14기 정규세션

ToBig's 13기 이지용

Neural Network 기초

Contents

14기 정규세션

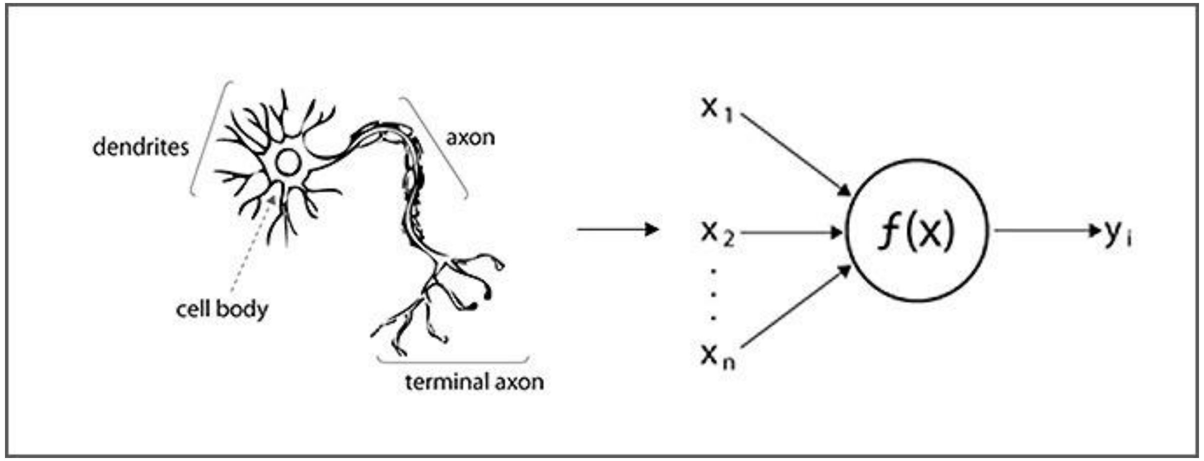
ToBig's 13기 이지용

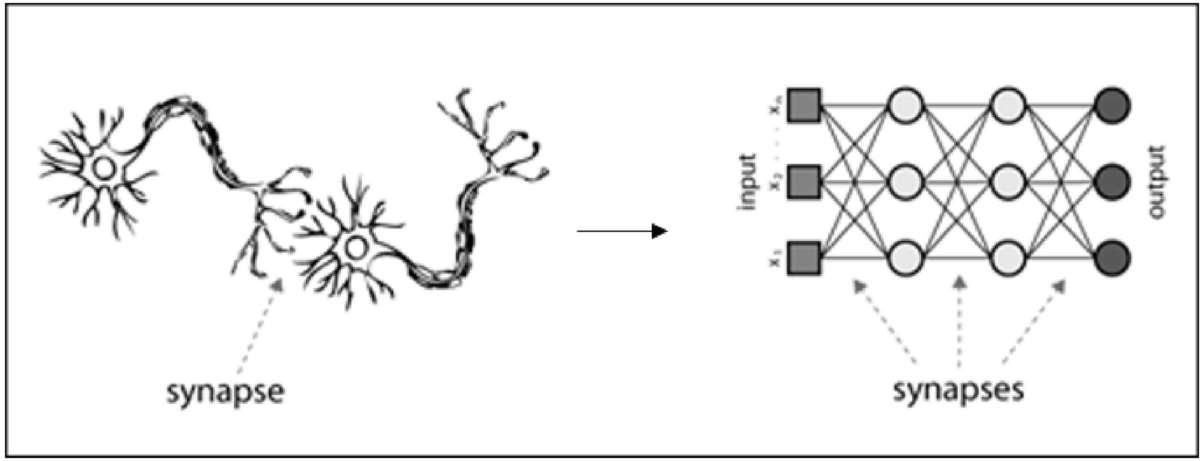
Neural Network

Unit 01 | Neural Network

Neural Network

생물학의 신경망에서 영감을 얻은 통계학적 기계학습 알고리즘

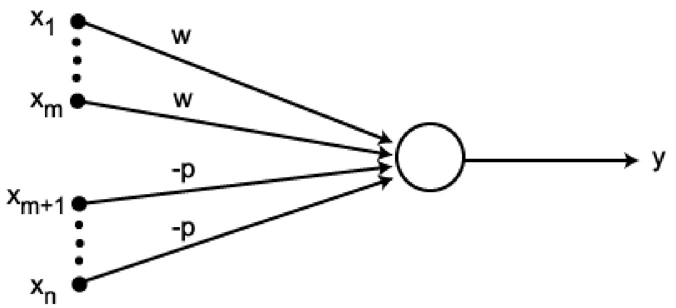




Neural Net의 역사 – 초기

McCulloch, W. S., & Pitts, W. (1943).
A logical calculus of the ideas immanent in nervous activity

인간의 뇌를 기계적으로 모델링한 최초의 논문



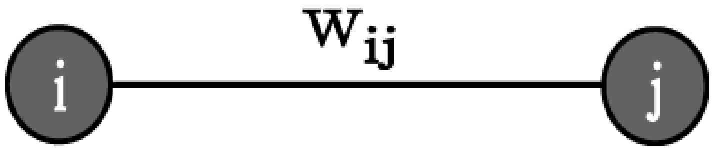
맥컬럭-피츠 모델

Unit 01 | Neural Network

Neural Net의 역사 - 초기

Hebb, D. O. (1949).
The organization of behavior; a neuropsychological theory

신경망은 서로 연결되어 있고, 지속적으로 학습하면서 네트워크를 수정한다



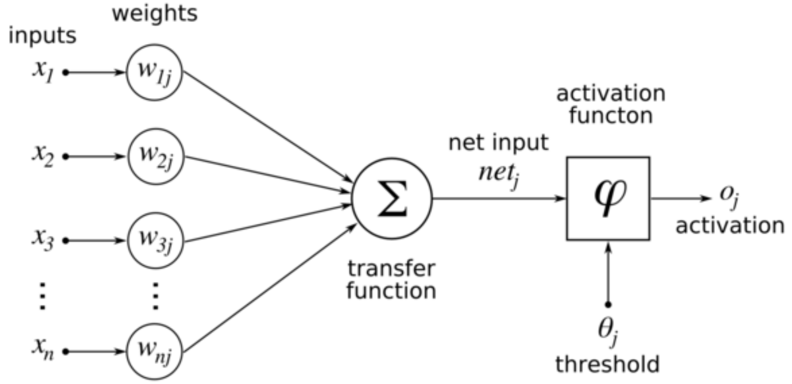
헵의 뉴런 연결

Unit 01 | Neural Network

Neural Net의 역사 – 1세대

Rosenblatt, F. (1958).
The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain

가장 단순한 형태의 계산에 의한 최초의 신경망 모델
Input과 Weight의 선형결합에 Activation 함수를 적용



Output

$$w_1x_1 + w_2x_2 + \dots + w_nx_n > \theta \Rightarrow 1$$

$$w_1x_1 + w_2x_2 + \dots + w_nx_n \leq \theta \Rightarrow 0$$

Neural Net의 역사 - 1세대

Widrow, B., & Hoff, M. (1960).
Adaptive Switching Circuits

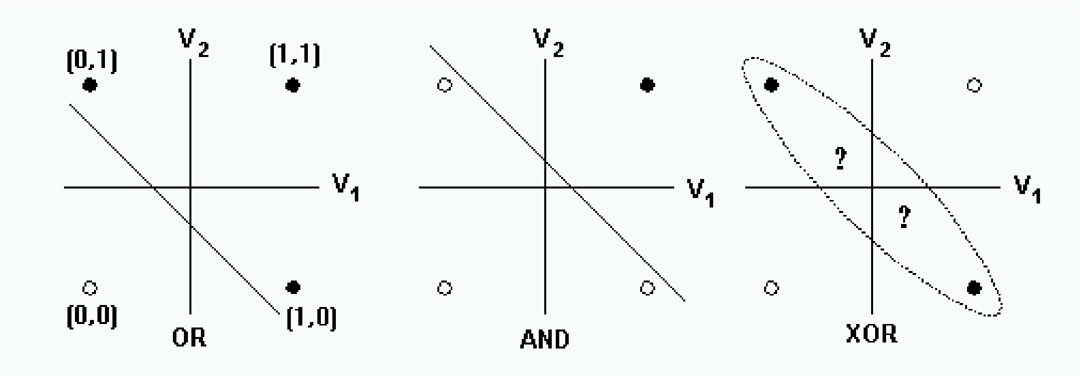
주어진 정보(오차)에 따라 단층 신경망의 가중치를 갱신하는 규칙
델타규칙(= Adaline, Widrow-Hoff 규칙)

$$w \leftarrow w + \eta(y - o)x$$

Neural Net의 역사 - 1세대

Minsky, M., & Papert, S. (1969).
Perceptrons: an introduction to computational geometry

선형 분류기인 퍼셉트론의 한계를 수학적으로 입증
단층 퍼셉트론 모델이 XOR 문제를 해결할 수 없다

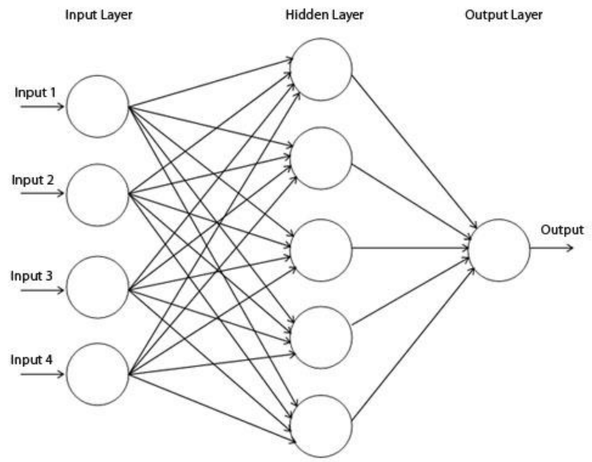


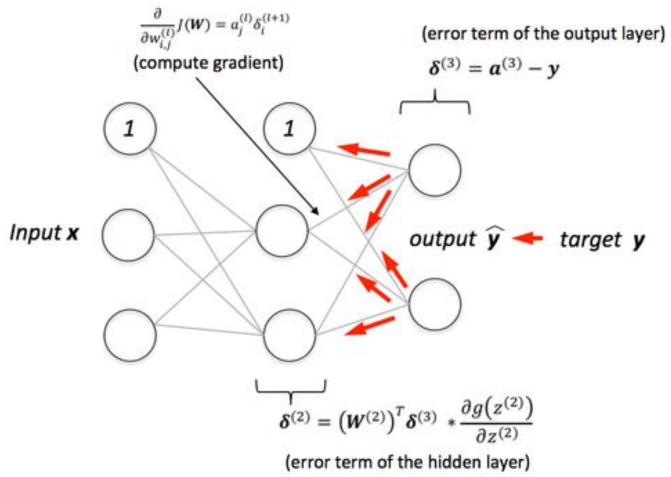
Unit 01 | Neural Network

Neural Net의 역사 - 2세대

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986).
Learning representations by back-propagating errors

은닉층을 추가시킨 다층 퍼셉트론이 XOR문제를 해결할 수 있음
이를 학습시키는 **오류 역전파** 방법



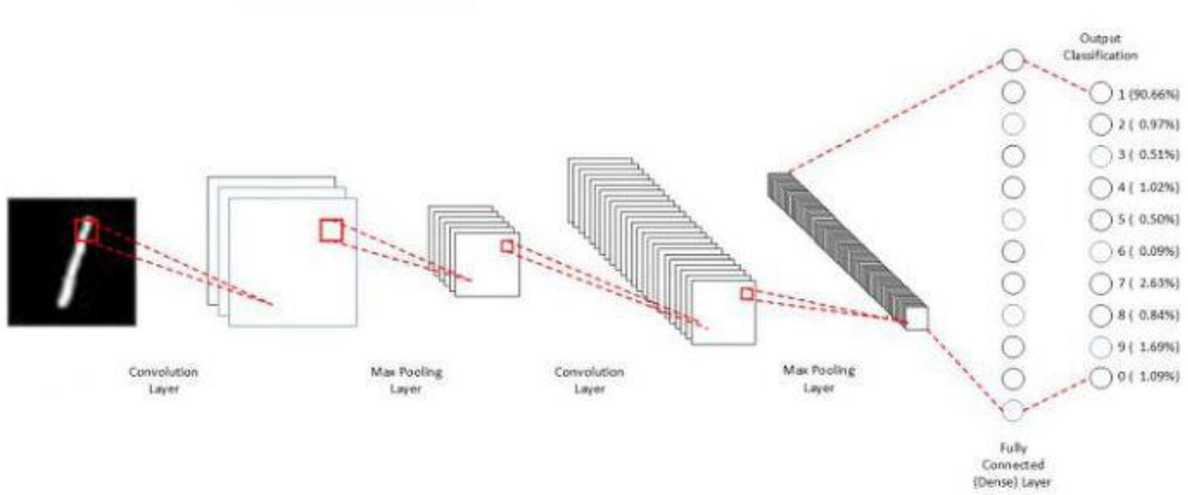


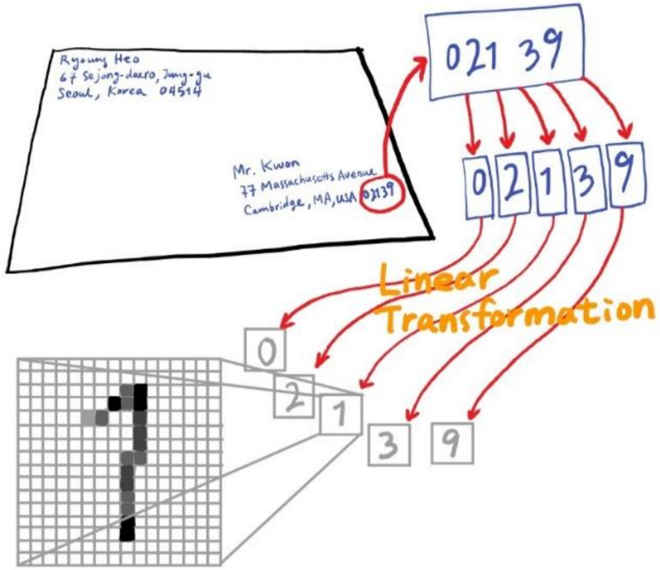
Unit 01 | Neural Network

Neural Net의 역사 – 2세대

LeCun, Y. et al. (1989).
Backpropagation Applied to Handwritten Zip Code Recognition

CNN을 이용한 손글씨 우편번호 인식. 이후 MNIST 데이터셋 구축





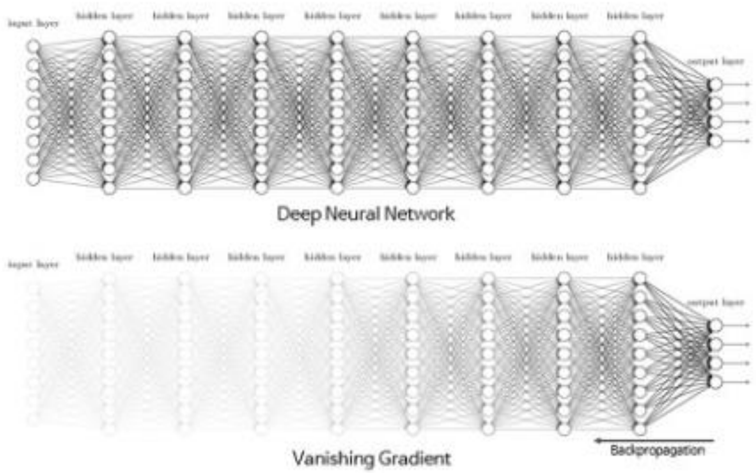
Unit 01 | Neural Network

Neural Net의 역사 - 2세대

Hochreiter, S. (1991).
Untersuchungen zu dynamischen neuronalen Netzen
Bengio, Y., Simard, P., & Frasconi, P. (1994).
Learning long-term dependencies with gradient descent is difficult

층이 깊어질수록 기울기 소실 문제가 발생함

부족한 데이터 셋과 과적합 문제
컴퓨터 성능의 한계

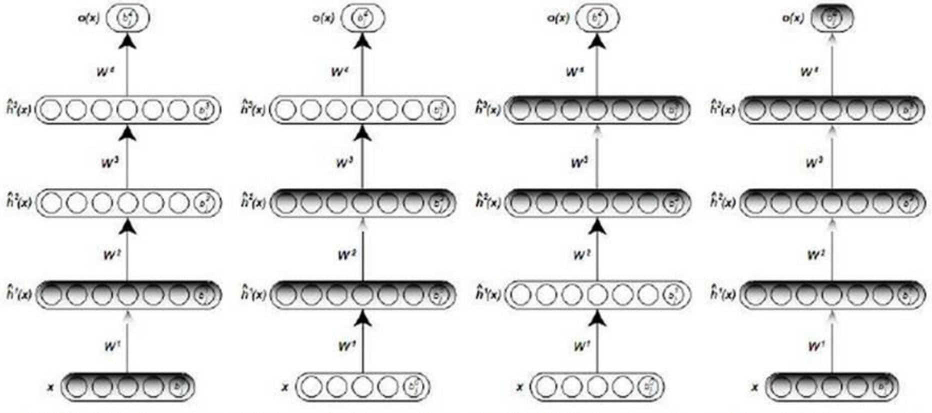


Unit 01 | Neural Network

Neural Net의 역사 – 3세대

Hinton, G. E., & Salakhutdinov, R. R. (2006).
Reducing the dimensionality of data with neural networks

제한된 볼츠만 머신을 기반으로 Label이 없는 데이터를 통한 사전 비지도 학습
기울기 소실, 과적합 문제를 극복
사전 학습으로 적절한 초기값 선정



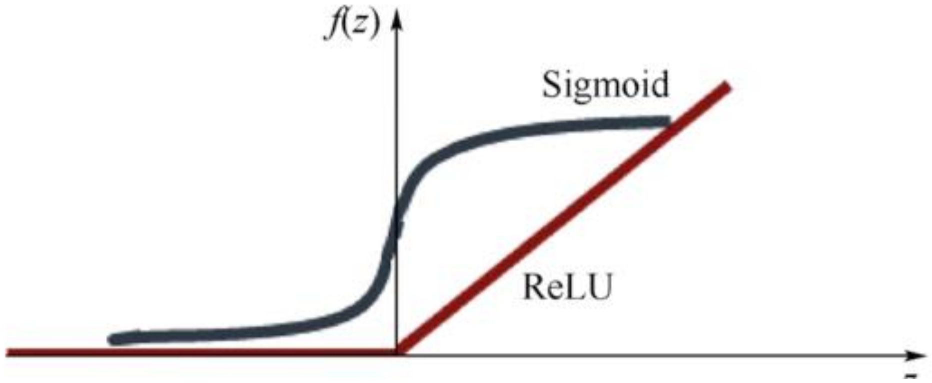
(a) First hidden layer pre-training (b) Second hidden layer pre-training (c) Third hidden layer pre-training (d) Fine-tuning of whole network

Unit 01 | Neural Network

Neural Net의 역사 – 3세대

Nair, V., & Hinton, G. E. (2010).
Rectified Linear Units Improve Restricted Boltzmann Machines
Glorot, X., Bordes, A. & Bengio, Y. (2011).
Deep Sparse Rectifier Neural Networks

ReLU 활성화 함수를 통한 기울기 소실 문제와 학습시간 문제 해결

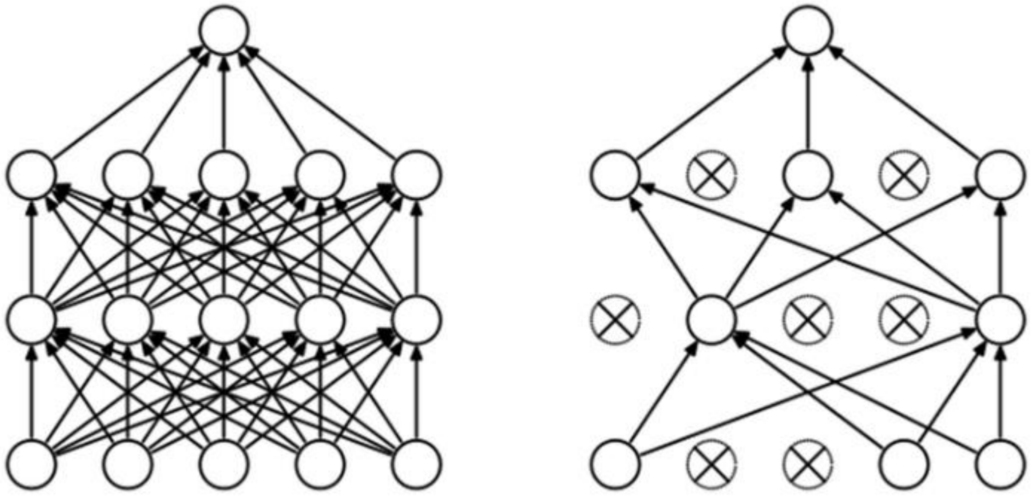


Unit 01 | Neural Network

Neural Net의 역사 – 3세대

Hinton, G. E., Srivastava, N, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. (2012). Improving neural networks by preventing co-adaptation of feature detectors

Dropout을 통한 과적합 방지

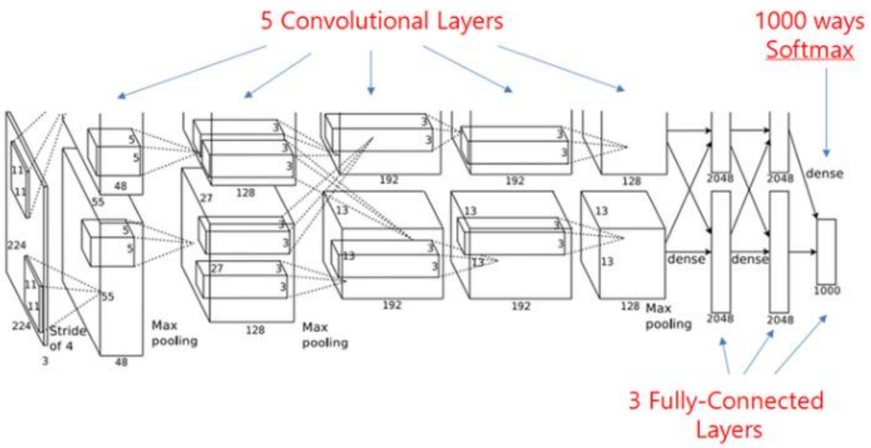


(a) Standard Neural Net (b) After applying dropout.

Neural Net의 역사 – 3세대

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012).
ImageNet Classification with Deep Convolutional Neural Networks

AlexNet
ReLU, Dropout, Augmetaion, GPU 연산 등을 활용해 ILSVRC 대회를 압도적으로 우승



Unit 01 | Neural Network

Neural Net의 역사 – Deep Learning

Lecun, Y., Bengio, Y., & Hinton, G. E. (2015).
Deep Learning

딥러닝은 탐지나 분류에 필요한 표현을 자동으로 발견하는 표현학습을 여러 레벨로 쌓은 것
Layer들의 Feature를 사람이 직접 구하지 않는다

이미지 인식, 음성 인식, 자연어 처리, 강화학습 등등 많은 분야에서 성과를 거두고 있다

DL의 알고리즘 중에는 DNN, RNN, CNN, GAN, DQN 등이 대표적이다

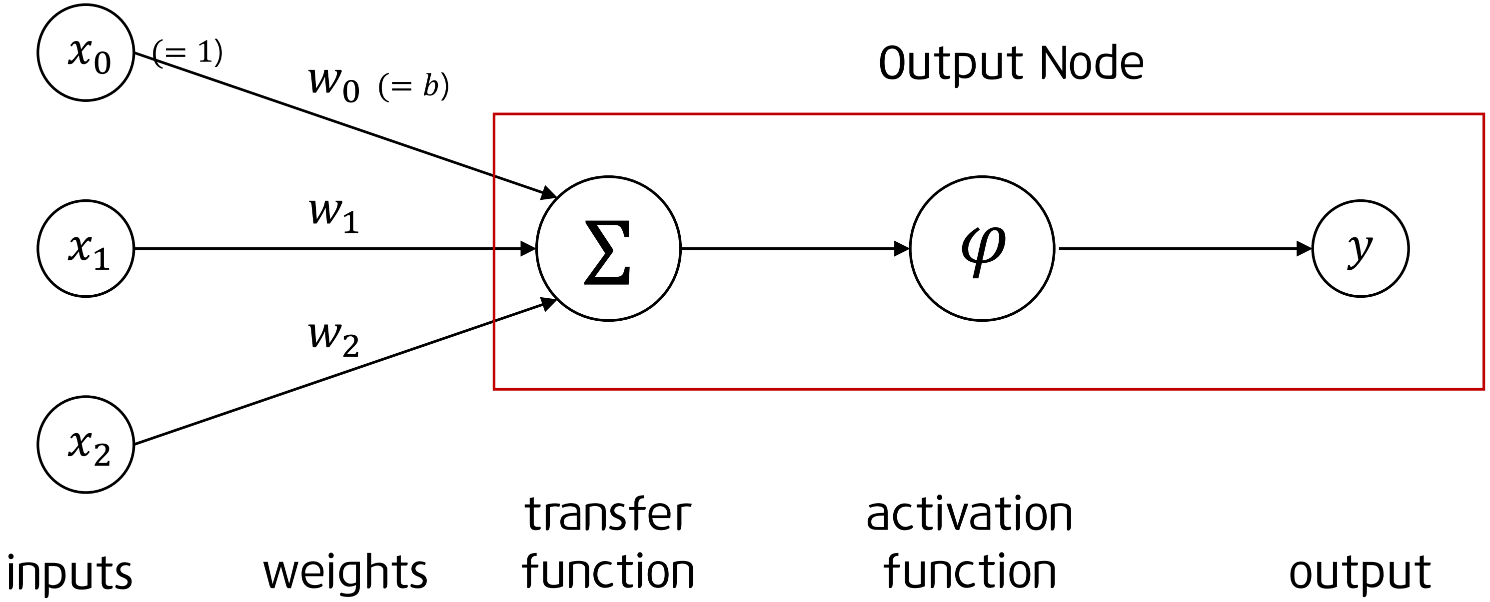
14기 정규세션

ToBig's 13기 이지용

Perceptron

Unit 02 | Perceptron

Perceptron의 구조



$$\varphi(w_0x_0 + w_1x_1 + w_2x_2) = y$$

$$\varphi(z) = \begin{cases} -1, & z < 0 \\ 1, & z \geq 0 \end{cases}$$

Unit 02 | Perceptron

Perceptron 수식 표현

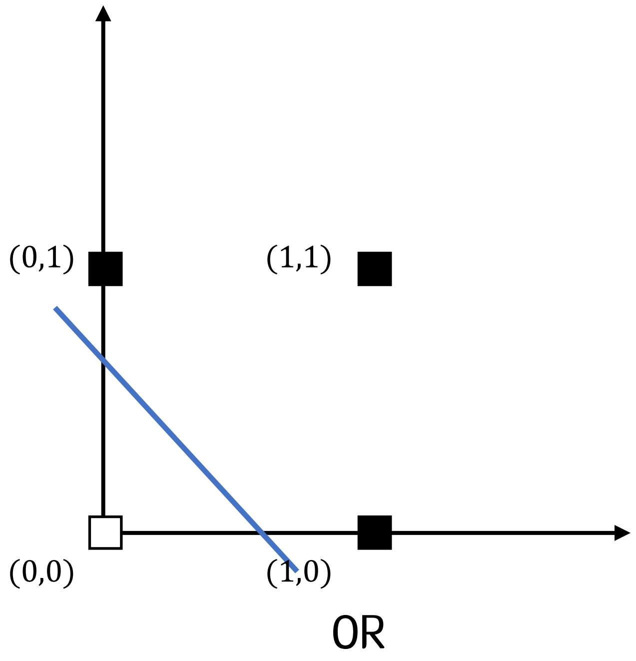
$$\varphi(w_0x_0 + w_1x_1 + w_2x_2) = y$$

$$\varphi(\sum w_jx_j) = y$$

$$\varphi(\boldsymbol{w}^T\boldsymbol{x}) = y$$

$$\varphi(\begin{pmatrix} w_0 & w_1 & w_2 \end{pmatrix}\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}) = y$$

Perceptron 분류 예시

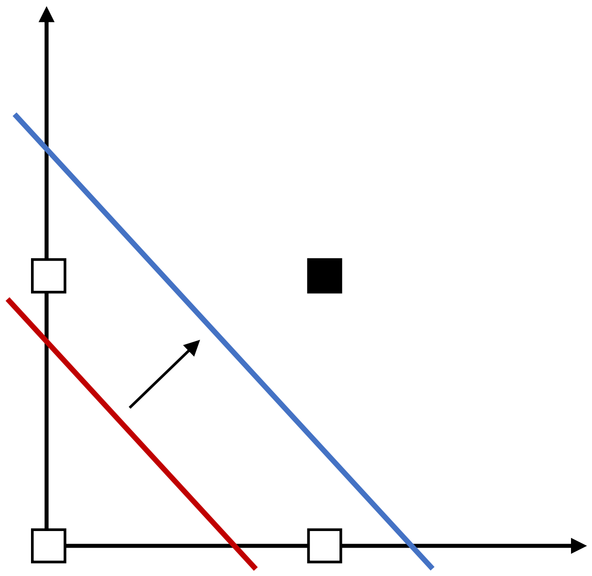


$$\varphi(w_0x_0 + w_1x_1 + w_2x_2) = y \qquad \varphi(z) = \begin{cases} -1, & z < 0 \\ 1, & z \geq 0 \end{cases}$$

$$w_0 = -0.4, w_1 = 0.6, w_2 = 0.6$$

$$\varphi(-0.4 + 0.6 * 0 + 0.6 * 0) = -1$$
$$\varphi(-0.4 + 0.6 * 0 + 0.6 * 1) = 1$$
$$\varphi(-0.4 + 0.6 * 1 + 0.6 * 0) = 1$$
$$\varphi(-0.4 + 0.6 * 1 + 0.6 * 1) = 1$$

Perceptron 학습, Adaline Gradient Descent 예시



AND

$$w_0 = -0.4, w_1 = 0.6, w_2 = 0.6$$

$$w_j \leftarrow w_j + \eta(y - o)x_j, \eta = 0.05$$

$$w_0 \leftarrow w_0 + 0.05((-1) - (1)) * 1$$
$$w_1 \leftarrow w_1 + 0.05((-1) - (1)) * 0$$
$$w_2 \leftarrow w_2 + 0.05((-1) - (1)) * 1$$

$$w_0 \leftarrow w_0 + 0.05((-1) - (1)) * 1$$
$$w_1 \leftarrow w_1 + 0.05((-1) - (1)) * 1$$
$$w_2 \leftarrow w_2 + 0.05((-1) - (1)) * 0$$

$$w_0 = -0.6, w_1 = 0.5, w_2 = 0.5$$

Perceptron 학습 규칙

$$J(w) = \frac{1}{2}\sum\left(y^{(i)} - \left(\boldsymbol{w}^T\boldsymbol{x}^{(i)}\right)\right)^2$$

$$\frac{\partial J}{\partial w_j} = -\sum\left(y^{(i)} - \left(\boldsymbol{w}^T\boldsymbol{x}^{(i)}\right)\right)\left(x_j^{(i)}\right)$$

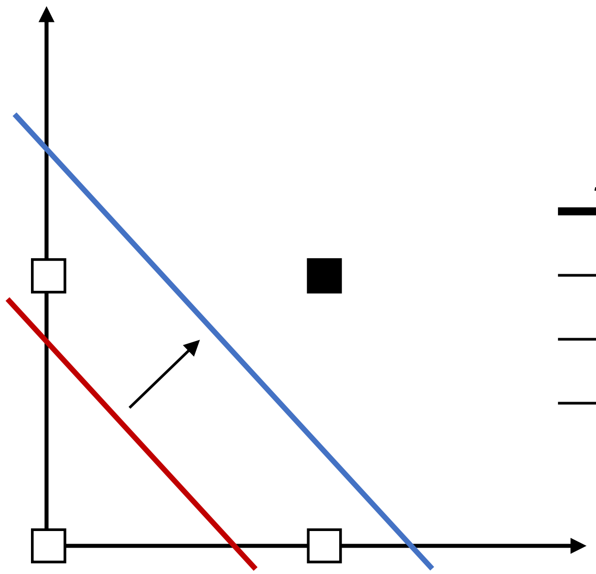
$$w_j \leftarrow w_j + \eta\left(y^{(i)} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}\right)x_j^{(i)}, \eta = 0.05$$

$$w_0 \leftarrow w_0 + 0.05 * (-2.8)$$
$$w_1 \leftarrow w_1 + 0.05 * (-1)$$
$$w_2 \leftarrow w_2 + 0.05 * (-1)$$

$$w_0 = -0.54, w_1 = 0.55, w_2 = 0.55$$

Adaline Gradient Descent

Adaline Gradient Descent 예시



AND

$w_0 = -0.4, w_1 = 0.6, w_2 = 0.6$

$x_0^{(i)}$	$x_1^{(i)}$	$x_2^{(i)}$	$y^{(i)}$	$\boldsymbol{w}^T\boldsymbol{x}^{(i)}$	$y^{(i)} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}$
1	0	0	−1	−0.4	−0.6
1	0	1	−1	0.2	−1.2
1	1	0	−1	0.2	−1.2
1	1	1	1	0.8	0.2

$$J(w) = \frac{1}{2}\sum\left(y^{(i)} - \left(\boldsymbol{w}^T\boldsymbol{x}^{(i)}\right)\right)^2$$

$$\frac{\partial J}{\partial w_j} = -\sum\left(y^{(i)} - \left(\boldsymbol{w}^T\boldsymbol{x}^{(i)}\right)\right)\left(x_j^{(i)}\right)$$

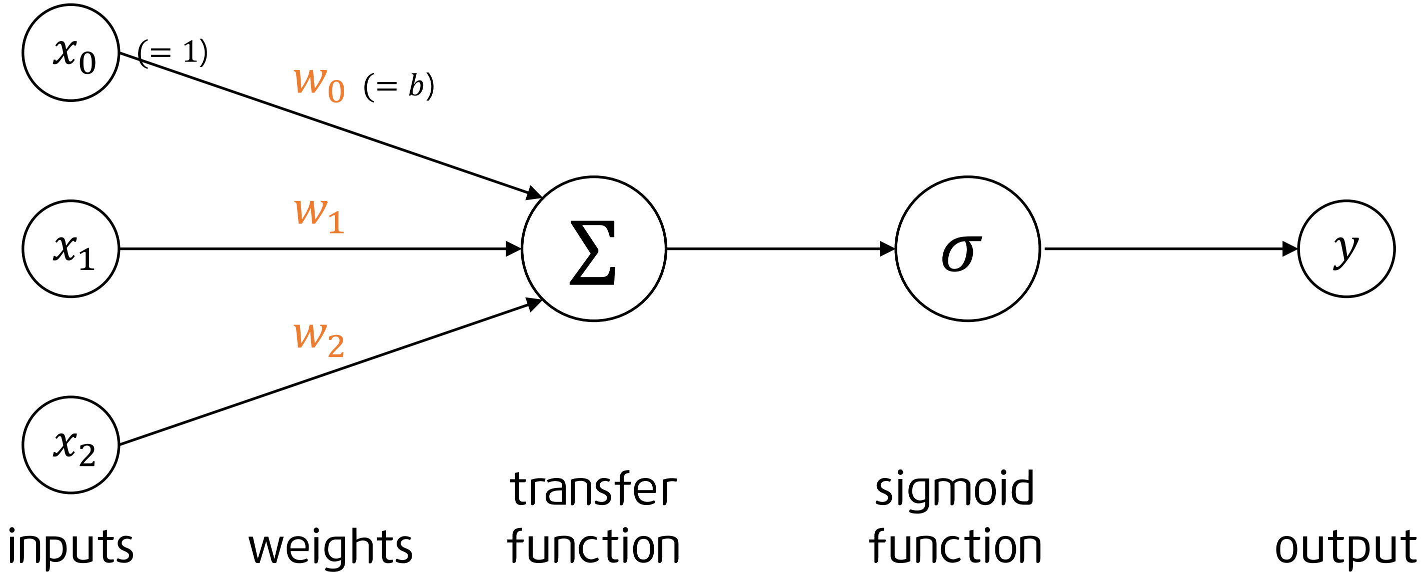
$$w_j \leftarrow w_j + \eta\left(y^{(i)} - \boldsymbol{w}^T\boldsymbol{x}^{(i)}\right)x_j^{(i)}, \eta = 0.05$$

$$w_0 \leftarrow w_0 + 0.05 * (-2.8)$$
$$w_1 \leftarrow w_1 + 0.05 * (-1)$$
$$w_2 \leftarrow w_2 + 0.05 * (-1)$$

$w_0 = -0.54, w_1 = 0.55, w_2 = 0.55$

Adaline Gradient Descent

Logistic Regression



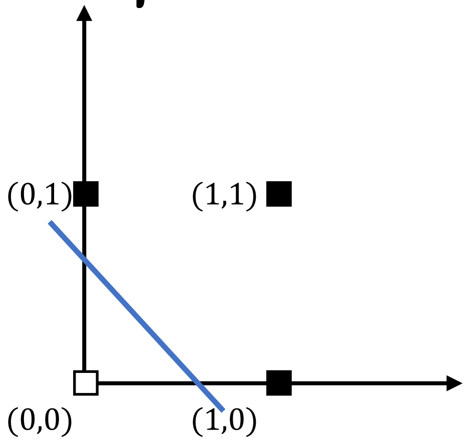
$$\phi(w_0x_0 + w_1x_1 + w_2x_2) = y$$

$$J(w) = -\sum\{y^{(i)}\log p^{(i)} + (1 - y^{(i)})\log(1 - p^{(i)})\}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

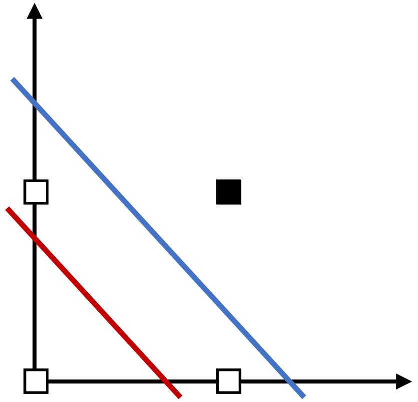
$$\frac{\partial J}{\partial w_j} = -\sum(y^{(i)} - (p^{(i)})\ )x_j^{(i)}, w_j \leftarrow w_j + \eta(y^{(i)} - p^{(i)})x_j^{(i)}$$

Perceptron XOR 분류 문제



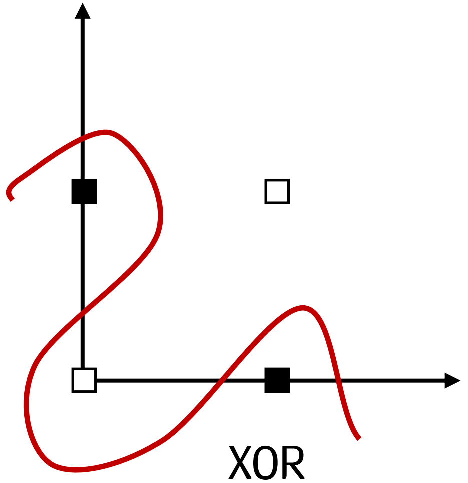
OR

$$w_0 = -0.4, w_1 = 0.6, w_2 = 0.6$$
$$\varphi(-0.4 + 0.6 * 0 + 0.6 * 0) = -1$$
$$\varphi(-0.4 + 0.6 * 0 + 0.6 * 1) = 1$$
$$\varphi(-0.4 + 0.6 * 1 + 0.6 * 0) = 1$$
$$\varphi(-0.4 + 0.6 * 1 + 0.6 * 1) = 1$$



AND

$$w_0 = -0.6, w_1 = 0.5, w_2 = 0.5$$
$$\varphi(-0.6 + 0.5 * 0 + 0.5 * 0) = -1$$
$$\varphi(-0.6 + 0.5 * 0 + 0.5 * 1) = -1$$
$$\varphi(-0.6 + 0.5 * 1 + 0.5 * 0) = -1$$
$$\varphi(-0.6 + 0.5 * 1 + 0.5 * 1) = 1$$



XOR

선형 분류 불가능

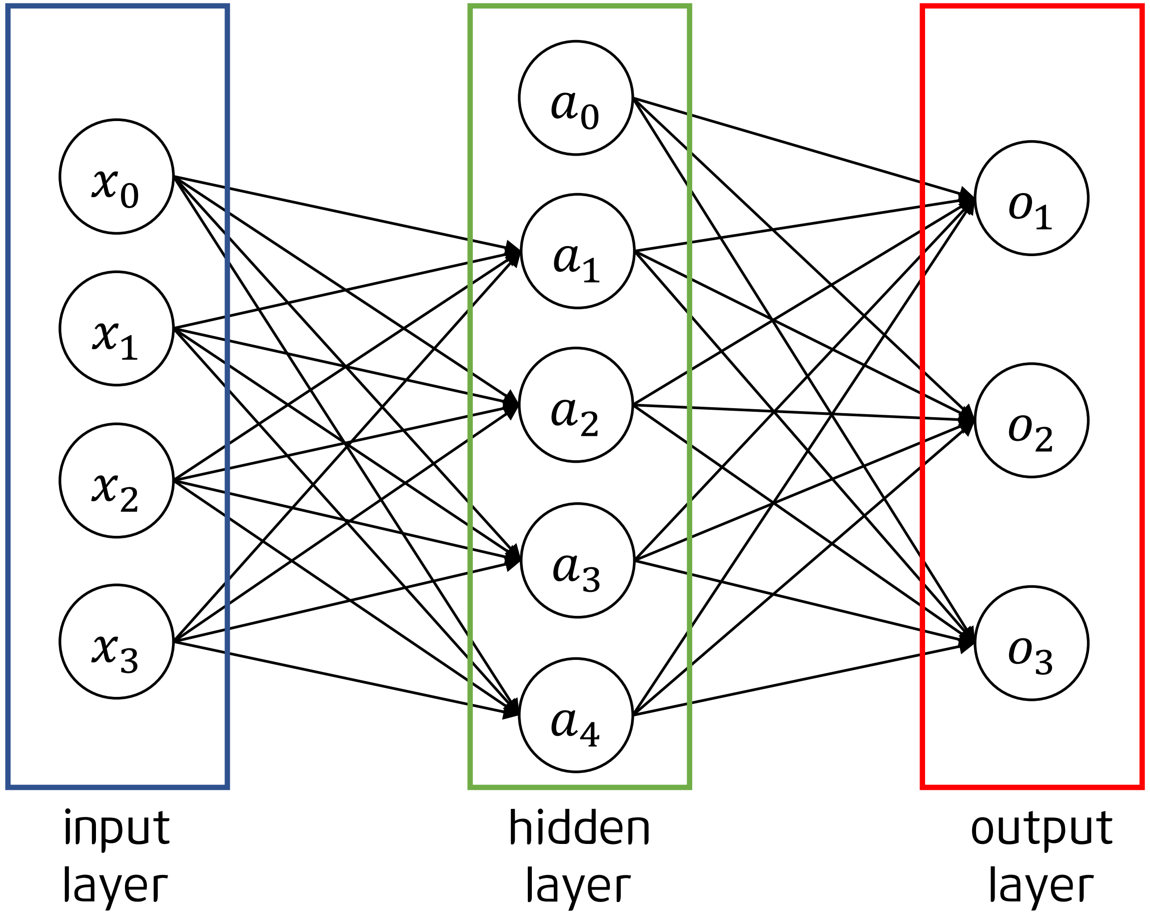
14기 정규세션

ToBig's 13기 이지용

Multilayer Perceptron

Unit 03 | Multilayer Perceptron

Multilayer Perceptron의 구조



d+1개의 입력 노드

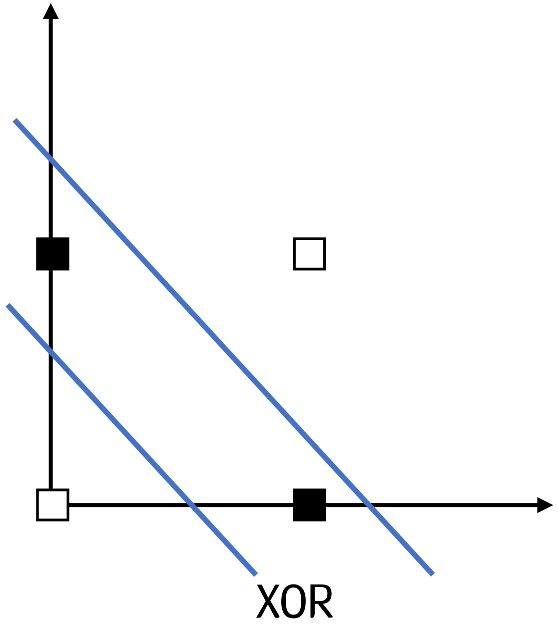
h+1개의 은닉층 노드

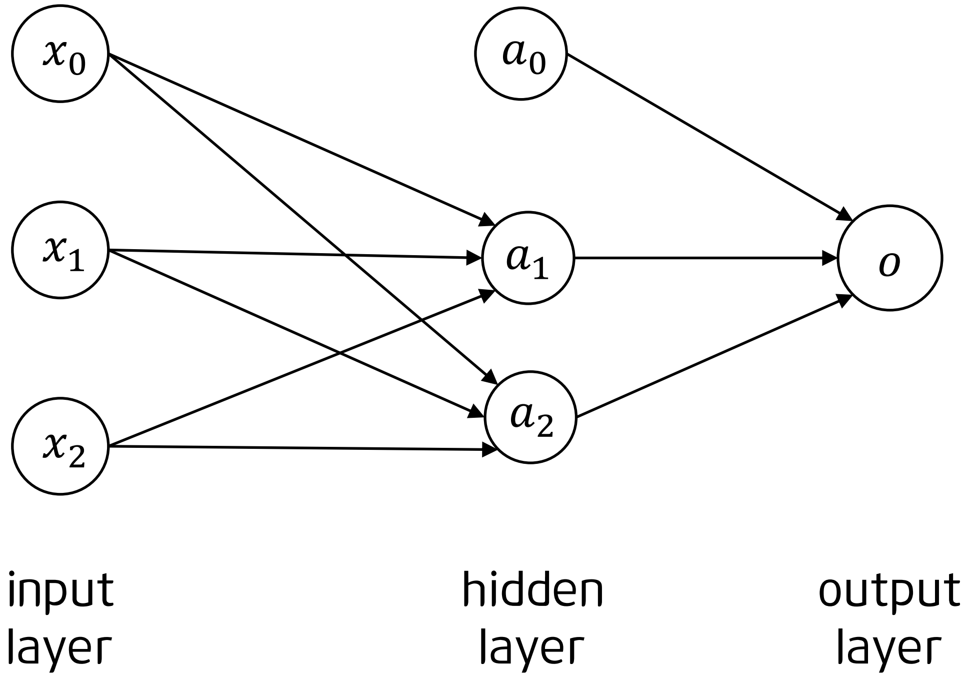
c개의 출력 노드(부류 개수)

(d+1)*h + (h+1)*c 개의 가중치 개수
(=파라미터의 개수)
(2layer의 경우)

Unit 03 | Multilayer Perceptron

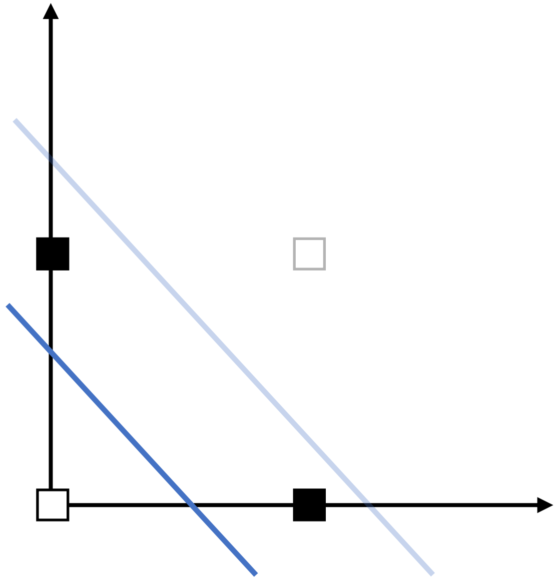
Multilayer Perceptron 으로 XOR 문제 해결

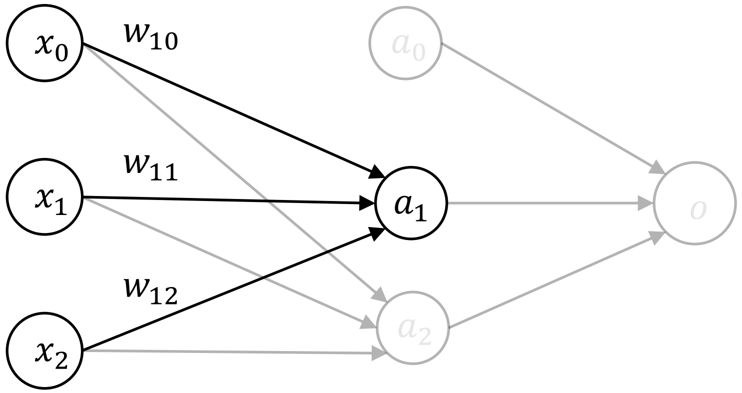




Unit 03 | Multilayer Perceptron

Multilayer Perceptron 으로 XOR 문제 해결



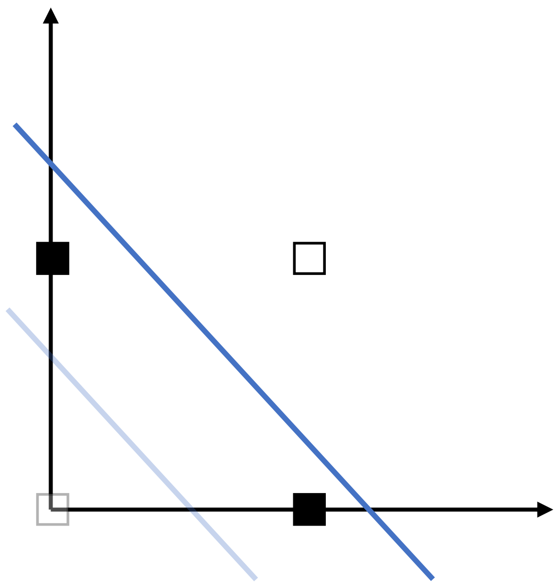


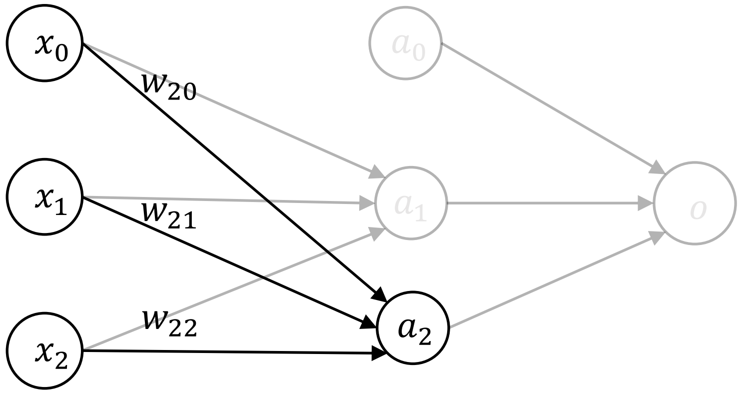
$$\varphi(\boldsymbol{w}_1^T \boldsymbol{x}) = \varphi\left(\begin{pmatrix} w_{10} & w_{11} & w_{12} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} \right) = a_1$$

$$w_{10} = -0.4, w_{11} = 0.6, w_{12} = 0.6$$

Unit 03 | Multilayer Perceptron

Multilayer Perceptron 으로 XOR 문제 해결



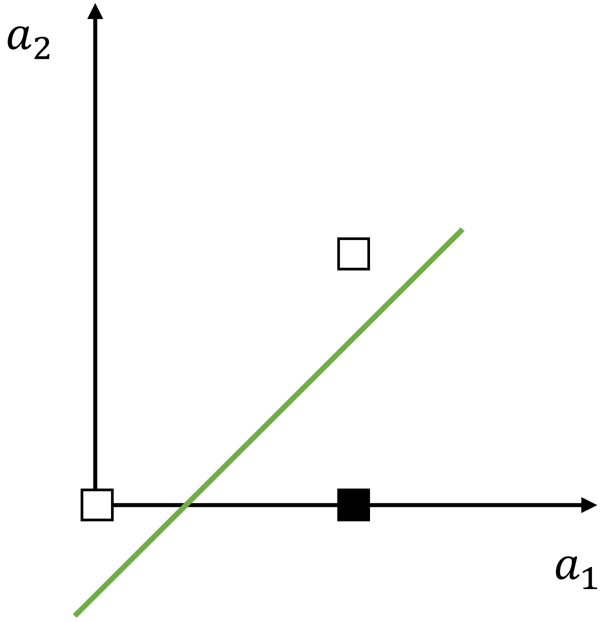


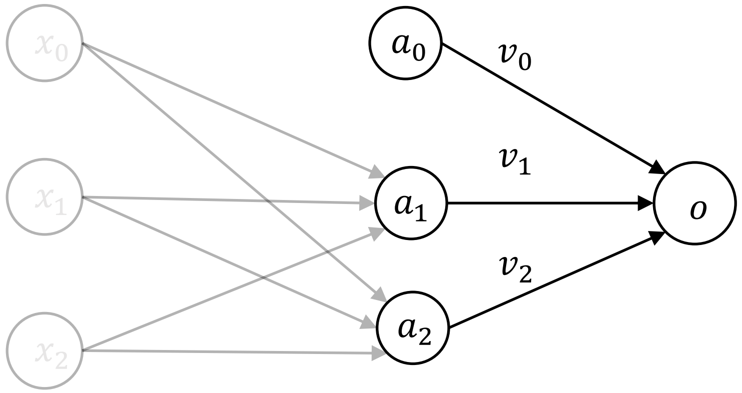
$$\varphi(\boldsymbol{w}_2^T\boldsymbol{x}) = \varphi\left(\begin{pmatrix} w_{20} & w_{21} & w_{22} \end{pmatrix}\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}\right) = a_2$$

$$w_{20} = 0.6, w_{21} = -0.5, w_{22} = -0.5$$

Unit 03 | Multilayer Perceptron

Multilayer Perceptron 으로 XOR 문제 해결

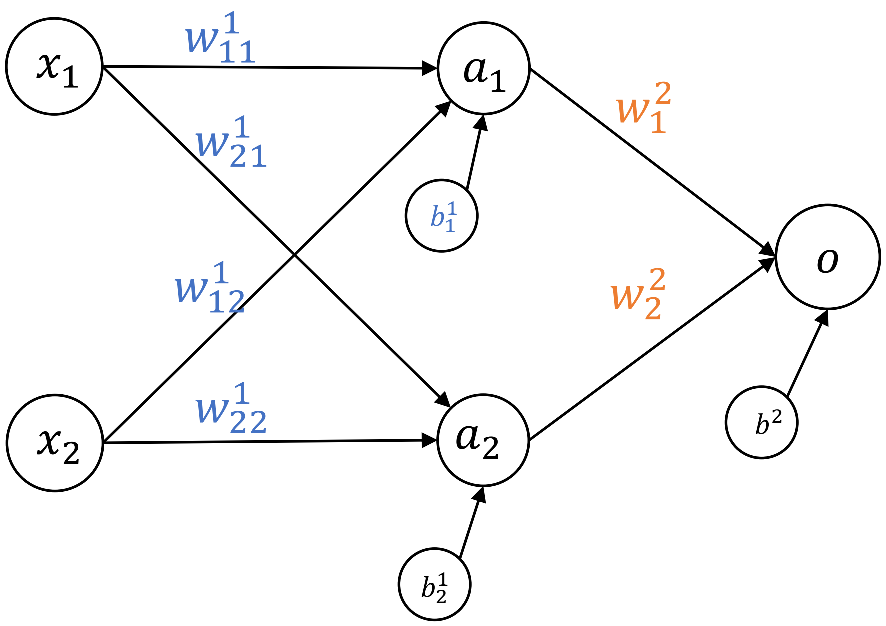




$$\varphi(\boldsymbol{v}^T\boldsymbol{x}) = \varphi\left(\begin{pmatrix} v_0 & v_1 & v_2 \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}\right) = y$$

$$v_0 = -0.4, v_1 = 0.6\ v_2 = -0.6$$

Multilayer Perceptron 행렬 계산



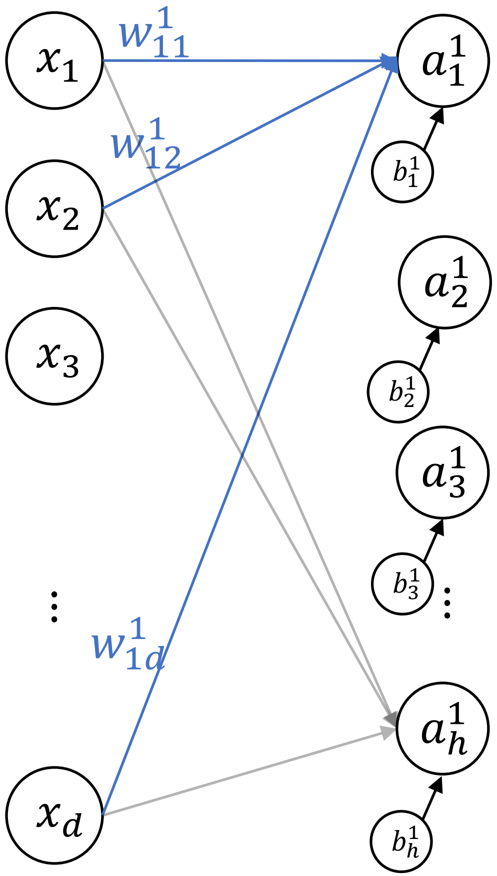
$$\varphi({W^1}^T \boldsymbol{x}^{(i)}) = \varphi\left(\begin{pmatrix} w_{11}^1 & w_{12}^1 \\ w_{21}^1 & w_{22}^1 \end{pmatrix}\begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix} + \begin{pmatrix} b_1^1 \\ b_2^1 \end{pmatrix}\right) = \begin{pmatrix} a_1^{(i)} \\ a_2^{(i)} \end{pmatrix}$$

$$\boldsymbol{a}^{(i)} = \varphi(W^1 \boldsymbol{x}^{(i)} + \boldsymbol{B}^1)$$

$$\varphi({W^2}^T \boldsymbol{a}^{(i)}) = \varphi\left(\begin{pmatrix} w_1^2 & w_2^2 \end{pmatrix}\begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix} + b_1^2\right) = y^{(i)}$$

$$y^{(i)} = \varphi(W^2 \boldsymbol{a}^{(i)} + \boldsymbol{B}^2)$$

Multilayer Perceptron 행렬 표현



$$\boldsymbol{w}_i^k = \left(w_{i1}^k, w_{i2}^k, \cdots, w_{id}^k\right)^T$$

$$W^k = \left(\boldsymbol{w}_1^k \quad \boldsymbol{w}_2^k \quad \cdots \quad \boldsymbol{w}_h^k\right)^T$$

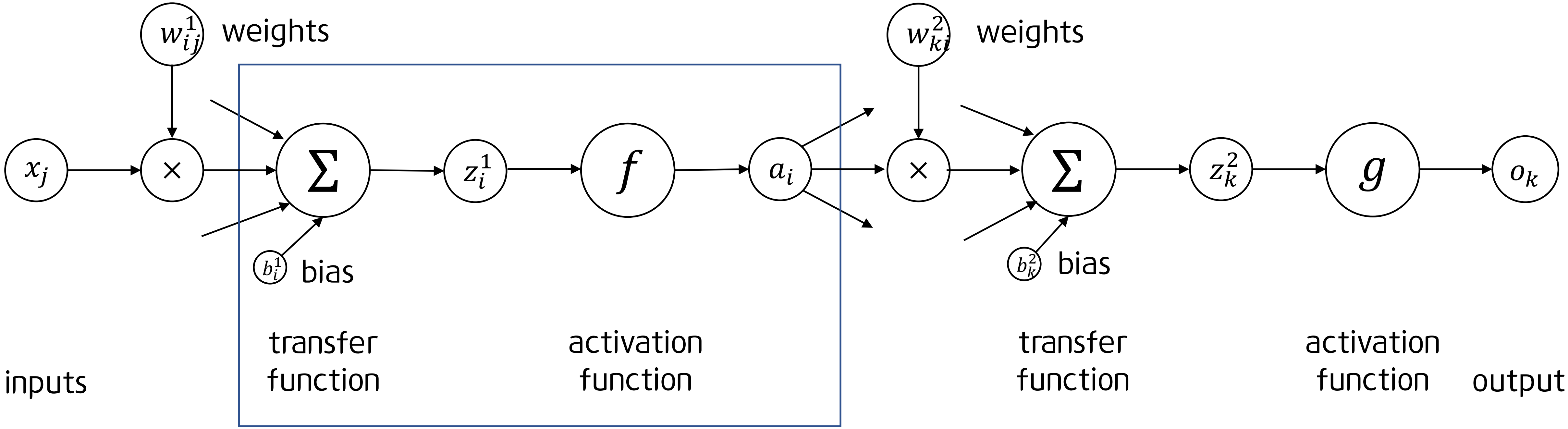
$$W^k = \begin{pmatrix} w_{11}^k & w_{21}^k & \cdots & w_{i1}^k & \cdots & w_{h1}^k \\ w_{12}^k & w_{22}^k & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ w_{1j}^k & & & w_{ij}^k & & w_{hj}^k \\ \vdots & & & & \ddots & \vdots \\ w_{1d}^k & \cdots & \cdots & w_{id}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$W^k \boldsymbol{x} + \boldsymbol{b}^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} + \begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix}$$

h∗d d∗1 h∗1

Unit 03 | Multilayer Perceptron

단일 노드들의 동작(순전파)



A node of a Hidden Layer

14기 정규세션

ToBig's 13기 이지용

Activation, Loss, Derivative

Unit 04 | Activation, Loss, Derivative

이번 강의에서의 기호 표기

i번 데이터의 j번 원소 $x_j^{(i)}$ 전체 데이터 셋 중 i번

Input data

L번 레이어의 i번 노드
$L-1$번 레이어의 j번 노드 w_{ij}^{L} L번 레이어

Weight

$x_0, a_0 = 1$ b_i^L w_{i0}^L으로 표현

Bias

z_i^L

WiX+bi

$sigmoid, ReLU$ f Hidden Layer에서
z_i^L 를 a_i^L 로 변환

Activation Function

a_i^L

f(z)

$softmax, linear$ g Output Layer에서
z를 o로 변환

Output Activation Function

k번 category $o_k^{(i)}$

Output

$cross\ entropy,$
$mean\ square\ error$ J

Loss Function

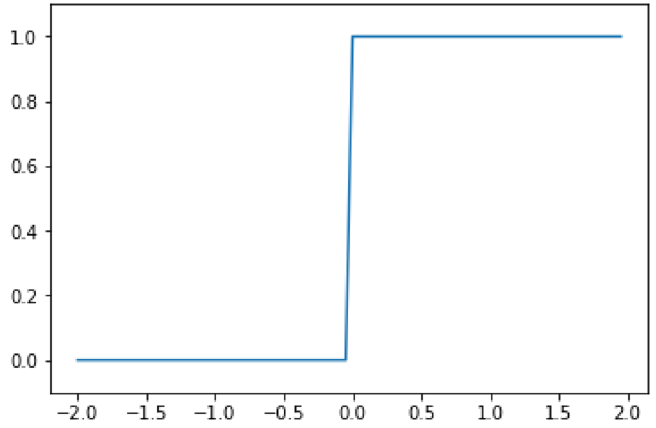
Activation Function

각 층마다의 특성을 가지기 위해서는 Activation Function이 필요하다
특히 non-linear한 분류와 은닉층을 쌓기 위해 비선형 함수를 사용해야함

$$f(x) = Wx + b$$
$$f\big(f(x)\big) = (Wx + b)x + b = Kx + b$$

Activation Function – Step Function

Threshold를 초과하면 1, 아니면 -1
0일때 미분이 불가능하고 미분값이 0이라 쓰이지 않는다

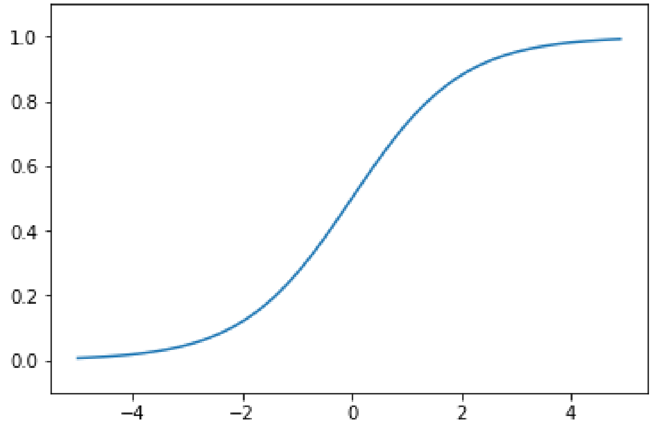


$$\varphi(z) = \begin{cases} -1, & z < 0 \\ 1, & z \geq 0 \end{cases}$$

Unit 04 | Activation, Loss, Derivative

Activation Function – Sigmoid Function

Logistic Function. 0~1의 출력 값을 갖는다



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

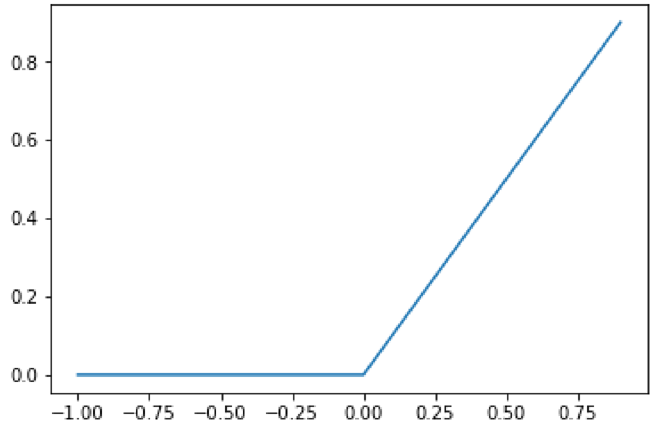
Activation Function – Sigmoid Function

Vanishing Gradient
은닉층이 깊어질수록 학습시 Sigmoid의 미분값이 계속 곱해짐
-> 미분값이 0에 수렴하는 결과

$$\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big) \leq 1/4$$

Activation Function – Rectified Linear Unit(ReLU)

Vanishing Gradient 문제를 해결하고 연산속도가 빠른 장점



$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} = \max(0, x), \qquad f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

Unit 04 | Activation, Loss, Derivative

Activation Function – Softmax

Sigmoid 함수를 Multi-class Classification에 맞게 일반화
분류 신경망의 출력층에서 많이 사용함

$$S(z) = \left(\frac{e^{z_1}}{\sum_{k=1}^{K} e^{z_k}}, \frac{e^{z_2}}{\sum_{k=1}^{K} e^{z_k}}, \dots, \frac{e^{z_K}}{\sum_{k=1}^{K} e^{z_k}} \right)$$

$$p_i = S_i(z) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \qquad \sum S_i(z) = 1$$

Loss Function – Mean Square Error

신경망의 학습을 위해 출력값과 실제값의 차이를 계산한 Loss Function을 정의해야 한다
연속형 변수에는 MSE가 많이 사용된다

$$MSE = \frac{1}{2}\sum(y_i - o_i)^2$$

$$\frac{\partial MSE}{\partial o_i} = -(y_i - o_i)$$

Loss Function – Cross Entropy

범주형 변수에는 CE가 많이 사용된다
MSE와 CE의 차이는
모든 output에 대해 오차를 계산하는지 vs 실제값이 1인 경우의 오차만을 계산하는지

$$CE = -\sum y_k \log(p_k)$$

이때 y_k는 하나의 class에 대해서만 1이므로
Σ연산이 모두 진행될 필요는 없다
Softmax로 p값을 계산할 경우

$$\boldsymbol{y} = (y_1, \ldots, y_k, \ldots) = (0,0, \ldots, 1, \ldots 0)$$
$$\boldsymbol{p} = (p_1, \ldots, p_k, \ldots) = \left(S_1(z_1), S_2(z_2), \ldots, S_k(z_k), \ldots S_K(z_K)\right)$$

Loss Function – Cross Entropy Derivative(Softmax)

$$Cross\ Entropy = J = -\sum y_j \log(p_j) = -\sum y_j \log\left(S_j(z)\right)$$

$$\frac{\partial J}{\partial z_i} = -\sum_j \left\{ y_j \frac{\partial}{\partial p_j} \log(p_j) \frac{\partial p_j}{\partial z_i} \right\}$$

$$= -\sum_j \left\{ \frac{y_j}{p_j} \frac{\partial p_j}{\partial z_i} \right\}$$

$$= -y_i(1 - p_i) + \sum_{i \neq j} y_j p_i$$

$$= p_i - y_i (= -(y_i - p_i))$$

$$\frac{\partial p_i}{\partial z_i} = \frac{e^{z_i} \sum e^{z_k} - e^{z_i} e^{z_i}}{(\sum e^{z_k})^2}$$

$$= \frac{e^{z_i}}{\sum e^{z_k}} \left(\frac{\sum e^{z_k} - e^{z_i}}{\sum e^{z_k}} \right)$$

$$= p_i(1 - p_i)$$

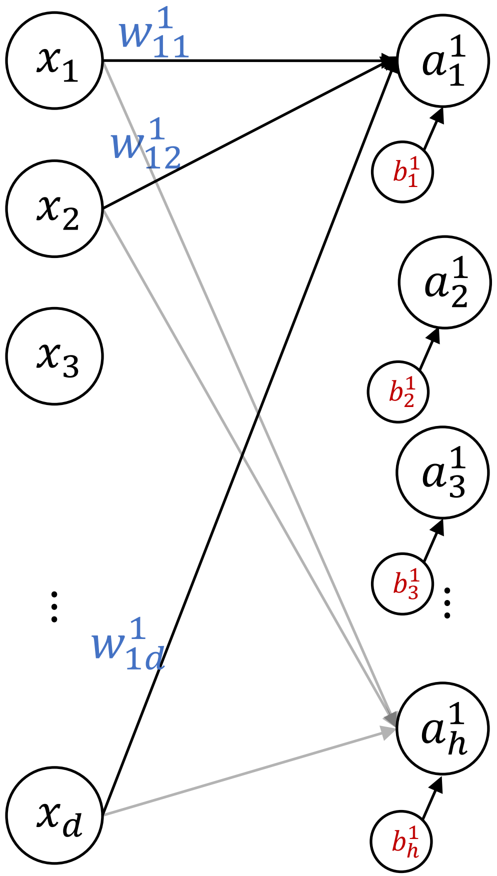
$$\frac{\partial p_j}{\partial z_i} = -\frac{e^{z_j}}{\sum e^{z_k}} \frac{e^{z_i}}{\sum e^{z_k}}, \qquad (i \neq j)$$

$$= -p_i p_j$$

* $\boldsymbol{y} = (y_1, \ldots, y_k, \ldots) = (0,0, \ldots, 1, \ldots 0),$

* $\frac{\partial J}{\partial z} = \begin{pmatrix} p_1 - y_1 \\ \vdots \\ p_k - y_k \end{pmatrix}$

Multilayer Perceptron 행렬 표현



$$\boldsymbol{w}_i^k = \left(w_{i1}^k, w_{i2}^k, \cdots, w_{id}^k\right)^T$$

$$W^k = \left(\boldsymbol{w}_1^k \quad \boldsymbol{w}_2^k \quad \cdots \quad \boldsymbol{w}_h^k\right)^T$$

$$W^k = \begin{pmatrix} w_{11}^k & w_{21}^k & \cdots & w_{i1}^k & \cdots & w_{h1}^k \\ w_{12}^k & w_{22}^k & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ w_{1j}^k & & & w_{ij}^k & & w_{hj}^k \\ \vdots & & & & \ddots & \vdots \\ w_{1d}^k & \cdots & \cdots & w_{id}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$W^k\boldsymbol{x} + \boldsymbol{b}^k = \underset{h*d}{\begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix}} \underset{d*1}{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}} + \underset{h*1}{\begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix}}$$

Unit 04 | Activation, Loss, Derivative

Scalar, Vector, Matrix

$$W^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} = \begin{pmatrix} w_{11}^k & \cdots & w_{h1}^k \\ \vdots & \ddots & \vdots \\ w_{1d}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$\boldsymbol{z}^k = W^k\boldsymbol{x} + \boldsymbol{b}^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} + \begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix} = \begin{pmatrix} z_1^k \\ z_2^k \\ \vdots \\ z_h^k \end{pmatrix}$$

$$z_1^k = w_{11}^k x_1 + w_{12}^k x_2 + \cdots + w_{1d}^k x_d + b_1^k = \boldsymbol{w}_1^{k^T}\boldsymbol{x}$$

$$z_i^k = w_{i1}^k x_1 + w_{i2}^k x_2 + \cdots + w_{id}^k x_d + b_i^k = \boldsymbol{w}_i^{k^T}\boldsymbol{x}$$

Unit 04 | Activation, Loss, Derivative

Scalar, Vector Derivative

$$\frac{\partial z_1^k}{\partial w_{11}^k} = x_1, \qquad \frac{\partial z_i^k}{\partial w_{ij}^k} = x_j$$

$$\frac{\partial z_i}{\partial \boldsymbol{w}_i^{k^T}} = \boldsymbol{x}^T, \qquad \frac{\partial z_i}{\partial \boldsymbol{w}_i^k} = \begin{pmatrix} \frac{\partial z_i}{\partial w_{i1}^k} \\ \vdots \\ \frac{\partial z_1}{\partial w_{id}^k} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} = \boldsymbol{x}$$

$$\frac{\partial z_i}{\partial x_j} = w_{ij}^k, \qquad \frac{\partial \boldsymbol{z}}{\partial x_j} = \begin{pmatrix} \frac{\partial z_1}{\partial x_j} \\ \vdots \\ \frac{\partial z_h}{\partial x_j} \end{pmatrix} = \begin{pmatrix} w_{11}^k \\ \vdots \\ w_{h1}^k \end{pmatrix}$$

Matrix Derivative

$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_d} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial z_h}{\partial x_1} & \cdots & \cdots & \frac{\partial z_h}{\partial x_d} \end{pmatrix} = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} = W^k$$

Unit 04 | Activation, Loss, Derivative

Scalar, Vector, Matrix, Derivative

$$W^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} = \begin{pmatrix} w_{11}^k & \cdots & w_{h1}^k \\ \vdots & \ddots & \vdots \\ w_{1d}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$\boldsymbol{z}^k = W^k \boldsymbol{x} + \boldsymbol{b}^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} + \begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix} = \begin{pmatrix} z_1^k \\ z_2^k \\ \vdots \\ z_h^k \end{pmatrix}$$

$$z_1^k = w_{11}^k x_1 + w_{12}^k x_2 + \cdots + w_{1d}^k x_d + b_1^k = \boldsymbol{w}_1^{k^T} \boldsymbol{x}$$

$$z_i^k = w_{i1}^k x_1 + w_{i2}^k x_2 + \cdots + w_{id}^k x_d + b_i^k = \boldsymbol{w}_i^{k^T} \boldsymbol{x}$$

$$\frac{\partial z_1^k}{\partial w_{11}^k} = x_1, \quad \frac{\partial z_i^k}{\partial w_{ij}^k} = x_j$$

$$\frac{\partial z_i}{\partial \boldsymbol{w}_i^{k^T}} = \boldsymbol{x}^T, \qquad \frac{\partial z_i}{\partial \boldsymbol{w}_i^k} = \begin{pmatrix} \frac{\partial z_i}{\partial w_{i1}^k} \\ \vdots \\ \frac{\partial z_1}{\partial w_{id}^k} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} = \boldsymbol{x}$$

$$\frac{\partial z_i}{\partial x_j} = w_{ij}^k, \qquad \frac{\partial \boldsymbol{z}}{\partial x_j} = \begin{pmatrix} \frac{\partial z_1}{\partial x_j} \\ \vdots \\ \frac{\partial z_h}{\partial x_j} \end{pmatrix} = \begin{pmatrix} w_{11}^k \\ \vdots \\ w_{h1}^k \end{pmatrix}$$

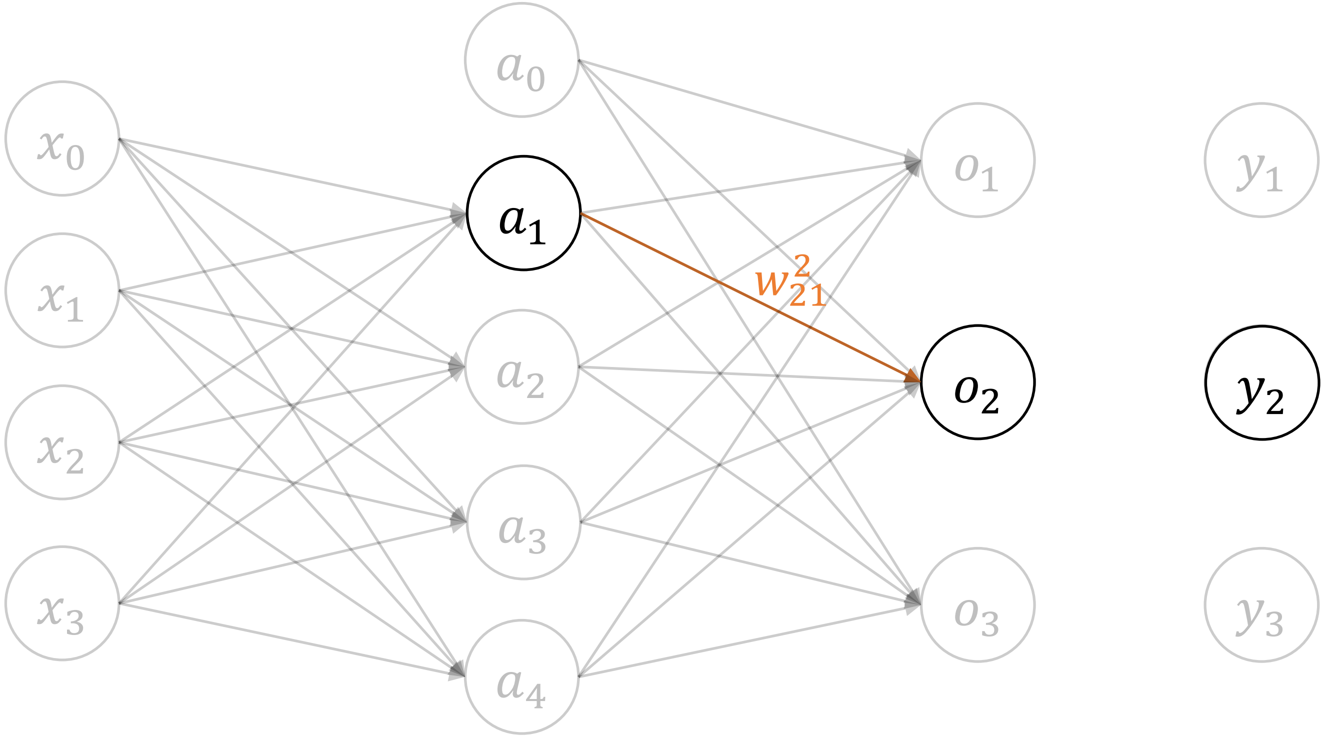
$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_d} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial z_h}{\partial x_1} & \cdots & \cdots & \frac{\partial z_h}{\partial x_d} \end{pmatrix} = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} = W^k$$

14기 정규세션

ToBig's 13기 이지용

Back Propagation

Hidden Layer - Output Layer 가중치 업데이트



$CE = -\sum y_i \log(o_i),\ MSE = \frac{1}{2}\sum(y_i - o_i)^2$

$softmax = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$

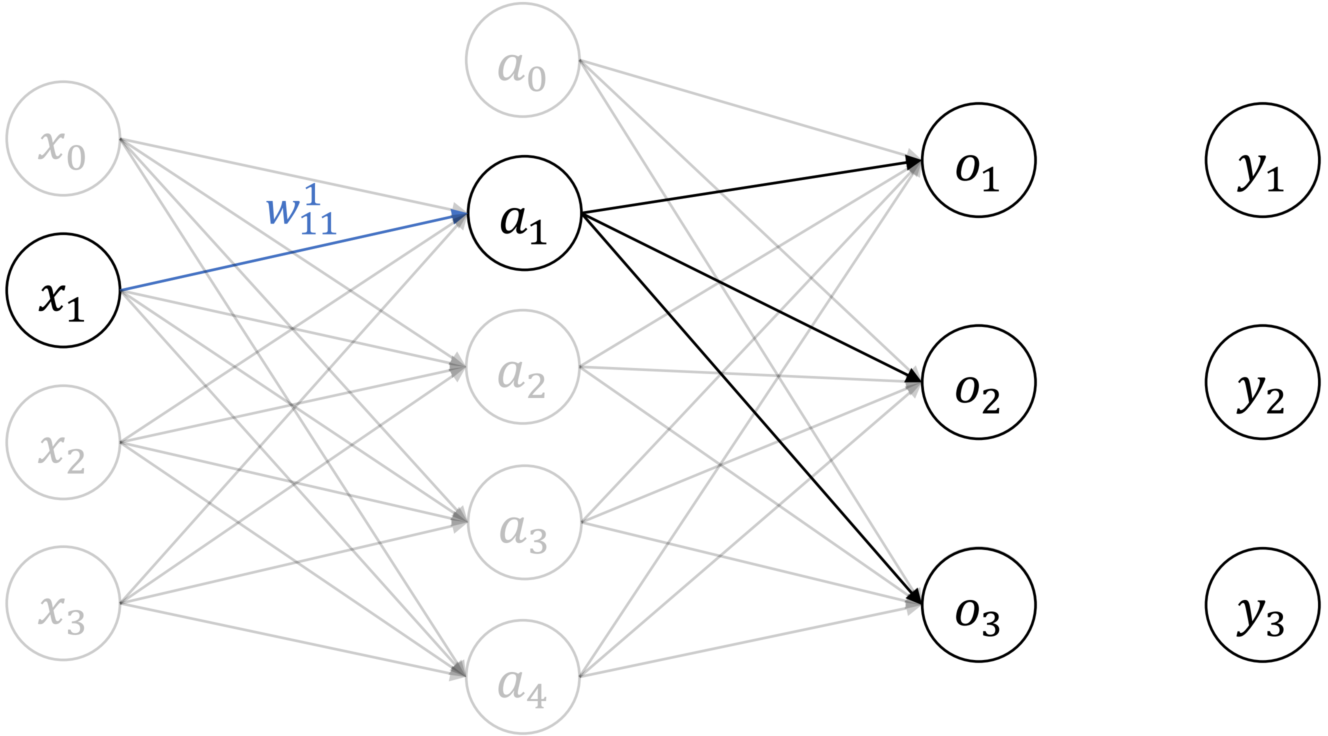
$$z_2^2 = \boldsymbol{w}_2^2 \boldsymbol{a} + b_2^2, \qquad o_2 = g(z_2^2)$$

$$\frac{\partial J}{\partial w_{21}^2} = \frac{\partial z_2^2}{\partial w_{21}^2} \frac{\partial o_2}{\partial z_2^2} \frac{\partial J}{\partial o_2}$$

$$w_{21}^2 \leftarrow w_{21}^2 + \eta(y_2 - o_2)a_1$$

Unit 05 | Back Propagation

Input Layer – Hidden Layer 가중치 업데이트



$sigmoid = \phi(z) = \frac{1}{1+e^{-z}}$

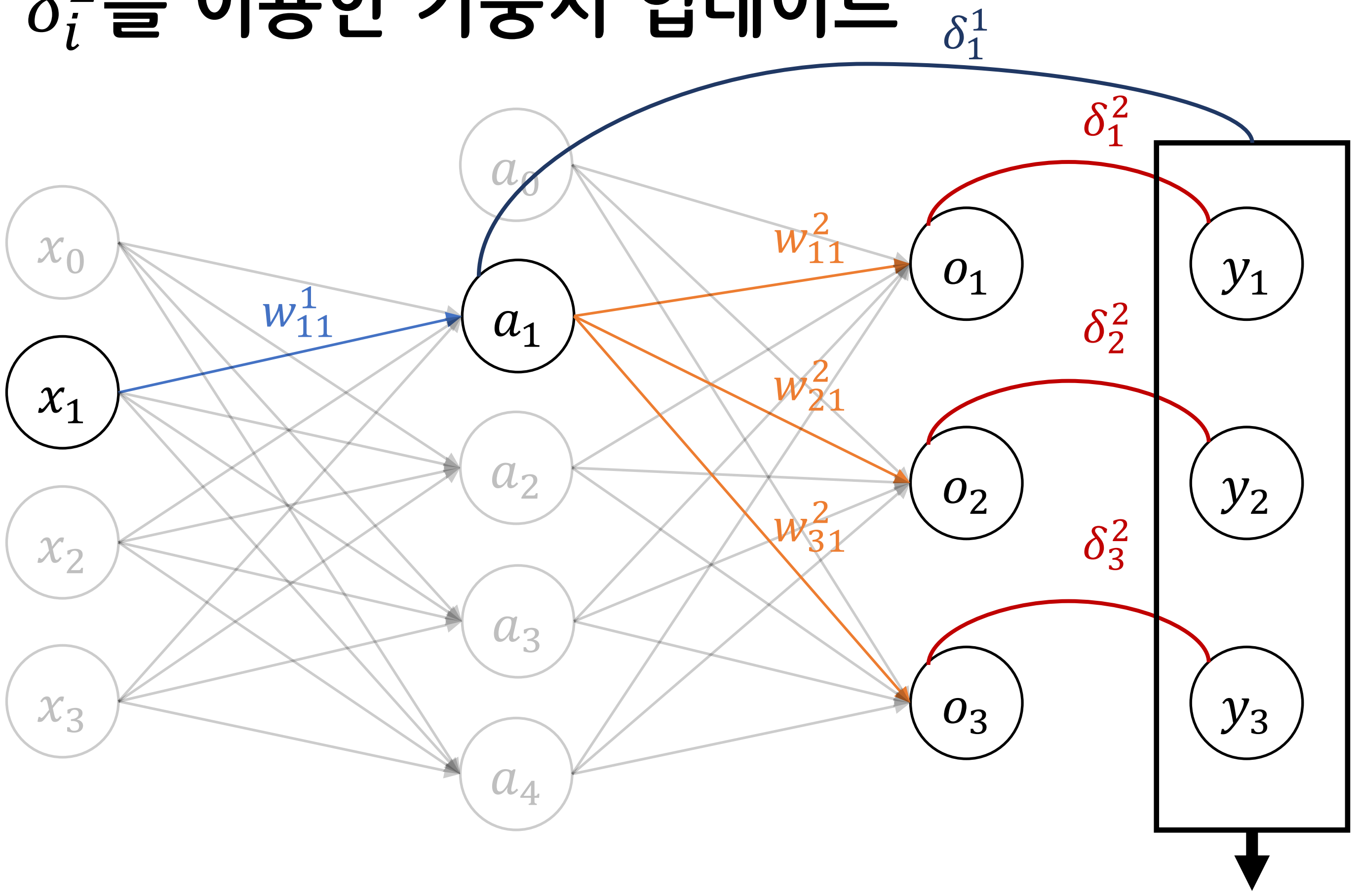
$ReLU = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} = \max(0, x)$

$z_1^1 = \boldsymbol{w}_1^1 \boldsymbol{x} + b_1^1, \qquad a_1 = f(z_1^1)$

$$\begin{aligned}
\frac{\partial J}{\partial w_{11}^1} &= \frac{\partial z_1^1}{\partial w_{11}^1} \frac{\partial a_1}{\partial z_1^1} \frac{\partial J}{\partial a_1} \\
&= \frac{\partial z_1^1}{\partial w_{11}^1} \frac{\partial a_1}{\partial z_1^1} \left(\frac{\partial J_{o_1}}{\partial a_1} + \frac{\partial J_{o_2}}{\partial a_1} + \frac{\partial J_{o_3}}{\partial a_1} \right) \\
&= \frac{\partial z_1^1}{\partial w_{11}^1} \frac{\partial a_1}{\partial z_1^1} \left(\sum \frac{\partial J_{o_k}}{\partial a_1} \right) \\
&= \frac{\partial z_1^1}{\partial w_{11}^1} \frac{\partial a_1}{\partial z_1^1} \left(\sum \frac{\partial z_k^2}{\partial a_1} \frac{\partial o_k}{\partial z_k^2} \frac{\partial J}{\partial o_k} \right)
\end{aligned}$$

$w_{11}^1 \leftarrow w_{11}^1 - \eta x_1 f'(z_1^1) \sum w_{k1}^2 \delta_k^2$

오차항 δ_i^L를 이용한 가중치 업데이트



$$\delta_2^2 = \frac{\partial J}{\partial z_2^2} = z\frac{\partial o_2}{\partial z_2^2}\frac{\partial J}{\partial o_2}$$

$$w_{21}^2 \leftarrow w_{21}^2 - \eta\delta_2^2 a_1$$

$$\delta_1^1 = \frac{\partial J}{\partial z_1^1} = \frac{\partial a_1}{\partial z_1^1}\left(\sum \frac{\partial z_k^2}{\partial a_1}\frac{\partial o_k}{\partial z_k^2}\frac{\partial J}{\partial o_k}\right)$$

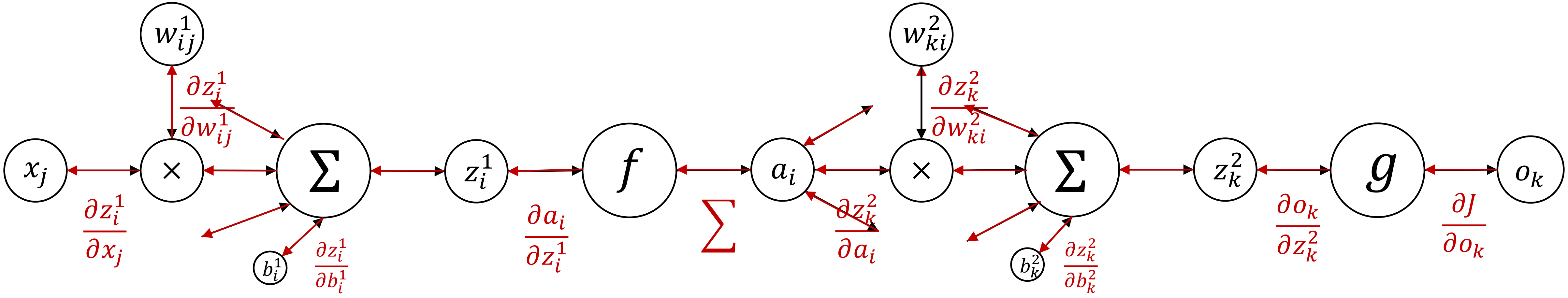
$$= f'(z_1^1)\sum w_{k1}^2\,\delta_k^2$$

$$w_{11}^1 \leftarrow w_{11}^1 - \eta\delta_1^1 x_1$$

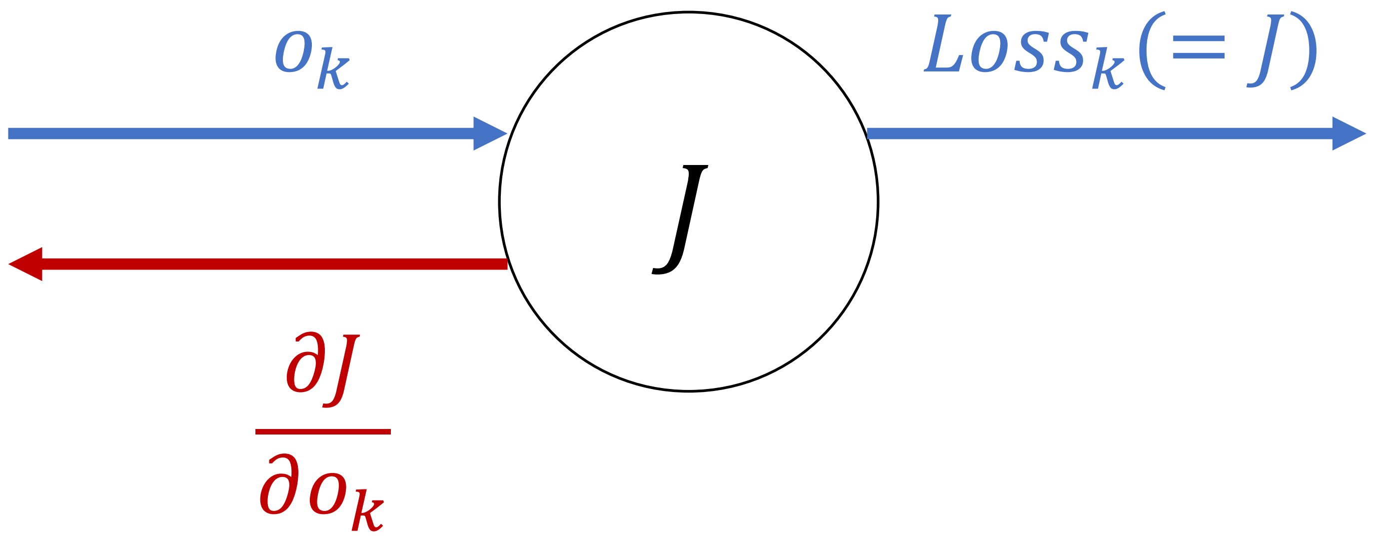
$$\delta_i^L = \frac{\partial J}{\partial z_i^L} = f'\left(z_i^L\right)\sum w_{k1}^{L+1}\,\delta_k^{L+1}$$

$$w_{ij}^L \leftarrow w_{ij}^L - \eta\delta_i^L a_j^{L-1}$$

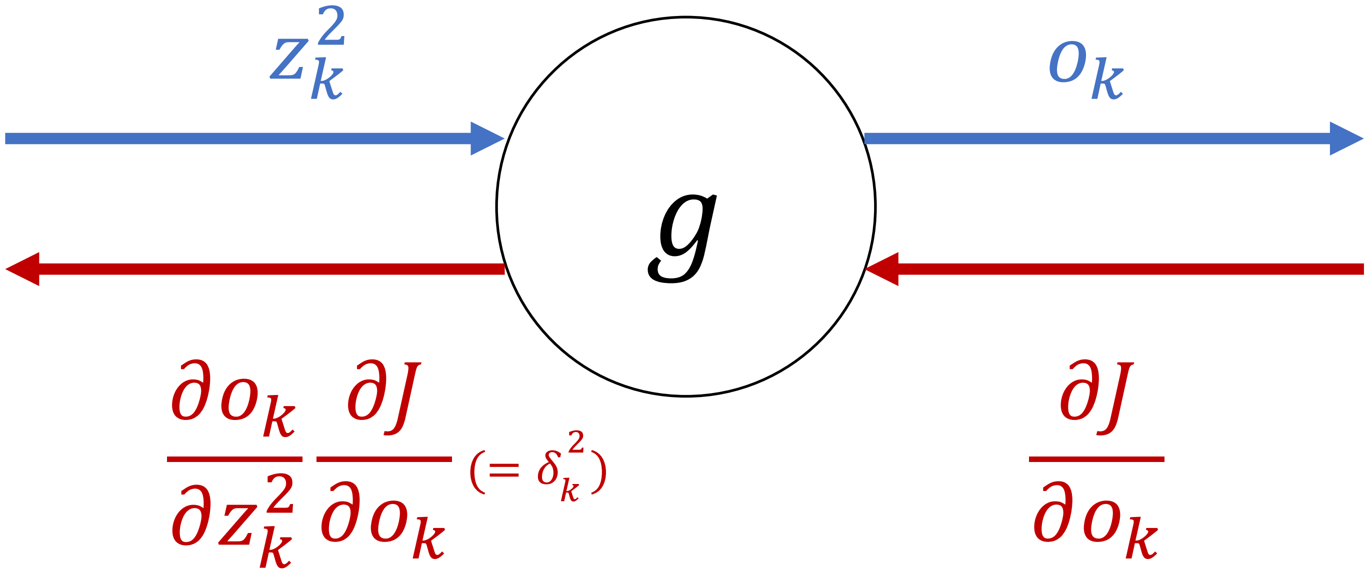
단일 노드들의 역전파


x_j
w_{ij}^1
$\times$
Σ
b_i^1
z_i^1
f
a_i
w_{ki}^2
$\times$
Σ
b_k^2
z_k^2
g
o_k
$\frac{\partial z_i^1}{\partial x_j}$
$\frac{\partial z_i^1}{\partial w_{ij}^1}$
$\frac{\partial z_i^1}{\partial b_i^1}$
$\frac{\partial a_i}{\partial z_i^1}$
$\sum$
$\frac{\partial z_k^2}{\partial a_i}$
$\frac{\partial z_k^2}{\partial w_{ki}^2}$
$\frac{\partial z_k^2}{\partial b_k^2}$
$\frac{\partial o_k}{\partial z_k^2}$
$\frac{\partial J}{\partial o_k}$

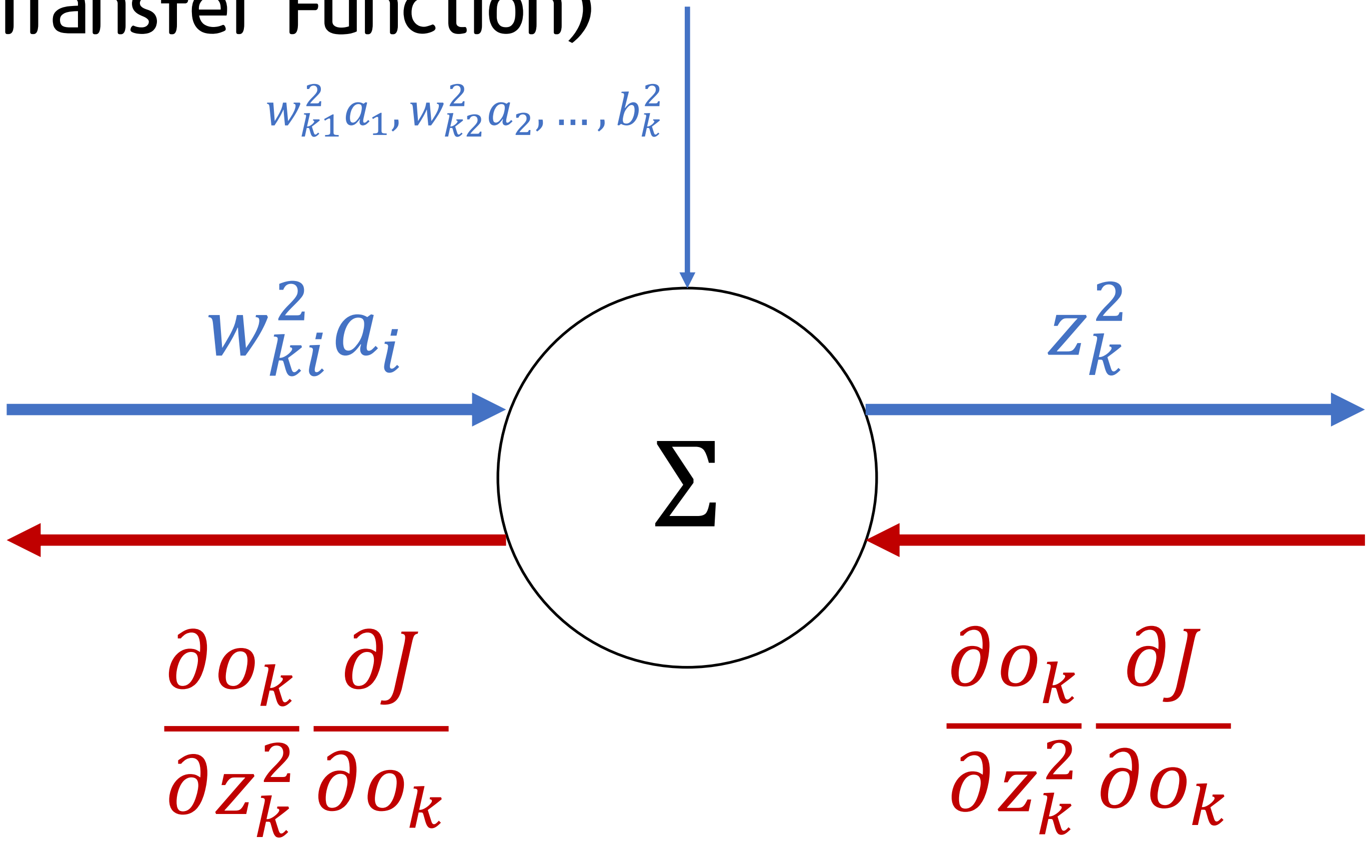
역전파 – Loss Function



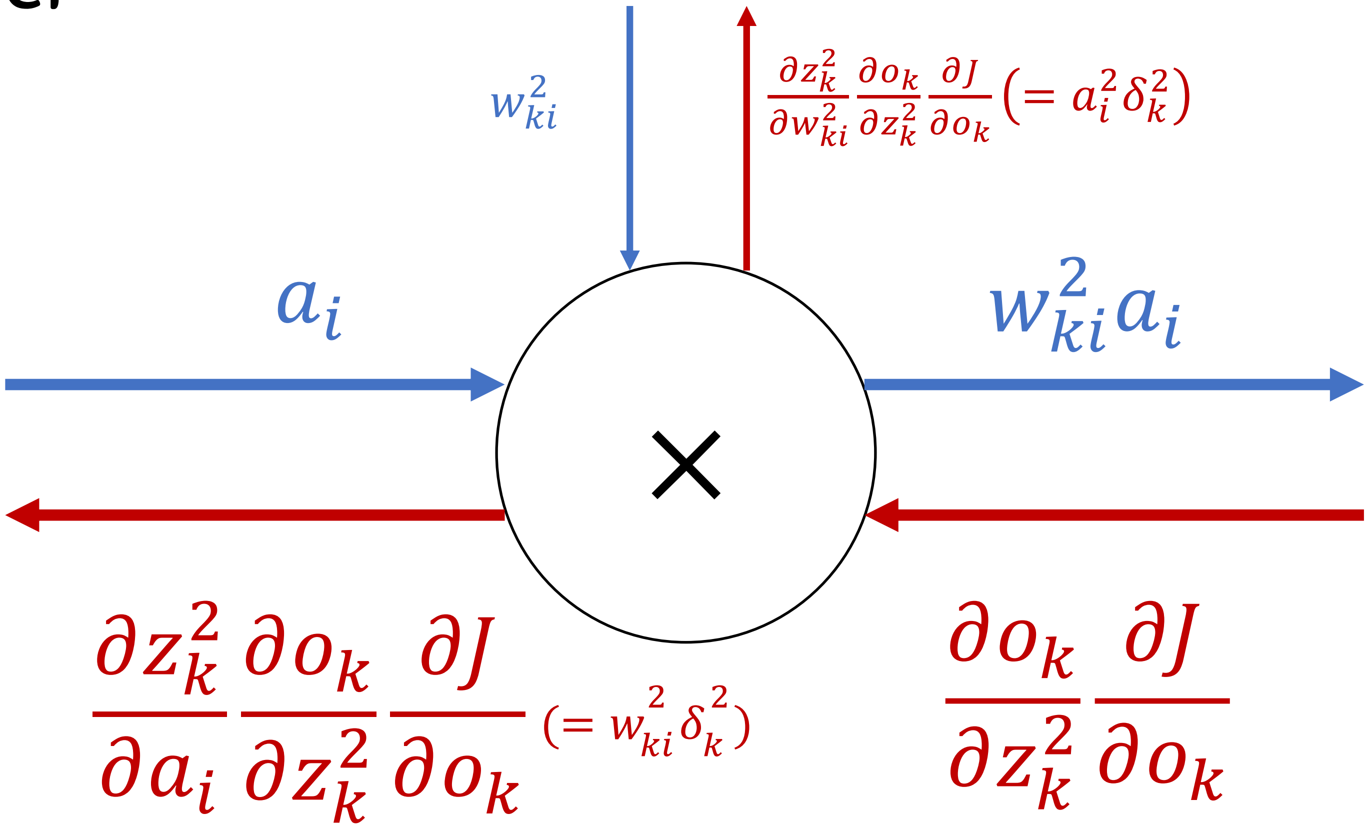
역전파 – Output Activation Function



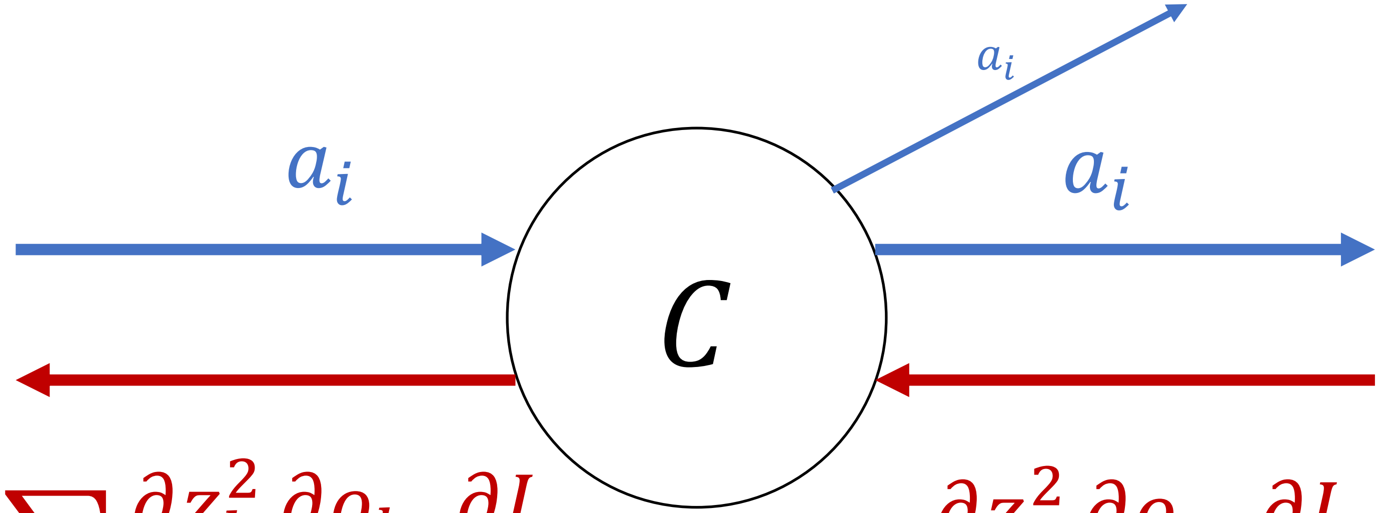
역전파 – Adder(Transfer Function)



역전파 - Multiplier



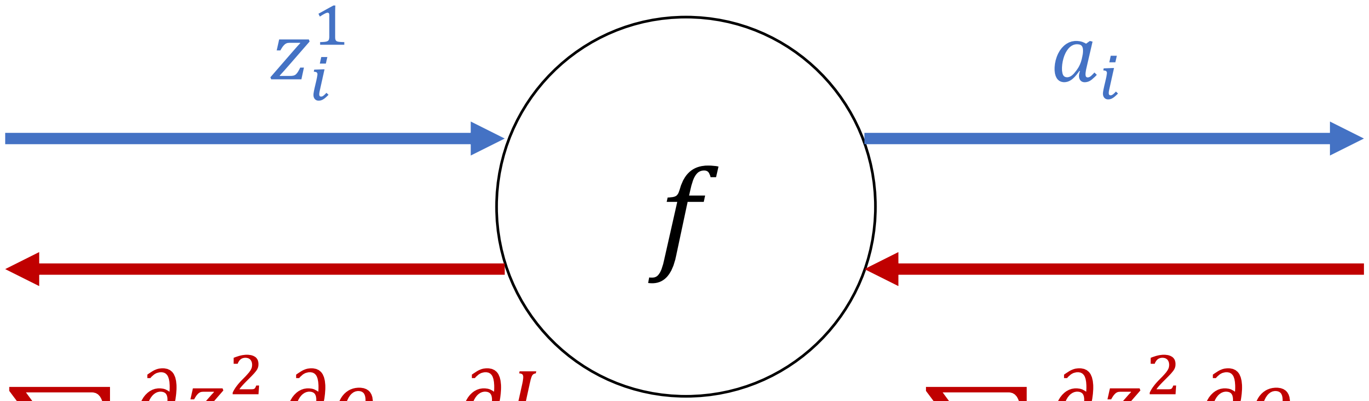
역전파 – Copier



$$\sum \frac{\partial z_k^2}{\partial a_i}\frac{\partial o_k}{\partial z_k^2}\frac{\partial J}{\partial o_k}$$

$$\frac{\partial z_k^2}{\partial a_i}\frac{\partial o_k}{\partial z_k^2}\frac{\partial J}{\partial o_k}$$

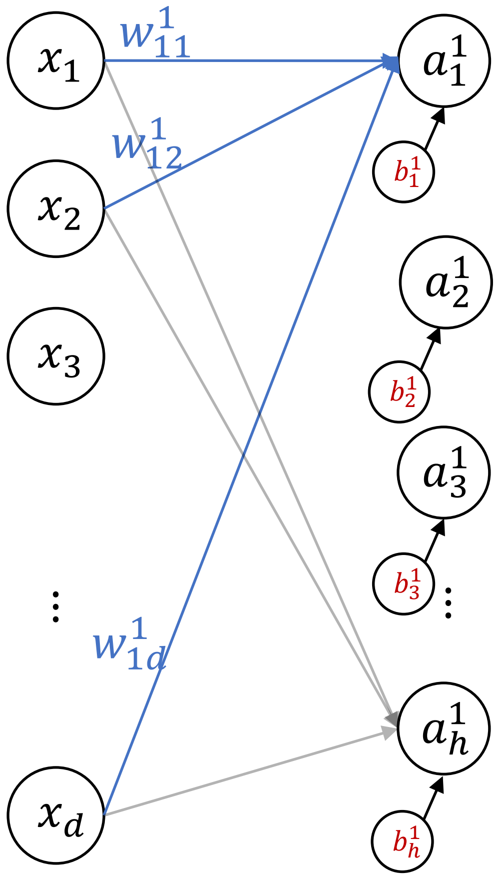
역전파 - Activation Function



$$\frac{\partial a_i}{\partial z_i^1} \sum \frac{\partial z_k^2}{\partial a_i} \frac{\partial o_k}{\partial z_k^2} \frac{\partial J}{\partial o_k} \; (= \delta_i^1) \qquad \sum \frac{\partial z_k^2}{\partial a_i} \frac{\partial o_k}{\partial z_k^2} \frac{\partial J}{\partial o_k}$$

Unit 03 | Multilayer Perceptron

Multilayer Perceptron 행렬 표현



$$\boldsymbol{w}_i^k = \left(w_{i1}^k, w_{i2}^k, \cdots, w_{id}^k\right)^T$$

$$W^k = \left(\boldsymbol{w}_1^k \quad \boldsymbol{w}_2^k \quad \cdots \quad \boldsymbol{w}_h^k\right)^T$$

$$W^k = \begin{pmatrix} w_{11}^k & w_{21}^k & \cdots & w_{i1}^k & \cdots & w_{h1}^k \\ w_{12}^k & w_{22}^k & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ w_{1j}^k & & & w_{ij}^k & & w_{hj}^k \\ \vdots & & & & \ddots & \vdots \\ w_{1d}^k & \cdots & \cdots & w_{id}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$W^k \boldsymbol{x} + \boldsymbol{b}^k = \underbrace{\begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix}}_{h*d} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}}_{d*1} + \underbrace{\begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix}}_{h*1}$$

Vector, Matrix, Derivative

$$W^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} = \begin{pmatrix} w_{11}^k & \cdots & w_{h1}^k \\ \vdots & \ddots & \vdots \\ w_{1d}^k & \cdots & w_{hd}^k \end{pmatrix}^T$$

$$\boldsymbol{z}^k = W^k \boldsymbol{x} + \boldsymbol{b}^k = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} + \begin{pmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_h^k \end{pmatrix} = \begin{pmatrix} z_1^k \\ z_2^k \\ \vdots \\ z_h^k \end{pmatrix}$$

$$z_1^k = w_{11}^k x_1 + w_{12}^k x_2 + \cdots + w_{1d}^k x_d + b_1^k = \boldsymbol{w}_1^{k^T} \boldsymbol{x}$$

$$z_i^k = w_{i1}^k x_1 + w_{i2}^k x_2 + \cdots + w_{id}^k x_d + b_i^k = \boldsymbol{w}_i^{k^T} \boldsymbol{x}$$

$$\frac{\partial z_1^k}{\partial w_{11}^k} = x_1, \quad \frac{\partial z_i^k}{\partial w_{ij}^k} = x_j$$

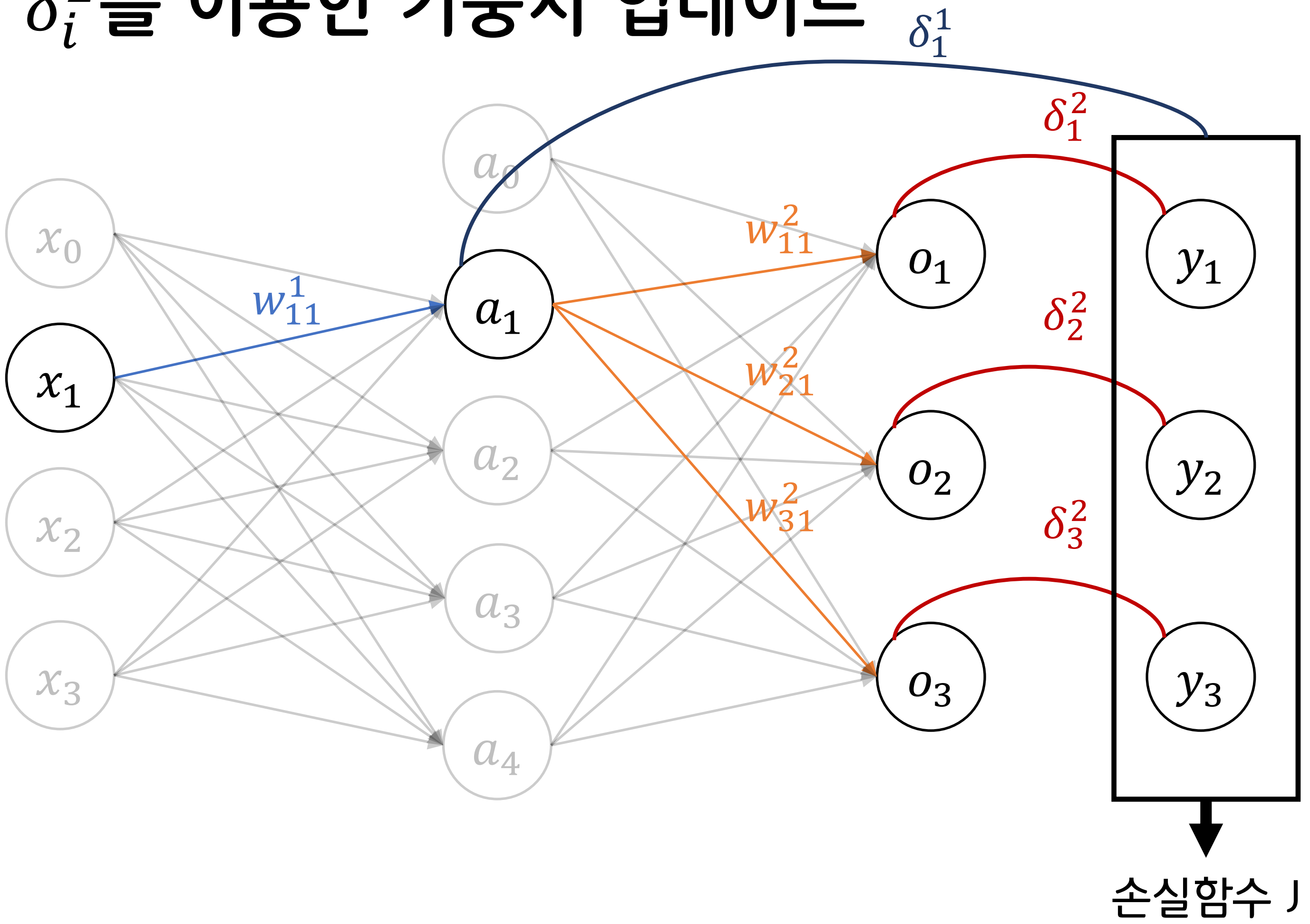
$$\frac{\partial z_i}{\partial \boldsymbol{w}_i^{k^T}} = \boldsymbol{x}^T, \qquad \frac{\partial z_i}{\partial \boldsymbol{w}_i^k} = \begin{pmatrix} \frac{\partial z_i}{\partial w_{i1}^k} \\ \vdots \\ \frac{\partial z_1}{\partial w_{id}^k} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} = \boldsymbol{x}$$

$$\frac{\partial z_i}{\partial x_j} = w_{ij}^k, \qquad \frac{\partial \boldsymbol{z}}{\partial x_j} = \begin{pmatrix} \frac{\partial z_1}{\partial x_j} \\ \vdots \\ \frac{\partial z_h}{\partial x_j} \end{pmatrix} = \begin{pmatrix} w_{11}^k \\ \vdots \\ w_{h1}^k \end{pmatrix}$$

$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_d} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial z_h}{\partial x_1} & \cdots & \cdots & \frac{\partial z_h}{\partial x_d} \end{pmatrix} = \begin{pmatrix} w_{11}^k & \cdots & w_{1d}^k \\ \vdots & \ddots & \vdots \\ w_{h1}^k & \cdots & w_{hd}^k \end{pmatrix} = W^k$$

Unit 05 | Back Propagation

오차항 δ_i^L를 이용한 가중치 업데이트



$$\delta_2^2 = \frac{\partial J}{\partial z_2^2} = \frac{\partial o_2}{\partial z_2^2}\frac{\partial J}{\partial o_2}$$

$$w_{21}^2 \leftarrow w_{21}^2 - \eta \delta_2^2 a_1$$

$$\delta_1^1 = \frac{\partial J}{\partial z_1^1} = \frac{\partial a_1}{\partial z_1^1}\left(\sum \frac{\partial z_k^2}{\partial a_1}\frac{\partial o_k}{\partial z_k^2}\frac{\partial J}{\partial o_k}\right)$$

$$= f'(z_1^1)\sum w_{k1}^2\,\delta_k^2$$

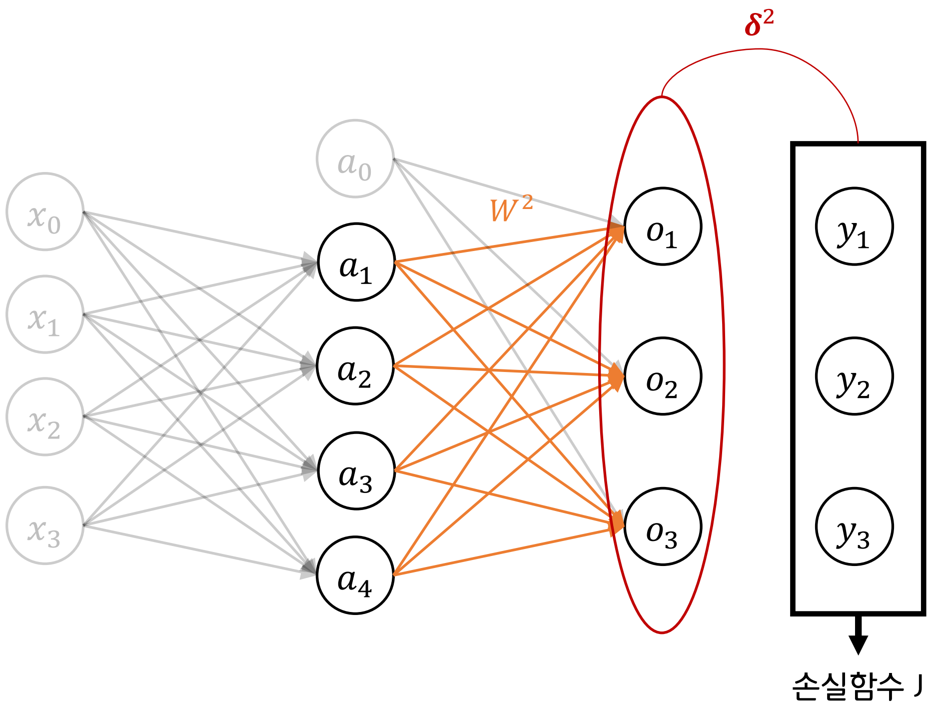
$$w_{11}^1 \leftarrow w_{11}^1 - \eta \delta_1^1 x_1$$

$$\delta_i^L = \frac{\partial J}{\partial z_i^L} = f'\left(z_i^L\right)\sum w_{k1}^{L+1}\,\delta_k^{L+1}$$

$$w_{ij}^L \leftarrow w_{ij}^L - \eta \delta_i^L a_j^{L-1}$$

Unit 05 | Back Propagation

역전파 행렬 표현



$$\boldsymbol{\delta}^2 = \frac{\partial J}{\partial \boldsymbol{z}^2} = \begin{pmatrix} \frac{\partial J}{\partial z_1^2} \\ \frac{\partial J}{\partial z_2^2} \\ \frac{\partial J}{\partial z_3^2} \end{pmatrix} = \begin{pmatrix} \delta_1^2 \\ \delta_2^2 \\ \delta_3^2 \end{pmatrix}$$

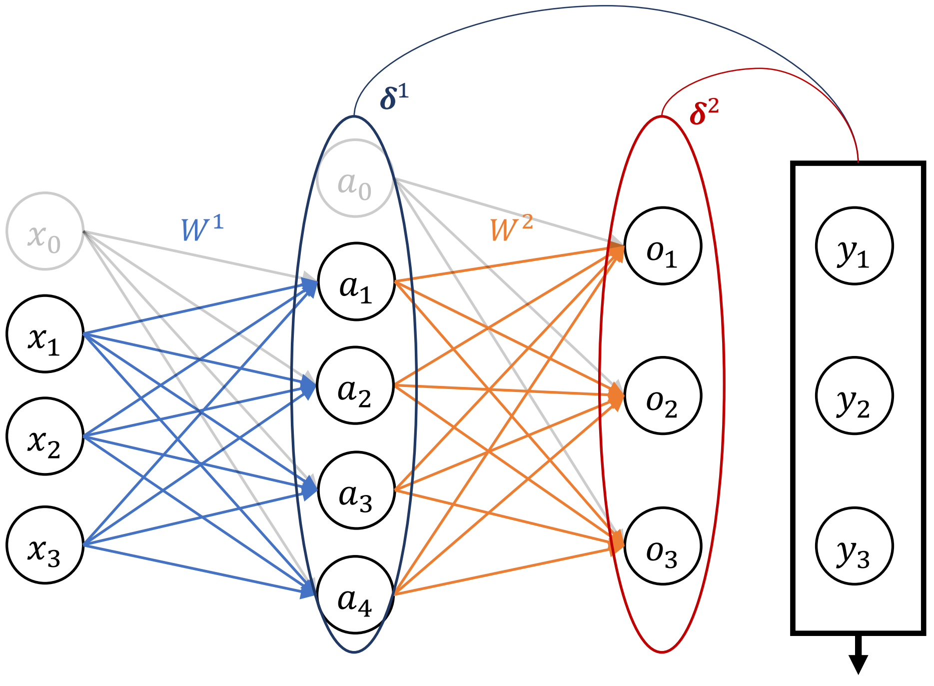
$$\boldsymbol{\delta}^2 \boldsymbol{a}^T = \begin{pmatrix} \delta_1^2 \\ \delta_2^2 \\ \delta_3^2 \end{pmatrix} (a_1 \quad a_2 \quad a_3 \quad a_4)$$

$$= \begin{pmatrix} \delta_1^2 a_1 & \delta_1^2 a_2 & \delta_1^2 a_3 & \delta_1^2 a_4 \\ \delta_2^2 a_1 & \delta_2^2 a_2 & \delta_2^2 a_3 & \delta_2^2 a_4 \\ \delta_3^2 a_1 & \delta_3^2 a_2 & \delta_3^2 a_3 & \delta_3^2 a_4 \end{pmatrix}$$

$$W^2 \leftarrow W^2 - \eta \boldsymbol{\delta}^2 \boldsymbol{a} \quad (= W^2 + \triangle W^2)$$

Unit 05 | Back Propagation

역전파 행렬 표현



$$\delta^1 = \frac{\partial J}{\partial \boldsymbol{z^1}} = \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{z^1}} \frac{\partial \boldsymbol{z^2}}{\partial \boldsymbol{a}} \frac{\partial J}{\partial \boldsymbol{z^2}}$$

$$\frac{\partial \boldsymbol{z^2}}{\partial \boldsymbol{a}} = \begin{pmatrix} \frac{\partial z_1^2}{\partial a_1} & \frac{\partial z_2^2}{\partial a_1} & \frac{\partial z_3^2}{\partial a_1} \\ \frac{\partial z_1^2}{\partial a_2} & \frac{\partial z_2^2}{\partial a_2} & \frac{\partial z_3^2}{\partial a_2} \\ \frac{\partial z_1^2}{\partial a_3} & \frac{\partial z_2^2}{\partial a_3} & \frac{\partial z_3^2}{\partial a_3} \\ \frac{\partial z_1^2}{\partial a_4} & \frac{\partial z_2^2}{\partial a_4} & \frac{\partial z_3^2}{\partial a_4} \end{pmatrix}^T = \begin{pmatrix} w_{11}^2 & w_{21}^2 & w_{31}^2 \\ w_{12}^2 & w_{22}^2 & w_{32}^2 \\ w_{13}^2 & w_{23}^2 & w_{33}^2 \\ w_{14}^2 & w_{24}^2 & w_{34}^2 \end{pmatrix}^T = W^2$$

$$\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{z^1}} = \begin{pmatrix} \frac{\partial a_1}{\partial z_1^1} \\ \frac{\partial a_2}{\partial z_2^1} \\ \frac{\partial a_3}{\partial z_3^1} \\ \frac{\partial a_4}{\partial z_4^1} \end{pmatrix} = \begin{pmatrix} f'(a_1) \\ f'(a_2) \\ f'(a_3) \\ f'(a_4) \end{pmatrix} = f'(\boldsymbol{a})$$

$$\delta^1 = f'(\boldsymbol{a}) \circ ({W^2}^T \boldsymbol{\delta}^2) = \begin{pmatrix} f'(a_1) \\ f'(a_2) \\ f'(a_3) \\ f'(a_4) \end{pmatrix} \circ \begin{pmatrix} w_{11}^2 & w_{21}^2 & w_{31}^2 \\ w_{12}^2 & w_{22}^2 & w_{32}^2 \\ w_{13}^2 & w_{23}^2 & w_{33}^2 \\ w_{14}^2 & w_{24}^2 & w_{34}^2 \end{pmatrix} \begin{pmatrix} \delta_1^2 \\ \delta_2^2 \\ \delta_3^2 \end{pmatrix} = \begin{pmatrix} \delta_1^1 \\ \delta_2^1 \\ \delta_3^1 \\ \delta_4^1 \end{pmatrix}$$

$$W^1 \leftarrow W^1 - \eta \boldsymbol{\delta}^1 \boldsymbol{x}$$

Assignment

Assignment 1 – 신경망 문제 풀이

문제를 풀어주세요. 자신만의 풀이과정을 상세하게 써 주셔야 합니다.
노트 필기, 인쇄본에 자필 작성 후 스캔 및 사진, iPad 필기본 등으로 제출해 주셔야 인정합니다.
복붙은 리젝할게요.(직접 한 수식 타이핑까지는 인정)
우수과제 기준 : 풀이 과정 > 점수 > 제출 시간

Assignment 2 – 신경망 코드 구현

week6_NeuralNetwork_Basic_assignment2.ipynb 파일의 빈칸을 채워주세요.
그리고 예측을 가장 잘 하는 신경망 모델을 짜주세요.
함수의 역할과 동작과정을 주석에 자세히 달아주세요.
우수과제 기준 : 주석 > 성능

각 과제의 우수과제자와 점수가 제일 높은 분께 상품을 드릴게요~

Reference

참고자료

- Tobig's 12기 배유나, 김태한 강의자료
- The hundred-page machine learning book
- 처음 배우는 딥러닝 수학
- 파이썬 머신러닝
- https://www.edwith.org/deepnlp/joinLectures/17363
- https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=1
- https://ko.wikipedia.org/wiki/%EC%9D%B8%EA%B3%B5_%EC%8B%A0%EA%B2%BD%EB%A7%9D
- https://wikidocs.net/37406
- https://ratsgo.github.io/
- https://reniew.github.io/12/
- https://jinseob2kim.github.io/deep_learning.html
- https://simpling.tistory.com/entry/Mean-Squared-Error-VS-Cross-Entropy-Error
- https://sacko.tistory.com/10?category=632408
- https://eehoeskrap.tistory.com/137
- https://www.slipp.net/wiki/
- http://www.aistudy.com/neural/model_kim.htm
- https://brunch.co.kr/@hvnpoet/
- https://nbviewer.jupyter.org/github/metamath1/ml-simple-works/blob/master/fitting/matrix-derivative.ipynb

논문 목록

McCulloch, W. S., & Pitts, W. (1943).
A logical calculus of the ideas immanent in nervous activity
http://aiplaybook.a16z.com/reference-material/mcculloch-pitts-1943-neural-networks.pdf

Hebb, D. O. (1949).
The organization of behavior; a neuropsychological theory
https://pure.mpg.de/rest/items/item_2346268_3/component/file_2346267/content

Rosenblatt, F. (1958).
The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain
http://www2.fiit.stuba.sk/~cernans/nn/nn_texts/neuronove_siete_priesvitky_02_Q.pdf

Widrow, B., & Hoff, M. (1960).
Adaptive Switching Circuits
https://apps.dtic.mil/dtic/tr/fulltext/u2/241531.pdf

Minsky, M., & Papert, S. (1969).
Perceptrons: an introduction to computational geometry
https://books.google.co.kr/books?hl=ko&lr=&id=PLQ5DwAAQBAJ&oi=fnd&pg=PR5&dq=Perceptrons:+an+introduction+to+computational+geometry&ots=zyLFwLvl31&sig=Kp_aP5TtLbAHZogzNLnosh9eyvI#v=onepage&q=Perceptrons%3A%20an%20introduction%20to%20computational%20geometry&f=false

논문 목록

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986).
Learning representations by back-propagating errors
http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf

LeCun, Y. et al. (1989).
Backpropagation Applied to Handwritten Zip Code Recognition
https://www.ics.uci.edu/~welling/teaching/273ASpring09/lecun-89e.pdf

Hochreiter, S. (1991).
Untersuchungen zu dynamischen neuronalen Netzen
https://www.bioinf.jku.at/publications/older/3804.pdf

Bengio, Y., Simard, P., & Frasconi, P. (1994).
Learning long-term dependencies with gradient descent is difficult
http://www.comp.hkbu.edu.hk/~markus/teaching/comp7650/tnn-94-gradient.pdf

논문 목록

Hinton, G. E., & Salakhutdinov, R. R. (2006).
Reducing the dimensionality of data with neural networks
https://dbirman.github.io/learn/hierarchy/pdfs/Hinton2006.pdf

Nair, V., & Hinton, G. E. (2010).
Rectified Linear Units Improve Restricted Boltzmann Machines
https://icml.cc/Conferences/2010/papers/432.pdf

Glorot, X., Bordes, A. & Bengio, Y. (2011).
Deep Sparse Rectifier Neural Networks
http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf

Hinton, G. E., Srivastava, N, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. (2012).
Improving neural networks by preventing co-adaptation of feature detectors
https://arxiv.org/pdf/1207.0580.pdf)

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012).
ImageNet Classification with Deep Convolutional Neural Networks
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Lecun, Y., Bengio, Y., & Hinton, G. E. (2015).
Deep Learning
https://s3.us-east-2.amazonaws.com/hkg-website-assets/static/pages/files/DeepLearning.pdf

Q & A

들어주셔서 감사합니다.