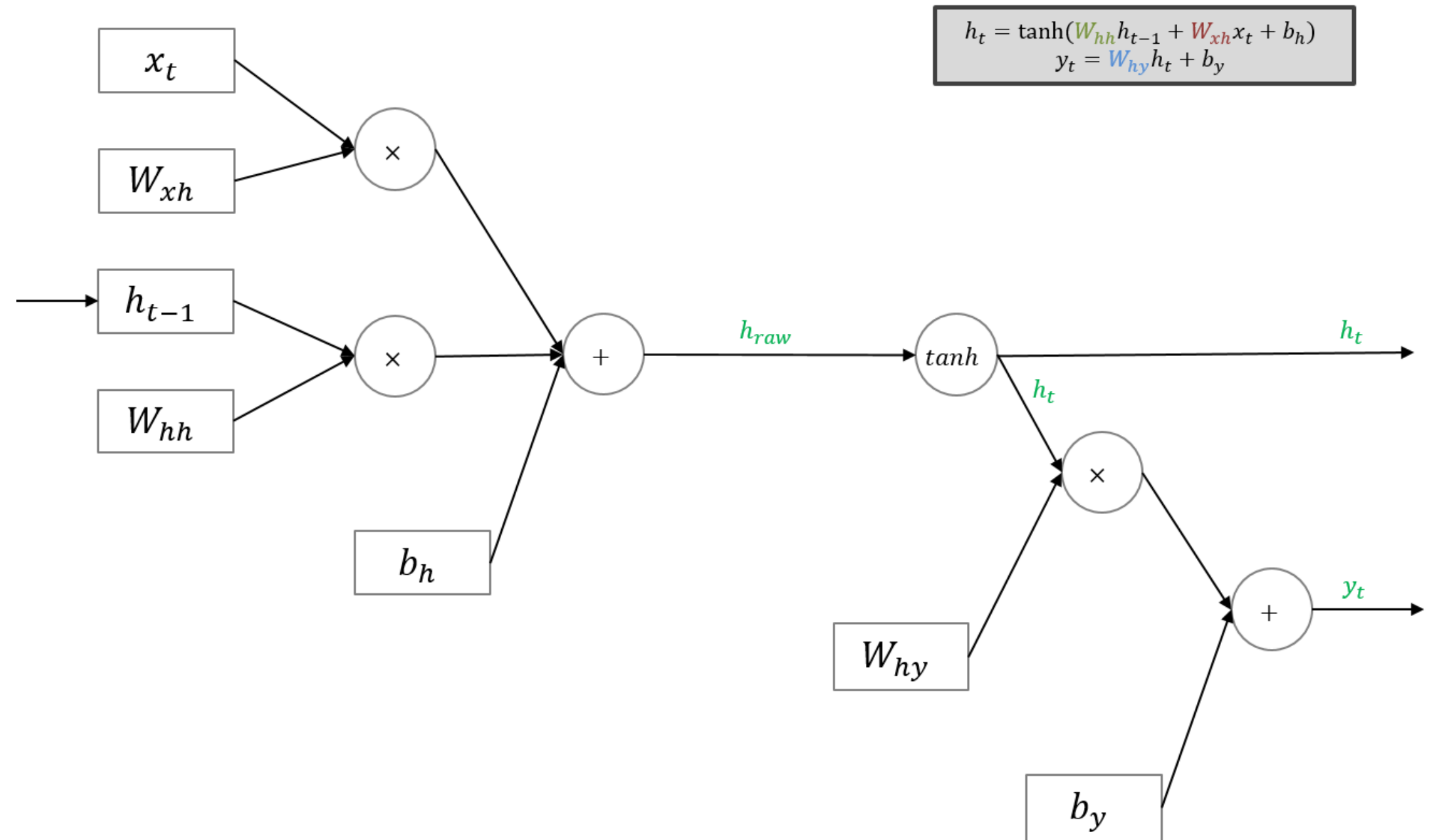
POS TAGGING

RNN

RNN의 Forward Propagation

특징 :

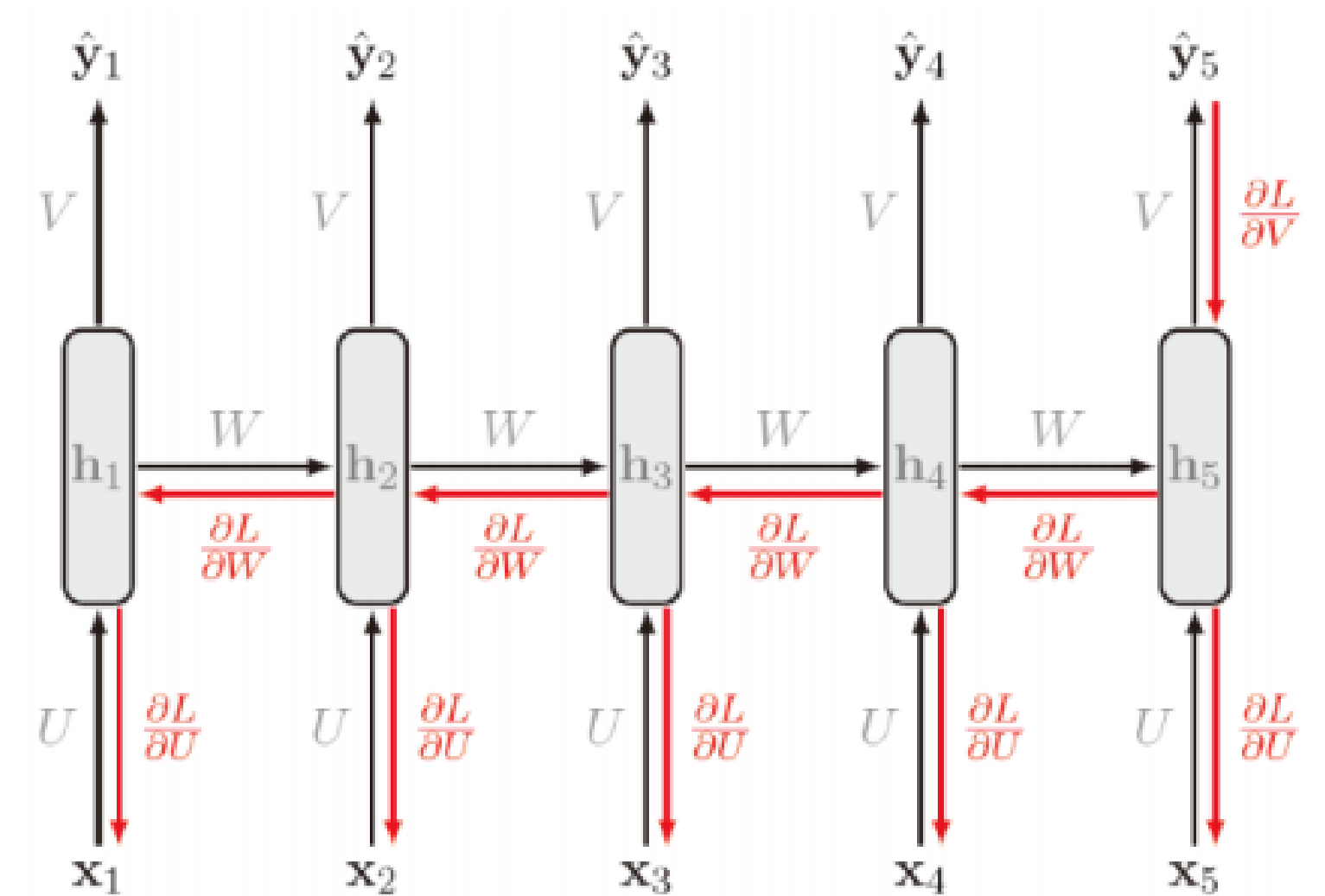
- CAN'T BE PARALLELIZED
- loss function은 전체 합치기



RNN

RNN의 Backward Propagation

- 이러한 과정을 BPTT라고 함



RNN

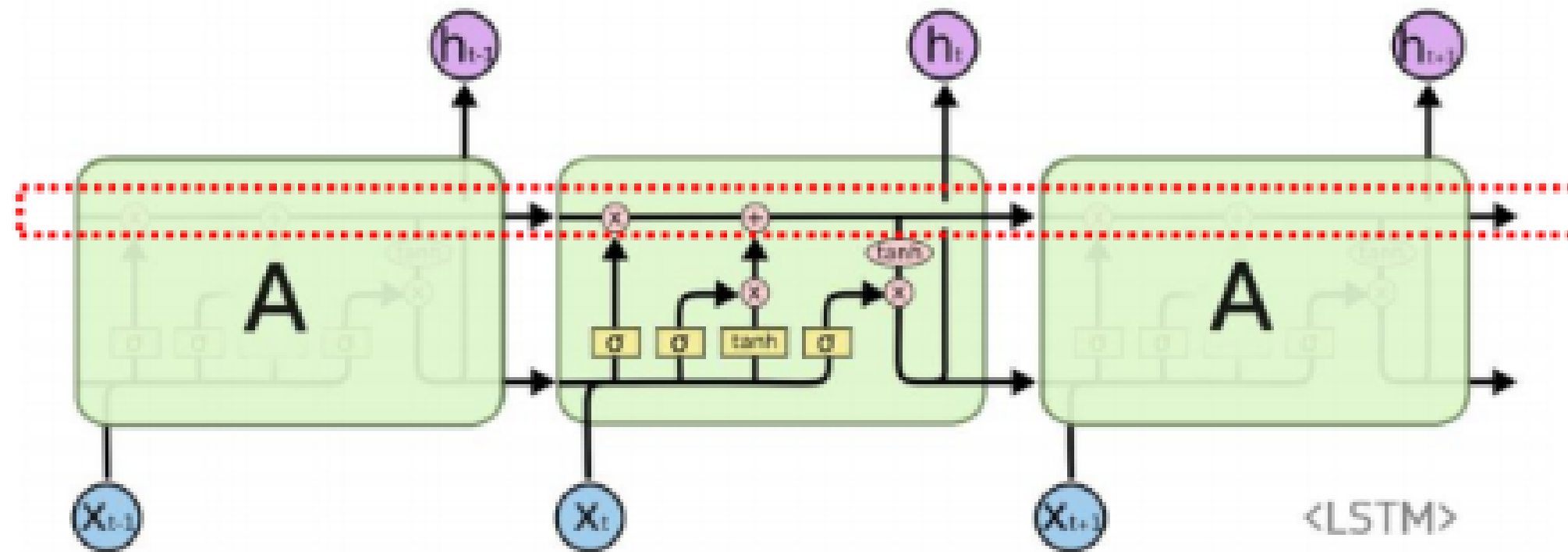
RNN의 한계점

- Gradient Vanishing / Exploding
 - 파라미터가 공유되기 때문에
 - (1) parameter가 1보다 크면 : gradient가 무한대로 가서 gradient exploding
 - (2) parameter가 1보다 작으면 : gradient가 0으로 가서 gradient vanishing

LSTM

LSTM 이란

- Long-short term memory
- 메모리를 여러 개로 나눔
- 컨베이어 벨트처럼 행동하며 state를 잘 유지해서 전파가 잘 되도록 함



<LSTM>

LSTM

LSTM Cell

- (개념 설명)

forget gate $f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f)$

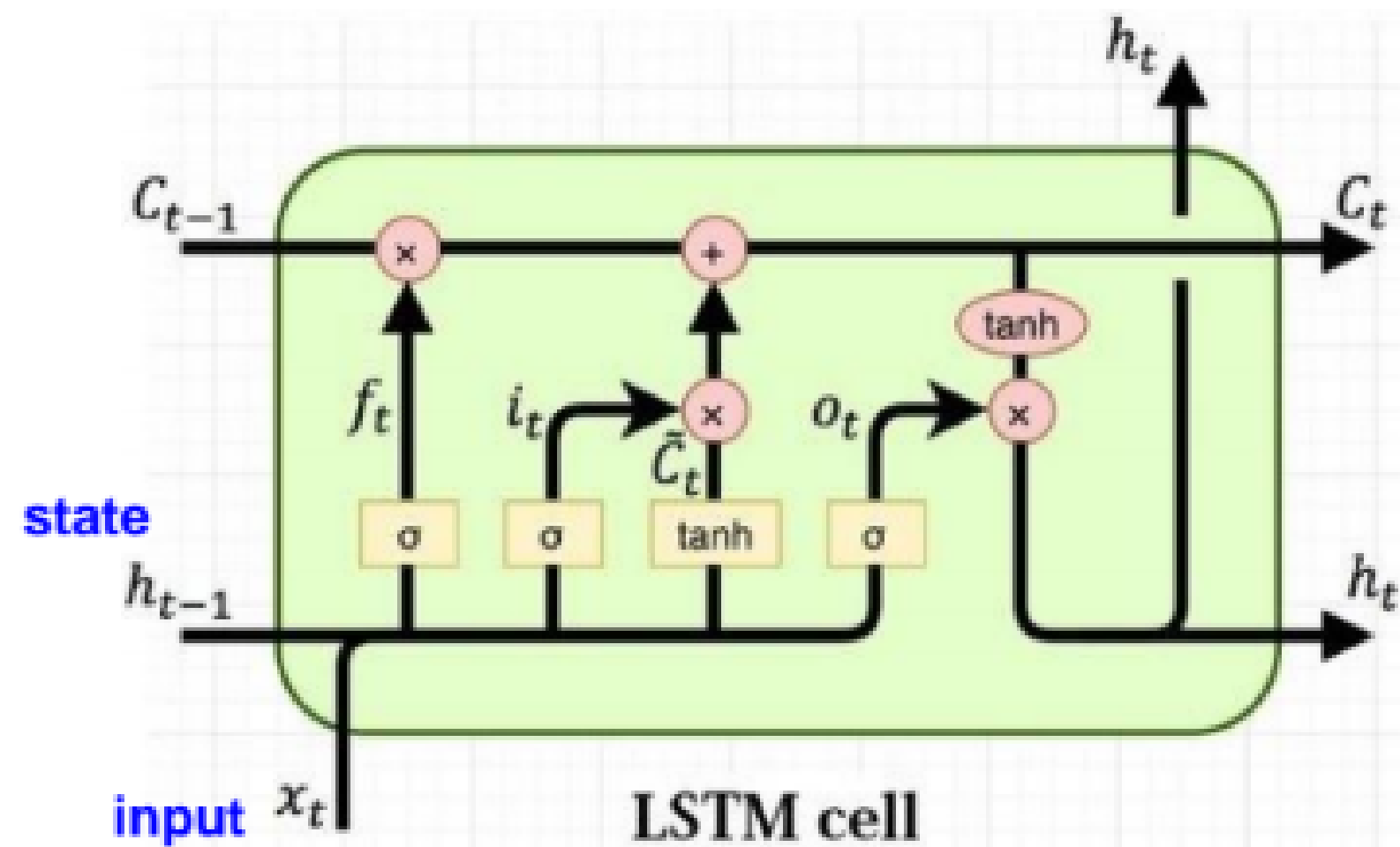
input gate $i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i)$

$$\tilde{C}_t = \tanh(W_g x_t + V_g h_{t-1} + b_g)$$

hidden output $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$

gate $o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o)$

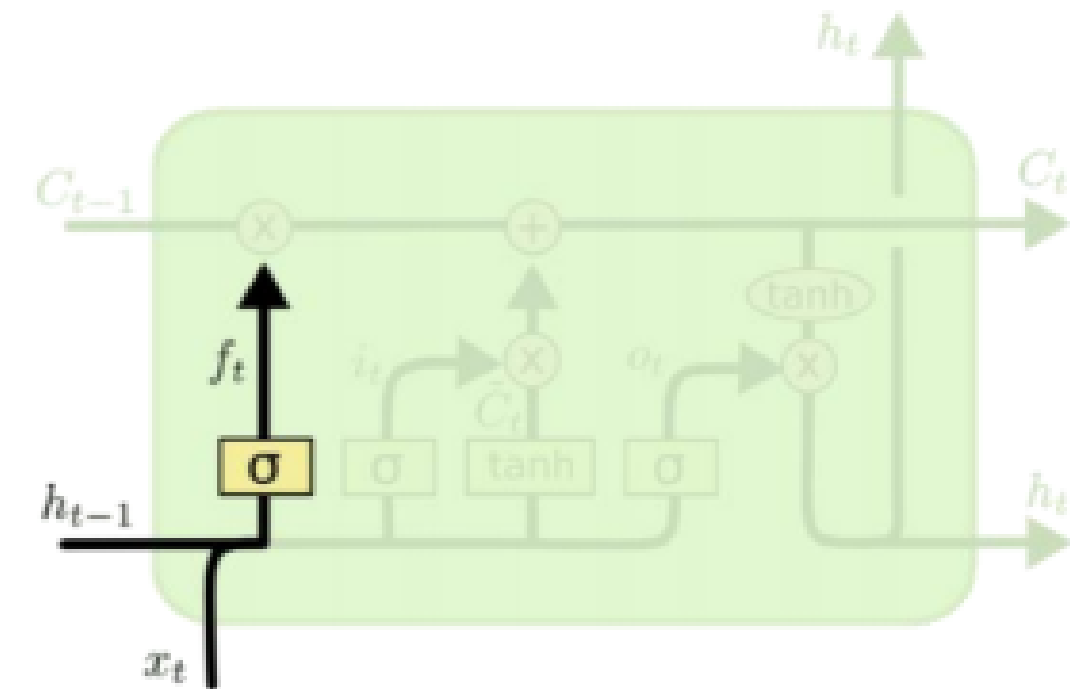
output $h_t = o_t \odot \tanh(C_t)$



LSTM

LSTM Cell – Forget Gate

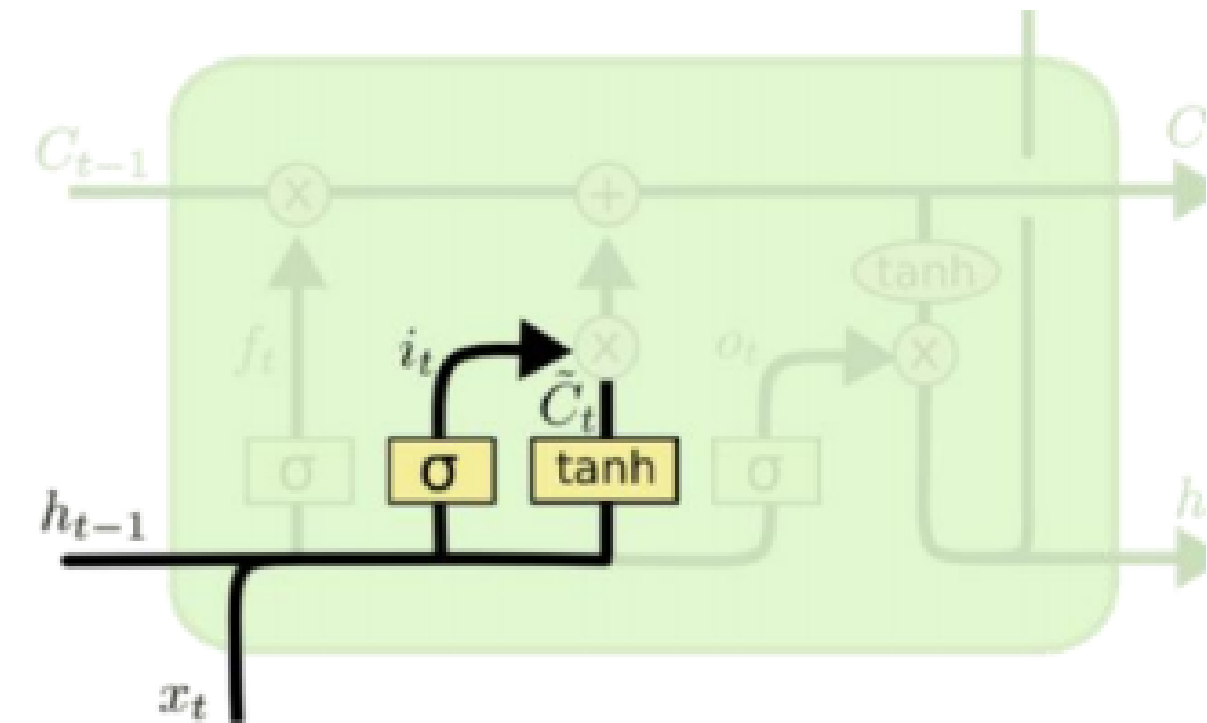
- 과거의 정보를 얻기 위한 게이트
- h_{t-1} 와 x_t 에 대해 sigmoid를 씌워준 값.
- Output 종류
 - 0 : forget
 - 1 : preserve (유지)



LSTM

LSTM Cell – Input Gate

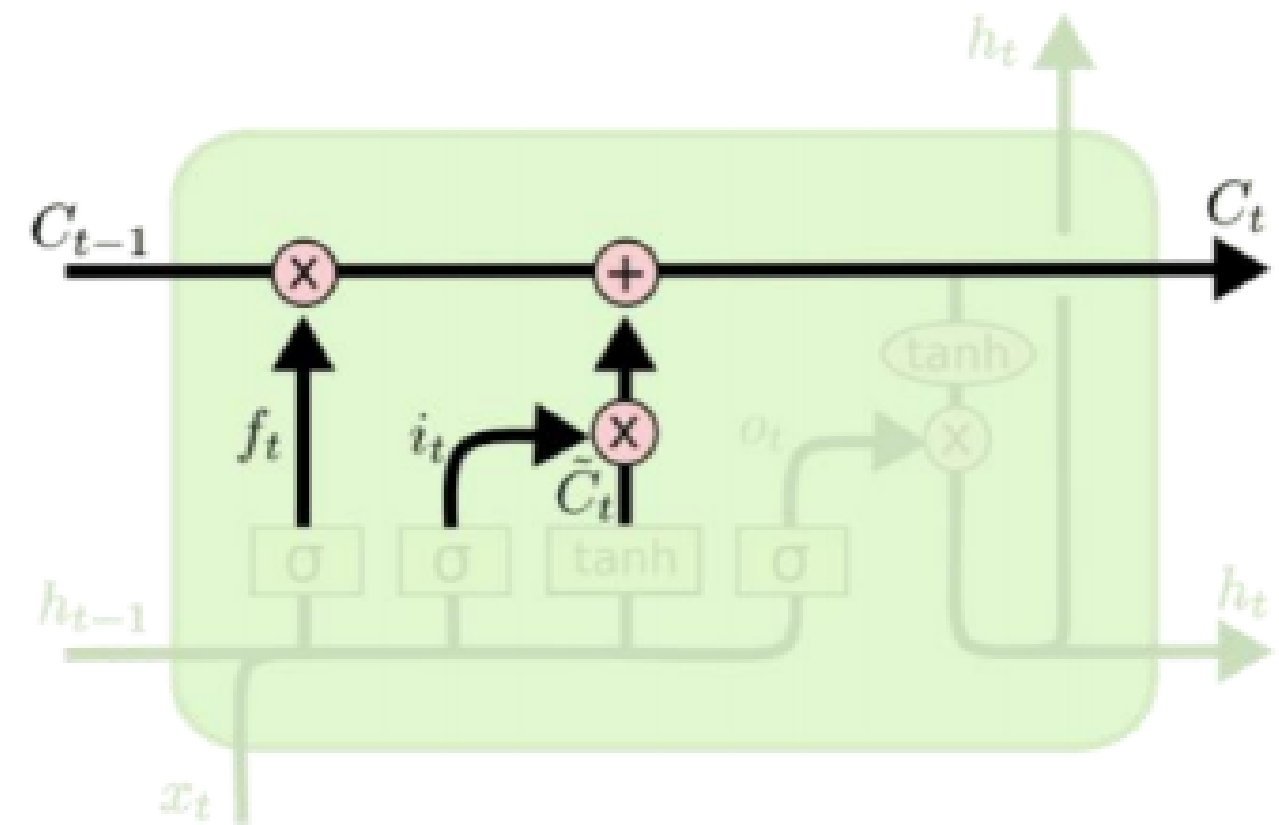
- 현재의 정보를 기억하려는 게이트
- h_{t-1} 와 x_t 에 대해 sigmoid와 tanh를 씌워준 값으로 hadamard 곱을 함



LSTM

LSTM Cell – Update Gate

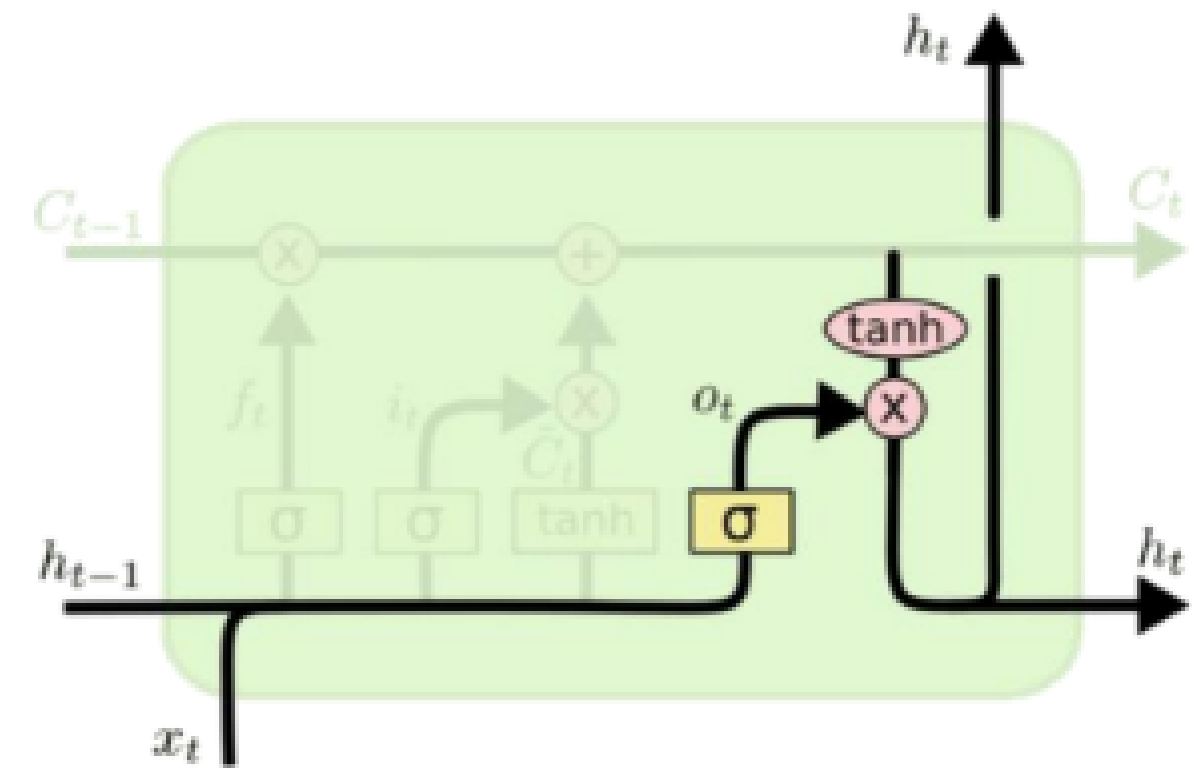
- 얼마나 업데이트를 할 건지 결정하는 게이트
- forget gate와 input 게이트를 hadamard 연산을 통해 계산함



LSTM

LSTM Cell – Output Gate

- 무엇을 결과물로 내보낼지를 결정함



LSTM

LSTM이 Vanishing Gradient을 해결한 방식

- f 가 sigmoid의 output이 되므로 explode는 완전히 해결함
- f 가 sigmoid의 output이 되므로 1에 가까울수록 vanishing이 최소화됨
- 완벽하게 Vanishing Gradient를 해결하진 않았지만, 더 오래 기억할 수 있도록 함

KNN / SVM / RNN / LSTM

Thank you