

< 2022/03/08 >

# Deep Learning

## 1. Introduction

Kyungwoo Song

- Basics Math
- Introduction to Machine Learning
  - Example) Decision Tree, Logistic Regression, ...
- Introduction to Deep Learning
  - Example) CNN, Transformer, ...
- The Recent Advanced ML/DL
  - Example) Deep Generative Model, Bayesian Neural Nets

- Instructor: Kyungwoo Song
  - [kyungwoo.song@uos.ac.kr](mailto:kyungwoo.song@uos.ac.kr)
  - 02-6490-2474
  - Office Hours: <https://mlai.uos.ac.kr/contact>
- Class Format
  - Mon 13:00~15:00: Recording
  - Tue 12:00~13:00: Live class with Zoom
    - ❖ Make-up class for paper review, programming, mathematics, ...
- Reference
  - No textbook
  - <http://web.stanford.edu/class/cs224n/>
  - <http://cs231n.stanford.edu/>
  - Neurips, ICML, ICLR, AAAI, AISTATS, ACL, EMNLP, NAACL, ...

비어 있는 시간은 모두 가능합니다.  
수업, 진로 등 도움이 필요한 학생  
은, 비어 있는 시간 중 편한 시간을  
메일로 연락 부탁드립니다.

본 수업은 3,4학년 및 대학원교과목으로,  
머신러닝 분야의 연구수행 능력 함양을 목표로 합니다.

- Grading Policy

- Attendance: 10%
- Midterm: 30%
- Final Exam: 30%
- Homework: 30%
  - ❖ Three Programming HW (Google Colab)
  - ❖ One Paper Review HW (Papers will be assigned)
  - ❖ Final Project (Project Proposal): 30%
    - 7min presentation
  - ❖ Define you own problems!

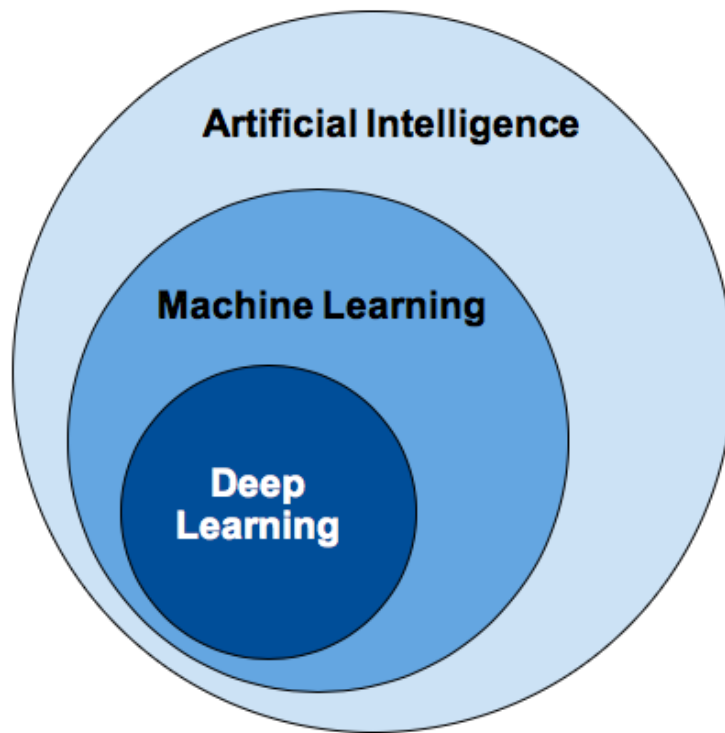
2~3명에서 한팀 (추후 공지)

프로젝트의 완성된 결과물이 나올 필요는 없습니다. 1) 기존에 어떤 연구가 존재하고, 2) 이러한 연구의 문제점은 무엇이고, 3) 해당 문제점을 해결하기 위해 이렇게 접근해보고 싶다 등의 “계획”을 작성하시면 됩니다.  
도움이 필요한 친구들은, 언제든지 연락주세요.

- 3,4학년 교과목이며, 대학원 공통 교과목 입니다.
  - 그 까닭에, 기본적인 내용뿐 아니라, 다양한 심화된 내용도 함께 다룰 예정입니다.
- 기본적인 수학 및 프로그래밍 지식을 가지고 계신것으로 전제로 수업이 진행 됩니다.
- 만약 그렇지 않을 경우에는, 개인적으로 많은 시간을 투자하시는것을 추천드립니다.
  - 예시
    - ❖Python 튜토리얼1: <https://programmers.co.kr/learn/courses/2>
    - ❖Python 튜토리얼2: <https://cs231n.github.io/python-numpy-tutorial/>
    - ❖수학 튜토리얼1: [http://www.kmooc.kr/courses/course-v1:SKKUK+SKKU\\_45+2020\\_T1/about#preview-video-modal](http://www.kmooc.kr/courses/course-v1:SKKUK+SKKU_45+2020_T1/about#preview-video-modal)

수업의 전반부에는 중요한 전통적인 내용들을,  
그리고 후반부로 갈 수록, 최근 연구들을  
기반으로 수업이 진행됩니다.

- AI, ML, DL은 대체 무엇인가요?
- AI: Artificial Intelligence (Rule-based model, ...)
- ML: Machine Learning (로지스틱 회귀분석, 서포트 벡터 머신, ...)
- DL: Deep Learning (뉴럴 넷, ...)



AI, ML, DL 굳이 구분하자면 다르긴 합니다.

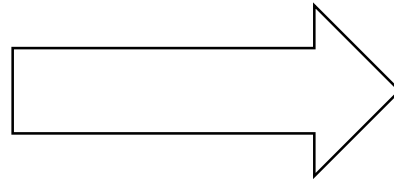
하지만 실제 연구자들은 많은 경우 구분하지 않는 편이며, 본 강의에서도 굳이 구분하지 않도록 하겠습니다.

- Neural net 기반의 Image classification 기준 예시
- 2개의 class 중에서 하나를 맞추는 것이라고 가정해보자.

< Li et al. 2021 >



아직 우리는, Text를 어떻게 표현하는지에 대해서는 배우지 않았기에, 이미지로 예시를 소개 하겠습니다.



**Cat**  
Dog

# AI는 어떻게 작동하나요?

AI101

컴퓨터가 보기에는...

|      |     |     |     |     |     |     |     |     |     |     |     |     |     |     |      |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| [105 | 112 | 108 | 111 | 104 | 99  | 106 | 99  | 96  | 103 | 112 | 119 | 104 | 97  | 93  | 87]  |
| [ 91 | 98  | 102 | 106 | 104 | 79  | 98  | 103 | 99  | 105 | 123 | 136 | 110 | 105 | 94  | 85]  |
| [ 76 | 85  | 90  | 105 | 128 | 105 | 87  | 96  | 95  | 99  | 115 | 112 | 106 | 103 | 99  | 85]  |
| [ 99 | 81  | 81  | 93  | 120 | 131 | 127 | 100 | 95  | 98  | 102 | 99  | 96  | 93  | 101 | 94]  |
| [106 | 91  | 61  | 64  | 69  | 91  | 88  | 85  | 101 | 107 | 109 | 98  | 75  | 84  | 96  | 95]  |
| [114 | 108 | 85  | 55  | 55  | 69  | 64  | 54  | 64  | 87  | 112 | 129 | 98  | 74  | 84  | 91]  |
| [133 | 137 | 147 | 103 | 65  | 81  | 80  | 65  | 52  | 54  | 74  | 84  | 102 | 93  | 85  | 82]  |
| [128 | 137 | 144 | 140 | 109 | 95  | 86  | 70  | 62  | 65  | 63  | 63  | 60  | 73  | 86  | 101] |
| [125 | 133 | 148 | 137 | 119 | 121 | 117 | 94  | 65  | 79  | 80  | 65  | 54  | 64  | 72  | 98]  |
| [127 | 125 | 131 | 147 | 133 | 127 | 126 | 131 | 111 | 96  | 89  | 75  | 61  | 64  | 72  | 84]  |
| [115 | 114 | 109 | 123 | 150 | 148 | 131 | 118 | 113 | 109 | 100 | 92  | 74  | 65  | 72  | 78]  |
| [ 89 | 93  | 90  | 97  | 108 | 147 | 131 | 118 | 113 | 114 | 113 | 109 | 106 | 95  | 77  | 80]  |
| [ 63 | 77  | 86  | 81  | 77  | 79  | 102 | 123 | 117 | 115 | 117 | 125 | 125 | 130 | 115 | 87]  |
| [ 62 | 65  | 82  | 89  | 78  | 71  | 80  | 101 | 124 | 126 | 119 | 101 | 107 | 114 | 131 | 119] |
| [ 63 | 65  | 75  | 88  | 89  | 71  | 62  | 81  | 120 | 138 | 135 | 105 | 81  | 98  | 110 | 118] |
| [ 87 | 65  | 71  | 87  | 106 | 95  | 69  | 45  | 76  | 130 | 126 | 107 | 92  | 94  | 105 | 112] |
| [118 | 97  | 82  | 86  | 117 | 123 | 116 | 66  | 41  | 51  | 95  | 93  | 89  | 95  | 102 | 107] |
| [164 | 146 | 112 | 80  | 82  | 120 | 124 | 104 | 76  | 48  | 45  | 66  | 88  | 101 | 102 | 109] |
| [157 | 170 | 157 | 120 | 93  | 86  | 114 | 132 | 112 | 97  | 69  | 55  | 70  | 82  | 99  | 94]  |
| [130 | 128 | 134 | 161 | 139 | 100 | 109 | 118 | 121 | 134 | 114 | 87  | 65  | 53  | 69  | 86]  |
| [128 | 112 | 96  | 117 | 150 | 144 | 120 | 115 | 104 | 107 | 102 | 93  | 87  | 81  | 72  | 79]  |
| [123 | 107 | 96  | 86  | 83  | 112 | 153 | 149 | 122 | 109 | 104 | 75  | 80  | 107 | 112 | 99]  |
| [122 | 121 | 102 | 80  | 82  | 86  | 94  | 117 | 145 | 148 | 153 | 102 | 58  | 78  | 92  | 107] |
| [122 | 164 | 148 | 103 | 71  | 56  | 78  | 83  | 93  | 103 | 119 | 139 | 102 | 61  | 69  | 84]  |

< Li et al. 2021 >



Size: 32\*32\*3 (RGB)

Text는 어떻게 입력해야 할 지,  
한번 고민해보세요!

**Cat**, Dog

$$f(x, W) = Wx$$

이미지  
(3072\*1)

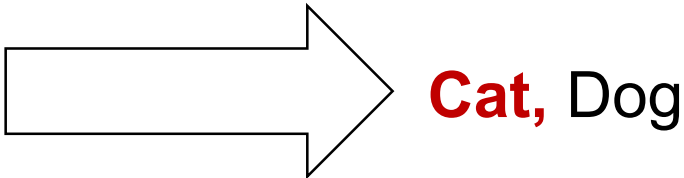
Weight  
Parameter  
(최적화 대상)  
(2\*3072)

(2\*1)

Source:



< Li et al. 2021 >



$$s = f(x, W) = Wx$$

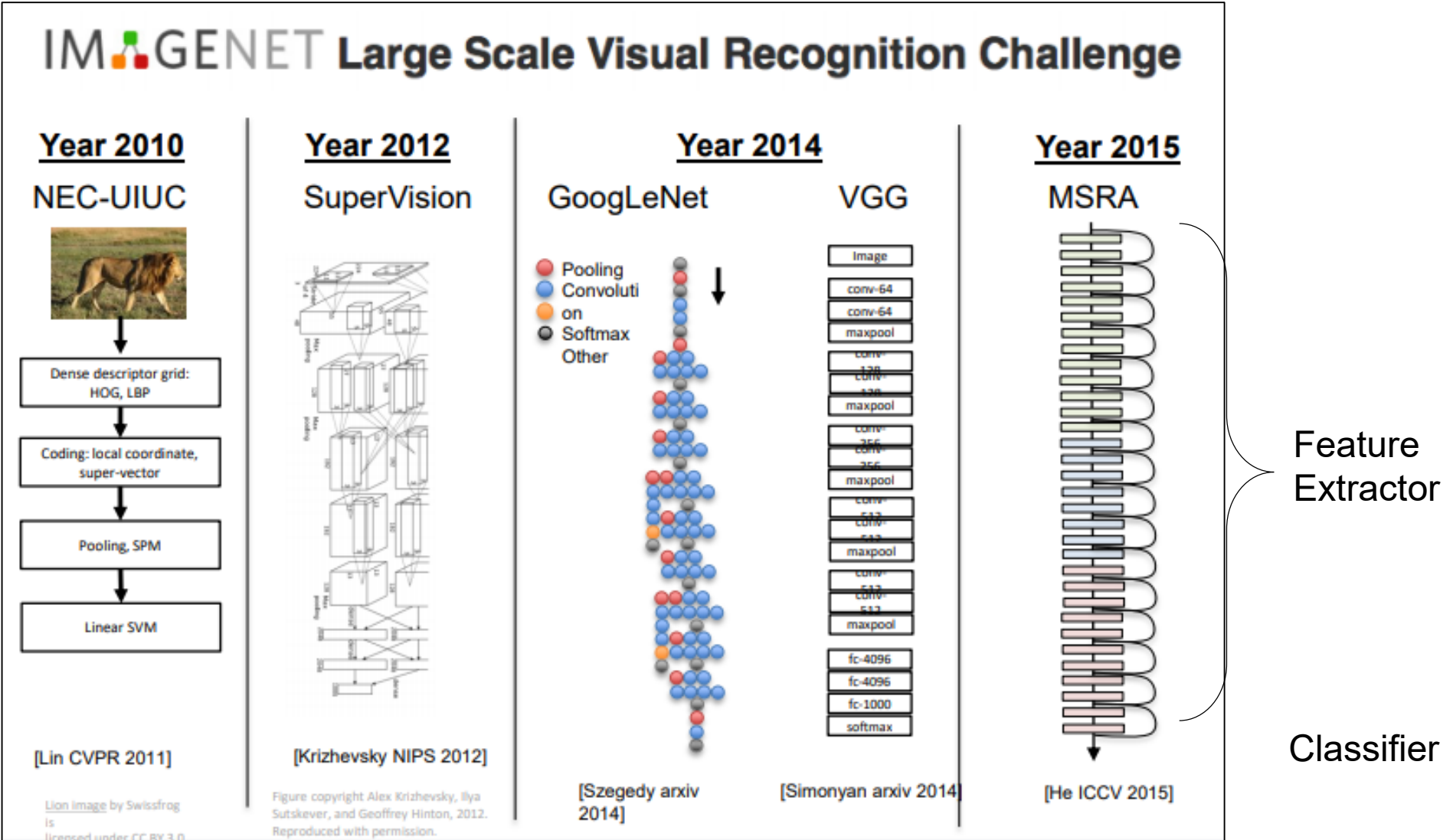
$$\hat{y}_i = \frac{\exp(s_i)}{\sum_{j=1}^2 \exp(s_j)}$$

$i$  번째 class일 확률 (우리의 예측치)

- 주어진 data:  $(x, y)$
- $x$ : 이미지,  $y$ : 레이블 (정답지)
- $\hat{y}$  (우리의 예측치)와  $y$  가 가까워지도록  $W$ 를 최적화!
- 목적식:  $L(y, \hat{y}) = -\sum_{i=1}^2 y_i \log \hat{y}_i$
- $W \leftarrow W - \frac{\partial L}{\partial W}$ 을 통해서 최적화

| $y$ | $\hat{y}$ |
|-----|-----------|
| 1   | 0.9       |
| 0   | 0.1       |

Source:



< Li et al. 2021 >

Source: [http://cs231n.stanford.edu/slides/2021/lecture\\_1\\_feifei.pdf](http://cs231n.stanford.edu/slides/2021/lecture_1_feifei.pdf)

# Colab

- Colaboratory
  - 줄여서 'Colab'
  - 브라우저에서 Python을 작성하고 실행할 수 있음
  - Colab 소개 영상:  
[https://www.youtube.com/watch?v=inN8seMm7UI&ab\\_channel=TensorFlow](https://www.youtube.com/watch?v=inN8seMm7UI&ab_channel=TensorFlow)
- Colab의 장점
  - 간편한 실행 환경 구축 (원하는 library 들을 쉽게 설치 가능)
  - GPU 무료 액세스 (유료 요금제도 있지만, 본 수업에서는 무료여도 충분)
  - 간편한 공유 (Google Drive와 연동)



[https://colab.research.google.com/#scrollTo=gJr\\_9dXGpJ05](https://colab.research.google.com/#scrollTo=gJr_9dXGpJ05)

[https://colab.research.google.com/github/pytorch/pytorch.github.io/blob/master/assets/hub/huggingface\\_pytorch-](https://colab.research.google.com/github/pytorch/pytorch.github.io/blob/master/assets/hub/huggingface_pytorch-transformers.ipynb#scrollTo=HZsm20AVO1d-)

Source: [transformers.ipynb#scrollTo=HZsm20AVO1d-](https://colab.research.google.com/github/pytorch/pytorch.github.io/blob/master/assets/hub/huggingface_pytorch-transformers.ipynb#scrollTo=HZsm20AVO1d-)

```

Tokenizer
The tokenizer object allows the conversion from character strings to tokens understood by the different models. Each model has its own
tokenizer, and some tokenizing methods are different across tokenizers. The complete documentation can be found here.


[] import torch
tokenizer = torch.hub.load('huggingface/pytorch-transformers', 'tokenizer', 'bert-base-uncased') # Download vocabulary from S3 and cache.
tokenizer = torch.hub.load('huggingface/pytorch-transformers', 'tokenizer', './test/bert_saved_model/') # E.g. tokenizer was saved using 'save_pretrained('./test/saved_model/')'

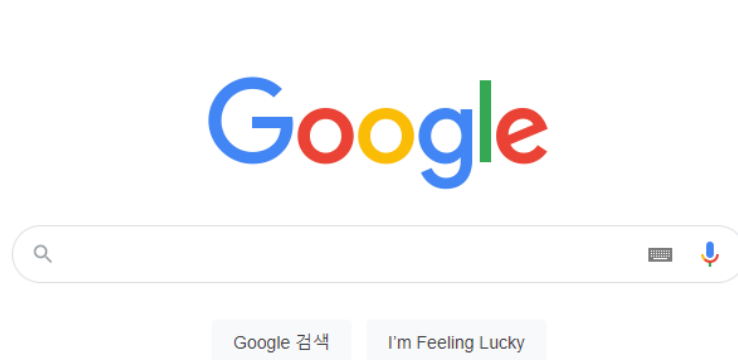
Models
The model object is a model instance inheriting from a nn.Module. Each model is accompanied by their saving/loading methods, either from a
local file or directory, or from a pre-trained configuration (see previously described config). Each model works differently, a complete overview
of the different models can be found in the documentation.

[] import torch
model = torch.hub.load('huggingface/pytorch-transformers', 'model', 'bert-base-uncased') # Download model and configuration from S3 and cache.
model = torch.hub.load('huggingface/pytorch-transformers', 'model', './test/bert_model/') # E.g. model was saved using 'save_pretrained('./test/saved_model/')'
model = torch.hub.load('huggingface/pytorch-transformers', 'model', 'bert-base-uncased', output_attentions=True) # Update configuration during loading
assert model.config.output_attentions == True
# Loading from a TF checkpoint file instead of a PyTorch model (slower)
config = AutoConfig.from_json_file('./tf_model/bert_tf_model_config.json')
model = torch.hub.load('huggingface/pytorch-transformers', 'model', './tf_model/bert_tf_checkpoint_ckpt_index', from_tf=True, config=config)

Models with a language modeling head
Previously mentioned model instance with an additional language modeling head.
    
```

- Google 창을 열어주세요

Gmail 이미지  AI

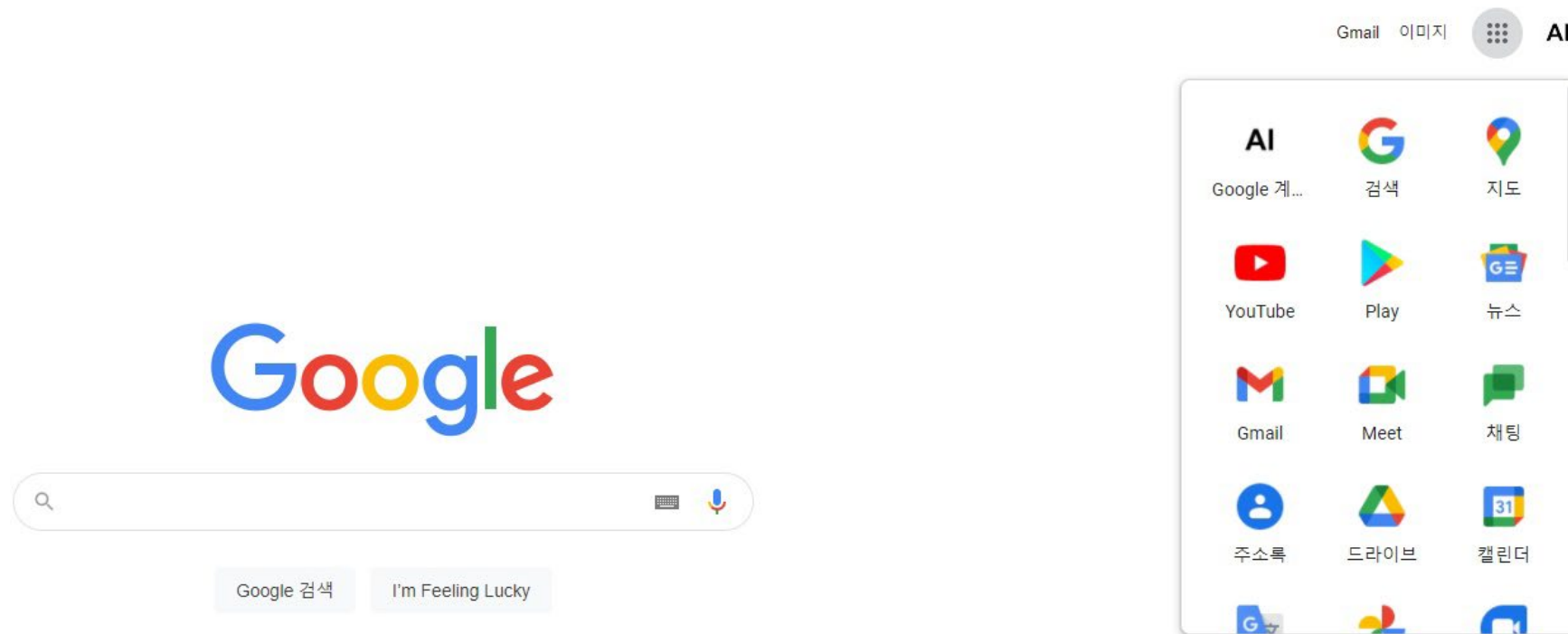


Source:

Kyungwoo Song, Department of AI, University of Seoul

13

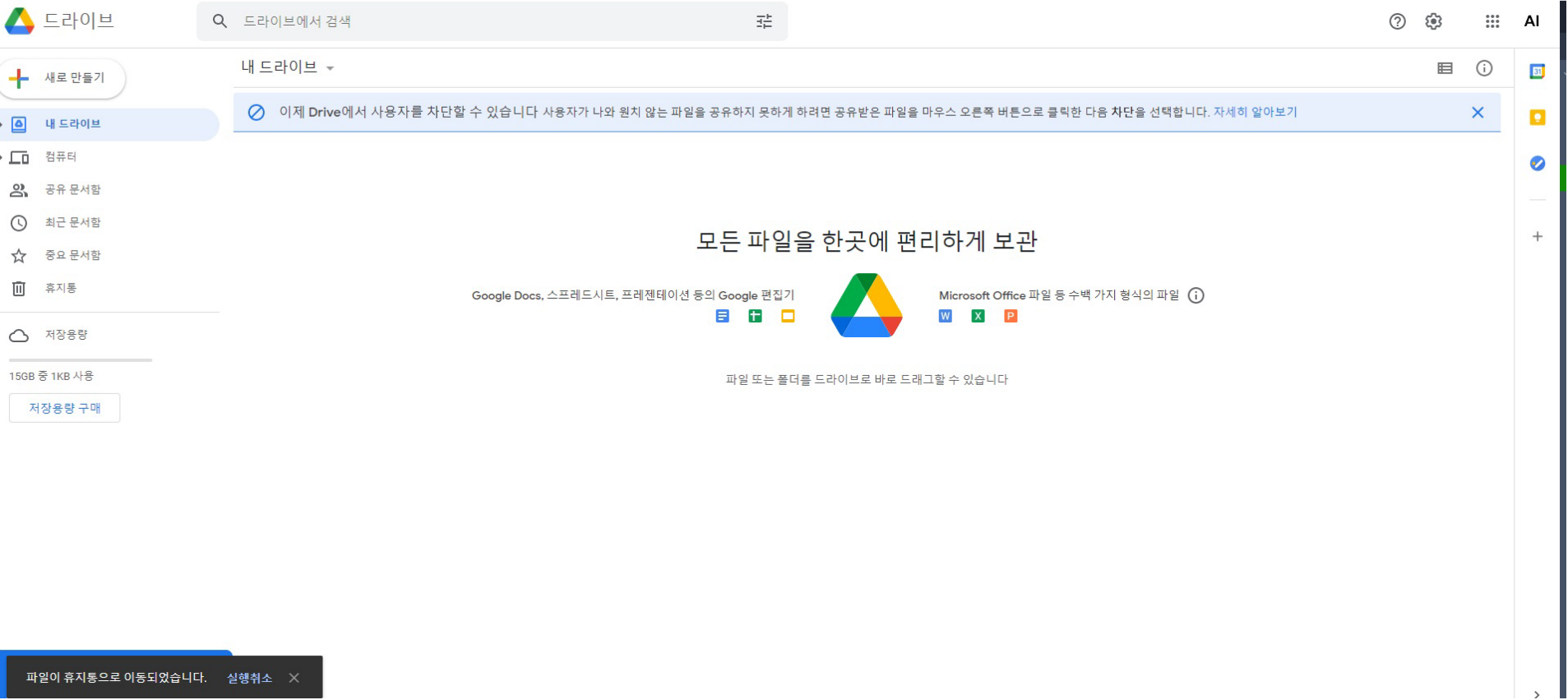
- 계정 옆의 아이콘을 클릭하여 구글 드라이브를 선택해 주세요.



# Google Colab 설치

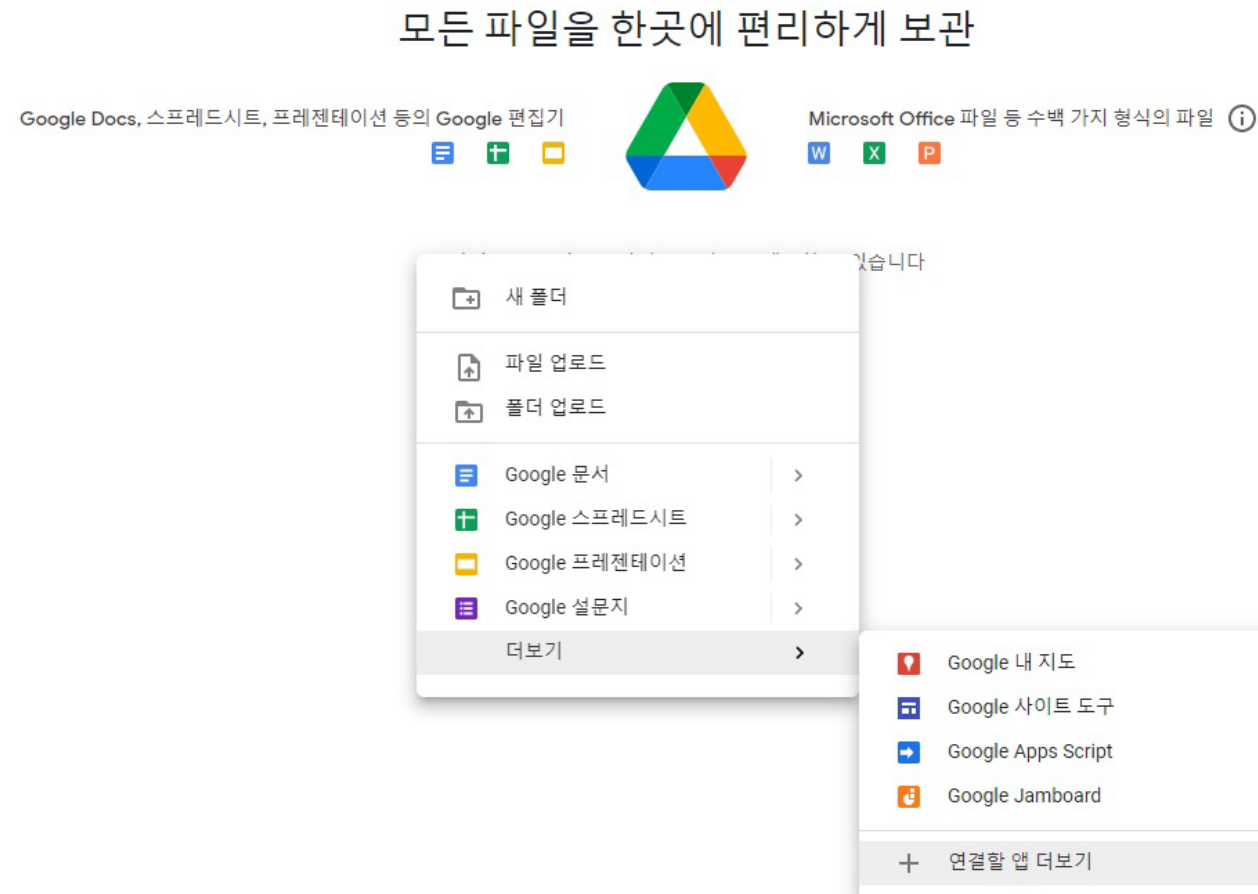
## Colab

- 구글 드라이브에 들어갑니다.



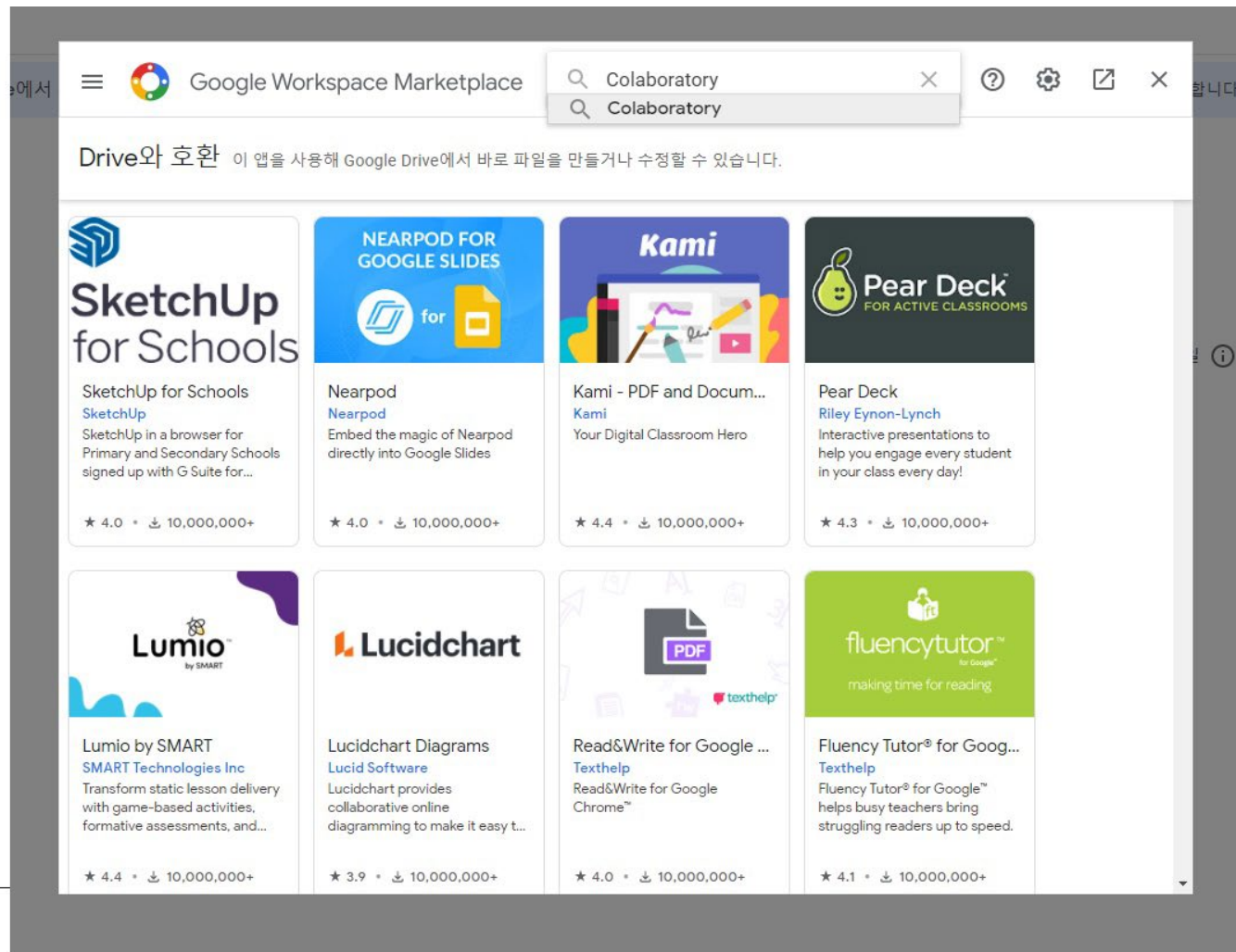
Source:

- 오른쪽 마우스를 클릭해서 연결할 앱 더보기를 클릭합니다.



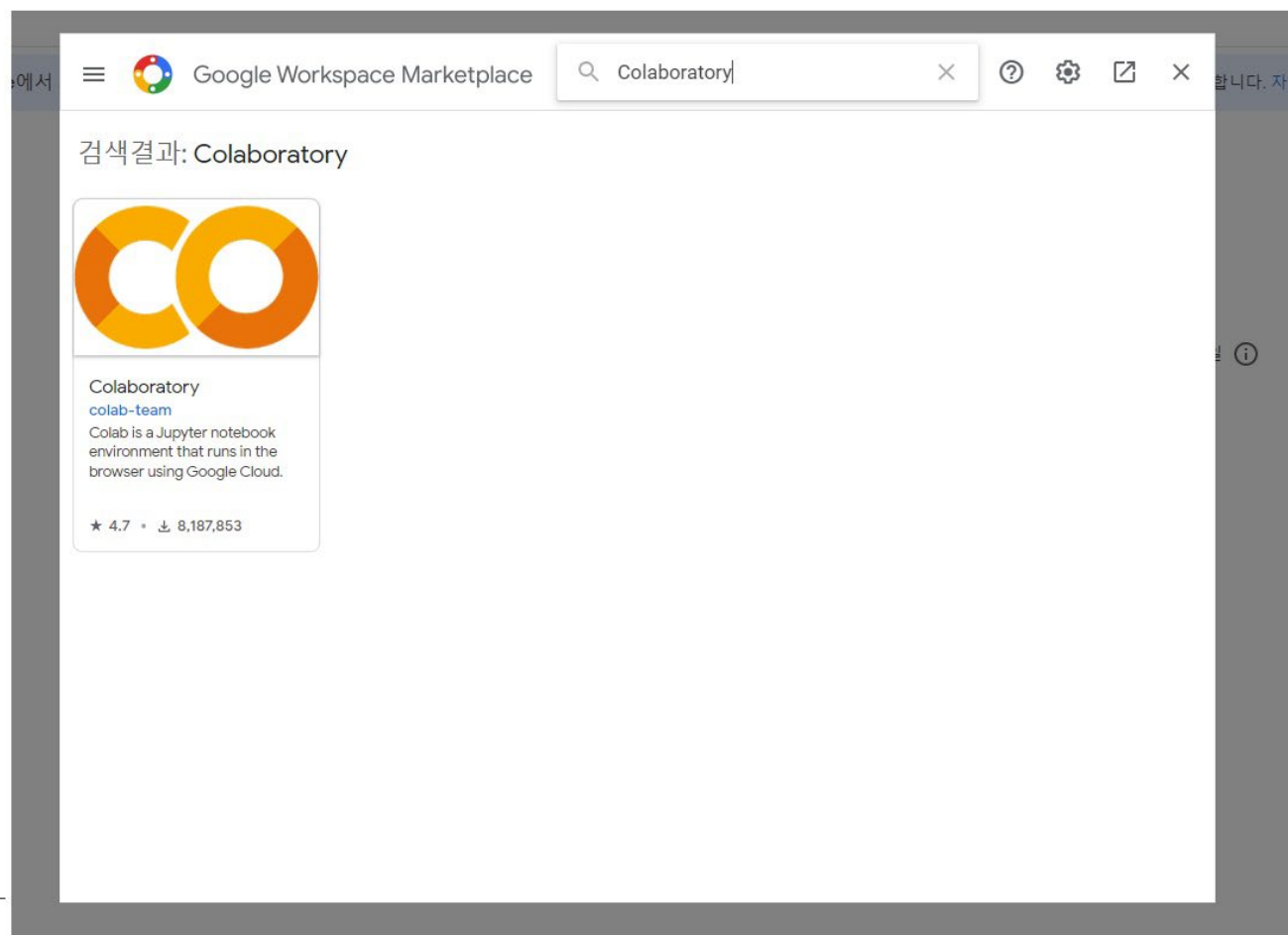


- Apps검색에서 Colaboratory를 검색합니다.



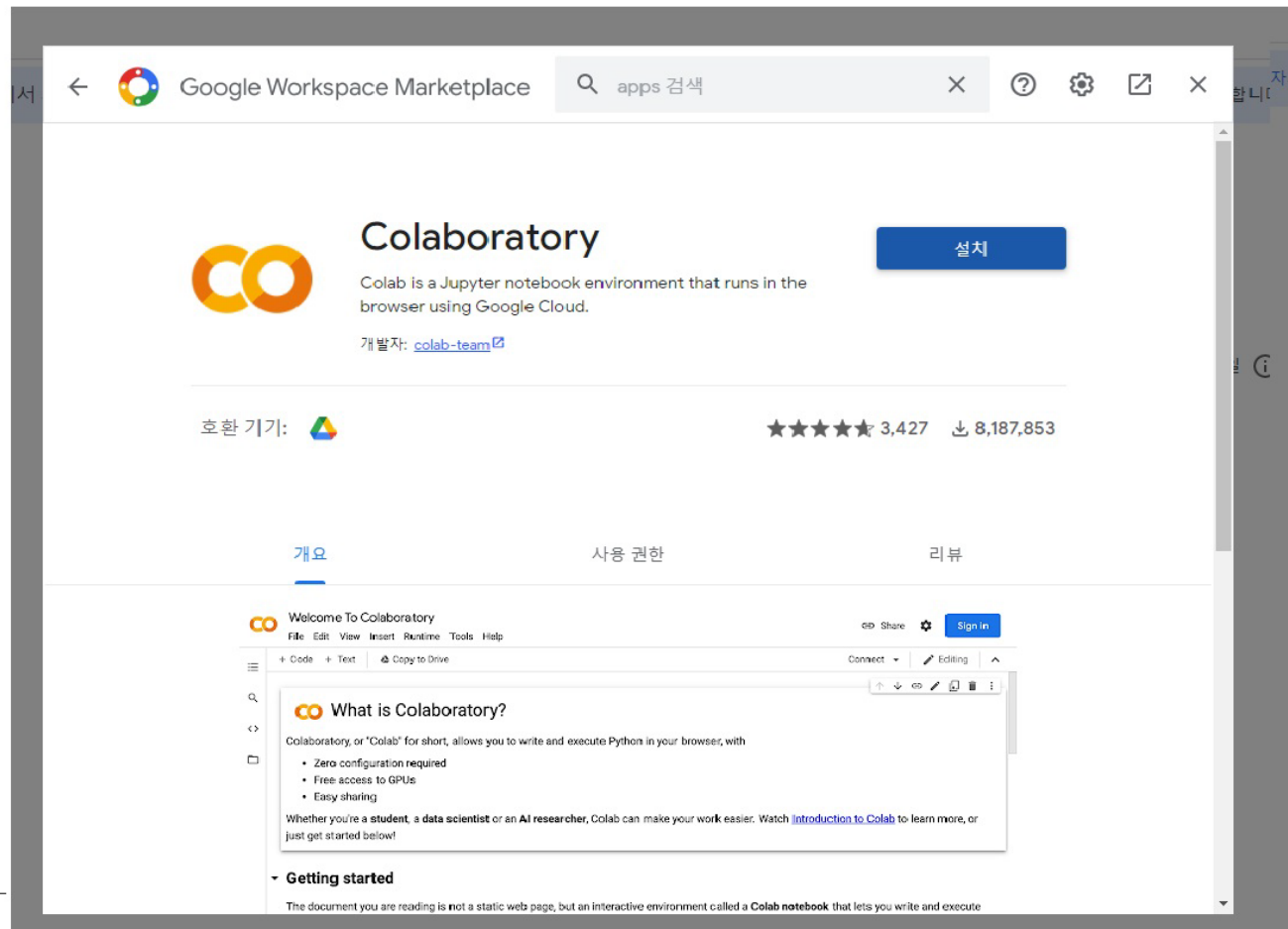
Source:

- 그러면 이와 같은 화면을 만날 수 있습니다. Colaboratory를 클릭합니다.



Source:

- Colaboratory 옆의 설치 버튼을 누릅니다.

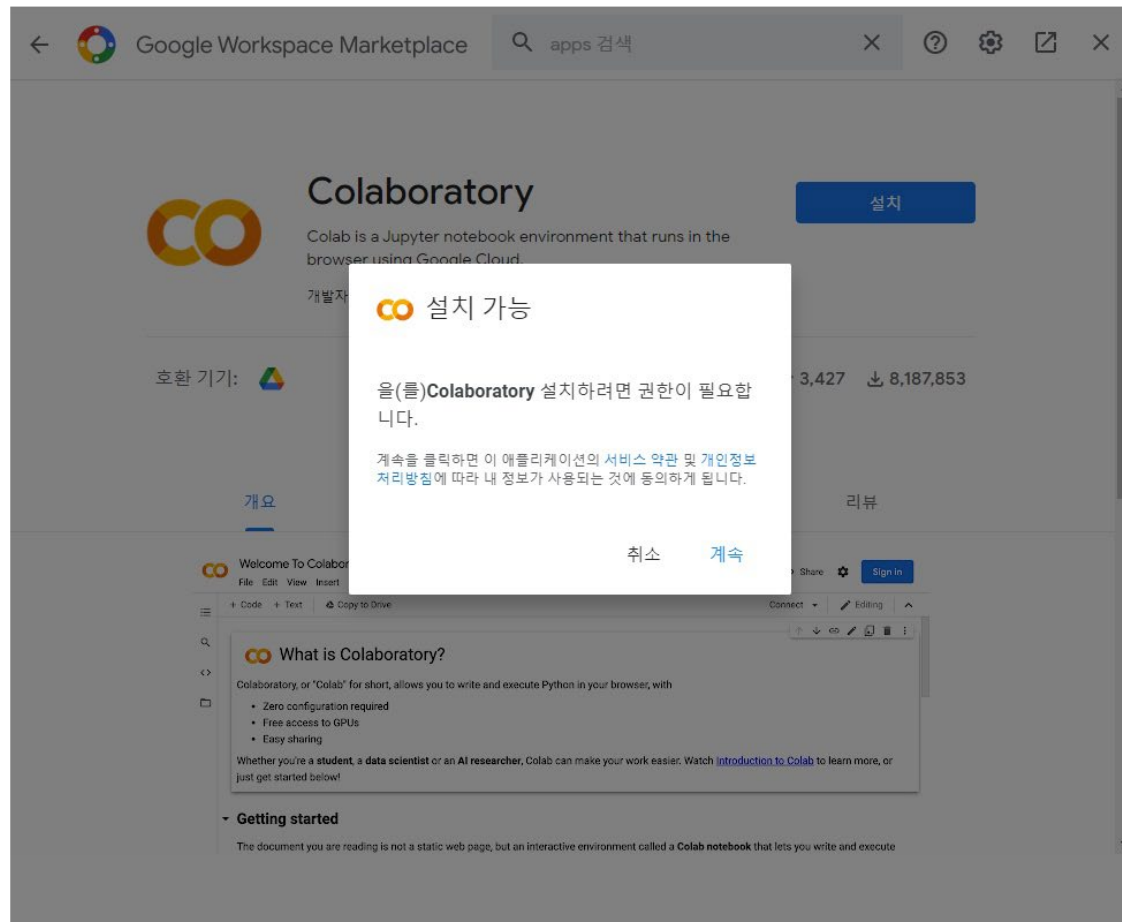


Source:

# Google Colab 설치

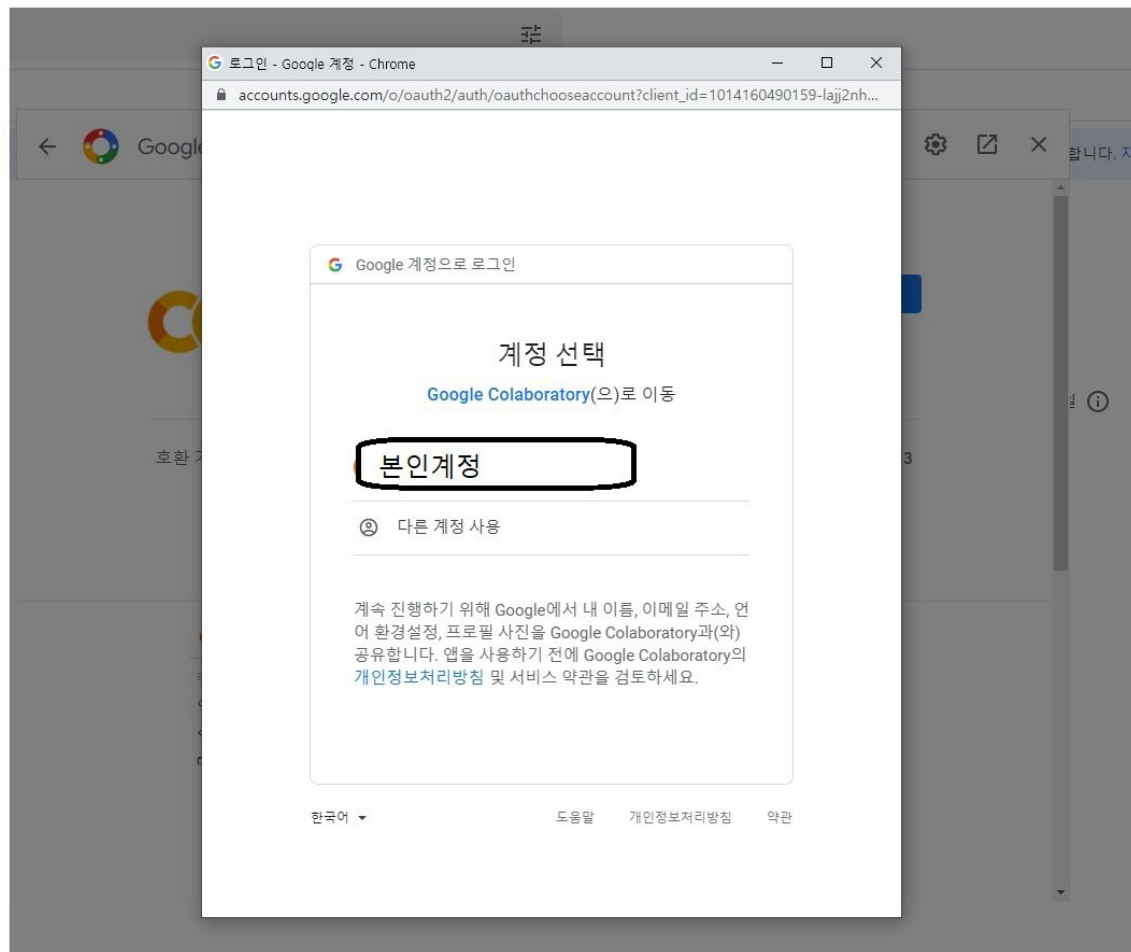
## Colab

- “계속” 버튼을 누릅니다.



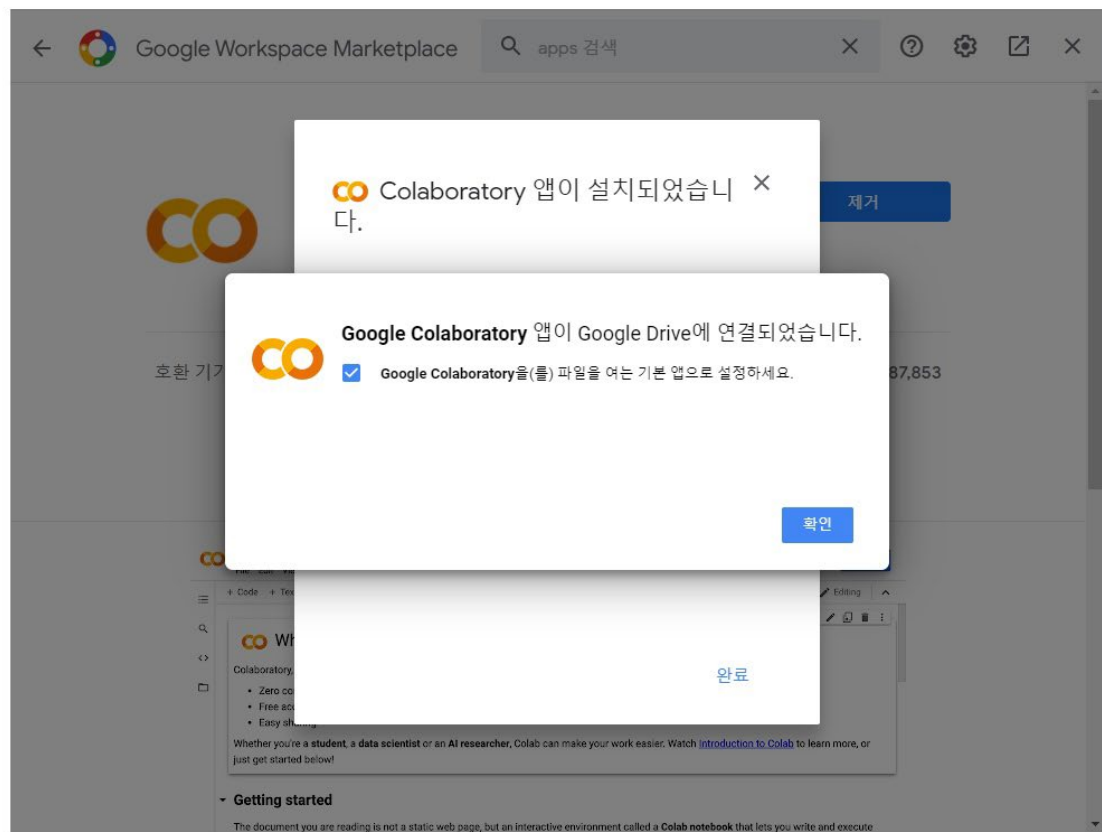
Source:

- 계정을 선택하는 창이 나옵니다. 본인의 생성된 계정을 클릭해주세요.



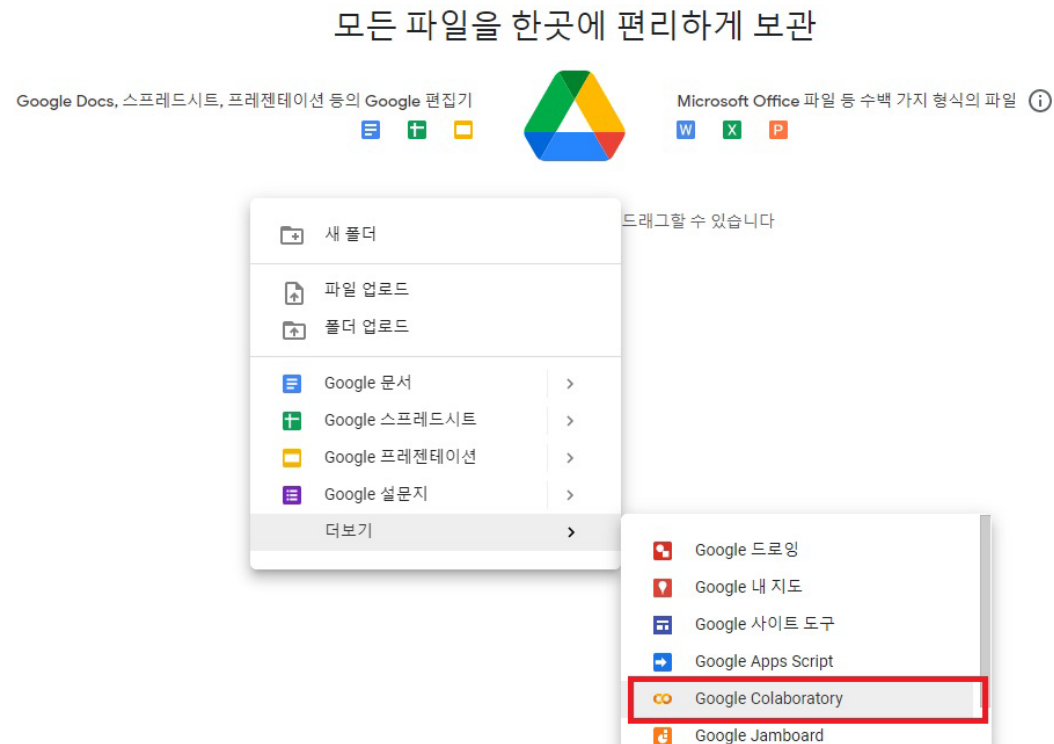
Source:

- 코랩이 생성되었다는 메시지가 나옵니다. 확인과 완료를 눌러주세요.



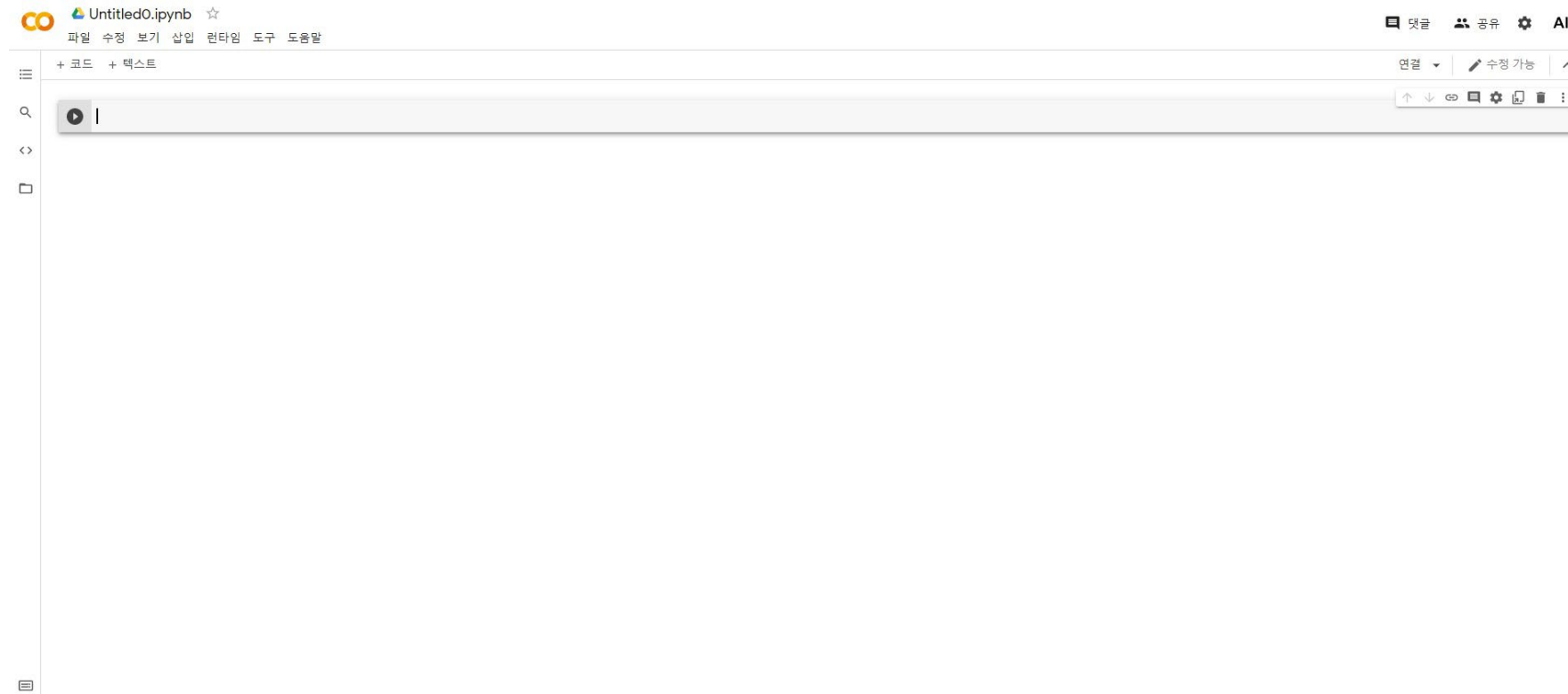
Source:

- 드라이브 메인에서 오른쪽 마우스를 클릭하여 코랩 노트북을 생성해 봅시다.



Source:

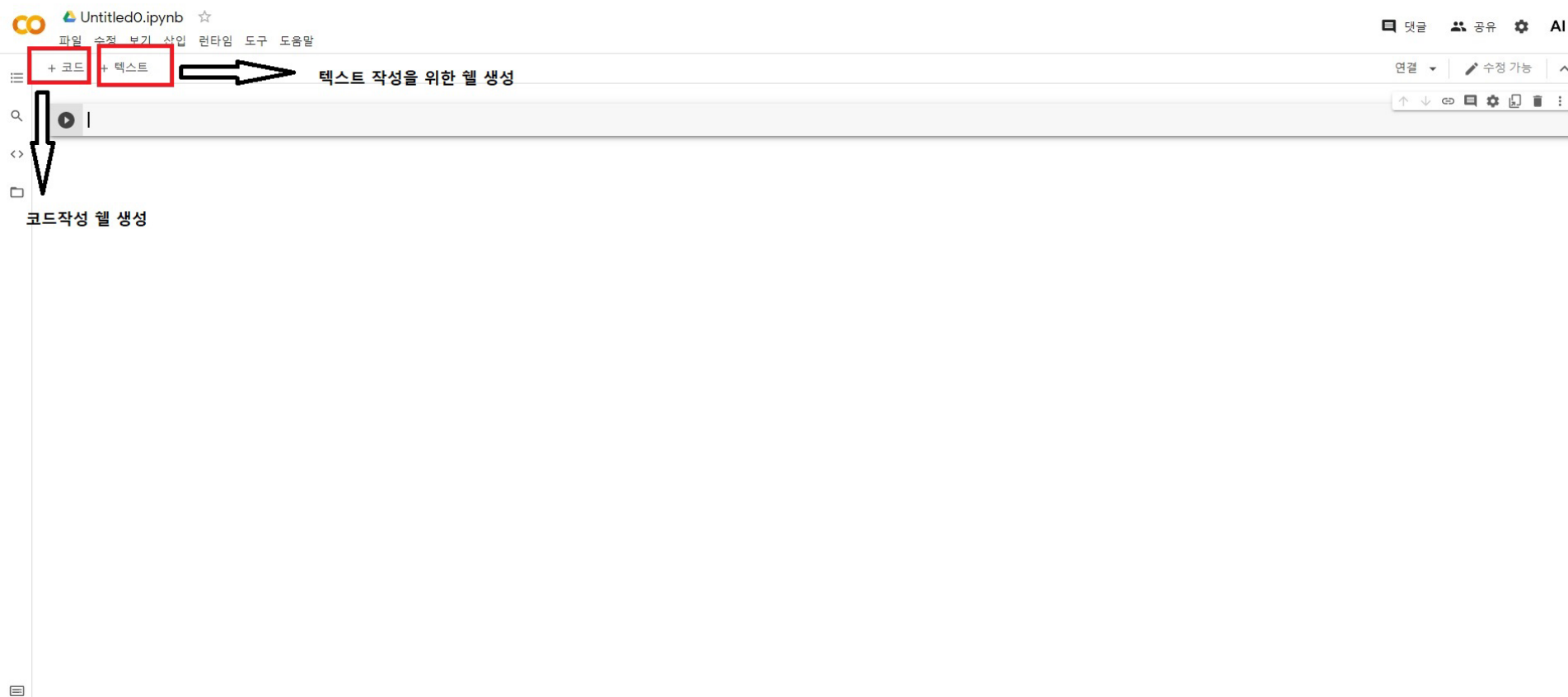
- 다음과 같은 화면을 만날 수 있습니다.



Source:

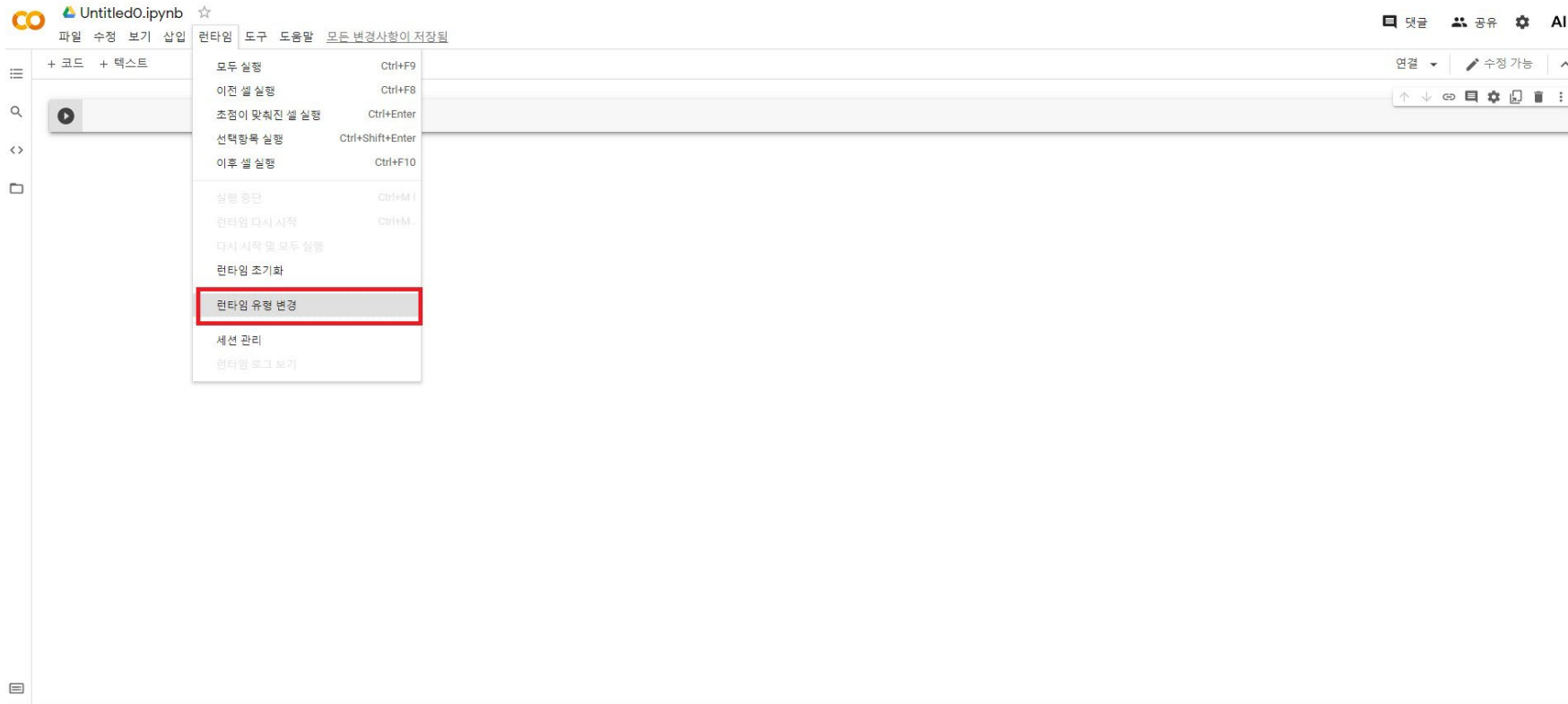


- 아래 두가지를 특히 많이 활용하게 됩니다.
  - 코드 작성을 위한 셀 생성
  - 중간 중간 내용 정리를 위한 텍스트 셀 생성



Source:

- 본 수업에서는 GPU를 사용할 일이 많기 때문에, 런타임에서 “런타임유형변경”을 클릭합니다.



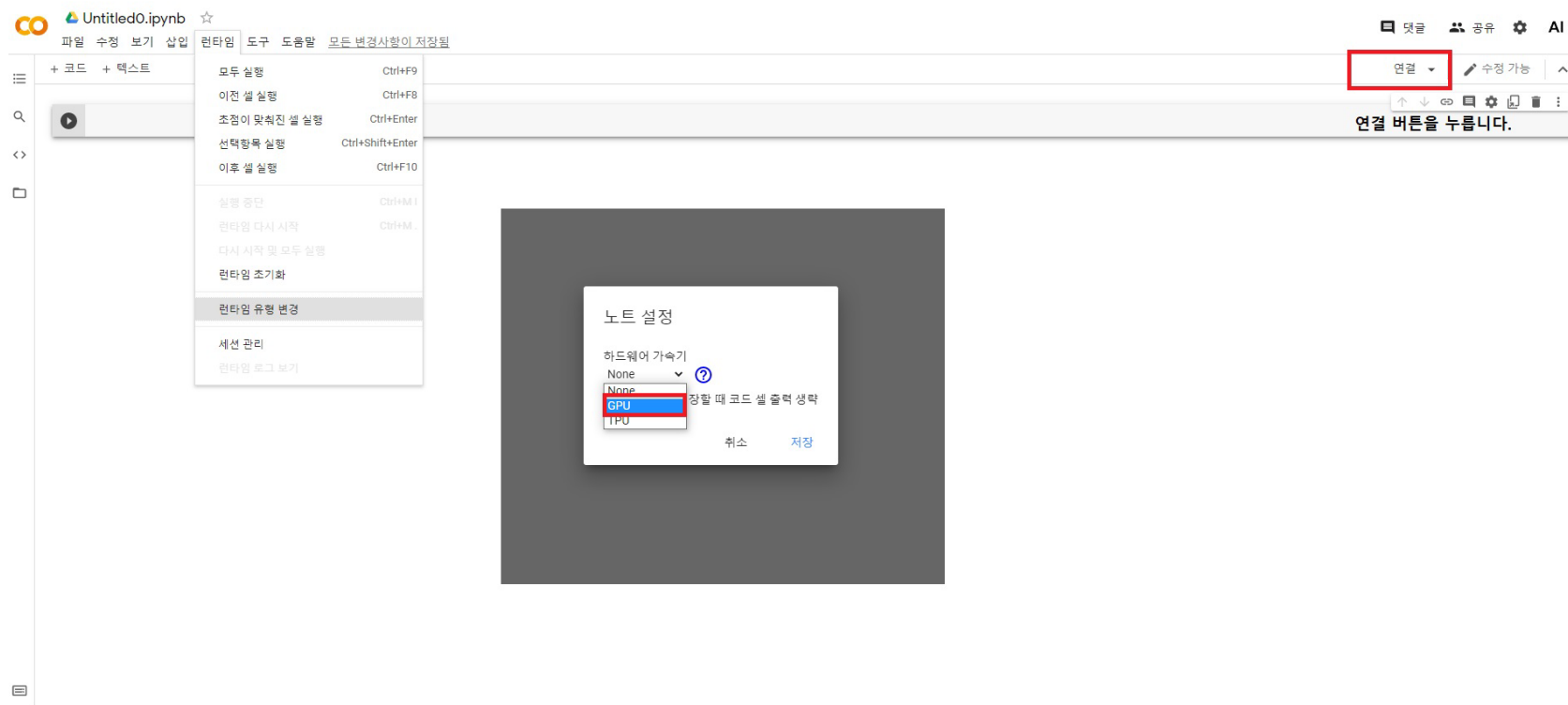
Source:

- 기본적으로 설정되어 있는 None을 GPU로 변경합니다.



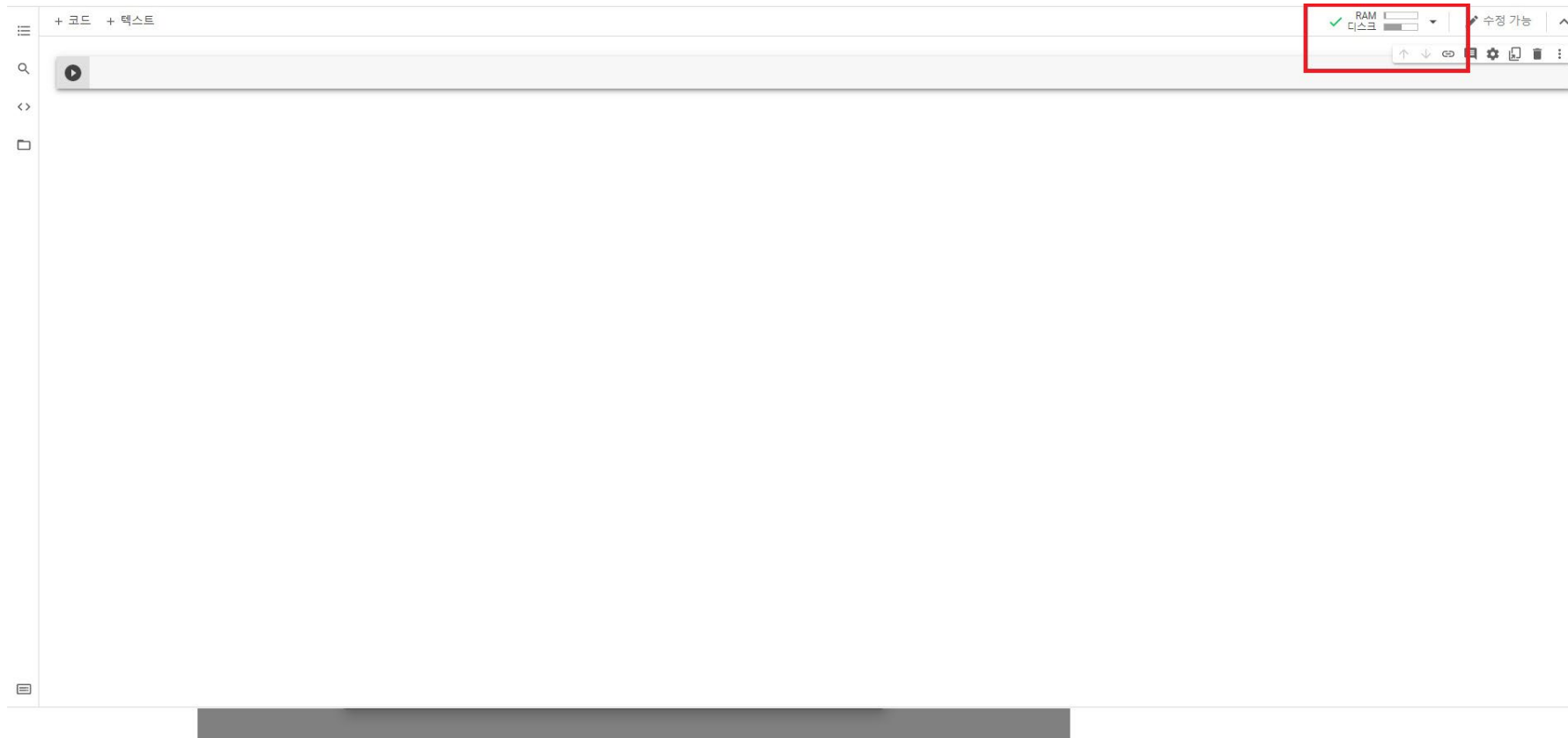
Source:

- 마지막으로 “연결” 버튼을 누릅니다.

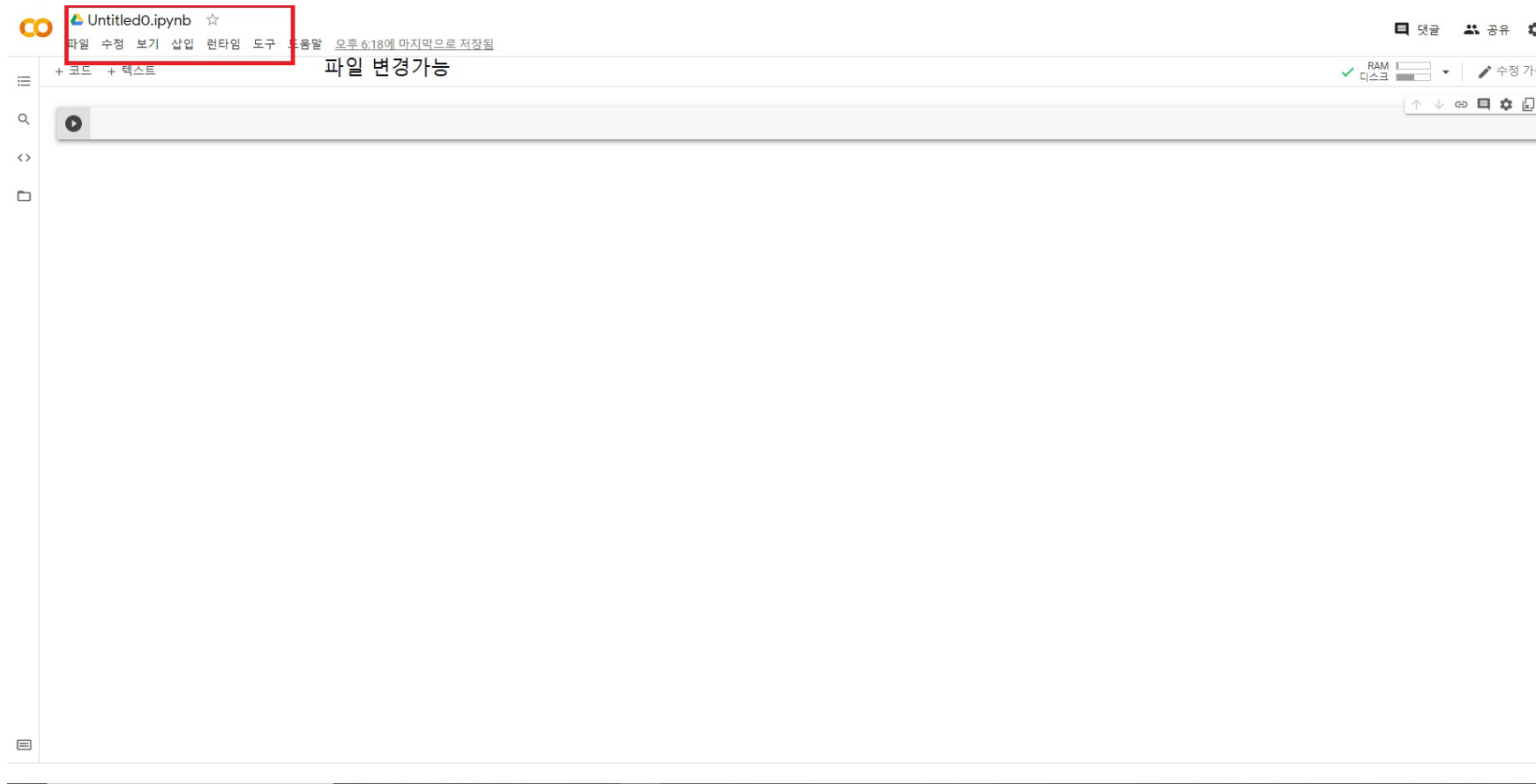


Source:

- 박스안이 “할당 중 → 연결중 → 초기화 중” 을 거쳐 아래와 같이 변경됩니다.



- 박스안을 더블클릭하여 파일 이름도 변경이 가능합니다.

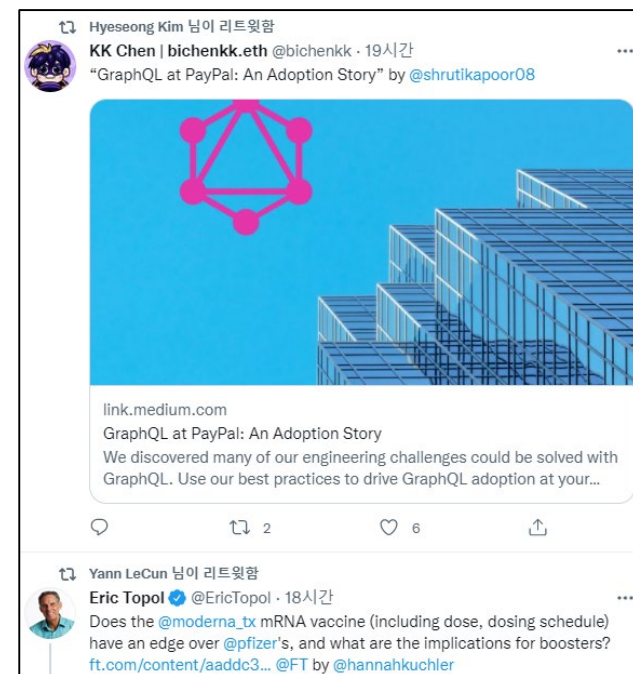


Source:

# Research

공부 해야하는데, 공부가 하기 싫을 때,  
죄책감 없이(?) 쉴 수 있습니다 :)

- Facebook
  - Tensorflow KR: <https://www.facebook.com/groups/TensorFlowKR/>
  - Pytorch KR: <https://www.facebook.com/groups/PyTorchKR>
- Twitter
  - <https://mobile.twitter.com/home>
- Reddit
  - <https://www.reddit.com/r/MachineLearning/>
- Arxiv Sanity
  - <http://www.arxiv-sanity.com/>



Source:



- ML/AI Conference and Journal
  - NeurIPS, ICML, ICLR, AAAI, AISTATS, JMLR, ...
    - ❖ <https://papers.nips.cc/paper/2021>
    - ❖ <https://icml.cc/Conferences/2021/Schedule?type=Poster>
    - ❖ <https://openreview.net/group?id=ICLR.cc/2021/Conference>
    - ❖ <https://aaai.org/Library/AAAI/aaai21contents.php>
    - ❖ <https://aistats.org/aistats2021/accepted.html>
    - ❖ <https://jmlr.org/papers/>
- NLP Conference and Journal
  - ACL, EMNLP, NAACL, TACL, ...
- Vision Conference and Journal
  - CVPR, ECCV, ICCV, TPAMI, ...
- Data Mining Conference
  - KDD, WWW, CIKM, ...
- 국내 논문
  - DBPIA: <https://www.dbpia.co.kr/>

좋은 논문을 많이 읽으시는 것을 추천합니다.

- Paper 구성
  - Abstract
  - Introduction
  - Related Works
  - Methodology
  - Experiments
  - Conclusion

## Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Example) Attention Is All You Need

<https://arxiv.org/pdf/1706.03762.pdf>

- Lecture
  - Stanford cs231n
    - ❖ <http://cs231n.stanford.edu/>
  - Stanford cs224n
    - ❖ <https://web.stanford.edu/class/cs224n>
  - Oxford with Deep Mind
    - ❖ <https://www.cs.ox.ac.uk/teaching/courses/2016-2017/dl/>
  - Stanford cs229
    - ❖ <http://cs229.stanford.edu/>
- Textbook
  - Murphy: <https://probml.github.io/pml-book/book1.html> (2판입니다)
  - Bishop: Pattern Recognition and Machine Learning
  - Tibshirani: <https://web.stanford.edu/~hastie/ElemStatLearn/>
  - Mathematics for Machine Learning: <https://mml-book.github.io/book/mml-book.pdf>