

누구나 할 수 있는!

# 인터페이스 AI 오픈 스터디

1주차. 인공지능을 위한 수학

# 수학과 영어의 필요성

- AI의 대부분들이 제시한 가치에 따라, 대부분의 인공지능 관련 논문이 arXiv를 통해 빠르게 공유되고, 기술이 매우 빠르게 발전함.
- 문제는 이 때문에 AI 관련 논문과 용어는 99%가 영어
  - AI 논문은 영어로 쓰는 게 국룰
- 또한, 수학은 공학자들 사이에서는, 기술이 아니라 하나의 언어다.

$$\mathcal{L}_{adv} = \mathbb{E}_x [\log D_{src}(x)] + \mathbb{E}_{x,c} [\log (1 - D_{src}(G(x, c)))],$$

The screenshot shows the arXiv page for the paper "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation" by Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. The page includes the Cornell University logo, a search bar, and a sidebar with download options (PDF, Other formats), current browse context (cs.CV), references and citations (NASAADS, Google Scholar, Semantic Scholar), 2 blog links, and a bibliography section. The main content area displays the title, authors, abstract, comments, subjects, journal reference, and submission history.

Computer Science > Computer Vision and Pattern Recognition

[Submitted on 24 Nov 2017 (v1), last revised 21 Sep 2018 (this version, v3)]

**StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation**

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo

Recent studies have shown remarkable success in image-to-image translation for two domains. However, existing approaches have limited scalability and robustness in handling more than two domains, since different models should be built independently for every pair of image domains. To address this limitation, we propose StarGAN, a novel and scalable approach that can perform image-to-image translations for multiple domains using only a single model. Such a unified model architecture of StarGAN allows simultaneous training of multiple datasets with different domains within a single network. This leads to StarGAN's superior quality of translated images compared to existing models as well as the novel capability of flexibly translating an input image to any desired target domain. We empirically demonstrate the effectiveness of our approach on a facial attribute transfer and a facial expression synthesis tasks.

Comments: Accepted to CVPR 2018 (Oral)

Subjects: Computer Vision and Pattern Recognition (cs.CV)

Journal reference: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6789-6797

Cite as: arXiv:1711.09020 [cs.CV]  
(or arXiv:1711.09020v3 [cs.CV] for this version)

**Bibliographic data**  
[Enable Bibex (What is Bibex?)]

**Submission history**  
From: Yunjey Choi [view email]  
[v1] Fri, 24 Nov 2017 16:37:30 UTC (5,462 KB)  
[v2] Fri, 14 Sep 2018 06:09:36 UTC (6,547 KB)  
[v3] Fri, 21 Sep 2018 08:17:49 UTC (6,547 KB)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

arXiv is a registered trademark of Cornell University. | Browse v3.0 released 2020-04-15 | Feedback?

About arXiv | Leadership Team | Contact | Follow us on Twitter | Help | Privacy Policy | Blog | Subscribe

If you have a disability and are having trouble accessing information on this website or need materials in an alternate format, contact web-accessibility@cornell.edu for assistance.

# 인공지능을 위해 필요한 수학

- 선형대수학

- 컴퓨터로 데이터를 표현하는데 벡터, 행렬 등을 이용
- 신경망 모델 자체를 행렬의 곱으로 나타낼 수도 있음
- 선형 함수들의 성질을 알아야 딥러닝이 가능함

- 멀티미디어

- 수학적 변환을 이용해 우리 생활 속 연속(Analog) 데이터를 컴퓨터가 이해 가능한 이산적(Digital) 데이터로 변환하는 학문

- 미분

- 딥러닝의 학습 부분에 사용
- 우리가 고등학교 때 배운 미분은 어렵지만, 딥러닝에서는 그렇게 어렵진 않음

# 선형대수학 Linear Algebra

# 스칼라와 벡터 (Scalar and Vector)

- 우리가 평소에 흔히 사용하는 하나의 수를 스칼라 값이라 한다.
  - 2, 3, 0, 0.12, 2.1, -5.2 ...
- 스칼라는 하나의 변수에, 하나의 값을 갖는다.
  - 크기, 무게, 질량 등...
- 우리는 때때로, 여러 값을 갖는 데이터를 마주한다.
  - 서울시의 좌표 (위도, 경도), 누군가의 키와 몸무게 (160, 50)
- 이때, 2개 이상의 값을 하나의 변수에 갖는 것들을 벡터라 한다.
  - 세종대 모 동아리 회원들의 학점 목록 (3.5, 4.0, 4.5, 4.2, 3.7, 2.8 ...)
  - 한 벡터안에 들어갈 수 있는 값의 개수에 제한은 없다.

# 행렬 (Matrix)

- 벡터에서 한 단계 확장된 데이터, 행(row)과 열(column)을 갖는 데이터
  - 우리가 흔히 보는 표, 장부 등, 흑백 이미지
  - 역시 행과 열의 개수에 제한은 없음

		열				
행	0	255	0	255	0	
	255	100	255	100	255	
	0	255	100	255	0	
	0	0	255	0	0	
	0	0	0	0	0	

흑백 이미지의 행렬 표현

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

수학적 표현

```
matrix = [[1, 2, 3],  
          [4, 5, 6],  
          [7, 8, 9]]
```

프로그램에서의 표현(2차원 배열)

# 행렬의 곱 (Matrix Multiplication)

- 행렬의 곱셈 연산은 두 종류가 있음.
  - Element-wise Multiplication (of Matrix)
  - Matrix Multiplication
- 이 중, element-wise는 간단하게 행렬 전체에 특정 스칼라 값을 곱하는 것.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$A \times 2 = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

# 행렬의 곱 (Matrix Multiplication)

- 행렬곱, MatMul은 조금 더 복잡함.
  - MatMul은 행렬과 행렬간의 곱셈
  - 곱하고자 하는 행렬 A 의 크기 (MxN)과 행렬 B의 크기 (TxY)에서, N과 T가 같아야 함
  - 두 행렬의 곱 C는 (MxY) 크기의 행렬이 됨

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

2x2 행렬

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

2x3 행렬

$$AB = \begin{bmatrix} 9 & 12 & 15 \\ 19 & 26 & 33 \end{bmatrix}$$



# 행렬의 곱 (Matrix Multiplication)

- 행렬 A와 B의 곱 C의 (1,2)에 있는 원소는, 행렬 A의 1행과 행렬 B의 2열의 곱의 합이다.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$AB = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

# 스칼라, 벡터, 행렬의 비교

- 스칼라와 벡터, 행렬은 차원이 점점 늘어가는 데이터이다.

2
---

Scalar, 0차원

1	2	3
---	---	---

vector, 1차원

1	2	3
4	5	6

Matrix, 2차원

1	2	3
4	5	6

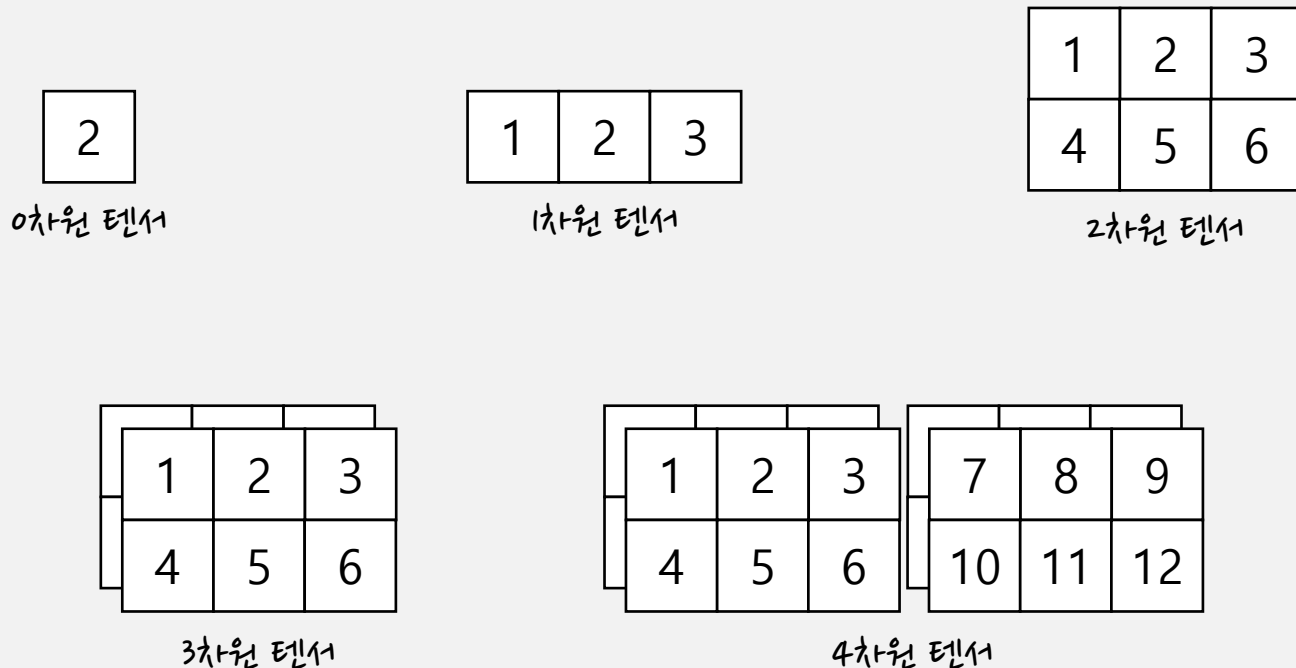
그럼 이 친구는 뭘까?

1	2	3	7	8	9
4	5	6	10	11	12

여는 ?????

# 텐서 (Tensor)

- 보통 행렬 이상으로 많은 차원(3차원 이상)을 갖는 데이터를 텐서라 함
- 그러나 스칼라나 행렬 등도 각각 0차원, 2차원 텐서로 볼 수 있음



# 텐서가 흐른다?

- 구글의 머신러닝 라이브러리 Tensorflow의 이름이 여기서 유래함.
- Tensor로 된 데이터들이 신경망 위를 흘러간다(flow)!



# 멀티미디어 Multimedia

어쩌면 이산수학에 가까울지도..?

# Digital vs Analog

- 이 세상은 연속적인 (Continuous) 데이터로 이루어져 있다.
  - 우리는 시간을 1초 단위로 나누지만, 사실 0.5초, 0.25초, 0.125초... 등 시간은 무한히 쪼갤 수 있다.
  - 빛의 세기를 밝음과 어두움 사이에서도 수많은 값으로 나눌 수 있다.
  - 이러한 현실의 데이터를 아날로그 데이터라 한다.



# Digital vs Analog

- 그러나, 컴퓨터 속에는 0과 1의 이산적인 데이터가 존재한다.  
(0과 1 사이의 값이 없다.)
  - 이러한 단위를 비트(bit)라 하고, 컴퓨터의 저장공간에는 한계가 있다.
  - 그래서 컴퓨터 공학자들은 아날로그 데이터를 이산적인 디지털 데이터로 변환하여 저장한다.
  - 그렇기에, 현실의 데이터를 컴퓨터에 저장할 때는 필연적으로 데이터의 손실(loss)이 생긴다.



# RGB Code

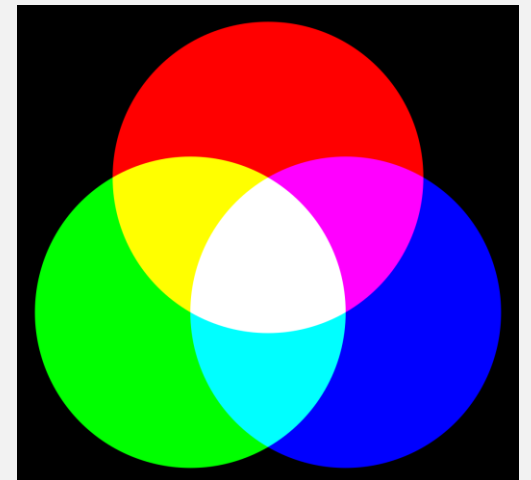
- 이미지를 컴퓨터에 저장하기 위해 빛의 3원색을 이용한 RGB를 이용한다.
  - 빛의 3원색(Red, Green, Blue)를 섞으면 이 세상의 모든 색을 만들 수 있다.
  - 컴퓨터에서는 이 3가지 색깔의 세기 0(없음) ~ 1(가장 밝음)까지를 256개로 나눠 저장한다. 이는 256이 8비트로 나타낼 수 있는 경우의 수이기 때문이다.
  - 각 색상의 값이 0 이면 해당 색상이 포함되지 않은 것이며, 255이면 최대로 포함 된 것이다.

빨간색 (255, 0, 0)

보라색 (255, 0, 255)

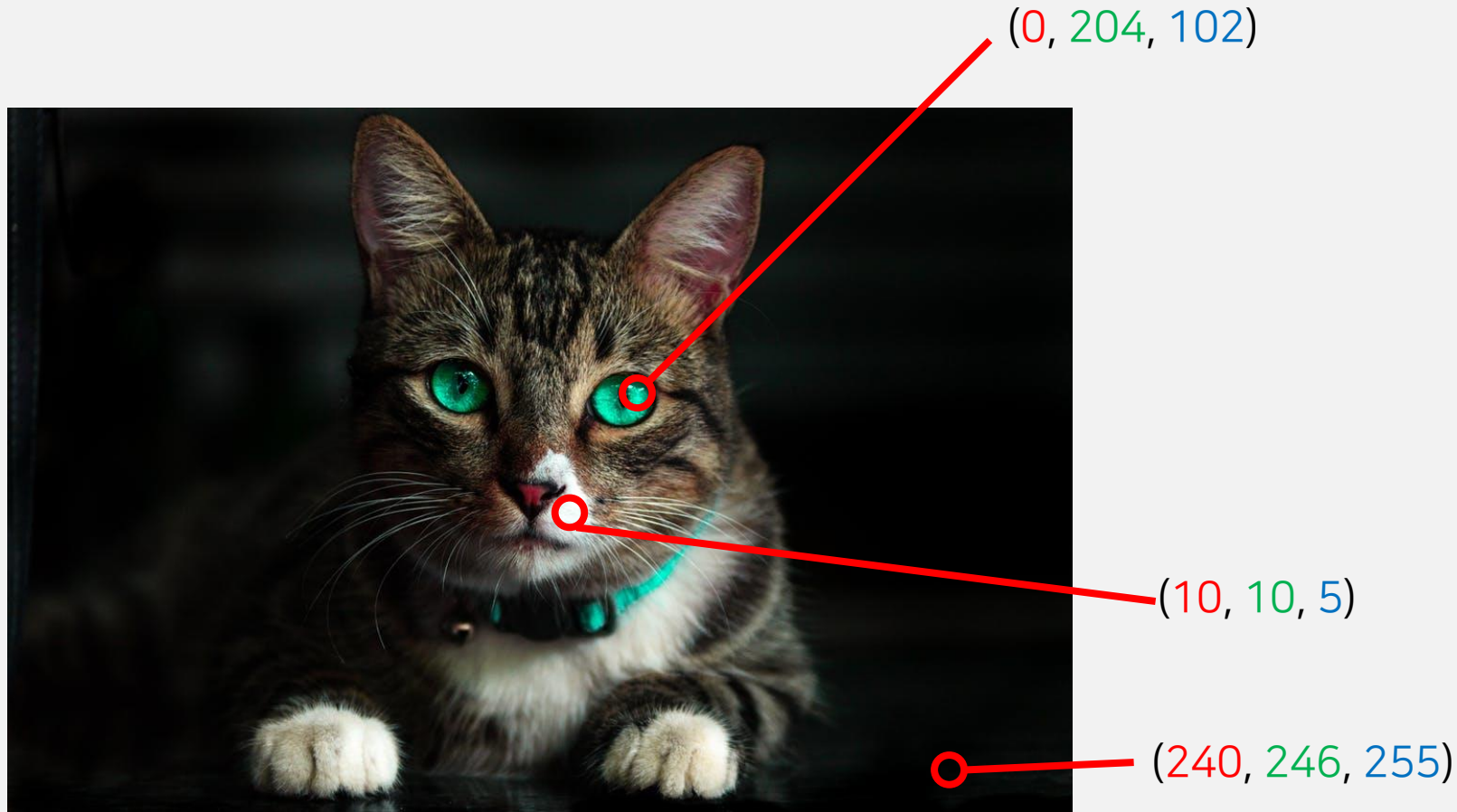
하얀색 (255, 255, 255)

검은색 (0,0,0)



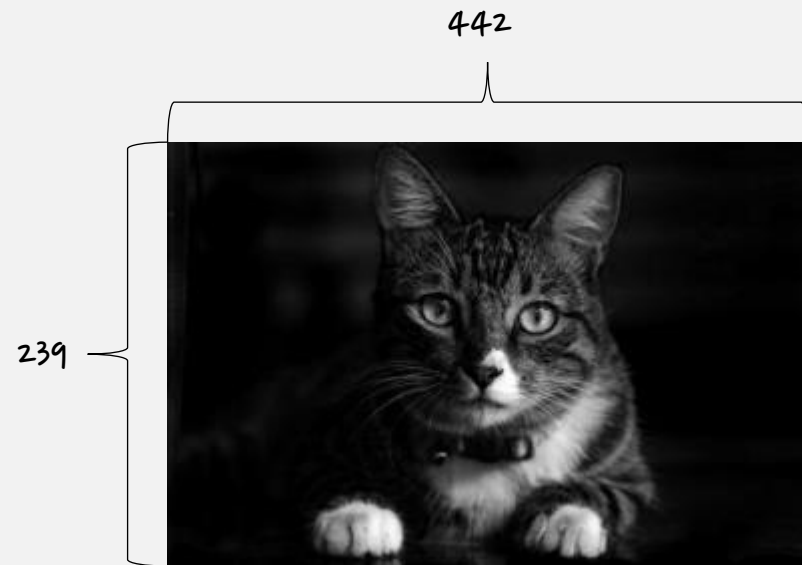


# RGB Representation of Picture



# Matrix Representation of Picture

- 이미지를 텐서로 나타내보자.



각 원소가 0~255 값을 갖는 행렬  
(2차원 텐서) (239x442)



각 원소가 0~255 값을 갖는 2차원 텐서 x RGB 3색상  
(3차원 텐서) (239x442x3)

# 미분 Differential

생각보다 쉬워요! 걱정 ㄴ ㄴ

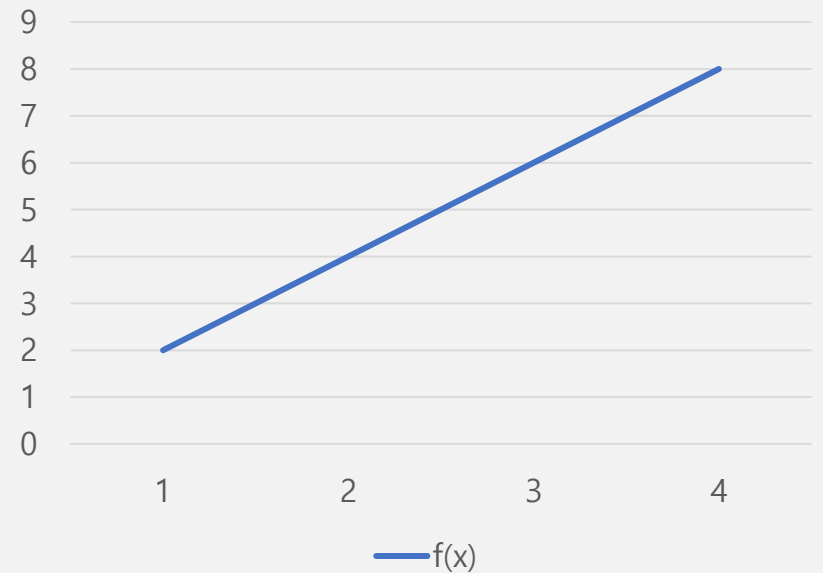
# 해석적 미분 vs 수치 미분

- 지금까지 우리는 겁나 어렵고 계산하기 뻘썩, 외울 것 많은 미분을 배웠다.
- 보통 우리까지 배운 미분, 공식을 이용해 정확한 기울기를 구하는 미분을 해석적 미분이라 한다. (Analytical Diff)
- 우리는 지금부터 쉽고 편한 수치 미분을 배울 것이다. (Numerical Diff)

# 미분과 기울기

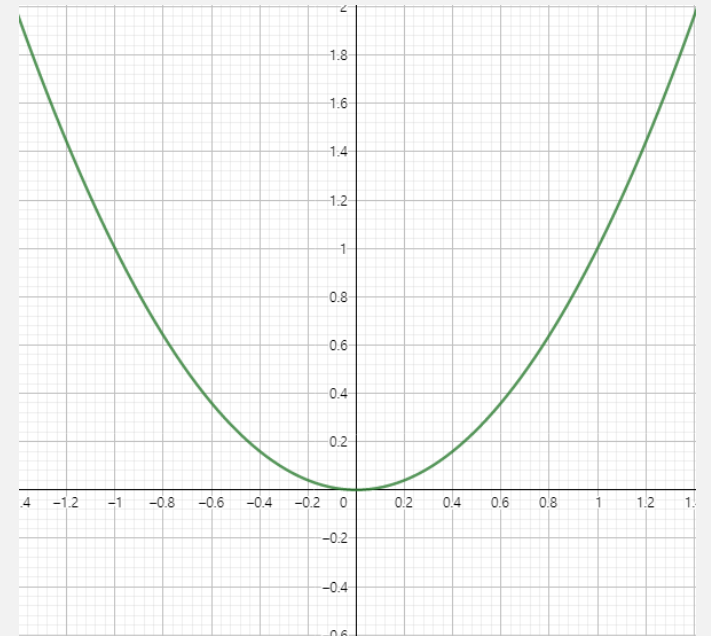
- 함수  $f$ 가 아래와 같은 그래프를 그린다고 생각해보자.
- $f(x) = 2x$
- $f(1) = 2, f(2) = 4$ 이다.  
즉,  $x$ 가 1 커지는 동안  $f(x)$ 는 2 커졌다.  
=  $x$  하나에 대한 함수  $f$ 의 변화율은 2이다.  
= 이것 식으로 나타내면...

$$\frac{df}{dx} = 2$$



# 기울기가 변하는 식

- 이 세상 모든 식이 일차방정식이면 좋겠지만, 그렇지 않다.
- $f(x) = x^2$
- 선형이 아닌 식들의 기울기는  $x$ 에 따라 계속 변한다.
- $x=2$  일 때 기울기는  $x=2.1$ 의 기울기와 대강 비슷하다.
- 이를 이용해서 변화율을 대강(근사해서) 구한다.
- $f(2.1) - f(2) = 4.41 - 4 = 0.41$
- $x$ 가 0.1 움직이는 동안 0.41 변했다.
- $x=2$ 에서  $x$  하나에 대한 변화율은 대강 4.1이다.
- 실제  $x=2$ 에서의 기울기를 해석적으로 구하면 4이다.



# 수치 미분의 장단점

## 장점

1. 아무리 복잡한 식이라도, 함수에 값 두 번 넣어보면 기울기를 구할 수 있다.
2. 복잡한 해석적 미분에 비해 컴퓨팅 자원을 아낄 수 있다. (핵심!!!)
3. 미분에 사용하는 두 값의 차를 아주 작게 하면, 오차가 아주 작게 기울기를 구할 수 있다.

## 단점

1. 정확한 기울기가 아니다.
2. 계산할 때마다 오차가 발생하기 때문에, 나비효과가 일어날 수도 있다.

# 미니 프로젝트 마스크 착용 여부 판별기 만들기

with Teachable Machine



# Teachable Machine

- Teachable Machine은 구글에서 제공하는 무료 웹 AI 개발 도구입니다.
- 이 도구를 이용해 데이터를 학습시켜, 마스크 착용 여부를 판별하는 AI를 만들어보세요!
- 원한다면 다른 주제로 하셔도 괜찮습니다.
- 간단한 프로젝트 소개를 작성하여 디스코드에 공유해주세요!
- 이름을 원하시는 분은 저에게 DM으로 제출하셔도 괜찮습니다.
- 제출된 프로젝트로 투표를 진행하여, 1등에게 스타벅스 기프티콘을 드립니다!
- <https://teachablemachine.withgoogle.com/>
- [참고자료](#)

# 감사합니다!

1주차가 끝났습니다. 앞으로 잘 부탁드립니다!