

• 12/13

• Ensemble (앙상블)

↳ 특별한 Machine Learning 알고리즘은
아니예요!



여러 모델을 결합해 성능을 높이는 기법!

크게 3가지 기법이 있어요

- ① Voting
- ② Bagging
- ③ Boosting

① Voting

hard Voting

Soft Voting

결과가 같아요

* hard Voting

($\frac{1}{2}$) 개의
class를 가짐

여러모드
모든!

3개의
서로 다른

model을
사용

최종결과는 ①

다른결과
설정

0 : 1개
1 : 2개
2 : 0개

①
=

②
=

③
=

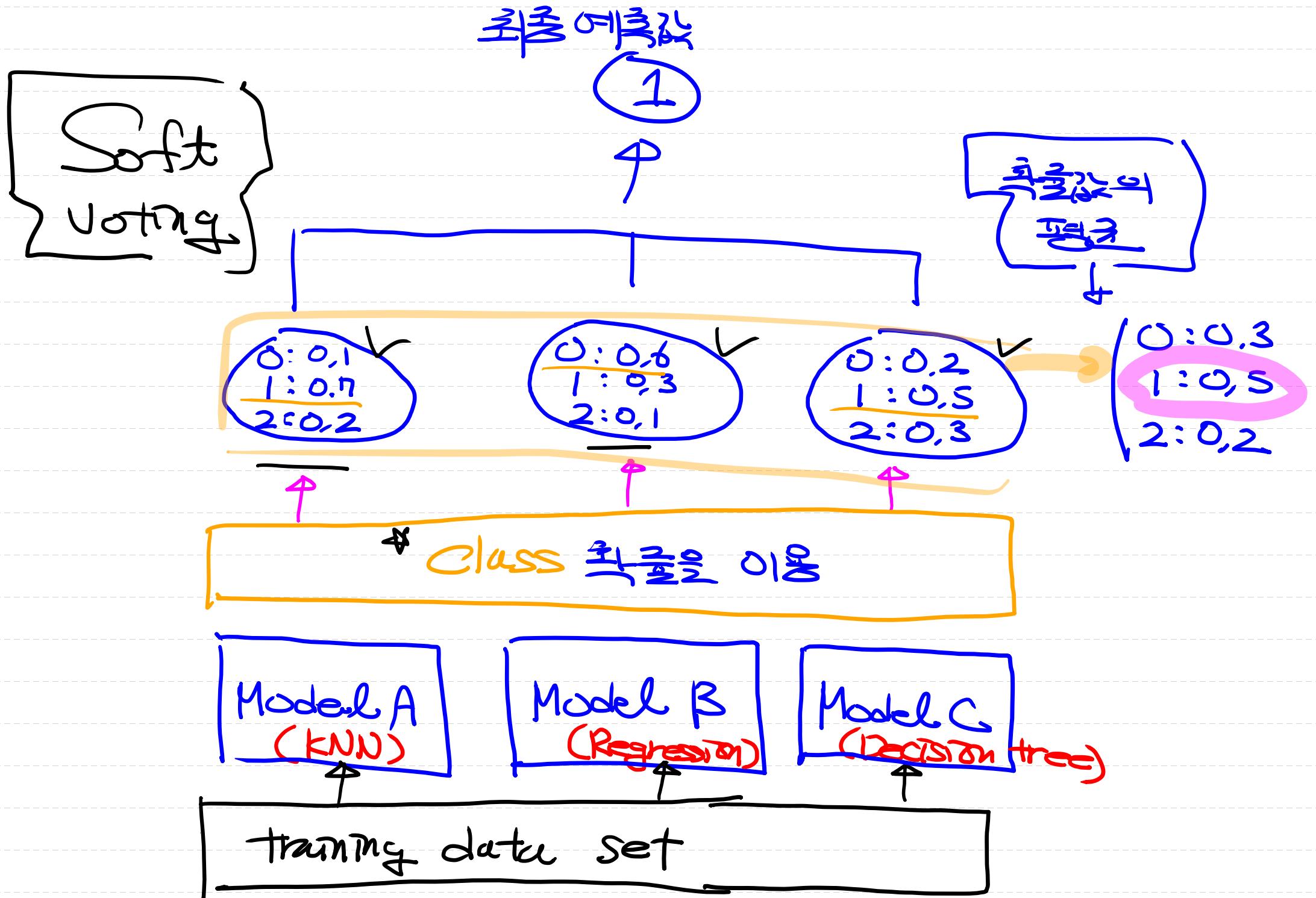
* Class로 예측

Model A
(KNN)

Model B
(Regression)

Model C
(Decision tree)

training data set



② 투표법 - Bagging

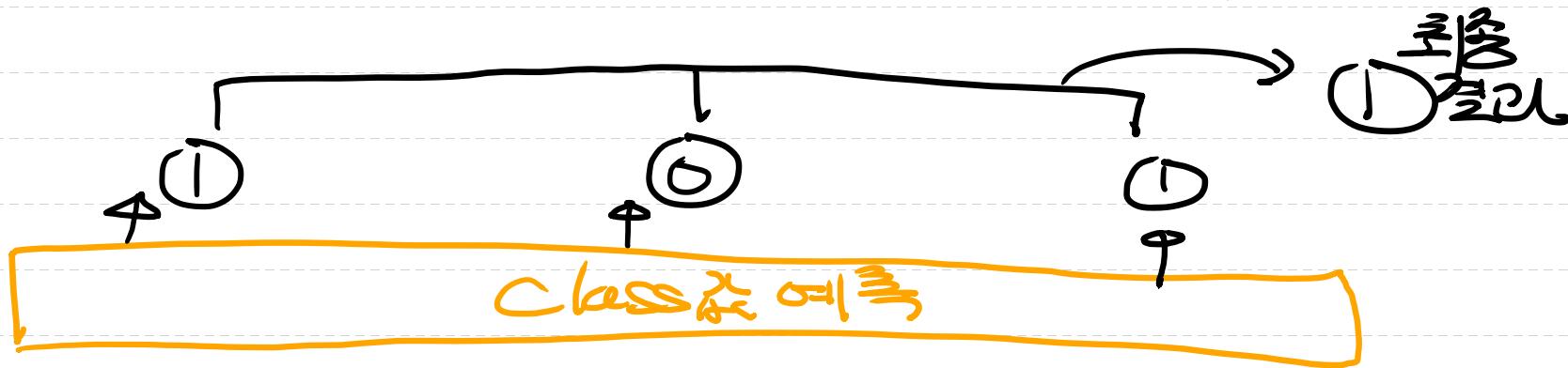
간단한 bagging의 방식을 이용할 때
우리가 사용하는 model이
Decision Tree OLPt

Voting 방식과 유사한데 Voting을 기본적으로 따른

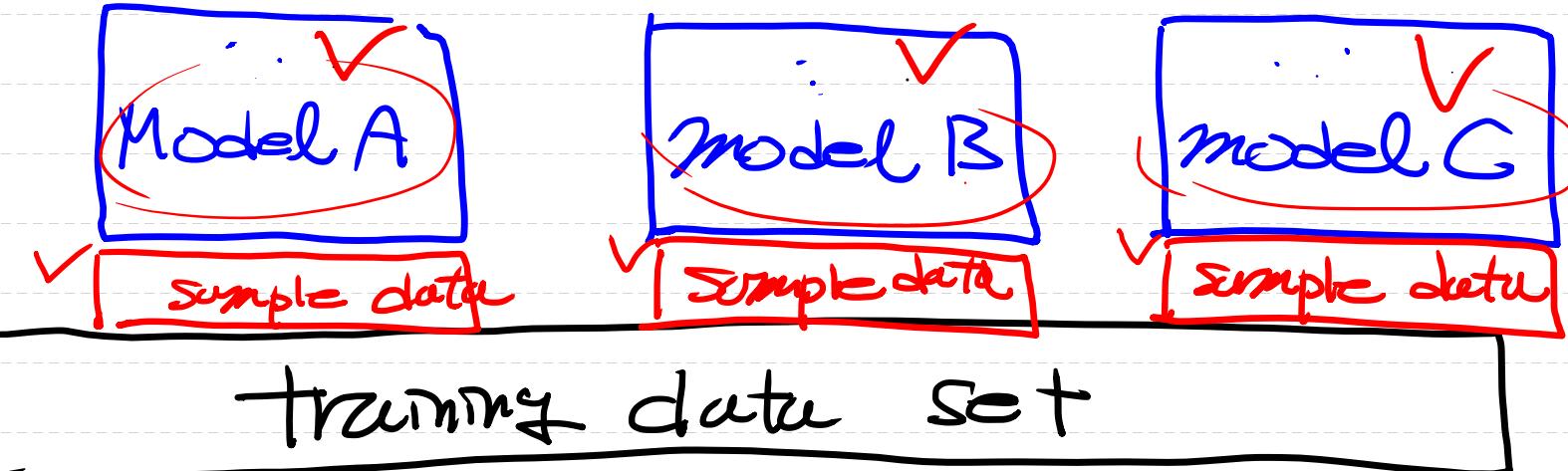
Machine Learning 알고리즘을 사용.

Bagging 방식을 같은 모델 여러개를 이용해요

Random Forest



각각의
model



③ 앙상블 - Boosting

Boosting은 여러개의 가벼운 model을 조합적으로 학습하는 기법

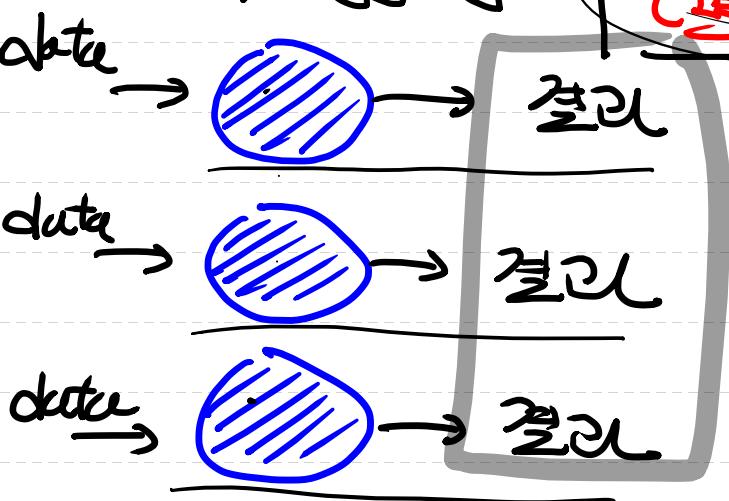
각각 예측한 데이터에 따라 오차를 줄일수 있는

방향으로 model을 update해

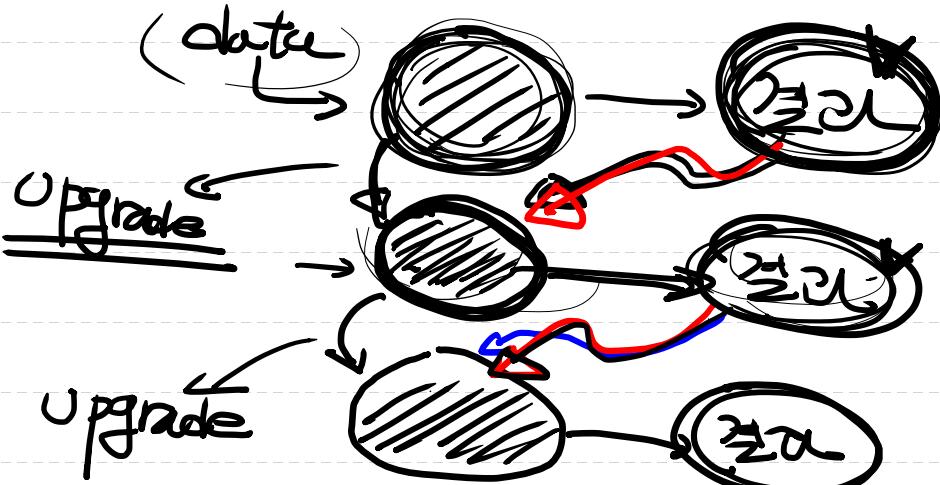
나가는 방향

* Bagging

Parallel
(병렬)



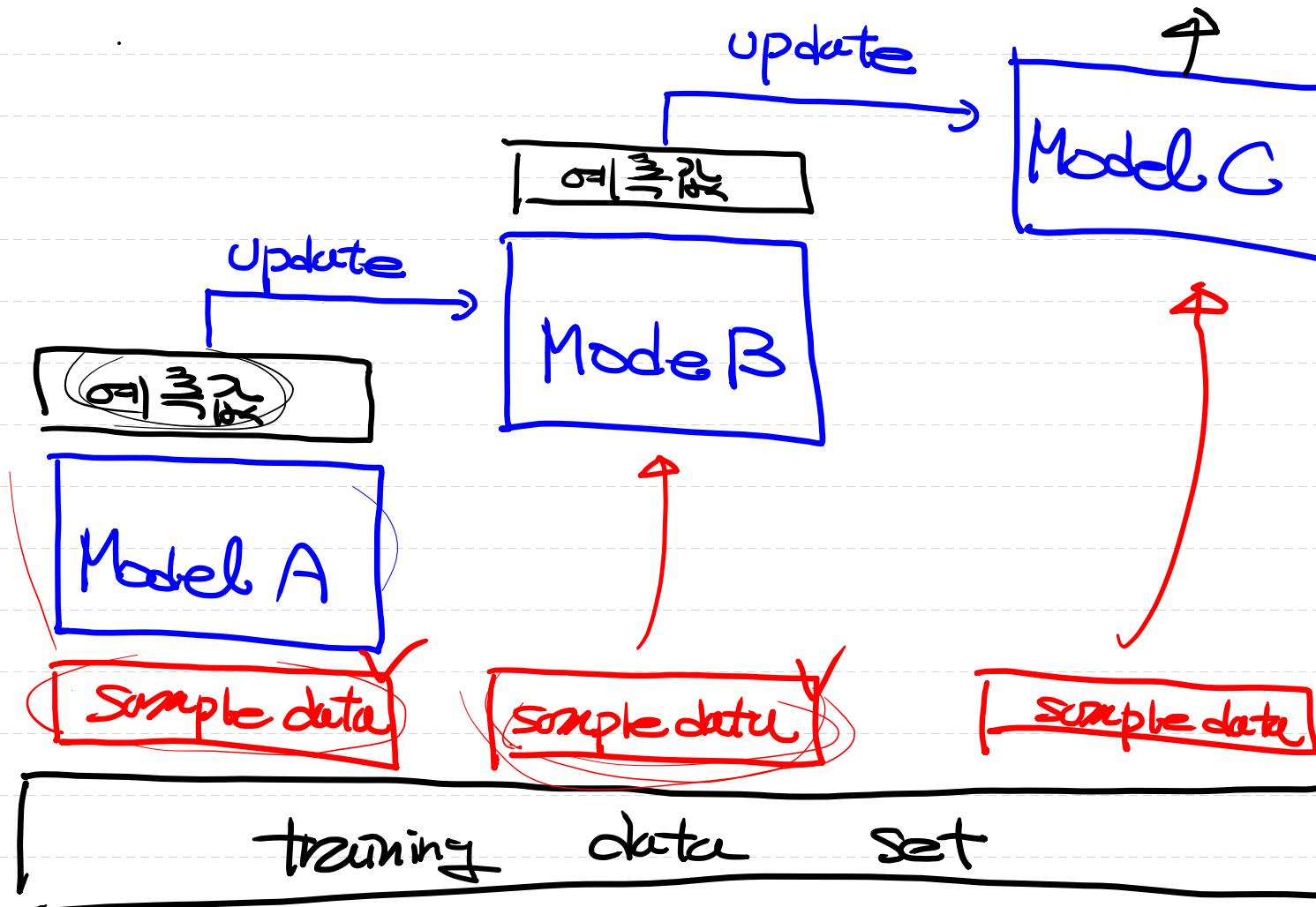
* Boosting *



수집된 결과

XGBoost model → 일반적으로 많이 사용되는
boosting 알고리즘.

그리고 Ensemble – boosting이나 DTM에 알아보면
최종예측값!



● Deep Learning → DNN (Deep Neural Network)

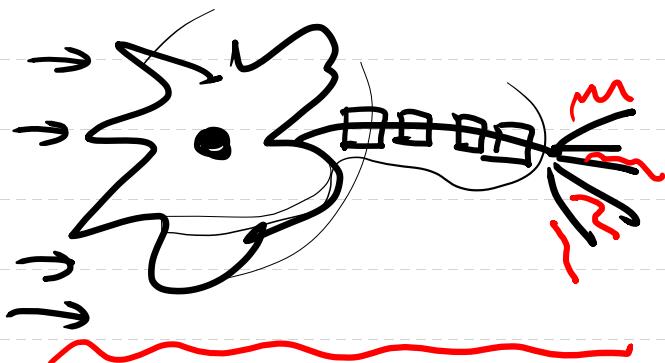
- 우리의 최종 목표 → AI를 구현하는 것



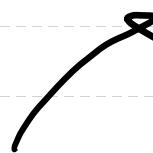
Machine Learning을 이용해서 구현하는 것

우리가 선택한 경로!

neuron



- SUM ✓
- Decision Tree ✓
- Regression ✓
- KNN ✓
- ✓ **Neural Network**
- K-Means, DBSCAN
- Ensemble
- PCA



이것의 끼를 model로
Machine Learning을
구현!

- 1957년 크리블렛이 가장 간단한 형태의

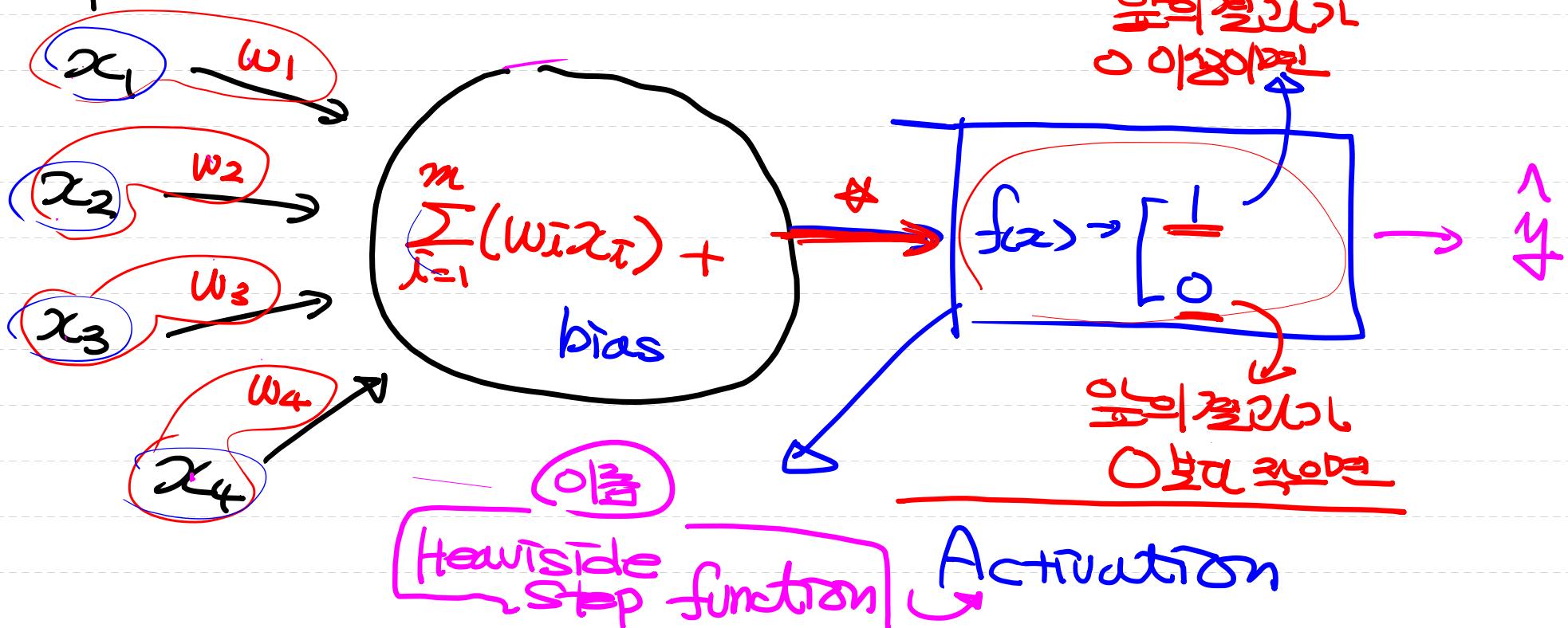
neural Network을
만들어요

* Perceptron

- * Single-layer perceptron Network



Perceptron



- 이런 perceptron으로 복잡한 문제를 해결 가능?

컴퓨터는 작동하기

↳ 논리회로로 구성 → GATE라고 표현되는

Perceptron은 CPU를 구성하고 있던 GATE를 표현하는
있으면

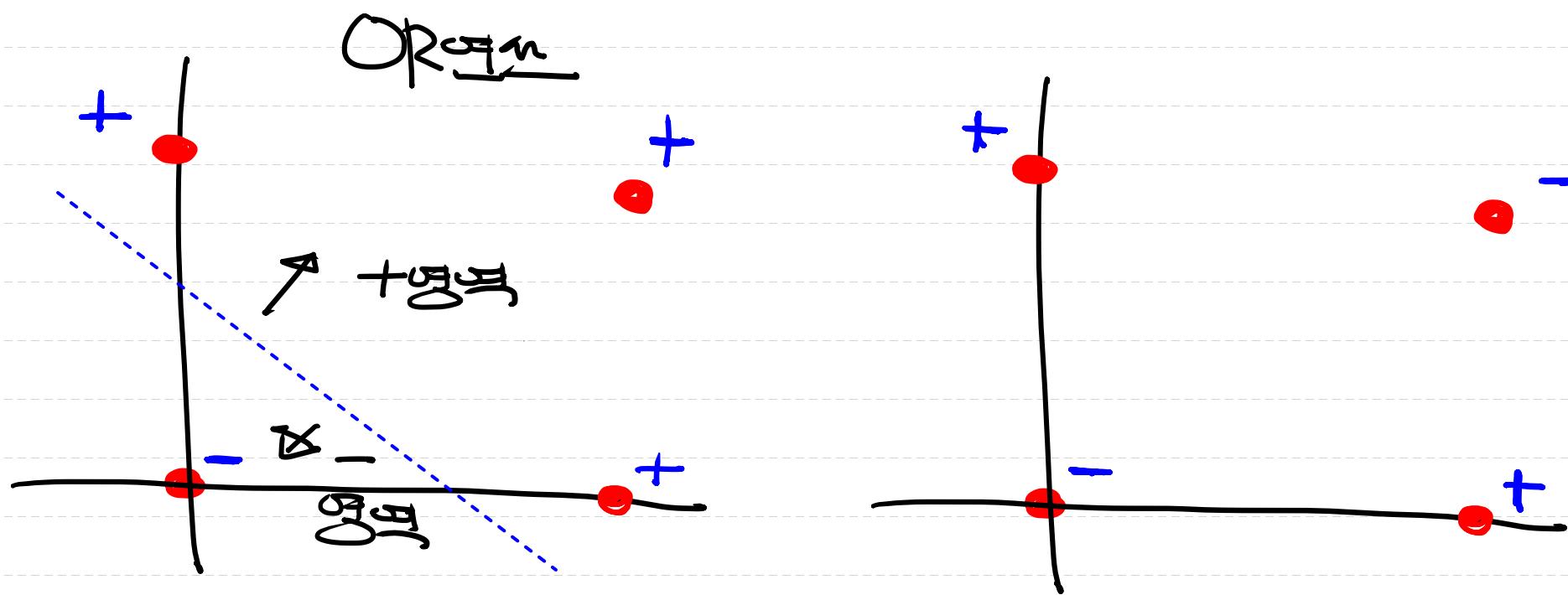
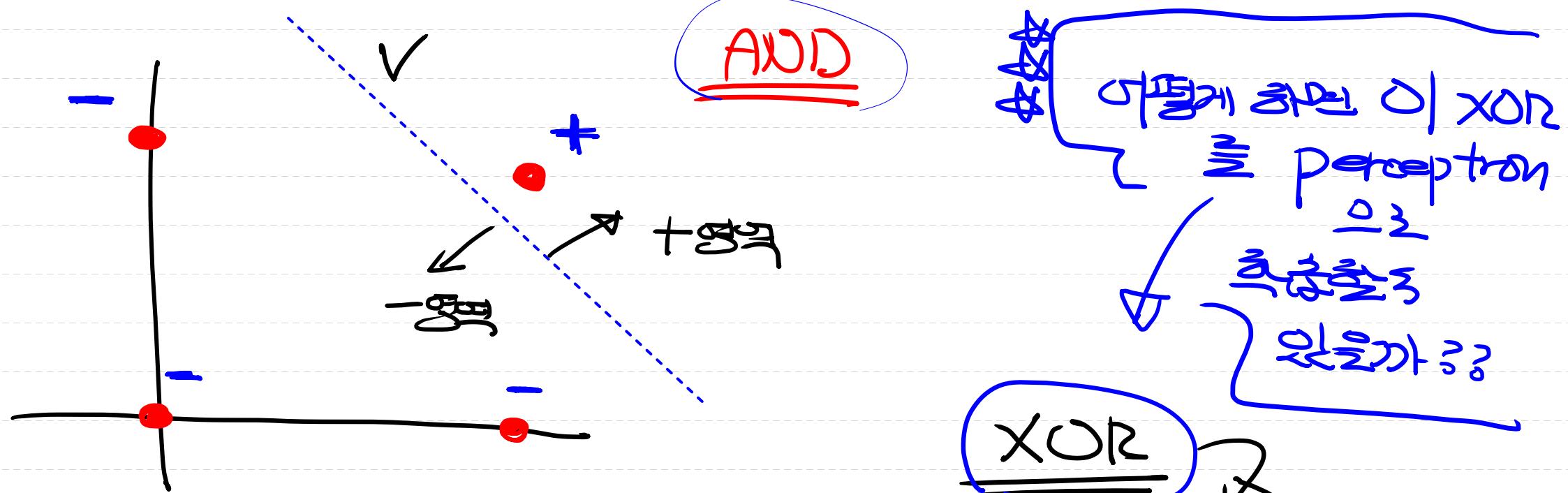
* Perceptron은 다양한 GATE 연산을 구현할 수 있는지 연구!

*** AND 연산 OR연산 NOR연산 XOR연산.

x	y	
0	0	0
0	1	0
1	0	0
1	1	1

x	y	
0	0	1
0	1	1
1	0	1
1	1	0

x	y	
0	0	1
0	1	0
1	0	0
1	1	1



1969년 마빈 미크로

"단일 layer의 perceptron"

단 하나의 perceptron으로 학습이 불가!

☞ multilayer perceptron으로 학습이 가능

그런데 이 방식은 전달을 할 때 어려워 그 방법은
할 수도 어려워!

수학적 풀이 → 책으로 봄.

→ 이 이후로 neural network이 급격히 연구
되어가게 되요!!

1974년

풀 웨이보스

"Back propagation"

→ 이렇게 하면 학습이 가능할까 같아요!! XX

82년 당시 논문을 봄XX

그림

없음

XX

1986 "제프리 힐튼"

✓ keras안에 이미 이기능이
적용되어
있어요

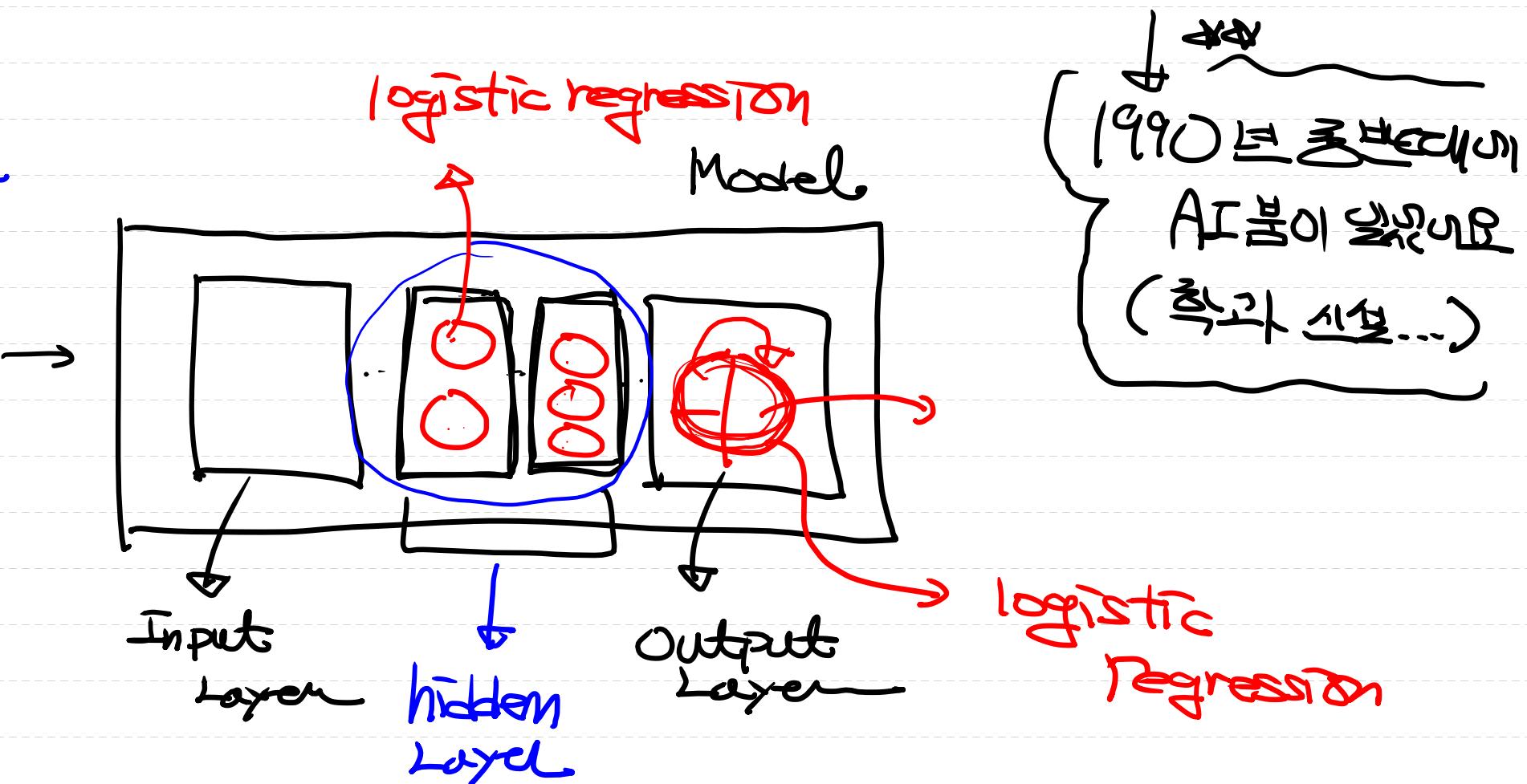
↳ Backpropagation 기법을 재조명!



→ 딥 Neural network의 전성기가 시작!!

XOR data

x	y	t
0	0	0
0	1	1
1	0	1
1	1	0



○ 아주 복잡한 문제를 해결하기 위함

Layer와 Layer 사이 node 간을 추가하면 해결 가능!!



많은 Layer를 사용할 경우



Vanishing Gradient

현상 발생



다시 이야기는 (ML - neural network)
의 시작!



다른 Machine Learning

알고리즘을 연구.

Regression
SUM
Decision Tree (Random Forest)
KNN
K-Means, DBSCAN

"Hinton" → 카카오로 이적.

2006, 2007년

(CIFAR)

2006년 흉터나 놀랄정도로 성과 발표.

✓ Vanishing Gradient 현상을 해결하려면
어떻게 해야 하는가??

(1) weight의 초기화

→ 초기 keras에서
구현

(2) Activation을

Sigmoid → relu

"Deep Network" → "Deep Learning"