What does “Fix: invalid value encountered in true_divide” mean and how can it be resolved?

Authored by

stats writer

May 12, 2024

The statement "Fix: invalid value encountered in true_divide" refers to encountering an error while performing a true division operation. This error occurs when the divisor (the number being divided by) is equal to zero or when both the dividend (the number being divided) and divisor are complex numbers. To resolve this issue, the user should check their input values and make sure that the divisor is not equal to zero. If the issue persists, they can try converting the input values into real numbers before performing the division. Additionally, checking for any potential coding errors or using alternative division methods can also help resolve this error.

Fix: invalid value encountered in true_divide

One warning you may encounter when using NumPy is:

RuntimeWarning: invalid value encountered in true_divide

This warning occurs when you attempt to divide by some invalid value (such as NaN, Inf, etc.) in a NumPy array.

It's worth noting that this is only a warning and NumPy will simply return a nan value when you attempt to divide by an invalid value.

The following example shows how to address this warning in practice.

How to Reproduce the Error

Suppose we attempt to divide the values in one NumPy

array by the values in another NumPy array:

```
import numpy as np

#define NumPy arrays
x = np.array()
y = np.array()

#divide the values in x by the values in ynp.divide(x, y)

array()

RuntimeWarning: invalid value encountered in true_divide
```

Notice that NumPy divides each value in x by the corresponding value in y, but a RuntimeWarning is produced.

This is because the last division operation performed was zero divided by zero, which resulted in a nan value.

How to Address this Warning

As mentioned earlier, this RuntimeWarning is only a warning and it didn't prevent the code from being run.

However, if you'd like to suppress this type of warning then you can use the following syntax:

```
np.seterr(invalid='ignore')
```

This tells NumPy to hide any warning with some "invalid" message in it.

So, if we run the code again then we won't receive any warning:

```
import numpy as np

#define NumPy arrays
x = np.array()
y = np.array()

#divide the values in x by the values in ynp.divide(x, y)

array()
```

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