Multilayer Perceptron

POSTECH CSE

Saemi Moon

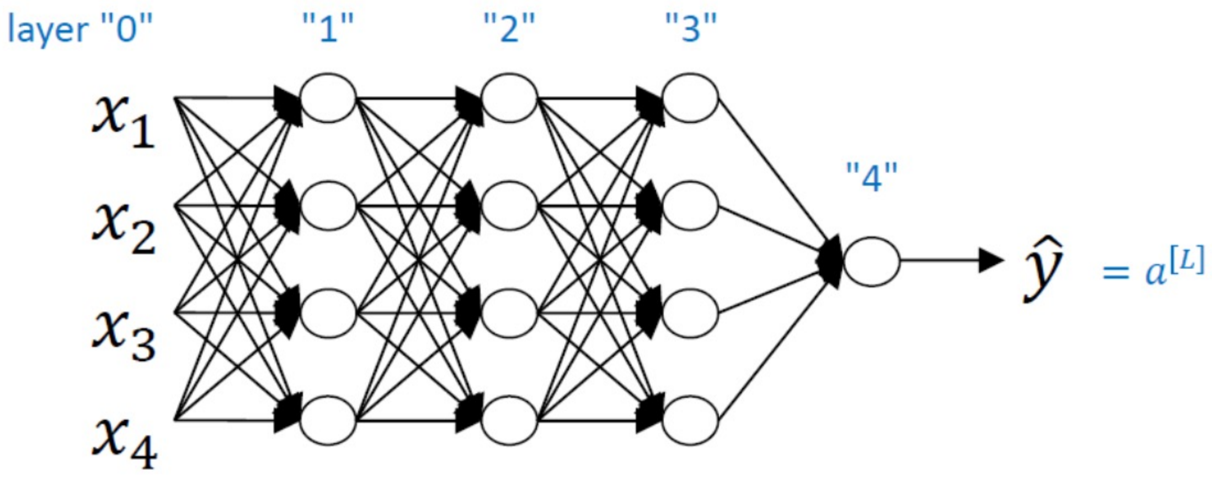
Contents

- Review the key points
  - Multilayer Perceptron (MLP)
  - Activation function
  - Backpropagation

Contents

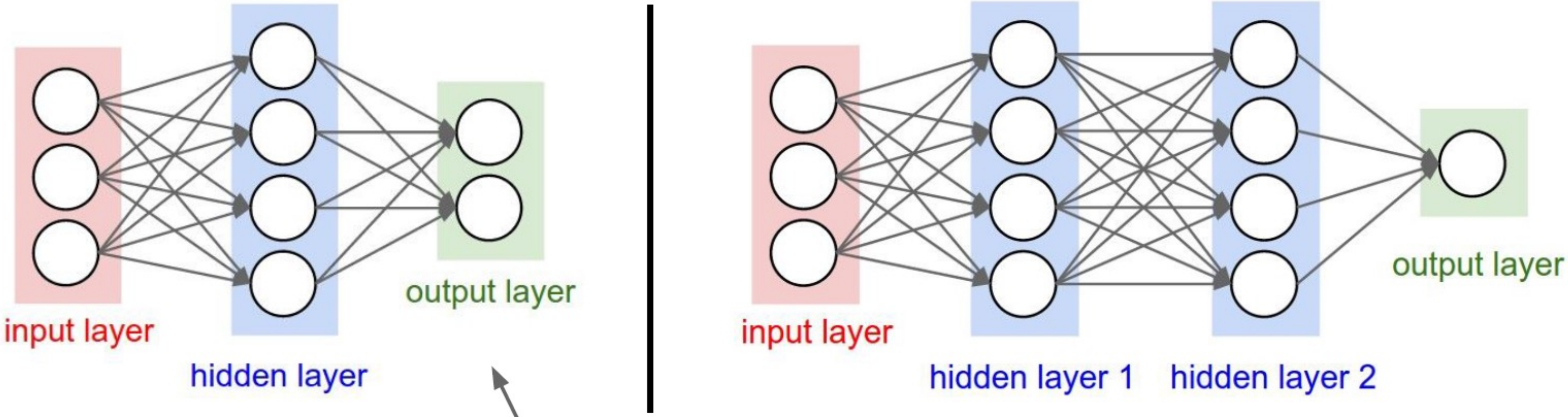
- Review the key points
  - Multilayer Perceptron (MLP)
  - Activation function
  - Backpropagation

MLP notation



- Input이 있는 layer를 0번째 layer라고 하기도 하지만, 주로 input layer라고 함
- 마지막 layer는 output layer라고 부름
- Input layer, output layer를 제외한 나머지 layer를 hidden layer라고 함

MLP notation



“2-layer Neural Net”, or
“1-hidden-layer Neural Net”

“3-layer Neural Net”, or
“2-hidden-layer Neural Net”

“Fully-connected” layers

MLP notation

- Neural Network에서 input $x \in \mathbb{R}^D$ 가 주어졌을 때, 뉴런은 다음과 같이 정의할 수 있다.

$$f = \sigma(w^\top x + b),\ w \in \mathbb{R}^D \quad b \in \mathbb{R}$$

$\sigma(.)$: non linear activation function

MLP notation

- Single layer NN

$$f = \sigma(W_1 x + b_1)$$

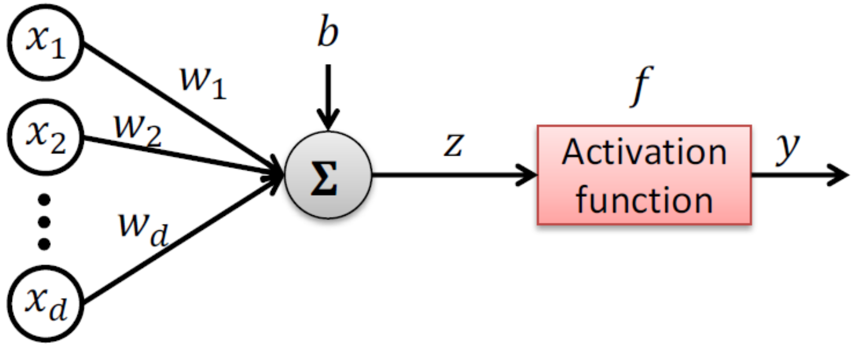
- 2-layer NN

$$f = W_2 \sigma(W_1 x + b_1)$$

- 3-layer NN

$$f = W_3 \sigma(W_2 \sigma(W_1 x + b_1) + b_2)$$

Single-Layer NN



- Framework
  - Input: $x = (x_1, x_2, \ldots, x_d)^\top$
  - Output: y
  - Model: weight vector $w = (w_1, w_2, \ldots, w_d)^\top$ and bias b

$$y = f(z) = f(w^\top x + b)$$

Limitation of single layer NN

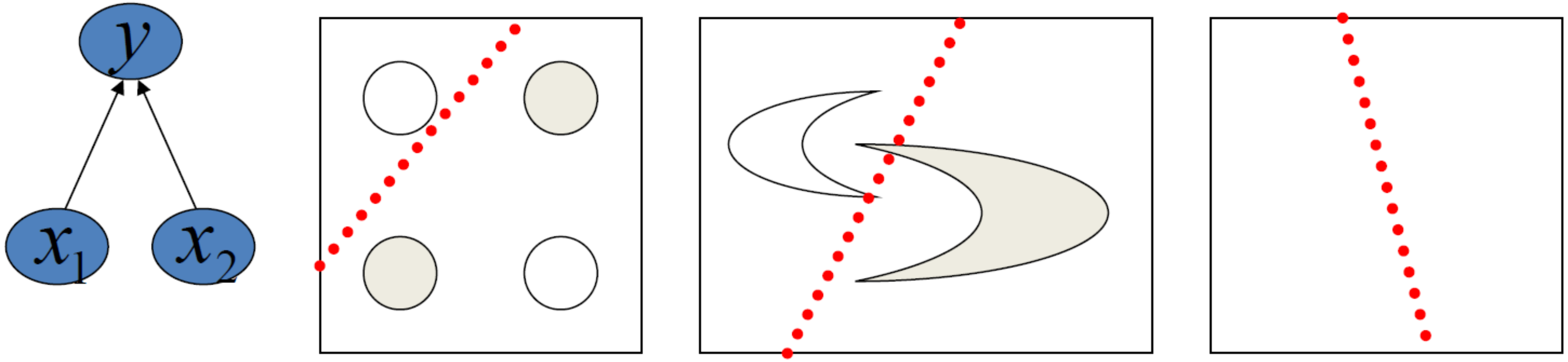


Figure: Let x_1 and x_2 be the inputs and y is the output from zero-hidden layer net. The network can represent any hyperplane separating input dimensions into two halves.

Why non-linear activation function?

- 데이터가 복잡해지고, feature들의 차원이 증가하면서 데이터의 분포가 선형적으로 표현되는 경우가 매우 적음
- 선형적이지 않은 데이터는 선형적인 boundary로는 표현이 불가능 하므로 비선형 boundary가 필요함

Role of Activation function

- Activation function의 역할?
  - 만약, activation function을 사용하지 않는다면?

$$\mathbf{f} = \mathbf{W}_2(\mathbf{W}_1\mathbf{x} + \mathbf{b})$$
$$\mathbf{f} = \mathbf{W}_2\mathbf{W}_1\mathbf{x} + \mathbf{W}_2\mathbf{b}$$
$$\mathbf{f} = \mathbf{W}'\mathbf{x} + \mathbf{b}'$$

  - 서로 다른 weight를 아무리 많이 곱하더라도 하나의 weigh를 곱한 것과 똑같은 표현력을 갖게 됨

Role of Activation function

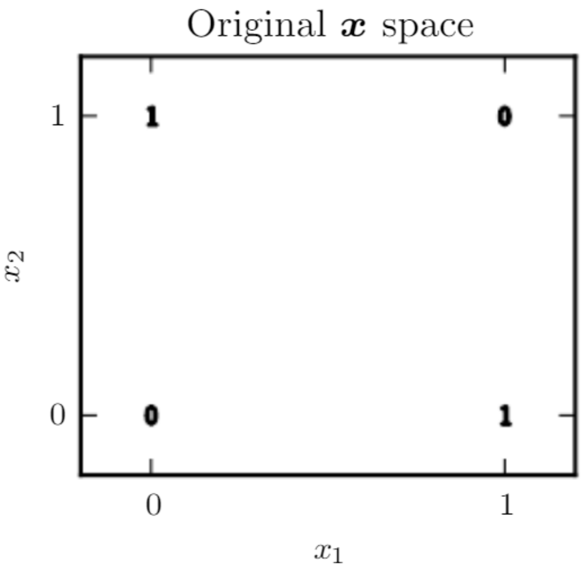
- Activation function의 역할?
  - 만약, activation function으로 linear function을 사용하게 된다면?

$$f(x) = c(c(c(x))) = c^3x = ax$$

  - 여러 개의 layer를 쌓더라도 하나의 layer와 같은 표현력을 갖게 됨

Representation Power

- Activation function의 역할?
  - Ex) 다음과 같은 **X**와 **y**가 데이터셋으로 주어졌다고 가정



$$\mathbf{X} = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vec{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016).
http://www.deeplearningbook.org

Representation Power

- Activation function의 역할?
  - Ex) 만약 activation function(ex. ReLU)을 활용해 x와 y 사이의 관계를 아래와 같이 모델링한다면?

$$y = w^{T} \max\{0, XW + c\} + b$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org

Representation Power

- Activation function의 역할?
  - Ex) 만약 activation function을 활용해 x와 y 사이의 관계를 아래와 같이 모델링 한다면?

$$y = w^T \max\{0, XW + c\} + b$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

넣고 계산

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016).
http://www.deeplearningbook.org

Representation Power

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vec{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

- Activation function의 역할?
  - Ex) 만약 activation function을 활용해 x와 y 사이의 관계를 아래와 같이 모델링 한다면?

$$y = w^T \max\{0, \boxed{XW} + c\} + b$$

$$XW = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org

Representation Power

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vec{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \; C = \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \; w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

- Activation function의 역할?
  - Ex) 만약 activation function을 활용해 x와 y 사이의 관계를 아래와 같이 모델링 한다면?

$$y = w^T \max\{0, \boxed{XW + C}\} + b$$

$$XW = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} \qquad XW + C = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016).
http://www.deeplearningbook.org

Representation Power

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vec{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

- Activation function의 역할?
  - Ex) 만약 activation function을 활용해 x와 y 사이의 관계를 아래와 같이 모델링 한다면?

$$y = w^T \boxed{\max\{0, XW + C\}} + b$$

$$XW + C = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \qquad \max\{0, XW + C\} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016).
http://www.deeplearningbook.org

Representation Power

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vec{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

- Activation function의 역할?
  - Ex) 만약 activation function을 활용해 x와 y 사이의 관계를 아래와 같이 모델링 한다면?

$$y = \boxed{w^T \max\{0, XW + c\}} + b$$

$$\max\{0, XW+c\} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \qquad w^T \max\{0, XW+c\} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org

Representation Power

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vec{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

- Activation function의 역할?
  - Ex) 만약 activation function을 활용해 x와 y 사이의 관계를 아래와 같이 모델링 한다면?

$$y = w^T \max\{0, XW + C\} + b$$

$$w^T \max\{0, XW + C\} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

<--비교-->

$$y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org

Representation Power

- One-hidden layer ($\mathbf{f} = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b})$): Open or close boundary of convex region

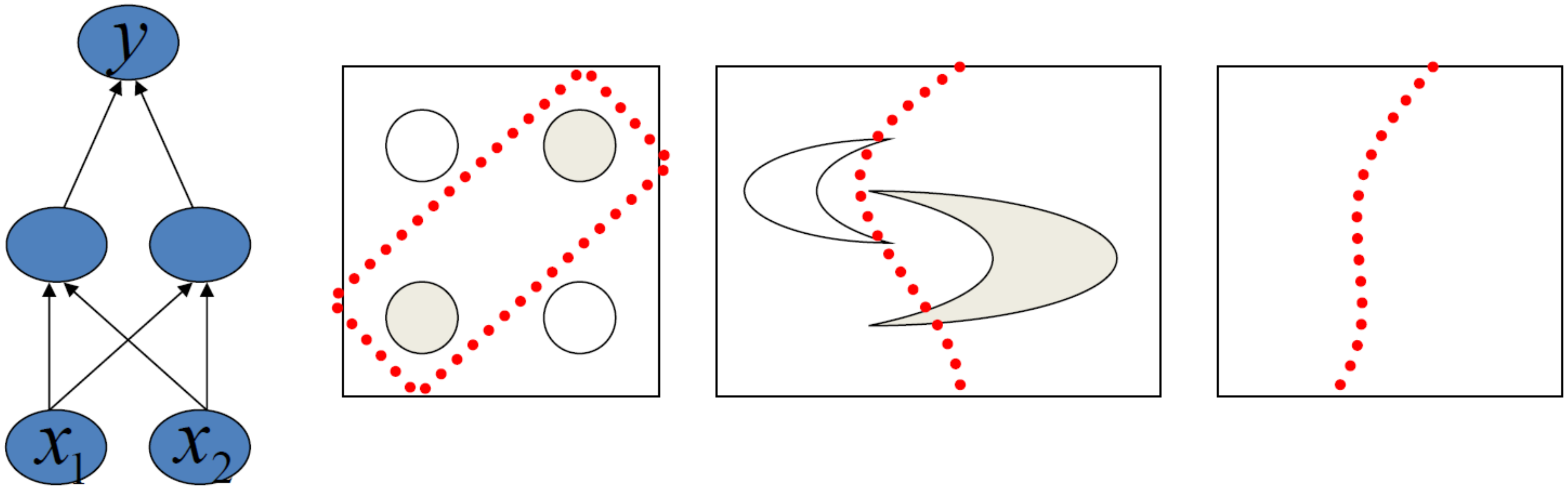
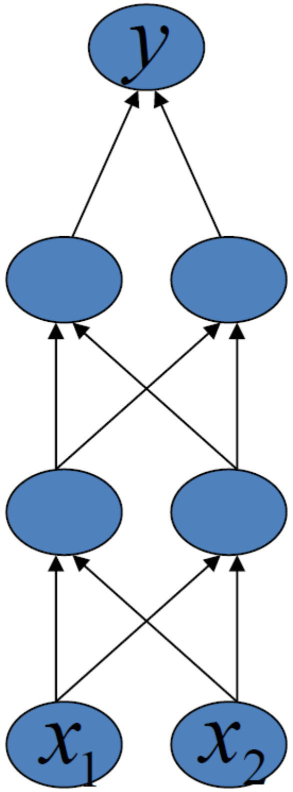


Figure: The one-hidden layer can represent any bounded convex region.

Representation Power



- 2 hidden layers
  - Combinations of convex regions

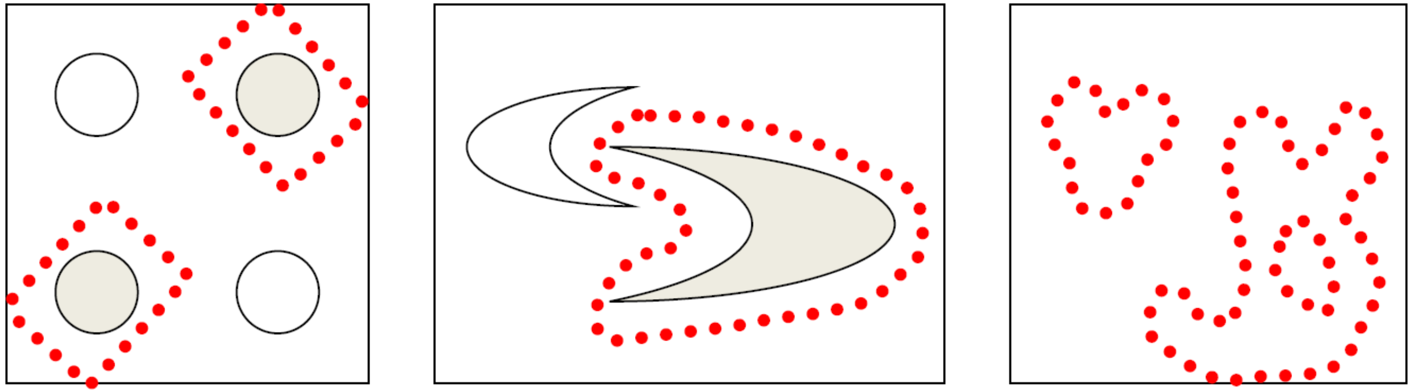
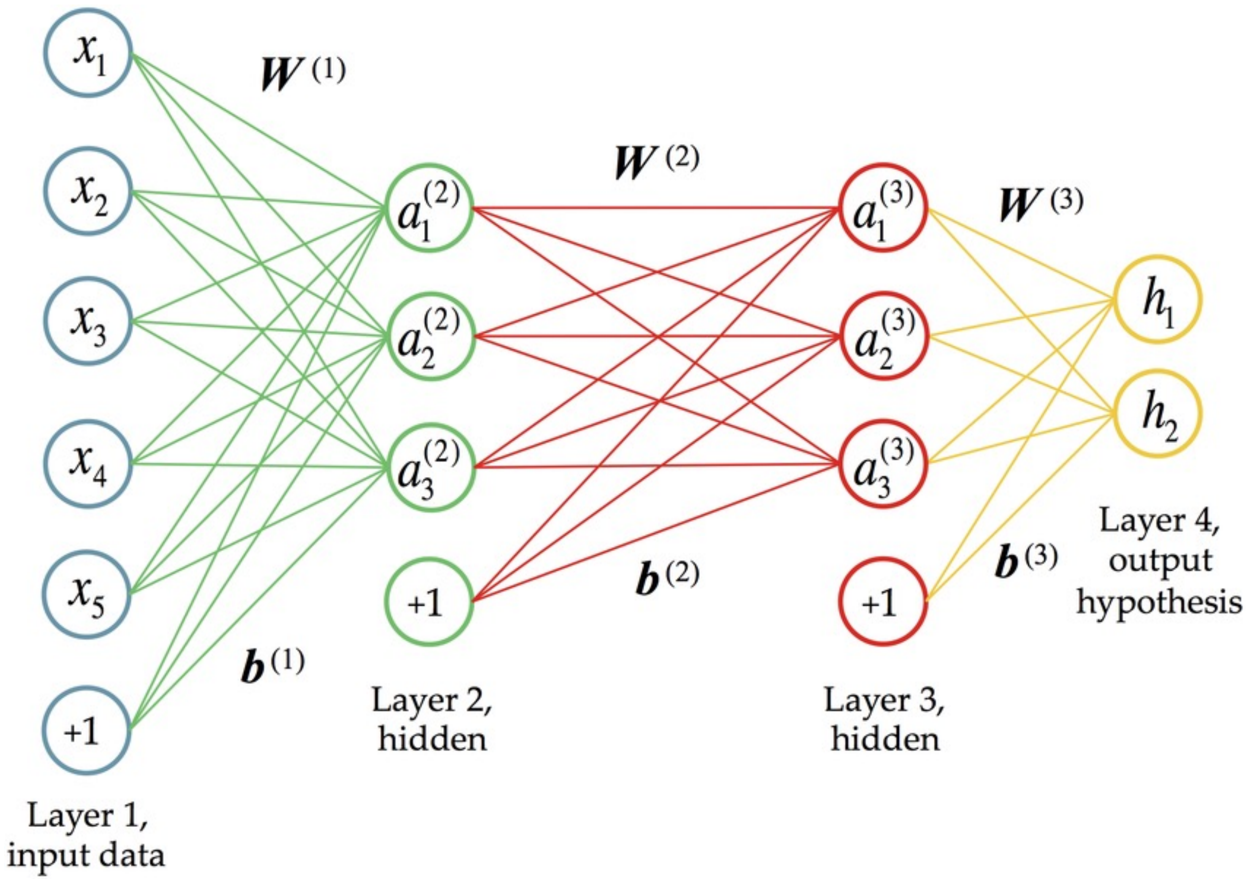


Figure: The two-hidden layers can represent any combination of convex region.

Multilayer Perceptron

- Bias term



Contents

- Review the key points
  - Multilayer Perceptron (MLP)
  - Activation function
  - Backpropagation

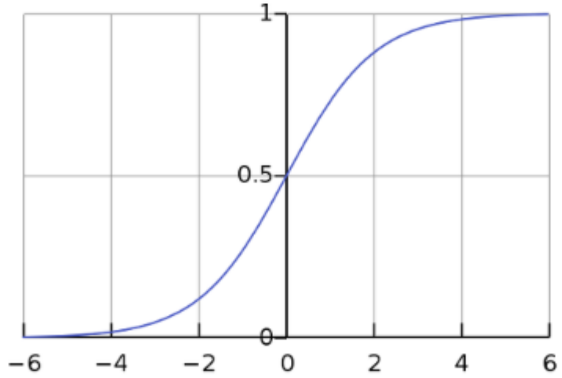
Activation function

- 대표적인 activation function
  - Sigmoid function
  - Tanh function
  - ReLU function
  - Leaky ReLU function

Sigmoid & Tanh function

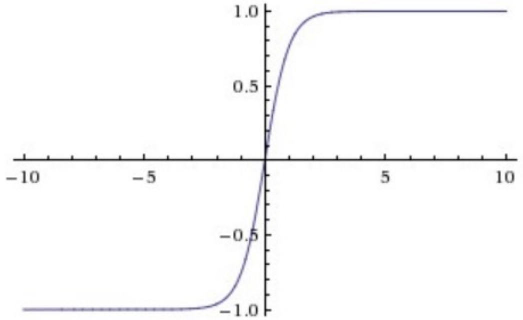
- Sigmoid

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



- Tanh Function

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
$$tanh'(x) = 1 - tanh^2(x)$$

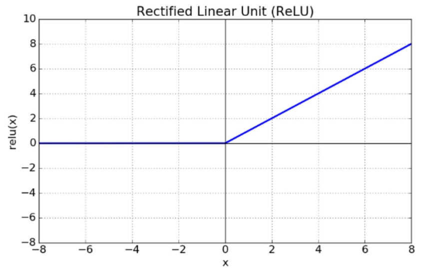


- Input이 매우 작거나 클 때, Gradient가 거의 0이 되는 문제가 생긴다.
- 학습이 안되는 문제가 생김

ReLU & leaky ReLU function

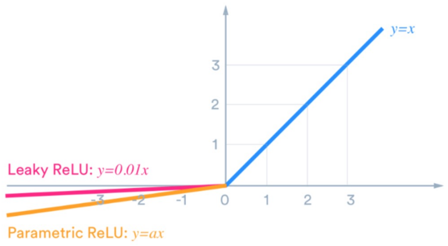
- ReLU

$$f(x) = max(0, x)$$



- Leaky ReLU

$$f(x) = max(\alpha x, x)$$



- Gradient가 없어지는 문제를 해결하고자 함
- ReLU는 음수일 때 Gradient가 없어지는 문제가 있지만, 잘 작동함
- Leaky ReLU는 음수일 때 Gradient가 없어지는 문제를 해결하여 더 성능이 좋지만 실제로는 ReLU를 많이 사용함

Contents

- Review the key points
  - Multilayer Perceptron (MLP)
  - Activation function
  - Backpropagation

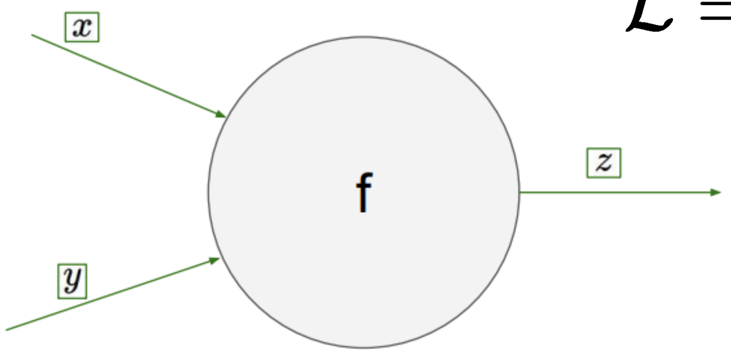
POSTECH

Backpropagation

- 각각의 weight와 bias에 대해서 gradient를 직접 계산해서 구해야 한다.
  - $\dfrac{\partial \mathcal{L}}{\partial W_\ell}, \dfrac{\partial \mathcal{L}}{\partial b_\ell}$
  - $\dfrac{\partial \mathcal{L}}{\partial W_\ell} =$ 와 같은 closed form으로 gradient를 구하는 것은 매우 복잡하고, cost가 큼
  - Loss가 달라지면 다시 계산을 해야함
  - Complex model은 이러한 방식으로 학습이 거의 불가능 함

→ solution: back propagation

Backpropagation

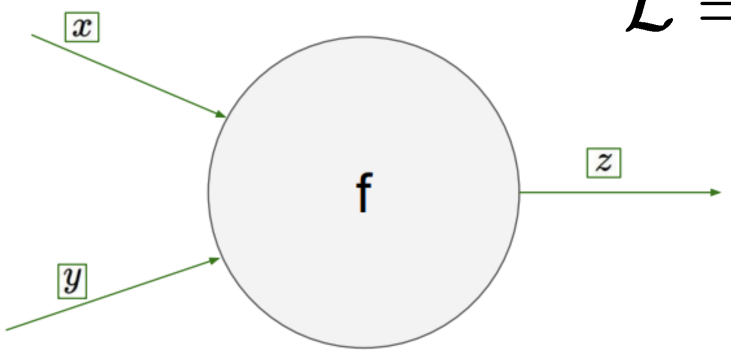


$$\mathcal{L} = (t - z)^2,\ t : label$$

- 다음과 같이 Neural Net이 있고, loss function은 L2-norm을 사용할 때,
  - $\frac{\partial \mathcal{L}}{\partial z}$는 쉽게 구할 수 있음
  - Backpropagation에서는 $\frac{\partial \mathcal{L}}{\partial z}$를 이용하여 $\frac{\partial \mathcal{L}}{\partial x}, \frac{\partial \mathcal{L}}{\partial y}$를 구함

Backpropagation

$$\mathcal{L} = (t - z)^2,\ t : label$$



- By chain rule, $\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial x}$
  - Ex) $z = f(x, y) = x^2 - xy$, $x = 3,\ y = 1,\ t = 5$
    - $z = 6,\ \mathcal{L} = 1,\ \frac{\partial \mathcal{L}}{\partial z} = -2 * (5 - 6) = 2$
    - $\frac{\partial z}{\partial x} = 2 * 3 - 1 = 5$
    - $\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial x} = 2 * 5 = 10$

Connect with Neural Net

- $\ell = 1, \ldots, N$ layer를 갖는 neural net
  - L번째 hidden layer를 다음과 같이 표현

$$h_L = W_L \sigma(h_{L-1}) + b_L$$

- $\frac{\partial \mathcal{L}}{\partial W_L} = \frac{\partial \mathcal{L}}{\partial h_L} \frac{\partial h_L}{\partial W_L}$ $\rightarrow$ $\frac{\partial \mathcal{L}}{\partial W_{L-1}} = \boxed{\frac{\partial \mathcal{L}}{\partial h_L}} \frac{\partial h_L}{\partial W_{L-1}}$

$$\frac{\partial \mathcal{L}}{\partial h_L} \boxed{\frac{\partial h_L}{\partial h_{L-1}}} \frac{\partial h_{L-1}}{\partial W_{L-1}}$$

  - 주황색 박스부분은 이미 이전에 계산된 부분을 재사용
  - 연두색 박스부분 역시 계산이 가능
  - 위 과정을 반복하면 모든 weight, bias에 대해서 gradient를 계산 가능

Table of Contents

- Practice
  - Implement Backpropagation
  - Multilayer Perceptron

Implement Backpropagation

- Logistic regression을 backpropagation으로 구현하기
  - $y = \sigma(Wx + b)$

$$z = w^T x + b$$

$$\hat{y} = a = \sigma(z)$$

$$\mathcal{L}(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$$

Implement Backpropagation(more hard)

- Two-layer neural network에 대한 backpropagation 구현하기

  - Model:

$$y = \sigma(W_2\sigma(W_1x + b_1) + b_2)$$

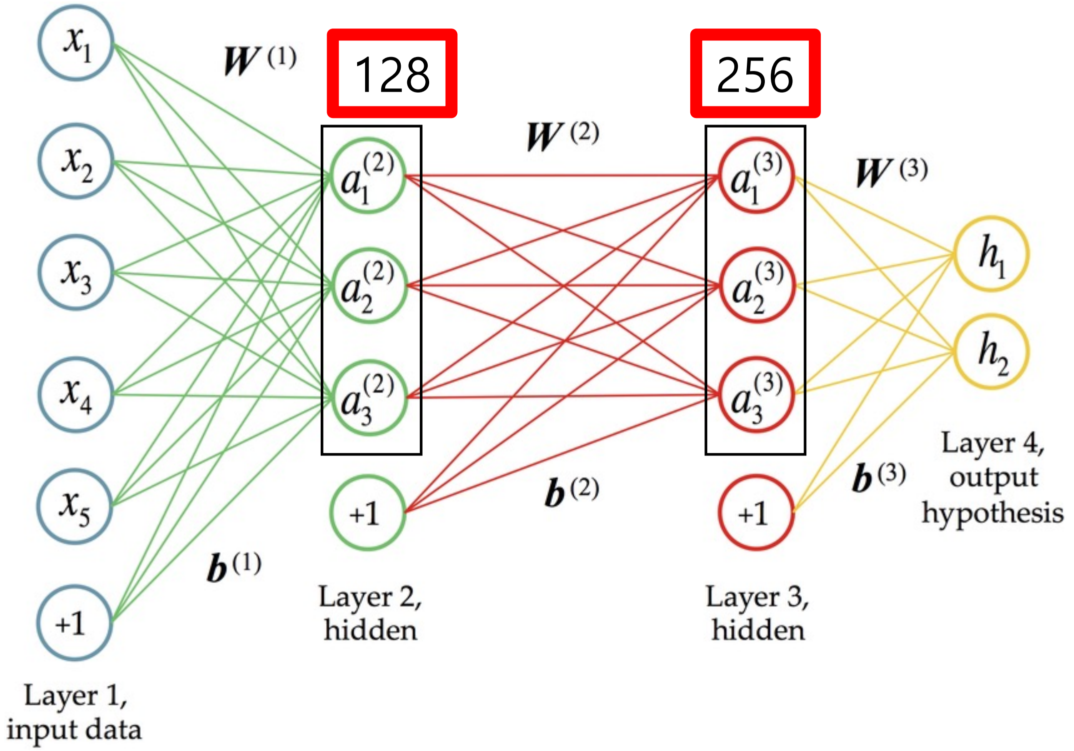
    - Activation function은 sigmoid function사용

  - Loss function :

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})$$

Multilayer Perceptron

- MNIST 데이터를 classify하는 multilayer neural network 구현하기
  - Network design : 2 hidden layer (1st hidden layer : 128, 2nd hidden layer : 256)



Thank You :)

saemi@postech.ac.kr