

AIoT AutoCar Prime 으로 배우는 온디바이스 AI 프로그래밍

6.1 Numpy

6.1.2 연산

연산

□ numpy 연산

- 파이썬에서 사용하는 연산자(+, -, *, /)를 사용 가능
- 파이썬의 일반적 연산과 달리 Numpy는 행렬곱 지원
- numpy 사칙연산 : 파이썬 보다 연산 속도가 빠름

```
01: import numpy as np
02:
03: x = np.arange(4).reshape(2,2)
04: print(x)
05: y = np.arange(3,-1,-1).reshape(2,2)
06: print(y)
07:
08: s1 = x + y
```

```
09: print(s1)
10: s2 = x - y
11: print(s2)
12: s3 = x * y
13: print(s3)
14: s4 = x / y
15: print(s4)
```

연산

□ Numpy의 메소드를 사용해 사칙연산 가능

```
01: import numpy as np
02:
03: x = np.arange(4).reshape(2,2)
04: print(x)
05: y = np.arange(3,-1,-1).reshape(2,2)
06: print(y)
07:
08: s1 = np.add(x,y)
```

```
09: print(s1)
10: s2 = np.subtract(x,y)
11: print(s2)
12: s3 = np.multiply(x,y)
13: print(s3)
14: s4 = np.divide(x,y)
15: print(s4)
```

행렬곱

- 행렬곱 : 머신러닝 연산에 주로 사용
 - ▣ Numpy는 연산 속도가 빠르고 행렬곱을 지원
 - 머신러닝 프로그래밍에서 매우 중요한 패키지로 사용됨
 - ▣ matmul : 행렬곱 메소드. 파라미터로 ndarray 2개 사용

```
01:         import numpy as np
02:
03:         x = np.arange(4).reshape(2,2)
04:         print(x)
05:         y = np.arange(3,-1,-1).reshape(2,2)
06:         print(y)
07:
08:         s1 = np.matmul(x,y)
09:         print(s1)
```

기타 연산

□ 배열의 절대값, 제곱, 제곱근, 로그를 구하는 연산 지원

```
01:      import numpy as np
02:
03:      x = np.arange(4).reshape(2,2)
04:      print(x)
05:
06:      s1 = np.abs(x)
07:      print(s1)
08:      s2 = np.square(x)
09:      print(s2)
10:      s3 = np.sqrt(x)
11:      print(s3)
12:      s4 = np.log(x)
13:      print(s4)
```

내용 정리

- 파이썬에서 사용하는 연산자(+, -, *, /)와 연산 메소드 사용 가능
- + 또는 `add(x, y)`: x와 y로 입력된 ndarray배열을 덧셈하여 반환
- - 또는 `subtract(x, y)`: 입력된 ndarray배열 x에 y를 뺄셈하여 반환
- * 또는 `multiply(x, y)`: x와 y로 입력된 ndarray배열을 곱셈하여 반환
- / 또는 `divide(x, y)`: 입력된 ndarray배열 x에 y를 나눗셈하여 반환

내용 정리

- `matmul(x, y)`: 입력된 ndarray배열 `x`에 `y`를 행렬곱하여 반환
- `abs(x)`: 입력된 ndarray배열 `x`를 절댓값으로 반환
- `square(x)`: 입력된 ndarray배열 `x`를 제곱하여 반환
- `sqrt(x)`: 입력된 ndarray배열 `x`를 제곱근으로 반환
- `log(x)`: 입력된 ndarray배열 `x`를 로그값으로 반환

연습문제

- 문제 11. 다음 코드를 읽고 출력을 작성해보세요.

```
01:      import numpy as np
02:
03:      arr = np.ones(5)
04:
05:      a1 = arr + arr
06:      a2 = np.add(arr, arr)
07:
08:      print(a1)
09:      print(a2)
```

연습문제

- 문제 12. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import numpy as np
02:
03:         arr = np.array([[1,2], [3,4]])
04:
05:         a1 = np.matmul(arr, arr)
06:
07:         print(a1)
```

연습문제

- 문제 13. 다음 코드를 읽고 arr을 제공하여 출력하는 코드를 작성해 보세요.

```
01:         import numpy as np
02:
03:         arr = np.array([0.1, 1, 10, 11])
```

연습문제

- 문제 14. 다음은 OLED의 (1,1)좌표에 ‘!’ 을 출력하는 예제입니다.

```
01:         from pop import Oled
02:
03:         display = Oled()
04:
05:         display.setCursor(1,1)
06:         display.print('!')
```

- A. 코드를 실행시켜 결과를 확인해 보세요.

연습문제

- ▣ B. 다음 코드에서 sqrt와 matmul 메소드만 사용하고, arr1과 arr2 배열만 사용하여 OLED에 '!' 가 (2,3) 좌표에 출력되도록 빈 칸을 채워보세요.

```
01:      from pop import Oled
02:      import numpy as np
03:
04:      display = Oled()
05:
06:      arr1 = np.array([[1,2],[4,1]])
07:      arr2 = np.array([1, 1])
08:
```

```
_____
```

```
...
```

```
_____
```

```
11:
12:      arr = np.abs(arr)
13:
14:      x = int(arr[0])
15:      y = int(arr[1])
16:
17:      display.setCursor(x,y)
18:      display.print('!')
```