Pytorch

numpy

Create array/Matrix

```
import numpy as np

array = np.array([[1,2,3],
                  [2,3,4]],dtype=np.float)

# create a 3x4 matrix with all 0
b = np.zeros((3,4))

c = np.ones((3,4))

d = np.empty((3,4))

# same with range
e = np.arange(10,20,2)

f = np.arange(12).reshape((3,4))

g = np.linspace(1,10,20).reshape

h = np.random.random((2,4))
```

Attributes of array

The number of dimesion of array

```
array.ndim
```

The shape of array

```
array.shape
```

The number of element in array

```
array.size
```

Matrix Operations

Determine the size of array elements

```
a = np.array([10,20,30,40])
b = np.arange(4)

# return True if element fulfill the condition
print(b<3)
```

Array addition and subtraction

```
c = a-b
print(c)
```

Array multiplication

```
e = np.array([[1,1],[0,1]])
f = np.arange(4).reshape((2,2))

# simple multiplication
g = e * f

# dot multiplication
g_dot = np.dot(e,f)

g_dot_2 = e.dot(f)
```

Sum and find the maximum and minimum

```
h = np.random.random((2,4))

# axis=1 means Cross-column operation(may change the number of column)
print(np.sum(h,axis=1)

print(np.min(h))

print(np.max(h))
```

Find specific elements

```
A = np.arange(2,14).reshape((3,4))

# Returns the index of the minimum value of the array
np.argmin(A)
no.argmax(A)

# Returns the mean of each row in the matrix
np.mean(A,axis=1)

np.average(A)

np.median(A)
```

Get new Matrix

```
# Array of elements accumulated
np.cumsum(A)

# In each line, the difference between every two numbers
np.diff(A)

# Returns the row and column indexes of nonzero elements in the array
np.nonzero(A)

# Sort each line
np.sort(A)
```

```
12
13 # Return transpose matrix (2 methods)
14 np.transpose(A)
15 A.T
16
17 # Change the elements in the matrix less than 5 to 5,
18 # the elements greater than 9 to 9, and the other unchanged
19 np.clip(A,5,9)
```

Matrix slicing

```
1  A = np.arange(3,15).reshape((3,4))
2
3  # Return specific row
4  print(A[2])
5  print(A[2,:])
6
7  # Return specific element
8  print(A[1,1])
9
10 # Print the Matrix line by line
11 for row in A:
12     print(row)
13
14 # Print the Matrix column by column using A.T
15 for column in A.T:
16     print(column)
17
18 # Convert matrix into array
19 A.flatten()
20
21 # A.flat works as an Iterator
22 # This for loop will print all the element in the matrix
23 for item in A.flat:
24     print(item)
25
```

Add dimension to matrix

```
1  A = np.array([1,1,1])   # The shape is (3,)
2  B = np.array([2,2,2])
3
```

```
print(A[np.newaxis,:].shape) # The shape is (1,3,)

print(A[:,np.newaxis].shape) # The shape is (3,1,)
```

Matrix Merging

```
# Vertical stack
C = np.vstack((A,B))

# Horizontal stack
D = np.hstack((A,B))

# concatenate

# axis=0, change the number of row
E = np.concatenate((A,B,B,A),axis = 0)

# axis=1, change the number of column
F = np.concatenate((A,B),axis = 1)
```

Matrix split

```
A = np.arange(12).reshape(3,4)

# split the matrix into 2 part
# axis=1 means based on column split the matrix
np.split(A,2,axis=1)

# axis=0 means based on row split the matrix
np.split(A,3,axis=0)

# Unequal split
np.array_split(A,3,axis=1)

# Vertical split
np.vsplit(A,3)

# Horizontal split
np.hsplit(A,2)

```

Copy

```
a = np.arange(4)

# Assign A to B, B is related to A, changing A will change B
b = a

# Assign A to B, but not related, changing A will not change B
b = a.copy()
```