2023 D&A

Deep Session 2차시

퍼셉트론, 오차역전파법

CONTENTS

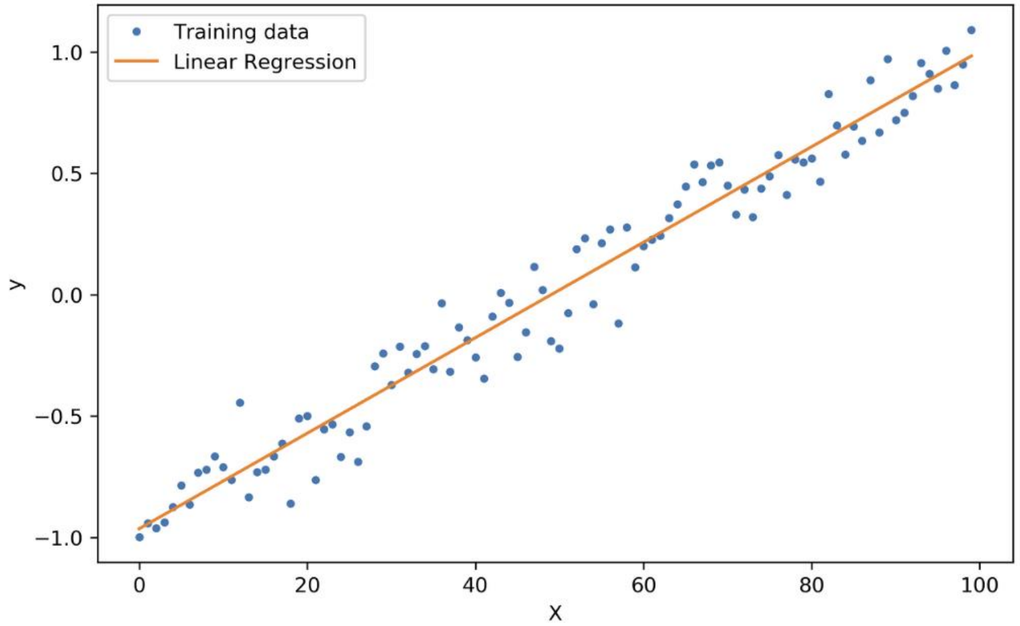
머신러닝

머신러닝 review

Task

데이터를 활용하여 현실의 문제를 해결 or 차선책을 제시

- 1인당 GDP와 삶의 만족도 간의 관계
- 대선 주자의 소득과 지지율 사이 관계 ex) 선형회귀, DT
- 고객의 사진을 보고 고객의 나이 예측



머신러닝 모델의 목적

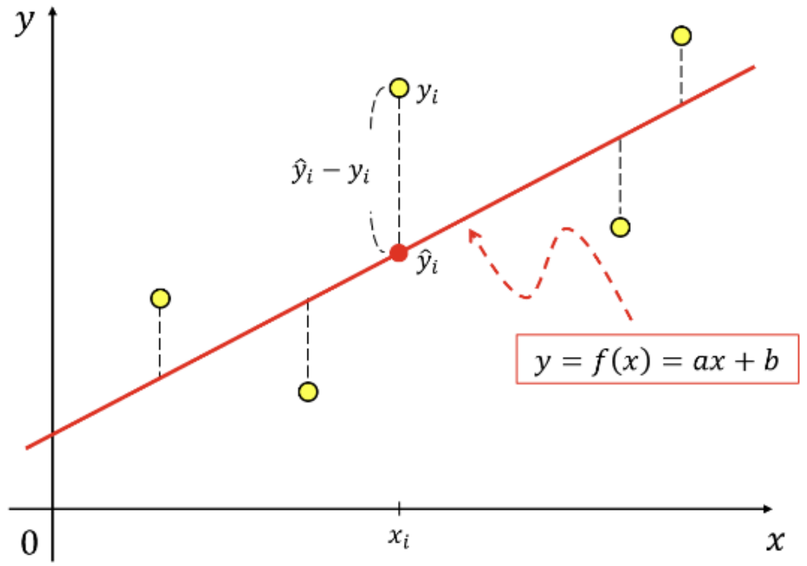
- 기계에게 목적을 부여하기 위한 작업을 수행
- 비용함수 최소화
- $ex)\ cost(W, b) = \frac{1}{N} \sum_{(x,y)\epsilon D} \left(y - prediction(x)\right)^2$

머신러닝
비용함수

비용함수

머신러닝의 핵심 : 기계에게 내 의도를 전달하는 것
- 컴퓨터에게 내 의도를 전달하기 위해 알맞은 목적을 부여해야 한다.

- 비용함수(Cost Function) 정의
  - 설계자의 의도에 맞게 정의 가능
  - 회귀분석에서 대표적인 비용함수 MSE
  - $\boldsymbol{ex)}\ \boldsymbol{cost(W,b)} = \frac{1}{N}\sum_{(x,y)\epsilon D}\left(y - \boldsymbol{prediction(x)}\right)^2$
  - 선형회귀식 : $\hat{y} = \boldsymbol{Wx + b}$
  - $\boldsymbol{cost(W,b)}$가 최소가 되게 하는 $\boldsymbol{W}$와 $\boldsymbol{b}$를 구하는 것이 목표



머신러닝
선형회귀

선형회귀

변수가 p인 선형회귀에서의 회귀식 (p개의 특성이 있는 회귀식)

$$\hat{y} = X\theta + bias$$

- θ는 가중치 행렬 p X 1
- X는 input 데이터 행렬 n X p
- $bias$는 편향 p X 1

  $ex)\ \theta_1 x_{11} + \theta_2 x_{12} + \ \ldots + \theta_p x_{1p}$

- 최솟값을 구하면, 비용이 최소가 되므로 나의 목적과 가까운 결과를 얻을 수 있다.
  - 그렇다면 어떻게?

① 쌩으로 미분하기
② 경사하강법

경사하강법

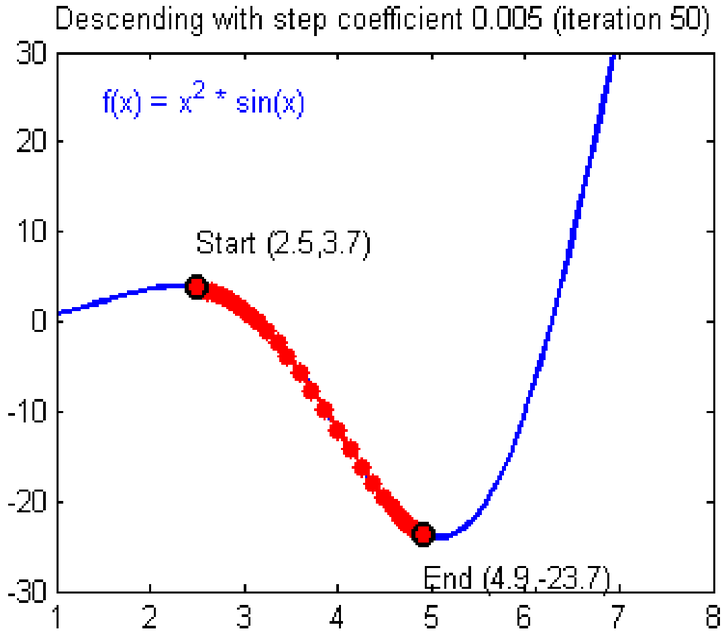
경사하강법이란?

쌩으로 미분하기

- $cost(W, b) = \frac{1}{N} \sum_{(x,y)\epsilon D} \left(y - prediction(x)\right)^2$
- 미분 결과값 : $\hat{\theta} = \left(X^T X\right)^{-1} X^T Y$ (선형회귀의 정규방정식)

위 식은 convex하기 때문에 전역 최소값을 갖는다. ➡ 식의 최솟값은 1개!

- but 행렬 곱, 역행렬 계산은 X가 많다면 매우 어렵다.
- 보다 효율적으로 계산할 수 있는 방법은 없을까?
  - 한번에 미분하지 말고 차근차근 내려가자!
  - 경사하강법의 등장



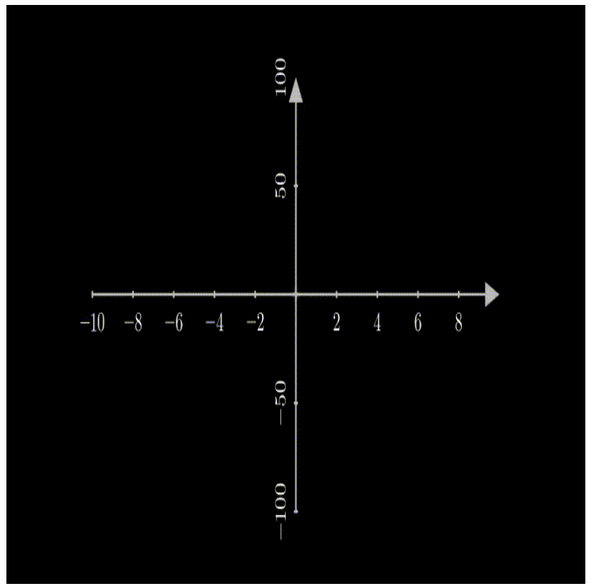
경사하강법

경사하강법이란?

경사하강법(Gradient Descent)

- 한 지점에서 기울기를 구한 뒤, 기울기가 감소하는 방향으로 차근차근 내려가는 방법
- 매개변수를 업데이트할 때, 비용 함수의 기울기를 사용하여 현재 위치에서 가장 가파른 경사 하강 방향으로 이동

- 최적화 과정에서 점진적으로 더 작은 손실 값을 구하는 iterative한 방법



주의할 점

- X는 input 데이터(= 고정 값 ≠ 비용함수 공간에서 움직이는 변수 값), y는 output 값
- 선형회귀에서의 기울기

- 우리가 찾는 것은 $\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{w}$
- $\boldsymbol{ex)}\ \frac{\partial}{\partial \boldsymbol{\theta}}\ \boldsymbol{MSE(\theta)} = \frac{\boldsymbol{2}}{\boldsymbol{m}} \sum^{\boldsymbol{m}} (\boldsymbol{\theta^T x^{(i)}}\ - \boldsymbol{y^{(i)}})\ \ \boldsymbol{x_j^{(i)}}$

경사하강법
Learning rate

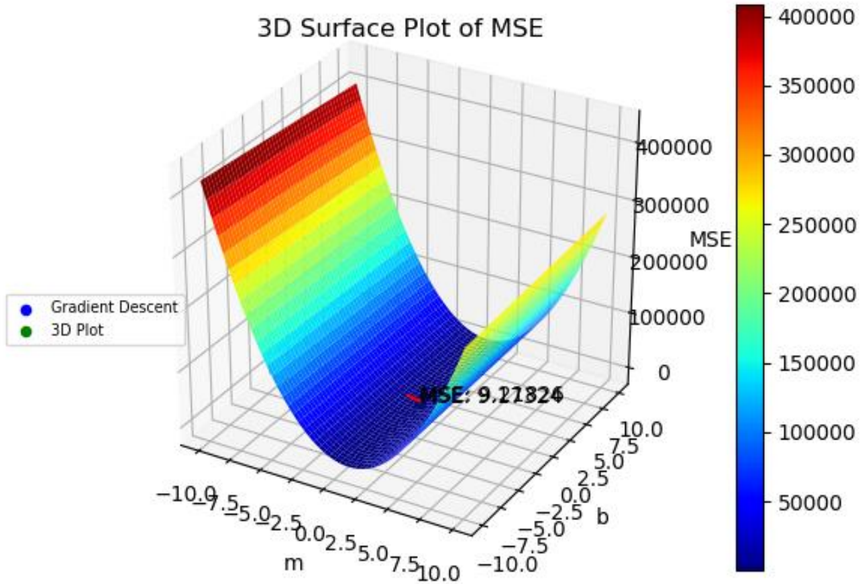
Learning rate

- 경사하강법 : 한 걸음씩 움직이면서 비용함수가 작아지는 지점을 찾는 방법
- 걸음마다 보폭은 어떻게 설정할 것인가? ➡ 보폭 : Learning rate

- learning rate가 지나치게 큰 경우(보폭이 매우 큰 경우)?
- learning rate가 지나치게 작은 경우(보폭이 매우 작은 경우)?

- $W := W - \alpha \frac{\partial}{\partial W} cost(W)$ … α = learning rate

- learning rate 설정은 설계자의 몫이다.



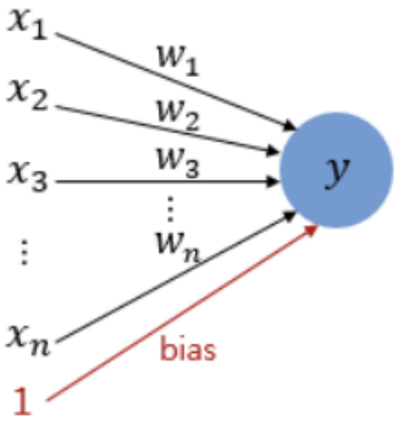
MSE 함수의 vector space

- Cost Function으로 정의한 함수 값의 파라미터 공간
- MSE에서 θ로 표현된 공간 (오른쪽 그림에서 변수는 총 2개 - θ_1, θ_2 / z축 값은 Cost Function)
- 데이터의 분포가 달라지면 함수 형태는 달라지지만 공간 차원은 불변

인공신경망
퍼셉트론

딥러닝 사용 이유

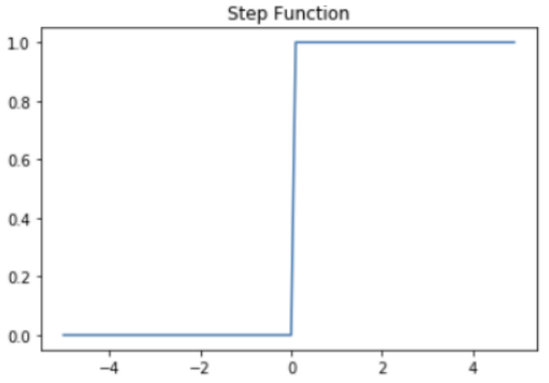
- 머신러닝 문제에 비해 보다 복잡한 task들을 해결한다.
- CNN, RNN 등 복잡한 비선형 문제들을 다룰 수 있다. ➡ 고전적 머신러닝의 한계점?



퍼셉트론

- Frank Rosenblatt가 1957년에 제안한 초기 형태의 인공 신경망
- 다수의 신호를 입력 받아, 하나의 결과를 출력하는 형태

- 다수의 입력 데이터를 넣어 신호가 일정 크기 이상이 되면 값을 출력한다.
- $if\ \sum_i^n W_i x_i + b \geq 0 \rightarrow y = 1$
- $if\ \sum_i^n W_i x_i + b < 0 \rightarrow y = 0$
- 각각의 입력신호를 통해 보내진 입력값 x는 각각의 가중치 w와 함께 인공 뉴런에 전달되어 y를 출력한다.

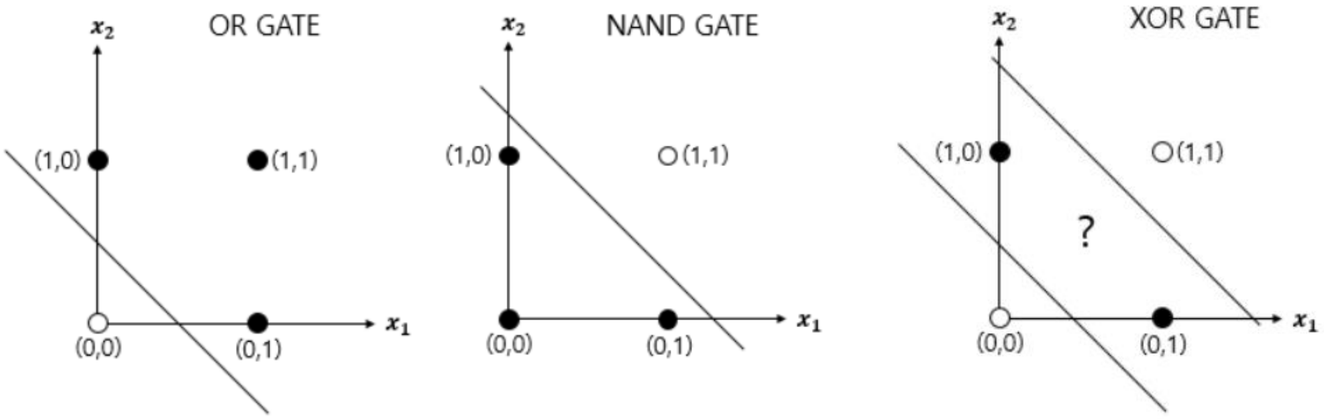


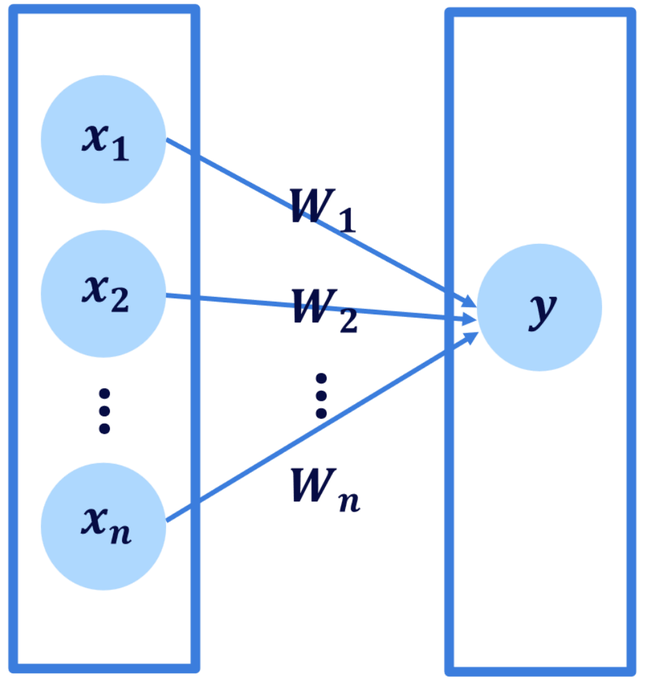
인공신경망
퍼셉트론

단층 퍼셉트론의 한계

- 선형분류만 가능
- XOR 게이트(같으면 0, 다르면 1 출력) 구현 불가능

- 선을 긋는 행위의 의미?





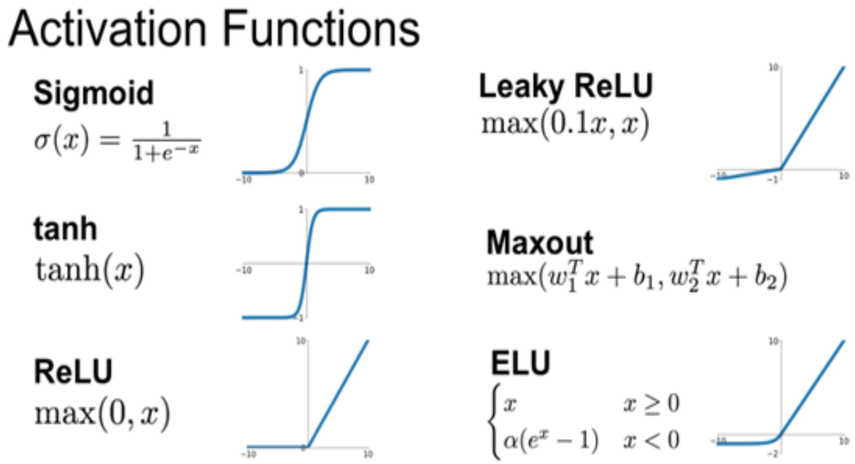
인공신경망

Activation Function

Activation Function(활성화 함수)

- 비선형 함수
- Sigmoid, ReLU 등의 함수를 의미한다.

- 활성화 함수는 비선형성을 가해주는 매우 중요한 역할을 한다.
- 활성화 함수를 사용하면 입력값에 대한 출력값이 nonlinear로 도출되므로 선형분류기를 비선형분류기로 변환 가능하다.
- 퍼셉트론은 선형 결합 후 비선형 함수를 통과한다.



인공신경망

Activation Function

신경망에서 선형 함수를 사용하게 된다면?

- 신경망의 층을 깊게 쌓는다는 의미가 사라진다.
- hidden layer가 없는 네트워크로도 같은 기능을 수행할 수 있다.

- $if\ h(x) = cx \rightarrow y(x) = h\left(h\left(h(x)\right)\right) = c \times c \times c \times x = c^3x$
- $y = ax$에서 '$a = c^3$인 선형함수 1층'으로 구성된 네트워크와 다른 점이 없다.

인공신경망

MLP

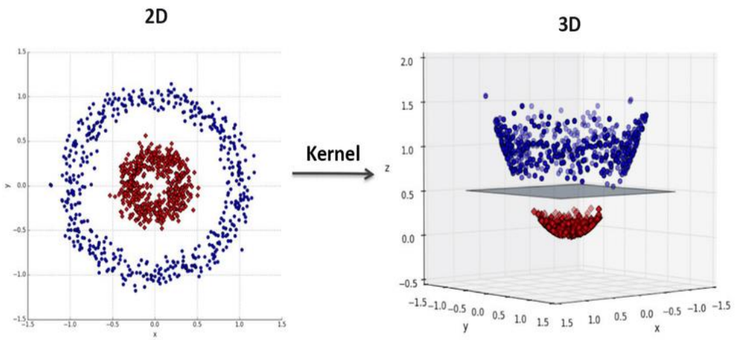
선형 분류만 가능하다는 것의 의미는?

- 퍼셉트론은 선형 결합만을 진행한다.
- 여러 개의 퍼셉트론을 쌓더라도 하나의 퍼셉트론에서 진행되는 일은 결국 '직선 긋기'이다.
- 직선을 긋는다고 하더라도 정의역 구간이 바뀐다면 어떻게 될까? (시점변환)

예시) $AND(\overline{x_1}, x_2), AND(x_1, \overline{x_2})$ 를 통해 비선형 데이터 정렬

- $AND(\overline{x_1}, x_2)$: $\overline{x_1}, x_2$ 둘 다 1이면 1 출력, 그렇지 않으면 0 출력
- $\overline{x_1}$ 은 x_1의 부정

- 하나의 퍼셉트론은 선형이지만, 벡터를 다른 관점에서 바라본다면?
- 하나의 퍼셉트론을 지날 때마다 관점이 달라진다면, 비선형 효과를 낼 수 있지 않을까?

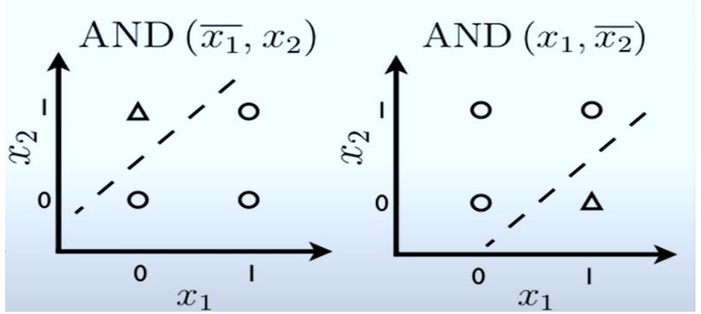


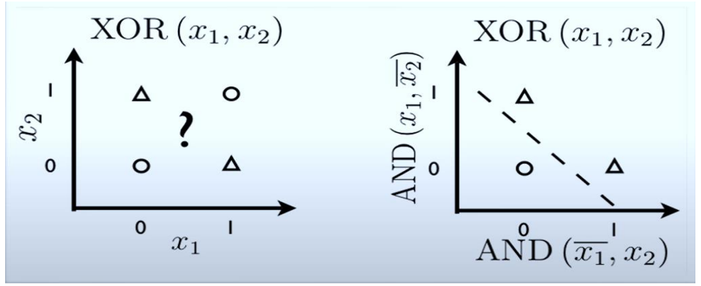
인공신경망

MLP

선형 분류만 가능하다는 것의 의미는?

$XOR(A_1, A_2)$	$XOR(A_1, A_2)$	$AND(x_1, \overline{x_2})$	x_1	x_2
0	0	0	0	0
0	0	0	1	1
1	1	0	0	1
1	0	1	1	0





- 각각의 퍼셉트론은 선형 결합만 가능하다. ➡ 직선 긋기의 문제
- 다음 퍼셉트론을 넘어 갈 때, 축은 변경된다. ➡ 합성함수의 정의역 공간, 시점변환

- 최종적으로 x에 대해 바라봤을 때, 비선형 형태로 출력이 된다.

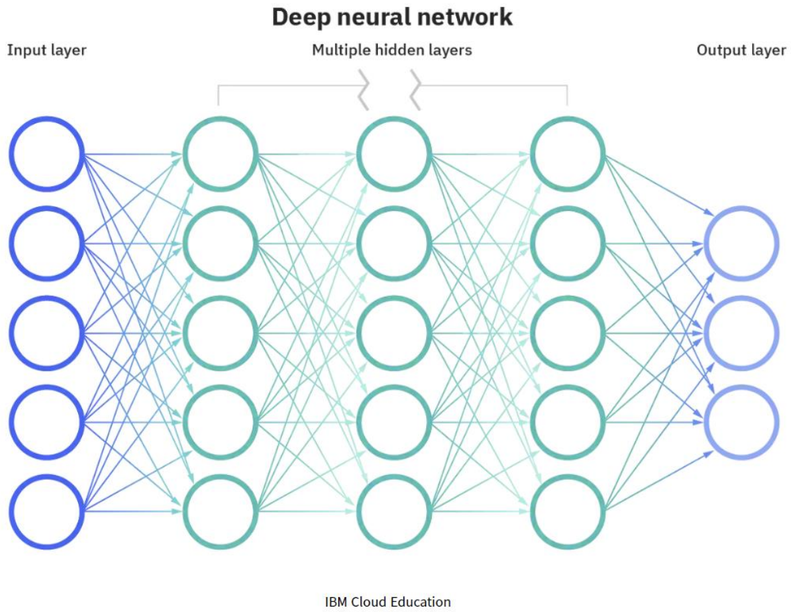
인공신경망

MLP

MLP(Multi Layer Perceptron)

- 여러 개의 layer를 쌓아 올린 형태로 구성되어 있는 모델
- 비선형 분류가 어렵다는 단일 퍼셉트론의 한계점을 극복하기 위해 등장

- 딥러닝의 기본 구조가 되는 신경망(Neural Network)를 의미한다.
- 심층 신경망(Deep Neural Network) ➡ 딥러닝(Deep Learning)



기본 구조

- 입력층(input layer)
- 은닉층(hidden layer)
- 출력층(output layer)
- 노드

오차역전파법

순전파와 역전파

딥러닝 모델의 목표

- Minimizing the cost function
- cost function을 정의하여 network의 실효성을 판별한다.

가정

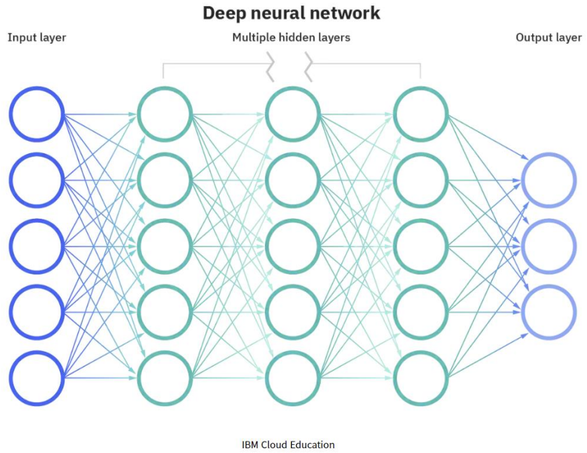
- cost function은 MSE로 정의한다.
- $\boldsymbol{cost}(\boldsymbol{W}, \boldsymbol{b}) = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{(\boldsymbol{x},\boldsymbol{y})\epsilon \boldsymbol{D}} \left(\boldsymbol{y} - \boldsymbol{prediction}(\boldsymbol{x})\right)^{\mathbf{2}}$
- 효율적인 학습법인 경사하강법을 어떻게 적용해야 할까?

오차역전파법
순전파와 역전파

순전파(Feed Forward, Forward Propagation)

- MLP의 Parameter를 활용하여 결과값을 계산하는 방법
- 이전 layer에서 넘어온 값에 가중치(w)와 편향(b)를 적용해 다음 layer로 넘기는 방식이다.

- $\hat{y}$ 와 실제 y 를 비교하여 $cost$(오차)를 계산한다.
- 계산그래프의 출발점(왼쪽)부터 종착점(오른쪽)으로의 전파를 의미한다.



역전파(Backpropagation)

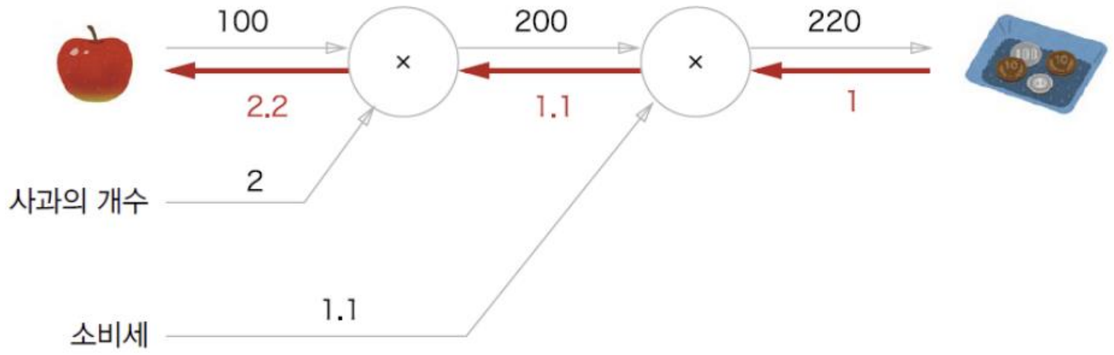
- MLP의 parameter를 update하는 과정
- 순전파 과정을 통해 나온 cost(오차)를 활용하여 각 layer의 weight와 bias를 최적화한다.

- 순전파의 결과로부터 cost function이 최소화되는 방향으로 weight와 bias를 수정한다.
- 계산그래프의 종착점(오른쪽)에서 출발점(왼쪽)으로의 전파를 의미한다. (↔ 순전파 방향)

오차역전파법

순전파와 역전파

계산그래프를 통한 미분 문제 해결



- 순전파 : → 방향
- 역전파 : ← 방향

- 역전파는 국소적 미분을 전달한다.
- 사과가 1원 오르면 최종 금액은 2.2원 오른다는 의미

오차역전파법

오차역전파법

연쇄법칙(Chain Rule)

- 역전파에서 '국소적 미분'을 순방향과 반대 방향으로 전달하는 원리는 연쇄법칙을 따른다.
- 연쇄법칙 : 합성 함수의 미분에 대한 성질

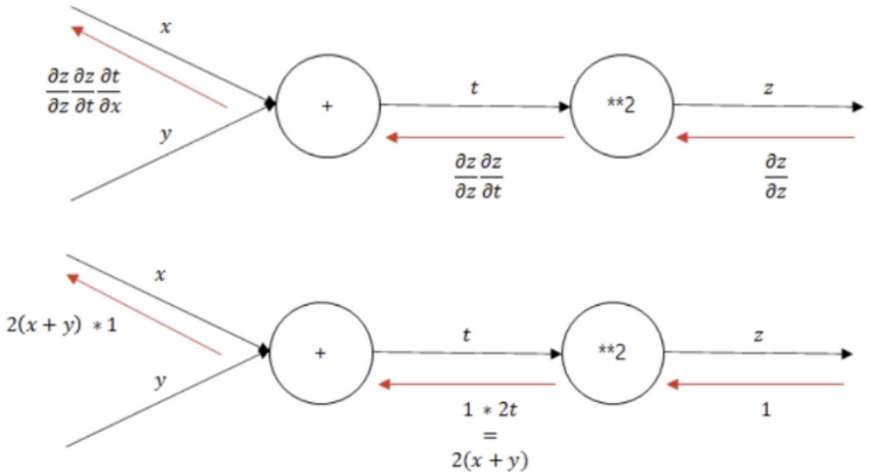
> 합성 함수의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다.

수학적으로 확인하기

- $z = (x + y)^2$
  - $z = t^2$
  - $t = x + y$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x}$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x} = 2t \times 1 = 2(x + y)$$

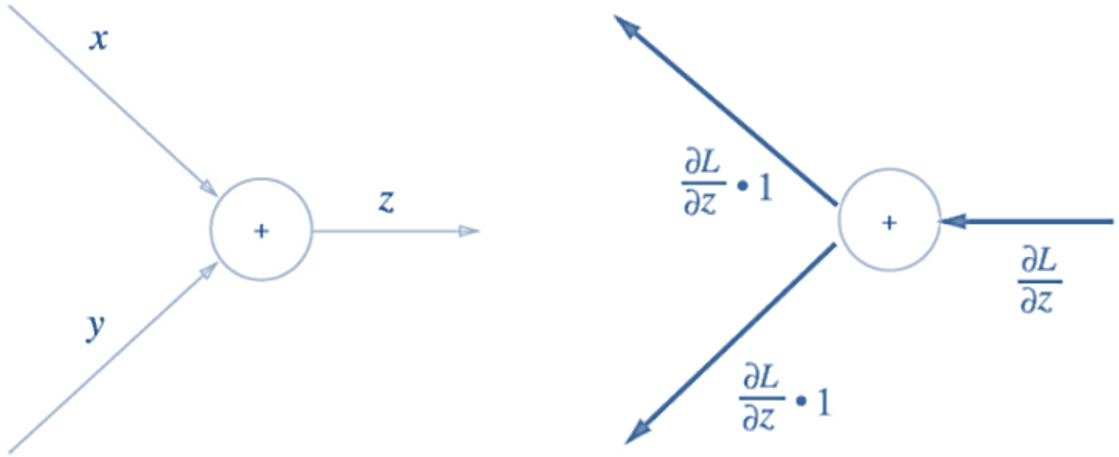


오차역전파법

오차역전파법

덧셈 노드의 역전파

- 입력값을 그대로 흘려 보낸다.
- 덧셈 노드의 역전파는 1을 곱하기만 할 뿐, 입력된 값을 그대로 다음 노드로 전달한다.

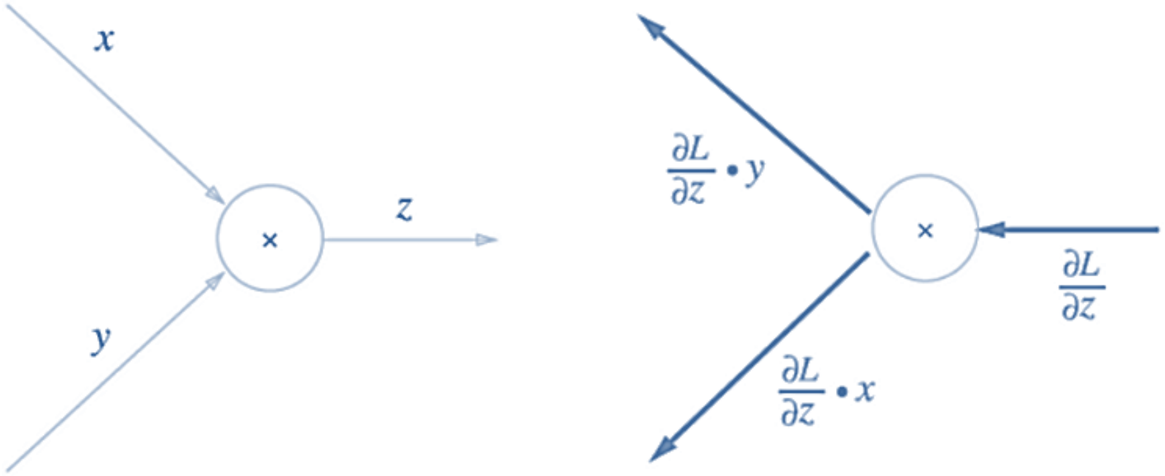


오차역전파법

곱셈 노드의 역전파

- 상류의 값에 입력 신호들을 서로 바꾼 값을 곱해서 하류로 보낸다.
- 순전파 때 x였다면 역전파에서는 y, 순전파 때 y였다면 역전파에서는 x로 바뀐다는 의미이다.

- 곱셈의 역전파는 덧셈의 역전파와 다르게 순방향 입력 신호값이 필요하므로, 곱셈 노드를 구현할 때 순전파의 입력 신호를 변수에 저장한다.



오차역전파법

출력층

출력층(신경망 학습 경우)

- 출력층 노드는 softmax 함수를 거쳐 각 정답 노드에 얼마만큼의 영향력이 있는지 살펴본다.

- $Softmax(t_i) = \frac{\exp(t_i)}{\sum j=1^{K} \exp(t_j)}$
- K개의 클래스에서 i번째 원소가 정답일 확률
- 각 출력을 확률 형태로 인식한다.

2023 D&A

Deep Session 2차시

THANK YOU

2023. 03. 16