Data Processing

In this lesson, we will learn about data processing tools using arrays.

WE'LL COVER THE FOLLOWING

- Mean and standard deviation
- min and max
- Complex arrays
- Some useful functions

NumPy arrays provide diverse functionalities and tools to manipulate extract meaning from raw data.

Mean and standard deviation

Suppose you are measuring the peak value of current passing through a transformer every two hours, but due to temperature changes, there is a fluctuation in the peak value. In order to get a good representation of the data, we will need to calculate the **mean** value of all 12 readings. The mathematical formula for the mean is given below:

$$\overline{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$$

where a_i are elements in the data set, n is the total number of elements and $\overline{a}$ is the mean.

The data's **standard deviation** and **variance** will represent the number of fluctuations in the peak value. The mathematical formula for standard deviation is given below:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a - \overline{a})^2}$$

$$variance = \sigma^2$$

where a_i are elements in the data set, n is the total number of elements, $\overline{a}$ is the mean, and σ is the standard deviation. Variance is simply the square of the standard deviation.

In the example below, we will see how easily we can compute mean and standard deviation.

```
import numpy as np

current = np.array([2.2, 2.5, 2.5, 2.7, 2.1, 2.5, 2.6, 2.5, 3.0, 2.9, 2.3, 2.8])
print("mean", np.mean(current))     # to calculate mean
print("standard deviation", np.std(current))    # to calculate standard deviation
print("variance", np.var(current))      # to calculate variance
```

We have used the functions `mean()`, `std()` and `var()` from the `numpy` to calcu;ate mean, standard deviation and variance respectively.

min and max

To make sure the current does not exceed the upper limit and burn the equipment, it is necessary to see the maximum value of the current during the fluctuations.

We use the `max()` function from the `numpy` module to find the maximum value of current in the dataset. Let's see en example of this below:

```
import numpy as np

current = np.array([2.2, 2.5, 2.5, 2.7, 2.1, 2.5, 2.6, 2.5, 3.0, 2.9, 2.3, 2.8])
print("maximum value:", np.max(current))     # to calculate maximum value
```

To make sure the current does not fall below the lower limit and that the equipment keeps functioning normally, it is necessary to see the minimum value of the current during the fluctuations.

We use the `min()` function from the `numpy` module to find the minimum value of current in the dataset. Let’s look at an example of this below:

```
import numpy as np

current = np.array([2.2, 2.5, 2.5, 2.7, 2.1, 2.5, 2.6, 2.5, 3.0, 2.9, 2.3, 2.8])
print("minimum value:", np.min(current))     # to calculate minimum value
```

Complex arrays

Arrays with complex numbers are commonly used by engineers when working with signal processing. It is useful to be able to extract the real and imaginary parts of a complex array. This can be done by using `.real` and `.imag` with the array. This is similar to what we learned here.

Suppose there is an array of complex number named `arr`

- `arr.real` will return an array with the real parts of `arr`.
- `arr.imag` will return an array with the imaginary parts of `arr`.

```
import numpy as np

arr = np.array([1+3j, 2+5j, 3+1j, 8-2j])
print("Real part of arr =", arr.real)
print("Imaginary part of arr = ", arr.imag)
```

Some useful functions

Some other useful functions are:

- `sum` - returns the sum of all elements in the array
- `prod` - returns the product of all elements in the array
- `cumsum` - returns an array with the cumulative sum of all elements in the array
- `cumprod` - returns an array with the cumulative product of all elements in the array

```
import numpy as np

v = np.arange(1, 10)


print("Sum:", np.sum(v))
print("Product:", np.prod(v))
print("Cumulative Sum:", np.cumsum(v))
print("Cumulative Product:", np.cumprod(v))
```

In the next lesson, we will learn about some smart programming tools for arrays.