

# Deep learning 실습

## Python



Ha-Jin Yu, Dept. of Computer Science, University of Seoul

서울시립대학교 컴퓨터과학부 유하진

2019.

HJYU@UOS.AC.KR

# Python 기본문법

- 변수 선언 없음 (형이 가변적)
- 끝에 세미콜론(;)을 붙이지 않아도 됨
- 중괄호({}) 대신 탭으로 구분
- 연산자: // (몫), \*\*(지수승)
- if / for/ while 끝에 콜론(:) 사용
- ‘&&’ 대신 ‘and’ 사용
- ‘||’ 대신 ‘or’ 사용
- ‘null’ 대신 ‘None’ 사용
- ‘def 함수명(인자):’의 형태로 함수 정의
- 함수 반환값이 여러 개가 될 수 있음

# Python 실습 – 변수 선언

## C / C++

```
#include <stdio.h>
#include <string.h>

void main()
{
    int a;
    double b;
    char c[10];

    a = 10;
    b = 3.14;
    strcpy(c, "apple");

    printf("%d %f %s", a, b, c);
}
```

## Python

```
a = 10
b = 3.14
c = 'apple'

print (a, b, c)

a,b,c = 10, 3.14, 'aa'
```

# Python 실습 – 조건문 (if)

## C / C++

```
#include <stdio.h>

void main()
{
    int length;

    printf("length of fish = ");
    scanf("%d", &length);

    if(length < 82)
        printf("salmon");
    else
        printf("sea bass");
}
```

## Python

```
print ('length of fish = ')
length = input( )

if int(length) < 82:
    print ('salmon')
else:
    print ('sea bass')
```

# Python 실습 – 반복문 (for)

C / C++

```
#include <stdio.h>

void main()
{
    int sum = 0;

    for(int i = 1; i <= 100; i++)
    {
        sum += i;
    }

    printf("%d", sum);
}
```

Python

```
sum = 0

for i in range(1, 101):
    sum += i

print (sum)
```

# Python 실습 – 자료구조

- tuple : ()
  - 상수형 리스트
- list : []
  - 배열, 연결리스트
- 사전 (dictionary) : {}
  - 인덱스가 지정된 리스트
  - Key와 value로 구성
  - Ex> a[‘apple’] = 0

# Python 실습 – list

```
a = [1, 3, 4, 56, 100]
print (a)
[1, 3, 4, 56, 100]
>>> print (a[0], a[1])
1 3
>>>
>>> a[0] = 99
>>>
>>> print (a)
[99, 3, 4, 56, 100]
```

```
>>> u = ' university    of    seoul    '
>>> u.strip()
'university    of    seoul'
>>> a = u.split()
>>> a
['university', 'of', 'seoul']
>>> b = u.split('of')
>>> b
[' university ', ' seoul ']
```

# Python 실습 – dictionary

```
>>> a = {'salmon': 85, 'sea bass': 100}  
>>> print (a)  
{'salmon': 85, 'sea bass': 100}  
>>> a['salmon'], a['sea bass']  
85 100  
>>> a['salmon'] = 99  
>>> a  
{'salmon': 99, 'sea bass': 100}  
>>> a.keys()  
['salmon', 'sea bass']  
>>> a.values()  
[85, 100]  
>>> a.items()  
[('salmon', 99), ('sea bass', 100)]
```

```
>>> for k in a.keys():  
...     print(k)  
...  
salmon  
sea bass  
>>> 'salmon' in a  
True
```

# Python 기본문법 2

- 주석
  - #을 붙인 곳부터 라인의 끝까지 주석 처리됨
  - ""이 시작된 곳부터 다음 ""이 등장하는 곳까지 주석 처리됨  
(여러 줄 주석 가능)

- 라이브러리 로드

- import 라이브러리명
    - 라이브러리를 통째로 로드
    - ‘라이브러리명.함수명’으로 사용
  - from 라이브러리명 import 함수명
    - 라이브러리에서 특정 함수만 로드  
(함수명 대신 '\*'를 붙이면 모든 함수 로드)
    - ‘함수명’으로 사용

```
from tensorflow import *
import tensorflow as tf
y = tf.matmul(x, w)
```

# Python 실습 – 파일 입출력

## 파일 읽기

```
# 파일 열기  
fd = open('test.txt', 'r')  
  
# 한 줄씩 읽기  
lines = fd.readlines()  
  
# 닫기  
fd.close()  
  
print (lines)
```

## 파일 쓰기

```
# 파일 열기  
fd = open('test.txt', 'w')  
  
# 파일에 쓰기  
fd.write('test1\n')  
fd.write('test2\n')  
  
# 닫기  
fd.close()
```

# Python 실습 – 리스트 다루기

## 리스트 맨 뒤에 추가하기

```
>>> a = [1, 2, 3, 4, 5]  
>>> a.append(6)  
>>> a.append(7)  
>>> a.append(8)  
>>> a  
[1, 2, 3, 4, 5, 6, 7, 8]
```

## 리스트끼리 연결하기

```
>>> a = [1, 2, 3, 4, 5]  
>>> b = [6, 7, 8]  
>>> a.extend(b)  
>>> a  
[1, 2, 3, 4, 5, 6, 7, 8]
```

# Python 실습 – 리스트 다루기

## 리스트의 일부만 선택하기

```
>>> a=[1,2,3,4,5,6]
>>> a[2:4]
[3, 4]
>>> a[2:-1]
[3, 4, 5]
>>> a[1:5:2]
[2, 4]
>>> a[1:-1:2]
[2, 4]
>>> b=a[2:5]
>>> b
[3, 4, 5]
```

## 리스트 내용 계산하기

```
>>> a = [5, 2, 3, 4, 1]
>>> a.sort()
>>> print (a)
[1, 2, 3, 4, 5]
```

# Python 실습 – 문자열 다루기

## 문자열도 일종의 리스트

```
>>> a = 'apple'  
>>> a = a[1:]  
>>> print a  
pple
```

## 문자열 비교

```
a = 'apple'  
  
if a == 'apple':  
    print ('ok')
```

# Python 실습 – 형 변환

## 문자 → 정수형 숫자

```
>>> a = '1'  
>>> b = 2  
>>> print (int(a) + b)  
3
```

## 문자열로 치환하기

```
>>> a = 'apple'  
>>> b = 1  
>>> c = 6.55  
>>> d = '%s %d %.2f' % (a, b, c)  
>>> print (d)  
apple 1 6.55
```

# Python - list

## List의 unpacking 1

```
>>> location = [1, 2]
>>> row = location[0]
>>> col = location[1]
>>> print (row, col)
1 2
```

## List의 unpacking 2

```
>>> location = [1, 2]
>>> row, col = location
>>> print (row, col)
1 2
```

# Class

```
class TestClass:  
    # 클래스 생성자  
    def __init__(self, testArg):  
        print( 'Init Class')  
        print( 'argument =', testArg)  
  
        # 멤버 변수 선언  
        self.testMember = 0  
  
    # 클래스 내부에서의 멤버 함수 호출  
    self.TestMemberFunc( )  
  
# 멤버 함수 예시  
def TestMemberFunc(self):  
    print( 'self.testMember =', self.testMember)  
    self.testMember += 1  
  
test = TestClass(123) # 클래스 생성  
  
test.TestMemberFunc() # 클래스 외부에서의 멤버 함수 호출
```

실행 결과 :

```
Init Class  
argument = 123  
self.testMember = 0  
self.testMember = 1
```

# Class inheritance

```
>>> class Parent:  
>>>     def f(self):  
        print 'Parent'  
>>>  
>>> class Child(Parent): #class Parent 의 속성을 상속 받음  
>>>     def g(self):  
        print 'Child'  
>>>  
>>> d = Child()  
>>> d.f()  
Parent  
>>> d.g()  
Child
```

# 구조 복사하기

- Python에서는 구조(tuple, list, dictionary 등)의 경우 reference가 복사되므로 이에 유의

```
a = [1, 2, 3]
b = a
a = [1, 2, 3, 4]
```

```
print (a, b)
```

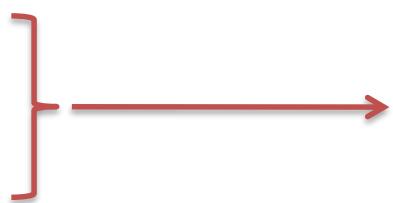
```
a = [1, 2, 3]
b = a
a.append(4)
```

```
print (a, b)
```



실행 결과 :

```
[1, 2, 3, 4] [1, 2, 3]
```

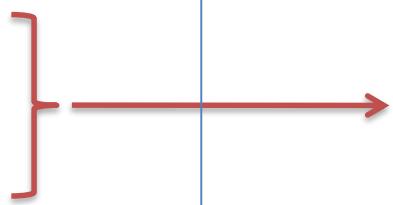


실행 결과 :

```
[1, 2, 3, 4] [1, 2, 3, 4]
```

```
a = [1, 2, 3]
b = a.copy()
a.append(4)
```

```
print (a, b)
```



실행 결과 :

```
[1, 2, 3, 4] [1, 2, 3]
```

# NumPy

```
>>> import numpy as np  
>>> a = np.arange(10)  
>>> a  
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])  
>>> np.zeros( (3,4) )  
array([[ 0.,  0.,  0.,  0.],  
       [ 0.,  0.,  0.,  0.],  
       [ 0.,  0.,  0.,  0.]])  
>>> np.ones( (2,3,4), dtype=np.int16 )  
array([[[1, 1, 1, 1],  
       [1, 1, 1, 1],  
       [1, 1, 1, 1]],  
      [[1, 1, 1, 1],  
       [1, 1, 1, 1],  
       [1, 1, 1, 1]]], dtype=int16)
```

```
>>> a = np.arange(15).reshape(3, 5)  
>>> a  
array([[ 0,  1,  2,  3,  4],  
       [ 5,  6,  7,  8,  9],  
       [10, 11, 12, 13, 14]])  
>>> a.shape  
(3, 5)  
  
a[0, 1] # 0행 1열의 원소  
a[:, 1] # 1열의 모든 원소  
a[0, :] # 0행의 모든 원소
```

# Numpy - array

## 순수 Python 사용

```
current = [1, 2]          # 현재 좌표  
goal = [5, 6]            # 도착 좌표  
  
# 거리 계산  
  
# 행 간 거리  
rowDist = abs(current[0] - goal[0])  
# 열 간 거리  
colDist = abs(current[1] - goal[1])  
# 총 거리  
dist = rowDist + colDist  
  
print (dist)
```

## NumPy 사용

```
from numpy import *  
  
# Numpy 형식인 array 형으로 변환한 뒤 동  
일한 연산 수행  
  
# 현재 좌표  
current = array([1, 2])  
# 도착 좌표  
goal = array([5, 6])  
dist = sum(abs(current - goal))  
  
print (dist)
```

# Numpy – 2D array

## 2D array shape / slicing

```
>>> from numpy import *
>>> # 2차원 list를 array로 변환
a = array([[1, 2], [3, 4], [5, 6]])
>>> print (a)      # 전체 출력
[[1 2]
 [3 4]
 [5 6]]
>>> print a.shape    # array 모양 출력
(3, 2)
>>> print a[0, 1]    # 0행 1열의 원소
2
>>> print a[:, 1]    # 1열의 모든 원소
[2 4 6]
>>> print a[0, :]    # 0행의 모든 원소
[1 2]
```

## Array reshape

```
>>> from numpy import *
>>> # 1차원 list를 2D array로 변환
a = [1, 2, 3, 4, 5, 6]
>>> b = array(a).reshape(3, 2)
>>> print (b)
[[1 2]
 [3 4]
 [5 6]]
>>> c = b.reshape(2,3)
>>> c
array([[1, 2, 3],
       [4, 5, 6]])
```

# Numpy – 파일 입출력

## 파일로 저장

```
from numpy import *

# 2차원 list를 array로 변환
a = array([[1, 2], [3, 4], [5, 6]])

# 파일로 저장 (1차원으로 저장됨)
a.tofile('test.txt', sep = '\t')
```

## 파일에서 읽기

```
from numpy import *

# 파일로부터 array 읽기
# (1차원으로 읽힘)
b = fromfile('test.txt', sep = '\t')

# 2차원으로 모양 재설정 후 형 변환
c = b.reshape(3, 2).astype('int')

print (c)
```

# NumPy array형의 연결

```
>>> from numpy import *
>>>
>>> a = array([1, 2])
>>> b = array([3, 4])
>>>
>>> # 상하로 연결
c = vstack((a, b))
>>>
>>> # 좌우로 연결
d = hstack((a, b))
>>>
>>> print c
[[1 2]
 [3 4]]
>>> print d
[1 2 3 4]
```

# numpy 를 이용한 perceptron 구현

```
from numpy import *
data = [
    (array([0,0,1]), 0), # x, y
    (array([0,1,1]), 1),
    (array([1,0,1]), 1),
    (array([1,1,1]), 1),
]
w = random.rand(3)
c = 0.2 # learning rate
n = 10 # number of epoch
```

```
# Trainning
for i in range(n):
    for j in range(len(data)):
        x, y = data[j]
        net = dot(w, x)
        if net<0:
            fx=0
        else:
            fx=1
        delta = y - fx
        w += c * delta * x

#test
for x, _ in data:
    net = dot(x, w)
    if net<0:
        fx=0
    else:
        fx=1
    print (x[:2], net, fx)
```