

Introducción a NumPy

NumPy Introduction

Autor: **Sebastián Arroyave Ramírez**

IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: s.arroyave@utp.edu.co

Introducción — python para los estudiosos de los sistemas ha sido uno de los lenguajes mas aceptados desde su entrada a escena, por esto es que muchas universidades en los programas de ingeniería de sistemas y la computación han dictado sus clases enseñando este lenguaje de programación, ya que es un lenguaje sencillo para un principiante ya que no cuesta nada aprenderlo, gracias a que hay muchas herramientas que le permiten a uno aprender de forma rápida y entendible este lenguaje de programación. Python no solo es un lenguaje de programación que solo utiliza una sintaxis para programar, a python se le puede llegar añadir muchas librerías que le ayudan no solo a programar funciones simples sino que también le ayuda a programar funciones matemáticas o de una complejidad matemática complicada, NumPy es una librería que le ayuda al programador para hacer funciones matemáticamente complejas haciéndolas así más fáciles la tarea de programar dichas funciones algebraicas.

Conceptos básicos de NumPy

Cuando hablamos de comenzar a programar con NumPy lo primero que debemos saber es que tenemos que importar la librería numpy para que así el editor donde estemos trabajando pueda saber que estamos realizando funciones que con numpy se pueden programar y luego ejecutar bien.

import NumPy as np

Así se vería al momento de importar la librería numpy para poder empezar a trabajar con ella, se puede apreciar que al lado de numpy se encuentran dos palabras desconocidas para el público poco conocedor lo que hace referencia con el “**as np**” es que se está abreviando la palabra numpy para que cuando se haga un llamado o se guarde alguna función, no se tenga que escribir la palabra numpy sino solo las letras “**np**”.

Luego después de esto se puede comenzar, en el caso de este artículo, en crear un vector (array) del cual se va a poder hacer mucho con la ayuda de numpy, lo primero que debemos de hacer es una función la cual no cree el vector.

a = np.array([0,1,2,3,4,5])

Aquí podemos ver que creamos un vector de 6 elementos los cuales van desde el 0 a 5 y los guardamos en la variable a y lo mandamos a imprimir en pantalla. Ya si uno quiere que se calcule el número de dimensiones que tiene ese vector se puede utilizar la función de “**a.ndim**” y le mostrará las ya dichas dimensiones que tiene ese array, también si se quiere hacer el respectivo cálculo de cuántos elementos contiene el vector se puede utilizar la función “**a.shape**”.

Uno se preguntará ¿con numpy puedo convertir ese vector en una matriz? la respuesta es sí, ya que con unas simples funciones uno le puede llegar a cambiar la forma a ese vector, para poder hacer el dichoso cambio de forma del array se usa la siguiente función “**a.reshape**” ya teniendo escrita esta función podremos darle el número de filas y columnas que nosotros queramos, luego de esto podemos imprimir esta función pero antes debemos de verificar los cambios.

NOTA: cabe recordar que como el array se le está cambiando la estructura se tiene que guardar en otra variable la cual es b.

Luego si se quiere llegar a cambiar algún elemento del array lo que se tiene que hacer es que por medio de una lista la cual es usando [] estas llaves se dice la fila y columna que se quieren cambiar. Sabiendo que el array b se construyó con la base del a este array a se cambiará también.

Si lo que se quiere es conservar el array o vector original que es el que se mostró antes se puede utilizar la siguiente función.

c = a.reshape((3,2)).copy()

Lo que podemos apreciar con esta función es que la guardamos en una variable c y lo que hace es que la reestructuración que le hizo al array le crea una copia. También a este array copiado se le puede cambiar algún elemento e imprimirlo.

Las operaciones en este tipo de funciones no podían faltar, nada más y nada menos que podemos hacer cualquier tipo de operación matemática ya sea una multiplicación sencilla

utilizando “*”, escribiendo por cuanto se multiplicara la variable que le escribamos, tambien podemos llegar a elevar todos los numeros del array poniendo “**” para asi que nos muestre en pantalla el resultado de elevar el numero a la potencia que nosotros queramos.

NumPy nos permite que con el array definido podamos iterar todos los elementos, esto se hace con la siguiente function.

print(a>4, '\n')

Lo que nos esta diciendo esta function es que cuando aplicamos la condicion de que si “a>4” todos los elementos que sean mayores a 4 lanzara un True pero si no se cumple esta function lanzara un False, luego podemos agregarle lo que se nos venga en mente como cambiar a 1 a todos los numeros mayores que 4 o que los numeros iguales a 1 se cambien por otro numero, se puede hacer la cantidad de cambios que uno desee.

NumPy nos da la posibilidad de poder hacer muchas cosas en las cuales se encuentra poder encontrar valores erroneos los cuales se pueden presentar en el array. “np.NaN” si la llegamos a poner en nuestro array y lo mandamos a imprimir nos aparecerá la palabra “nan”, ya luego tendremos funciones las cuales nos permitirán verificar si llegan a existir valores nan como “np.isnan” tambien calcular valores que NO sean nan tal cual lo permite la función “~np.isnan” y por ultimo y no menos importante poder hacer el calculo promedio de los valores que NO son nan “np.mean(~np.isnan)”.

Proyecto de Machine Learning

el enunciado es el siguiente:

```
# Una empresa vende el servicio de proporcionar algoritmos
# de aprendizaje automático a través de HTTP.
# Con el éxito creciente de la empresa, aumenta la demanda de una mejor
# infraestructura para atender todas las solicitudes web entrantes.
# No queremos asignar demasiados recursos, ya que sería demasiado costoso.
# Por otro lado, perderemos dinero si no hemos reservado suficientes recursos
# para atender todas las solicitudes entrantes.
# Ahora, la pregunta es, ¿cuándo alcanzaremos el límite de nuestra
# infraestructura actual, que se estima en 100.000 solicitudes por hora?
# Nos gustaría saberlo de antemano cuando tenemos que solicitar servidores
# adicionales en la nube para atender todas las solicitudes
# con éxito sin pagar por las no utilizadas.
```

Para poder empezar hacer un Proyecto de machine learning lo que principalmente necesitamos son los datos del estudio que deseamos hacer, los datos al no ser generados por alguna function sino que se traen de otra parte diferente se pueden mostrar en el programa gracias a la siguiente function.

**data = np.genfromtxt(“web_traffic.tsv”,
delimiter=“\t”)**

ya con esta function se van a mostrar todos los datos que tengamos en ese archivo de texto que pusimos en la function, ya si uno es calculador y quiere ver cuantos datos hay en ese archivo de texto puede utilizar la funcion “data.shape”.

Ya si lo que se quiere es dividir el array que tenemos en dos vectores podemos utilizar es “x = data[:, 0]” y el mismo caso si fuera con un segundo vector y, luego como se hace en todos los casos se manda a imprimir la function en pantalla.

Para finalizar la fase de analisis de datos, se pueden utilizar muchas de las funciones ya mencionadas como la function de las dimensiones del array, tambien podemos utilizar la function de buscar cuales elementos son nan dentro del array y muchas otras mas.

Graficacion de los datos

lo primero que debemos de hacer para poder graficar es importar la libreria.

import matplotlib.pyplot as plt

con esta libreria ya podremos permitir al programa graficar los datos anteriormente analizados y asi poder ver como se comportan segun lo que hallamos investigado, para hacer los dibujos de la grafica lo que necesitaremos es funciones como scatter, title, xlabel, ylabel, xticks y muchas otras mas, que nos permitan ponerle un titulo al grafico, que tamaño queremos que se muestren los puntos que se dibujan en este grafico.

NOTA: a todas estas anteriores funciones de graficacion ya mencionadas se le tiene que poner antes de sus nombre el “plt.”