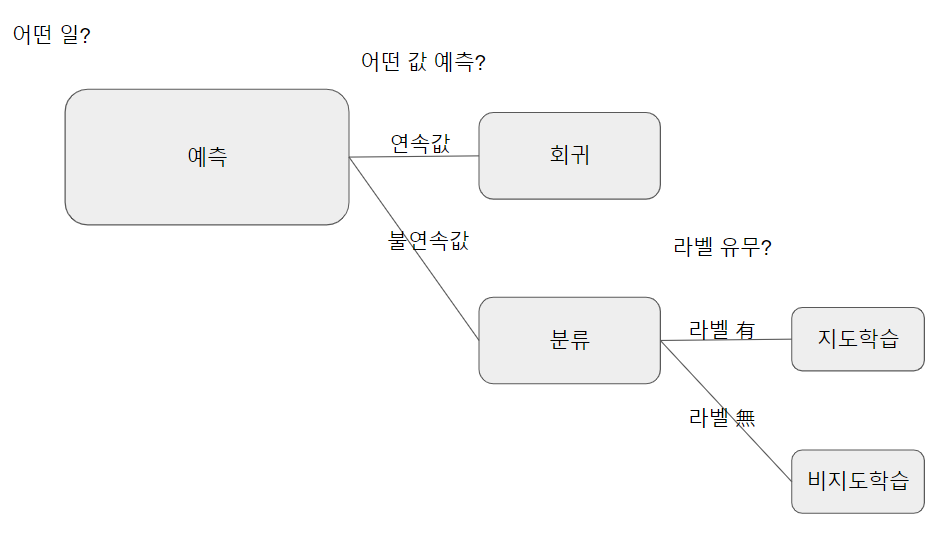
# 1.Learning?

: 컴퓨터에게 예측하는 일을 시키기 위해 1.‘학습 시키고’ 2.‘예측 하게하는 것’.

## 1-1.일의 종류? :



예측

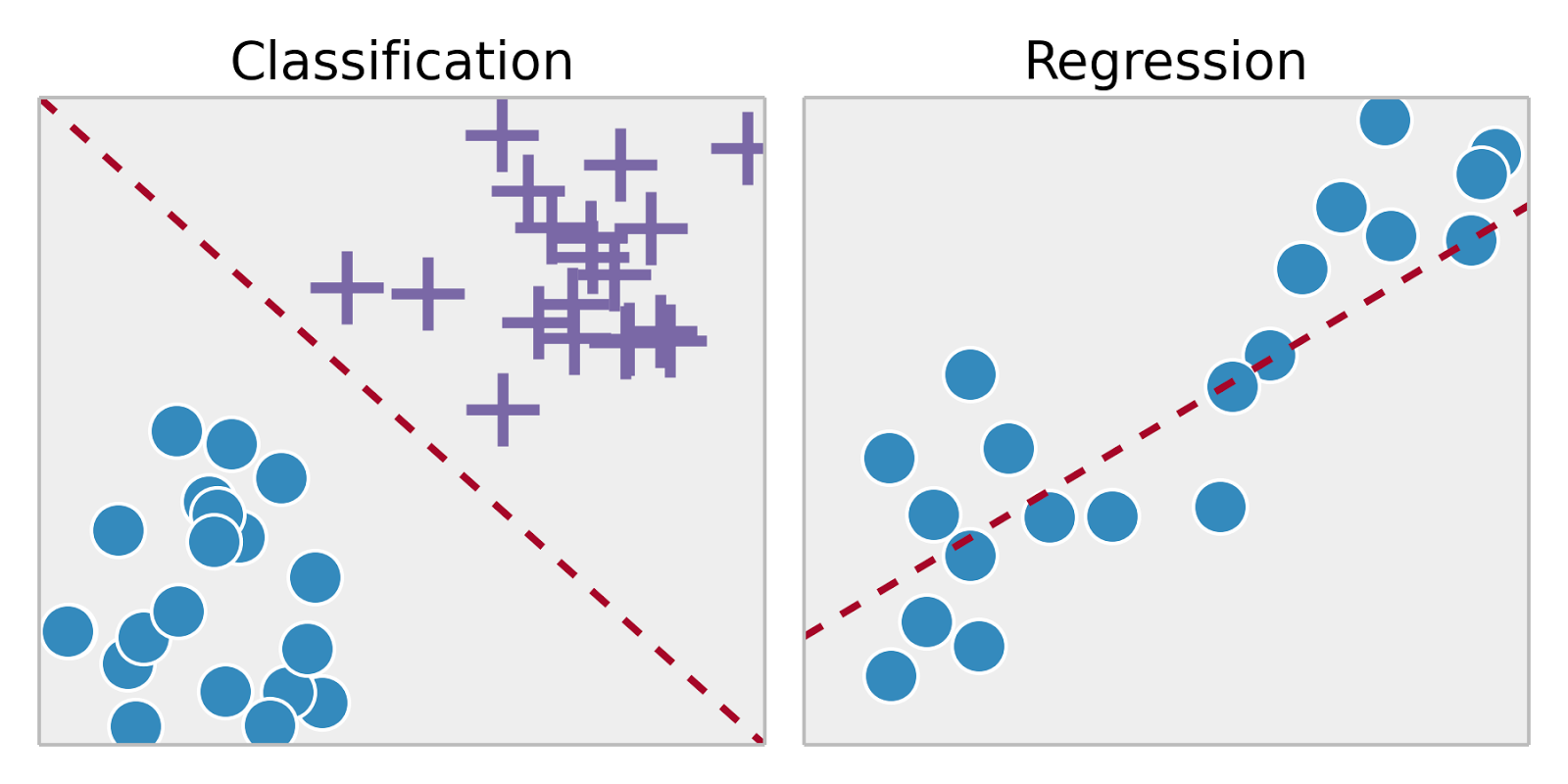
- 값이 몇 이냐?(연속된 값) : 회귀

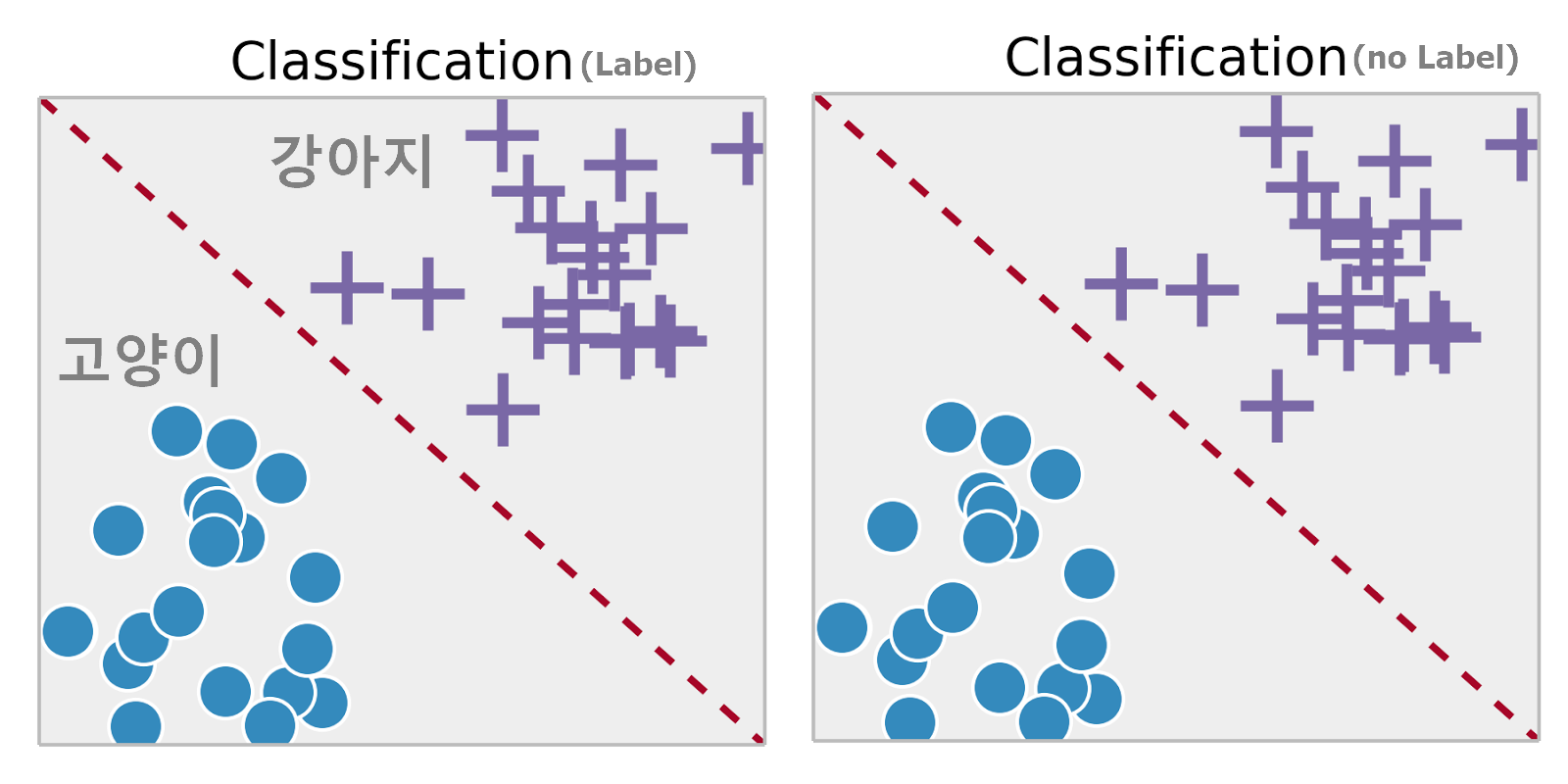
- 값이 무엇이냐?(불연속 값) : 분류 - 라벨이 있느냐? : 지도학습

(이거 이거 중에 분류해줘)

- 라벨이 없느냐? : 비(非)지도학습

(데이터가 이런데 몇개로 분류해줘)





## 1-2.학습과 예측?

학습 : 예측하고자 하는 값을 잘 설명하는 빈 모델을 선택하고,

학습용 데이터로 학습시키는 것.

즉 빈 모델을 최적화해 나가는 과정.

예측 : 학습된 모델에 실제 데이터를 넣어 예측값을 뽑아 내는 것.

# 2.Machine Learning(ML)?

컴퓨터를 ‘학습(모델 학습)’시키고, ‘일(예측) 시키기

## 2-1.ML에서 학습?

학습이란 빈 모델을 선택하고 학습용 데이터를 학습하는 것.

빈 모델은 학습을 통해 무엇이 최적화 될까?

‘가중치’가 최적화 된다.

가중치??

예:



공용 자전거 이용 데이터

특성은 시간만 있다고 쳐보자.

시간에 따른 이용자 수의 데이터가 있다.

이 때 우리는 좌표를 찍어보고,

가장 데이터를 잘 대변하는 직선을 하나 그릴 수 있다.

y(이용자 수) = w1(가중치 = 선의 기울기)\*x + b1(편향)

(기울기는 미분을 통해 구할 수 있음.)

그러면 특성이 더 있다고 해보자.

시간, 날짜

시간과 날짜에 따른 이용자 수의 데이터가 있다.

이 때 좌표를 찍을 수 있나? 물론 3차원에 찍을 수 있다.

y = w1\*x1+w2\*x2 + b

평면이 나올 것.

[https://www.wolframalpha.com/input/?i=W\_1\*X\_1%2BW\_2\*X\_2](https://www.wolframalpha.com/input/?i=W_1*X_1%2BW_2*X_2)

(마찬가지로 각각의 x1, x2에 대해 미분(편미분)을 하면 기울기를 얻을 수 있다.

다만 귀찮고 복잡해질 뿐이다.)

만약 이런거라면?

<https://www.wolframalpha.com/input/?i=log%28x_1%29%2Blog%281-x_2%29>

시간, 날짜, 습도가 있다면?

y = w1\*x1+w2\*x2+w3\*x3 + b

[https://www.wolframalpha.com/input/?i=W\_1\*X\_1%2BW\_2\*X\_2%2BW\_3\*X\_3](https://www.wolframalpha.com/input/?i=W_1*X_1%2BW_2*X_2%2BW_3*X_3)

시간, 날짜, 습도, 풍량, 온도..?

즉 모델을 최적화 한다는 것은 데이터를 잘 대변하는 가중치 w들과 편향 b를 구하는 것

## 2-2.ML 과정

1.데이터 관찰 및 가공(데이터 분석 및 전처리)

2.데이터 나누기(학습용/평가용)

3.학습 시키기(모델에 학습용 데이터 학습)

4.모델 평가(평가용 데이터 사용) + 2~4 반복

5.실제 값 예측 시키기

## 2-2.어떤 데이터냐? => 어떤 모델

어떤 값을 예측해야하냐에 따라

ML 모델(학습 모델, 학습 알고리즘)을 선택한다.

연속값?(회귀 문제)

=>

Linear Regressor : 선형 회귀

Random Forest Regressor

XGBoost

...

불연속값?(분류 문제)

=>

Logistic Regressor : (이항/다항) 회귀

Decision tree & Random Forest Classifier: 결정 트리 모델, 랜덤 포레스트 모델

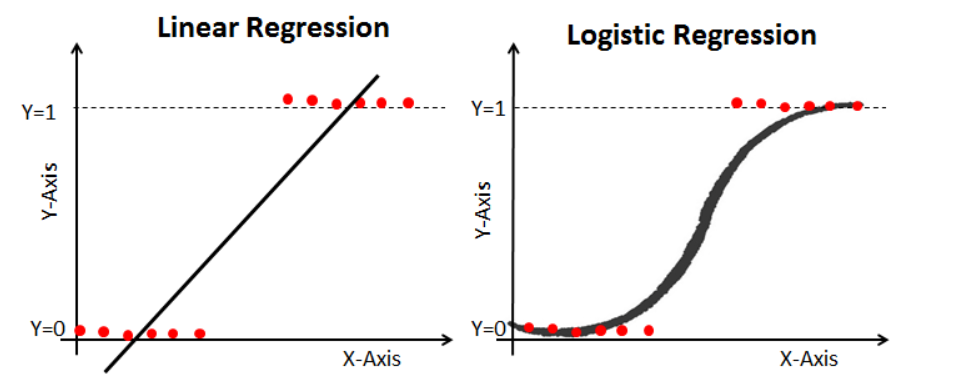
Suport Vector Machine(SVM)

Naiv Baise

XGBoost

...

## 2-3.실습



2-3-1.회귀 - Linear Regressor

2-3-2.분류 - Logistic Regressor

## 2-4.[정리](#cm130okwcvie)

# 3.Deep Learning(DL)

컴퓨터를 ‘학습(모델 학습)’시키고, ‘일(예측) 시키기

## 3-1.ML vs DL?

ML : 특성들이 이런게 있으니 답을 예측하는 모델(y=w1x+w2x+...) 만들어봐.

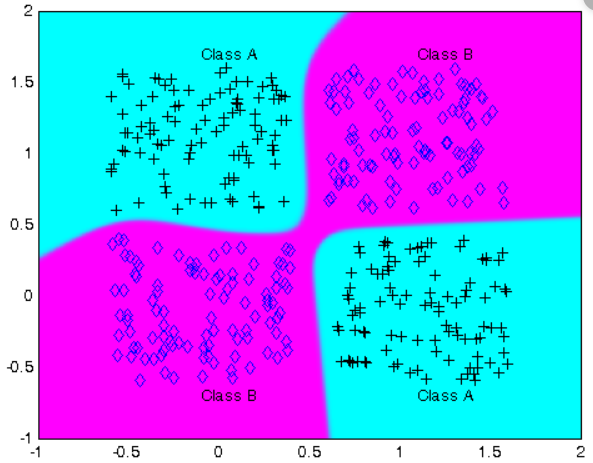
(물론 모든 특성을 다 써서 만들라고 할 수도 있고, 답 예측과 밀접한 연관이 있는 특성들만 골라서 만들어 줄 수도 있다.)

DL : deep 해서 특성이 어떤게 있을진 모르겠다. 특성 갯수(가중치 혹은 필터)는 내가 정해줄게. 예측하는 모델 만들어봐.

## 3-2.Why deep?

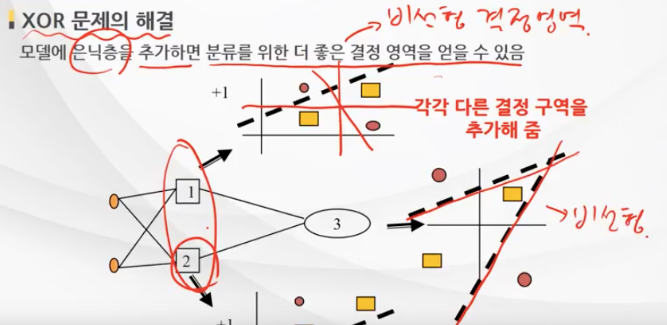
머신러닝으로 모든 분류나 회귀가 되지 않는다.

가령



이런 문제. XOR 문제라고 한다.

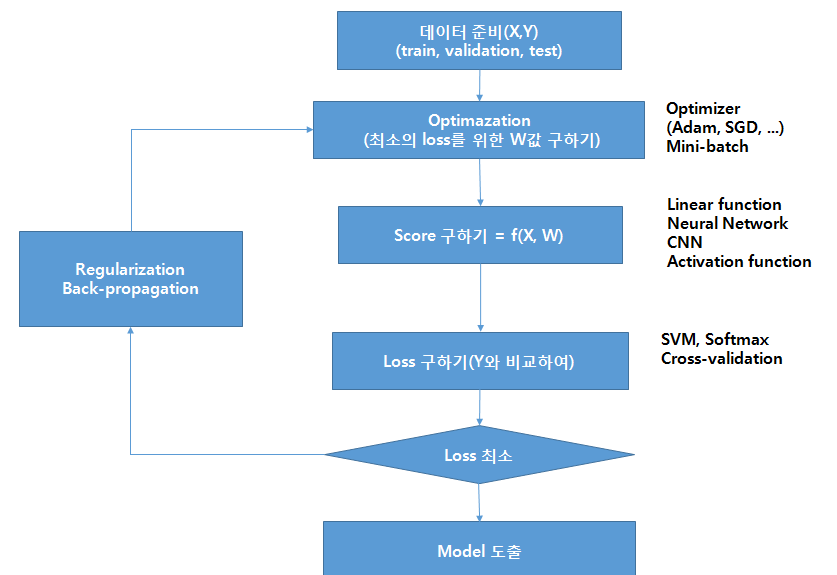




W -> W^2 -> W^3 이 되면서 비 선형적인 형태가 된다.

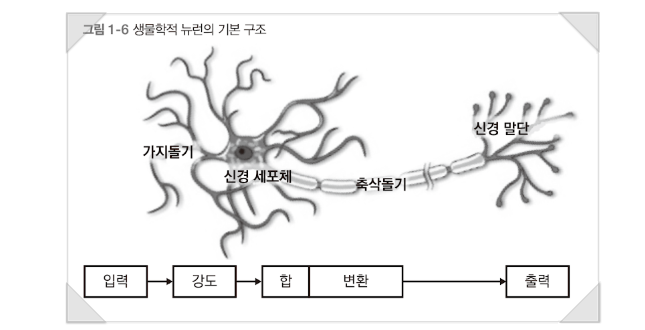
은닉층이 생기면서 우리는 그 모델의 형태를 알 수 없을지라도 분류/회귀를 할 수 있게 된다.

## 3-3. DL 과정

****

## 3-4.DL 모델

### 3-4-1. Perceptron 모델



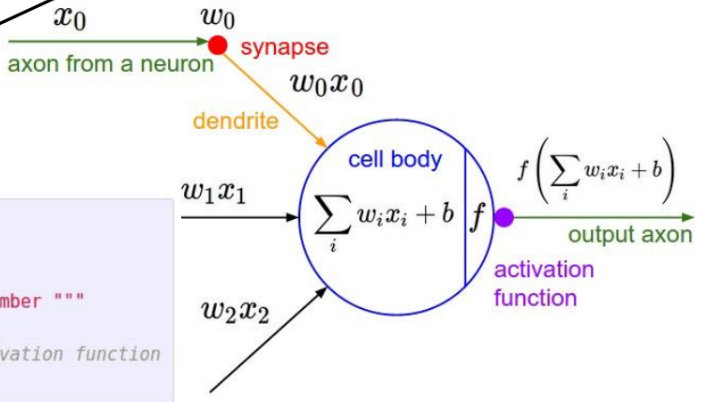
퍼셉트론 모델은 위의 생물학적 뉴런 원리를 가져옴.

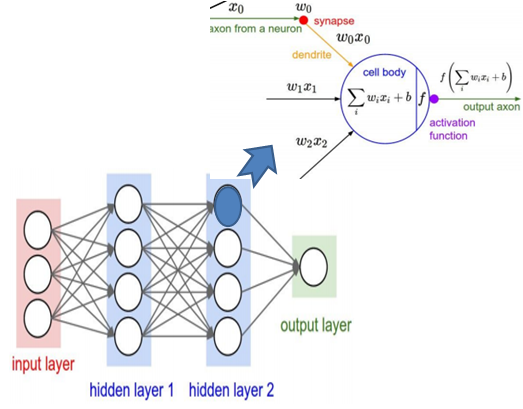
다수의 입력을 받아서 입력의 강도에 따라 다른 가중치를 부여한다.

입력값과 가중치값의 곱을 한다.

다수니까 그 값들을 합한다.

그 결과를 변환하여 임계점 보다 높으면 결과를 이 후에 신경망에 전달하는 방식.





### 3-4-2 CNN

NN이 파라미터가 너무 많음

CNN은 파라미터 수를 줄일 수 있음.

## ML vs DL

ML : 특성들이 이런게 있으니 답을 예측하는 모델(y=w1x+w2x+...) 만들어봐.

(물론 모든 특성을 다 써서 만들라고 할 수도 있고, 답 예측과 밀접한 연관이 있는 특성들만 골라서 만들어 줄 수도 있다.)

DL :

XOR 문제를 해결하기 위해 은닉층을 만들어줌.

그러다 보니 특성이 어떤게 있을진 모르겠음.

근데 특성 갯수(가중치 혹은 필터)는 내가 정해줄게. 예측하는 모델 만들어봐.