

손쉽게 개발환경 만들기

DEEP LEARNING AND COMPUTER VISION

02

APPLICATION

ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE (AI)
REFERS TO THE SIMULATION OF THE
HUMAN INTELLIGENCE IN COMPUTERS
AND THE PROGRAMS TO THESE
DATA ANALYSIS AND RECOGNITION
SYSTEMS.

The term may also be applied
to any machine that performs
tasks associated with a human,
such as learning and
problem-solving.

Artificial intelligence (AI) refers
to the simulation of human
intelligence.

Notice

Artificial intelligence (AI) refers
to the simulation of human
intelligence.

ARTIFICIAL INTELLIGENCE (AI)
REFERS TO THE SIMULATION OF
HUMAN INTELLIGENCE IN COMPUTERS
AND THE PROGRAMS TO THESE
DATA ANALYSIS AND RECOGNITION
SYSTEMS.

이 교육과정은 교육부 '성인학습자 역량 강화 교육콘텐츠 개발' 사업의 일환으로써
교육부로부터 예산을 지원 받아 고려사이버대학교가 개발하여 운영하고 있습니다.
제공하는 강좌 및 학습에 따르는 모든 산출물의 저작권은 교육부, 한국교육학술정보원,
한국원격대학협의회와 고려사이버대학교가 공동 소유하고 있습니다.

생각해보기

다차원 배열의 데이터 정의 방식에 대해 생각해봅시다.

각각 다른 크기와 무게 속성을 가진 과일 3개의 정의

```
# 개수: 3개, 크기: 1, 2, 3, 무게: 10, 20, 30
```

```
# 첫 번째 정의 방식
```

```
fruit_type1 = [ [ 1, 2, 3], [ 10, 20, 30]]
```

```
# 두 번째 정의 방식
```

```
fruit_type2 = [ [1, 10], [2, 20], [3, 30]]
```

- 두 가지 정의 중에서 사용해야 한다면 어느 쪽을 사용하시겠습니까?
- 두 정의의 장단점에 대해 생각해 봅시다.

학습목표

Artificial Intelligence (AI) refers
to the simulation of human

GOALS

Artificial Intelligence has
helped to the simulation of
human intelligence in various
fields and contributed to their
useful human and social value
addition.

The technology will be applied
to various fields and contribute
to the development of a smart
society and the realization of
human dreams.

- 1 텐서의 개념을 설명할 수 있다.
- 2 구글 코랩을 사용할 수 있다.
- 3 PC에서 Jupyter등의 개발 환경을 구성할 수 있다.

- 1 Keras-Tensorflow 소개
- 2 인터넷만 되면
무료로 쓸 수 있는 구글 코랩
- 3 내 PC에서 직접 돌려보고 싶은 사람을 위한
개발 환경 만들기

학습내용

APPLICATION

1차시

Keras-Tensorflow 소개

가장 많이 쓰이는 Deep Learning Framework

01 Keras-Tensorflow 소개

01 텐서의 정의

다차원, 선형성, 변환 등에 주로 쓰임

수학

벡터공간에서 정의되는 다차원 선형 대수 객체

물리학

좌표 변환에 대해 선형 변환하는 다차원 객체

머신러닝에서는 훨씬 단순한 정의를 사용

머신러닝

숫자 객체(**numpy**)의 다차원 배열



다차원 행렬

01 텐서의 정의 (Rank 1 Tensor)



```
import numpy as np
# 딸기0의 크기
x = np.array( [ 1.0 ] )
print(x.shape)
출력: (1,)
```

한 개의 특징벡터이므로 shape은 (1)



$x[0]$

01 텐서의 정의 (속성 추가)



```
# 딸기0의 [ 크기, 무게 ]
x = np.array( [ 1.0, 1.1 ] )
print(x.shape, x[0].shape)
출력: (2,) ()
```

두 개의 특징벡터이므로 shape은 (2)



$x[0] = \text{크기}$

$x[1] = \text{무게}$

01 텐서의 정의 (Rank 2 Tensor)



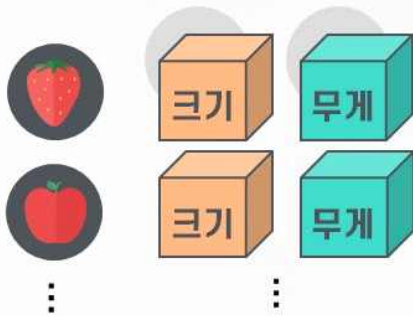
과일0, 과일1, .. , 과일99의 [크기, 무게]

```
x = np.array([[ 1.0, 1.1 ], [ 2.0, 2.1 ], ..., [ 1.0, 1.1 ]])
```

```
print(x.shape, x[0].shape, x[0,0].shape)
```

출력: (100,2,) (2,) ()

→ shape은 (100,2,) → 두 개의 특징 벡터가 100개



$x[0] == 0$ 번째 과일의 특징 [크기, 무게]

$x[35,0] == 35$ 번째 과일의 무게

01 텐서의 정의 (Tensor for category)



과일0, 과일1, .. , 과일99의 종류

```
y = np.array([[ 1, 0 ], [ 0, 1 ], ..., [ 0, 1 ]])
```

→ shape은 (100,2,) → 두 개의 카테고리가 100개



$y[0] == 0$ 번째 과일의 분류 == [1,0] (딸기)



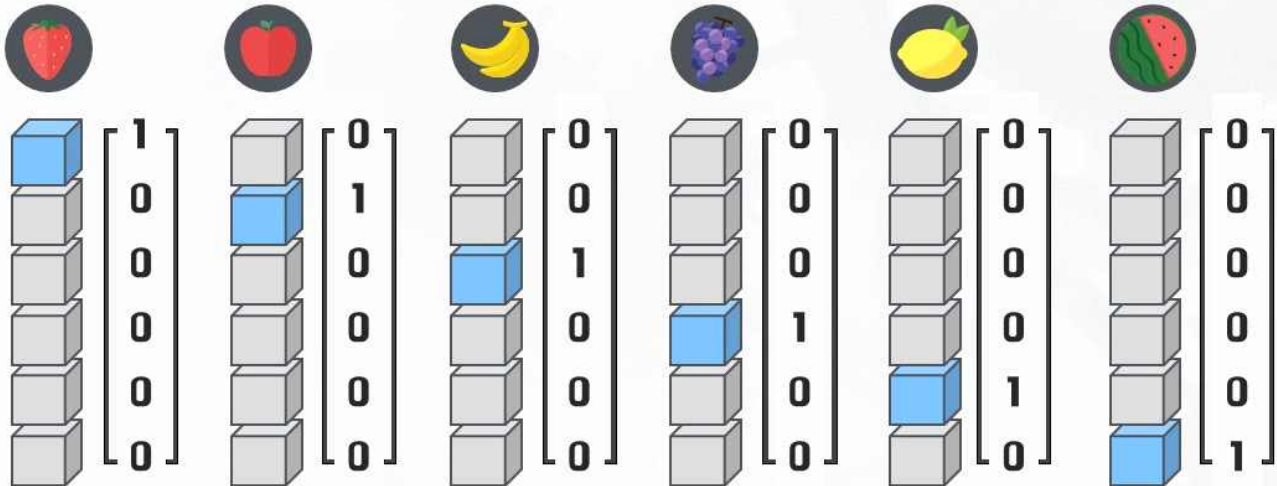
$y[1] == 1$ 번째 과일의 분류 == [0,1] (사과)

⋮

01 텐서의 정의 (Category 확장)



분류가 6개인 경우의 시각화



02 다차원 배열과 텐서의 차이

다차원 배열

```

y = [
    [ 1.0, 1.1 ],
    [ 1.0, 3, 2.0 ],
]
print(y*2) # y+y와 같음
[[1.0, 1.1], [1.0, 3, 2.0], [1.0, 1.1],
[1.0, 3, 2.0]]
print(y**2) # error

```

비정형화된 타입에도 많이 사용
대량 수치 연산이 없음

텐서

```

x = np.array([
    [ 1.0, 1.1 ],
    [ 2.2, 1.0 ]
])
print(x*2)
[[2.  2.2]
 [4.4 2.  ]]
print(x**2) # OK: 제곱연산 수행

```

정형화된 타입을 주로 사용
대량 수치 연산 가능

03 머신러닝의 특징 (Classical Program)



03 머신러닝의 특징 (Machine Learning)

학습 단계



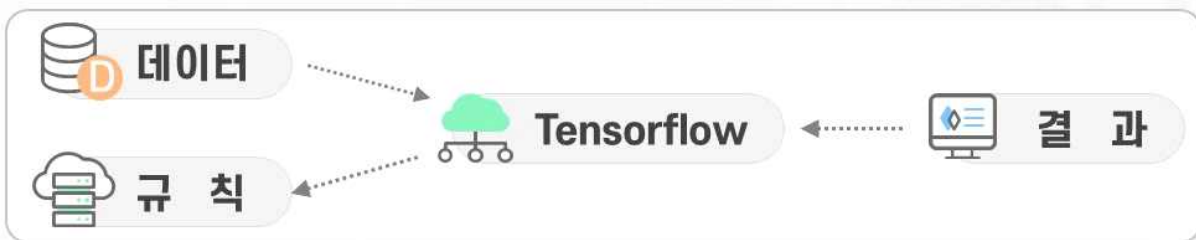
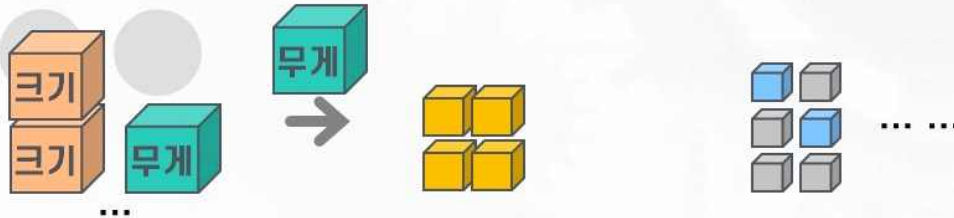
실행 단계



04 Tensorflow란?



정의 입력 Tensor(데이터)와 결과 Tensor사이의 흐름(flow)



04 Tensorflow란? (Keras vs. Tensorflow 1)

High Level

K Keras

TensorFlow theano
Microsoft CNTK

Flexible & High Performance

TensorFlow

Python C/C++

Since 2015

01 Keras-Tensorflow 소개

04 Tensorflow란? (Keras == Tensorflow 2)



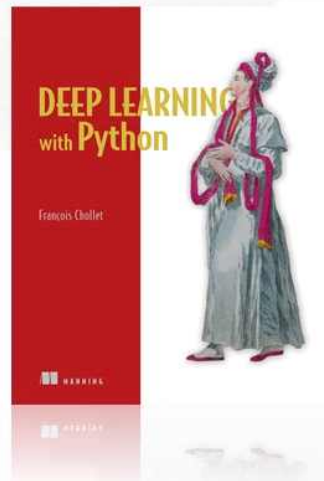
Keras는 Tensorflow의 공식 상위 API가 됨

Keras 최초개발자 :
François Chollet

Keras의 개발자이자 구글에서 텐서플로를 공동 개발



Since 2019



이미지 출처 : <https://livebook.manning.com/book/deep-learning-with-python/about-this-book/>

01 Keras-Tensorflow 소개

05 머신러닝 (입문자를 위한) 팁



Simply use Keras!

❖ Life is short.



Run away from Tensorflow 1.x.

❖ `tf.placeholder`, `tf.Session.run()`

❖ Still lots of 1.x codes there!

APPLICATION

2차시

인터넷만 되면 무료로 쓸 수 있는 구글 코랩

웹기반 Jupyter Notebook + GPU 지원

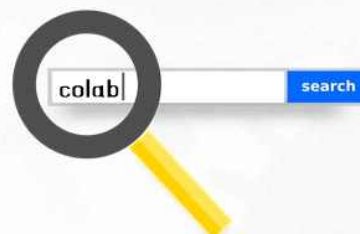
02 인터넷만 되면 무료로 쓸 수 있는 구글 코랩

01 구글 코랩이란?



웹기반(Jupyter) 파이썬 머신러닝 학습 도구

- ❖ 무료 (구글 로그인)!
- ❖ 런타임 당 12시간까지 GPU 사용 가능!
- ❖ 웹기반: <https://colab.research.google.com>



02 인터넷만 되면 무료로 쓸 수 있는 구글 코랩

Artificial intelligence (AI) relies on the collection of human

01 구글 코랩이란?



웹기반(Jupyter) 파이썬 머신러닝 학습 도구



02 인터넷만 되면 무료로 쓸 수 있는 구글 코랩

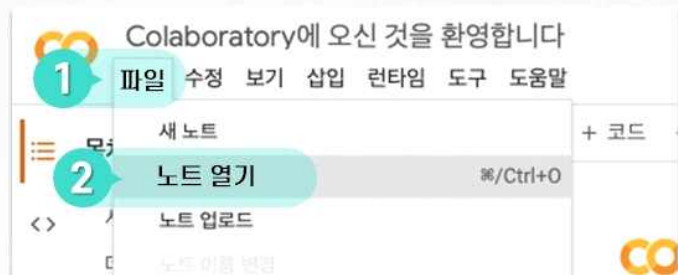
Artificial intelligence (AI) relies on the collection of human

01 구글 코랩이란?



코랩으로 노트북 파일 읽어들이기

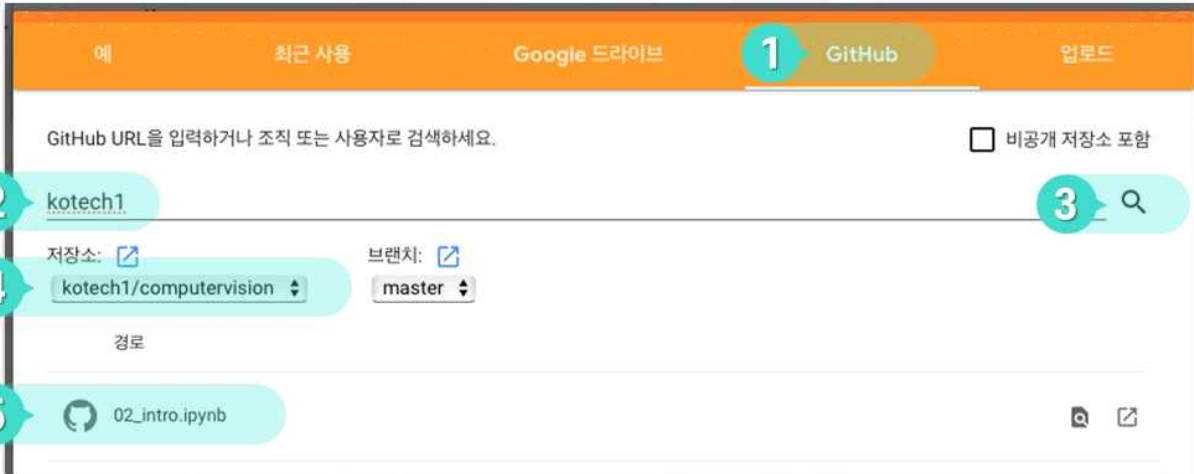
- 1 파일 메뉴 ➡ 노트 열기
- 2 GitHub선택 ➡ 사용자: kotech1 ➡ 검색 아이콘 클릭
- 3 저장소: kotech1/computervision 선택
- 4 파일: 02_intro.ipynb 클릭



01 구글 코랩이란?



코랩으로 노트북 파일 읽어들이기



01 구글 코랩이란?



코랩으로 노트북 파일 읽어들이기

- 1 파일 메뉴 ➡ 노트 열기
- 2 GitHub선택 ➡ 사용자: kotech1 ➡ 검색 아이콘 클릭
- 3 저장소: kotech1/computervision 선택
- 4 파일: 02_intro.ipynb 클릭
- 5 파일 메뉴 ➡ 드라이브에 사본 저장



다음부터는
[파일 ➡ 노트 열기 ➡ 구글 드라이브]
에서 내 노트를 불러올 수 있다!

02 Welcome to Python!



코드 실행 방법

- ▶ 클릭
- ctrl-Enter 혹은 shift-Enter

```
print('hello, world.')
```



실행

실행순서

[1] print('hello, world.')

hello, world.

출력

셀 배치 변경

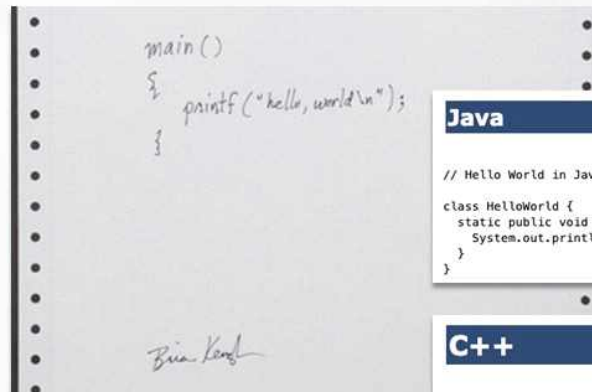
↑ ↓ ↻ ⚙️ 🗑️ ⋮

출력 지우기 등

셀 삭제

02 Welcome to Python!

“hello, world”,
누가 처음 쓰기 시작했을까?



C언어 개발자인 Kernighan의
1978년 C교재에서 사용

Java

```
// Hello World in Java
class HelloWorld {
    static public void main( String args[] ) {
        System.out.println( "Hello World!" );
    }
}
```

C++

```
// Hello World in C++ (pre-ISO)
#include <iostream.h>

main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

02 Welcome to Python!



Python 이스터에그

- ❖ The Zen of python

```
import this
```



The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.

...

Python의 모토

단순한 것이 복잡한 것보다 낫다

03 Tensorflow



텐서플로우 버전 확인

```
import tensorflow as tf  
tf.__version__
```



'2.2.0'

2.x 버전 사용

03 Tensorflow



라이브러리 확인

```
!pip list
!pip install <some package>==<version>
```



```
Package Version
absl-py 0.9.0
alabaster 0.7.12
...
```

03 Tensorflow



텐서 만들기

❖ Numpy 배열 만들기

```
import numpy as np
x = np.linspace(-1, 1, num=21)
x
```

-1에서 1까지 모두 21개의 데이터를 생성



```
array([-1. , -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0. ,
        0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ])
```


03 Tensorflow



텐서로부터 텐서 만들기

❖ $y = x^2$ 데이터 생성

```
y = x**2
y
```

Vectorized

➤ `array([1. , 0.81, 0.64, 0.49, 0.36, 0.25, 0.16, 0.09, 0.04, 0.01, 0. ,
0.01, 0.04, 0.09, 0.16, 0.25, 0.36, 0.49, 0.64, 0.81, 1.])`

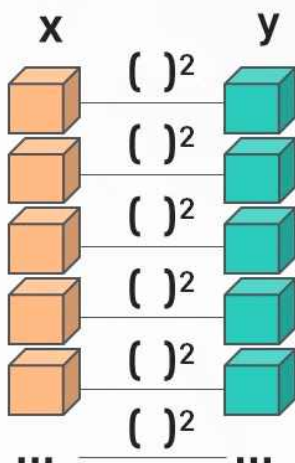


03 Tensorflow



입출력 시각화

❖ $y = x^2$ 데이터 흐름 시각화



단순 연산의 경우 shape은 같음

- x의 shape이 (21)이므로,
- y의 shape도 (21)이다.

```
print(x.shape, y.shape)
```

➤ (21,) (21,)

03 Tensorflow

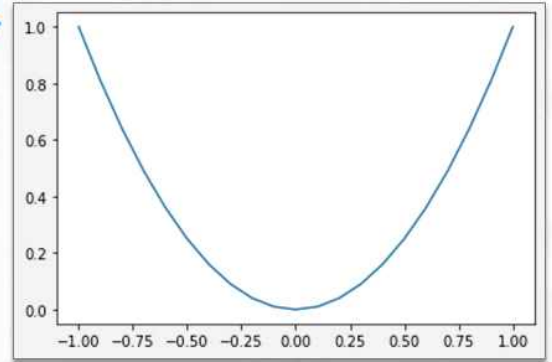


matplotlib로 손쉽게 그래프 그리기

❖ $y = x^2$ 를 그래프로 그리기

```
from matplotlib import pyplot as plt
plt.plot(x, y, '-')
plt.show()
```

x, y점들을 연결하는 그래프
('-'는 상형문자)



03 Tensorflow

일반 연산

```
n = 100000
x = np.random.random(n).tolist()
tic = time.time()
y = []
for i in range(0, n, 1):
    y.append(x[i]*x[i])
toc = time.time()
print("list time =",
      (toc-tic)*1000, "ms")
```

➤ list time = 23.8985271453857422 ms

텐서 연산

```
n = 100000
x = np.random.random(n)
tic = time.time()
y = x**2
toc = time.time()
print("tensor time =",
      (toc-tic)*1000, "ms")
```

➤ tensor time = 1.16546249389648 ms

03 Tensorflow



인터넷 이미지 다운로드 및 표시

- 인터넷에서 이미지를 다운로드하고 matplotlib로 그려 보기

```
from tensorflow import keras
import cv2

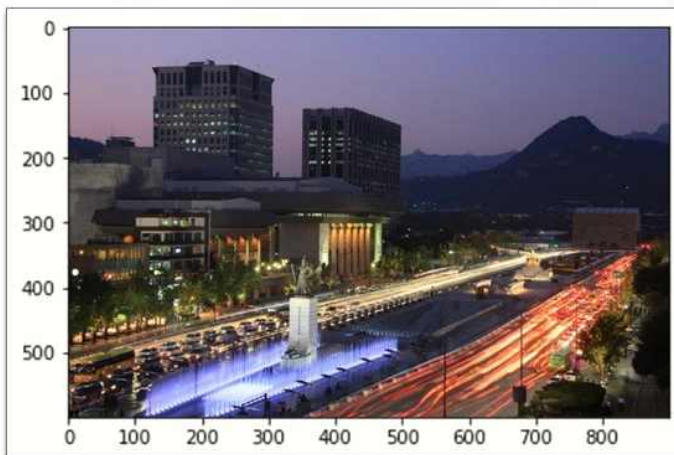
image_path = keras.utils.get_file("seoul.jpg",
    "http://data.si.re.kr/photo_download/21555")
image = cv2.imread(image_path)
# convert opencv RGB for opencv
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
plt.show()
```

03 Tensorflow



인터넷 이미지 다운로드 및 표시

- 결과 출력



본 저작물은 '서울시 사진기록화사업 2010'에서 '2009년'작성하여
공공누리 제1유형으로 개방한 '광화문광장'을 이용하였으며, 해당 저작물은
'서울연구데이터서비스, <http://data.si.re.kr/collection/view/360>'에서
무료로 다운로드할 수 있습니다.

APPLICATION

3차시

내 PC에서 직접 돌려 보고 싶은 사람을 위한 개발환경 만들기

윈도우와 맥을 중심으로 한 Local 개발 환경



실습 PRACTICE

ARTIFICIAL INTELLIGENCE
REFERS TO THE SIMULATION OF
HUMAN INTELLIGENCE IN MACHINES
THAT ARE PROGRAMMED TO LEARN
AND REASON AND TAKE
ACTIONS.

This part aims to explore
the deep learning framework
and understand the basic
concepts of the framework and
the basic concepts.

개발 환경 만들기

Local 개발 환경 구축의 필요성

- IDE(Integrated Development Environment)를 이용한 본격 개발
PyCharm이나 Visual Studio Code 등
- 보안성이 요구되는 환경
인트라넷 만 이용가능할 경우, 학습데이터의 보안이 필요할 경우
- GPU 및 리소스 활용
Local PC의 GPU 활용, 대용량 SSD 등 로컬 리소스 활용

윈도우에서 개발환경 설치

- 사전 요구 사항
윈도우 10 64bit, 3GB이상의 저장공간 여유
- 아나콘다란?
공개소스 기반 파이썬 배포 도구 - 과학 기술목적 (딥러닝 등)
GUI 설치 지원, Jupyter 노트북 지원
다운로드: www.anaconda.com

-
- Keras
A deep learning library
- ▶
- 텐서플로 / 제논 / CNTK / ...
- CUDA / cuDNN BLAS / Eigen
- GPU CPU
- Copyright 2015, all rights reserved.

학습 정리

학습정리

- ❖ 구글 코랩을 이용하여 웹 기반으로 파이썬과 딥러닝을 공부하거나 프로그래밍하는 법을 배웠습니다.
- ❖ 로컬 PC 환경에서 파이썬 및 머신러닝을 구축하는 방법을 실습 영상을 통하여 학습하였습니다.
- ❖ Jupyter 노트북이나 IDE같은 다양한 개발 도구가 활용 가능합니다.

케라스 랑 놀면 뭐 할까?

Rank 2 텐서 데이터 변형 연습

과일 3개의 크기 및 무게 속성을 변화시켜 보기

과일 3개의 크기 및 무게 데이터

```
x = np.array([[ 1.0, 1.1 ], [ 2.0, 2.1 ], [ 0.9, 1.1 ]])
```

크기와 무게를 2배로 만들기

```
x_big = x*2  
print(x_big)  
[[2.  2.2]  
 [4.  4.2]  
 [1.8 2.2]]
```


케라스랑 놀면 뭐 할까?

Rank 2 텐서 데이터 특정 칼럼 속성 변경

과일 3개의 무게만 변화시켜 보기

```
x_size = x[:, 1:2] # 두 번째 컬럼만 추출
x_size *= 2
x_big2 = x
x_big2[:, 1:2] = x_size # 두 번째 열 데이터 교체
print(x_big2)
[[1.  2.2]
 [2.  4.2]
 [0.9 2.2]]
```

참고 문헌

REFERENCE

This document contains information
on artificial intelligence (AI) and
its application to various fields.
It is intended for educational purposes
only and should not be used for
commercial or other purposes.

강의에 필요한 URL

- 구글 코랩: colab.research.google.com
- 아나콘다 다운로드: www.anaconda.com/products/individual
- 맥 홈브루 설치: brew.sh/index_ko

참고 URL

- 텐서플로우-케라스 튜토리얼:
www.tensorflow.org/tutorials/keras/classification?hl=ko
- PyCharm Community Edition:
www.jetbrains.com/ko-kr/pycharm/download/#section=mac
- 서울시연구데이터서비스:
<http://data.si.re.kr/collection/view/360>