2. 머신러닝 분석 개요

박경미

목차

- 머신러닝 분석 개요
- 머신러닝 수식
- 미분코드

1. 머신러닝 개요

❖ 인공지능, 머신러닝, 딥러닝

- 인공지능 (Artificial Intelligence)

  인간의 학습능력, 추론능력 등을 컴퓨터를 통해 구현하는 포괄적인 개념

- 머신러닝 (Machine Learning)

  데이터를 이용하여 명시적으로 정의되지 않은 패턴을 학습하여 미래 결과(값, 분포)를 예측

  ※ 데이터마이닝(Data Mining): 데이터간의 상관관계나 속성을 찾는 것이 주목적

- 딥러닝 (Deep Learning)

  머신러닝의 한 분야로서 신경망(Neural Network)을 통하여 학습하는 알고리즘의 집합

1. 머신러닝 개요

❖ 머신러닝 정의

- <u>머신러닝은 데이터에서부터 학습하도록 컴퓨터를 프로그래밍하는 과학(또는 예술</u>

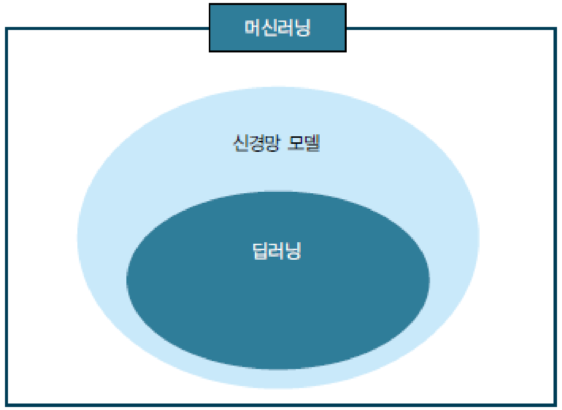
- <u>“머신러닝은 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구 분야”</u>
<u>-아서 새뮤얼(Arthur Samuel), 1959</u>

- <u>“어떤 작업 T에 대한 컴퓨터 프로그램의</u> 성능을 P로 측정했을 때 경험 E로 인해 성능이 향상됐다면, 이 컴퓨터 프로그램은 작업 T와 성능 측정 P에 대해 경험 E로 학습한 것”
-톰 미첼(Tom Mitchell), 1997

1. 머신러닝 개요

❖ 머신러닝?

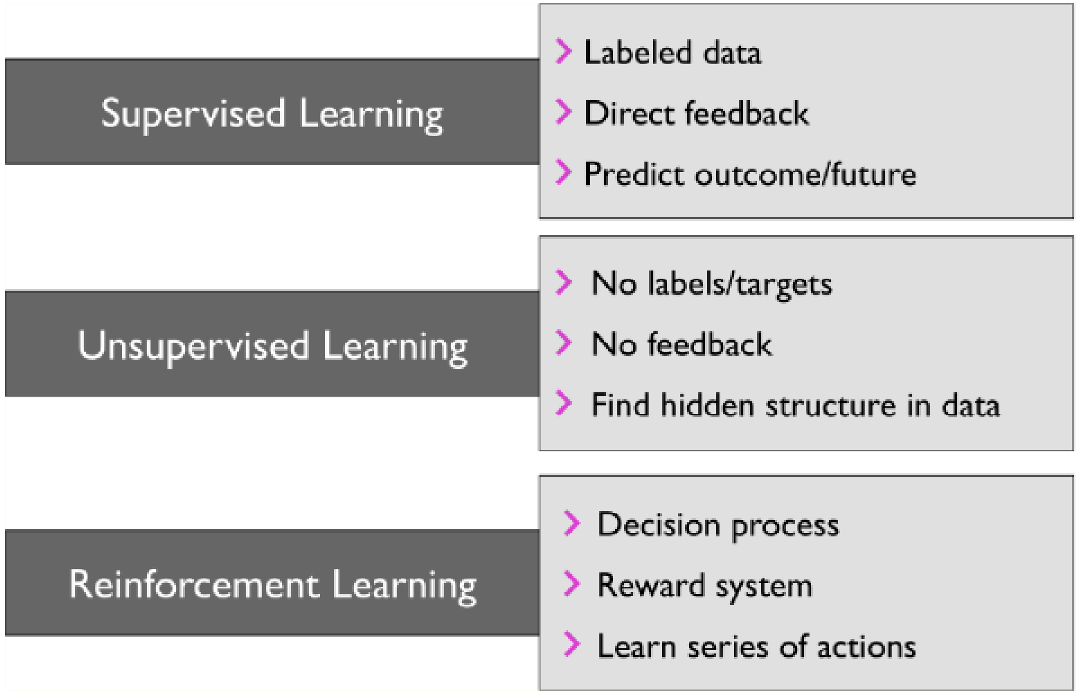
- 데이터에서 법칙성을 추출하는 통계적 방법'의 하나.
- 법칙을 추출하여 예측, 분류하는 다양한 모델(알고리즘).
- 손글씨 문자 식별, 물체 식별, 질병 진단 등 다양한 분야에 걸쳐 응용.
- 딥러닝도 머신러닝의 일부로, 신경망 모델의 한 형태.



1. 머신러닝 개요

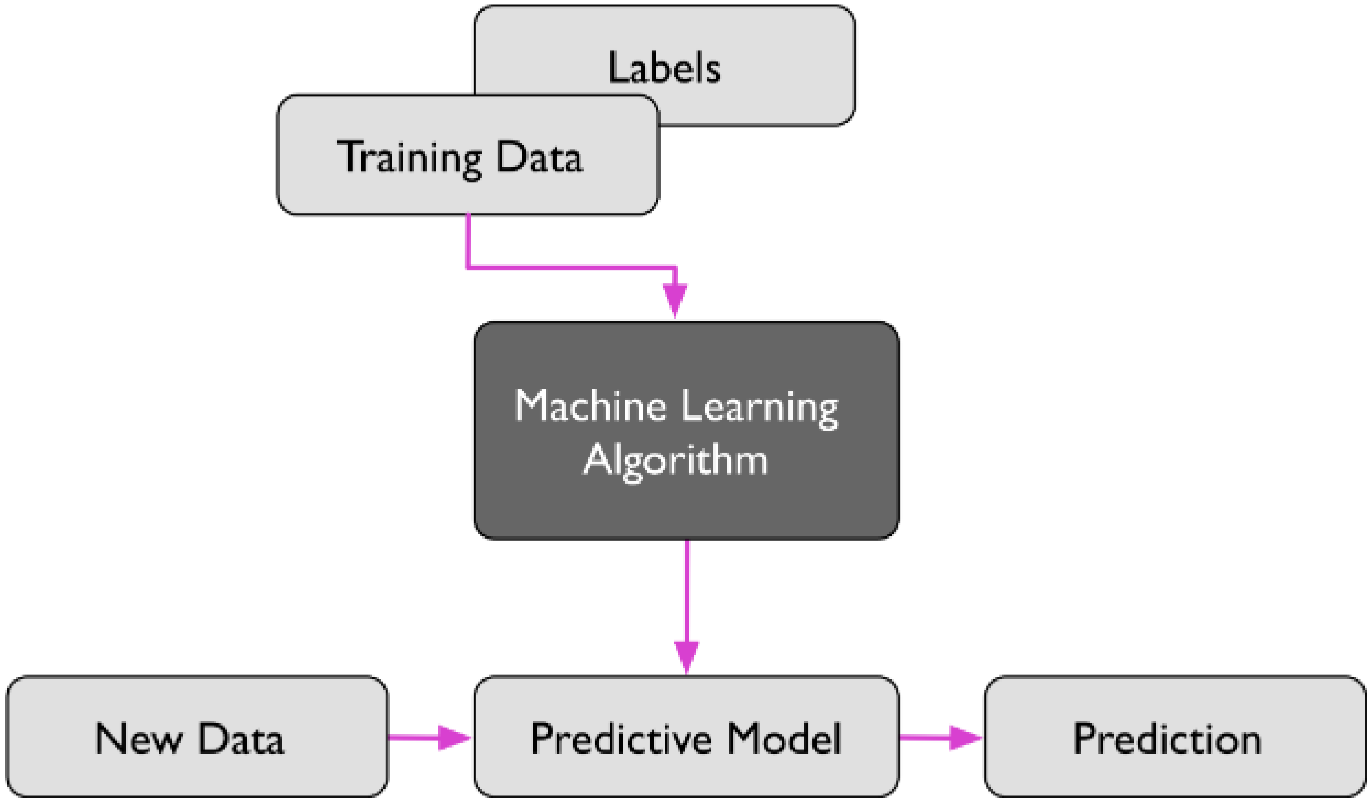
❖ 머신러닝 문제의 분류

- 지도 학습 Supervised Learning 문제
  - 입력에 대한 적절한 출력을 구하는 문제
  - Regression, Classification
- 비지도 학습Unsupervised Learning 문제
  - 비지도 학습은 입력 정보의 특징을 찾는 문제
  - Clustering
- 강화 학습 문제
  - 강화 학습은 장기나 체스와 같이 마지막 결과가 가장 좋은 행동을 찾는 문제.



1. 머신러닝 개요

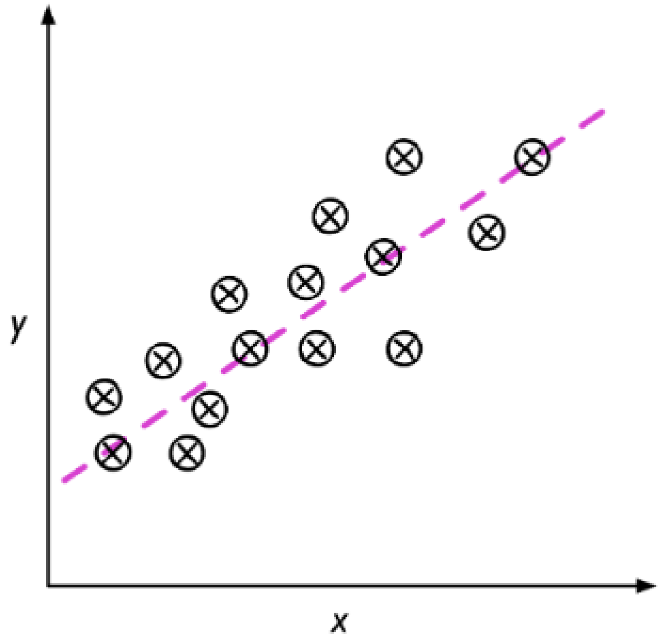
❖ 지도 학습으로 미래 예측하기

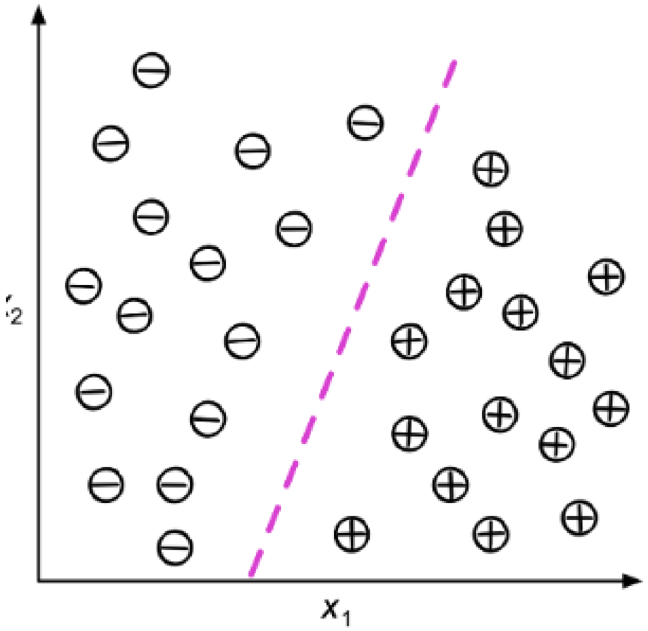


1. 머신러닝 개요

❖ 지도 학습으로 미래 예측하기

- Regression, Classification



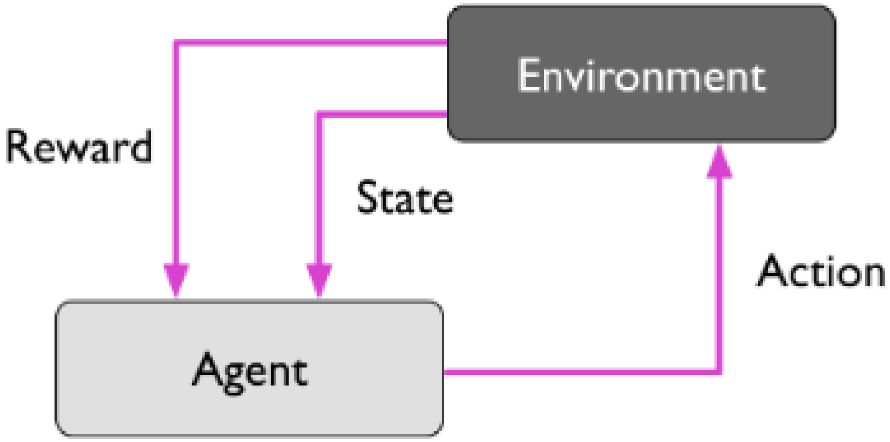


- 예측 변수(특성) vs 반응 변수(타깃)
- 선형 회귀(linear regression)

- 이진 분류(binary classification) vs 다중 분류(multiclass classification)
- 양성 클래스(positive class) / 음성 클래스(negative class)
- 결정 경계(decision boundary)

1. 머신러닝 개요

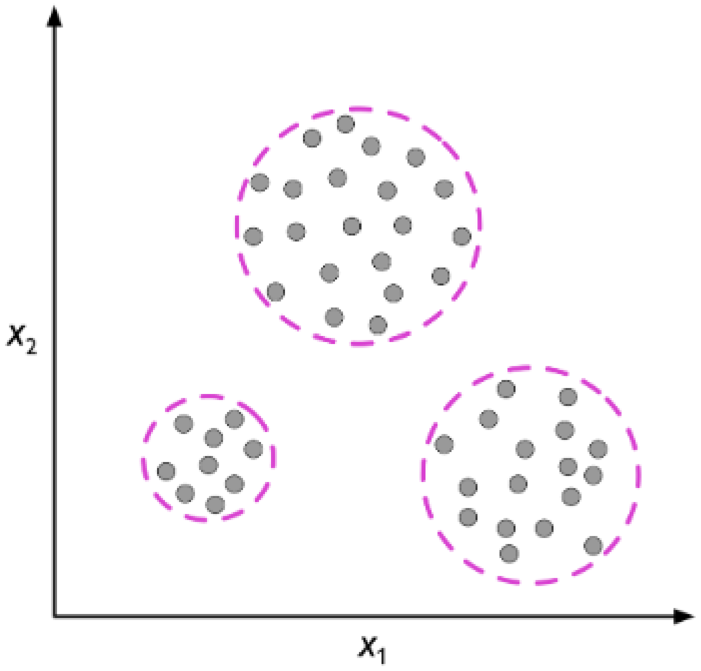
❖ 강화 학습으로 반응형 문제 해결



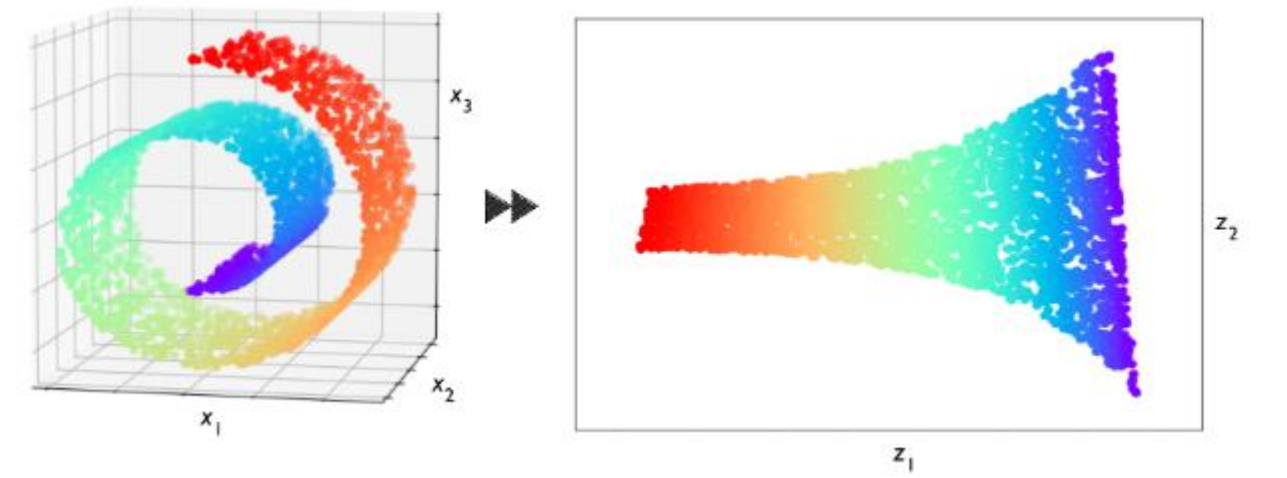
1. 머신러닝 개요

❖ 비지도 학습으로 숨겨진 구조 발견

- 군집: 서브 그룹 찾기

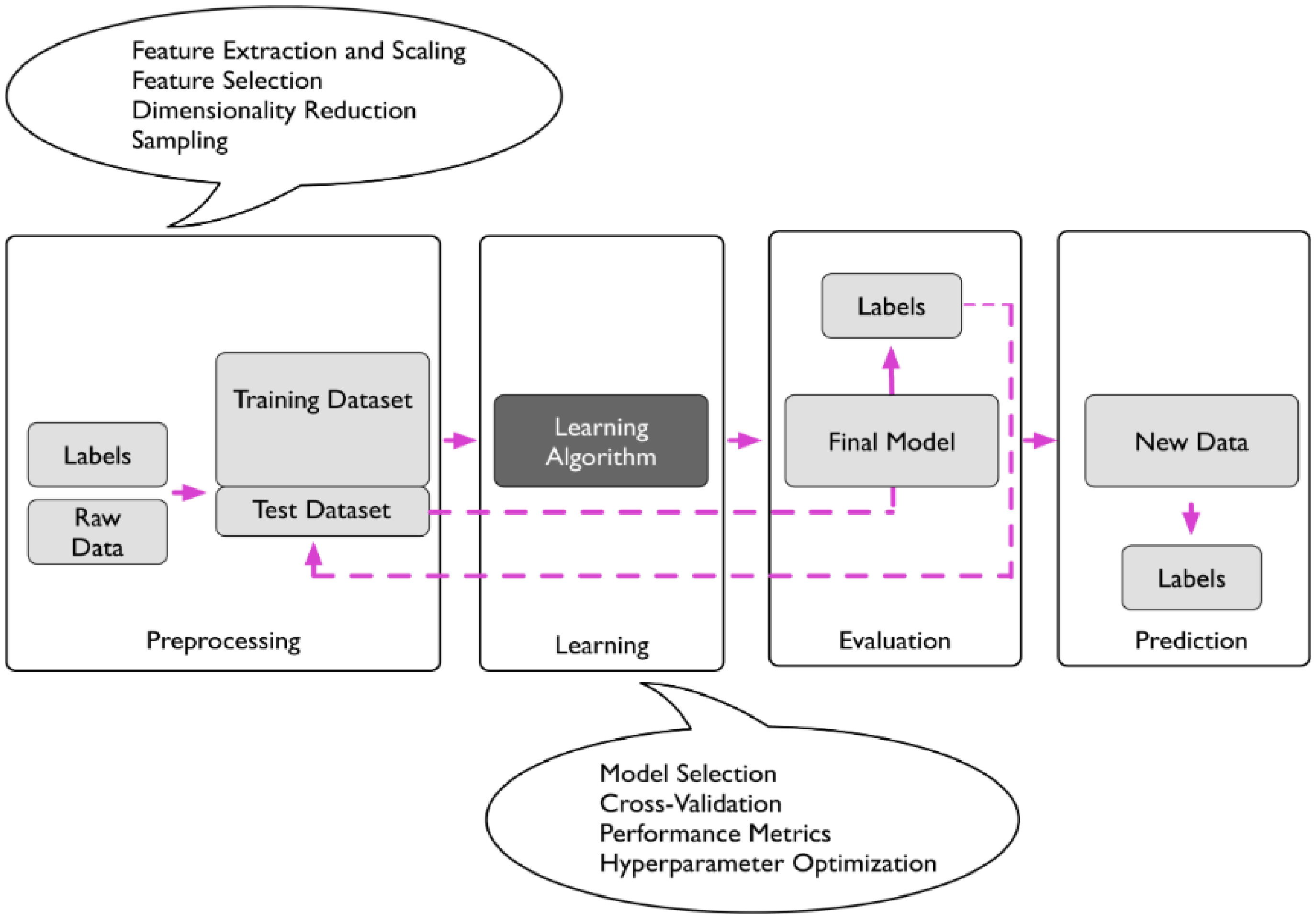


- 차원축소 : 데이터 압축



1. 머신러닝 개요

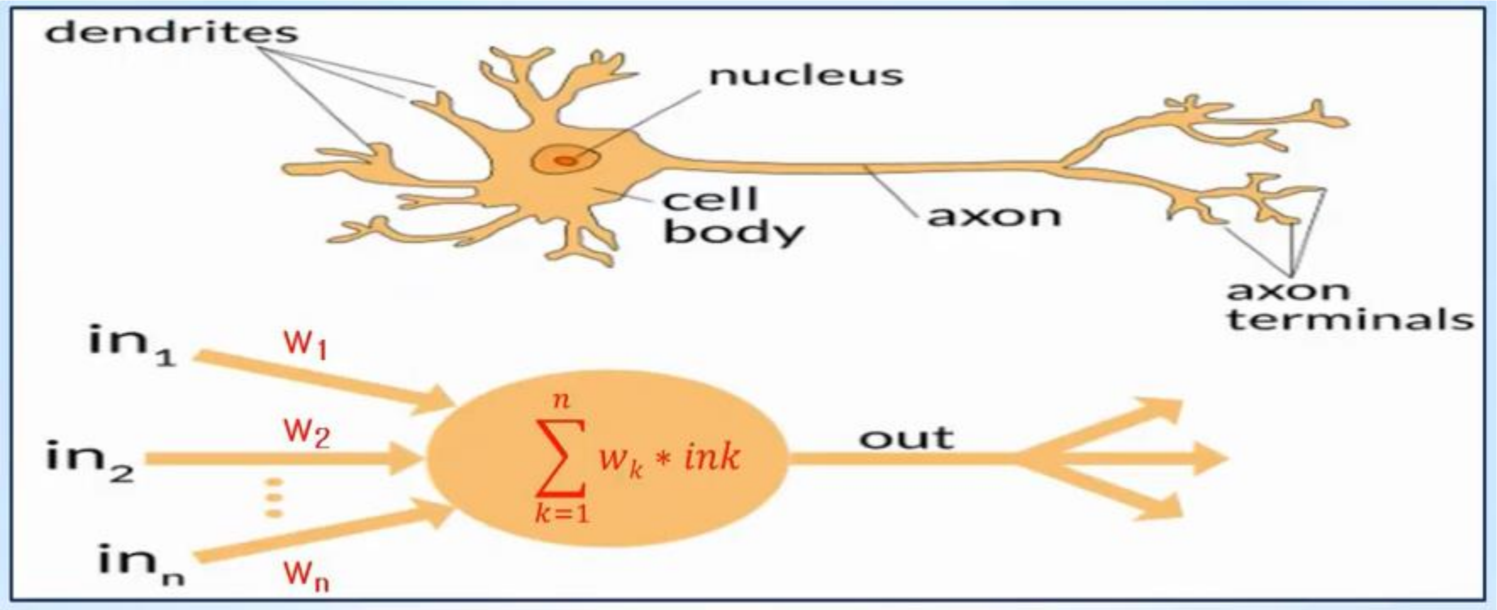
❖ 머신러닝 시스템 구축 로드맵



1. 머신러닝 개요

❖ 딥러닝 – Neural Networ

- 신경망(Neural Network) 인간의 뇌 세포(Neuron) 상호작용을 모방함
- 이전 뉴런 출력을 각각의 입력으로 받아, + 또는 –가중치를 통해 전체 합을 합하여 특정 임계치(threshold)를 넘으면 다음 뉴런으로 전달하는 원리



❖ 머신러닝/딥러닝 구현 - 프레임워크

Caffe

TensorFlow

theano

- 머신러닝 프레임워크(Framework) 사용 단점
  - 머신러닝의 알고리즘을 API로 추상화 함으로서 개발을 쉽고 빠르게 할 수 있지만, 동작원리와 내부 구현을 자세히 알 수 없는 블랙박스(Black Box) 로서 동작함

INPUT → BLACK BOX → OUTPUT

python™

- 파이썬(Python) 직접 구현 시 장점
  - 머신러닝 동작원리를 자세히 알 수 있어 재미있으며 동시에 알고리즘에 대한 깊은 이해 가능
  - 새로운 머신러닝 알고리즘이 나왔을 때 빠르게 코드를 이해할 수 있는 insight 획득

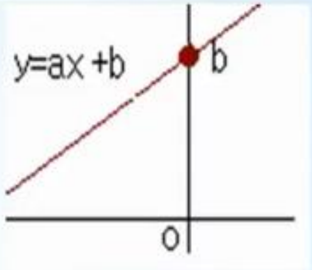
2. 머신러닝 수학

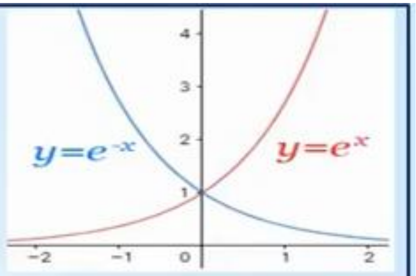
❖ 머신러닝 수식

➢ 기본 수학 개념

$y = ax + b$ $\qquad$ $y = e^x$

$$\sum_{k=1}^{2} A_k = A_1 + A_2 \qquad \prod_{k=1}^{2} A_k = A_1 \times A_2$$





➢ 행렬 (matrix) 연산

- 산술연산 (+, −, *, /)

$$\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} + \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} = \begin{pmatrix}2 & 2\\3 & 5\end{pmatrix}$$

(2 X 2) + (2 X 2) = (2 X 2)

- 행렬 곱 (dot product)

$$\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} \cdot \begin{pmatrix}1 & 0 & 0\\0 & 1 & 0\end{pmatrix} = \begin{pmatrix}1 & 2 & 0\\3 & 4 & 0\end{pmatrix}$$

(2 X 2) · (2 X 3) = (2 X 3)

2. 머신러닝 수학

❖ 미분

미분 – derivative

미분을 왜 하는가? 미분으로 얻을 수 있는 인사이트?

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

f(x) 를 미분하라

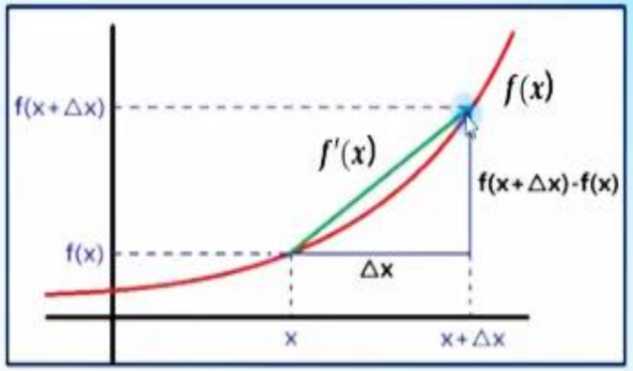
⇒ 입력변수 x 가 미세하게 변할때, 함수 f 가 얼마나 변하는지 알 수 있는 식을 구해라

⇒ 함수 f(x) 는 입력 x의 미세한 변화에 얼마나 민감하게 반응하는지 알 수 있는 식을 구해라

insight

⇒ 입력 x 를 현재 값에서 아주 조금 변화시키면, 함수 f(x)는 얼마나 변하는가 ?

⇒ 함수 f(x)는 입력 x의 미세한 변화에 얼마나 민감하게 반응하는가 ?



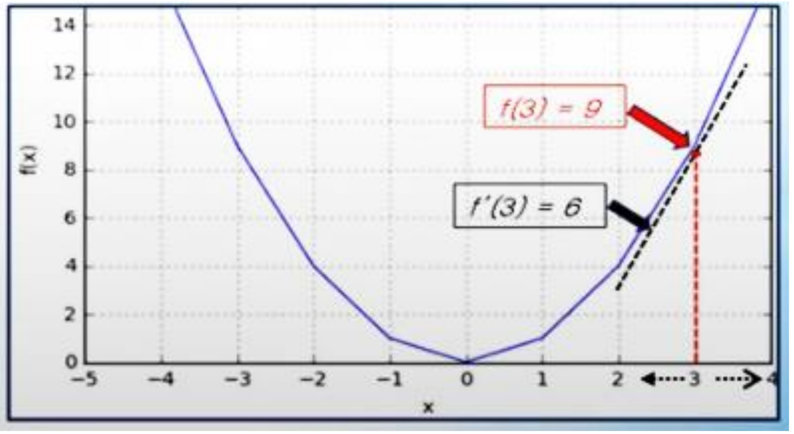
[예] 함수 $f(x) = x^2$ 일 경우, 미분 $f'(x) = 2x$

$f(3) = 9$ 해석

=> 입력 x = 3 에서 출력은 9 임을 의미

$f'(3) = 6$ 해석

=> 입력 x = 3 을 미세하게 변화시킬 때 함수는 현재 입력 값의 2배인 6 배 변화를 일으킴을 의미



2. 머신러닝 수학

❖ 머신러닝/딥러닝에 자주 사용되는 함수 미분

$f(x) = \text{상수} \Rightarrow f'(x) = 0$

$f(x) = ax^n \Rightarrow f'(x) = nax^{n-1}$

$f(x) = e^x \Rightarrow f'(x) = e^x$

$f(x) = \ln x \Rightarrow f'(x) = \dfrac{1}{x}$

$f(x) = e^{-x} \Rightarrow f'(x) = -e^{-x}$

[예 1] $f(x) = 3x^2 + e^x + 7 \Rightarrow f'(x) = 6x + e^x$

[예 2] $f(x) = \ln x + \dfrac{1}{x} \Rightarrow f'(x) = \dfrac{1}{x} - \dfrac{1}{x^2}$

참고

$\dfrac{1}{x} = x^{-1}$

2. 머신러닝 수학

❖ 편미분-partial derivative

- 입력 변수가 하나 이상인 다 변수 함수에서 미분하고자하는 변수 하나를 제외한 나머지 변수들을 상수로 취급하고 해당 변수를 미분하는 것
- 예를 들어 f(x, y)를 변수 x에 대해 편미분 하는 경우 다음과 같이 나타냄

$$\boxed{\frac{\partial f(x,y)}{\partial x}}$$

[예1] $f(x,y) = 2x + 3xy + y^3$, 변수 x 에 대하여 편미분

$$\frac{\partial f(x,y)}{\partial x} = \frac{\partial(2x+3xy+y^3)}{\partial x} = 2 + 3y$$

[예2] $f(x,y) = 2x + 3xy + y^3$, 변수 y 에 대하여 편미분

$$\frac{\partial f(x,y)}{\partial y} = \frac{\partial(2x+3xy+y^3)}{\partial y} = 3x + 3y^2$$

[예3] 체중 함수가 '체중(야식, 운동)' 처럼 야식/운동에 영향을 받는 2변수 함수라고 가정할 경우, 편미분을 이용하면 각 변수 변화에 따른 체중 변화량을 구할 수 있음

현재 먹는 야식의 양에서 조금 변화를 줄 경우 체중은 얼마나 변하는가 ? ➡ $\dfrac{\partial 체중}{\partial 야식}$

현재 하고 있는 운동량에 조금 변화를 줄 경우 체중은 얼마나 변하는가 ? ➡ $\dfrac{\partial 체중}{\partial 운동}$

2. 머신러닝 수학

❖ 연쇄법칙- chain rule

- 합성 함수란 여러 함수로 구성된 함수로서 이러한 합성 함수를 미분하려면 ‘합성함수를 구성하는 각 함수를 미분의 곱 ' 으로 나타내는 chain rule(연쇄법칙을 이용)

[합성함수 예1] $f(x) = e^{3x^2}$ ➡ 함수 e^t , 함수 $t = 3x^2$ 조합

[합성함수 예2] $f(x) = e^{-x}$ ➡ 함수 e^t , 함수 $t = -x$ 조합

- $f(x) = e^{3x^2}$ 을 chain rule로 미분하는 경우, $t = 3x^2$ 으로 놓으면 $f(x) = e^t$

chain rule 적용(약분개념)

t = 3x² 대입

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\cancel{\partial t}}\frac{\cancel{\partial t}}{\partial x} = \frac{\partial(e^t)}{\partial t}\frac{\partial(3x^2)}{\partial x} = (e^t)(6x) = (e^{3x^2})(6x) = 6xe^{3x^2}$$

- $f(x) = e^{-x}$ 을 chain rule로 미분하는 경우, $t = -x$ 으로 놓으면 $f(x) = e^t$

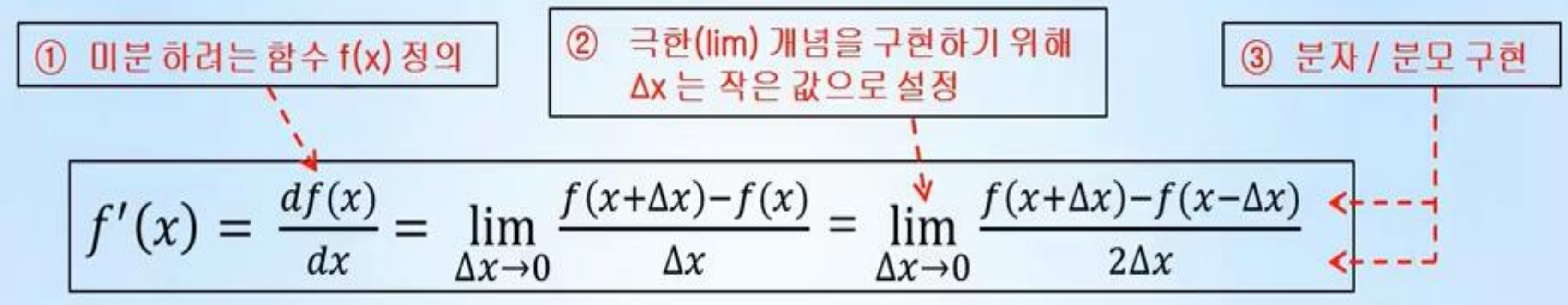
chain rule 적용(약분개념)

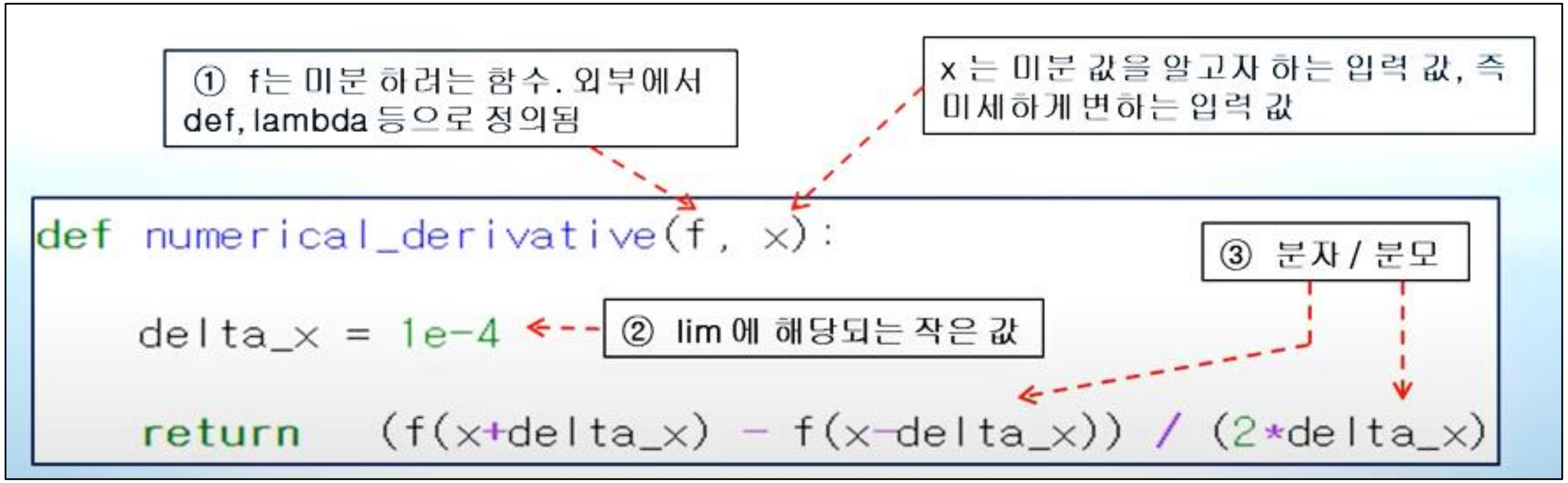
t = −x 대입

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\cancel{\partial t}}\frac{\cancel{\partial t}}{\partial x} = \frac{\partial(e^t)}{\partial t}\frac{\partial(-x)}{\partial x} = (e^t)(-1) = (e^{-x})(-1) = -e^{-x}$$

수치미분 1차 버전-numerical derivative

- 수치미분은 주어진 입력 값이 미세하게 변할 때 함수 값 f는 얼마나 변하는지를 계산하는 것





2. 머신러닝 수학

[예제 1] 함수 $f(x) = x^2$ 에서 미분계수 $f'(3)$ 을 구하기. 즉, x=3 에서 값이 미세하게 변할 때, 함수 f 는 얼마나 변하는지 계산하라는 의미

$f'(3)$ 계산 과정 (참고: $f'(x) = 2x$)

$$f'(3) = \lim_{\Delta x \to 0} \frac{f(3+\Delta x) - f(3-\Delta x)}{2\Delta x}$$

Δx 는 10^{-4} 대입

$$f'(3) = \frac{f(3+1e^{-4}) - f(3-1e^{-4})}{2 * 1e^{-4}}$$

$f(x) = x^2$

$$f'(3) = \frac{(3+1e^{-4})^2 - (3-1e^{-4})^2}{2 * 1e^{-4}}$$

result

$$f'(3) = 6.0$$

```
def my_func1(x):

    return  x**2

def numerical_derivative(f, x):

    delta_x = 1e-4

    return  (f(x+delta_x) - f(x-delta_x)) / (2*delta_x)

result = numerical_derivative(my_func1, 3)

print("result ==", result)
```

```
result == 6.000000000012662
```

2. 머신러닝 수학

[예제 2] 함수 $f(x) = 3xe^x$ 를 미분한 함수를 $f'(x)$ 라고 할 경우, $f'(2)$ 을 구하기.
x=2 에서 값이 미세하게 변할 때, 함수 f 는 얼마나 변하는지 계산하라는 의미

수치 미분

```
import numpy as np

def my_func2(x):

    return  3*x*(np.exp(x))
```

```
def numerical_derivative(f, x):

    delta_x = 1e-4

    return   (f(x+delta_x) - f(x-delta_x)) / (2*delta_x)
```

```
result = numerical_derivative(my_func2, 2)

print("result ==", result)
```

```
result == 66.50150507518049
```

수학공식 검증

$f(x) = 3xe^x$ 미분

↓

$f'(x) = 3e^x + 3xe^x$

↓ $x = 2$ 대입

$f'(2) = 3e^2 + 3 \cdot 2e^2$

↓

```
print("3*exp(2) + 3*2*exp(2) == ", end='')
print(3*np.exp(2) + 3*2*np.exp(2))
```

```
3*exp(2) + 3*2*exp(2) == 66.50150489037586
```

수치미분-다변수

- 입력 변수가 하나 이상인 다 변수 함수의 경우 입력 변수는 서로 독립적이기 때문에 수치미분 또한 변수의 개수 만큼 개별적으로 계산하여야 함

[예] $f(x, y) = 2x + 3xy + y^3$ 라면, 입력 변수 x, y 두 개 이므로 $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}$ 각각 수치미분 수행.

$f'(1.0,\ 2.0)$ 값을 계산하기 위해서는,

$f(x, 2) = 2x + 6x + 8$

⇒ x = 1.0 에서의 미분계수는 변수 y = 2.0 을 상수로 대입하여 $\frac{\partial f(x,\ 2)}{\partial x}$ 를 수행

$f(1, y) = 2 + 3y + y^3$

⇒ y = 2.0 에서의 미분계수 또한 변수 x = 1.0 인 상수로 대입하여 $\frac{\partial f(1, y)}{\partial y}$ 를 수행

[insight] $f(x, y) = 2x + 3xy + y^3$,인 경우 $f'(1.0,\ 2.0)$ = $(8.0,\ 15.0)$ 직관적 이해

⇒ x = 1.0 에서 미분 값을 구한다는 것은, y 값은 2.0 으로 고정한 상태에서, x = 1.0 을 미세하게 변화시킬 때 f(x, y) 는 얼마나 변화는지 알아보겠다는 의미. 즉, y = 2.0 으로 고정된 상태에서 x = 1.0 을 미세하게 변화시키면 f(x, y) 는 8.0 만큼 변한다는 의미

⇒ y = 2.0 에서 미분 값을 구한다는 것은, x 값은 1.0 으로 고정한 상태에서, y = 2.0 을 미세하게 변화시킬 때 f(x, y) 는 얼마나 변화는지 알아보겠다는 의미. 즉, x = 1.0 으로 고정된 상태에서 y = 2.0 을 미세하게 변화시키면 f(x, y) 는 15.0 만큼 변한다는 의미

수치미분-다변수

다변수 함수

모든변수를 포함하고 있는 numpy객체(배열, 행렬…)

```
def numerical_derivative(f, x):    # 수치미분 debug version
    delta_x = 1e-4
    grad = np.zeros_like(x)

    it = np.nditer(x, flags=['multi_index'], op_flags=['readwrite'])

    while not it.finished:
        idx = it.multi_index

        tmp_val = x[idx]
        x[idx] = float(tmp_val) + delta_x
        fx1 = f(x)   # f(x+delta_x)

        x[idx] = tmp_val - delta_x
        fx2 = f(x)   # f(x-delta_x)
        grad[idx] = (fx1 - fx2) / (2*delta_x)

        x[idx] = tmp_val
        it.iternext()

    return grad
```

계산된 수치미분 값 저장 변수

모든 입력변수에 대해 편미분하기 위해 iterator 획득

Numpy 타입은 mutable이므로 원래 값 보관

하나의 변수에 대한 수치미분 계산