

Fundamentals Of Deep Learning

Numpy

Goals

1. Understand why to use Numpy
2. Understand Numpy data structures (Numpy arrays)
3. Perform Matrix operations on these numpy arrays
4. Understand how to use Boolean logical operations
5. Use Numpy to find min,max,variance,standard deviation
6. Reshape arrays
7. Generate random values using np.random and linspace

Data Preprocessing

In any machine learning or deep learning problem we need to understand the data we are dealing with. This is a very crucial step. Without understanding what data we are dealing with it's like playing cricket while closing your eyes:-)

Since mistakes, redundancies, missing values, and inconsistencies all compromise the integrity of the set, you need to fix all those issues for a more accurate outcome. Imagine you are training a Machine Learning algorithm to deal with your customers' purchases with a faulty dataset. Chances are that the system will develop biases and deviations that will produce a poor user experience.

Thus, before using that data for the purpose you want, you need it to be as organized and “clean” as possible. There are several ways to do so, depending on what kind of problem you’re tackling

Numpy

NumPy is one of the most powerful Python libraries. Numpy packages all the important essentials required for Data Science, ML and Deep Learning!

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Numpy Arrays

Importing Numpy as:

```
import numpy as np
```

How do we create arrays and matrices in numpy?


Simplest way to create an array in Numpy is to use Python List

```
myPythonList = [1, 9, 8, 3]
```

To convert python list to a numpy array by using the object `np.array`.

```
numpy_array_from_list = np.array(myPythonList)
```

To display the contents of the list



```
numpy_array_from_list
```

Output

```
array([1, 9, 8, 3])
```

In practice, there is no need to declare a Python List. The operation can be combined.

```
a = np.array([1, 9, 8, 3])
```

NOTE: Numpy documentation states use of `np.ndarray` to create an array. However, this the recommended method

You can also create a numpy array from a Tuple

Import statistical functions:

```
np.min()
```

```
np.max()
```

```
np.std()
```

```
np.mean()
```

```
np.var()
```

References:

<https://www.guru99.com/numpy-array.html>

<https://becominghuman.ai/an-essential-guide-to-numpy-for-machine-learning-in-python-5615e1758301>

https://www.learnpython.org/en/Pandas_Basics

<https://www.w3resource.com/numpy/manipulation/reshape.php>

<https://www.geeksforgeeks.org/numpy-reshape-python/>

https://docs.scipy.org/doc/numpy/reference/generated/numpy.logical_and.html

https://docs.scipy.org/doc/numpy/reference/generated/numpy.logical_or.html

https://docs.scipy.org/doc/numpy/reference/generated/numpy.logical_and.html

https://docs.scipy.org/doc/numpy/reference/generated/numpy.logical_not.html

https://chrisalbon.com/python/basics/generating_random_numbers_with_numpy/

■ ■ ■