딥러닝 몸풀기

누구나 이해할 수 있는 딥러닝

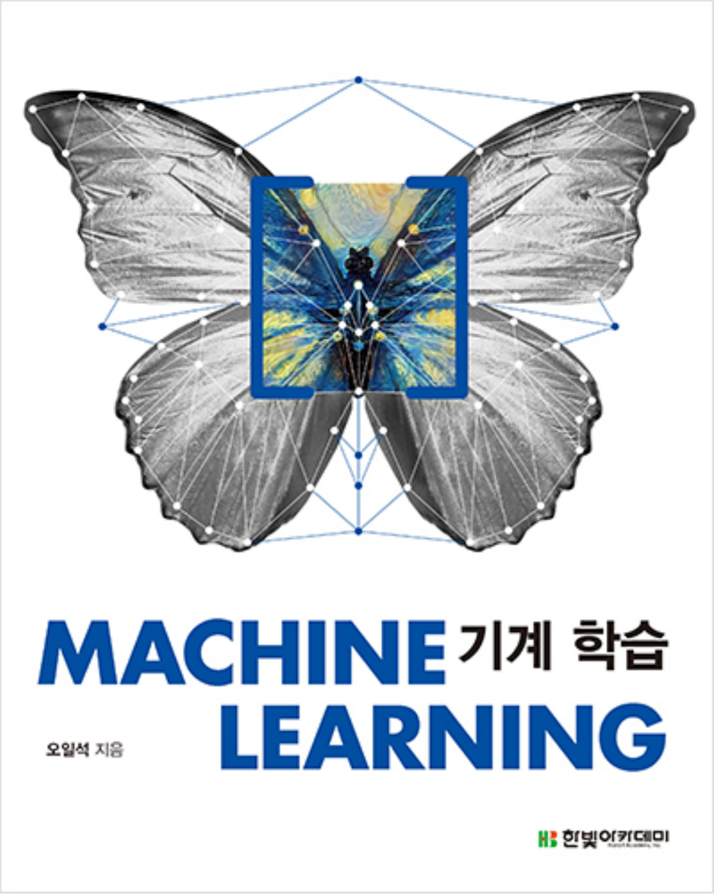
Local Laboratory

딥러닝 몸풀기

Local Laboratory

1. 인공지능과 기계학습 그리고 딥러닝
2. 다층 퍼셉트론
3. 딥러닝의 기초

Reference



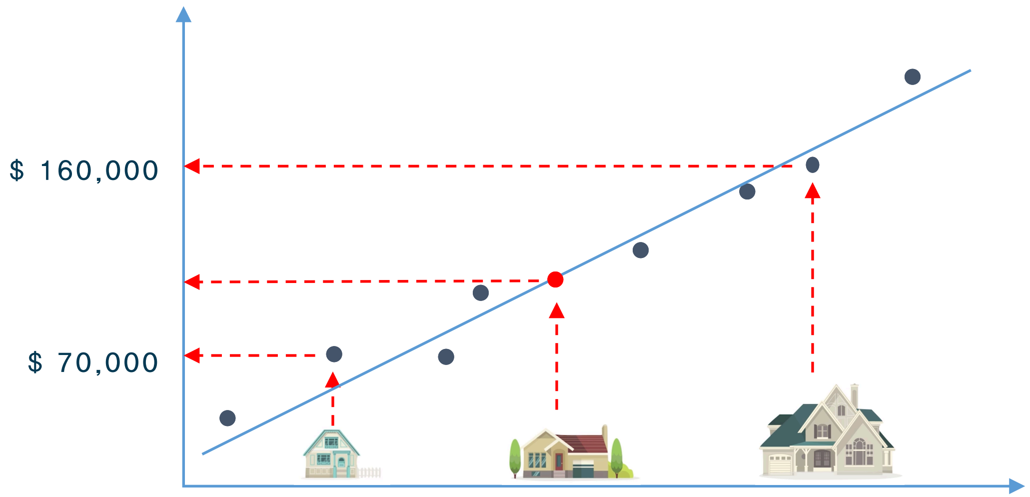
기계학습, 오일석

UDACITY

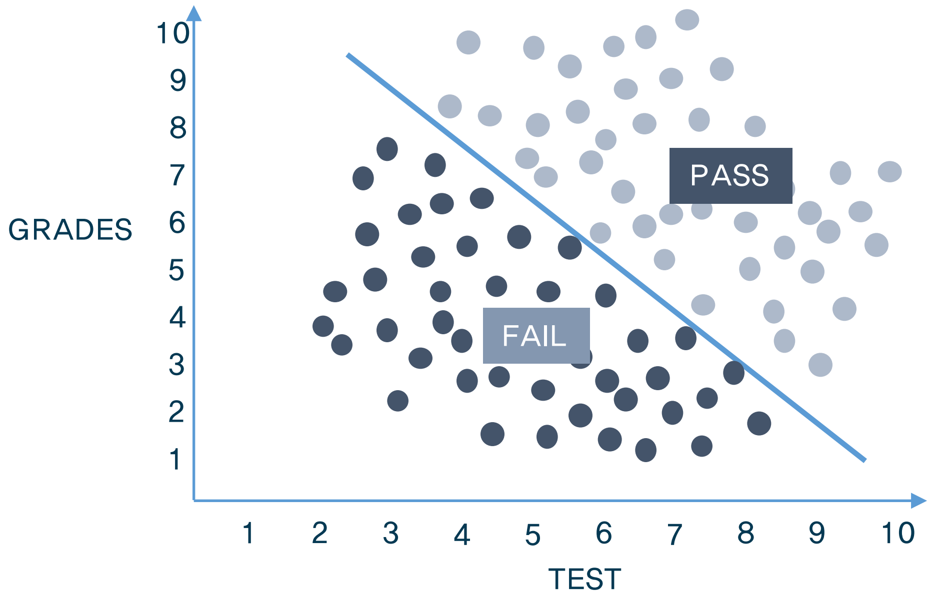
Self-Driving Car Nano degree, Udacity

회귀와 분류 Regression, Classification

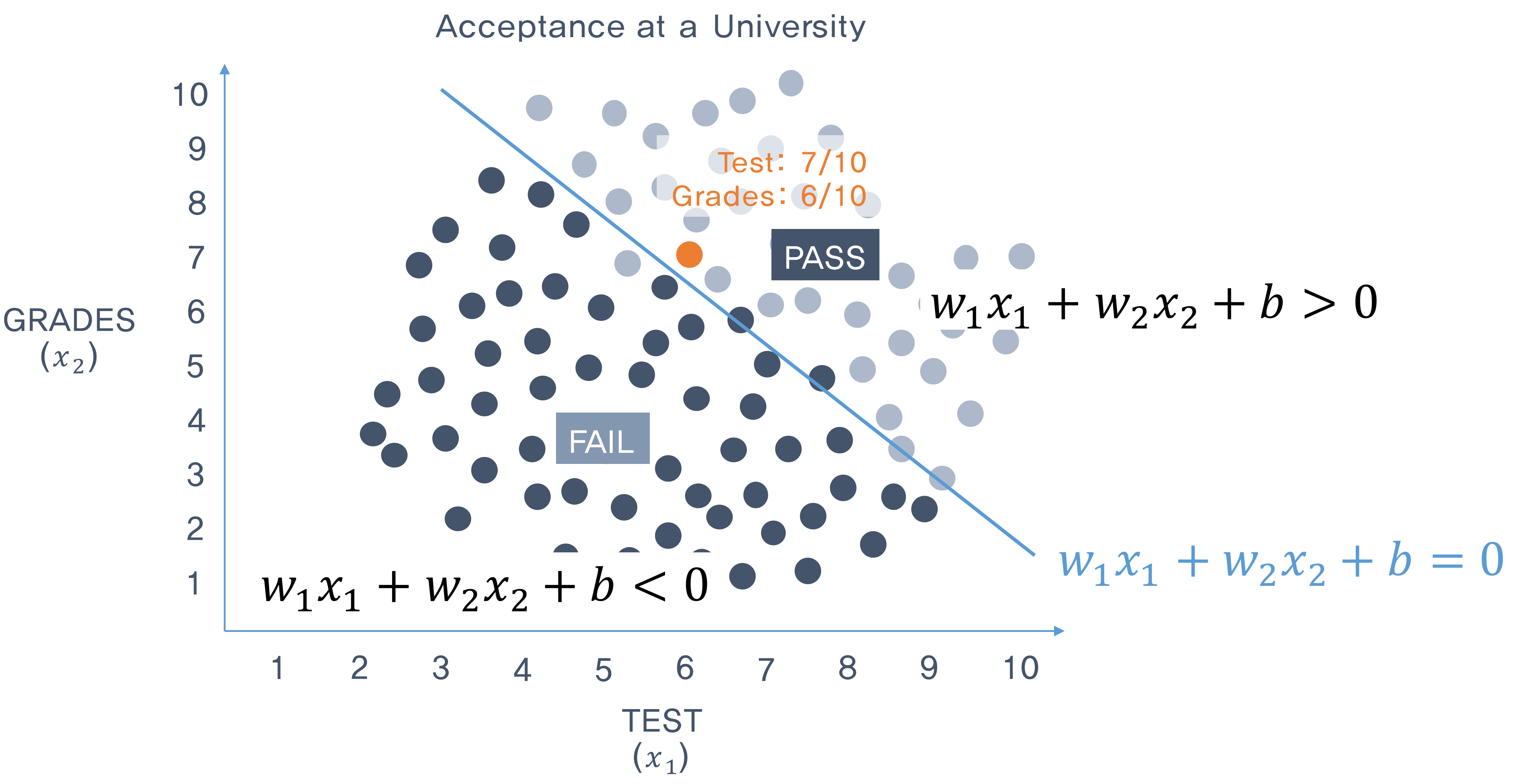
Regression



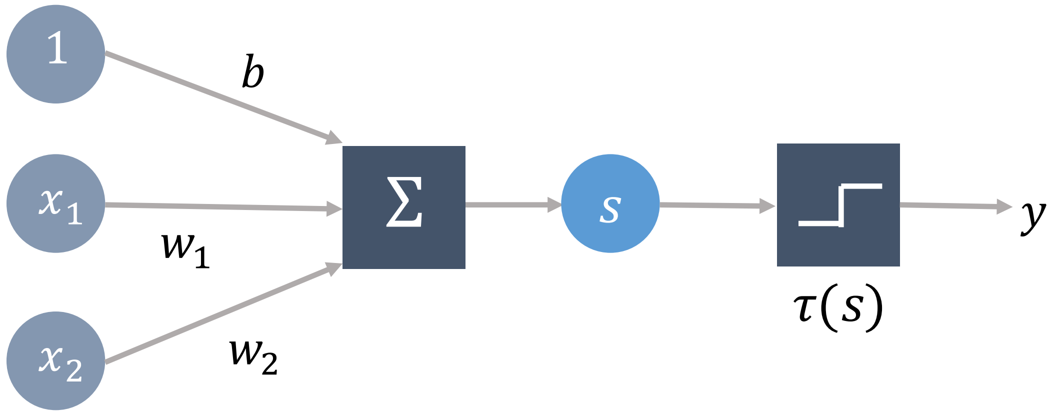
Classification

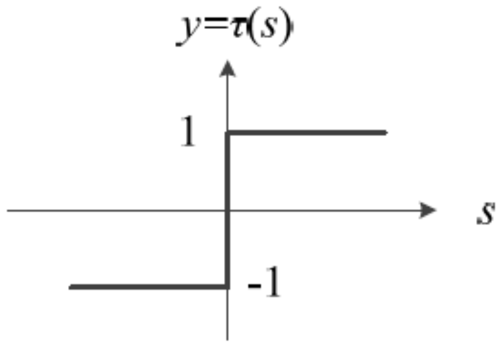


분류 Logistic Regression



분류 Logistic Regression





(b) 계단함수를 활성함수 $\tau(s)$로 이용함

활성함수 Activation function

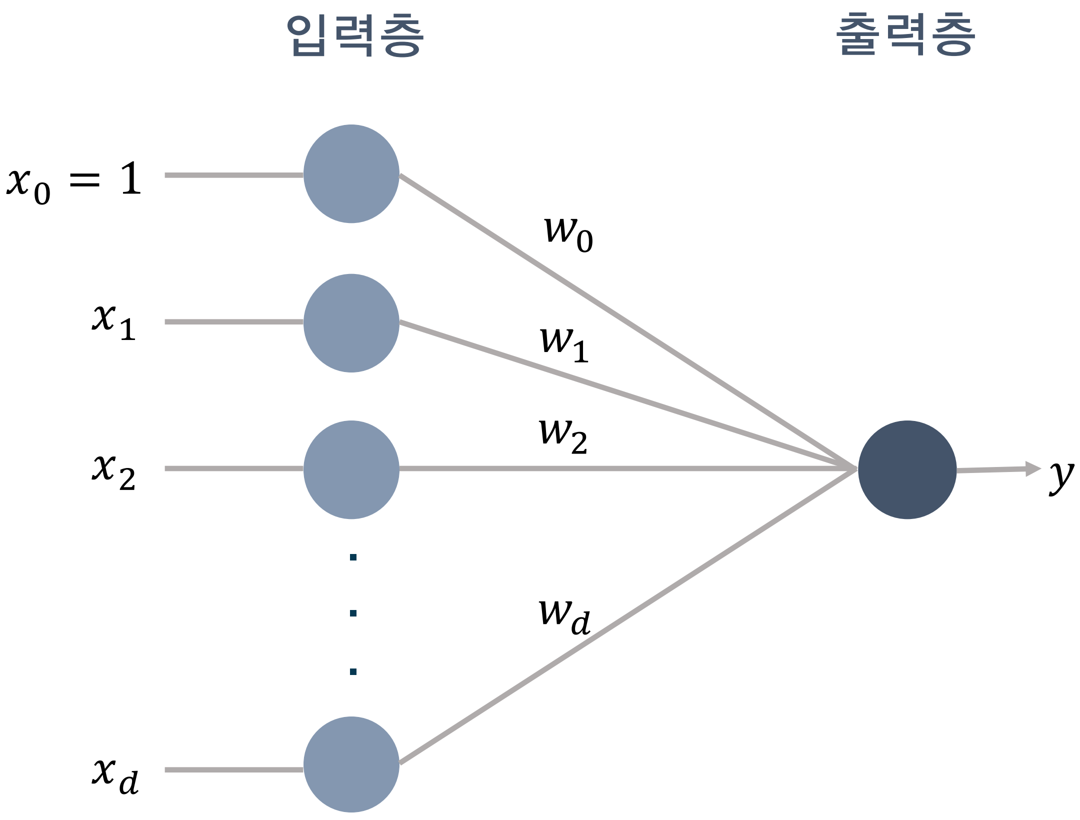
$$s = w_1 x_1 + w_2 x_2 + b$$

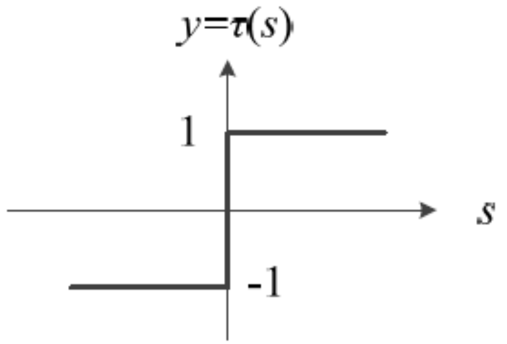
$$s = b + \sum_{i=1}^{2} w_i x_i$$

$$y = \tau(s)\,,$$

$$\tau(s) = \begin{cases} 1 & if\ s \geq 0 \\ -1 & if\ s < 0 \end{cases}$$

퍼셉트론 Perceptron





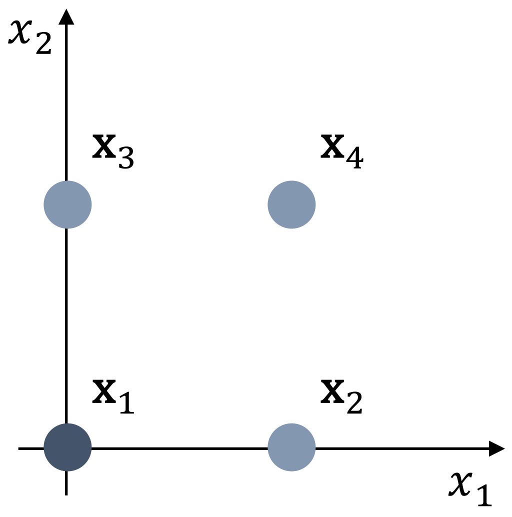
(b) 계단함수를 활성함수 $\tau(s)$로 이용함

$$s = w_0 + \sum_{i=1}^{d} w_i x_i$$

$$y = \tau(s)$$

$$\tau(s) = \begin{cases} 1 & if\ s \geq 0 \\ -1 & if\ s < 0 \end{cases}$$

퍼셉트론 Perceptron



직선 모델

$$f_\Theta(\mathbf{x}) = w_1 x_1 + w_2 x_2 + b$$

추정해야할 매개변수

$$\Theta = (w_1, w_2, b)^T$$

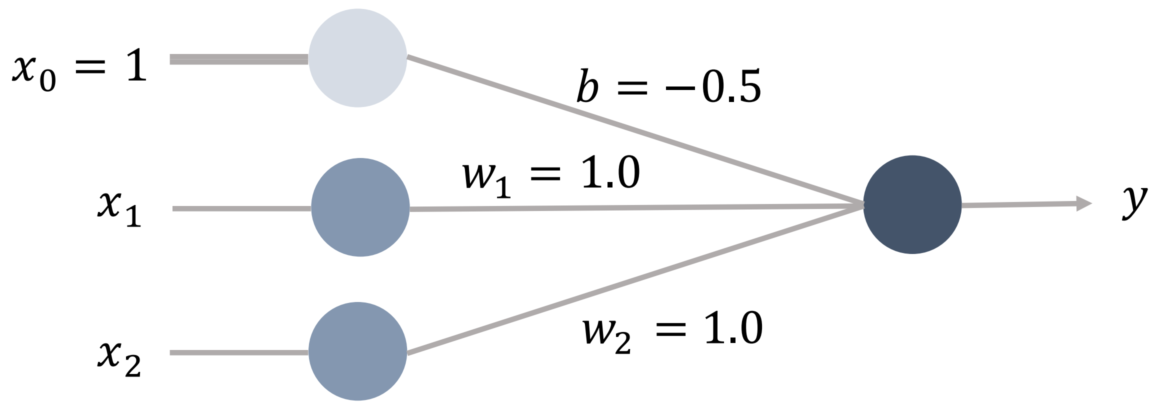
훈련 데이터 집합

$$\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}, \qquad \mathbb{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$$

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$y_1 = -1 \qquad y_2 = 1 \qquad y_3 = 1 \qquad y_4 = 1$$

퍼셉트론 Perceptron



$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $\quad \mathbf{x}_1 : s = -0.5 + 0 * 1.0 + 0 * 1.0 = -0.5,$ $\quad \tau(-0.5) = -1$ $\quad y_1 = -1$

$\mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $\quad \mathbf{x}_2 : s = -0.5 + 1 * 1.0 + 0 * 1.0 = 0.5,$ $\quad \tau(0.5) = 1$ $\quad y_2 = 1$

$\mathbf{x}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $\quad \mathbf{x}_3 : s = -0.5 + 0 * 1.0 + 1 * 1.0 = 0.5,$ $\quad \tau(0.5) = 1$ $\quad y_3 = 1$

$\mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\quad \mathbf{x}_4 : s = -0.5 + 1 * 1.0 + 1 * 1.0 = 0.5,$ $\quad \tau(1.5) = 1$ $\quad y_4 = 1$

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

$$\mathbf{x} = (x_1, x_2, \cdots, x_d)^{\mathrm{T}}, \qquad \mathbf{w} = (w_1, w_2, \cdots, w_d)^{\mathrm{T}}$$

$$s = \begin{bmatrix} w_1 & w_2 & \cdots & w_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} + w_0$$

$$s = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + w_0$$

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

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

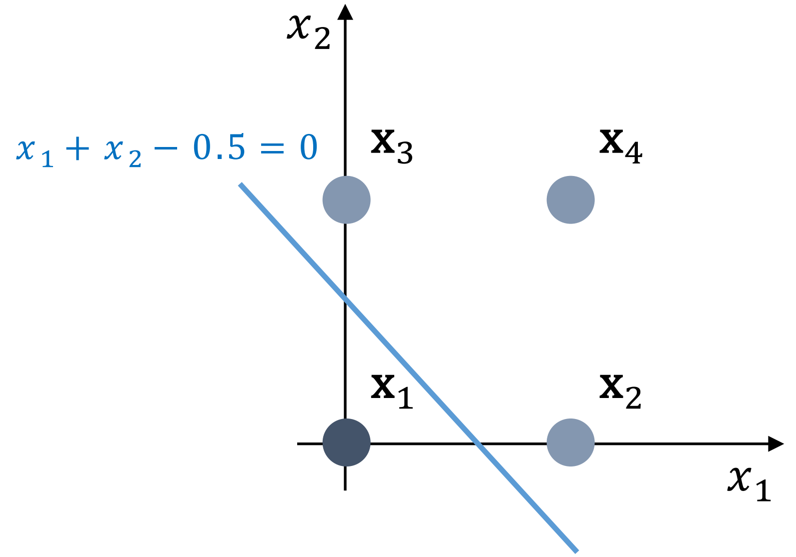
$$y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x})$$

퍼셉트론 기하학적 설명

$$d(\mathbf{x}) = d(x_1, x_2) = w_1 x_1 + w_2 x_2 + w_0 = 0 \quad \rightarrow \quad x_1 + x_2 - 0.5 = 0$$

w_1과 w_2는 직선의 방향, w_0은 절편을 결정

결정 직선은 전체 공간을 +1과 1의 두 부분공간으로 분할하는 분류기 역할



d차원 공간에서는 $d(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + w_0 = 0$

2차원은 결정 직선, 3차원은 결정 평면, 4차원 이상은 결정 초평면

퍼셉트론 목적함수 설계

퍼셉트론의 매개변수 $w = (w_0, w_1, w_2, \cdots, w_d)$

목적함수를 $J(\Theta)$ 또는 $J(\mathbf{w})$로 표기

목적함수의 조건

$J(\Theta) \geq 0$

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

틀리는 샘플이 많은 $\mathbf{w}$ 일수록 $J(\mathbf{w})$는 큰 값을 가진다.

Y는 $\mathbf{w}$가 틀리는 샘플의 집합

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

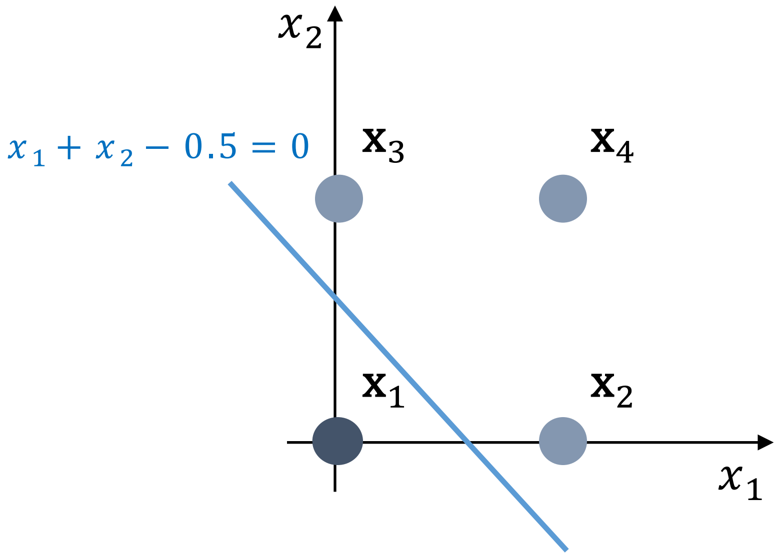
목적함수 설계

Y는 $\mathbf{w}$가 틀리는 샘플의 집합

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

$$y_k = \begin{cases} 1 \\ -1 \end{cases} \qquad \mathbf{w}_k^{\mathrm{T}}\mathbf{x}_k = \begin{cases} 1 \\ -1 \end{cases}$$

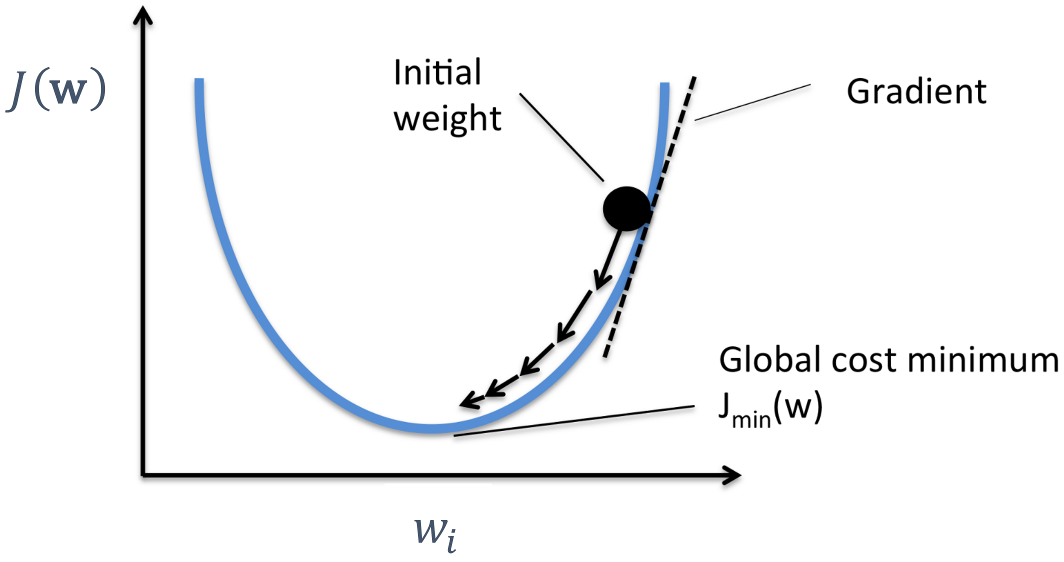
y_k와 $\mathbf{w}_k^{\mathrm{T}}\mathbf{x}_k$가 같은 값을 가지면 1



퍼셉트론 목적함수 설계

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

경사하강법 Gradient Descent



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

$$= \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}, \qquad i = 0, 1, \cdots, d$$

퍼셉트론 가중치 업데이트

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

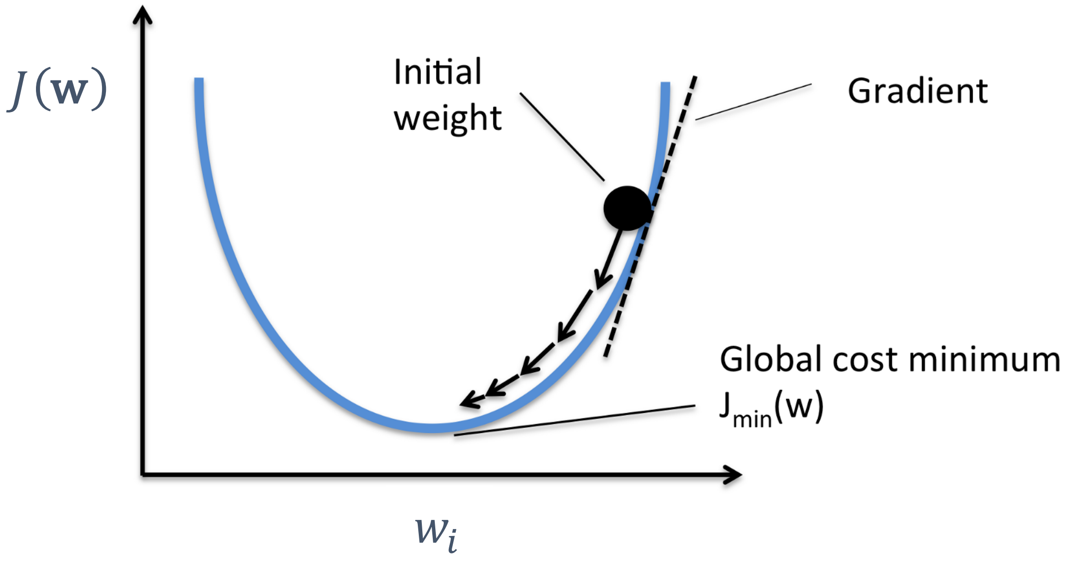
$$= \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}, \qquad i = 0,1,\cdots,d$$

가중치 업데이트

$$w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}, \qquad i = 0,1,\cdots,d$$

ρ: 학습률

경사하강법 Gradient Descent



배치 학습

입력: 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ

출력: 최적 가중치 $\hat{\mathbf{w}}$

난수를 생성하여 초기해 w를 설정한다.
repeat
    $Y = \emptyset$
    for $j = 1$ to n
        $y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j)$
        if $(y \neq y_j)$ $Y = Y \cup \mathbf{x}_j$
    if $(Y \neq \emptyset)$
        for $i = 0$ to d
            $w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki}$
until $(Y \neq \emptyset)$

$\hat{\mathbf{w}} = \mathbf{w}$

스토캐스틱(Stochastic) 학습

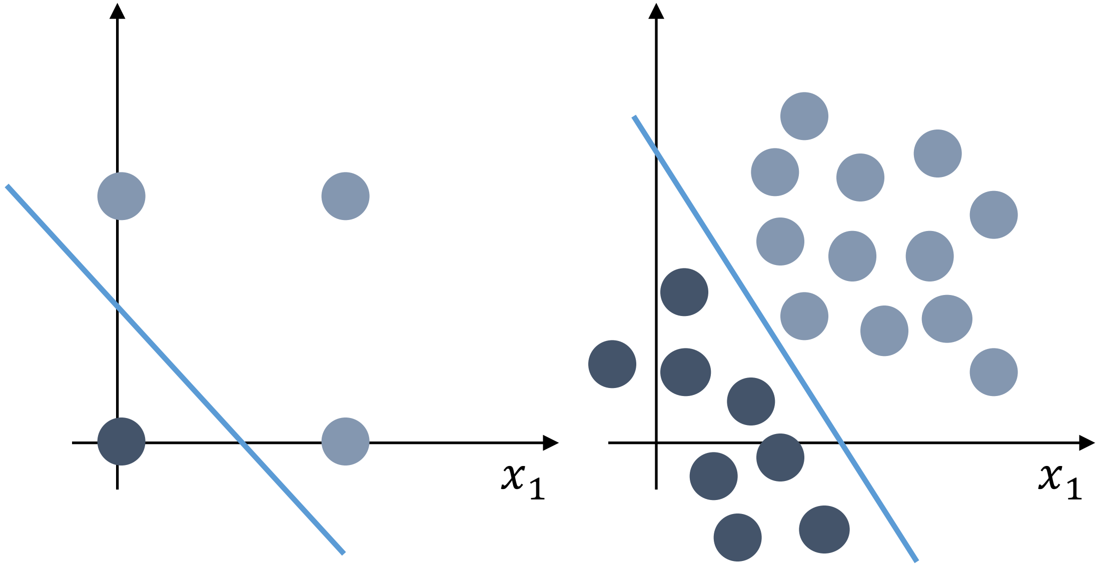
입력: 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ

출력: 최적 가중치 $\hat{\mathbf{w}}$

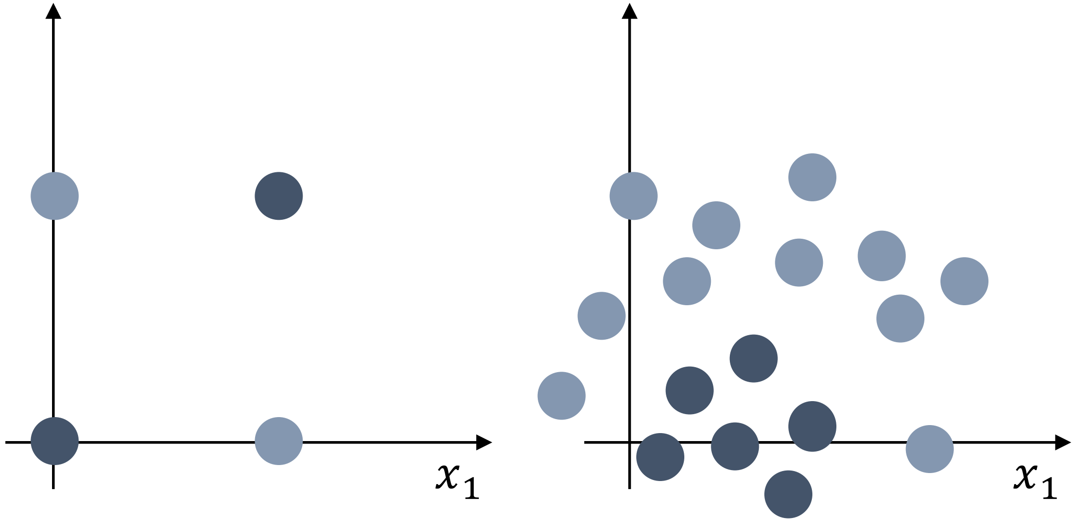
난수를 생성하여 초기해 w를 설정한다.
repeat
    $\mathbb{X}$의 샘플 순서를 섞는다.
    quit = true
    for $j = 1$ to n
        $y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j)$
        if $(y \neq y_j)$
            quit = false
            for $i = 0$ to d
                $w_i = w_i + \rho y_j x_{ji}$
until (quit)

$\hat{\mathbf{w}} = \mathbf{w}$

다층 퍼셉트론 Multi-Layer Perceptron



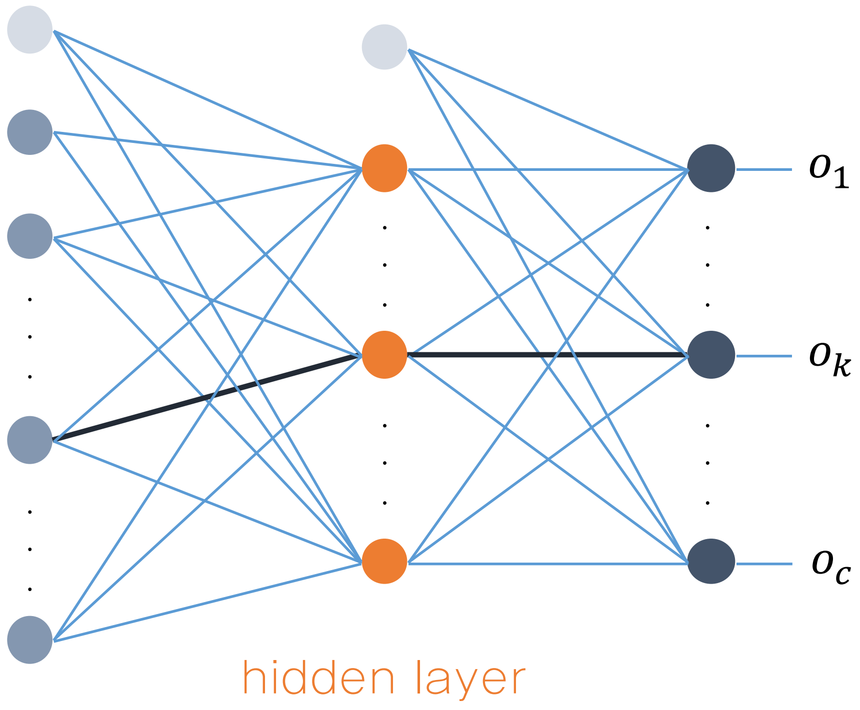
선형분리 가능



선형분리 불가능

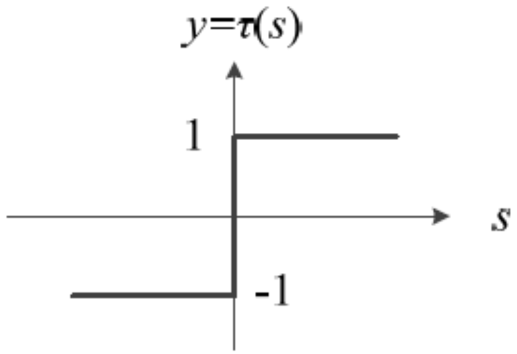
다층 퍼셉트론 핵심 아이디어

- 은닉층(Hidden Layer)
  은닉층은 원래 특징 공간을 분류하는데 훨씬 유리한 새로운 특징 공간으로 변환

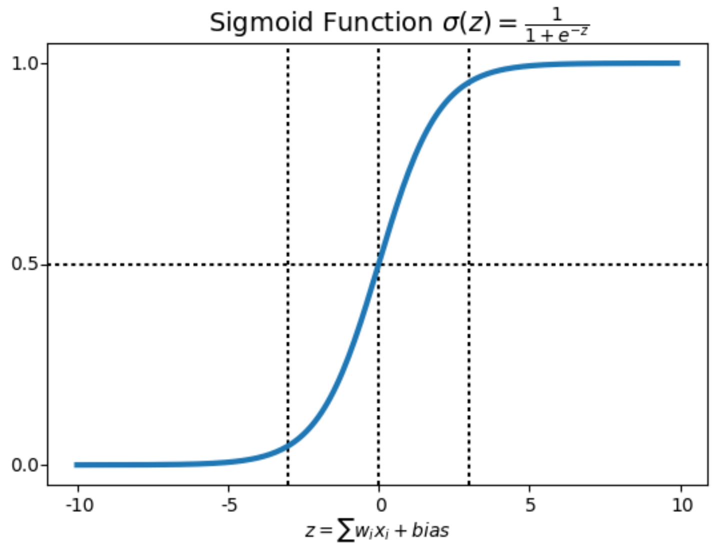


다층 퍼셉트론 핵심 아이디어

- **시그모이드 활성함수**
  퍼셉트론은 계단함수를 활성함수로 사용.
  시그모이드 함수는 출력값이 연속값이기 때문에 이를
  신뢰도로 간주함으로써 더 융통성 있게 의사결정을 할 수 있다.



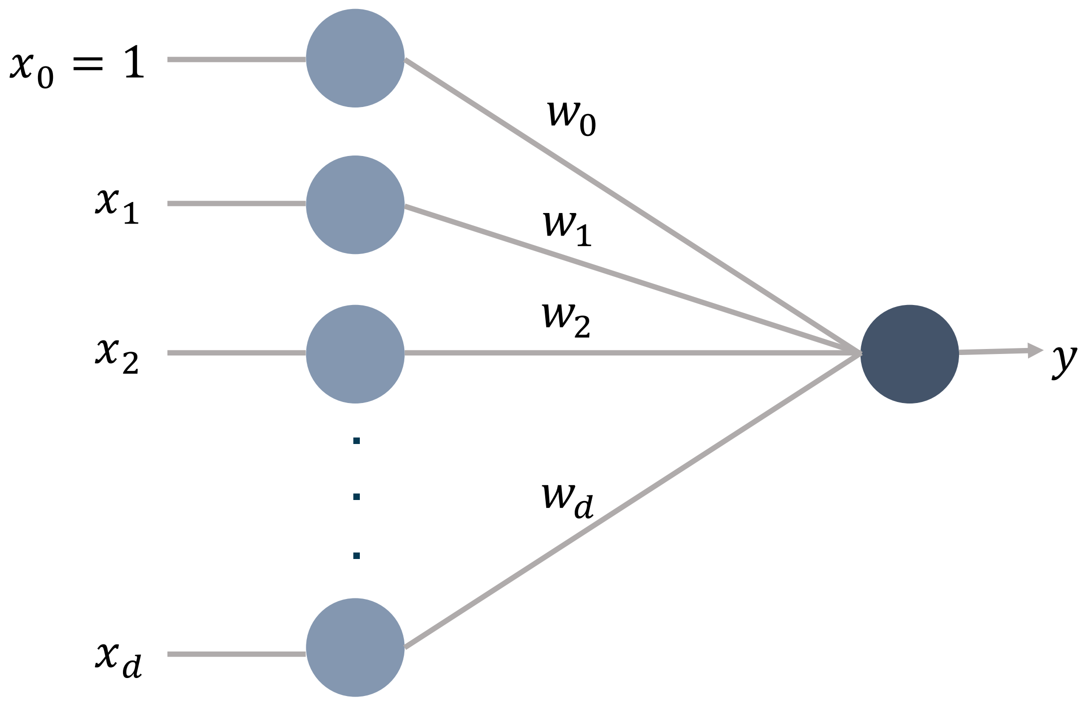
계단 함수

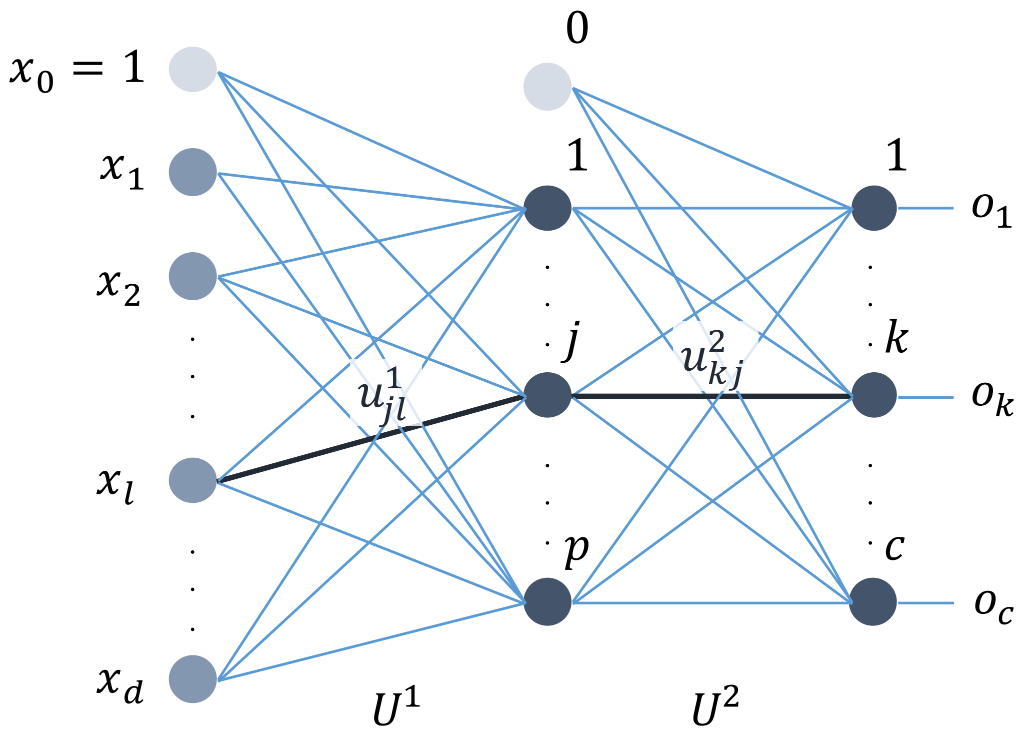


시그모이드 함수

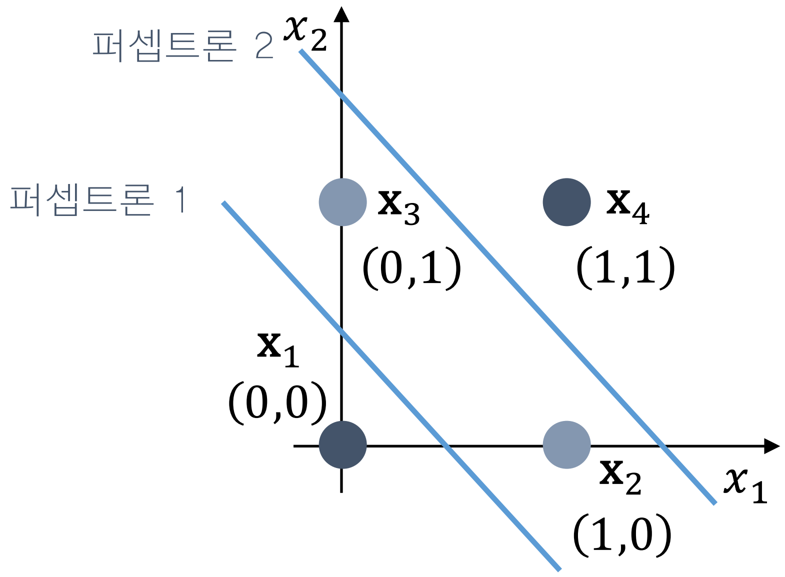
다층 퍼셉트론 핵심 아이디어

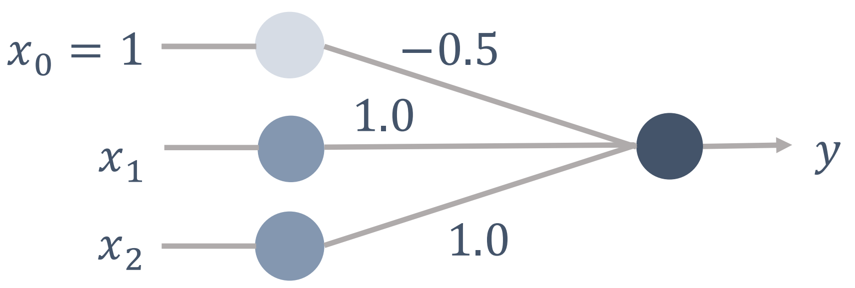
- 오류 역전파 알고리즘 사용
  다층 퍼셉트론은 여러 층이 순차적으로 이어진 구조
  역방향으로 진행하면서 한번에 한 층씩 그레이디언트를 계산하고 가중치를 갱신하는 방식

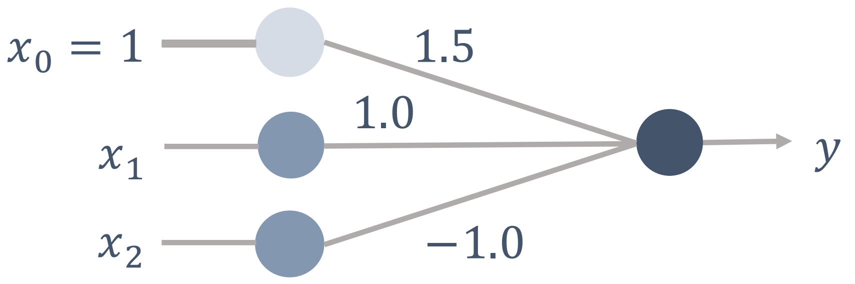




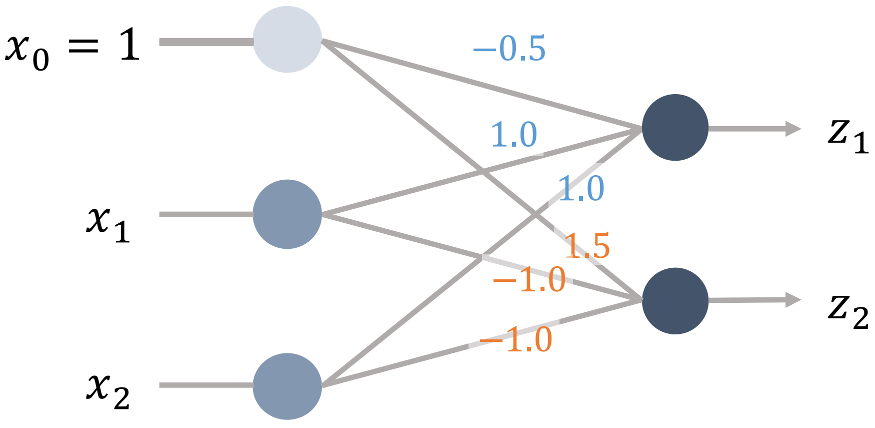
다층 퍼셉트론 특징공간 변환

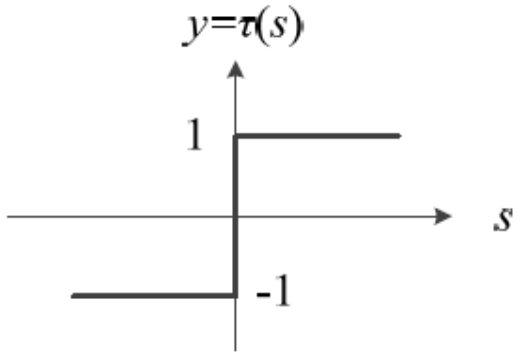






다층 퍼셉트론 특징공간 변환





$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \longrightarrow \mathbf{z}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$s_1 = 1 * -0.5 + 0 * 1.0 + 0 * 1.0 = -0.5$ $\quad z_1 = \tau(-0.5) = -1$

$s_2 = 1 * 1.5 + 0 * -1.0 + 0 * -1.0 = 1.5$ $\quad z_2 = \tau(1.5) = 1$

$$\mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \longrightarrow \mathbf{z}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$s_1 = 1 * -0.5 + 1 * 1.0 + 0 * 1.0 = 0.5$ $\quad z_1 = \tau(0.5) = 1$

$s_2 = 1 * 1.5 + 1 * -1.0 + 0 * -1.0 = 0.5$ $\quad z_2 = \tau(-0.5) = 1$

$$\mathbf{x}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \longrightarrow \mathbf{z}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$s_1 = 1 * -0.5 + 0 * 1.0 + 1 * 1.0 = 0.5$ $\quad z_1 = \tau(0.5) = 1$

$s_2 = 1 * 1.5 + 0 * -1.0 + 1 * -1.0 = 0.5$ $\quad z_2 = \tau(0.5) = 1$

$$\mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \longrightarrow \mathbf{z}_4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

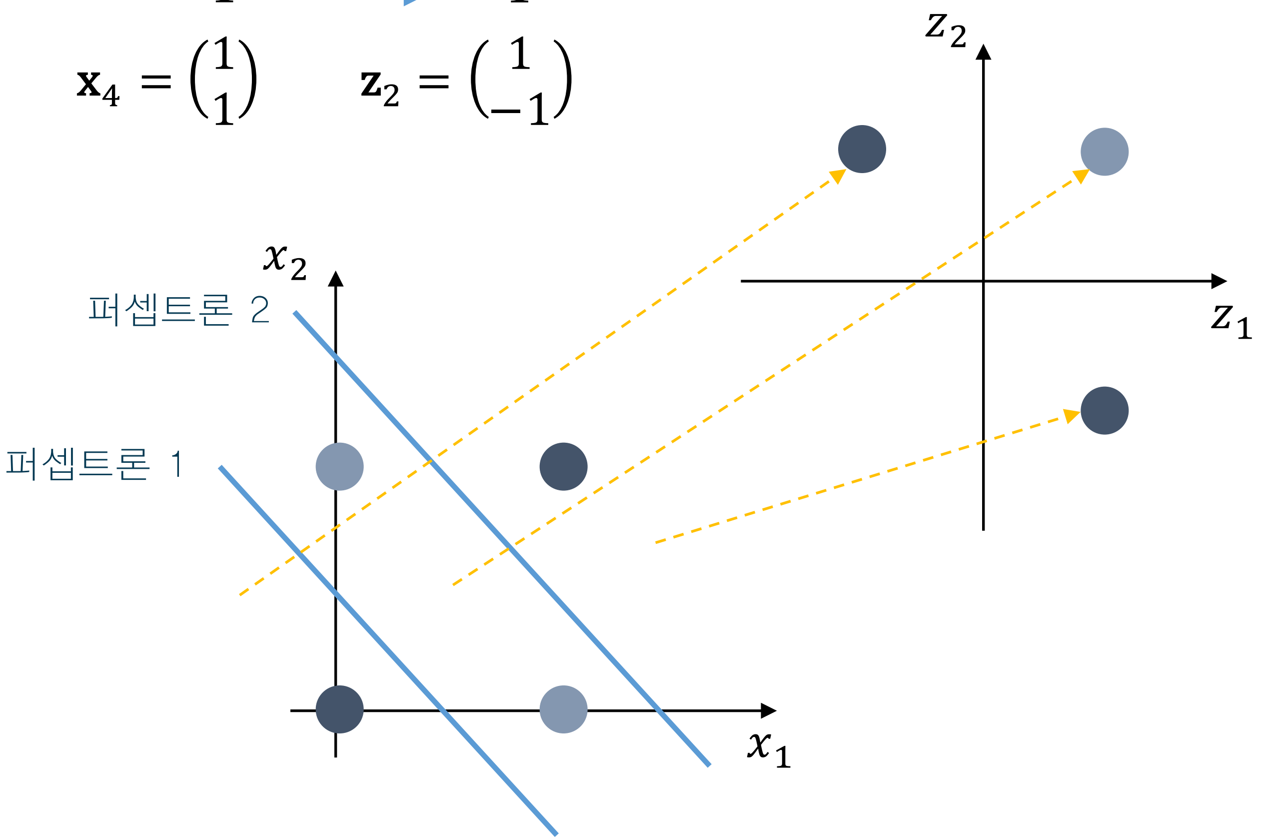
$s_1 = 1 * -0.5 + 1 * 1.0 + 1 * 1.0 = 1.5$ $\quad z_1 = \tau(1.5) = 1$

$s_2 = 1 * 1.5 + 1 * -1.0 + 1 * -1.0 = -0.5$ $\quad z_2 = \tau(-0.5) = -1$

다층 퍼셉트론 특징공간 변환

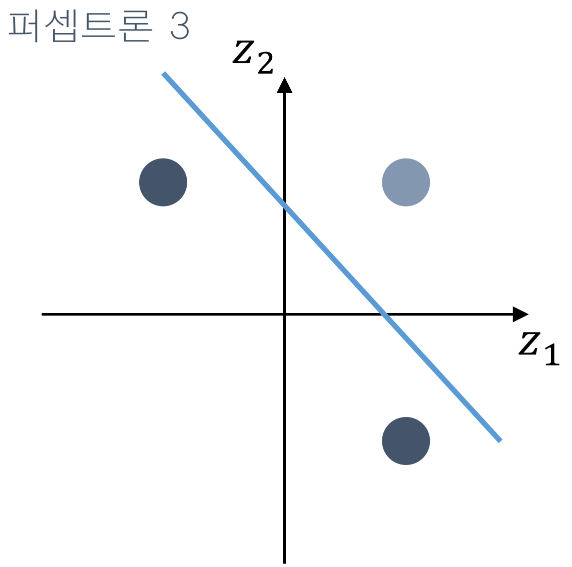
$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \mathbf{z}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \qquad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \mathbf{z}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

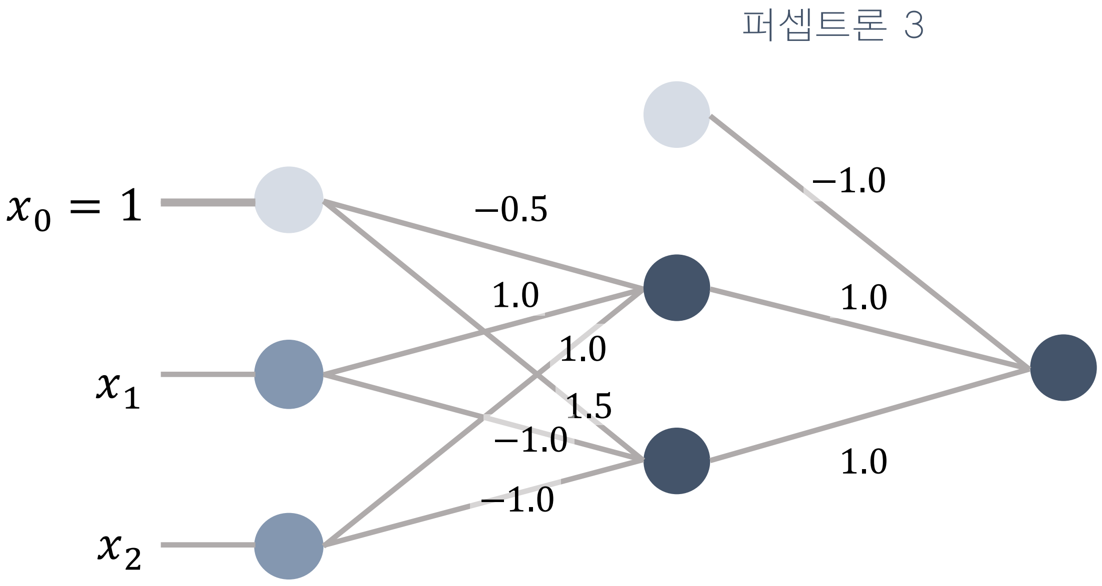
$$\mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \mathbf{z}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \mathbf{z}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$



원래 특징 공간 $\mathbf{x}$를 새로운 특징 공간 $\mathbf{z}$로 변환

다층 퍼셉트론 특징공간 변환





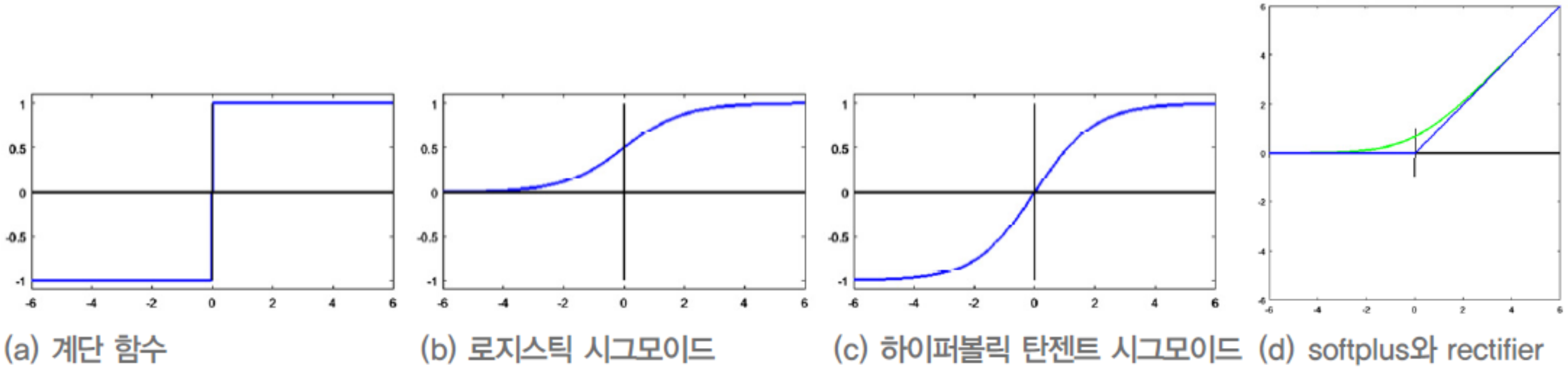
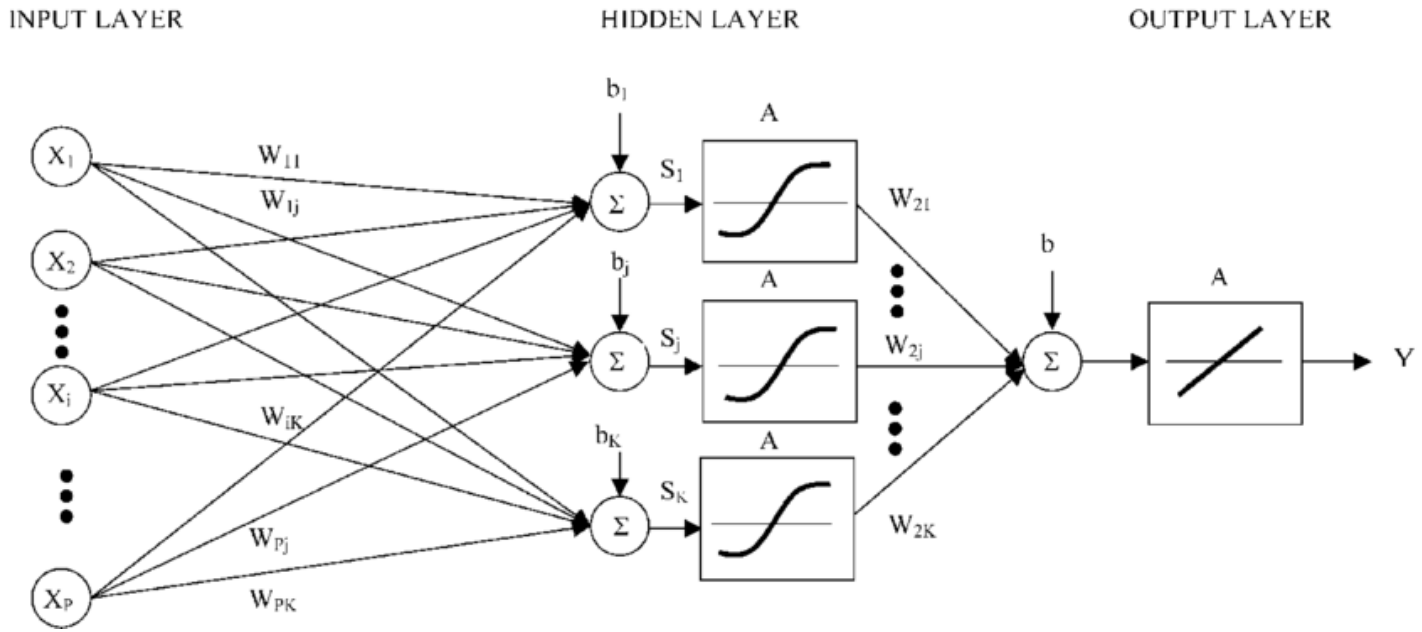


그림 3-12 신경망이 사용하는 활성함수

다층 퍼셉트론 활성함수 Activation Function

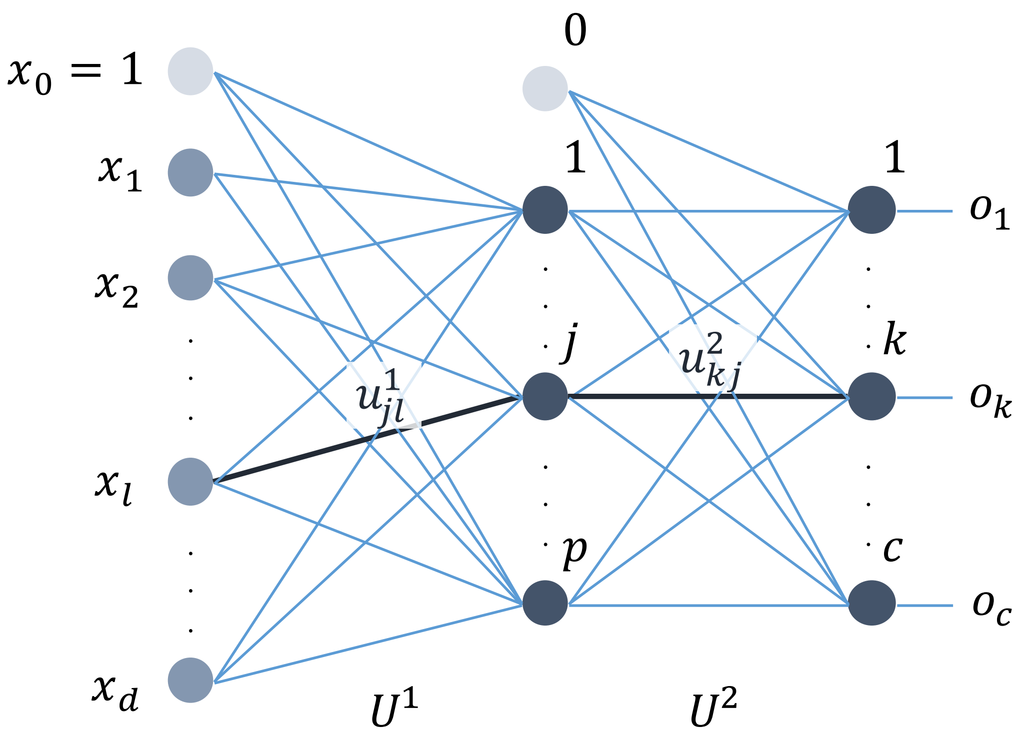


다층 퍼셉트론 활성함수

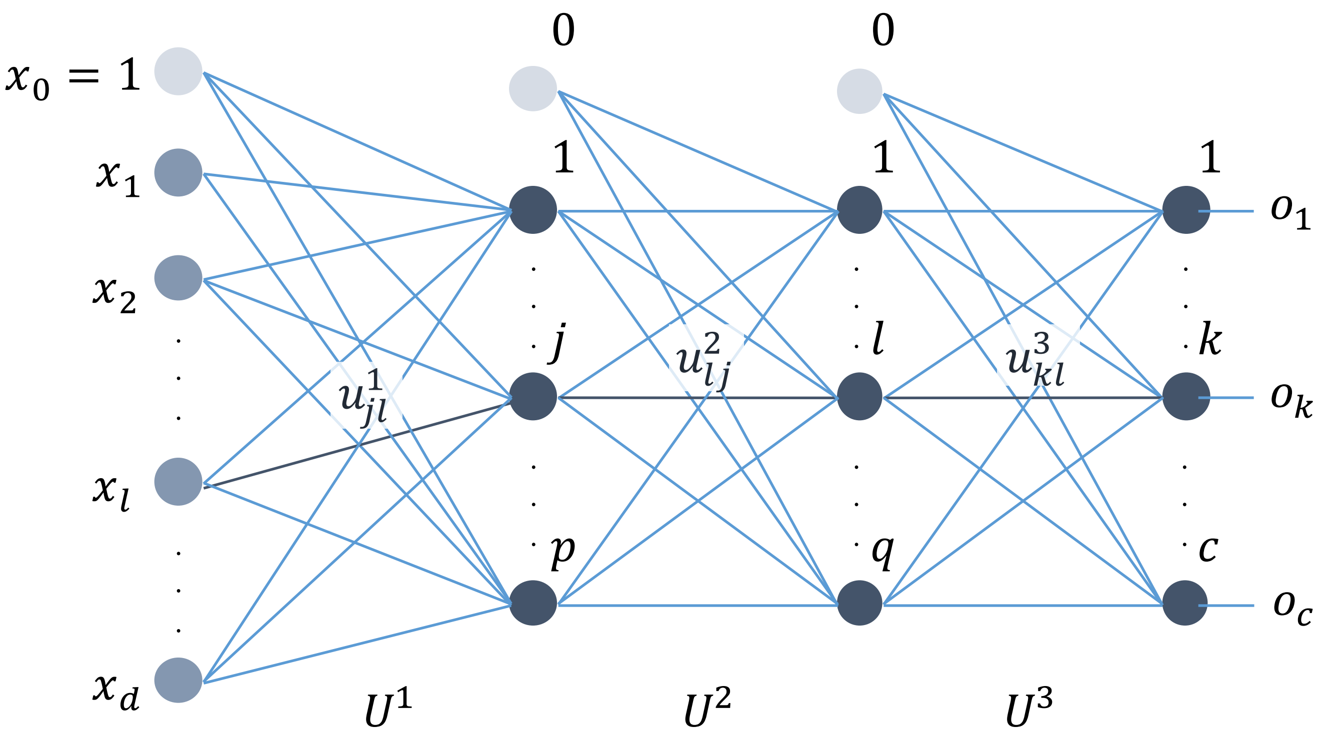
표 3-1 활성함수로 사용되는 여러 함수

함수 이름	함수	1차 도함수	범위
계단	$\tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases}$	$\tau'(s) = \begin{cases} 0 & s \neq 0 \\ \text{불가} & s = 0 \end{cases}$	-1과 1
로지스틱 시그모이드	$\tau(s) = \frac{1}{1+e^{-as}}$	$\tau'(s) = a\tau(s)\big(1-\tau(s)\big)$	(0,1)
하이퍼볼릭 탄젠트	$\tau(s) = \frac{2}{1+e^{-as}} - 1$	$\tau'(s) = \frac{a}{2}(1-\tau(s)^2)$	(-1,1)
소프트플러스	$\tau(s) = \log_e(1+e^s)$	$\tau'(s) = \frac{1}{1+e^{-s}}$	(0, ∞)
렉티파이어(ReLU)	$\tau(s) = \max(0, s)$	$\tau'(s) = \begin{cases} 0 & s < 0 \\ 1 & s > 0 \\ \text{불가} & s = 0 \end{cases}$	[0, ∞)

다층 퍼셉트론 MLP 구조



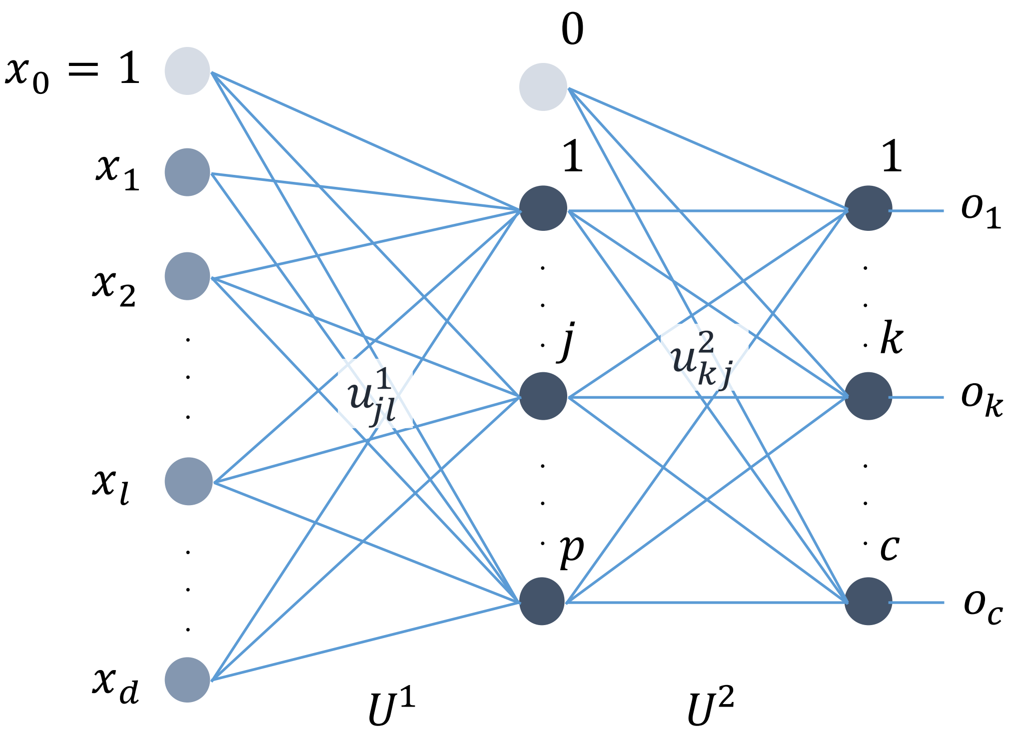
2층 퍼셉트론



3층 퍼셉트론

다층 퍼셉트론 가중치 행렬

입력층-은닉층을 연결하는 $\mathbf{U}^1$(u^2_{ji}은 입력층의 i번째 노드를 은닉층의 j번째 노드와 연결)
은닉층-출력층을 연결하는 $\mathbf{U}^1$ (u^2_{kj}는 은닉층의 j번째 노드를 출력층의 k번째 노드와 연결)



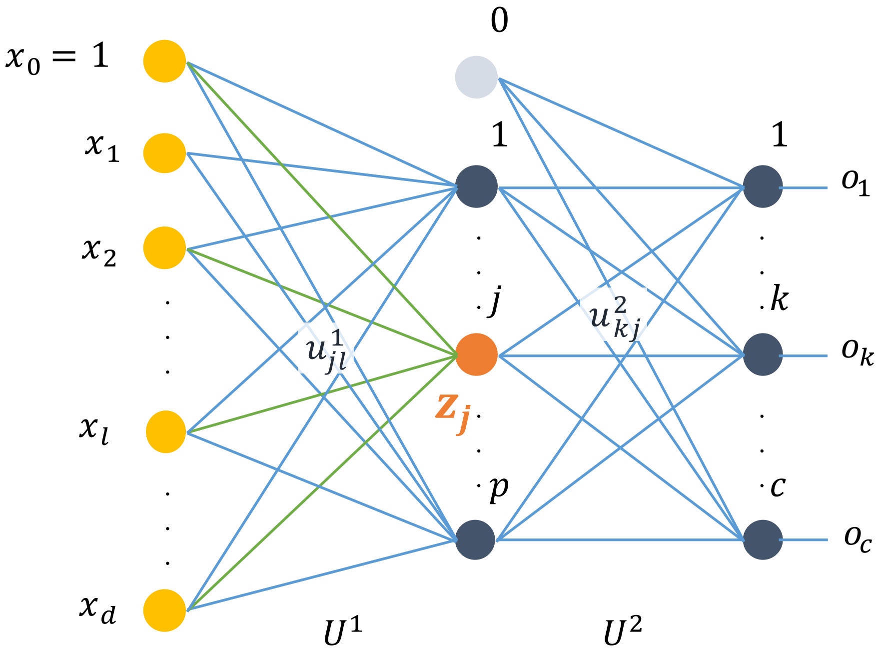
2층 퍼셉트론

2층 퍼셉트론 가중치 행렬

$$U^1 = \left(\begin{bmatrix} u^1_{10} & u^1_{11} & \cdots & u^1_{1d} \\ u^1_{20} & u^1_{21} & & u^1_{2d} \\ \vdots & & \ddots & \vdots \\ u^1_{p0} & u^1_{p1} & \cdots & u^1_{pd} \end{bmatrix}\right)$$

$$U^2 = \left(\begin{bmatrix} u^2_{10} & u^2_{11} & \cdots & u^2_{1p} \\ u^2_{20} & u^1_{21} & & u^2_{2p} \\ \vdots & & \ddots & \vdots \\ u^2_{c0} & u^2_{c1} & \cdots & u^2_{cp} \end{bmatrix}\right)$$

다층 퍼셉트론 동작



j번째 은닉 노드의 연산

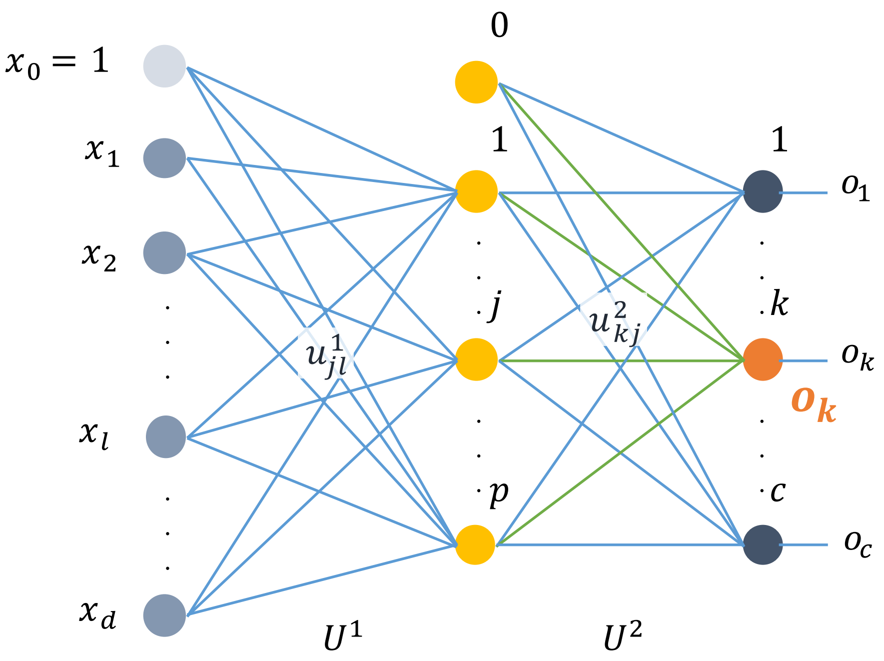
$$z_j = \tau(zsum_j), \qquad j = 1,2,\cdots,p$$

$$zsum_j = \mathbf{u}_j^1 \mathbf{x}$$

$$zsum_j = u^1_{j0} + u^1_{j1}x_1 + u^1_{j2}x_2 + \cdots + u^1_{jd}x_d$$

$$\mathbf{u}_j^1 = \left(u^1_{j0}, u^1_{j1}, \cdots, u^1_{jd}\right), \qquad \mathbf{x} = (1, x_1, x_2, \cdots, x_d)^T$$

다층 퍼셉트론 동작



k번째 출력 노드의 연산

$$o_k = \tau(osum_k), \qquad k = 1,2,\cdots,c$$

$$osum_k = \mathbf{u}_k^2\mathbf{z}$$

$$osum_k = u_{k0}^2 + u_{k1}^2 z_1 + u_{k2}^2 z_2 + \cdots + u_{kp}^2 z_p$$

$$\mathbf{u}_k^2 = \left(u_{k0}^2, u_{k1}^2, \cdots, u_{kp}^2\right), \qquad \mathbf{z} = (1, z_1, z_2, \cdots, z_p)^T$$

다층 퍼셉트론 행렬 표기

은닉 노드의 행렬 표기

$$zsum_j = u_{j0}^1 + u_{j1}^1 x_1 + u_{j2}^1 x_2 + \cdots + u_{jd}^1 x_d$$

$$\mathbf{zsum} = \mathbf{U}^1\mathbf{x} = \begin{bmatrix} u_{10}^1 & u_{11}^1 & \cdots & u_{1d}^1 \\ u_{20}^1 & u_{21}^1 & & u_{2d}^1 \\ \vdots & & \ddots & \vdots \\ u_{p0}^1 & u_{p1}^1 & \cdots & u_{pd}^1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

다층 퍼셉트론의 동작을 행렬로 표기하면,

은닉 노드의 행렬 표기

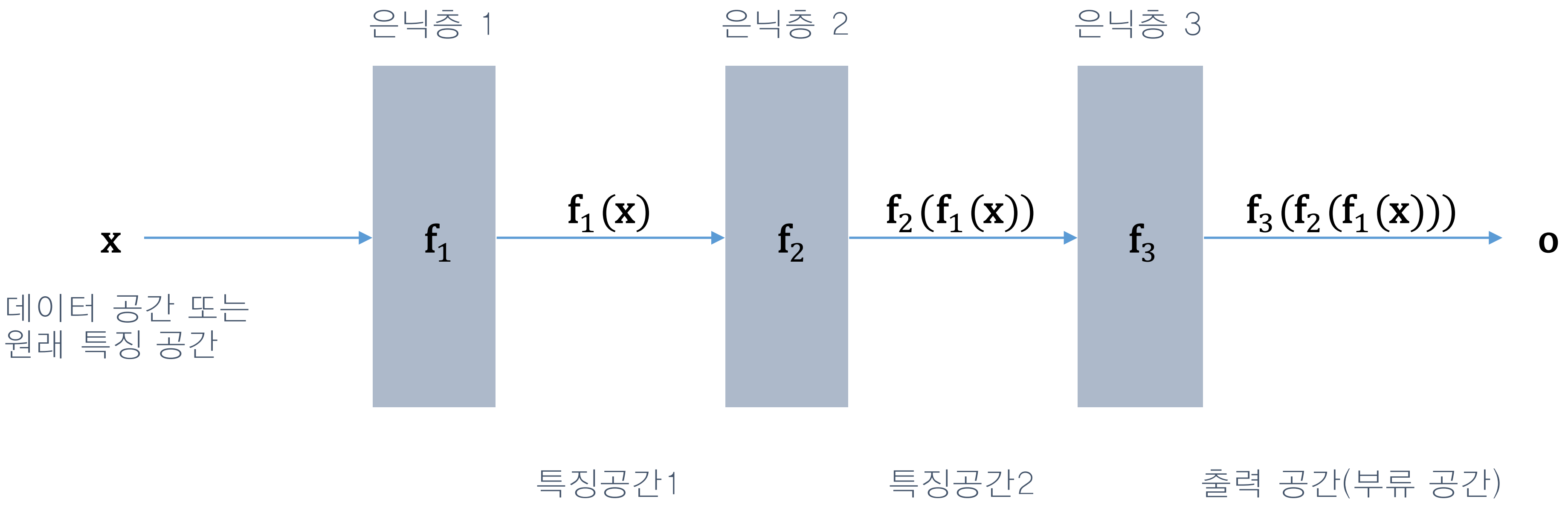
$$osum_k = u_{k0}^2 + u_{k1}^2 z_1 + u_{k2}^2 z_2 + \cdots + u_{kp}^2 z_p$$

$$\mathbf{osum} = \mathbf{U}^2\mathbf{z} = \begin{bmatrix} u_{10}^2 & u_{11}^2 & \cdots & u_{1p}^2 \\ u_{20}^2 & u_{21}^2 & & u_{2p}^2 \\ \vdots & & \ddots & \vdots \\ u_{c0}^2 & u_{c1}^2 & \cdots & u_{cp}^2 \end{bmatrix} \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_p \end{bmatrix}$$

$$\mathbf{o} = \tau\left(\mathbf{U}^2 \underbrace{\tau_h(\mathbf{U}^1\mathbf{x})}_{\mathbf{z}}\right)$$

다층 퍼셉트론 특징 학습

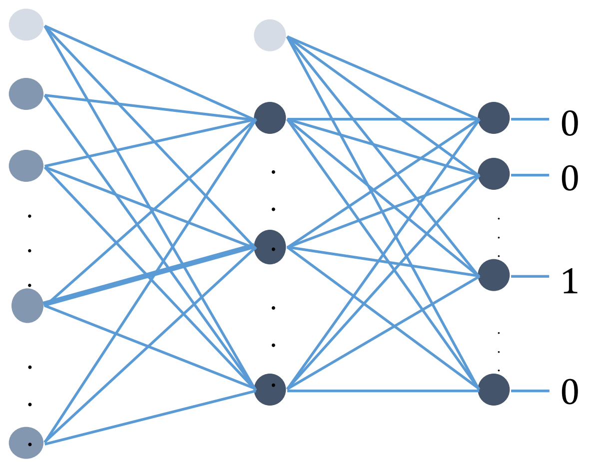
은닉층은 특징 벡터를 분류에 더 유리한 새로운 특징 공간으로 변환



다층 퍼셉트론 목적 함수 Cost Function

훈련 집합

$$\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}, \mathbb{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \quad \mathbf{y}_1 = (0, 0, \cdots, 1, \cdots, 0)^{\mathrm{T}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$



행렬 표현

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \end{pmatrix}, \qquad \mathbf{Y} = \begin{pmatrix} \mathbf{y}_1^{\mathrm{T}} \\ \mathbf{y}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{y}_n^{\mathrm{T}} \end{pmatrix}$$

기계학습의 목적

$$\mathbf{Y} = \mathbf{f}(\mathbf{X})$$

$$y_i = \mathbf{f}(\mathbf{x}_i), \qquad i = 1, 2, 3, \cdots, n$$

다층 퍼셉트론 목적 함수 Cost Function

목적 함수

배치모드

$$e = \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{y}_i - \mathbf{o}_i\|_2^2$$

온라인모드

$$e = \frac{1}{2}\|\mathbf{y} - \mathbf{o}\|_2^2$$

2층 퍼셉트론의 경우 파라미터, $\Theta = \{\mathbf{U}^1, \mathbf{U}^2\}$

$$J(\Theta) = \frac{1}{2}\|\mathbf{y} - \mathbf{o}(\Theta)\|_2^2 = \frac{1}{2}\sum_{i=1,c}(y_i - o_i)^2$$

$J(\Theta) = J(\{\mathbf{U}^1, \mathbf{U}^2\})$ 의 최저점을 찾아주는 경사하강법

$$\mathbf{U}^1 = \mathbf{U}^1 - \rho\frac{\partial J}{\partial \mathbf{U}^1}$$

$$\mathbf{U}^2 = \mathbf{U}^2 - \rho\frac{\partial J}{\partial \mathbf{U}^2}$$

- 식 (3.21)을 알고리즘 형태로 쓰면,

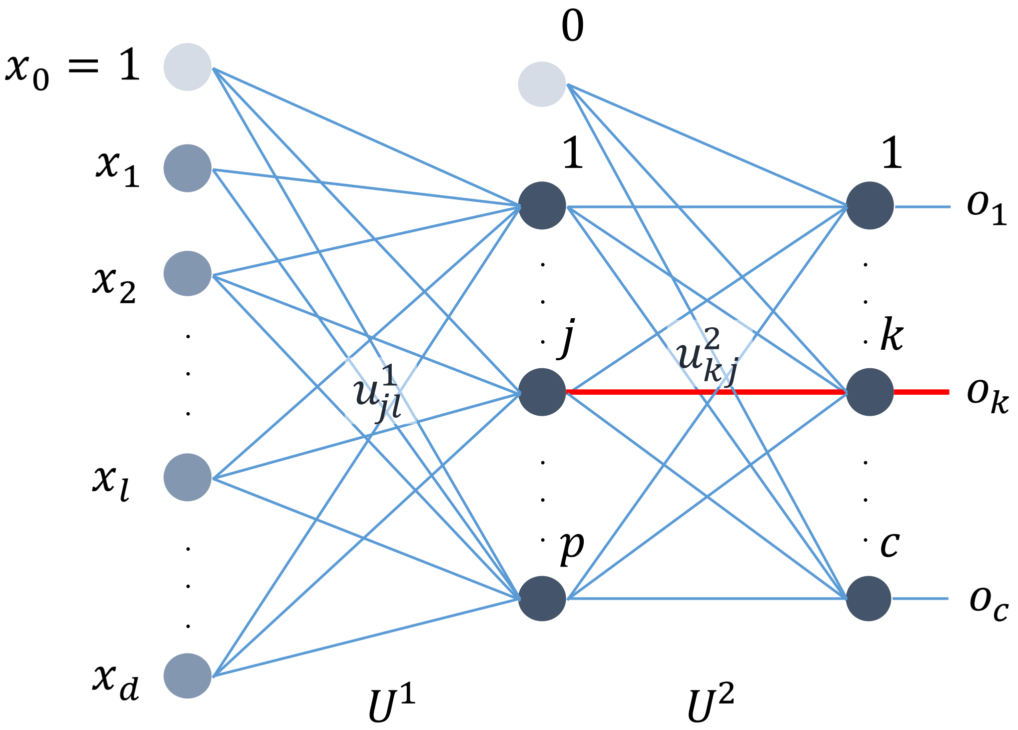
알고리즘 3-3 다층 퍼셉트론을 위한 스토케스틱 경사 하강법

입력: 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $\mathbb{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$, 학습률 ρ

출력: 가중치 행렬 $\mathbf{U}^1$과 $\mathbf{U}^2$

$\mathbf{U}^1$과 $\mathbf{U}^2$를 초기화한다.
repeat
    $\mathbb{X}$의 순서를 섞는다.
    for ($\mathbb{X}$의 샘플 각각에 대해)
        식 (3.15)로 전방 계산을 하여 $\mathbf{o}$를 구한다.
        $\frac{\partial J}{\partial \mathbf{U}^1}$와 $\frac{\partial J}{\partial \mathbf{U}^2}$를 계산한다.
        식 (3.21)로 $\mathbf{U}^1$과 $\mathbf{U}^2$를 갱신한다.
until (멈춤 조건)

도함수 값 $\frac{\partial J}{\partial \mathbf{U}^2}$의 계산 과정



$$\frac{\partial J}{\partial u_{kj}^2} = \frac{\partial (0.5\|\mathbf{y} - \mathbf{o}(\mathbf{U}^1, \mathbf{U}^2)\|_2^2)}{\partial u_{kj}^2}$$

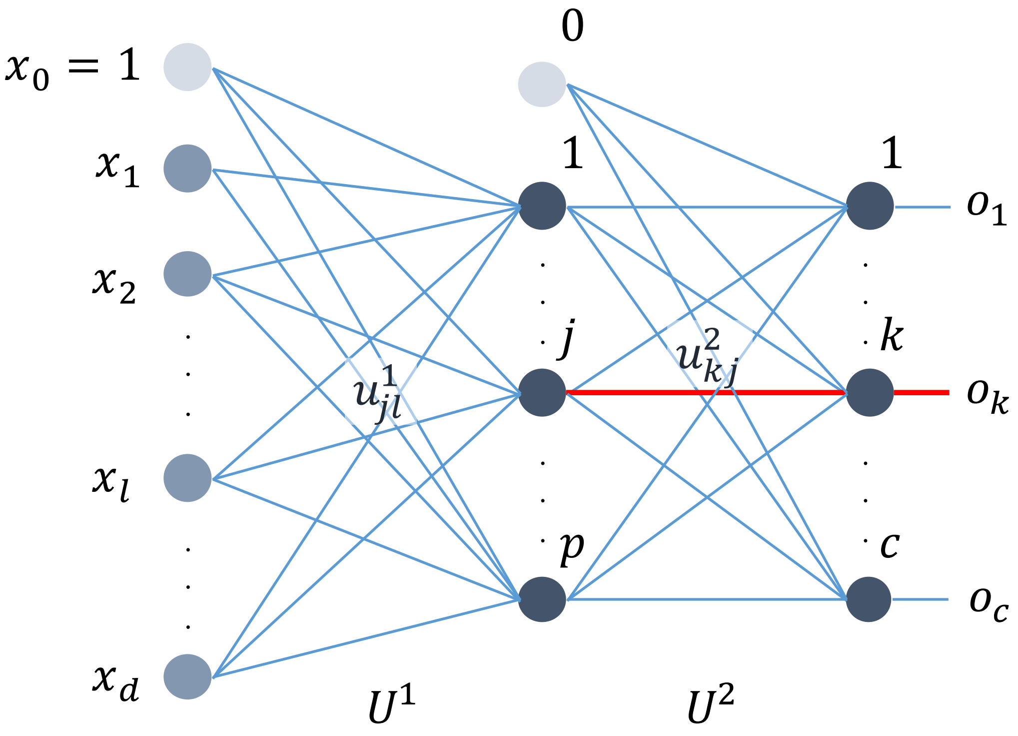
$$\frac{\partial J}{\partial u_{kj}^2} = \frac{\partial \left(0.5 \sum_{q=1}^{c} \left(y_q - o_q\right)^2\right)}{\partial u_{kj}^2}$$

$$\frac{\partial J}{\partial u_{kj}^2} = \frac{\partial \left(0.5\left((y_1 - o_1)^2 + (y_2 - o_2)^2 + \cdots + (y_k - o_k)^2 + \cdots + (y_c - o_c)^2\right)\right)}{\partial u_{kj}^2}$$

$$\frac{\partial J}{\partial u_{kj}^2} = \frac{\partial (0.5(y_k - o_k)^2)}{\partial u_{kj}^2}$$

다층 퍼셉트론 오류 역전파 유도 Backpropagation

도함수 값 $\frac{\partial J}{\partial \mathbf{U}^2}$의 계산 과정



$$\frac{\partial J}{\partial u_{kj}^2} = \frac{\partial(0.5(y_k - o_k)^2)}{\partial u_{kj}^2}$$

$y = f\big(g(x)\big)$의 도함수

$y' = f'\big(g(x)\big)g'(x)$

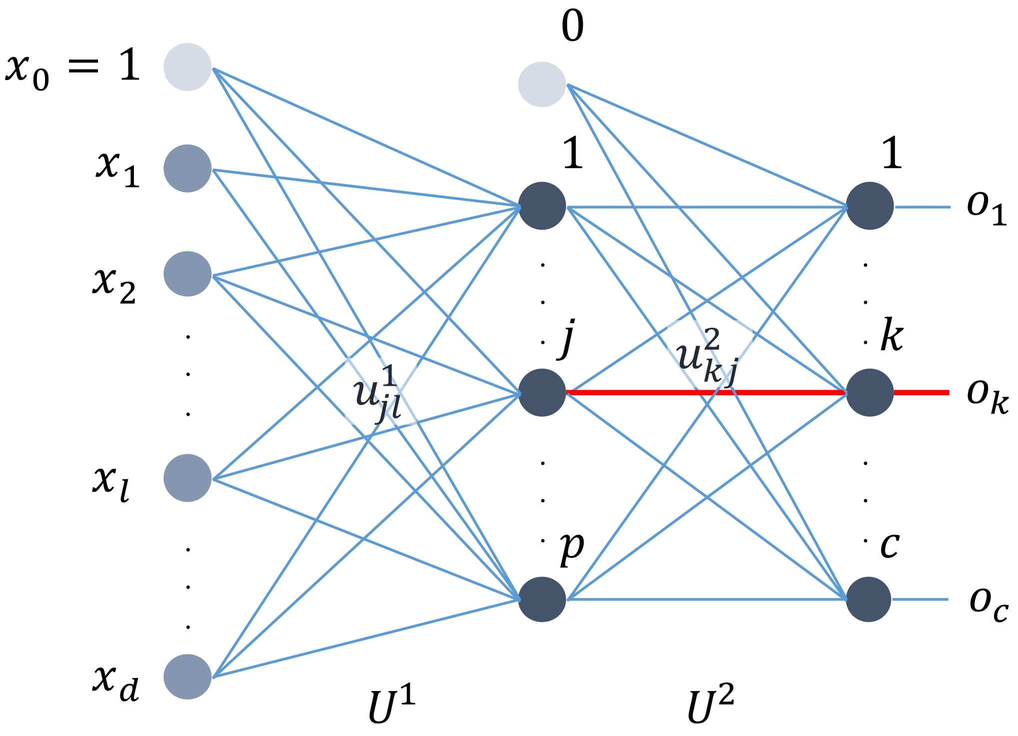
$$\frac{\partial J}{\partial u_{kj}^2} = -(y_k - o_k)\frac{\partial o_k}{\partial u_{kj}^2}$$

$$o_k = \tau(osum_k)$$

$$\frac{\partial J}{\partial u_{kj}^2} = -(y_k - o_k)\frac{\partial \tau(osum_k)}{\partial u_{kj}^2}$$

다층 퍼셉트론 오류 역전파 유도 Backpropagation

도함수 값 $\frac{\partial J}{\partial \mathbf{U}^2}$의 계산 과정



$$\frac{\partial J}{\partial u^2_{kj}} = -(y_k - o_k)\frac{\partial \tau(osum_k)}{\partial u^2_{kj}}$$

$$o_k = \tau(osum_k)$$

$$\frac{\partial J}{\partial u^2_{kj}} = -(y_k - o_k)\tau'(osum_k)\frac{\partial (osum_k)}{\partial u^2_{kj}}$$

$$osum_k = \mathbf{u}^2_k \mathbf{z}$$

$$\frac{\partial J}{\partial u^2_{kj}} = -(y_k - o_k)\tau'(osum_k)\frac{\partial (\mathbf{u}^2_k\ \mathbf{z})}{\partial u^2_{kj}}$$

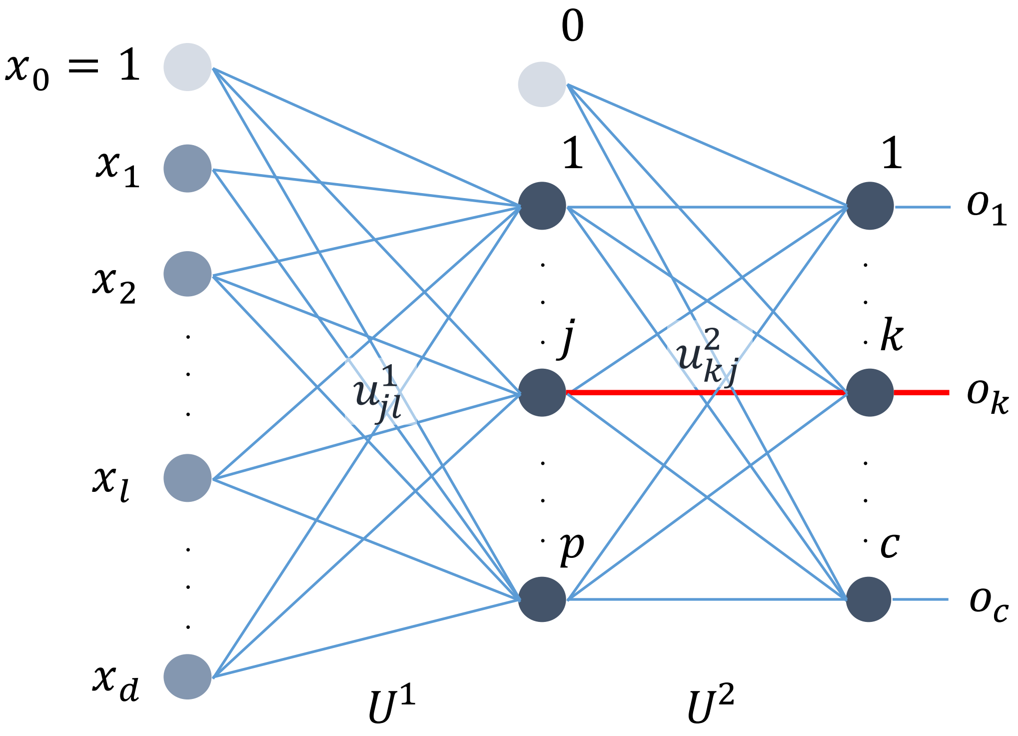
$$\mathbf{u}^2_k = \left(u^2_{k0}, u^2_{k1}, \cdots, u^2_{kp}\right),$$
$$\mathbf{z} = \left(1, z_1, z_2, \cdots, z_p\right)^T$$

$$\frac{\partial J}{\partial u^2_{kj}} = -(y_k - o_k)\tau'(osum_k)\frac{\partial (u^2_{k0} + u^2_{k1}z_1 + u^2_{k2}z_2 + \cdots + u^2_{kj}z_j + \cdots u^2_{kp}z_p)}{\partial u^2_{kj}}$$

$$\frac{\partial J}{\partial u^2_{kj}} = -(y_k - o_k)\tau'(osum_k)\frac{\partial (u^2_{kj}z_j)}{\partial u^2_{kj}}$$

$$\frac{\partial J}{\partial u^2_{kj}} = -(y_k - o_k)\tau'(osum_k)\ z_j$$

도함수 값 $\frac{\partial J}{\partial \mathbf{U}^2}$의 계산 과정



$$\frac{\partial J}{\partial u_{kj}^2} = -(y_k - o_k)\tau'(osum_k)\, z_j$$

$$\delta_k = (y_k - o_k)\tau'(osum_k), \qquad 1 \le k \le c$$

$$\frac{\partial J}{\partial u_{kj}^2} = \Delta u_{kj}^2 = -\delta_k z_j, \qquad 0 \le j \le p, 1 \le k \le c$$

$$\frac{\partial J}{\partial u_{k1}^2} = \Delta u_{k1}^2 = -\delta_k z_1 \qquad \frac{\partial J}{\partial u_{k2}^2} = \Delta u_{k2}^2 = -\delta_k z_2$$

$$\frac{\partial J}{\partial u_{ji}^1} = \frac{\partial(0.5\|\mathbf{y} - \mathbf{o}(\mathbf{U}^1, \mathbf{U}^2)\|_2^2)}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = \frac{\partial\left(0.5 \sum_{q=1}^{c}\left(y_q - o_q\right)^2\right)}{\partial u_{ji}^1}$$

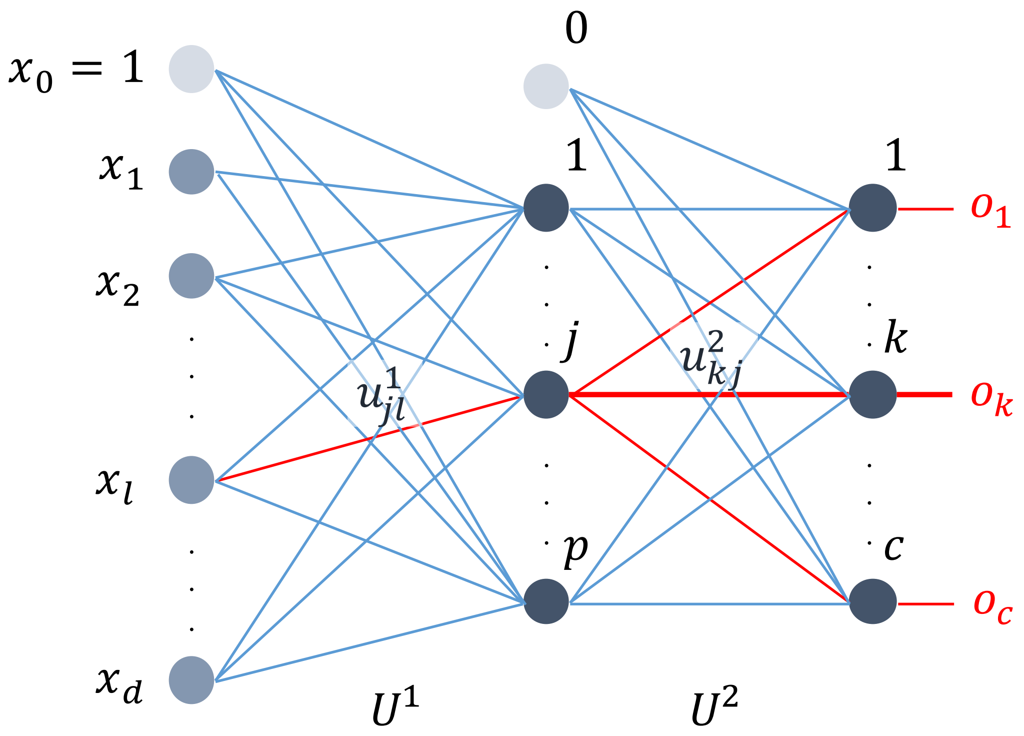
$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}\left(y_q - o_q\right)\frac{\partial o_q}{\partial u_{ji}^1}$$

$$o_q = \tau\left(osum_q\right)$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}\left(y_q - o_q\right)\frac{\partial \tau\left(osum_q\right)}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}\left(y_q - o_q\right)\tau'\left(osum_q\right)\frac{\partial osum_q}{\partial u_{ji}^1}$$

도함수 값 $\frac{\partial J}{\partial \mathbf{U}^1}$의 계산 과정



다층 퍼셉트론 오류 역전파 유도 Backpropagation

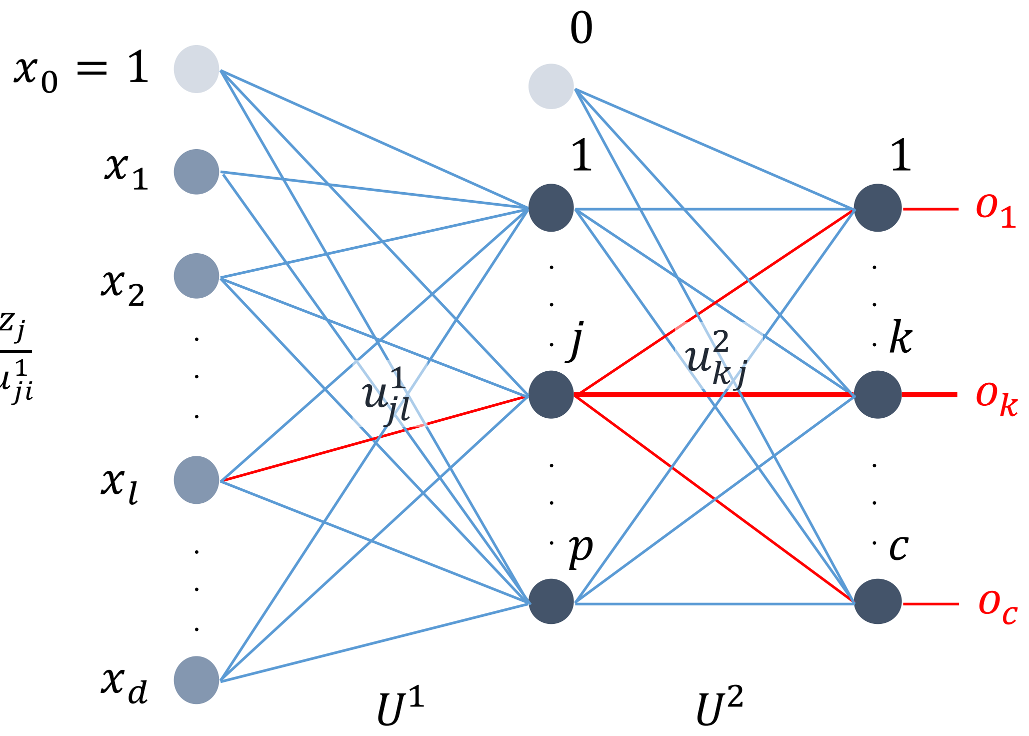
$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)\frac{\partial osum_q}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)\frac{\partial osum_q}{\partial z_j}\frac{\partial z_j}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)\frac{\partial\left(u_{q0}^2 + u_{q1}^2 z_1 + u_{q2}^2 z_2 + \cdots + u_{qj}^2 z_j + \cdots u_{qp}^2 z_p\right)}{\partial z_j}\frac{\partial z_j}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\frac{\partial z_j}{\partial u_{ji}^1}$$

도함수 값 $\frac{\partial J}{\partial \mathbf{U}^1}$의 계산 과정



다층 퍼셉트론 오류 역전파 유도 Backpropagation

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\frac{\partial z_j}{\partial u_{ji}^1}$$

$$z_j = \tau(zsum_j)$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\frac{\partial \tau(zsum_j)}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\tau'(zsum_j)\frac{\partial (zsum_j)}{\partial u_{ji}^1}$$

$$zsum_j = \mathbf{u}_j^1\mathbf{x}$$

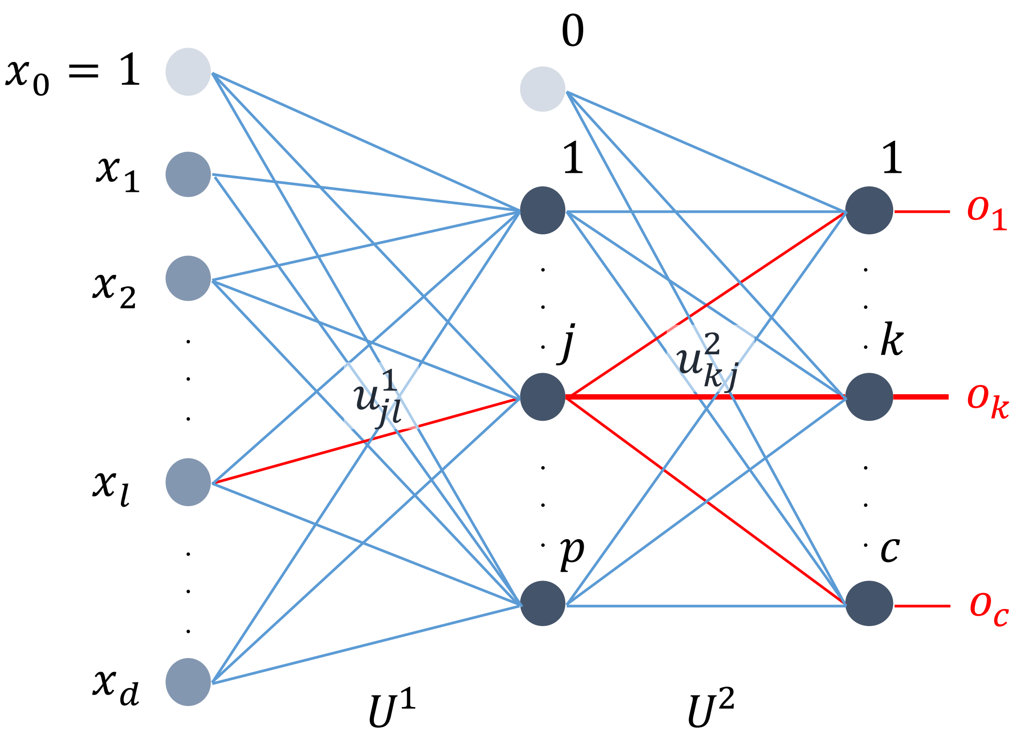
$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\tau'(zsum_j)\frac{\partial \tau(\mathbf{u}_j^1\mathbf{x})}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\frac{\partial \tau(u_{j0}^1 + u_{j1}^1x_1 + u_{j2}^1x_2 + \cdots + u_{ji}^1x_i + u_{jd}^1x_d)}{\partial u_{ji}^1}$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2\tau'(zsum_j)x_i$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\tau'(zsum_j)x_i\sum_{q=1}^{c}(y_q - o_q)\tau'(osum_q)u_{qj}^2$$

다층 퍼셉트론 오류 역전파 유도 Backpropagation



$$\delta_k = (y_k - o_k)\tau'(osum_k), \qquad 1 \le k \le c$$

$$\frac{\partial J}{\partial u_{ji}^1} = -\tau'(zsum_j)x_i \sum_{q=1}^{c} (y_q - o_q)\tau'(osum_q)u_{qj}^2$$

$$\eta_j = \tau'(zsum_j) \sum_{q=1}^{c} \delta_q u_{qj}^2, \qquad 1 \le j \le p$$

$$\frac{\partial J}{\partial u_{ji}^1} = \Delta u_{ji}^1 = -\eta_j x_i, \qquad 0 \le i \le d, 1 \le j \le p$$

$$\frac{\partial J}{\partial u_{j1}^1} = \Delta u_{j1}^1 = -\eta_j x_1 \qquad \frac{\partial J}{\partial u_{j2}^1} = \Delta u_{j2}^1 = -\eta_j x_2$$

다층 퍼셉트론 오류 역전파 유도 Backpropagation

$$\delta_k = (y_k - o_k)\tau'(osum_k), \qquad 1 \le k \le c$$

$$\frac{\partial J}{\partial u_{kj}^2} = \Delta u_{kj}^2 = -\delta_k z_j, \qquad 0 \le j \le p, 1 \le k \le c$$

$$\eta_j = \tau'(zsum_j) \sum_{q=1}^{c} \delta_q u_{qj}^2, \qquad 1 \le j \le p$$

$$\frac{\partial J}{\partial u_{ji}^1} = \Delta u_{j1}^1 = -\eta_j x_i, \qquad 0 \le i \le d, 1 \le j \le p$$

위의 식을 이용하여 출력층의 오류를 역방향으로 전파하며 그레이디언트를 계산

알고리즘 3-4 다층 퍼셉트론 학습을 위한 스토캐스틱 경사 하강법

입력: 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력: 가중치 행렬 $\mathbf{U}^1$과 $\mathbf{U}^2$

$\mathbf{U}^1$과 $\mathbf{U}^2$를 초기화한다.
repeat
  $\mathbb{X}$의 순서를 섞는다.
  for ($\mathbb{X}$의 샘플 각각에 대해)
    현재 처리하는 샘플을 $\mathbf{x}=(x_0,x_1,x_2,\cdots,x_d)^{\mathrm{T}}$, $\mathbf{y}=(y_1,y_2,\cdots,y_c)^{\mathrm{T}}$라 표기한다.
    x_0과 z_0을 1로 설정한다. // 바이어스
        // 전방 계산
    for (j=1 to p) $zsum_j=\mathbf{u}_j^1\mathbf{x}$, $z_j=\tau(zsum_j)$ // 식 (3.13)
    for (k=1 to c) $osum_k=\mathbf{u}_k^2\mathbf{z}$, $o_k=\tau(osum_k)$ // 식 (3.14)
        // 오류 역전파
    for (k=1 to c) $\delta_k=(y_k-o_k)\tau'(osum_k)$ // 식 (3.22)
    for (k=1 to c) for (j=0 to p) $\Delta u_{kj}^2=-\delta_k z_j$ // 식 (3.23)
    for (j=1 to p) $\eta_j=\tau'(zsum_j)\sum_{q=1}^{c}\delta_q u_{qj}^2$ // 식 (3.24)
    for (j=1 to p) for (i=0 to d) $\Delta u_{ji}^1=-\eta_j x_i$ // 식 (3.25)
        // 가중치 갱신
    for (k=1 to c) for (j=0 to p) $u_{kj}^2=u_{kj}^2-\rho\Delta u_{kj}^2$ // 식 (3.21)
    for (j=1 to p) for (i=0 to d) $u_{ji}^1=u_{ji}^1-\rho\Delta u_{ji}^1$ // 식 (3.21)
until (멈춤 조건)

알고리즘 3-5 다층 퍼셉트론 학습을 위한 스토캐스틱 경사 하강법(행렬 표기)

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력 : 가중치 행렬 $\mathbf{U}^1$과 $\mathbf{U}^2$

$\mathbf{U}^1$과 $\mathbf{U}^2$를 초기화한다.
repeat
  $\mathbb{X}$의 순서를 섞는다.
  for ($\mathbb{X}$의 샘플 각각에 대해)
    현재 처리하는 샘플을 $\mathbf{x}=(x_0,x_1,x_2,\cdots,x_d)^{\mathrm{T}}$, $\mathbf{y}=(y_1,y_2,\cdots,y_c)^{\mathrm{T}}$라 표기한다.
    x_0과 z_0을 1로 설정한다. // 바이어스
        // 전방 계산
    $\mathbf{zsum}=\mathbf{U}^1\mathbf{x}$, $\tilde{\mathbf{z}}=\boldsymbol{\tau}(\mathbf{zsum})$ // 식 (3.13), $\mathbf{zsum}_{p*1}$, $\mathbf{U}^1{}_{p*(d+1)}$, $\mathbf{x}_{(d+1)*1}$, $\tilde{\mathbf{z}}_{p*1}$
    $\mathbf{osum}=\mathbf{U}^2\mathbf{z}$, $\mathbf{o}=\boldsymbol{\tau}(\mathbf{osum})$ // 식 (3.14), $\mathbf{osum}_{c*1}$, $\mathbf{U}^2{}_{c*(p+1)}$, $\mathbf{z}_{(p+1)*1}$, $\mathbf{o}_{c*1}$
        // 오류 역전파
    $\boldsymbol{\delta}=(\mathbf{y}-\mathbf{o})\odot\boldsymbol{\tau}'(\mathbf{osum})$ // 식 (3.22), $\boldsymbol{\delta}_{c*1}$
    $\Delta\mathbf{U}^2=-\boldsymbol{\delta}\mathbf{z}^{\mathrm{T}}$ // 식 (3.23), $\Delta\mathbf{U}^2{}_{c*(p+1)}$
    $\boldsymbol{\eta}=(\boldsymbol{\delta}^{\mathrm{T}}\widetilde{\mathbf{U}}^2)^{\mathrm{T}}\odot\boldsymbol{\tau}'(\mathbf{zsum})$ // 식 (3.24), $\widetilde{\mathbf{U}}^2{}_{c*p}$, $\boldsymbol{\eta}_{p*1}$
    $\Delta\mathbf{U}^1=-\boldsymbol{\eta}\mathbf{x}^{\mathrm{T}}$ // 식 (3.25), $\Delta\mathbf{U}^1{}_{p*(d+1)}$
        // 가중치 갱신
    $\mathbf{U}^2=\mathbf{U}^2-\rho\Delta\mathbf{U}^2$ // 식 (3.21)
    $\mathbf{U}^1=\mathbf{U}^1-\rho\Delta\mathbf{U}^1$ // 식 (3.21)
until (멈춤 조건)

다층 퍼셉트론 미니배치 스토케스틱 경사 하강법

알고리즘 3-6 다층 퍼셉트론 학습을 위한 '미니배치' 스토캐스틱 경사 하강법

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ, 미니배치 크기 t

출력 : 가중치 행렬 $\mathbf{U}^1$과 $\mathbf{U}^2$

$\mathbf{U}^1$과 $\mathbf{U}^2$를 초기화한다.

repeat

    $\mathbb{X}$와 $\mathbb{Y}$에서 t개의 샘플을 무작위로 뽑아 미니배치 $\mathbb{X}'$와 $\mathbb{Y}'$를 만든다.

    $\Delta\mathbf{U}^2 = \mathbf{0}$, $\Delta\mathbf{U}^1 = \mathbf{0}$

    for ($\mathbb{X}'$의 샘플 각각에 대해)

        현재 처리하는 샘플을 $\mathbf{x} = (x_0, x_1, x_2, \cdots, x_d)^\mathrm{T}$, $\mathbf{y} = (y_1, y_2, \cdots, y_c)^\mathrm{T}$라 표기한다.

        x_0와 z_0를 1로 설정한다. // 바이어스

        // 전방 계산

        $\mathbf{zsum} = \mathbf{U}^1\mathbf{x}$, $\tilde{\mathbf{z}} = \boldsymbol{\tau}(\mathbf{zsum})$ // 식 (3.13), $\mathbf{zsum}_{p*1}$, $\mathbf{U}^1{}_{p*(d+1)}$, $\mathbf{x}_{(d+1)*1}$, $\tilde{\mathbf{z}}_{p*1}$

        $\mathbf{osum} = \mathbf{U}^2\mathbf{z}$, $\mathbf{o} = \boldsymbol{\tau}(\mathbf{osum})$ // 식 (3.14), $\mathbf{osum}_{c*1}$, $\mathbf{U}^2{}_{c*(p+1)}$, $\mathbf{z}_{(p+1)*1}$, $\mathbf{o}_{c*1}$

        // 오류 역전파

        $\boldsymbol{\delta} = (\mathbf{y} - \mathbf{o}) \odot \boldsymbol{\tau}'(\mathbf{osum})$ // 식 (3.22), $\boldsymbol{\delta}_{c*1}$

        $\Delta\mathbf{U}^2 = \Delta\mathbf{U}^2 + (-\boldsymbol{\delta}\mathbf{z}^\mathrm{T})$ // 식 (3.23)을 누적, $\Delta\mathbf{U}^2{}_{c*(p+1)}$

        $\boldsymbol{\eta} = \left(\boldsymbol{\delta}^\mathrm{T}\tilde{\mathbf{U}}^2\right)^\mathrm{T} \odot \boldsymbol{\tau}'(\mathbf{zsum})$ // 식 (3.24), $\tilde{\mathbf{U}}^2{}_{c*p}$, $\boldsymbol{\eta}_{p*1}$

        $\Delta\mathbf{U}^1 = \Delta\mathbf{U}^1 + (-\boldsymbol{\eta}\mathbf{x}^\mathrm{T})$ // 식 (3.25)를 누적, $\Delta\mathbf{U}^1{}_{p*(d+1)}$

        // 가중치 갱신

    $\mathbf{U}^2 = \mathbf{U}^2 - \rho\left(\frac{1}{t}\right)\Delta\mathbf{U}^2$ // 식 (3.21) - 평균 그레이디언트로 갱신

    $\mathbf{U}^1 = \mathbf{U}^1 - \rho\left(\frac{1}{t}\right)\Delta\mathbf{U}^1$ // 식 (3.21) - 평균 그레이디언트로 갱신

until (멈춤 조건)

batch

epoch

다층 퍼셉트론 예측

알고리즘 3-7 다층 퍼셉트론을 이용한 인식

입력: 테스트 샘플 $\mathbf{x}$ // 신경망의 가중치 $\mathbf{U}^1$과 $\mathbf{U}^2$는 이미 설정되었다고 가정함.
출력: 부류 y

$\mathbf{x} = (x_0, x_1, x_2, \cdots, x_d)^\mathrm{T}$로 확장하고, x_0과 z_0을 1로 설정한다.
$\mathbf{zsum} = \mathbf{U}^1\mathbf{x}$
$\tilde{\mathbf{z}} = \boldsymbol{\tau}(\mathbf{zsum})$ // 식 (3.13)
$\mathbf{osum} = \mathbf{U}^2\mathbf{z}$
$\mathbf{o} = \boldsymbol{\tau}(\mathbf{osum})$ // 식 (3.14)
$\mathbf{o}$에서 가장 큰 값을 가지는 노드에 해당하는 부류 번호를 y에 대입한다.