분류

- 가중치 벡터 $\underline{w}$을 알고 있다면, 새로운 특징벡터 $\underline{x}$가 주어진 경우 퍼셉트론을 사용해서 출력값을 계산할 수 있음 ⇒ 가중치 벡터 $\underline{w}$를 찾기 위해 사용한 $\underline{x}$가 아닌 새로운 $\underline{x}$

⇒ $O = \tau(\underline{w}\cdot\underline{x})$ 　활성화 함수 　$\left(\begin{array}{l} w^T \cdot w \geq T \Rightarrow 1 \\ \text{otherwise} \Rightarrow -1 \end{array}\right)$ example

- 좌변 : 모르는 값 ⇒ Output
- 우변 : 아는 값 ⇒ Input

⇒ *분류 : $\underline{w}$가 모두 주어진 상황에서 분류 (최종 판단)
↳ $\underline{x}$가 주어지면 계산을 통해 O을 찾음

학습 → $\underline{x}$와 label이 주어짐

- 학습 과정에서는 특징벡터 $\underline{x}$와, 이미 대응하는 O가 주어져 있음 ⇒ $O = \tau(\underline{w}, \underline{x})$ (output (분류된 라벨), 모르는 값, 아는 값)
- 모르는 값 $\underline{w}$를, sample을 제대로 분류하는 최적해로 구하는 것이 학습과정
- 훈련 집합에는 sample의 개수가 아주 많고, 이 모든 sample을 잘 분류하는 $\underline{w}$를 구해야 함
  ↳ model에 따라 완벽히 분류가 가능한 수도 있고, 불가능한 수도 있음
- 최적해를 구하기 위해 최적화 알고리즘을 사용
  ↳ 인공지능의 핵심
  → 인공지능 모델에서 사용하기에 좋다는 수학적 알고리즘을 가져와서 사용 (model, data에 따라 최적화 알고리즘 변화)

PLA (Perceptron Learning Algorithm) ⇒ 적절한 학습, 이해

- 벡터 $\underline{w}$가 어떤 값을 가져야 제대로 분류할 수 있게 되는지 찾아내는 최적화 알고리즘이 필요
- 최적화를 위해서는 최적화 비용함수 (목적함수)가 먼저 정의되어야 함
  ↳ 어떤 조건을 가장 만족하는 값을 찾음
  ↳ ex) 어떤 값을 가질 때 가장 효율적이냐고 물었을 때 모든 값을 하나하나 평가할 수 있는 방법
  ↳ 목적 함수의 입력으로서 가능한 해 (정보, 분류기)를 넣어보아서 결과를 평가해봄

⇒ 최적화 비용함수를 먼저 정의하지 않으면 어떤 것이 좋은 방법론인지 알 수 없음

*낮을수록 최적해에 적합

- * $J(\underline{w}) > 0$ (가중치 (학습대상)) (어떤 $\underline{w}$가 들어오더라도 0보다 큰 값) ⇒ 비용함수가 0보다 크도록 설계
- * $J(\underline{w})$ 값이 최소가 되게 하는 $\underline{w}$를 찾음
- * 분류에 실패하는 sample이 많을수록 $J(\underline{w})$ 값이 커지도록 설계
  성공할수록 $J(\underline{w})$ 값이 작아짐
  ↳ 어떤 방식으로 최적화를 하느냐에 따라 J가 다르게 설계
  ↳ 어떤 문제를 해결하느냐에 따라서 다르게 설계

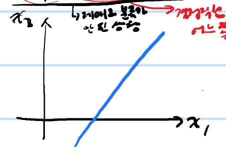
1. Data point $\underline{x}_i$, 대응하는 label y_i ← N개
   - $(\underline{x}_1, y_1), \cdots, (\underline{x}_N, y_N)$
   - $\underline{x}_k = [x_{k0}, x_{k1}, \cdots, x_{kd}]^T$ ⇒ Data point (vector x_k)

2. 비용함수 설계 ⇒ 문제의 목적에 맞게 설계

⇒ $J(\underline{w}) = \sum_{\underline{x}_k \in Y} -y_k(\underline{w}\cdot\underline{x}_k)$

← $\underline{w}$를 사용하여 분류했을 때, 분류에 실패하는 데이터의 집합.

- $-y_k(\underline{w}\cdot\underline{x}_k) > 0$ ⇒ 비용함수가 목적에 맞게 잘 설계 되었는지 확인
  (label; ↳ 제대로 분류가 안 된 상황; → 결정하는 기준, 어느 쪽에 있는지 알려줌)



$\underline{w}\cdot\underline{x}_k > 0$ ⇒ $y_k = 1$ (잘 분류) → $-y_k(\underline{w}\cdot\underline{x}_k) < 0$ (제대로 분류)
　　　　　　　$y_k = -1$ (제대로 분류 X) → $-y_k(\underline{w}\cdot\underline{x}_k) > 0$ (제대로 분류 X)
　↳ y_k와 부호가 같아야 함
$\underline{w}\cdot\underline{x}_k < 0$ ⇒ $y_k = -1$ (잘 분류) → $-y_k(\underline{w}\cdot\underline{x}_k) < 0$ (제대로 분류)
　　　　　　　$y_k = 1$ (제대로 분류 X) → $-y_k(\underline{w}\cdot\underline{x}_k) > 0$ (제대로 분류 X)
　↳ y_k와 부호가 같아야 함

⇒ 틀리는 sample이 많을수록 비용함수가 커짐

3. 비용함수를 가중치 $\underline{w}$에 대해 미분

⇒ $\dfrac{\partial J(\underline{w})}{\partial w_i} = \sum_{\underline{x}_k \in Y} \dfrac{\partial(-y_k(w_0 x_{k0} + \cdots + w_d x_{kd}))}{\partial w_i} = \sum_{\underline{x}_k \in Y} -y_k \cdot x_{ki}$ ⇒ 각각의 w_i에 대해 편미분

$i = 0, 1, \cdots, d$

$\times(-1)$

4. 가중치 벡터를 업데이트

⇒ $w_i \leftarrow w_i + \rho \sum_{\underline{x}_k \in Y} y_k \cdot x_{ki}, \ (i = 0, 1, \cdots, d)$ 　⇒ $\underline{w}$를 결정해줌
　↳ learning rate (constant) → 임의로 설정해줌
　⇒ w_i가 update 될 때마다 달라지기 때문에 ρ를 작게 잡을 때 유리할 수도 있고 크게 잡을 때 유리할 때도 있음 (상황에 따라 다름)

→ 1, 2, 3, 4의 과정을 반복하여 최적해에 수렴. (J가 제일 작은 값)
　↳ 분류 시 틀린 sample의 수가 최소 →
　　↳ (수학적으로 증명이 되어 있음)

- PLA는 훈련집합이 선형분리 가능하다면 반드시 최적해로 수렴.
- 훈련집합이 선형분리 가능하지 않으면 무한 반복 　J가 일정한 이상으로 떨어지지 않음
  ⇒ 데이터가 항상 선형분리 가능하다는 것이 보장되지 않기 때문에 몇 번까지만 해보고 안 되면 중간에 끊어 사용하는 식으로 사용.