배치 PLA (Batch) ⇒ 여러가지 데이터 포인트들에 대한 정보를 한번에 모두 고려하여 파라미터를 결정.

- Input : 훈련집합, 학습률 $(\underline{x}_i, y_i)$, $i \in \{1, 2, \cdots, N\}$, ρ (constant) — ($\underline{x}_i$: feature, y_i: label)
- Output : 최적 가중치 벡터 $\underline{w}^*$, $\hat{\underline{w}}$ 처럼 최적값을 따로 표현하기도 함.

1. 초기값 $\underline{w}$를 설정 ← 무작위 $\underline{w}$ (PLA는 무조건 수렴점으로 수렴하기 때문) (어떤 값이든 이변 상관 없음)
2. repeat (분류 실패한 샘플에 대해서 계속 update)
3. $Y = \phi$ ← 틀린 샘플 집합.
4. for $j = 1$ to N
5. $y = \tau(\underline{w} \cdot \underline{x}_j)$ ← 현재 $\underline{w}$를 사용해 분류
6. if $(y \neq y_j)$, $Y = Y \cup \underline{x}_j$ ← 분류가 틀리면 틀린 샘플 추가 (y: 예측값, y_j: 실제값)
7. if $(Y \neq \phi)$ ← 틀린 샘플이 존재
8. for $i = 0$ to d ← w vector의 길이 또는 Feature Vector의 길이
9. $w_i = w_i + \rho \sum_{\underline{x}_k \in Y} y_k x_{ki}$
10. until $(Y = \phi)$ ← 틀린 샘플이 없을 때까지 update (무한 loop를 방지하기 위해 limit를 줌)
11. $\underline{w}^* = \underline{w}$ (상황에 따라 값이 달라짐)

(3–6: 모든 data를 검사해서 ⇒ 7–9: w 업데이트)

* 틀린 샘플들을 모아서 한꺼번에 업데이트함.

스토캐스틱 PLA (Stochastic) ⇒ Random의 느낌을 갖고 있음.

- 샘플 순서를 무작위로 섞음
- 틀린 샘플을 발견하면, 즉시 업데이트함. (샘플 하나씩 확인함) (→ 확인이 되면 바로 업데이트)

1. 초기해 $\underline{w}$를 설정함.
2. repeat.
3. $\underline{X} = \{\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_N\}$의 샘플 순서를 섞음 (매번 섞어줌)
4. quit = true ← 종료 flag 초기화
5. for $j = 1$ to N
6. $y = \tau(\underline{w} \cdot \underline{x}_j)$
7. if $(y \neq y_j)$
8. quit = false
9. for $i = 0$ to d
10. $w_i = w_i + \rho y_j x_{ji}$

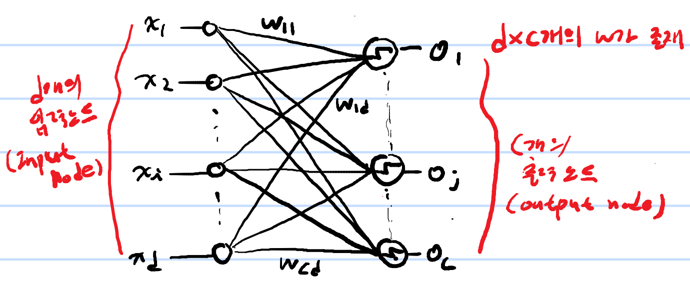
(7–10 ⇒ 분류가 틀린 것을 발견하면 w를 update)

11. until(quit)
12. $\underline{w}^* = \underline{w}$.

* mini Batch
⇒ N개의 data를 각기 작게 나누어 각각 나누어 주는 w에 대해 stochastic
→ Model과 목적에 따라 Batch 방식 선택

퍼셉트론을 사용한 다중분류

- 여러 개의 퍼셉트론 묶기



- 출력이 여러 개인 퍼셉트론 묶음의 가중치는 가중치 벡터의 묶음이 됨.
- j번째 퍼셉트론의 가중치 벡터를 $\underline{w}_j = (w_{j1}, \cdots, w_{jd})^T$로 표현
- 출력벡터는 $\underline{o} = (o_1, o_2, \cdots, o_c)^T$임.
- 벡터함수 $\tau(\cdot)$는 입력받은 벡터의 요소 각각에 활성화함수를 적용 (→ Input: vector, Output: vector, → 활성값)

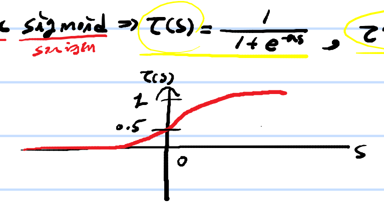
$$\Rightarrow \underline{o} = \tau\begin{pmatrix} \underline{w}_1 \cdot \underline{x} \\ \vdots \\ \underline{w}_c \cdot \underline{x} \end{pmatrix} = \tau(\overline{W} \cdot \underline{x}) \qquad \overline{W} = \begin{pmatrix} \underline{w}_1^T \\ \vdots \\ \underline{w}_c^T \end{pmatrix} \neq (\overline{w}_1 \cdots \overline{w}_c)^T$$

($\underline{w}_1$: 열벡터, $\underline{w}_1^T$: 행벡터)

- 출력이 c개인 퍼셉트론 묶음은 c개의 카테고리를 갖는 다중분류문제에 적용가능
  - o_1은 특징벡터 $\underline{x}$가 1번째 카테고리에 속하는지 여부를 판단
  - o_i는 특징벡터 $\underline{x}$가 i번째 카테고리에 속하는지 여부를 판단.
  - 출력벡터 $\underline{o}$에서, 여러 카테고리에 속하는 것으로 분류될 수 있음 ← 다중 레이블 다중분류 분류 (→ vector, → vector)
  - 단 하나의 카테고리에만 속하는 분류문제는?
    → 다른 활성화함수와 다른 최적화 알고리즘을 적용. (→ step func이 아닌 다른 func) (→ PLA가 아닌 다른 알고리즘)

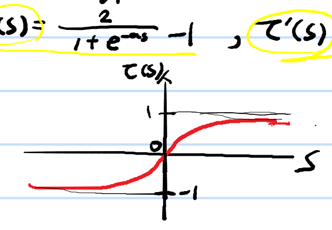
활성화함수 example

- 계단 함수 (Step func.) ⇒ $\tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases}$, $\tau'(s) = \begin{cases} 0 & s \neq 0 \\ \text{불가능} & s = 0 \end{cases}$ (정의 ×) ⇒ 미분 불가능
- Logistic sigmoid (logistic, s자 형태) ⇒ $\tau(s) = \frac{1}{1 + e^{-as}}$, $\tau'(s) = a\tau(s)(1 - \tau(s))$ 치역: $\tau(s) \in (0, 1)$

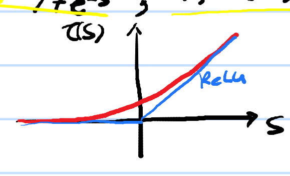


⇒ 확률 예측에서 주로 사용 (→ 확률을 표현하기도 함) (0~1 사이의 값이 결과)

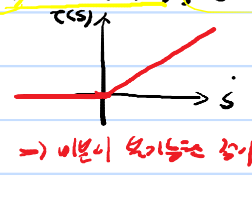
- 하이퍼볼릭 탄젠트 (Hyperbolic tangent)
⇒ $\tau(s) = \frac{2}{1 + e^{-as}} - 1$, $\tau'(s) = \frac{a}{2}(1 - \tau(s)^2)$, 치역 $= \tau(s) \in (-1, 1)$



- 소프트플러스 (softplus) (정확도 증가)
⇒ $\tau(s) = \log_e(1 + e^s)$, $\tau'(s) = \frac{1}{1 + e^{-s}}$, 치역 $= \tau(s) \in (0, \infty)$
ex) 큰 양수와 비슷...



- 렉티파이어 (ReLu) (연산속도 증가)
⇒ $\tau(s) = \max(0, s)$, $\tau'(s) = \begin{cases} 0 & s < 0 \\ 1 & s > 0 \\ \text{불가능} & s = 0 \end{cases}$ ⇒ 미분불가능, 치역 $= \tau(s) \in [0, \infty)$



(소프트플러스와 매우 흡사)
⇒ 미분이 불가능한 점이 있기 때문에 소프트플러스로 완화하여 사용.

* Logistic sigmoid를 활성화함수로 사용하는 퍼셉트론은 각 카테고리에 속할 확률을 추출하는 모델로 볼 수 있음
  → 더 정확한 의사결정 추론이 가능 (output이 확률) (→ model을 사용하여 Input을 넣고 output을 받는 것)
  → 확률이 가장 높은 카테고리에 속하는 것으로 판단 ⇒ 최적화 알고리즘은 model과 활성화함수에 따라 잘 선택해야 함.