Linear Algebra – Đại Số Tuyến Tính:

1. Cách Import?

from numpy import linalg as la

1. Trả Về Eigen Vector Và Eigen Value Của Ma Trận?

<Eigen Value>, <Eigen Vector>] = la.eig(<Ma Trận Vuông>)

* <Eigen Vector> là 1 ma trận mà các cột của nó là Eigen Vector của

<Ma Trận Vuông>, các Eigen Vector này có độ dài = 1

* <Eigen Value> là 1 dãy các Eigen Value tương ứng với mỗi Eigen Vector trong <Eigen Vector>

Numpy:

1. Cách Import?

import numpy as np

1. Nhân Từng Phần Tử Trong Mảng?

<Kết Quả> = np.multiply(<Mảng 1>, <Mảng 2>)

* Ví dụ

foo = np.multiply([1, 2, 3], [4, 5, 6])

* “foo” có giá trị là [4, 10, 18]

1. Chồng Các Numpy Array Lên Nhau?

<Numpy Array Chồng> = torch.stack(<Các Numpy Array>, <Chiều Quay Mặt>)

* Mặc định

|  |  |
| --- | --- |
| <Chiều Quay Mặt> | 0 |

* Hoạt động y chang cách chồng các Tensor lên nhau

1. Liệt Kê Tất Cả Các Giá Trị Khác Nhau Của 1 Numpy Array Theo Thứ Tự Tăng Dần?

<Các Giá Trị> = np.unique(<Numpy Array>)

* Ví dụ

foo = np.unique(bar)

* Ta có

|  |  |
| --- | --- |
| bar | [[1, 1, 2, 2, 5, 4,  9, 9, 4, 4, 3, 2]] |
| foo | [1, 2, 3, 4, 5, 9] |

1. Tất Cả Kiểu Dữ Liệu?

* Số nguyên

np.<Có Dấu Không>int<Số Bit>

* <Số Bit> = 8, 16, 32, 64
* <Có Dấu Không> = u thì không dấu, mặc định có dấu
* Số Floating Point

np.float<Số Bit>

* <Số Bit> = 16, 32, 64, 128
* Logic

bool

Numpy Array:

1. Numpy Array Chỉ Chạy Trên CPU?

* Đúng

1. Cách Nhanh Nhất Để Đổi Kiểu Dữ Liệu Của 1 Numpy Array?

<Numpy Array>.dtype = <Kiểu Dữ Liệu>

* Ví dụ

foo.dtype = float

* Để Copy ra 1 Numpy Array mới

<Numpy Array Mới> = <Numpy Array>.astype(<Kiểu Dữ Liệu>)

1. Chuyển Đổi PIL Image Sang Numpy Array?

<Numpy Array> = np.array(<PIL Image>)

1. Cách Hoán Vị Shape Của 1 Numpy Array?

<Numpy Array Có Shape Hoán Vị> = <Numpy Array Gốc>.transpose(<Hoán Vị>)

* Hoạt động đéo khác gì hoán vị Shape của 1 Tensor

1. Cách Gộp Tổng Giá Trị Vào 1 Trục?

<Numpy Array Ép> = <Numpy Array Gốc>.sum(

<Chiều Không Gian>, keepdims = True

)

* Hoạt động y chang như trong Torch

Random – Ngẫu Nhiên:

1. Cách Import?

from numpy import random

1. Tạo Numpy Array Có Giá Trị Ngẫu Nhiên Thuộc Phân Phối Chuẩn Tắc?

random.randn(<Shape>)

* Hoạt động đéo khác gì việc tạo Tensor với phân phối chuẩn tắc

Numpy

.c\_[Array, Array]:

Concat 2 Array theo chiều ngang

Ex:

numpy.c\_ =

.stack(Arrays, Axis):

Stack các Array lên nhau

Ex:

numpy.stack((,)) =

.concatnate(Arrays, Axis):

Concat các Array theo Axis

Ex:

numpy.concatnate((), axis = 0)

=

.linspace(Low, High, Num):

Trả về Array nội suy tuyến tính từ Low đến High gồm Num phần tử (đã bao

gồm Low và High)

Ex:

numpy.linspace(2, 6, 5) = [2, 3, 4, 5, 6]

.meshgrid(X, Y):

Trả về 1 Array gồm 2 phần tử, một là 2D Array có các Row = X, hai là 2D

array có các Column = Y, kết hợp mỗi phần tử tương ứng trong X với Y tạo

thành Grid

Ex:

X = [1, 2, 3], Y = [4, 5, 6]

X, Y = numpy.meshgrid(X, Y)

X = , Y = , (X, Y) =

array

.ndim :

Trả về số Dimension của Array

Ex:

[[1]].ndim = 2

.repeat(Time, Axis):

Lặp lại Array Time lần theo Axis

Ex:

numpy.repeat(, 2, axis = 0) =

.argsort():

Trả về Array gồm các Index đã được sắp xếp sao cho giá trị Array tăng dần

Ex:

array = [3, 7, 2, 4, 8]

indices = array.argsort() = [2, 0, 3, 1, 4]

array[indices] = [2, 3, 4, 7, 8]

linalg

.norm(Matrix || Vector):

Trả về Frobenius Norm của Matrix hoặc Two Norm của Vector

Ex:

numpy.linalg.norm() =

.eig(Matrix):

Trả về Array gồm 2 phần tử, một là Eigen Values List, hai là Normalized

Eigen Vectors Matrix, không theo thứ tự

Ex:

numpy.linalg.eig() =

.svd(Matrix):

Trả về Tuple gồm 3 phần tử, một là Left Singular Vectors Matrix, hai là

Singular Values Vector, ba là Transposed Right Singular Vectors Matrix

Ex:

numpy.linalg.svd () =

random

.exponential(Scale, Size):

Trả về Array là Samples từ Exponential Distribution với độ rộng Scale

Ex:

numpy.random.exponential(5, (3, )) =