Learning Numpy

Yankun (Alex) Meng

Lecture Outline

- **Introduction and Basics**
- Array Manipulation
- Calculus / Statistical Functions

What is NumPy?

NumPy is a package for dealing with arrays that's very fast and efficient.

Creating array:

```
arr1 = np.array([3, 5, 7, 3]) # convert list to array
arr2 = np.zeros(10) # array with 10 zeroes
arr3 = np.ones(10) # array with 10 ones
arrR = np.random.random(10) # sample 10 numbers from uniform distribution
arrG = np.random.randn(10) # sample 10 numbers from normal distribution
```

Two most common functions

linspace - values equally spaced apart given the number of values

```
arr = np.linspace(0, 10, 100) # from 0 to 10, 100 values
```

arrange - values equally spaced apart given the spacing

```
arr = np.arrange(0, 10, 0.02) # from 0 to 10, spacing of 0.02
```

Lecture Outline

- Introduction and Basics
- **Array Manipulation**
- Calculus / Statistical Functions

Common Array Operation

Element-wise operation

```python
arr + 2      # addition
arr - 5      # subtraction
2*arr        # multiplication
1/arr        # division
arr % 5      # Modulus
```

Let’s try to do the same element-wise operation but using functions:

```python
def f(x):
    return x**2 * np.sin(x) / np.exp(-x)

x = np.linspace(0, 10, 100)
y = f(x) # Element-wise application of f
```

Common Plotting

Most basic Plotting

```python
x = np.linspace(0, 1, 100)  # x-axis definition
y = x**2                    # element-wise x^2
plt.plot(x, x**2)           # plot(x, y)
plt.show()                  # show the output
```

Histogram

```python
plt.hist(arr) # plotting the array as histogram
```

Indexing and Slicing

Normal Indexing

```
arr = np.array([2, 4, 6, 8, 10])
arr[2]   # return 6
arr[2:]  # return 6, 8, 10 (including index 2)
arr[:-2] # return 2, 4, 6 (exluding index -2)
arr[1:2] # return 4
```

Boolean Indexing

```
arr > 5 # return a boolean array with true or false as elemenets
arr[arr>5] # filtering ⟶  return the elements that are true
```

Using Boolean Indexing to filter data

```
names = np.array(['Jim', 'Luke', 'Josh', 'Pete'])
first_letter_j = np.vectorize(lambda s: s[0])(names) == 'J'
print(names[first_letter_j]) # gets the names that have first letter as J
```

Now what is this doing?

What is that doing?

```python
first_letter_j = np.vectorize(lambda s: s[0])(names) == 'J'
```

Lambda Function

```python
f = lambda s: s[0]
```

This is a function that says give me a string and I'll give you the first character of the string. Note that `String` is not an array in python, but it supports the same indexing operations as arrays.

What is np.vectorize?

It creates a for-loop, applies the function `func` to every element in the array `arr`, and return the result as an array.

```python
np.vectorize(func)(arr) # this is an array
```

Lecture Outline

- Introduction and Basics
- Array Manipulation
- **Calculus / Statistical Functions**

Common Statistics Functions

```python
arr = 2*np.random.randn(10000) + 10 # 2 times std dev with a shift of 10 to the right\
np.mean(arr) # approx 10
np.std(arr) # approx 2
np.percentile(arr, 80) # 80% numbers less than this number
```

Integrals and Derivatives

```python
x = np.linspace(1, 10, 100)
y = 1/x**2 + np.sin(x)
plt.plot(x, y)
```

Derivatives

```python
dydx = np.gradient(y, x)
```

Integrals

```python
np.cumsum([1, 2, 3, 4]) # return [1,3,6,10]
y_int = np.cumsum(y) * (x[1] - x[0]) # int y dx
```

Resources Used

- [1] NumPy Tutorial
- [2] 100 numpy practice problems