

대학 수학 학습을 위한 파이썬 부트캠프

IV: Python 라이브러리 활용

구성

Python 프로그래밍의 이해 (변수와 연산)

9/25(토) 13:00 ~ 15:00	주제	· Python 소개 · 변수 및 기본 연산자
	실습 예제	· Python 개발 환경 세팅(Anaconda 설치, Jupyter notebook 활용) · Python 코딩 기초 실습

Python의 문법과 자료형 (벡터와 행렬 및 수치 미분)

10/2(토) 13:00 ~ 15:00	주제	· 기초 문법 (Syntax, comment, indentation) · 기본자료형 · 복합자료형 (list, vector, matrix) · 함수의 정의
	실습 예제	· Python 자료형 활용 예제 - 미적분학 1: 행렬의 연산 · 함수 활용 예제 - 수치 미분

Python 제어문 활용 (테일러 급수와 근 찾기 알고리즘)

10/9(토) 13:00 ~ 15:00	주제	· 조건문, 반복문 · 재귀함수
	실습 예제	· 반복문 실습 예제 - 미적분학 1: 테일러 급수 연산 · 재귀함수 실습 예제 - 미적분학 1: Newton's method

Python 라이브러리 활용 (다변수함수 시각화)

10/16(토) 13:00 ~ 15:00	주제	· 라이브러리 활용(Matplotlib, numpy, math, sympy 등) · Python 응용
	실습 예제	· 미적분학 2: 다변수 함수 3D 시각화 (그래프, 등위면, 극점, 안장점 등)

※ Zoom 온라인 미팅으로 실시간 진행됩니다.

※ 주차별 강의는 Co-티칭으로 진행되며, 불참시 녹화 영상이 제공되지 않습니다.

Remind: 함수 사용의 이유

- 재사용성
 - 코드의 불필요한 반복을 피할 수 있음

```
# 함수명: average
# 매개변수: score1, score2, score3
# 반환값: avg
def average(score1, score2, score3):
    avg = (score1 + score2 + score3) / 3
    return avg

student_1_score = [100,50,70]
student_2_score = [55,60,55]
student_3_score = [40,30,30]
student_4_score = [20,10,15]
student_5_score = [80,70,20]

result1 = average(student_1_score[0], student_1_score[1], student_1_score[2])
result2 = average(student_2_score[0], student_2_score[1], student_2_score[2])
result3 = average(student_3_score[0], student_3_score[1], student_3_score[2])
result4 = average(student_4_score[0], student_4_score[1], student_4_score[2])
result5 = average(student_5_score[0], student_5_score[1], student_5_score[2])
```

Q. 프로그램마다 매번 모든 함수를 정의해서
써야 하는걸까?

라이브러리

- 응용 프로그램 개발을 위해 필요한 함수들의 집합
 - 다수의 패키지와 모듈을 가지고 있음 (패키지의 모음)
 - 표준 라이브러리 & 외부 라이브러리가 존재
- 모듈과 패키지
 - 모듈(module): 서로 연관된 함수들이나 클래스를 담고 있는 한 개의 파이썬 코드 파일(.py 확장자)
 - 패키지(package): 여러 개의 모듈들을 모아놓아 폴더(디렉토리)에 묶어서 기능 제공
 - 폴더안에 __init__.py 파일이 존재한다

```
def randrange(self, start, stop=None, step=1, _int=int):
    """Choose a random item from range(start, stop[, step]).

    This fixes the problem with randint() which includes the
    endpoint; in Python this is usually not what you want.

    """

    # This code is a bit messy to make it fast for the
    # common case while still doing adequate error checking.
    istart = _int(start)
    if istart != start:
        raise ValueError("non-integer arg 1 for randrange()")
    if stop is None:
        if istart > 0:
            return self._randbelow(istart)
        raise ValueError("empty range for randrange()")

    # stop argument supplied.
    istop = _int(stop)
    if istop != stop:
        raise ValueError("non-integer stop for randrange()")
    width = istop - istart
    if step == 1 and width > 0:
        return istart + self._randbelow(width)
    if step == 1:
        raise ValueError("empty range for randrange() (%d,%d, %d)" % (istart, istop, width))

    # Non-unit step argument supplied.
    istep = _int(step)
    if istep != step:
        raise ValueError("non-integer step for randrange()")
    if istep > 0:
        n = (width + istep - 1) // istep
    elif istep < 0:
        n = (width + istep + 1) // istep
    else:
        raise ValueError("zero step for randrange()")

    if n <= 0:
        raise ValueError("empty range for randrange()")

    return istart + istep*self._randbelow(n)

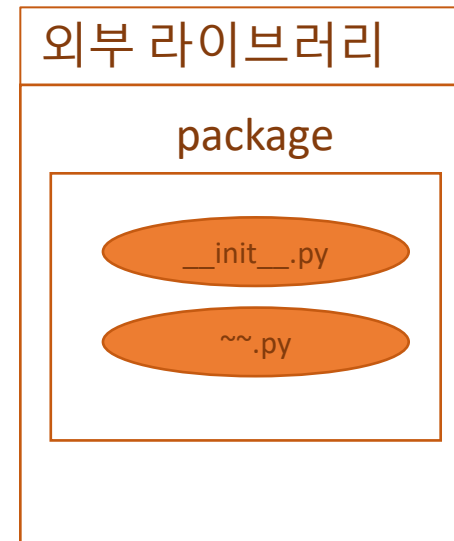
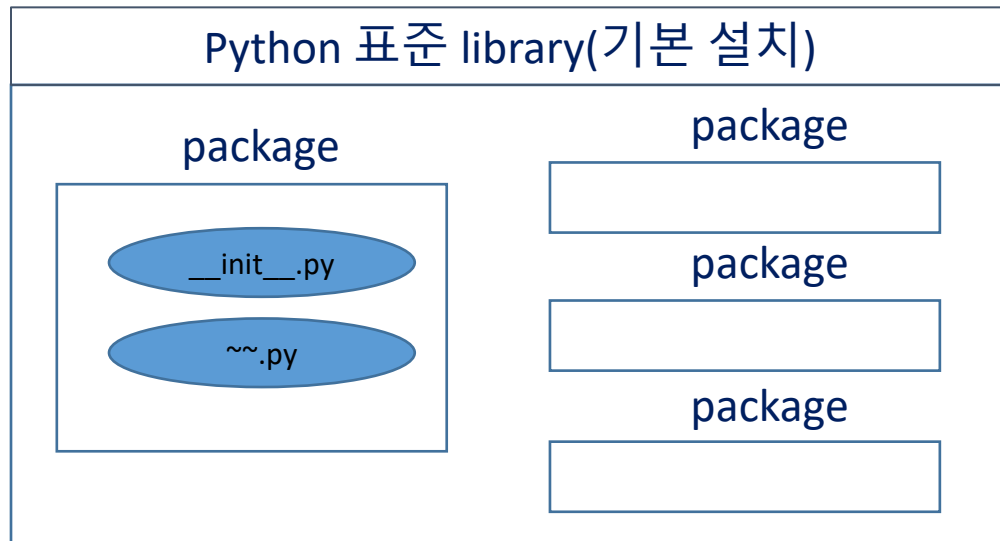
def randint(self, a, b):
    """Return random integer in range [a, b], including both end points.
    """

    return self.randrange(a, b+1)
```

random 모듈 코드

라이브러리

- 표준 라이브러리
 - 프로그래밍 언어가 기본적으로 가지고 있는 라이브러리
 - 파이썬 표준 라이브러리: csv, math, time, random 등 다양한 기능 지원
 - <https://docs.python.org/ko/3/library/index.html>
- 외부 라이브러리
 - 설치가 필요한 라이브러리(≈ 패키지)

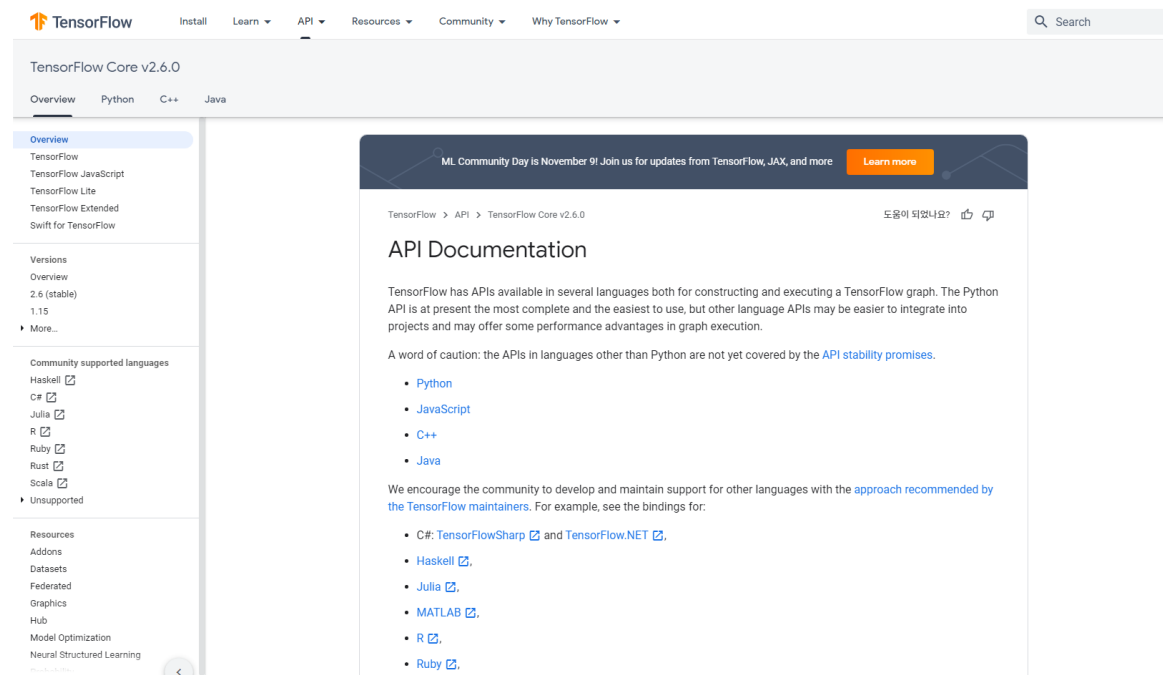


API

- 애플리케이션 프로그래밍 인터페이스 (Application Programming Interface)
 - 응용 프로그램에서 사용할 수 있도록, 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스
 - 사용자는 API의 내부 기능을 어떻게 구현하는지 몰라도 사용만 하면된다

- API Document
 - API의 사용법과 규격을 제공하는 문서
 - 라이브러리 홈페이지에 API Document가 존재

* 파이썬은 다양한 library를 제공하고,
API doc을 통해 사용법을 제공한다



라이브러리 검색과 활용

Q. 임의의 로또 번호 6개를 생성해 출력하는 프로그램

- 1~45 사이의 중복되지 않은 랜덤 숫자 6개를 생성해야 한다



랜덤 숫자를 어떻게 생성하지?

라이브러리 검색과 활용

Google

python generate random integers in range

전체 동영상 이미지 뉴스 쇼핑 더보기 도구

검색결과 약 18,100,000개 (0.35초)

Generating random number list in Python

```
1. import random
n = random.random()
print(n)
2. import random
n = random.randint(0,22)
print(n)
3. import random
randomlist = []
for i in range(0,5):
    n = random.randint(1,30)
    randomlist.append(n)
...
4. import random
#Generate 5 random numbers between 10 and 30
randomlist = []
for i in range(0,5):
    n = random.randint(10,30)
    randomlist.append(n)
print(randomlist)
```

2019. 8. 8.

<https://www.tutorialspoint.com/generating-random-num...>

Generating random number list in Python - Tutorialspoint

추천 스니펫 정보 · 사용자 의견

<https://stackoverflow.com/questions/generate-rando...>

Generate random integers between 0 and 9 - Stack Overflow

2016. 3. 12. — 22 Answers · A: random.random returns a random float in the range [0.0, 1.0) · B: random.randint returns a random integer N such that a <= N <= b · C: random.randrange(10) returns a random integer in the range [0, 10) · Try: from random import randrange print(randrange(10)) Docs: <https://docs.python.org/3/library/random.html>

Generate 'n' unique random numbers within a range - Stack Overflow · 답변 4개 · 2014년 4월 3일

How to get a random number between a float range? - Stack Overflow · 답변 5개 · 2018년 4월 2일

python - generate random number [duplicate] - Stack Overflow · 답변 3개 · 2021년 5월 25일

In python, how to generate a random integer in a range ... · 답변 3개 · 2014년 4월 17일

<https://docs.python.org/3/library/random.html>

random — Generate pseudo-random numbers — Python 3.10 ...

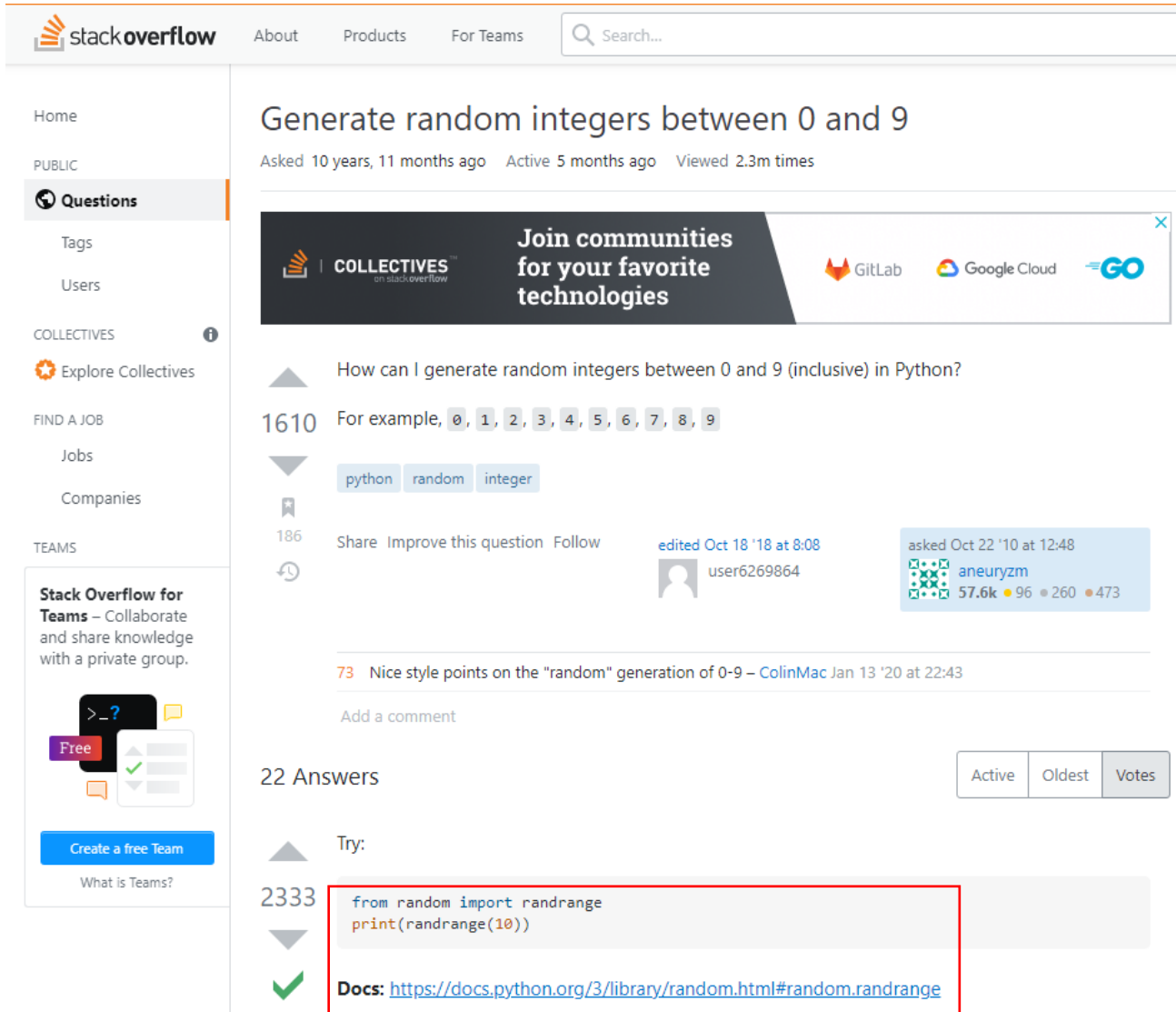
Almost all module functions depend on the basic function random(), which generates a random float uniformly in the semi-open range [0.0, 1.0). Python uses the ...

적당한 키워드로
검색



- Stack overflow
- Reference document
- Github
- ...

라이브러리 검색과 활용



stackoverflow About Products For Teams Search...

Home PUBLIC Questions Tags Users COLLECTIVES Explore Collectives FIND A JOB Jobs Companies TEAMS Stack Overflow for Teams – Collaborate and share knowledge with a private group. Create a free Team What is Teams?

Generate random integers between 0 and 9

Asked 10 years, 11 months ago Active 5 months ago Viewed 2.3m times

Join communities for your favorite technologies

How can I generate random integers between 0 and 9 (inclusive) in Python?

1610 For example, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

python random integer

186 Share Improve this question Follow edited Oct 18 '18 at 8:08 user6269864 asked Oct 22 '10 at 12:48 aneurym 57.6k 96 260 473

73 Nice style points on the "random" generation of 0-9 – ColinMac Jan 13 '20 at 22:43

Add a comment

22 Answers

Try:

2333

```
from random import randrange
print(randrange(10))
```

Docs: <https://docs.python.org/3/library/random.html#random.randrange>

다른 정보는 없나
Doc을 보자



라이브러리 검색과 활용

`random.randrange(stop)`

`random.randrange(start, stop[, step])`

Return a randomly selected element from `range(start, stop, step)`. This is equivalent to `choice(range(start, stop, step))`, but doesn't actually build a range object.

The positional argument pattern matches that of `range()`. Keyword arguments should not be used because the function may use them in unexpected ways.

Changed in version 3.2: `randrange()` is more sophisticated about producing equally distributed values. Formerly it used a style like `int(random()*n)` which could produce slightly uneven distributions.

Deprecated since version 3.10: The automatic conversion of non-integer types to equivalent integers is deprecated. Currently `randrange(10.0)` is losslessly converted to `randrange(10)`. In the future, this will raise a `TypeError`.

Deprecated since version 3.10: The exception raised for non-integral values such as `randrange(10.5)` or `randrange('10')` will be changed from `ValueError` to `TypeError`.

`random.randint(a, b)`

Return a random integer N such that $a \leq N \leq b$. Alias for `randrange(a, b+1)`.

`random.getrandbits(k)`

Returns a non-negative Python integer with k random bits. This method is supplied with the MersenneTwister generator and some other generators may also provide it as an optional part of the API. When available, `getrandbits()` enables `randrange()` to handle arbitrarily large ranges.

Changed in version 3.9: This method now accepts zero for k .

random.randrange 말고
random.randint(1,45)를
사용해도 1~45의 숫자를
얻을 수 있겠군



라이브러리 검색과 활용

```
#random모듈의 randint 함수를 추가
from random import randint

#로또번호를 저장할 리스트 변수
lottery_numbers = []

#로또 번호 5개가 생성될때 까지
while len(lottery_numbers) < 5:
    #새로운 번호 생성
    new_number = randint(1,45)

    #로또에는 중복숫자가 존재하면 안된다
    if new_number in lottery_numbers:
        continue
    else:
        lottery_numbers.append(new_number)

print(lottery_numbers)
```



import

- 모듈 가져오기- import / from ... import

1. import 모듈

ex) import random
random.randint(1,45)

2. from 모듈 import 함수이름

ex) from random import randint
randint(1,45)

3. import 패키지.모듈

ex) import os.path

4. from 패키지 import 모듈

ex) from os import path

```
def randrange(self, start, stop=None, step=1, _int=int):
    """Choose a random item from range(start, stop[, step]).

    This fixes the problem with randint() which includes the
    endpoint; in Python this is usually not what you want.

    """

    # This code is a bit messy to make it fast for the
    # common case while still doing adequate error checking.
    istart = _int(start)
    if istart != start:
        raise ValueError("non-integer arg 1 for randrange()")
    if stop is None:
        if istart > 0:
            return self._randbelow(istart)
        raise ValueError("empty range for randrange()")

    # stop argument supplied.
    istop = _int(stop)
    if istop != stop:
        raise ValueError("non-integer stop for randrange()")
    width = istop - istart
    if step == 1 and width > 0:
        return istart + self._randbelow(width)
    if step == 1:
        raise ValueError("empty range for randrange() (%d,%d, %d)" % (istart, istop, width))

    # Non-unit step argument supplied.
    istep = _int(step)
    if istep != step:
        raise ValueError("non-integer step for randrange()")
    if istep > 0:
        n = (width + istep - 1) // istep
    elif istep < 0:
        n = (width + istep + 1) // istep
    else:
        raise ValueError("zero step for randrange()")

    if n <= 0:
        raise ValueError("empty range for randrange()")

    return istart + istep*self._randbelow(n)

def randint(self, a, b):
    """Return random integer in range [a, b], including both end points.
    """

    return self.randrange(a, b+1)
```

random 모듈 코드

①

②

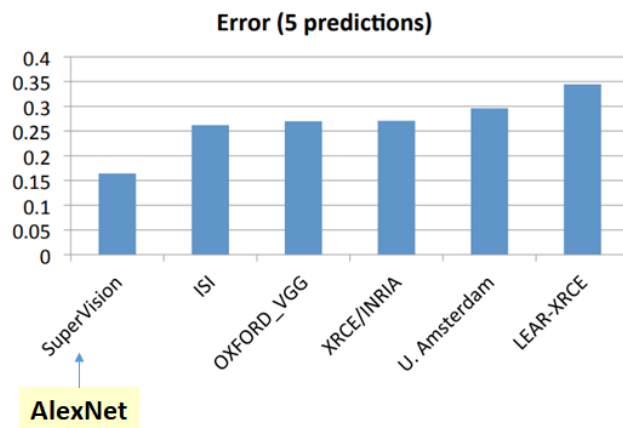
Remind: 파이썬의 장점

- 간결한 코드, 쉬운 문법: 직관적이고 간결한 문법 형식으로 초보자도 쉽게 배울 수 있음
- 다양성: 다양한 응용 프로그램 개발에 범용적으로 사용 가능
 - 데이터 분석
 - 웹 개발
 - 시스템 관리
 - 기계 학습, 딥러닝
 - 소프트웨어 테스트
- 풍부한 라이브러리 및 개발 도구
 - Numpy: 다차원 행렬 계산과 수치 계산 도구를 갖춘 파이썬 라이브러리
 - Scipy: 과학 컴퓨팅 및 기술 프로젝트에 사용되는 라이브러리 (선형대수, 신호 처리 등)
 - Tensorflow: 기계 학습 모델 구축 작업을 용이하게 하기 위해 구글에서 만든 라이브러리
 - Keras: 딥러닝 모델의 생성과 운영을 용이하게 하는 라이브러리
 - Matplotlib: 데이터 시각화를 용이하게 하는 프레임워크
 - Pandas: 데이터 계산 및 분석에 사용되는 라이브러리
- 대규모 개발자 커뮤니티: 다양한 참고자료와 정보가 존재
 - Stackoverflow: 프로그래밍 관련 Q&A 사이트
 - Github: 오픈 소스 코드 공유 및 협업 서비스

DeepLearning - ImageNet 2012

- ImageNet Large-Scale Visual Recognition Challenge(ILSVRC)-2012
 - 각 사진마다 정답을 5개까지 예측, 정확도를 측정
 - AlexNet이 압도적인 성능으로 우승

Ranking of the best results from each team



ImageNet 데이터셋

Tensorflow

- 기계 학습 모델 구축 작업을 용이하게 하기 위해 구글에서 만든 라이브러리
 - 파이썬 언어로 사용 가능
 - <https://www.tensorflow.org/tutorials/quickstart/beginner>
 - API: https://www.tensorflow.org/api_docs
- Tensorflow는 외부 라이브러리기 때문에 설치가 필요하다
 1. 구글 검색
 2. <https://www.tensorflow.org/install?hl=ko>

외부 라이브러리 설치

- Anaconda
 - 각종 수학/과학 라이브러리들을 같이 패키징한 python 배포판
 - 유명한 외부 라이브러리의 경우 이미 설치된 경우가 다수
 - Jupyter notebook 상에서 `!conda install [라이브러리] -y` 명령어로 설치

```
!conda install tensorflow -y
```

```
import tensorflow  
print(tensorflow.__version__)
```

```
2.3.0
```


Tensorflow 튜토리얼

```
import tensorflow as tf
# tensorflow, 대신 tf,으로 사용가능

mnist = tf.keras.datasets.mnist

# 데이터셋을 다운로드
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

#딤러닝 모델
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

#학습 방법
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

#학습 횟수 설정
model.fit(x_train, y_train, epochs=5)

#결과 출력
model.evaluate(x_test, y_test, verbose=2)
```



MNIST Dataset

Tensorflow 튜토리얼

```
import tensorflow as tf
# tensorflow, 대신 tf,으로 사용가능
```

```
mnist = tf.keras.datasets.mnist

# 데이터셋을 다운로드
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
#딤러닝 모델
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
#학습 방법
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
#학습 횟수 설정
model.fit(x_train, y_train, epochs=5)
```

```
#결과 출력
model.evaluate(x_test, y_test, verbose=2)
```

Sequential 함수
입력값: 리스트 []

Tensorflow 튜토리얼

```
import tensorflow as tf
# tensorflow, 대신 tf,으로 사용가능
```

```
mnist = tf.keras.datasets.mnist

# 데이터셋을 다운로드
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
#딤러닝 모델
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
#학습 방법
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
#학습 횟수 설정
model.fit(x_train, y_train, epochs=5)
```

```
#결과 출력
model.evaluate(x_test, y_test, verbose=2)
```

Dense 함수
입력값: 정수형 값, 문자열 값

tf.keras.layers.Dense



TensorFlow 1 version



View source on GitHub

Args

units

Positive integer, dimensionality of the output space.

activation

Activation function to use. If you don't specify anything, no activation is applied (ie. "linear" activation: $a(x) = x$).

Tensorflow 튜토리얼

What does Dense do?

Asked 4 years, 5 months ago · Active 7 months ago · Viewed 12k times

What is the meaning of the two `Dense` in this code?

```
4 self.model.add(Flatten())
  self.model.add(Dense(512))
  self.model.add(Activation('relu'))
  self.model.add(Dropout(0.5))
  self.model.add(Dense(10))
  self.model.add(Activation('softmax'))
  self.model.summary()
```

keras convolution keras-layer

Share Follow

edited Aug 23 '20 at 14:54



Donald Duck
7,066 ● 18 ● 63 ● 85

asked May 3 '17 at 8:47



K.SUP
41 ● 1 ● 1 ● 3

Add a comment

2 Answers

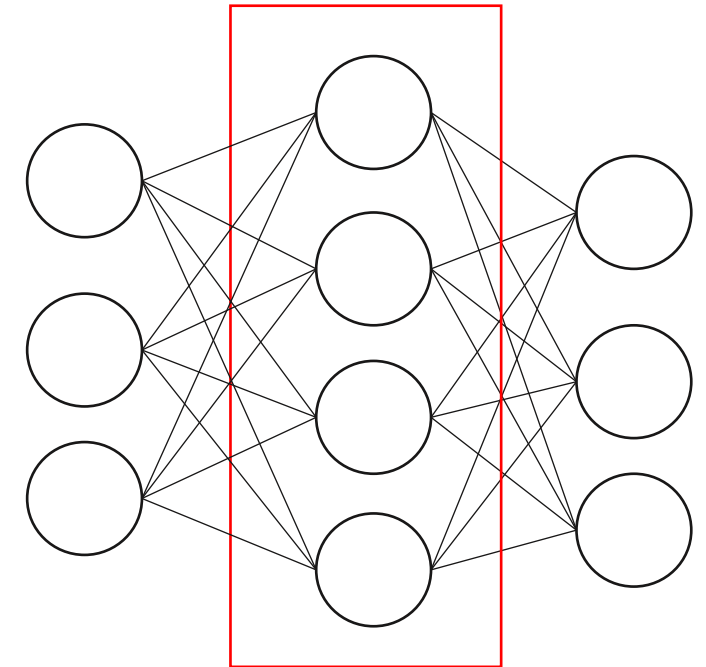
Active Oldest Votes

`Dense` is the only actual network layer in that model.

9 A `Dense` layer feeds all outputs from the previous layer to all its neurons, each neuron providing one output to the next layer.

It's the most basic layer in neural networks.

A `Dense(10)` has ten neurons. A `Dense(512)` has 512 neurons.



Dense

Tensorflow 튜토리얼

```
import tensorflow as tf
# tensorflow, 대신 tf,으로 사용가능
```

```
mnist = tf.keras.datasets.mnist

# 데이터셋을 다운로드
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
#딤러닝 모델
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

```
#학습 방법
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
#학습 횟수 설정
model.fit(x_train, y_train, epochs=5)
```

```
#결과 출력
model.evaluate(x_test, y_test, verbose=2)
```

Softmax는 뭐지?

Example usage: ⇔

```
>>> softmax = tf.nn.softmax([-1, 0., 1.])
>>> softmax
<tf.Tensor: shape=(3,), dtype=float32,
numpy=array([0.09003057, 0.24472848, 0.66524094], dtype=float32)>
>>> sum(softmax)
<tf.Tensor: shape=(), dtype=float32, numpy=1.0>
```


Tensorflow 튜토리얼

```
import tensorflow as tf
# tensorflow, 대신 tf,으로 사용가능
```

```
mnist = tf.keras.datasets.mnist

# 데이터셋을 다운로드
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
#딤러닝 모델
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

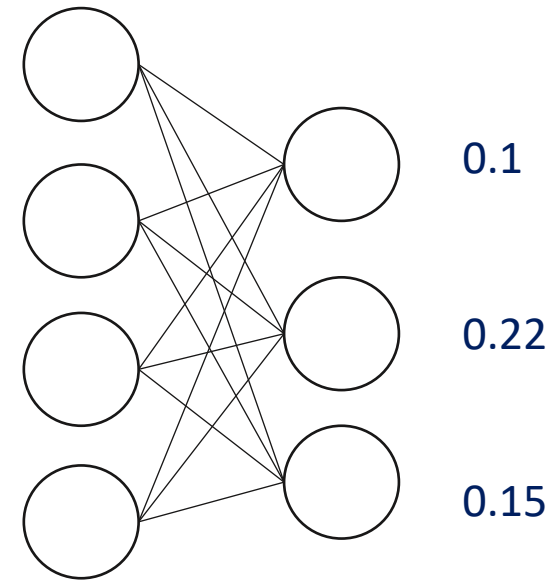
```
#학습 방법
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
#학습 횟수 설정
model.fit(x_train, y_train, epochs=5)
```

```
#결과 출력
model.evaluate(x_test, y_test, verbose=2)
```



[1,5,4,6,2, ...]



Dense(128) Dense(10)

*사진 데이터를 받아 128개의 뉴런을 가진 dense layer를 지난 뒤 10개의 확률 값 (각각 사진이 0~9의 숫자일 확률)을 계산