

AIoT AutoCar Prime 으로 배우는 온디바이스 AI 프로그래밍

6.1 Numpy

6.1.3 데이터 처리

데이터 처리

- 데이터를 다양하게 활용할 수 있도록 데이터 처리 관련 메소드 지원
 - ▣ 데이터 필터링, 통계, 정렬 등 유용하게 사용할 수 있는 메소드

조건부 선택

- 두 배열에서 특정 조건에 따라 어느 배열의 원소를 가져올지 선택해 새로운 배열 생성 가능

- ▣ ex. 두 배열을 비교해 더 큰 값을 가져와 배열 생성

```
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.where(x > y, x, y)
09:         print(s1)
```

배열 통계

□ 배열 요소들의 합, 평균, 최대값, 최소값 연산

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

배열 통계

□ 표준편차, 분산, 원소 누적 합과 곱 연산 지원

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

집합 연산

□ 배열 비교, 정렬, 중복 제거, 합집합 등 집합 연산 메소드 사용가능

```
01:         import numpy as np
02:
03:         x = np.ones((2,2))
04:         print(x)
05:         y = np.arange(3,-1,-1).reshape(2,2)
06:         print(y)
07:
08:         s1 = np.sort(y)
09:         print(s1)
10:         s2 = np.unique(x)
11:         print(s2)
12:         s3 = np.union1d(x,y)
13:         print(s3)
14:         s4 = np.setdiff1d(y,x)
15:         print(s4)
```

파일 저장 및 로드

- `save(name, x)` : 파일 저장 메소드
 - ▣ 'name' 파일명, '.npy' 확장자인 바이너리 형식으로 저장
- `load(name)` : 파일 로드 메소드
 - ▣ 파일명은 확장자까지 모두 포함해야함.

```
01:         import numpy as np
02:
03:         x = np.random.random(20) #랜덤 값 20개를 생성
04:         print(x)
05:
06:         np.save("randoms", x)
07:
08:         y=np.load("randoms.npy")
09:         print(y)
```

파일 저장 및 로드

- savez(name, arr_name=x, ...) : 여러 개의 배열을 한 파일로 저장
 - '.npz' 확장자인 바이너리 형식으로 저장
 - 파일 사용 후 close메소드로 닫아줘야 함

```
01: import numpy as np
02:
03: x = np.random.random(20) #랜덤 값 20개를 생성
04: print(x)
05: y = np.random.random(2) #랜덤 값 20개를 생성
06: print(y)
07:
```

```
08: np.savez("randoms", arr1=x, arr2=y)
09:
10: z=np.load("randoms.npz")
11: print(z["arr1"])
12: print(z["arr2"])
13: z.close()
```

파일 저장 및 로드

- `savetxt(name, x)` : 텍스트 파일 형식으로 저장
- `loadtxt(name)` : 텍스트 파일 로드

```
01: import numpy as np
02:
03: x = np.arange(20) # 0부터 19까지 20개의 값을 생성
04: print(x)
05:
06: np.savetxt("texts.txt", x)
07:
08: y=np.loadtxt("texts.txt")
09: print(y)
```

내용 정리

- `where(cond, x, y)` : 조건 연산자 또는 True, False 배열을 이용
 - ▣ 조건이 참이면 x의 원소, 거짓이면 y의 원소를 선택해 새로운 배열 반환
- `sum(x)`: 입력된 ndarray배열의 모든 원소 값을 덧셈하여 반환
- `mean(x)`: 입력된 ndarray배열의 모든 원소 값의 평균을 반환
- `max(x)`: 입력된 ndarray배열의 모든 원소 값 중 최대값을 반환
- `min(x)`: 입력된 ndarray배열의 모든 원소 값 중 최솟값을 반환
- `std(x)`: 입력된 ndarray배열의 표준편차를 반환

내용 정리

- `var(x)`: 입력된 ndarray배열의 분산을 반환
- `cumsum(x)`: 입력된 ndarray배열의 원소 순 누적 합을 반환
- `cumprod(x)`: 입력된 ndarray배열의 원소 순 누적 곱을 반환
- `sort(x)`: 입력된 ndarray배열을 정렬하여 반환
- `unique(x)`: 입력된 ndarray배열의 중복을 제거하여 반환
- `union1d(x, y)`: 입력된 ndarray `x`와 `y`배열의 합집합을 반환
- `setdiff1d(x, y)`: 입력된 ndarray `x`에 대한 `y`배열의 차집합을 반환

내용 정리

- `save(name, x)` :
 - ▣ `x`를 `name` 파일명, `‘.npy’` 확장자인 바이너리 형식으로 저장
- `load(name)` :
 - ▣ 파일 로드 메소드. 파일명은 확장자까지 모두 포함해야 함
- `savez(name, arr_name=x, ...)` : 여러 개의 배열을 한 파일로 저장
 - ▣ `name`은 파일명, `arr_name`은 배열 `x`의 이름을 지정
 - ▣ 파일 확장자는 `‘.npz’` 로 저장되며 바이너리 형식으로 저장
 - ▣ 파일 사용 종료시 `close` 메소드를 이용해 파일을 닫아줘야 함

내용 정리

- `savetxt(name, x) / loadtxt(name)` : `x`를 `name`이름의 텍스트 파일
저장 / 로드

연습문제

- 문제 15. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import numpy as np
02:
03:         arr = np.array([1,2,3,4])
04:
05:         a1 = np.sum(arr)
06:         a2 = np.min(arr)
07:         a3 = np.mean(arr)
08:
09:         print(a1)
10:         print(a2)
11:         print(a3)
```

연습문제

- 문제 16. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import numpy as np
02:
03:         arr1 = np.array([1,2,3,4])
04:         arr2 = np.array([4,3,2,1])
05:
06:         a1 = np.where(arr1>arr2, arr1, arr2)
07:
08:         print(a1)
```

연습문제

- 문제 17. 다음 코드를 읽고 arr을 정렬하여 출력하는 코드를 작성해 보세요.

```
01:         import numpy as np
02:
03:         arr = np.array([5, 1, -10, 11, 3])
```

연습문제

- 문제 18. 다음 코드는 Cds센서의 밝기 값을 10개씩 arr1과 arr2에 저장하고 출력하는 코드입니다.

```
01:      from pop import Cds,delay
02:      import numpy as np
03:
04:      cds = Cds(7)
05:
06:      arr1 = np.zeros(10)
07:      arr2 = np.zeros(10)
08:
09:      for i in range(len(arr1)):
```

```
10:          arr1[i] = cds.readAverage()
11:          delay(300)
12:
13:          for i in range(len(arr2)):
14:              arr2[i] = cds.readAverage()
15:              delay(300)
16:
17:          print(arr1)
18:          print(arr2)
```

연습문제

- ▣ A. 코드를 실행시키고 CDS센서에 변화를 주었을 때 출력을 확인해 보세요.
- ▣ B. arr1, arr2 두개의 배열을 비교하여 작은 값이 저장되는 새로운 배열 arr3을 생성해 보세요.
- ▣ C. B에서 생성된 배열을 오름차순으로 정렬해 보세요.
- ▣ D. C에서 정렬된 배열을 ARRAY.txt로 저장하고 로드해 보세요.