Can arrays of size larger than 1 be converted to Python scalars?

Authored by

stats writer

July 2, 2024

The conversion of arrays of size larger than 1 to Python scalars refers to the process of transforming an array data structure containing multiple elements into a single value in Python. This conversion is not possible as Python scalars are limited to representing single values only, while arrays can hold multiple values. Attempting to convert an array of size larger than 1 to a Python scalar will result in an error. Therefore, it is not feasible to convert arrays of size larger than 1 to Python scalars.

Fix: Only size-1 arrays can be converted to Python scalars

One error you may encounter when using Python is:

TypeError: only size-1 arrays can be converted to Python scalars

This error occurs most often when you attempt to use np.int() to convert a NumPy array of float values to an array of integer values.

However, this function only accepts a single value instead of an array of values.

Instead, you should use x.astype(int) to convert a NumPy array of float values to an array of integer values because this function is able to accept an array.

The following example shows how to fix this error in practice.

How to Reproduce the Error

Suppose we create the following NumPy array of float values:

```
import numpy as np

#create NumPy array of float values
x = np.array()
```

Now suppose we attempt to convert this array of float values to an array of integer values:

```
#attempt to convert array to integer values
np.int(x)
```

TypeError: only size-1 arrays can be converted to Python scalars

We receive a TypeError because the np.int() function only accepts single values, not an array of values.

How to Fix the Error

In order to convert a NumPy array of float values to integer values, we can instead use the following code:

#convert array of float values to integer values
x.astype(int)array()

Notice that the array of values has been converted to integers and we don't receive any error because the astype() function is able to handle an array of values.

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

Additional Resources

The following tutorials explain how to fix other common errors in Python: