Computación Blanda

Soft Computing

Autor: Angie Paola Villada Ortiz, Luz Adriana Quitumbo Santa
IS&C, Universidad Tecnológica de Pereira, Pereira Colombia
Correo-e:

paola.villada@utp.edu.co,Adriana.quitumbo@utp.edu.co

Resumen— Este documento se mostrarán algunas funciones que se implementaron con la librería numpy la cual significa "Numerical Python". Esta librería se puede utilizar en Python, sirve principalmente cuando se está aprendiendo Machine Learning, dado a que esta librería proporciona potentes estructuras de datos, donde se implementan matrices y matrices multidimensionales, por ende, se pueden acceder más rápido y los cálculos serán más eficientes. Ahora que se sabe para qué sirve la librería numpy, se utilizara algunas de sus funciones, primero se empezó creando un vector seguidamente se utilizó shape para saber el tamaño de los datos de ese vector. Y si se desea cambiar la estructura del vector se utiliza reshape. De igual modo se evidencio que en Python cuando un número es erróneo es decir sin ningún valor a este se le denomina NAN, estos datos no se deben tener en cuanta, en los cálculos, por eso con la librería numpy hay varias funciones para saber si en un vector determinado hay datos basura y eliminarlos.

Palabras clave— Numpy, Machine Learning, Array, Vector

Abstract— This document will show some functions that were implemented with the numpy library which means "numeric Python". This library can be used in Python, it is mainly used when learning Machine Learning, since this library provides powerful data structures, where matrices and multidimensional matrices are implemented, therefore, they can be accessed faster and the calculations will be more efficient . Now that we know what the numpy library is for, some of its functions will be used, first we started creating a vector, then shape was used to know the size of the data of that vector. And if you want to change the structure of the vector, you use reshape. In the same way, it was evident that in Python when a number is wrong, that is, without any value, it is called NAN, this data should not be taken into account, in the calculations, so with the numpy library there are several functions to know if in a given vector there is junk data and delete it.

Key Word— Numpy, Machine Learning, Array, Vector

I. INTRODUCCIÓN

Este articulo recoge las notas personales tomadas en clase durante la visualización del manejo de la librería numpy en la herramienta jupyter cuyo libro o interfaz es anaconda.

El aprendizaje fue respecto a la temática machine learning donde se evidencia cuatro aspectos fundamentales en el proceso de aprendizaje los cuales son:

1. Repasos de conceptos básicos de numpy.
2. Proyecto básico de machine learning.
3. Manejo de datos.
4. Graficación de datos.

Empezaremos por entender que es o para que sirve numpy en el editor de jupyter (anaconda).

Numpy es una extensión de Python y es uno de los paquetes de software que no debe perderse al aprender el aprendizaje automático, principalmente porque la estructura de datos de matriz proporcionada por la biblioteca tiene algunas ventajas sobre las listas normales de Python.

Algunos de los beneficios son: acceso más compacto, más rápido para leer y escribir artículos, más conveniente y más eficiente.

Numpy es un paquete de programa Python que significa "Python numérico". Es la biblioteca principal para la computación científica. Proporciona una estructura de datos poderosa para realizar matrices unidimensionales, matrices bidimensionales y matrices multidimensionales. Estas estructuras de datos aseguran el cálculo eficiente de la matriz.

El concepto básico de una matriz Numpy es que es un poderoso objeto de matriz N-dimensional, que tiene la forma de filas y columnas, en el que se almacenan varios elementos en sus respectivas ubicaciones de almacenamiento.

Donde se pude conocer las dimensiones de un array y los elementos que están dentro de el de una manera más fácil además de esto nos permite hacer un cambio de estructura de un array, se pueden modificar elementos que están dentro del vector bien sea por valores positivos o negativos y hacer múltiples operaciones con los elementos es decir elevar al cuadrado o multiplicar etc. La herramienta de numpy nos permite filtrar valores, sacar promedios de los datos ,tamaño y graficar.

Trabajar con Numpy es mucho más fácil y conveniente en comparación con las listas propias de Python, es por esta razón que es utilizado bastante en el desarrollo de los algoritmos de Machine Learning.

En los siguientes ítems se presenta el método de como implementar un array con la librería numpy y como jugar con los elementos que están dentro de él.

1.1 METODOLOGIA

Lo primero que se hizo es importar la librería numpy con el alias np (import numpy as np) después de esto se procede a crear un vector (a) que tiene seis elementos. Este vector se crea llamando la librería numpy es decir np con un operador punto y se hace un array con los elementos 0,1,2,3,4,5 por tanto la función queda de la siguiente manera: a = np.array([0,1,2,3,4,5]).

Después de haber creado el vector con seis elementos se imprime el vector (a) con un print(a, '\n').

El numero de dimensiones de un array (en este caso de seis elementos)se establece mediante un a.ndim y el número de elementos con a.shape cabe resaltar que (a) es un vector y que ndim, shape hacen parte de la librería de numpy.

Si se desea cambiar la estructura de array utilizamos el operador reshape (este término está definido por la librería de numpy);el cual me permite tener tres filas y dos columnas. En este ítem se puede evidenciar que el vector (b) está ligado con el vector (a) por tanto se procede a imprimir el vector (b) para observar el cambio de estructura del array creado inicialmente.

Luego se procede a mostrar el array con el número de elementos y dimensiones del vector (b) con shape, ndim.

El proceso para modificar un elemento de un array es haciendo referencia al nombre del array en decir (b),la fila, la columna y asignando un numero directo en este caso es de setenta siete. El cambio del elemento dentro del vector 8B9 se puede evidenciar imprimiendo (b).

Se debe tener en cuenta que el array (b) está ligado al array (a) es decir el vector (a) es el mismo(b) por tanto si se imprime notaremos que (a) se alteró.

Para que esto no ocurra utilizamos el operador copy en la función de la estructura del array b = a.reshape((3,2)) cabe resaltar que tres es de la fila y dos de la columna de la matriz o vector; por tanto, la función queda de la siguiente forma:

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

Hay que tener en cuenta que se crea el vector como una copia y se procede a imprimir el vector (c) para verificar que elementos tiene el vector dentro de él.

Se cambia el primer elemento en la posición cero de la columna y de la fila del vector (c) por un menos noventa nueve e imprimimos el array.

Es importante recalcar que las operaciones se propagan a lo largo del array como se presenta en la siguiente línea:

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

Si multiplicamos el array (d) por dos es decir (d*2);Los elementos dentro del array se modifican ya que cada elemento es multiplicado por dos por tanto el array cambia y lo mismo pasa si a los elementos dentro del array los elevamos por dos(d**2).

En machine learning los datos, con los que se trabajan llegan por medio de repositorios desconocidos y la información que entra llega con mucho ruido, basura (señal no deseada que se mezcla con la señal útil que se quiere transmitir). Y todas estas perturbaciones deben ser eliminadas.

En Python, si un número es erróneo (número sin asignación, sin valor) a este se le denomina como NAN. Por eso dentro de Python esta la constante np.NAN, es la que indica donde hay un valor basura, este no se debe formar parte de los cálculos. Para comprobar si hay algún dato basura dentro del vector, primero se crea el vector y se coloca un valor erróneo como se muestra a continuación:

c = np.array ([1, 2, np. NAN, 3, 4])

En la siguiente función ayuda mostrar la existencia de valores NAN, gracias a la constante de Python np.NAN. Este valor no se debe tener en cuenta en los cálculos.

print (np.isnan(c), '\n')

En esta instrucción sirve para verificar donde están los NAN. Con la función np.isnan, esta recibe un valor y determina si es NAN. Cuando no es NAN devuelve False y si es NAN devuelve True.

print(c[~np.isnan(c)], '\n')

Esta instrucción sirve para mostrar el promedio de los valores que no son NAN. Primero se imprime el vector, luego con la función np.isnan para mostrar los que no son NAN se le antepone el negador ~ para que el valor que salga False se convierta en True, dado a que Python trabaja con los valores verdaderos.

```
print(np.mean(c[~np.isnan(c)]))
```

Esta instrucción sirve para filtrar valores que no son NAN y sacarle promedio a esos datos que se filtraron utilizando la función np.mean.

Después de haber visto estas funciones, ahora se implementarán en el siguiente ejemplo: una empresa vende servicio para proporcionar algoritmos de aprendizaje automático a través de HTTP. Como ha sido un éxito, se aumentó la demanda de una mejor infraestructura para atender las solicitudes de web entrante.

Se requiere saber cuál es límite de la infraestructura actual, que se estima en 100.000 solicitudes por hora. Se requiere saber aproximadamente cuando se tiene que adquirir servidores adicionales a la nube para poder seguir atendiendo todas las solicitudes.

Para dar respuesta a esas preguntas anteriormente planteadas en ese ejemplo, se realizará un programa básico de machine learning en Python. Primero se obtiene los datos, los cuales se van a procesar. En la primera columna están ubicados el número de horas en que llego cada solicitud y en la segunda columna son el número de tareas que fueron ejecutadas.

```
data=np.genfromtxt("web_traffic.ts",delimiter="\t")
```

Primero se hace le llamado a la instrucción np.genfromtxt, esto hace que a partir del texto (es el archivo donde están recolectados los datos) genere información y luego con la instrucción delimiter="\t"), para delimitar y diferenciar un dato de otro dato.

```
print(data[:10], '\n')
```

En esta parte se muestran los datos con un rango de 10.

Para saber el tamaño de esos datos se puede utilizar la instrucción print (data.shape), en donde data, son los datos que están el archivo y shape es el tamaño.

Ahora se va a dividir el array(matriz), para crear dos vectores. La primera columna será x (número de horas) y la segunda y (filas). Para dividir el array se utiliza la siguiente instrucción:

```
x = data[: ,0]
y = data[: ,1]
```

Para mostrar los valores en x, y

```
print(x, '\n')
print(y, '\n')
```

para saber la dimensiones y los tamaños de esos vectores se utiliza las siguientes instrucciones.

```
print(x.ndim, '\n')
print(y.ndim, '\n')
print(x.shape, '\n')
print(y.shape)
```

para saber cuántos valores NAN hay en el vector y se utiliza la instrucción.

```
print(np.sum(np.isnan(y)))
```

Para comprimir los 2 vectores, para eliminar los valores NAN. Primero se muestran cuantos hay antes de que se compriman los 2 vectores.

```
print(x.shape, '\n')
print(y.shape, '\n')
```

Para comprimir los 2 vectores.

```
x = x[~np.isnan(y)]
y = y[~np.isnan(y)]
```

Se cuenta el número de elementos tanto de x como de y

```
print(x.shape, '\n')
print(x.shape, '\n')
```

Por último, se va a graficar x,y. se importa la librería para poder graficar, import matplotlib.pyplot as plt.

se va a hacer un gráfico donde se va a utilizar el vector x y el vector y. con un tamaño de figura de 10 pixeles.

```
Plt.scatter(x, y, s=10)
```

Para agregar título a la grafica

```
plt.title("Tráfico Web del último mes")
```

Las siguientes instrucciones son para poner en el eje x el tiempo y en el eje y la hora.

```
plt.xlabel("Tiempo")
```

```
plt.ylabel("Solicitudes/Hora")
```

Esta instrucción pinta todos los puntos y los agrupa por semanas.

```
plt.xticks([w*7*24 for w in range(10)],
['semana %i' % w for w in range(10)])
```

Esta instrucción es para acomodarse y no se salga de la pantalla.

```
plt.autoscale(tight=True)
```

Dibuja una cuadrícula punteada ligeramente opaca

```
plt.grid(True, linestyle='-', color='0.75')
plt.show()
```

1.2 RESULTADOS

Después de haber creado el vector (a) cuyos elementos dentro de él son: [0 1 2 3 4 5] podemos evidenciar el número de dimensiones del array (1) y el numero de elementos dentro del vector (6,).

Cuando cambiamos la estructura del vector por tres filas y dos columnas el resultado del vector es:

```
[[0 1]
 [2 3]
 [4 5]]
```

El número de dimensiones es de tres filas y dos columnas y el número de elementos de dos.

Al modificar un elemento dentro del array podemos evidenciar que en la posición cero de la columna y fila el numero dos dentro del vector es cambiado por un setenta y siete.

```
[[ 0 1]
 [77 3]
 [ 4 5]]
```

El resultado del vector (a) al imprimir es:

[0 1 77 3 4 5]

Podemos evidenciar que el resultado del vector (a) se alteró en el ítem anterior.

Para corregir esto se implementa una copia con la función c = a.reshape((3,2)).copy() por tanto proporciona el vector (a) donde se evidencia que no cambio ya que sigue con el mismo valor de setenta siete.

```
[[ 0 1]
 [77 3]
 [ 4 5]]
```

[0 1 77 3 4 5]

Pero el vector (c) si se modifica por un valor negativo en la posición fila cero y columna cero.

```
[[-99   1]
 [ 77   3]
 [ 4   5]]
```

Si se toma el array (d) y multiplicamos los elementos dentro de el con un dos es decir (d*2).

Podemos visualizar que el array cambia por completo ya que el vector (d) original cuyos elementos son [1,2,3,4,5] es modificado totalmente ya que cada elemento es elevado al cuadrado y por consiguiente el vector (d) queda de la siguiente manera:

[2 4 6 8 10]

Lo mismo pasa si elevamos al cuadrado los elementos [1,2,3,4,5] que se encuentran dentro del vector(d) dando como resultado el siguiente vector:

[1 4 9 16 25]

Para verificar si dentro de un vector hay algún valor erróneo se puede saber con diferentes funciones. En este ejemplo primero se creó un vector y en el mismo se insertó un NAN.

Además, se implementó otras funciones como: calcular el promedio de los valores que no son NAN, como se ve a continuación.

Primero se creó un array, donde contiene valores NAN.

[1. 2. nan 3. 4.]

Después de crear el array se hace uso de la función np.isnan (c), la cual recibe a c y lo va analizando en el momento cuando encuentra un valor NAN, pone un True y dado el caso contrario pone False.

[False False True False False]

Cuando se imprime la función np.isnan(c), se eliminan los valores NAN.

[1. 2. 3. 4.]

Para saber el promedio de los datos que no son NAN se utiliza la función np.mean.

2.5

Se mostrarán los resultados , luego de haber implementado las siguientes instrucciones.

```
data = np.genfromtxt("web_traffic.tsv", delimiter="\t")
print(data[:10], '\n')
```

```
[[1.000e+00 2.272e+03]
 [2.000e+00          nan]
 [3.000e+00 1.386e+03]
 [4.000e+00 1.365e+03]
 [5.000e+00 1.488e+03]
 [6.000e+00 1.337e+03]
 [7.000e+00 1.883e+03]
 [8.000e+00 2.283e+03]
 [9.000e+00 1.335e+03]
 [1.000e+01 1.025e+03]]
```

Para saber el tamaño de los datos anteriores se puede utilizar la instrucción print (data.shape) y el resultado es:

(743, 2) Hay 743 filas en 2 columnas.

Para dividir un array se utiliza la instrucción

```
x = data[: ,0]
y = data[: ,1]
```

```
[ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14.15. 16. 17.
18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28.29. 30. 31.
32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42.43. 44. 45.
46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56.57. 58. 59.
60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70.71. 72. 73.
74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84.85. 86. 87.
88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98.99. 100.
101. 102. 103. 104. 105. 106. 107. 108. 109.
110. 111. 112……
```

Cuando se utiliza las instrucciones:

Esta es para conocer la dimensión de los vectores

```
print(x.ndim, '\n')
print(y.ndim, '\n')
```

el resultado es: 1 y 1. Dado que son dos vectores.

Para conocer los tamaños de los vectores se hace con:

```
print(x.shape, '\n')
print(y.shape)
```

y da como resultado: (743,) y (743,). Dado a que hay tantos valores de horas como valores de compras.

```
print(np.sum(np.isnan(y)))
```

Hay que recordar que los que no son NAN son False y los que si son NAN son True. Entonces se suman los que si son NAN dando como resultado **8** valores NAN en el vector y.

Cuando se comprime los dos vectores da como resultado:

```
(743,)
(743,)
(735,)
(735,)
```

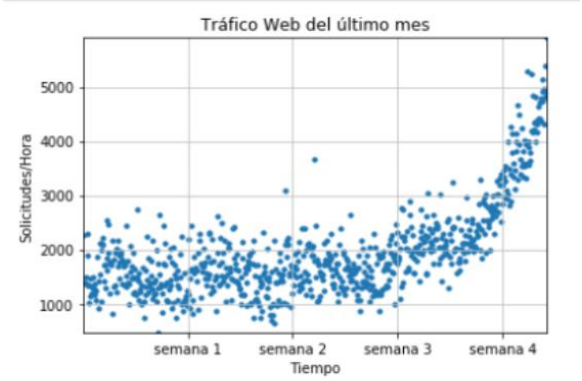

Figura 1 grafica de los vectores x,y.

1.3 CONCLUSIONES

- Después de haber visto en funcionamiento la librería Numpy, se puede concluir que esta es muy practica en el tratamiento de datos.

  Aún más cuando esos datos que llegan no siempre se pueden utilizar esto dado a que algunos son valores basura, no tienen ningún valor y es por eso, por lo que es mejor deshacerse de estos y que no se tengan en cuenta en los cálculos. Con numpy se pueden ver fácilmente esos datos NAN para luego quitarlos del vector.

- También algo importante es que se pueden graficar los datos que se han obtenido un ejemplo que se vio fue el tráfico web, donde se contrastaban las solicitudes que llegaban por hora contra el tiempo.

- En comparación con la propia lista de Python, usar Numpy es más fácil y conveniente, por lo que se usa ampliamente en el desarrollo de algoritmos de aprendizaje automático.

1.4 INFORMACION ACEDEMICA

- **Anggie Paola Villada Ortiz.** Nacida en 18/07/1998Guática Risaralda identificada con cedula de ciudadanía 1089721336. Me considero una persona responsable, dinámica y creativa, con facilidad de adaptación y capacidad de trabajar en equipo, en condiciones de alta presión para resolver problemas eficientemente y lograr las metas y objetivos trazados al nivel educativo. Estudios realizados Técnico en contabilización de operaciones comerciales y financieras. Colegio María Reina. Guática. (2015),Secretariado sistematizado. Instituto Panamericano. Guática. (2016),Curso de Inglés. Colombo Americano. Guática. (2014-2015),Curso de Salud Ocupacional. Sena. Guática. (2016),Curso de Excel. Sena. Guática. (2013).

- **Adriana Quitumbo**. Nací el 26 de Julio de 1993 en

Marsella Risaralda, después de cinco años mi familia y yo nos mudamos a Pereira. Inicie mis estudios a los cinco años en el colegio manos unidas, en ese colegio curse hasta noveno grado. Luego entre al colegio la julita donde me gradué como técnico bachiller en ventas de productos y servicios en el año 2012. En el año 2014 me gradué en el Sena como técnico en asistencia administrativa, en el 2015 realice prácticas laborales en una empresa de telecomunicaciones por dos años, finalmente me inscribe en un programa de la alcaldía para una beca de estudio la cual finalmente me gane esto fue en el año 2017, para estudiar ingeniería en sistemas y telecomunicaciones en la universidad tecnológica de Pereira aún estoy realizando mis estudios actualmente estoy en el séptimo semestre.