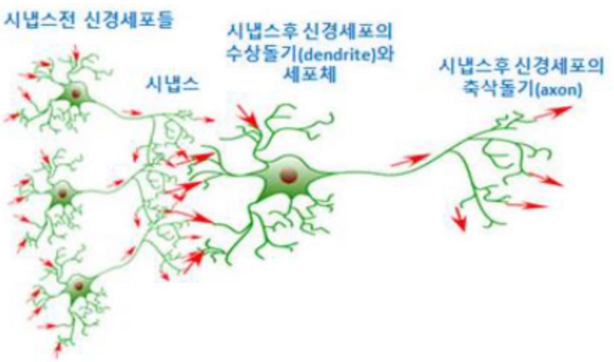
다층 퍼셉트론

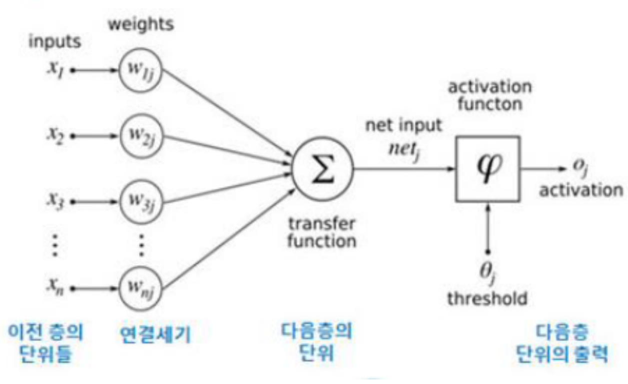
신경망 기초

- 신경망
    - 기계학습 역사에서 가장 오래된 기계학습모델, 현재 가장 다양한 형태를 가짐
    - 1950년대 퍼셉트론(인공두뇌학) ▶ 1980년대 다층 퍼셉트론 (결합설)
    - 딥러닝의 기초

인공신경망과 생물신경망

- 두 줄기 연구의 시너지 효과
    - 컴퓨터 과학
        - 계산 능력의 획기적 발전으로 지능 처리에 대한 욕구 확대
    - 뇌 과학
        - 뇌의 정보처리 방식 연구
    - **뇌**의 정보러치 **모방**하여 사람의 지능 처리할 수 있는 인공지능 도전
        - 뉴런의 동작 이해를 모방한 **인공 신경망(ANN)** 연구 수행됨
        - **퍼셉트론** 고안
- 사람의 신경망과 인공신경망 비교





사람 신경망	인공 신경망
세포체	노드
수상돌기	입력
축삭	출력
시냅스	가중치

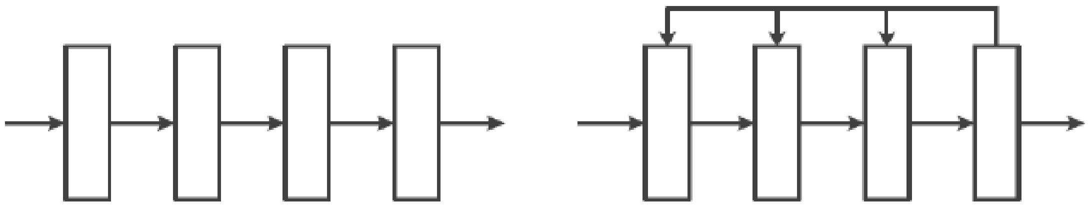
신경망의 간략한 역사

- 1943년 매컬럭과 피츠의 최초의 신경망
- 1949년 헤브는 최초로 학습 알고리즘 제안
- **1958년** 로젠블렛은 **퍼셉트론** 제안
- 위드로와 호프의 Adaline과 Madaline

- **1960년대**의 **과대평가**
- **1969년** 민스키와 페퍼트의 저서 Perceptrons는 **퍼셉트론의 한계를 수학적으로 입증**
  - 선형분류기에 불과하여, **XOR 문제조차 해결 못함**
  - 신경망 연구 **퇴조**
- **1986년** 루멜하트의 저서 Parallel Distributed Processing은 **다층 퍼셉트론** 제안
  - 신경망 연구 **부활**
- 1990년대 SVM에 밀리는 형국
- 2000년대 **딥러닝**이 실현되어 신경망이 기계 학습의 **주류 기술**로 자리매김

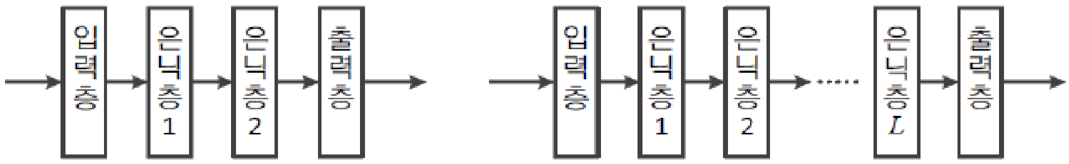
신경망의 종류

- **전방 신경망**과 **순환 신경망**



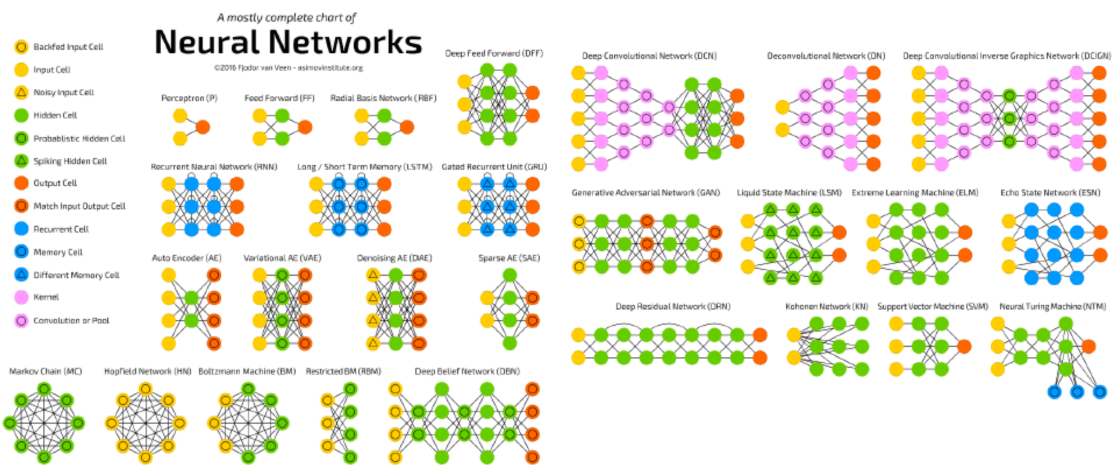
(a) 전방 신경망과 순환 신경망

- **얕은 신경망**과 **깊은 신경망**



(b) 얕은 신경망과 깊은 신경망

- **결정론 신경망**과 **스토캐스틱 신경망**
  - 결정론 신경망 : 모델의 매개변수와 조건에 의해 출력이 완전히 결정되는 신경망
  - 스토캐스틱 신경망 :
    고유의 임의성을 가지고 매개변수와 조건이 같더라도 다른 출력을 가지는 신경망
- 다양한 종류

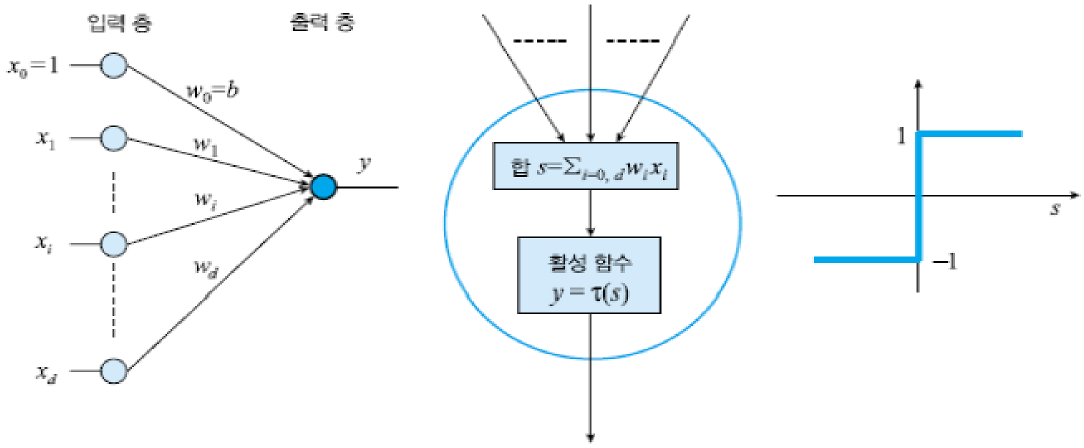


퍼셉트론

- 퍼셉트론은 **노드, 가중치, 층**과 같은 새로운 개념을 도입하고 **학습 알고리즘**을 창안함
- 원시적 신경망이지만, 딥러닝을 포함한 현대 신경망은 퍼셉트론을 병렬과 순차구조로 결합하여 만듬 ▶ 현대 신경망의 중요한 구성 요소

구조

- 입력층과 출력층을 가짐
    - 입력층은 연산을 하지 않으므로 퍼셉트론은 **단일 층 구조**라고 간주
- **입력층**의 i번째 노드는 특징 벡터 $\mathbf{x} = (x_1, x_2, \cdots, x_d)^\mathbf{T}$ 요소를 담당
- 항상 1이 입력되는 바이어스 노드
- **출력층**은 한 개의 노드
- i번째 입력 노드와 출력 노드를 연결하는 변은 가중치 w_i를 가짐



(a) 전체 구조 (b) 출력 노드의 연산 (c) 활성 함수

동작

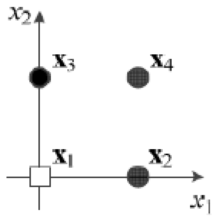
- 해당하는 입력값과 가중치를 곱한 결과를 모두 더하여 s를 구하고, 활성함수 τ를 적용함
- 활성함수 τ로 계단함수를 사용하므로 최종 출력 y는 +1 또는 -1

$$\left.\begin{array}{l} y = \tau(s) \\ \text{이때 } s = w_0 + \sum_{i=1}^{d} w_i x_i, \qquad \tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases} \end{array}\right\} \tag{3.1}$$

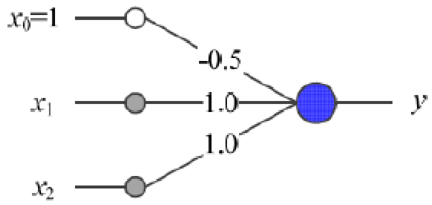
예제 3-1 **퍼셉트론의 동작**

2차원 특징 벡터로 표현되는 샘플을 4개 가진 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$, $\mathbb{Y} = \{y_1, y_2, y_3, y_4\}$를 생각하자. [그림 3-4(a)]는 이 데이터를 보여준다.

$$\mathbf{x}_1 = \begin{pmatrix}0\\0\end{pmatrix},\ y_1 = -1,\ \ \mathbf{x}_2 = \begin{pmatrix}1\\0\end{pmatrix},\ y_2 = 1,\ \ \mathbf{x}_3 = \begin{pmatrix}0\\1\end{pmatrix},\ y_3 = 1,\ \ \mathbf{x}_4 = \begin{pmatrix}1\\1\end{pmatrix},\ y_4 = 1$$



(a) 훈련집합



(b) 퍼셉트론

그림 3-4 **OR 논리 게이트를 이용한 퍼셉트론의 동작 예시**

샘플 4개를 하나씩 입력하여 제대로 분류하는지 확인해 보자.

$$\begin{aligned}
&\mathbf{x}_1\!: s = -0.5 + 0 * 1.0 + 0 * 1.0 = -0.5, && \tau(-0.5) = -1\\
&\mathbf{x}_2\!: s = -0.5 + 1 * 1.0 + 0 * 1.0 = 0.5, && \tau(0.5) = 1\\
&\mathbf{x}_3\!: s = -0.5 + 0 * 1.0 + 1 * 1.0 = 0.5, && \tau(0.5) = 1\\
&\mathbf{x}_4\!: s = -0.5 + 1 * 1.0 + 1 * 1.0 = 1.5, && \tau(1.5) = 1
\end{aligned}$$

결국 [그림 3-4(b)]의 퍼셉트론은 샘플 4개를 모두 맞추었다. 이 퍼셉트론은 훈련집합을 100% 성능으로 분류한다고 말할 수 있다.

- 행렬 표기(Matrix Vector Notation)

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0, \qquad \text{여기서 } \mathbf{x} = (x_1, x_2, \cdots, x_d)^{\mathrm{T}},\ \mathbf{w} = (w_1, w_2, \cdots, w_d)^{\mathrm{T}} \tag{3.2}$$

  - 바이어스 항을 벡터에 추가하면,

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x}, \qquad \text{여기서 } \mathbf{x} = (1, x_1, x_2, \cdots, x_d)^{\mathrm{T}},\ \mathbf{w} = (w_0, w_1, w_2, \cdots, w_d)^{\mathrm{T}} \tag{3.3}$$

  - 퍼셉트론의 동작을 식 (3.4)로 표현할 수 있음

$$y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \tag{3.4}$$

- 그림 3-4(b)를 기하학적으로 설명하면
  - 결정 직선 $d(\mathbf{x}) = d(x_1, x_2) = w_1x_1 + w_2x_2 + w_0 = 0 \rightarrow x_1 + x_2 - 0.5 = 0$
    - w1과 w2는 직선의 방향, w0은 절편을 결정
    - 결정 직선은 전체 공간을 +1과 -1의 두 부분공간으로 분할하는 **분류기** 역할

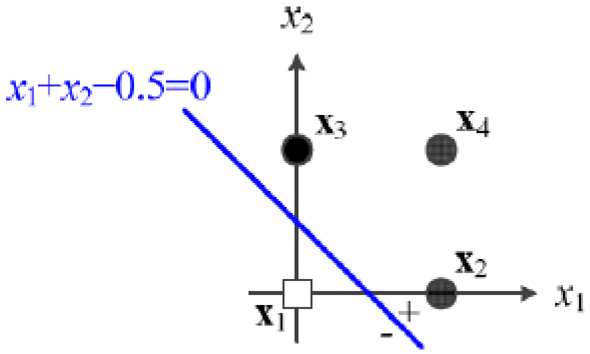
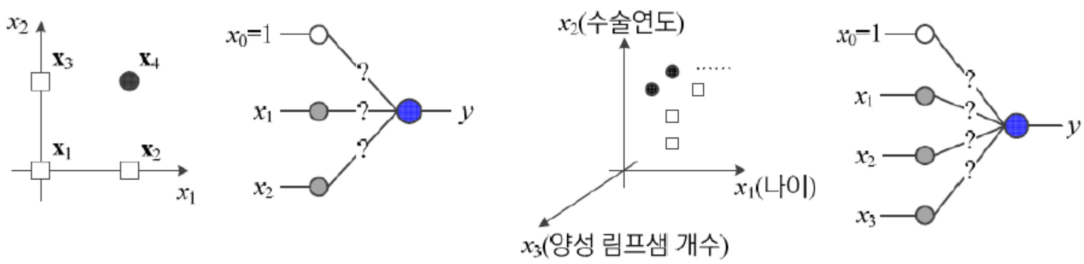


그림 3-5 [그림 3-4(b)]의 퍼셉트론에 해당하는 결정 직선

- d차원 공간에서는 $d(\mathbf{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + w_0 = 0$
    - 2차원은 결정직선, 3차원은 결정 평면, 4차원 이상은 결정 초평면

학습

- 학습 문제
    - 지금까지는 **학습을 마친 퍼셉트론**을 가지고 **동작**을 설명한 셈
    - 그림 3-6은 **학습 문제**: w1과 w2, w0이 어떤 값을 가져야 100% 옳게 분류할까?
    - 그림 3-6은 2차원 공간에 4개 샘플이 있는 훈련집합이지만,
      현실 세계는 d차원 공간에 수백 ~ 수만 개의 샘플이 존재



(a) AND 분류 문제

(b) Haberman survival 분류 문제
UCI 데이터 (유방암 수술 생존 관련 데이터)

그림 3-6 어떻게 학습시킬 것인가?

-
    - 일반적인 분류기의 **학습 수행 과정**
        - 단계 1 : **분류기의 정의**와 분류 과정의 수학적 정의
        - 단계 2 : 해당 분류기의 **목적함수 J(세타) 정의**
        - 단계 3 : J(세타)를 최소화하는 세타값을 찾기 위한 **최적화** 방법 수행
- 목적함수 설계 (단계 1과 단계 2)
    - 퍼셉트론의 매개변수를 $\mathbf{w} = (w_0, w_1, w_2, \cdots, w_d)^\mathrm{T}$라 표기하면, **매개변수 집합**은
      $\Theta = \{\mathbf{w}\}$
    - 목적함수를 $J(\Theta)$ 또는 $J(\mathbf{w})$로 표기함
    - **목적함수의 조건**

      $J(\mathbf{w}) \geq 0$이다. (1)

      $\mathbf{w}$가 최적이면, 즉 모든 샘플을 맞히면 $J(\mathbf{w}) = 0$이다. (2)

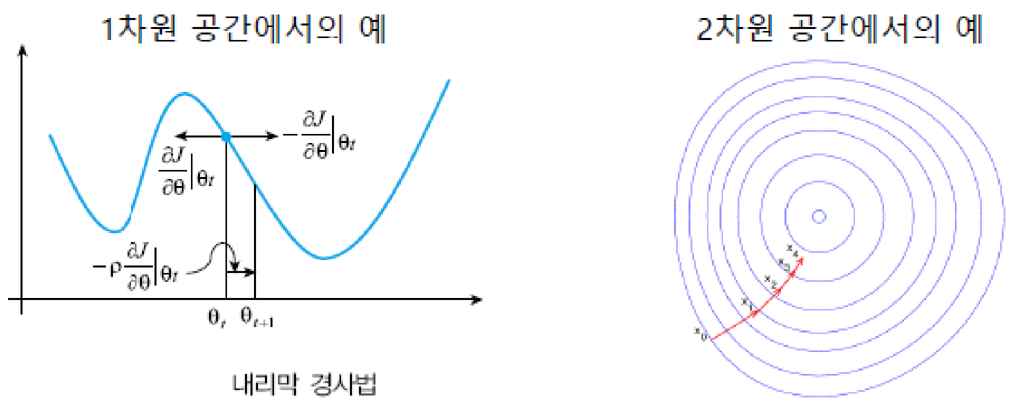
      틀리는 샘플이 많은 $\mathbf{w}$일수록 $J(\mathbf{w})$는 큰 값을 가진다. (3)
    - 식 (3.7)은 세가지 조건을 만족하므로, 퍼셉트론의 목적함수로 적합
        - y는 w가 틀리는 샘플의 집합

$$J(\mathbf{w}) = \sum_{\mathbf{x}_k \in Y} -y_k \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_k \right) \qquad (3.7)$$

- 조건 (1), (2), (3) 을 만족
    - 임의의 샘플 x_k가 Y에 속한다면, 퍼셉트론의 예측값 $\boldsymbol{w}^{\mathbf{T}}\mathbf{x}_{\boldsymbol{k}}$와 실제값 $\boldsymbol{y}_{\boldsymbol{k}}$는 부호가 다름

      → $-\boldsymbol{y}_{\boldsymbol{k}}(\boldsymbol{w}^{\mathbf{T}}\mathbf{x}_{\boldsymbol{k}})$는 항상 양수를 가짐: 조건(1) 만족
    - 결국 Y가 클수록(틀린 샘플이 많을수록), J(w)는 큰 값을 가짐 : **조건 (3) 만족**
    - Y가 공집합일 때(퍼셉트론이 모든 샘플을 맞출 때), J(w) = 0임 : **조건 (2) 만족**

- **경사 하강법**(3단계)
    - 최소 **J(세타) 기ㅇ울기**를 이용하여 **반복 탐색**하여 **극값**을 찾음



내리막 경사법

- 그레이디언트 계산
    - 식 (2.58)의 **가중치 갱신 규칙** $\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho\mathbf{g}$를 적용하려면 그레이디언트 g가 필요
    - 식 (3.7)을 **편미분**하면

$$\frac{\partial J(\mathbf{w})}{\partial w_i} = \sum_{\mathbf{x}_k \in Y} \frac{\partial(-y_k(w_0 x_{k0} + w_1 x_{k1} + \cdots + w_i x_{ki} + \cdots + w_d x_{kd}))}{\partial w_i} = \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}$$

$$\frac{\partial J(\mathbf{w})}{\partial w_i} = \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}, \qquad i = 0, 1, \cdots, d \qquad (3.8)$$

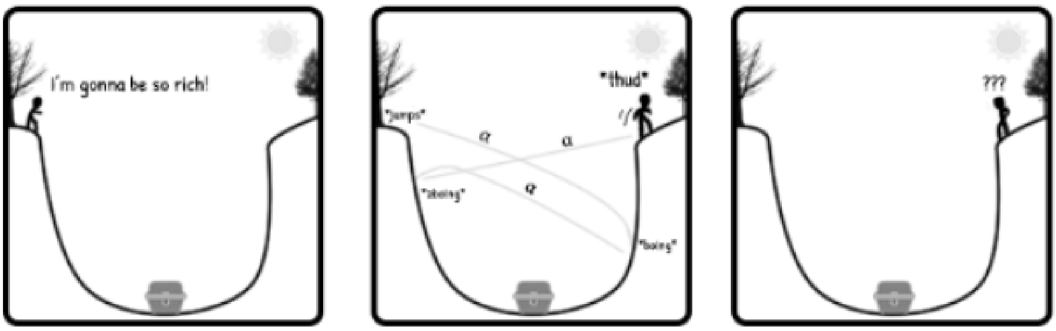
  - x_{ki}는 $\mathbf{x}_k$의 i번째 요소임
  - $\mathbf{x}_k = (x_{k0}, x_{k1}, \ldots, x_{kd})^{\mathrm{T}}$

- 편미분 결과인 식 (3.8)을 식 (2.58)에 대입하면

$$\text{델타 규칙:} \quad w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki}, \qquad i = 0, 1, \cdots, d \qquad (3.9)$$

  - 델타 규칙은 퍼셉트론 학습 규칙

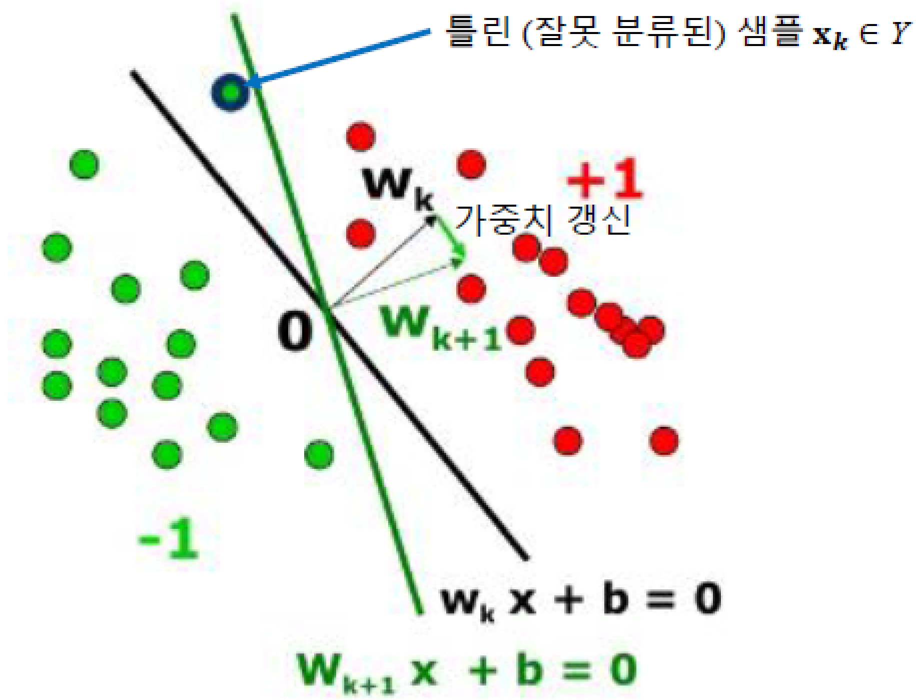
- **학습률**의 중요성



α too big

α too small

- 퍼셉트론 학습 알고리즘 **동작**



  - 식 (3.9)를 이용하면 학습 알고리즘을 쓰면
    - 훈련집합의 샘플을 모두 맞출 때까지 **세대**를 반복함

알고리즘 3-1 퍼셉트론 학습(배치 버전)

입력: 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력: 최적 가중치 $\hat{\mathbf{w}}$

1 난수를 생성하여 초기해 $\mathbf{w}$를 설정한다.
2 repeat
3 　$Y = \emptyset$ 　// 틀린 샘플 집합
4 　for j=1 to n
5 　　$y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j)$ 　// 식 (3.4)
6 　　if($y \neq y_j$) $Y = Y \cup \mathbf{x}_j$ 　// 틀린 샘플을 집합에 추가한다.
7 　if($Y \neq \emptyset$)
8 　　for i=0 to d 　// 식 (3.9)
9 　　　$w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki}$
10 until ($Y = \emptyset$)
11 $\hat{\mathbf{w}} = \mathbf{w}$

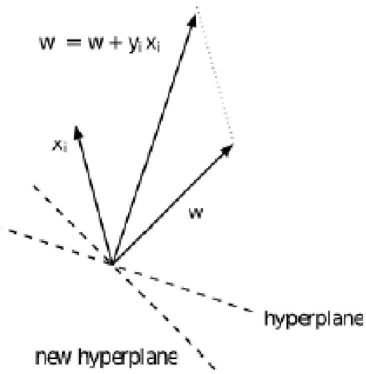
- 퍼셉트론 학습 알고리즘의 스토캐스틱 형태
  - 샘플 순서를 섞음, 틀린 샘플이 발생하면 즉시 갱신

알고리즘 3-2 퍼셉트론 학습(스토캐스틱 버전)

입력: 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력: 최적 가중치 $\hat{\mathbf{w}}$

난수를 생성하여 초기해 $\mathbf{w}$을 설정한다.
repeat
　$\mathbb{X}$의 샘플 순서를 섞는다.
　quit=true
　for j=1 to n
　　$y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j)$ 　// 식 (3.4)
　　if($y \neq y_j$)
　　　quit=false
　　　for i=0 to d
　　　　$\boxed{w_i = w_i + \rho y_j x_{ji}}$
until(quit) 　// 틀린 샘플이 없을 때까지
$\hat{\mathbf{w}} = \mathbf{w}$

갱신의 기하학적 의미



- 행렬 표기
  - 행렬을 사용하여 간결하게 표기 : 델타 규칙: $\mathbf{w} = \mathbf{w} + \rho \sum_{\mathbf{x}_k \in Y} y_k \mathbf{x}_k$
  - 행렬 표기로 [알고리즘 3-1]을 수정하면

$$\left.\begin{array}{l} 8.\ \ \text{for } i = 0 \text{ to } d \\ 9.\ \ \quad w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki} \end{array}\right\} \rightarrow \ 8.\ \ \mathbf{w} = \mathbf{w} + \rho \sum_{\mathbf{x}_k \in Y} y_k \mathbf{x}_k$$

  - 행렬 표기로 [알고리즘 3-2]를 수정하면

$$\left.\begin{array}{ll} 9. & \text{for } i = 0 \text{ to } d \\ 10. & \quad w_i = w_i + \rho y_j x_{ji} \end{array}\right\} \rightarrow \; 9.\;\; \mathbf{w} = \mathbf{w} + \rho y_j \mathbf{x}_j$$

- 선형분리 불가능한 경우에는 무한 반복

  until($Y = \emptyset$) 또는 until(quit)를 until(더 이상 개선이 없다면)으로 수정해야 함

- 학습 예제

  $\mathbf{w}(0)=(-0.5,0.75)^T$, $b(0)=0.375$ t_a: 샘플 a의 목표치 (실제값 == y_a)

  ① $d(\mathbf{x}) = -0.5x_1 + 0.75x_2 + 0.375$
  $Y=\{\mathbf{a}, \mathbf{b}\}$

  $$\mathbf{w}(1) = \mathbf{w}(0) + 0.4(t_a \cdot \mathbf{a} + t_b \cdot \mathbf{b}) = \begin{pmatrix} -0.5 \\ 0.75 \end{pmatrix} + 0.4\left[-\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right] = \begin{pmatrix} -0.1 \\ 0.75 \end{pmatrix}$$

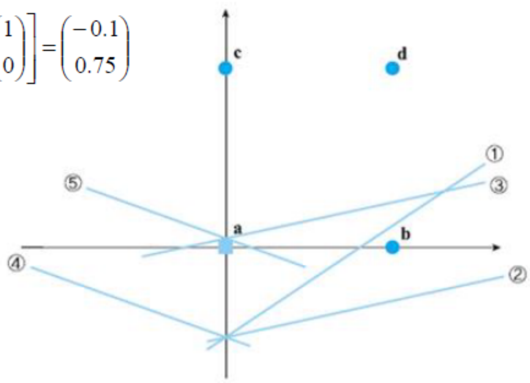
  $$b(1) = b(0) + 0.4(t_a + t_b) = 0.375 + 0.4 * 0 = 0.375$$

  ② $d(\mathbf{x}) = -0.1x_1 + 0.75x_2 + 0.375$
  $Y=\{\mathbf{a}\}$

  $$\mathbf{w}(2) = \mathbf{w}(1) + 0.4(t_a \mathbf{a}) = \begin{pmatrix} -0.1 \\ 0.75 \end{pmatrix} + 0.4\left[-\begin{pmatrix} 0 \\ 0 \end{pmatrix}\right] = \begin{pmatrix} -0.1 \\ 0.75 \end{pmatrix}$$

  $$b(2) = b(1) + 0.4(t_a) = 0.375 - 0.4 = -0.025$$

  ⋮



퍼셉트론 학습 과정의 시각화

- 퍼셉트론 학습 동작 예제