Week 2. Basic MathMatics
(Differential Method & Optimization)

Kisoo Kim

Kwangwoon University, Information Convergence

Kisooofficial@naver.com

목차

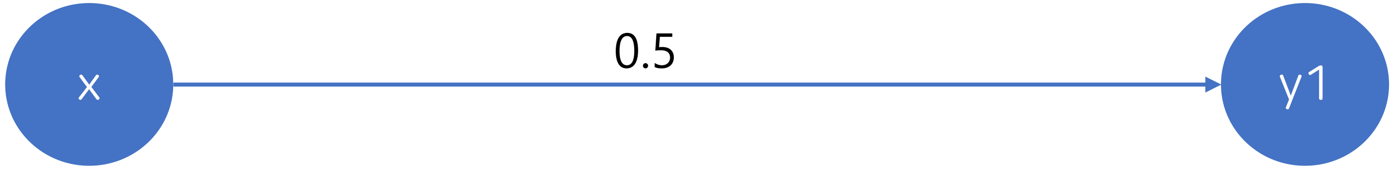
- 여러 가지 함수
  - 일변수-스칼라, 일변수-벡터, 다변수-스칼라, 다변수-벡터 함수, 합성함수
  - 각 함수에 대한 Neural Network 표현
- 미분법
  - 미분계수, 접선의 방정식
  - 일변수 함수의 미분법
  - 다변수 함수의 미분법
- 연쇄 법칙(Chain Rule)
- 최적화
  - Gradient Descent

여러 가지 함수

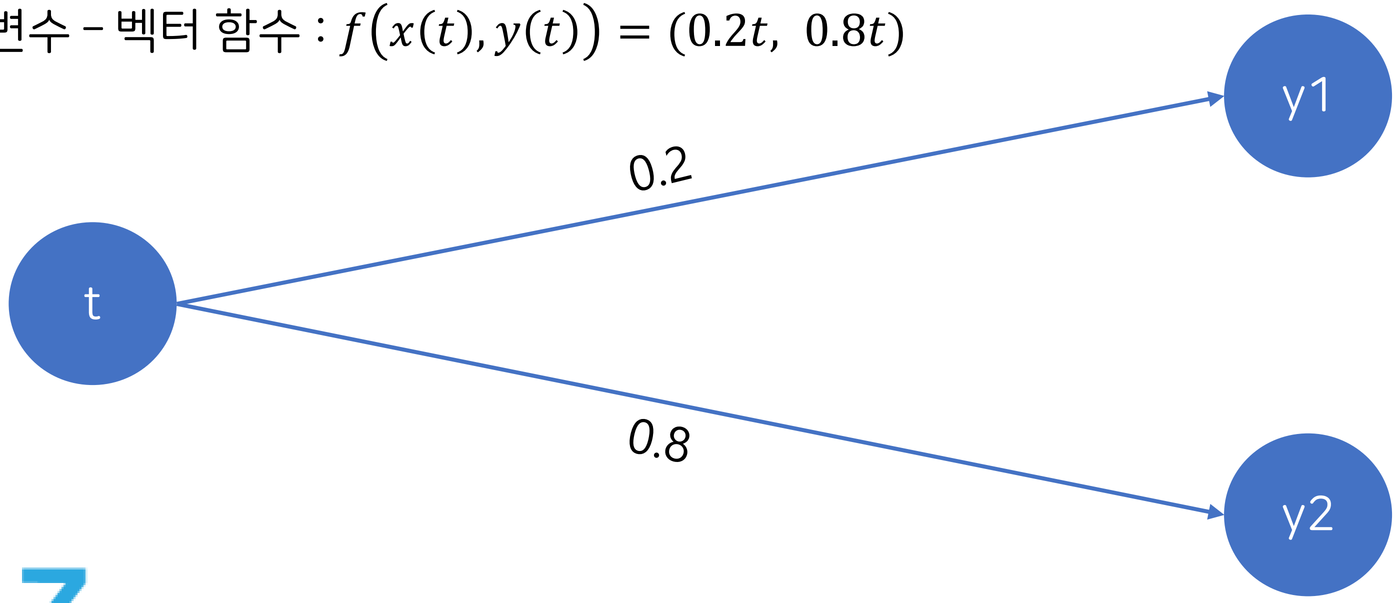
- 입력과 출력의 형태에 따라서 4가지로 구분 가능
  - 일변수 – 스칼라 함수 : $f(x) = 2x + 1$
    - 지하철을 타고 다니는 사람 중, BOAZ 시각화 세션에 참여한 횟수를 바탕으로 교통비를 예측하는 문제
    - 고등학교 과정에서 배웠던 모든 다항함수들이 해당
  - 일변수 – 벡터 함수 : $f(x(t),\ y(t),\ z(t)) = (0.5t, 0.3t, 0.2t)$
    - BOAZ 20기 전체 사람 수를 통해 분석, 시각화, 엔지니어링 세션에 참여하는 사람의 수를 예측하는 문제
  - 다변수 – 스칼라 함수 : $f(x,\ y,\ z,\ w) = x + 2y + 3z + w$
    - BOAZ 20기 사람들의 뒷풀이 참여 횟수, 출석 횟수, 결석 횟수, 과제 완료 횟수를 통해 활동 점수를 예측하는 문제
  - 다변수 – 벡터 함수 : $f(x,\ y,\ z,\ w) = (10x + 5y - 0.5z - w,\ 0.01x + 0.2y + 0.1z + 0.3w)$
    - BOAZ 20기 사람들의 스터디 참여 횟수, 소모임 참여 수, 외부활동 참여 수, 공모전 입상 횟수를 통해 세션 출석 및 결석 횟수 예측 문제

여러 가지 함수

- 각 함수에 따른 Neural Network 표현
  - 일변수 – 스칼라 함수 : $f(x) = 0.5x$

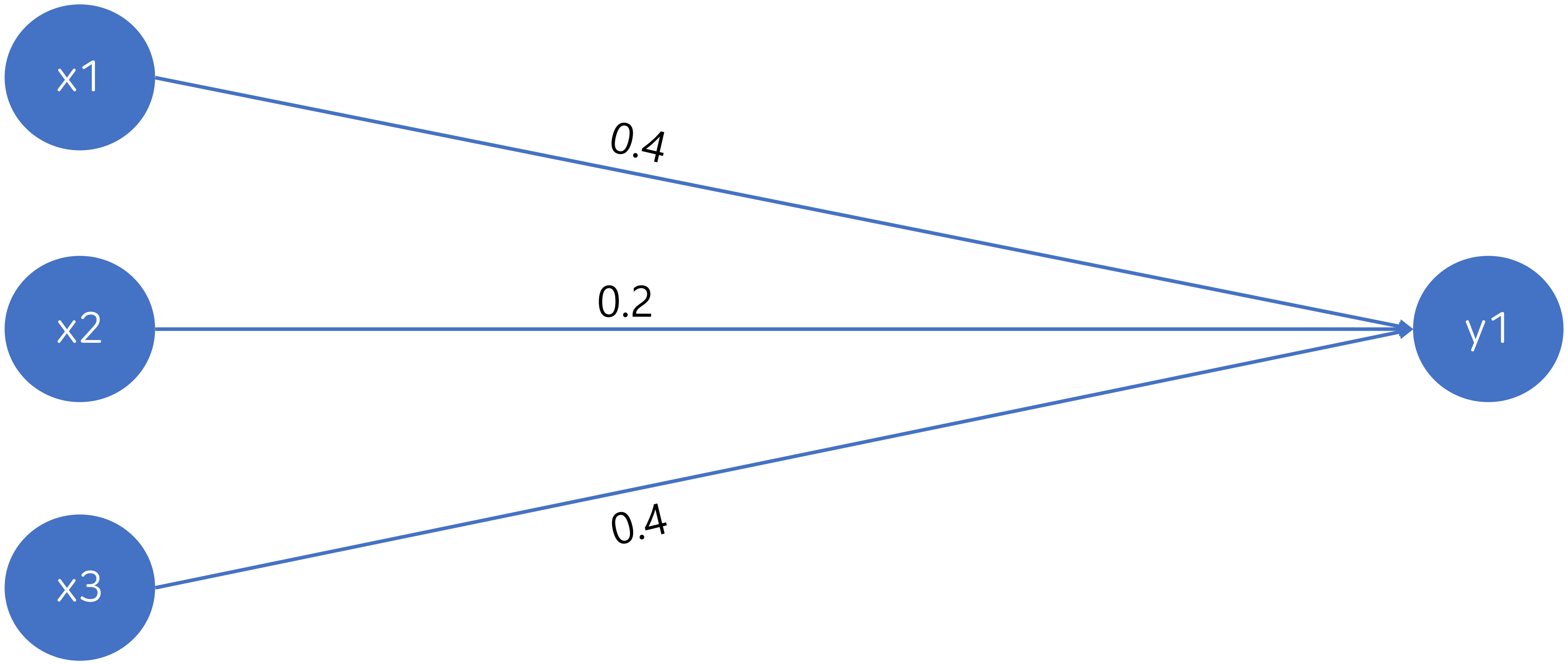


  - 일변수 – 벡터 함수 : $f\big(x(t), y(t)\big) = (0.2t,\ 0.8t)$



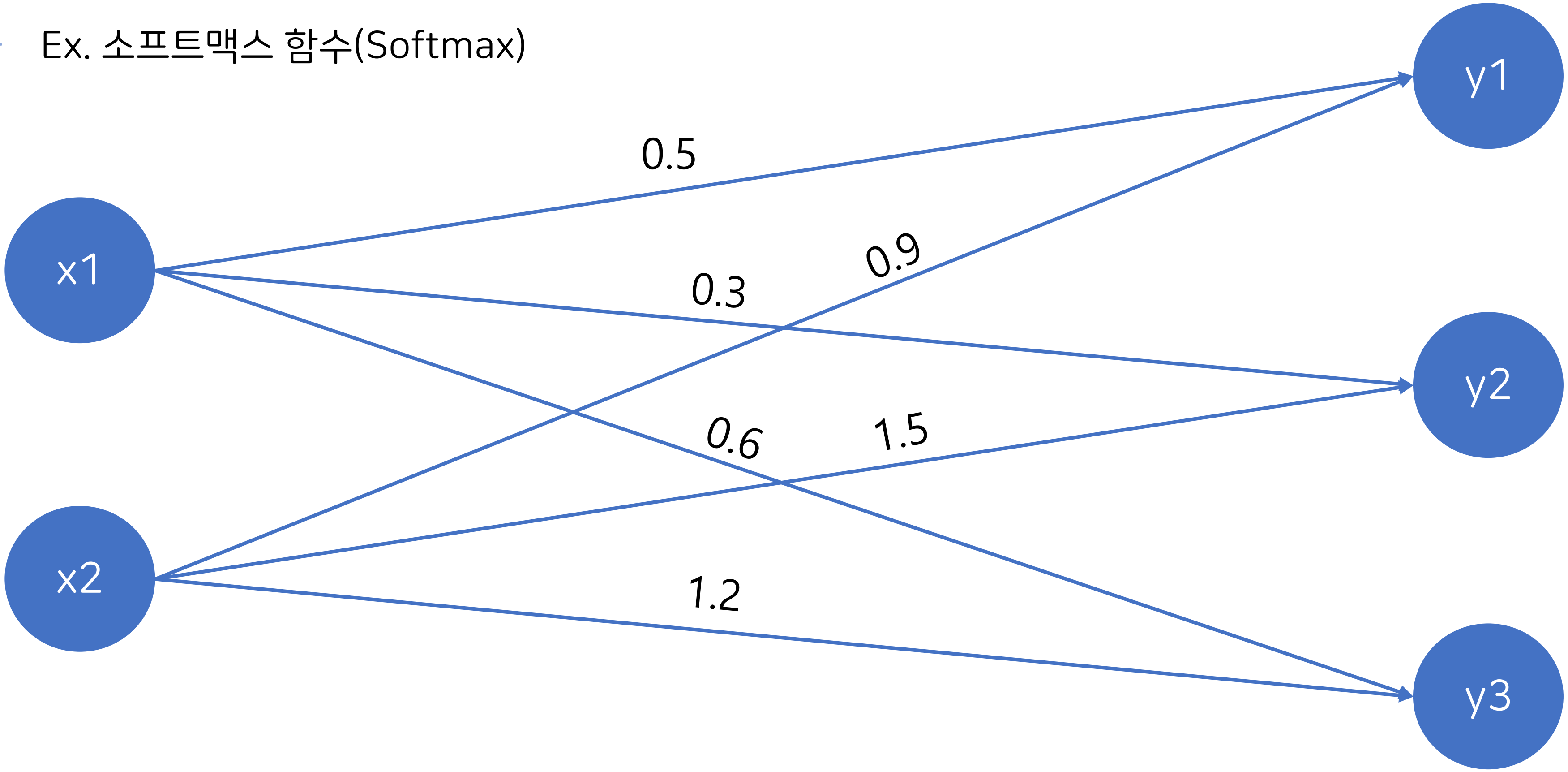
여러 가지 함수

- 각 함수에 따른 Neural Network 표현
  - 다변수 - 스칼라 함수 : $f(x_1, x_2, x_3) = 0.4x_1 + 0.2x_2 + 0.4x_3$



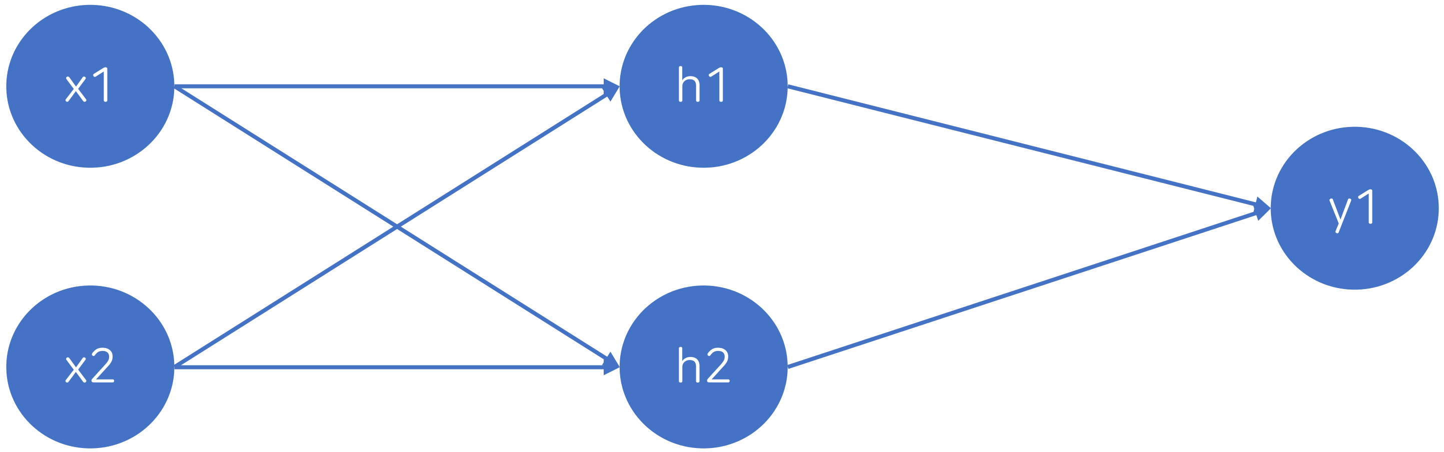
여러 가지 함수

- 각 함수에 따른 Neural Network 표현
  - 다변수 – 벡터 함수 : $f(x_1, x_2) = (0.5x_1 + 0.9x_2,\ \ 0.3x_1 + 1.5x_2,\ \ 0.6x_1 + 1.2x_2)$
    - Ex. 소프트맥스 함수(Softmax)



여러 가지 함수

- 합성함수
  - 어떤 함수의 출력이 또 다시 다른 함수의 입력으로 활용
  - Neural Network에서 hidden layer의 출력값이 또 다시 입력값으로 사용되는 것에 비유할 수 있음
    - $f(x) = x^3 - 12x + 1,\ \ g(x) = \log_3 x\,, g\big(f(x)\big) = g(x^3 - 12x + 1) = \log_3(x^3 - 12x + 1)$
  - 합성함수와 Neural Network의 예시
    - $f(x_1, x_2) = (h_1, h_2),\ \ g(h_1, h_2) = y$라고 하면, $g\big(f(x_1, x_2)\big) = y$의 형태로 나타낼 수 있음.



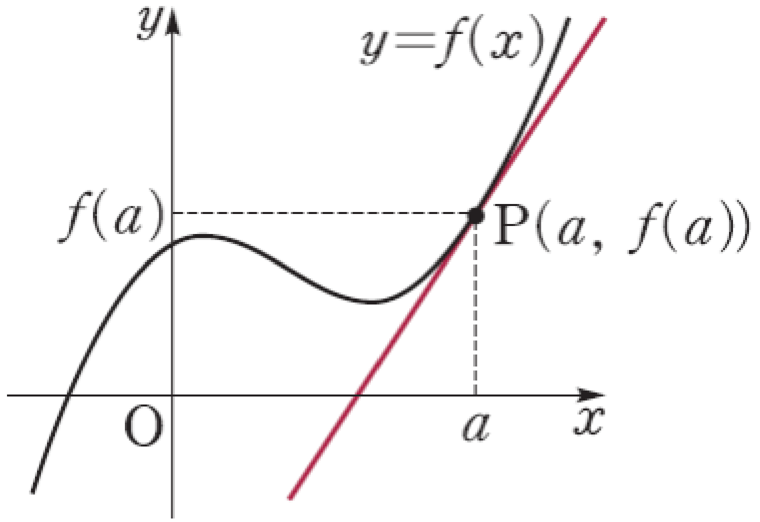
미분법

- 미분
  - 함수 f(x)의 도함수를 구하는 것을 f(x)를 미분한다고 함
  - $f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$
    - 여러 가지 표기법 존재 $f'(x),\ y', \frac{dy}{dx}, \ \frac{d}{dx}f(x),\ Dy$
    - 미분한 것을 또 미분하는 것을 이계도함수, 그 이후로부터 n계도함수라는 명칭을 사용함
      - ✓ N계 도함수의 표기법 : $y^{(n)},\ f^{(n)}(x),\ \frac{d^n y}{dx^n}$
  - $f'(a)$를 $x = a$에서의 미분계수라고 하고, $f'(a)$의 값은 x = a에서의 접선의 기울기와 같음.
    - Neural Network에서 $f'(a)$를 그레디언트(gradient)라고 말함. (수학적으로는 미분계수)
  - 데이터 분석에서 Loss function의 최솟값을 구하려고 할 때, 매우 많이 사용함

BOAZ

미분법

- 접선의 방정식
  - $y = f(x)$위의 점 $(a, f(a))$에서의 접선의 기울기인 $f'(a)$를 구한다.
  - $y - f(a) = f'(a)(x - a)$를 이용하여 접선의 방정식을 구한다.



미분법

- 여러 가지 함수의 미분법
  - 다항함수
    - $y = c$ (c는 상수)이면 $y' = 0$
    - $y = x^n$이면 $y' = nx^{n-1}$ (n은 자연수)
    - $y = cf(x)$이면 $y' = cf(x)$ (c는 상수)
    - $y = f(x) \pm g(x)$이면 $y' = f'(x) \pm g'(x)$
    - $y = f(x)g(x)$이면 $y' = f'(x)g(x) + f(x)g'(x)$
    - $y = \frac{f(x)}{g(x)}$이면 $y' = \frac{f'(x)g(x) - f(x)g'(x)}{\{g(x)\}^2}$

미분법

- 여러 가지 함수의 미분법
  - 삼각함수
    - $y = \sin x$이면 $y' = \cos x$
    - $y = \cos x$ 이면 $y' = \sin x$
    - $y = \tan x$ 이면 $y' = sec^2 x$
    - $y = \csc x$ 이면 $y' = -\csc x \cot x$
    - $y = \sec x$ 이면 $y' = \sec x \tan x$
    - $y = \cot x$ 이면 $y' = -csc^2\, x$

미분법

- 여러 가지 함수의 미분법
  - 지수, 로그함수
    - $y = a^x$이면 $y' = a^x \ln a$
    - $y = e^x$이면 $y' = e^x$
    - $y = \log_a x$ 이면 $y' = \frac{1}{x} \times \frac{1}{\ln a}$
    - $y = \ln x$ 이면 $y' = \frac{1}{x}$
    - $y = \sec x$ 이면 $y' = \sec x \tan x$
    - $y = \cot x$ 이면 $y' = -csc^2\ x$

미분법

- 여러 가지 함수의 미분법
  - 합성함수의 미분법
    - $z = f\left(g(x)\right)$라하고, $g(x) = y$라고 하면 $z = f(y)$이다. z를 미분한 결과는 $z' = f'\left(g(x)\right)g'(x)$ => $\frac{dz}{dy} \times \frac{dy}{dx}$
    - 각 함수의 미분계수의 곱의 형태라고 생각할 수 있음
    - 신경망 역전파 알고리즘의 기본적인 아이디어
  - 매우 많은 함수가 겹겹이 쌓여있다면?
    - $z = f(g(h(i(j(k(x))))))$와 같은 함수라면?
    - $\frac{dz}{dx} = \frac{dz}{df}\frac{df}{dg}\frac{dg}{dh}\frac{dh}{di}\frac{di}{dj}\frac{dj}{dk}\frac{dk}{dx}$

미분법

- $y = (3x^2 + 2x)^3$, $\mathrm{y} = \sin(\cos x)$, $y = \log_3(x^2 + 2x)$, $y = x^2 e^x$의 도함수를 구하시오.

다변수 함수의 미분법

- 편도함수
  - $y = f(x_1, x_2, \cdots, x_k, \cdots, x_n)$의 다변수 함수일 때, 하나의 변수에 대해서만 미분한 도함수
    - 표기 : $\frac{\partial f}{\partial x_k}(x_1, x_2, \cdots, x_k, \cdots, x_n) = \lim_{h \to 0} \frac{f(x_1, x_2, \cdots, x_k + h, \cdots, x_n) - f(x_1, x_2, \cdots, x_n)}{h}$
    - 다시 말해, x_k를 제외한 나머지 변수들은 상수로 생각하고 미분함

다변수 함수의 미분법

- $z(x, y) = x^2 + 4xy + 2y - 3$에 대해 $\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}$를 구하고, (4, -1)에서의 편미분계수를 구하시오.

연쇄 법칙(Chain Rule)

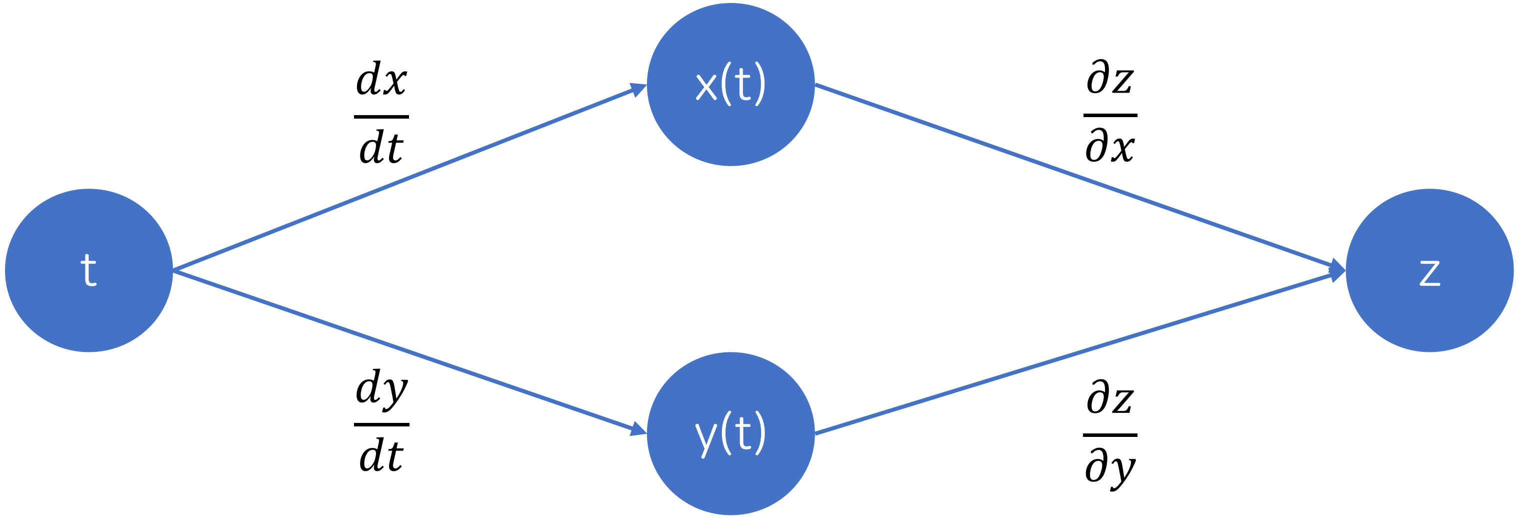
- 연쇄 법칙(Chain Rule)
  - $z = g(y),\ y = f(x)$인 $z = g(f(x))$에서 $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$
    - X가 입력 변수, y가 중간 변수, z가 출력 변수

입력변수 중간변수 출력변수

x $\quad \frac{dy}{dx} \quad$ $f(x) = y$ $\quad \frac{dz}{dy} \quad$ $g(y) = z$

연쇄 법칙(Chain Rule)

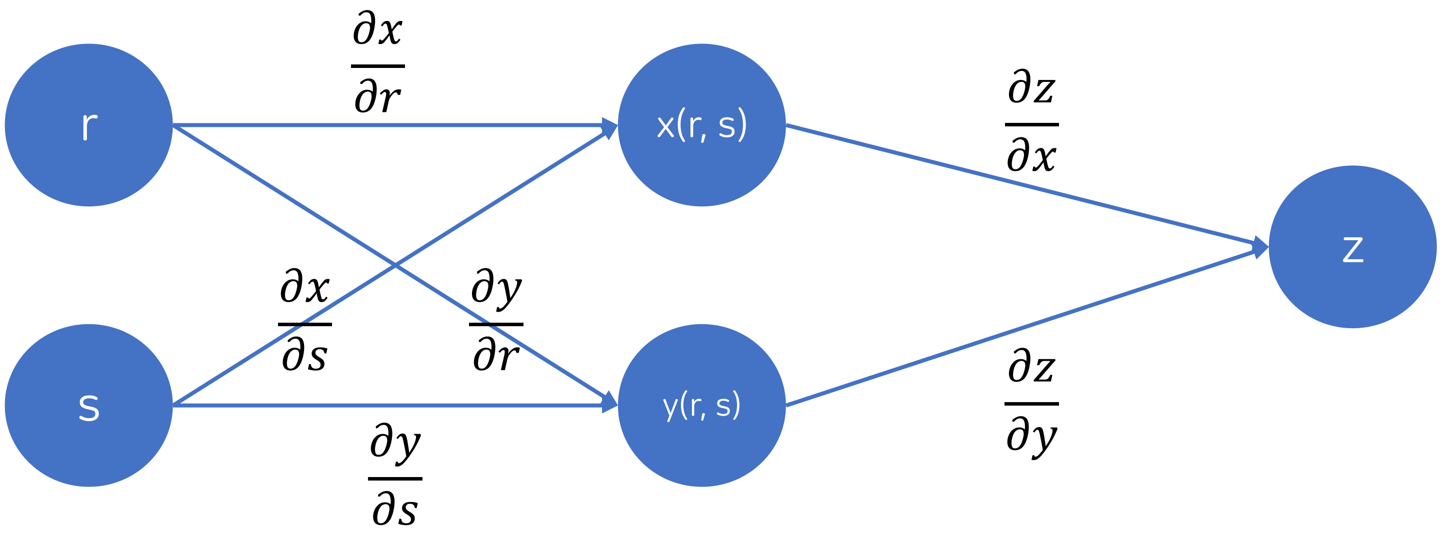
- 연쇄 법칙(Chain Rule)
  - 입력, 중간, 출력 변수가 1개 이상인 경우
    - $x = f_1(t),\ y = f_2(x)$일 때, $z = f(x, y)$의 t에 대한 변화
  - Neural Network처럼 그리기 : 쉽게 미분하기 위해



$$\frac{dz}{dt} = \frac{\partial z}{\partial x}\frac{dx}{dt} + \frac{\partial z}{\partial y}\frac{dy}{dt}$$

연쇄 법칙(Chain Rule)

- 연쇄 법칙(Chain Rule)
  - 입력, 중간, 출력 변수가 1개 이상인 경우
    - $x = f_1(r, s),\ y = f_2(r, s)$일 때, $z = f(x, y)$의 r, s에 대한 변화
  - Neural Network의 형태로 그리기



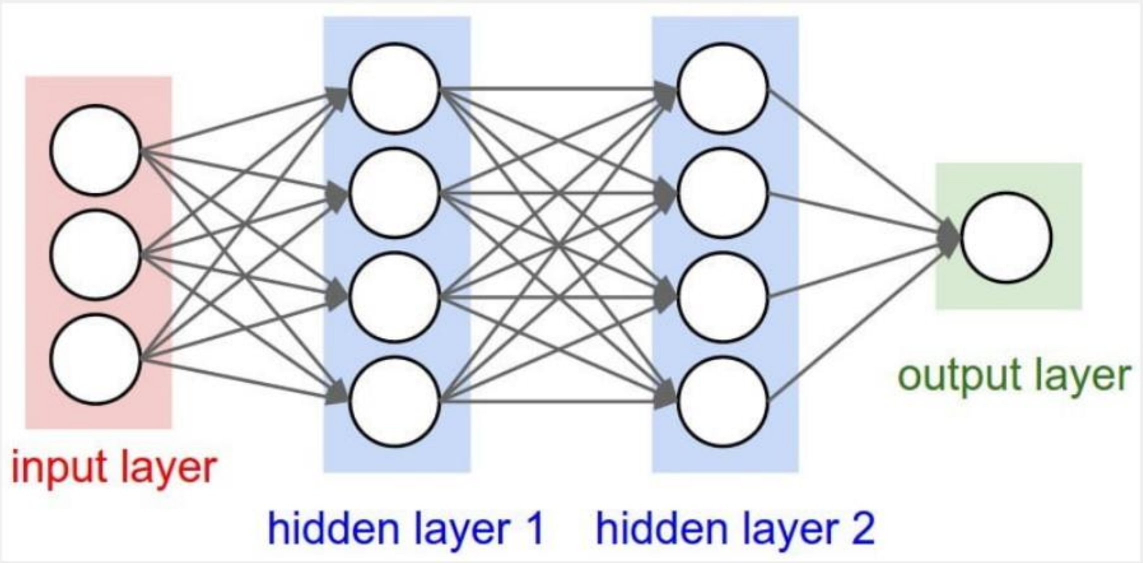
$$\frac{\partial z}{\partial r} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial r}, \qquad \frac{\partial z}{\partial s} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial s}$$

연쇄 법칙(Chain Rule)

- $z(x,y) = x^2y,\ x = u^2 + v^2,\ \ y = 2uv$일 때, $\frac{\partial z}{\partial u}$를 구하시오.

연쇄 법칙(Chain Rule)

- 연쇄 법칙(Chain Rule)이 중요한 이유
  - 아래와 같은 신경망 가정
    - 수치 미분은 시간이 너무 오래 걸린다는 단점이 존재. 역전파 알고리즘 진행 시, 연쇄 법칙을 이용함



Two or more hidden layers comprise a Deep Neural Network

최적화(Optimization)

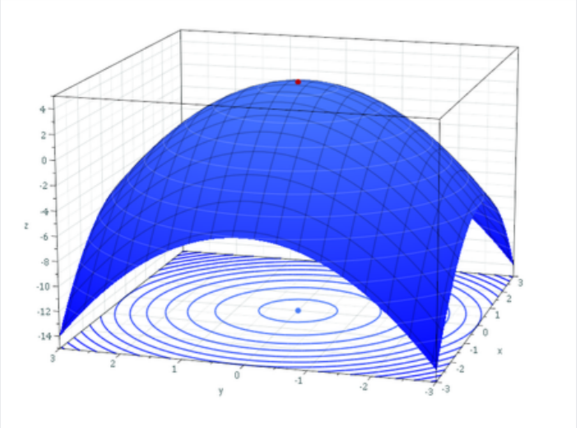
수학적 최적화

53개 언어

문서 토론 읽기 편집 역사 보기

위키백과, 우리 모두의 백과사전.

최적화(最適化, 영어: mathematical optimization 또는 mathematical programming)는 특정의 집합 위에서 정의된 실수값, 함수, 정수에 대해 그 값이 최대나 최소가 되는 상태를 해석하는 문제이다. **수리 계획** 또는 **수리 계획 문제**라고도 한다. 물리학이나 컴퓨터에서의 최적화 문제는 생각하고 있는 함수를 모델로 한 시스템의 에너지를 나타낸 것으로 여김으로써 **에너지 최소화 문제**라고도 부른다.



포물면 $f(x,y) = -(x^2+y^2)+4$. 붉은 점 $(0,0,4)$에서의 최댓값을 갖는다.

최적화 문제 [편집]

이 부분의 본문은 최적화 문제입니다.

최적화 문제는 다음과 같은 방법으로 표현한다.

수식: 함수 $f : A \to \mathbf{R}$ 어떤 집합 A에서 실수 x 에 대해

의미: an element x_0 in A such that $f(x_0) \leq f(x)$ for all x in A ("minimization") or such that $f(x_0) \geq f(x)$ for all x in A ("maximization").

위와 같은 공식은 선형 계획법 (linear programming)이라 한다. 실생활 및 이론적 문제 모두가 이와 같은 보편적 방법으로 해결할 수 있다.

함수 f의 값이 최소이거나 최대인 값을 찾으면 최적화 해법(optimal solution)을 찾은 것이 된다.[1] 최적화 문제의 종류에 따라서 최적해를 찾기 위한 방법은 최소화(minimization) 혹은 최대화(maximization)로 나눌 수 있다.

최적화(Optimization)

- 여러 가지 최적화 문제
  - 선형/비선형 계획법
  - 최댓값/최솟값 찾는 문제 등
- 딥러닝을 하기 위해 필요한 최적화
  - 경사 하강법(Gradient Descent)

여러 가지 최적화 문제

- 선형계획법(Linear Programming)
  - 목적함수와 제약조건이 모두 선형으로 이루어져 있어야 함.
  - Ex. $f(x_1, x_2) = 3x_1 + 2x_2 \ \ (x_1 + x_2 \leq 80)$
  - 현실 속의 문제가 대부분 비선형계획법과 관련이 있다는 점에서 현실과 조금은 동 떨어지는 느낌이 있음

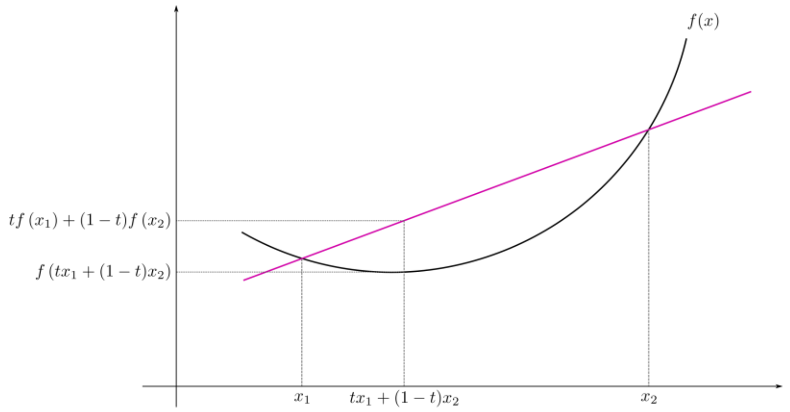
- 볼록 최적화
  - 수학적 최적화의 한 방법으로 많은 문제에서 효과적임
  - 모든 국소해(local solution)는 전역해(global solution)

여러 가지 최적화 문제

- 볼록 최적화
  - 볼록 함수(Convex function)

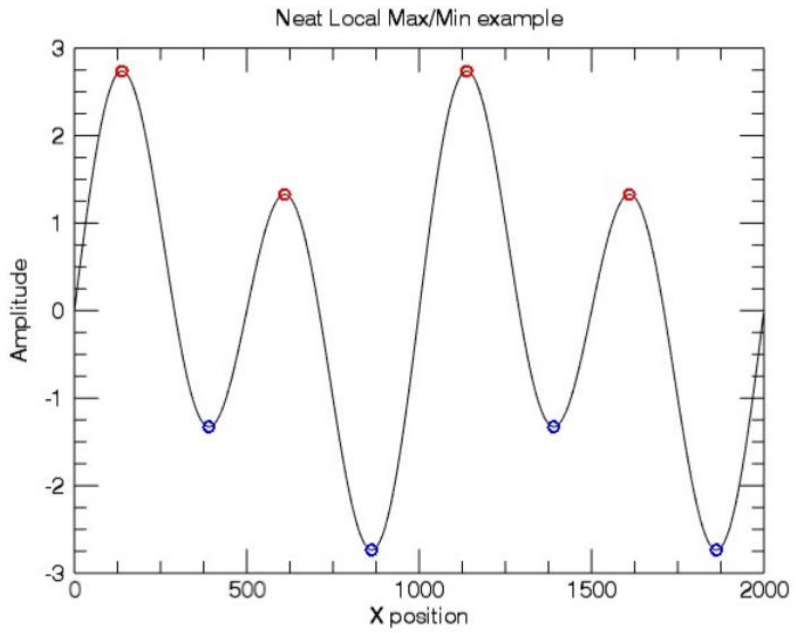
$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y)$$
$$for\ all\ x, y \in \mathbb{R}^n\ and\ 0 \le \theta \le 1$$

  - 볼록 함수(Convex function)의 예
    - 가장 대표적인 것이 이차함수, 이차함수는 최고차항의 계수가 양수이면 아래로 볼록, 음수이면 위로 볼록한(concave) 그래프
    - 기하학적인 의미 : 임의로 두 점을 잡아 이은 선분이 곡선보다 위에 있어야 함



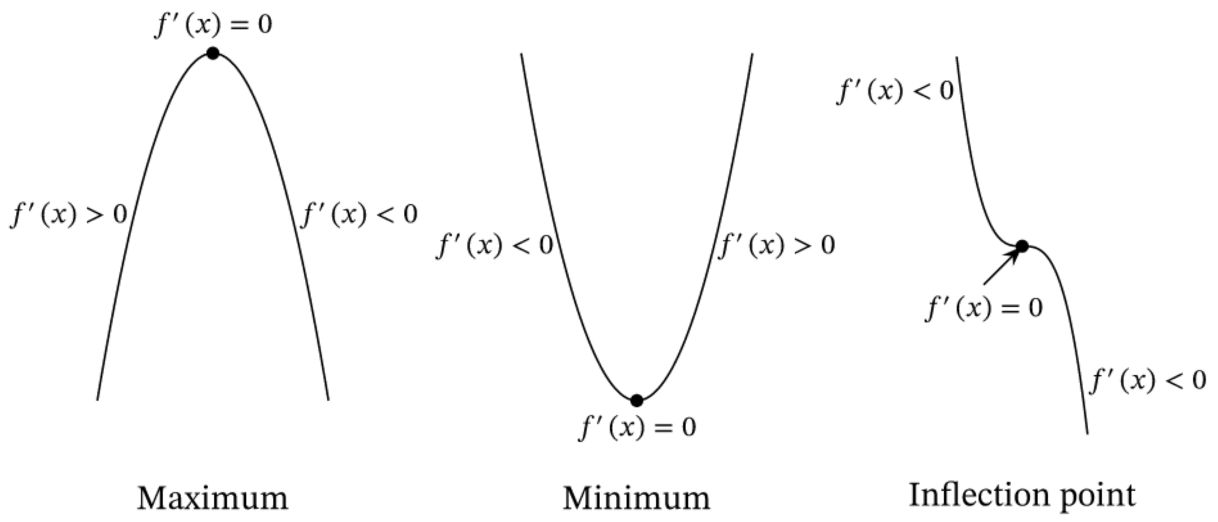
여러 가지 최적화 문제

- 국소해(local minimum)와 전역해(global minimum)
  - 국소해 : 특정 지점 x_a 근방 $B(x_a, \epsilon)$에 포함된 모든 지점 $(y \in B)$에 대해서 $f(x_a) \leq f(y)$이면, x_a는 국소해
  - 전역해 : 모든 지점 $y \in X$에 대해, $f(x_a) \leq f(y)$ 이면 x_a는 전역해



여러 가지 최적화 문제

- 국소해(local minimum)과 전역해(global minimum)을 찾는 방법
  - 도함수 $f'(x) = 0$을 만족하는 극점(stationary point)을 파악
  - 특정한 극점(stationary point)에서 아래를 만족한다.
    - $f''(x_a) > 0 \;\rightarrow local\ minimum$
    - $f''(x_a) < 0 \;\rightarrow local\ maximum$

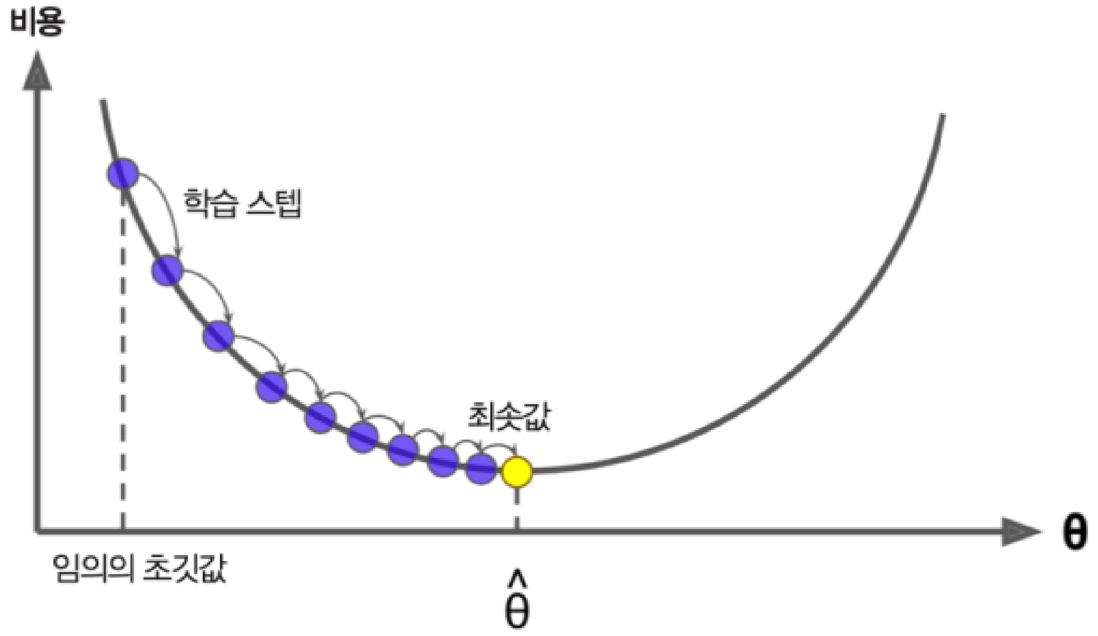


여러 가지 최적화 문제

- $f(x) = x^3 - 3x$의 local minimum이나 local maximum을 찾으시오.

경사 하강법(Gradient Descent)

- 경사 하강법(Gradient Descent)
  - 어떤 완벽한 solution을 주지는 않음.
  - Analytical solution이 아닌 iterative한 방법을 활용
  - Negative gradient 방향으로 이동시켜서 gradient가 0에 가까워지는 지점을 찾음
- 아래 그래프는 데이터 분석에서 Loss Function이라고 생각할 수 있음

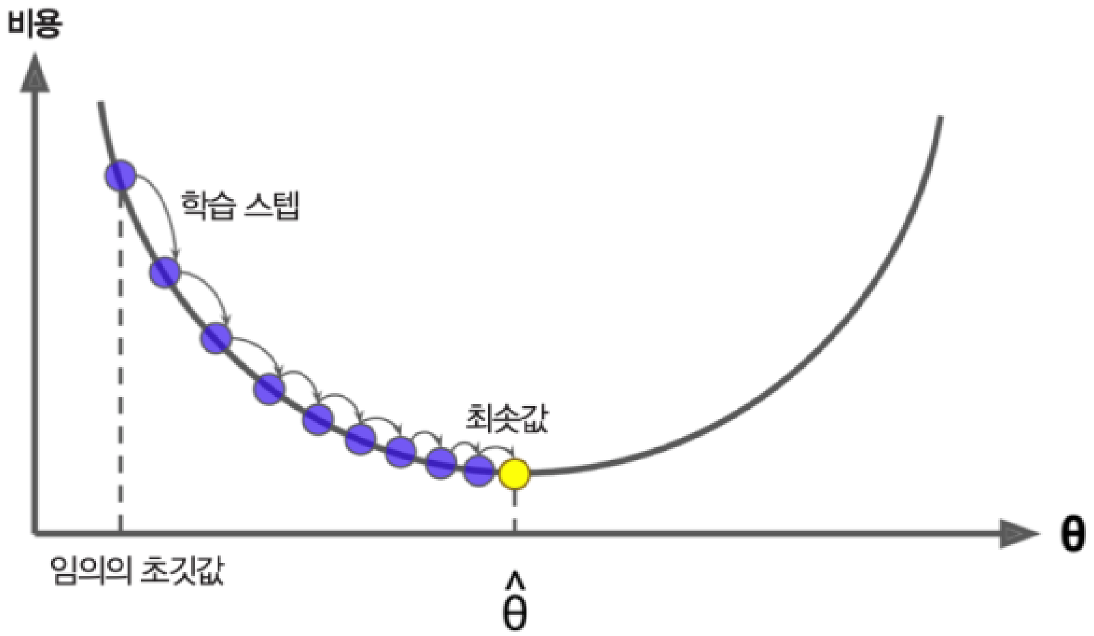


경사 하강법(Gradient Descent)

- 경사 하강법(Gradient Descent)
  - 어떤 완벽한 solution을 주지는 않음.
  - Analytical solution이 아닌 iterative한 방법을 활용
  - Negative gradient 방향으로 이동시켜서 gradient가 0에 가까워지는 지점을 찾음

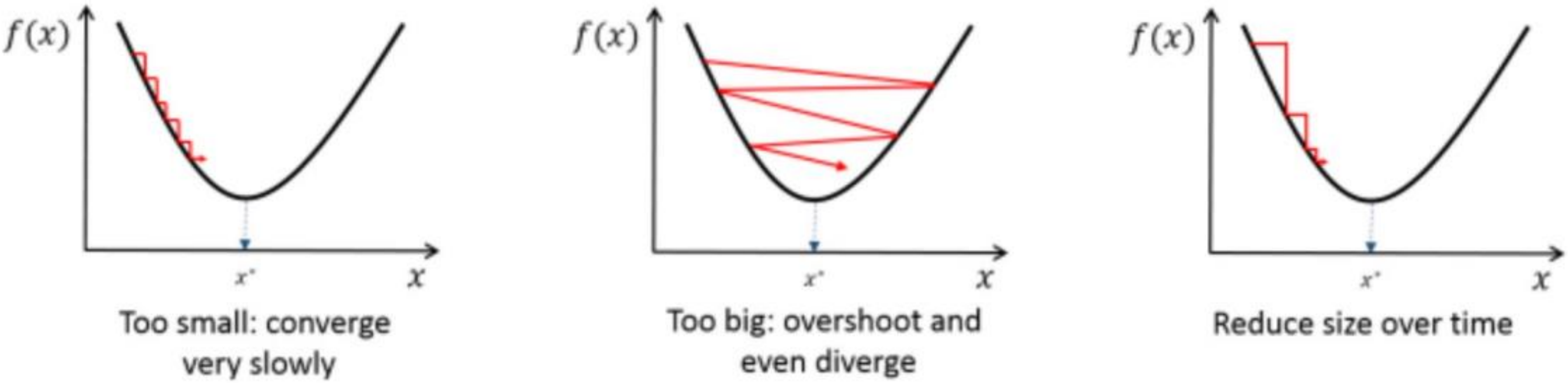
$$x \leftarrow x - \alpha \nabla_x f(x)$$

$$\alpha = learning\ rate, \qquad \nabla_x f(x) : gradient\ from\ loss\ function(MSE)$$



경사 하강법(Gradient Descent)

- 경사 하강법(Gradient Descent)
  - Learning rate를 어떻게 설정하냐에 따라, 최적의 minimum을 찾을 수 있음



경사 하강법(Gradient Descent)

- 방법
  1. 학습해야 하는 모든 파라미터를 초기화한다.
  2. 각 파라미터에 대한 Loss Function의 gradient를 구한다.
  3. Learning rate를 이용하여 gradient와 함께 파라미터를 업데이트한다.
  4. 단계 2, 3 반복하기

여러 가지 Loss Function

- MAE(Mean Absolute Error)
  - $MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$
- MSE(Mean Squared Error)
  - $MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$
- RMSE(Root Mean Squared Error)
  - $MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$
- Cross-Entropy
  - Binary Cross-Entropy(BCE-Loss), Categorical Cross-Entropy

Classfication Loss Functions

Binary Cross Entropy

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

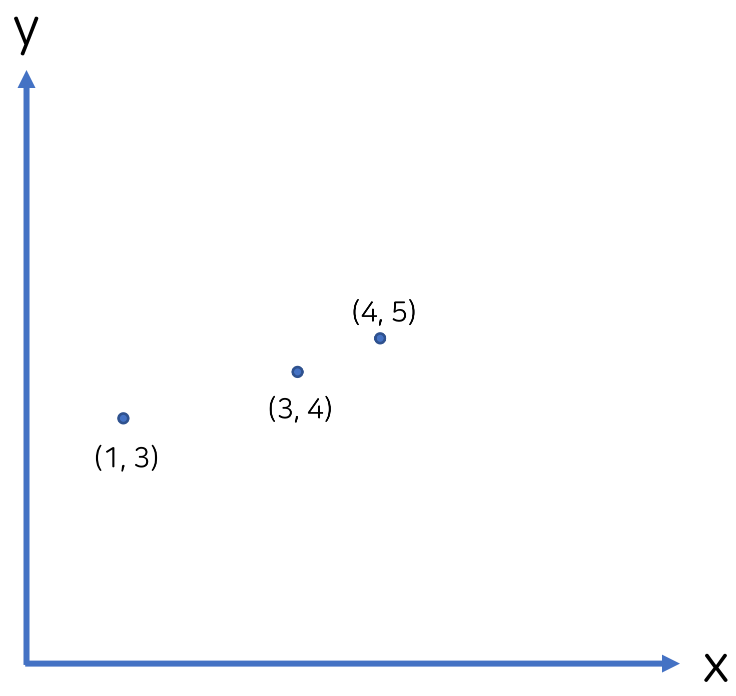
Binary Cross Entropy (BCE) Loss Function

경사 하강법(Gradient Descent)

- 아래 그림과 같이 좌표평면 위에 (1, 3), (3, 4), (4, 5)가 있다. 이 세 점을 직선 y = ax+b를 이용하여 예측한다.

Gradient descent를 이용하여 두 파라미터 a, b를 정하려고 할 때, 첫 번째 에폭을 지난 후 a, b의 값을 구하시오.

(단, learning_rate = 0.01이고, a와 b의 초기값은 각각 1과 0이며, Loss Function은 MSE를 사용)



Next Lecture

- 지도학습 기반의 머신러닝 알고리즘