How can I export a NumPy array to a CSV file?

Authored by

stats writer

May 11, 2024

Exporting a NumPy array to a CSV file allows for the data to be easily shared and analyzed by others. This process involves using the NumPy library's built-in function "np.savetxt()" to convert the array into a CSV file format. The function allows for customizable parameters such as delimiter, header, and precision to be specified. Once the array is converted, it can be saved as a CSV file using the "np.savetxt()" function. This allows for seamless integration of NumPy arrays into data analysis and visualization tools.

Export a NumPy Array to a CSV File (With Examples)

You can use the following basic syntax to export a NumPy array to a CSV file:

```
import numpy as np

#define NumPy array
data = np.array(,,])

#export array to CSV file
np.savetxt("my_data.csv", data, delimiter=",")
```

The following examples show how to use this syntax in practice.

Example 1: Export NumPy Array to CSV

The following code shows how to export a NumPy array to a CSV file:

```
import numpy as np
```

#define NumPy array
data = np.array(, , , ,])

#export array to CSV file
np.savetxt("my_data.csv", data, delimiter=",")

If I navigate to the location where the CSV file is saved on my laptop, I can view the data:

	1.0000000000000000e+00	2.0000000000000000e+00	3.0000000000000000e+00
1	4.0000000000000000e+00	5.0000000000000000e+00	6.0000000000000000e+00
2	7.0000000000000000e+00	8.0000000000000000e+00	9.0000000000000000e+00
3	1.0000000000000000e+01	1.1000000000000000e+01	1.2000000000000000e+01
4	1.3000000000000000e+01	1.4000000000000000e+01	1.5000000000000000e+01

Example 2: Export NumPy Array to CSV With Specific Format

The default format for numbers is "%.18e" - this displays 18 zeros. However, we can use the fmt argument to specify a different format.

For example, the following code exports a NumPy array to CSV and specifies two decimal places:

import numpy as np

#define NumPy array

```
data = np.array(, , , , ])

#export array to CSV file (using 2 decimal places)
np.savetxt("my_data.csv", data, delimiter=",", fmt="%.2f")
```

If I navigate to the location where the CSV file is saved, I can view the data:

	1.00	2.00	3.00
1	4.00	5.00	6.00
2	7.00	8.00	9.00
3	10.00	11.00	12.00
4	13.00	14.00	15.00

Example 3: Export NumPy Array to CSV With Headers

The following code shows how to export a NumPy array to a CSV file with custom column headers:

```
import numpy as np

#define NumPy array
data = np.array(, , , , ])

#export array to CSV file (using 2 decimal places)
np.savetxt("my_data.csv", data, delimiter=",",
```

fmt="%.2f",
header="A, B, C", comments="")

If I navigate to the location where the CSV file is saved, I can view the data:

	A	B	C
1	1.00	2.00	3.00
2	4.00	5.00	6.00
3	7.00	8.00	9.00
4	10.00	11.00	12.00
5	13.00	14.00	15.00

Note: You can find the complete documentation for the numpy.savetxt() function .

The following tutorials explain how to perform other common read and write operations in Python: