

AIoT AutoCar Prime 으로 배우는 온디바이스 AI 프로그래밍

6.1 Numpy

6.1.1 배열

numpy

- numpy(넘파이) : C언어 기반 고속 연산 라이브러리
 - ▣ 벡터 및 텐서 연산에 편리한 기능들을 사용 가능
 - ▣ 선형 대수 계산에 유용한 패키지
 - ▣ 파이썬의 편의성과 C언어의 연산 능력을 동시에 이용 가능

배열

- ndarray : 빠른 연산을 위한 numpy의 배열 타입
 - ▣ array(list): 파이썬의 리스트, 튜플, 배열 타입을 Numpy의 ndarray타입으로 생성
 - ▣ zeros(shape): 값을 0으로 채운 shape 형태로 ndarray를 생성
 - ▣ ones(shape): 값을 1으로 채운 shape 형태로 ndarray를 생성
 - ▣ full(shape, n): 값을 n으로 채운 shape 형태로 ndarray를 생성
 - ▣ eye(size): size x size 크기의 ndarray를 대각선은 1, 나머지는 0으로 채워 생성

배열

- ▣ `random.random(shape)`: 랜덤 값으로 채운 shape 형태의 ndarray 생성
- ▣ `arange(start, stop, step)`: start 부터 stop까지 step 만큼 증감시킨 값을 갖는 ndarray를 생성
 - start 파라미터만 입력에 생성 가능.
 - 이 경우 0부터 start 까지 1씩 증감하는 값으로 생성
- ▣ `reshape(shape)`: ndarray를 shape 형태로 재구성

생성

- ndarray는 numpy.array 메소드로부터 생성
 - ▣ 파이썬의 리스트, 튜플, 배열 타입 입력 가능

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

생성

- ndarray는 파이썬의 배열 타입처럼 다차원으로 생성 가능
 - ▣ ndim과 shape 속성을 이용해 차원과 배열 형태를 확인가능

```
01:         import numpy as np
02:
03:         arr = np.array([[1,2,3],[4,5,6]])
04:         print(arr.ndim)
05:         print(arr.shape)
```

생성

□ numpy.arange메소드

- ▣ 리스트 중에서 패턴이 반복되는 형태 배열을 생성할 때 사용
- ▣ 1개나 3개의 파라미터 사용
 - 파라미터를 1개만 받은 경우 : 0부터 n까지 1씩 증감하는 형태로 생성
 - 파라미터를 3개를 받은 경우 : 값 범위와 증가 폭을 지정 가능

```
01:         import numpy as np
02:
03:         arr = np.arange(5)
04:         print(arr)
05:         arr = np.arange(5,15,2)
06:         print(arr)
```

초기와

□ 특정 값을 채워 초기화시킨 ndarray를 생성 가능

▣ numpy.zeros : 배열값을 0으로 채운 ndarray배열 생성

```
01: import numpy as np
02:
03: arr = np.zeros(5)
04: print(arr)
```

▣ shape를 여러 가지 형태로 생성 가능

```
01: import numpy as np
02:
03: arr = np.zeros((2,5))
04: print(arr)
05: arr = np.zeros((2,2,2))
06: print(arr)
```

소개와

▣ numpy.ones : 값을 1로 채운 ndarray배열 생성

```
01:         import numpy as np
02:
03:         arr = np.ones(5)
04:         print(arr)
05:         arr = np.ones((2,2))
06:         print(arr)
```

▣ numpy.full : 입력한 파라미터 값으로 채운 ndarray배열 생성

```
01:         import numpy as np
02:
03:         arr = np.full(5,-1)
04:         print(arr)
05:         arr = np.full((2,2),-1)
06:         print(arr)
```

소개와

- ▣ `numpy.eye` : 사이즈 값을 파라미터로 받아 대각선으로 1을 채우고 나머지는 0을 채운 2차원 ndarray 배열 생성

```
01: import numpy as np
02:
03: arr = np.eye(5)
04: print(arr)
```

소개와

- ▣ numpy.random 클래스의 random 메소드 : shape를 파라미터로 입력하면 ndarray를 반환
 - 머신러닝에서 랜덤 노이즈나 가중치의 초기값 등으로 활용

```
01:         import numpy as np
02:
03:         arr = np.random.random(5)
04:         print(arr)
05:         arr = np.random.random((2,2))
06:         print(arr)
```

변형

- numpy에서는 형태 변형이나 배열 추출에 관한 메소드를 제공
 - ▣ arange : 패턴이 있는 배열을 생성하면 1차원 배열로 생성
 - ▣ ndarray.reshape : arange로 생성된 배열을 다차원 배열로 변환
 - 파라미터로 shape를 입력하면 호출한 ndarray를 shape 형태로 재구성

```
01:      import numpy as np
02:
03:      arr = np.arange(10)
04:      print(arr)
05:      reshaped = arr.reshape(2,5)
06:      print(reshaped)
```

변형

□ 파이썬과 같이 특정 배열 잘라내기 가능

```
01:         import numpy as np
02:
03:         arr = np.arange(10)
04:         print(arr)
05:         s1 = arr[5]
06:         print(s1)
07:         s2 = arr[:5]
08:         print(s2)
09:         s3 = arr[2:5]
10:         print(s3)
```

변형

- 다차원 배열에서 특정 원소, 행, 열의 범위로 지정으로 추출가능

```
01:         import numpy as np
02:
03:         arr = np.arange(10).reshape(2,5)
04:         print(arr)
05:         s1 = arr[1,4]
06:         print(s1)
07:         s2 = arr[1:]
08:         print(s2)
09:         s3 = arr[0,:]
10:         print(s3)
11:         s4 = arr[:, 0]
12:         print(s4)
```

내용 정리

- Numpy : 벡터 및 텐서 연산에 편리한 기능들을 사용 가능.
 - ▣ 선형 대수 계산에 유용한 패키지
- Numpy 설치 명령어 : `pip3 install numpy`
- ndarray : 빠른 연산을 위한 Numpy의 배열 타입
- `array(list)`: 파이썬의 리스트, 튜플, 배열 타입을 ndarray타입으로 생성
- `zeros(shape)`: 값을 0으로 채운 shape 형태로 ndarray를 생성

내용 정리

- `ones(shape)`: 값을 1으로 채운 shape 형태로 ndarray를 생성
- `full(shape, n)`: 값을 n으로 채운 shape 형태로 ndarray를 생성
- `eye(size)`: size x size 크기의 ndarray를 대각선은 1, 나머지는 0으로 채워 생성
- `random.random(shape)`: 값을 랜덤으로 채운 shape 형태로 ndarray를 생성

내용 정리

- `arange(start, stop, step):`
 - ▣ `start` 부터 `stop`까지 `step` 만큼 증감시킨 값을 갖는 `ndarray`를 생성
 - ▣ `start` 파라미터만 입력 : 0부터 `start` 까지 1씩 증감하는 값으로 생성
- `reshape(shape): ndarray를 shape 형태로 재구성`
- `Numpy 배열 추출, 잘라내기 :`
 - ▣ 파이썬의 리스트를 추출, 잘라내기 방법과 동일

연습문제

- 문제 7. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import numpy as np
02:
03:         arr = np.array([[1,2,3],[4,5,6]])
04:
05:         print(arr.ndim)
06:         print(arr.shape)
```

연습문제

- 문제 8. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import numpy as np
02:
03:         arr = np.zeros((3,3))
04:
05:         print(arr)
```

연습문제

- 문제 9. 다음과 같은 출력을 하는 코드를 작성해보세요.

[출력]

[-2, 0, 2, 4, 6, 8, 10]

연습문제

- 문제 10. 다음 코드는 Cds센서를 이용하여 주변 밝기 값을 출력하는 코드입니다.

```
01:         import numpy as np
02:         from pop import Cds,delay
03:
04:         cds = Cds(7)
05:
06:         for i in range(10):
07:             value = cds.readAverage()
08:             print(value)
09:             delay(500)
```

연습문제

- ▣ A. 코드를 실행시켜 결과를 확인해 보세요.
- ▣ B. 10개의 data를 저장할 수 있는 ndarray배열을 0으로 초기화하여 생성하는 코드를 작성해보세요.
- ▣ C. Cds센서의 값 10개를 B에서 생성한 배열에 저장하는 코드를 작성해보세요.
- ▣ D. C에서 생성된 배열을 (2,5)의 shape으로 변경하고 이를 출력하는 코드를 추가해 보세요.
- ▣ E. D에서 변경된 배열 중 첫 행만 추출하여 출력해 보세요.