

# 파이썬 텐서플로 기반 딥러닝

컴퓨터정보공학과  
강 환수 교수

# 깃허브 주소

- <https://github.com/ai7dnn/2021-2-DNN>

Google JDK 15 Docume... 2장 첫 번째 자... java.util.function... 11\_Colab\_Basics... 로그인 : edwith CS 230 - 답 러... Counting No. of...

Search or jump to... Pull requests Issues Marketplace Explore

ai7dnn / 2021-2-DNN Unwatch 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

ai7dnn Update README.md 7843ff8 5 minutes ago 4 commits

code	Colaboratory를 통해 생성됨	20 hours ago
README.md	Update README.md	5 minutes ago
[01주 텐서플로 기반 답러닝] 쉬운 인...	Add files via upload	7 minutes ago
[01주 텐서플로 기반 답러닝] 쉬운 인...	Add files via upload	7 minutes ago

README.md

## 2021-2-DNN

2021학년도 2학기 답러닝

About  
No description, website, or provided.  
Readme

Releases  
No releases published  
[Create a new release](#)

Packages  
No packages published  
[Publish your first package](#)

Languages  
Jupyter Notebook 100.0%

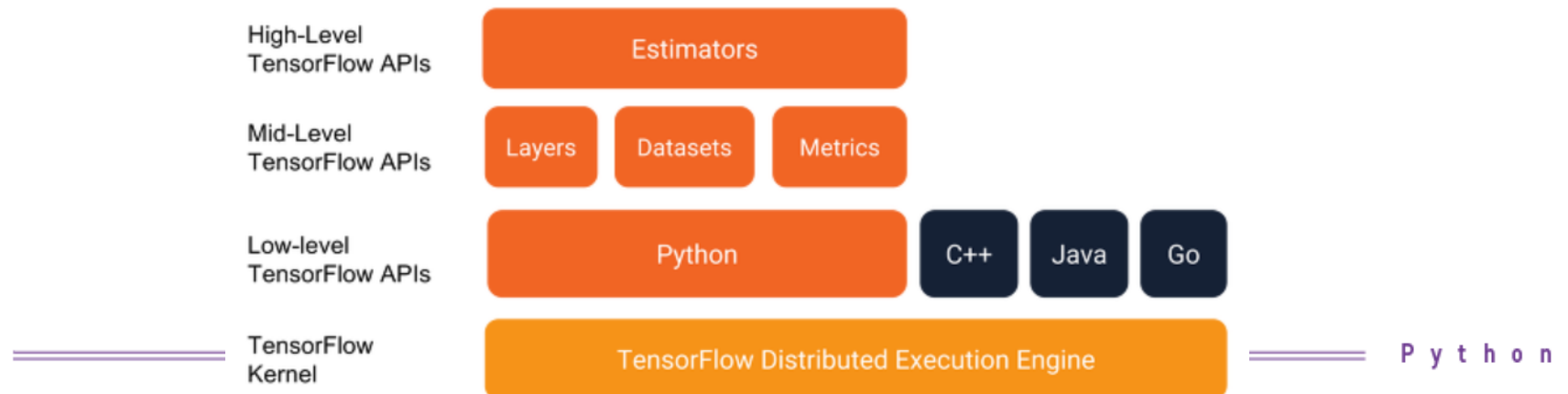


# 텐서플로 개요

참고 <https://excelsior-cjh.tistory.com/148> [EXCELSIOR]

# 텐서플로(TensorFlow) 개요

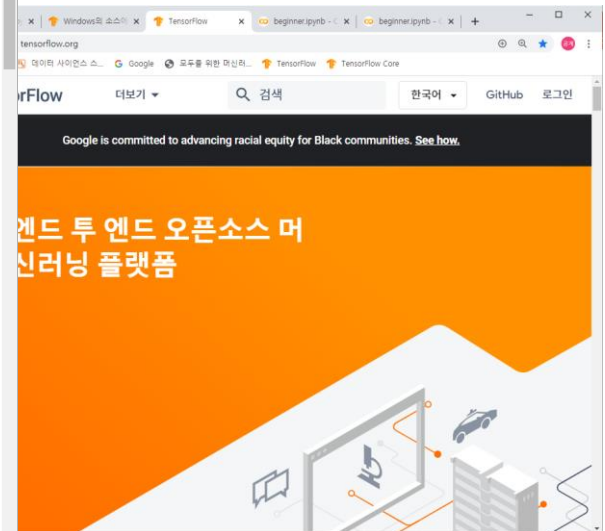
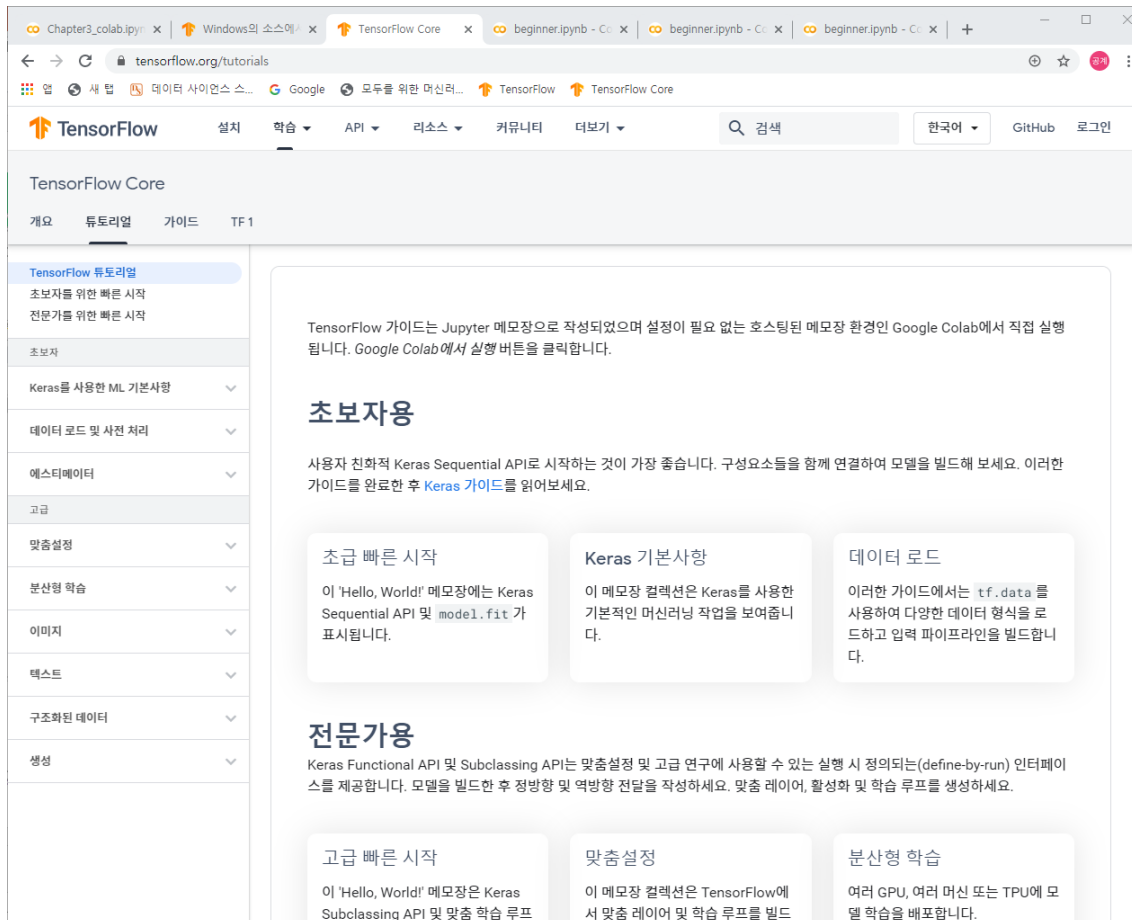
- 구글(Google)에서 만든 라이브러리
  - [www.tensorflow.org](http://www.tensorflow.org)
  - 딥러닝 프로그램을 쉽게 구현할 수 있도록 다양한 기능을 제공
    - 초보자 및 전문가에게 데스크톱, 모바일, 웹, 클라우드 개발용 API를 제공
    - 연구 및 프로덕션용 오픈소스 딥러닝 라이브러리
- 구현 및 사용
  - 텐서플로 자체는 기본적으로 C++로 구현
    - Python, Java, Go 등 다양한 언어를 지원
  - 파이썬을 최우선으로 지원
    - 대부분의 편한 기능들이 파이썬 라이브러리로만 구현되어 있어 Python에서 개발하는 것이 편함



# 텐서플로 홈페이지

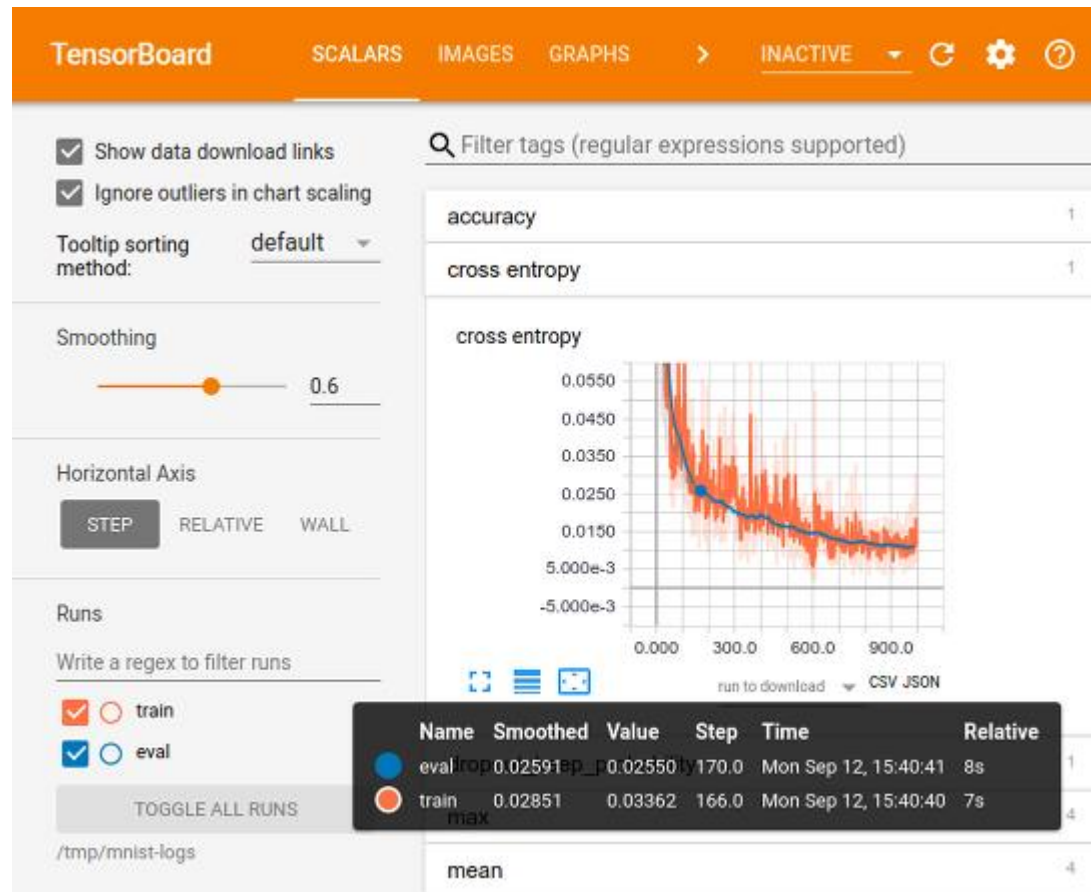
## • 튜토리얼

— <https://www.tensorflow.org/tutorials>



# 텐서보드(TensorBoard)

- 브라우저에서 실행 가능한 시각화 도우미
  - 딥러닝 학습 과정을 추적하는데 유용하게 사용



# 텐서 개요

- **Tensor(텐서): 모든 데이터**
  - 딥러닝에서 데이터를 표현하는 방식
    - 0-D 텐서 : 스칼라
    - 1-D 텐서 : 벡터
    - 2-D 텐서 : 행렬 등
  - n차원 행렬(배열)
    - 텐서는 행렬로 표현할 수 있는 n차원 형태의 배열을 높은 차원으로 확장

Scalar    Vector    Matrix    Tensor

1

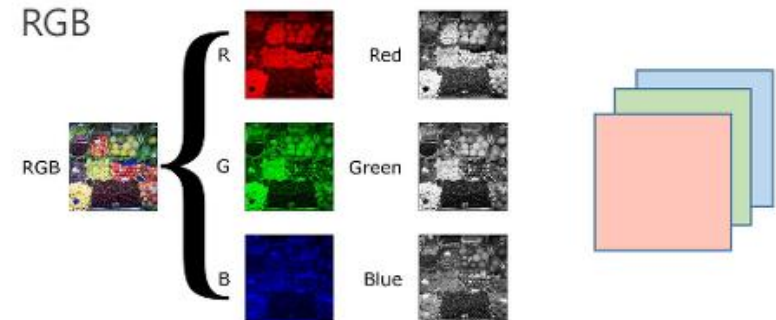
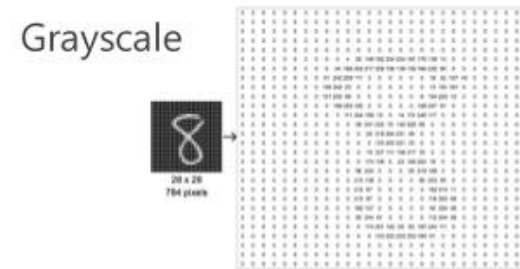
$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

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

$$\begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

# 텐서의 사례

- 스칼라: 차원이 없는 텐서
  - 10
- 벡터 값: 1차원 텐서
  - [10, 20, 30]
- 2차원 행렬: 2차원 텐서
  - 회색조( grayscale ) 이미지
    - 하나의 채널(channel)에 2차원 행렬(배열)로 표현
  - [[1, 2, 3], [4, 5, 6]]
- 텐서: n차원 행렬
  - 텐서의 차원을 텐서의 rank(순위)라 함
  - RGB 이미지
    - R(ed), G(reen), B(lue) 각 3개의 채널마다 2차원 행렬(배열)로 표현하는데, 이를 텐서(3차원의 값을 가지는 배열)로 표현
  - [[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]]





# TensorFlow 계산 과정

- TensorFlow에서 텐서(Tensor) 계산 과정

- 모두 그래프(Graph)라고 부르는 객체 내에 저장되어 실행
- 그래프를 계산하려면 외부 컴퓨터에 이 그래프 정보를 전달하고 그 결과값을 받아야 함

- Session

- 이 통신과정을 담당하는 것이 세션(Session)이라고 부르는 객체
- 생성, 사용, 종료 과정이 필요
- 세션 생성
  - Session 객체 생성
- 세션 사용
  - run 메서드에 그래프를 입력하면 출력 값을 계산하여 반환
- 세션 종료
  - close 메서드.
  - with 문을 사용하면 명시적으로 호출 불필요

```
x = tf.constant(3)
y = x**2
```

```
sess = tf.Session()
print(sess.run(x))
print(sess.run(y))
sess.close()
```

# 데이터 흐름 그래프(dataflow graph)

## TensorFlow에서 계산

- 데이터 흐름 그래프 (dataflow graph)로 이루어짐
- 텐서 형태의 데이터들이 딥러닝 모델을 구성하는 연산들의 그래프를 따라 흐르면서 연산이 일어남

## Tensor + DataFlow

- 딥러닝에서 데이터를 의미하는 Tensor 와 DataFlow Graph를 따라 연산이 수행되는 형태 (Flow)를 합쳐 TensorFlow란 이름이 나오게 됨

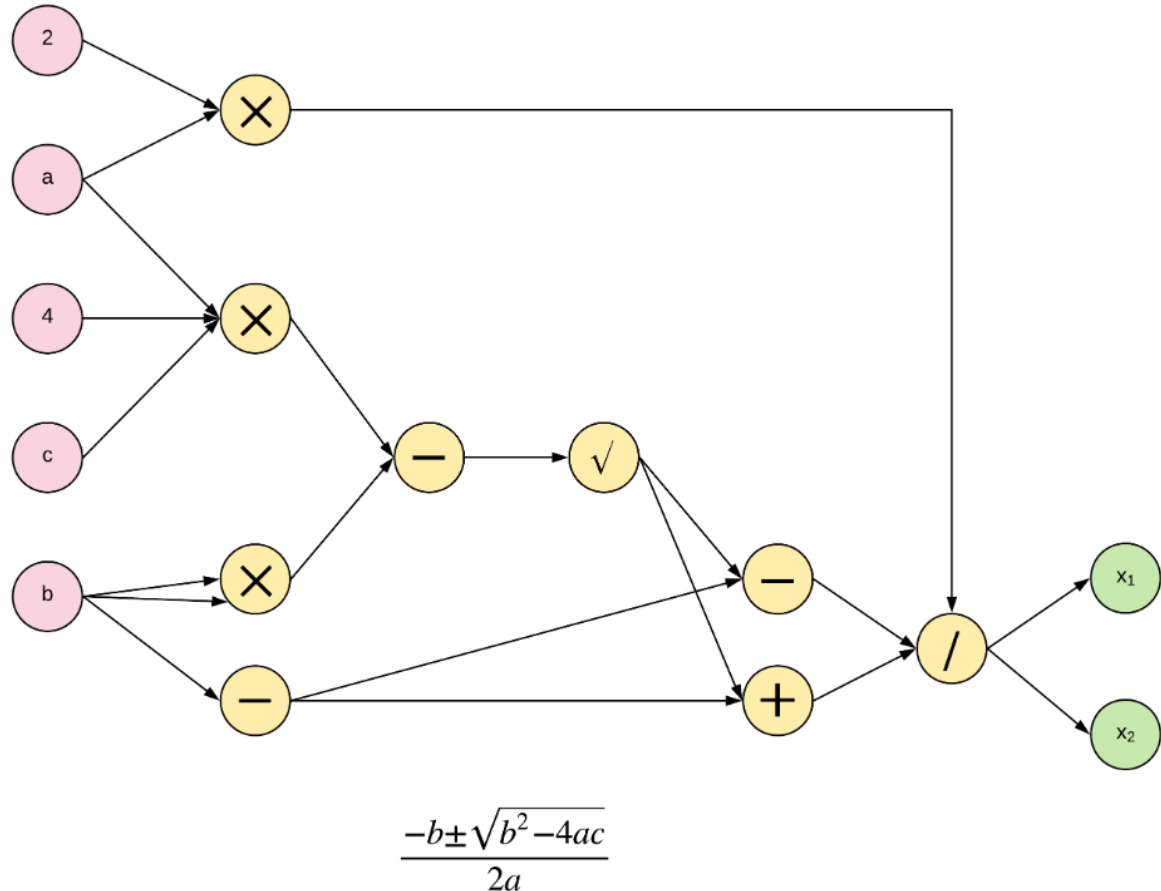


그림 1: 2 차 공식을 사용하여 2 차 표현식의 근을 계산하기위한 계산 그래프.

# TensorFlow API 탐색

## TensorFlow API 계층

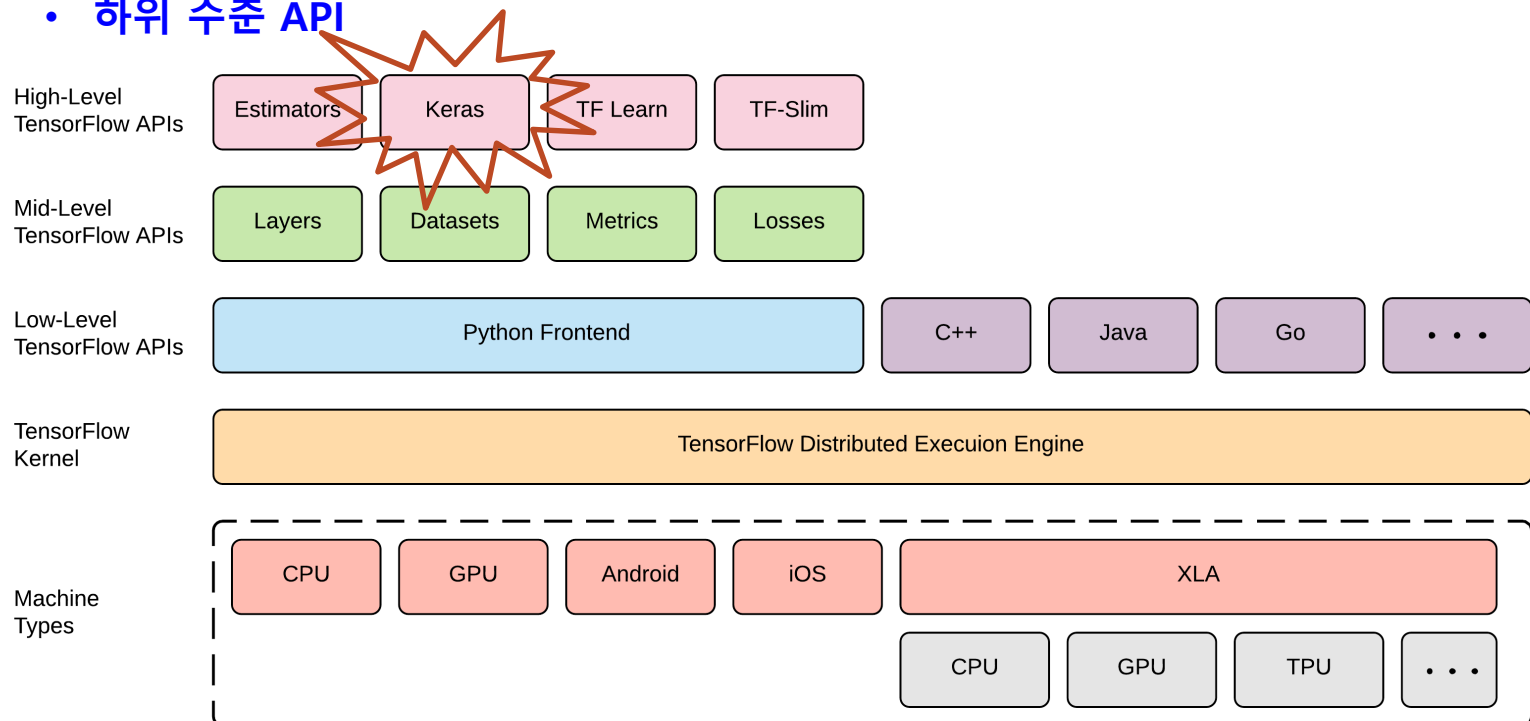
TensorFlow 딥 러닝 모델 구축 작업은 서로 다른 API 수준을 사용하여 해결

### 고급 API

- Keras나 TF-Slim 과 같은 추상화 라이브러리를 제공
- 저수준 텐서플로 라이브러리에 대해 손쉽게 고수준 접근이 가능하게 해줌

### 중급 API

### 하위 수준 API



# 텐서플로 vs 케라스

- 케라스, 텐서플로 뭐가 좋아요?

**If you're asking "*Keras or TensorFlow?*"**



**VS.**



**Then you're asking the *wrong* question (and here's why...)**

# Tensor Ranks, Shapes, and Types(1)

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

Rank	Math entity	Python example
0	Scalar (magnitude only)	<code>s = 483</code>
1	Vector (magnitude and direction)	<code>v = [1.1, 2.2, 3.3]</code>
2	Matrix (table of numbers)	<code>m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]</code>
3	3-Tensor (cube of numbers)	<code>t = [[[2], [4], [6]], [[8], [10], [12]], [[14], [16], [18]]]</code>
n	n-Tensor (you get the idea)	<code>....</code>

[https://www.tensorflow.org/programmers\\_guide/dims\\_types](https://www.tensorflow.org/programmers_guide/dims_types)

# Tensor Ranks, Shapes, and Types(2)

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

Data type	Python type	Description
DT_FLOAT	<code>tf.float32</code>	32 bits floating point.
DT_DOUBLE	<code>tf.float64</code>	64 bits floating point.
DT_INT8	<code>tf.int8</code>	8 bits signed integer.
DT_INT16	<code>tf.int16</code>	16 bits signed integer.
DT_INT32	<code>tf.int32</code>	32 bits signed integer.
DT_INT64	<code>tf.int64</code>	64 bits signed integer.

<https://www.quora.com/When-should-I-use-tf-float32-vs-tf-float64-in-TensorFlow>

# 딥러닝 참고 사이트

- 텐서플로
  - <https://www.tensorflow.org/>
- 머신러닝 단기집중과정
  - <https://developers.google.com/machine-learning/crash-course>
- Naver D2
  - <https://d2.naver.com/home>
- Naver Tech Talks
  - <https://d2.naver.com/news/2657726>
- Naver edwith 인공지능
  - <https://www.edwith.org/search/index?categoryId=71>
- 논문으로 짚어보는 딥러닝의 맥
  - <https://www.edwith.org/deeplearningchoi>
- 모두를 위한 머신러닝/딥러닝(성김 교수)
  - <https://hunkim.github.io/ml/>
- 모두를 위한 딥러닝 시즌 2
  - <https://deeplearningzerotoall.github.io/season2/>

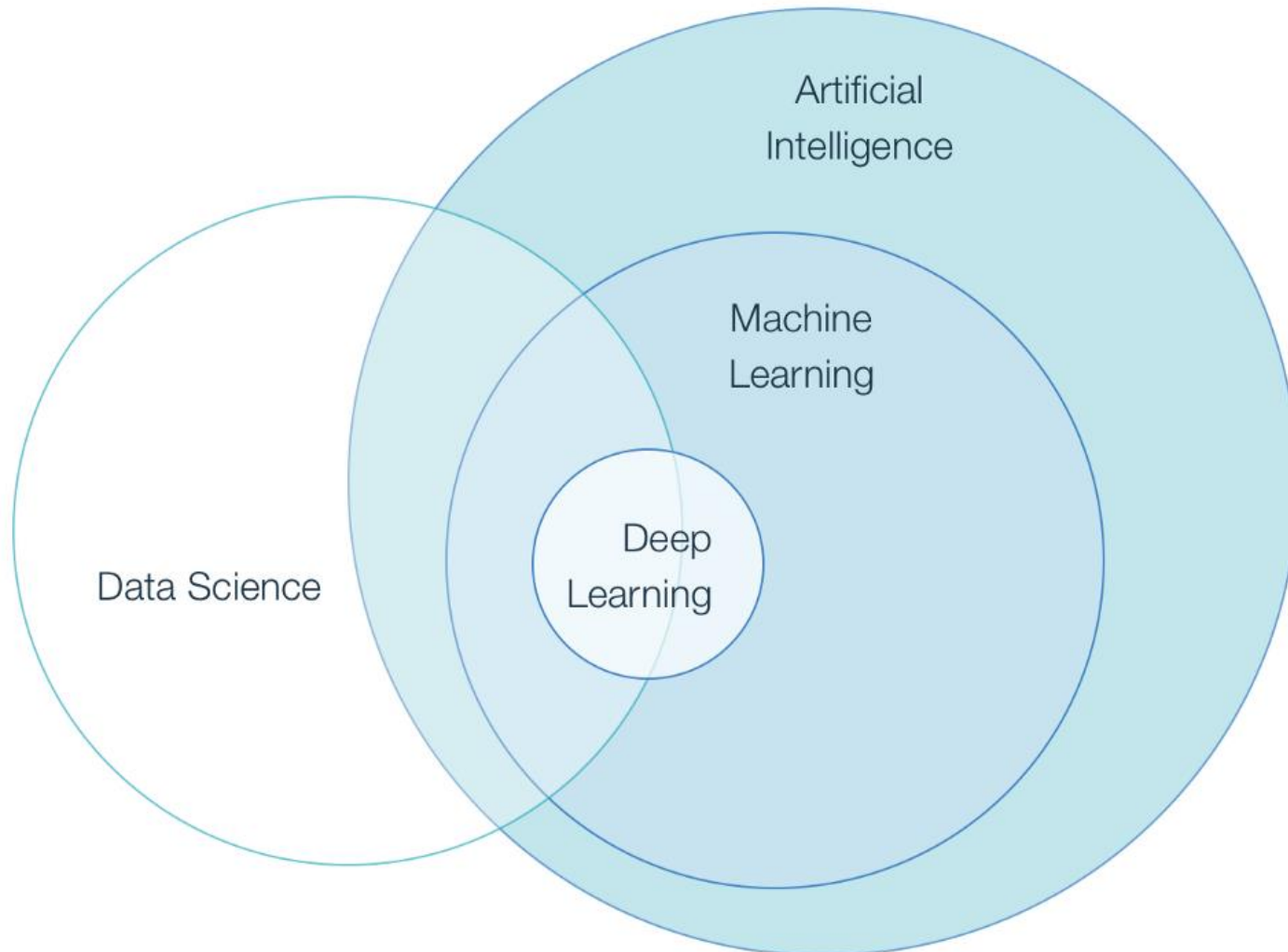
# 딥러닝 이해 영상

- 딥러닝 영상
  - <https://www.youtube.com/watch?v=aircAruvnKk>



# 코랩으로 시작하는 텐서플로 기초 프로그래밍

# Data Science & AI

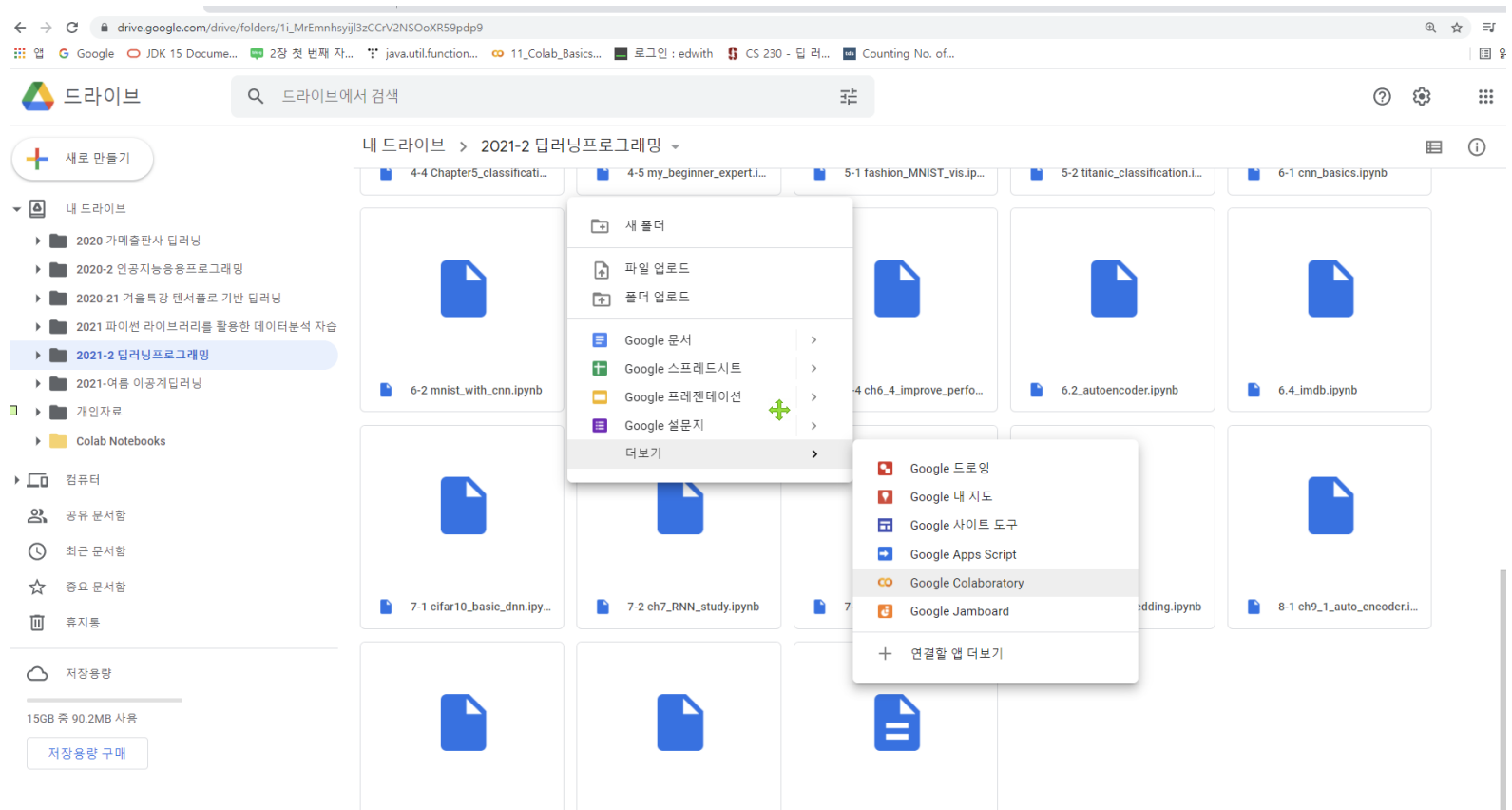


# 텐서플로

- **Tensorflow**
  - 머신러닝을 위한 오픈소스 플랫폼
    - 가장 널리 쓰이는 딥러닝 프레임워크
  - 지원 언어
    - 파이썬, C++, 자바스크립트, 자바, Go, Swift
- **버전 변화**
  - 2017, 1월: 1.0 알파
  - 2019, 3월: 2.0 알파
  - 2019, 9월: 2.0 정식
    - 즉시 실행 모드(eager execution)가 기본
    - 세션 대신 함수 사용
    - 정식으로 TPU 지원
    - 최신 버전 2.6
- **홈페이지**
  - <http://tensorflow.org>

# 코랩 드라이브에서 노트북 파일 열기

- 자신의 구글 드라이브 적당한 폴더에 저장
  - 파일 21-1-myfirst-tf.ipynb



# 실습 파일

- 21-1-myfirst-tf.ipynb

The screenshot shows a GitHub repository page for 'ai7dnn / 2021-2-DNN'. The file '21\_1\_myfirst\_tf.ipynb' is selected, showing its content in a Jupyter Notebook format. The notebook contains three code cells, each with an input prompt and an output.

Repository details: ai7dnn / 2021-2-DNN. Latest commit 37616f8 20 hours ago. 1 contributor.

File details: 633 lines (633 sloc) | 14.1 KB. Options: Open in Colab, Raw, Blame.

```

In [1]: import tensorflow as tf
         tf.__version__
Out[1]: '2.6.0'

In [2]: import sklearn as sk
         sk.__version__
Out[2]: '0.22.2.post1'

In [3]: import site
         site.getsitepackages()
Out[3]: ['/usr/local/lib/python3.7/dist-packages',
         '/usr/lib/python3/dist-packages']
  
```

# Tensorflow 불러오기

- `import tensorflow as tf`
  - `tf.__version__`
    - 현재는 자동으로 2.6.0

# 매직 명령어로 버전 바꾸기

- 코랩에서 쉽게 버전 사용방법

- %tensorflow\_version 1.x
- %tensorflow\_version 2.x

- Import 하기 전

- 위 매직 명령어 사용
- 사용 중에 바꾸려면 '런타임 다시 시작' 후 바로
  - %tensorflow\_version 1.x
  - %tensorflow\_version 2.x

- 2.2 사용 중에 1.x으로 변경

- 1. 메뉴, 런타임 | 런타임 다시 시작
  - 단축키: **ctrl+M** .
- 2. 바로 실행
  - %tensorflow\_version 1.x

```
[1] 1 import tensorflow as tf
```

```
[2] 1 tf.__version__
```

```
↳ '2.2.0'
```

```
[3] 1 # 3.4 텐서플로 2.0 버전 선택
2 try:
3     # %tensorflow_version only exists in Colab.
4     %tensorflow_version 1.x
5 except Exception:
6     pass
7 import tensorflow as tf
8 tf.__version__
```

```
↳ TensorFlow is already loaded. Please restart the runtime to change versions.
'2.2.0'
```

```
[1] 1 # 3.4 텐서플로 2.0 버전 선택
2 try:
3     # %tensorflow_version only exists in Colab.
4     %tensorflow_version 1.x
5 except Exception:
6     pass
7
8 import tensorflow as tf
9 tf.__version__
```

```
↳ TensorFlow 1.x selected.
'1.15.2'
```

# 텐서플로 2.0으로 실행

- 매직 명령어 사용
  - 또는 그대로 import
- 기본이 즉시 실행

```
[1] 1 # 텐서플로 2.0 버전 선택
    2 try:
    3     # %tensorflow_version only exists in Colab.
    4     %tensorflow_version 2.x
    5 except Exception:
    6     pass
    7
    8 import tensorflow as tf
    9 tf.__version__
```

↪ '2.2.0'

```
[6] 1 tf.executing_eagerly()
```

↪ True



# 좀 더 쉬운 2.0

- **텐서 출력**
  - 첫 항목에 내용
  - 값만 보려면
    - **메소드 numpy()**

```
[7] 1 import tensorflow as tf
    2
    3 hello = tf.constant('Hello World!')
    4 print(hello)
    5 print(hello.numpy())
```

```
↳ tf.Tensor(b'Hello World!', shape=(), dtype=string)
   b'Hello World!'
```

```
[8] 1 a = tf.constant(5)
    2 b = tf.constant(3)
    3 print(a+b)
    4 print((a+b).numpy())
```

```
↳ tf.Tensor(8, shape=(), dtype=int32)
   8
```

```
[9] 1 c = tf.constant('Hello, world!')
    2
    3 with tf.Session() as sess:
    4     print(sess.run(c))
```

```
↳ -----
AttributeError                                Traceback (most recent call last)
<ipython-input-9-7a2fa5301d92> in <module>()
      1 c = tf.constant('Hello, world!')
      2
----> 3 with tf.Session() as sess:
      4     print(sess.run(c))
```

AttributeError: module 'tensorflow' has no attribute 'Session'