

수치미분 최종 버전 – numerical derivative

```
import numpy as np

def numerical_derivative(f, x):
    delta_x = 1e-4
    grad = np.zeros_like(x)

    it = np.nditer(x, flags=['multi_index'], op_flags=['readwrite'])

    while not it.finished:
        idx = it.multi_index

        tmp_val = x[idx]
        x[idx] = float(tmp_val) + delta_x
        fx1 = f(x)  # f(x+delta_x)

        x[idx] = float(tmp_val) - delta_x
        fx2 = f(x)  # f(x-delta_x)
        grad[idx] = (fx1 - fx2) / (2*delta_x)

        x[idx] = tmp_val
        it.iternext()

    return grad
```

1번문제

다음과 같은 결과가 나오도록 수치미분 코드를 변경하시오. 즉 debug 문장을 추가하시오

2번문제

$f(x,y) = 2x + 3xy + y^3$ 에 대하여 수치미분 코드 numerical_derivative 함수를 이용해서 $f'(1.0, 2.0)$ 계산하는 코드를 구현 하시오

```
# 입력변수 1 개의 함수  $f(x) = x^2$ 
```

```
def func1(w):
```

```
    x = w
```

```
    return x**2
```

```
# lambda function 정의
```

```
f = lambda w : func1(w)
```

```
w = np.array([3.0])
```

```
#  $x = 3.0$  에서의 편미분 값
```

```
ret = numerical_derivative( f, w )
```

```
print('type(ret) = ', type(ret), ', ret_val = ', ret)
```

```
debug 1. initial input variable = [3.]
```

```
debug 2. initial grad = [0.]
```

```
=====
```

```
debug 3. idx = (0,) , x[idx] = 3.0
```

```
debug 4. grad[idx] = 6.000000000039306
```

```
debug 5. grad = [6.]
```

```
=====
```

```
type(ret) = <class 'numpy.ndarray'> , ret_val = [6.]
```

3번문제

수치미분 코드 numerical_derivative 함수를 이용해서 다음과 같은 행렬을 입력으로 받는 4변수 함수 $f(w, x, y, z)$ 에서 미분 값을 구하는 코드를 구현하시오

$$\begin{pmatrix} w & x \\ y & z \end{pmatrix} = \begin{pmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{pmatrix}$$

$$f(w,x,y,z) = wx + xyz + 3w + zy^2$$