Computación Blanda

Soft Computing

Autor: Ana Manuela Gamboa Piedrahita, Orfilia Castillo Maturana
ISC, Universidad Tecnológica de Pereira, Pereira Colombia
Correo-e: Orfilia.castillo@utp.edu.co
Manuela.gamboa@utp.edu.co

Resumen— Este documento mostrará algunas funciones implementadas usando la biblioteca numpy, que significa "Python digital". La biblioteca se puede usar en Python, principalmente cuando se aprende aprendizaje automático, porque la biblioteca proporciona una estructura de datos poderosa en la que se implementan matrices y matrices multidimensionales, por lo que se puede acceder a ellas más rápido y los cálculos serán más eficientes. Ahora que conocemos el propósito de la biblioteca numpy, usaremos algunas de sus funciones. Primero, comenzamos a crear un vector, y luego usamos shape para comprender el tamaño de los datos del vector.
Si desea cambiar la estructura del vector, use reshape. De igual forma, es obvio que en Python, cuando un número es incorrecto, es decir, no hay valor, se llama NAN, este dato no debe ser considerado en el cálculo, por lo que para la biblioteca numpy existen varias funciones para saber si está dado. En el vector, hay datos basura y elimínelos.

Palabras clave— Numpy, Machine Learning, Array, Vector

Abstract— This document will show some functions implemented using the numpy library, which means "digital Python". The library can be used in Python, mainly when you learn machine learning, because the library provides a powerful data structure in which matrices and multidimensional arrays are implemented, so they can be accessed faster and the calculations will be more efficient. Now that we know the purpose of the numpy library, we will use some of its functions. First, we start creating a vector, and then we use shape to understand the size of the vector data.
If you want to change the structure of the vector, use reshape. Similarly, it is obvious that in Python, when a number is incorrect, is no value, is called NAN, this data should not be considered in the calculation, so for the numpy library there are several functions to know if it is given. In the vector, there is garbage data and delete it.

Key words— Numpy, Machine Learning, Array, Vector

I. INTRODUCCION

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.2 Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Python se puede trabajar desde la interfaz de jupyter que viene incorporado en el paquete de anaconda, especialmente en la parte de jupyter notebook. En esta interfaz se pueden trabajar diversas formas de programacion ya sea orientada a objeto, paradigmica entre otras, especialmente se facilita mucho el trabajo o el manejo de los arreglos (vectores) o matrices, python tiene bibliotecas exclusivamente para el manejo de estas como lo es numpy, una de las bibliotecas que te permite analizar datos en grandes cantidades que estan digitados en este tipo de arreglos.

En este document trataremos de hacer un repaso de lo que es esta librerira y de hacer un resumen de algunas funciones utilizadas en un documento elavorado en jupyter notebook.

Cabe enfatisar que el analisis que se le hace a este document es para reforsar aspectos anteriormente mencionados.

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.

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.

II. METODOLOGIA

Lo primero que se debe hacer es importar la biblioteca numpy con alias np (import numpy as np), y luego continuamos creando un vector (a) con seis elementos. Este vector se crea llamando a la biblioteca numpy, es decir, usando el operador de punto np y usando los elementos 0, 1, 2, 3, 4 y 5 para crear una matriz, por lo que la función es la siguiente: 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 número 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 desea cambiar la estructura de la matriz, puede usar el operador de remodelación (el término está definido por la biblioteca numpy); esto me permite tener tres filas y dos columnas. En este proyecto, puede ver que el vector (b) está vinculado al vector (a), por lo que el vector (b) se imprime para observar los cambios de estructura de la matriz creada inicialmente.

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

El proceso de modificación de los elementos de la matriz se realiza citando el nombre, la fila y la columna de la matriz en (b) y asignando un número directo (en este caso 77). El cambio de elementos en el vector 8B9 se puede demostrar imprimiendo (b).

Se debe considerar que el arreglo (b) está vinculado al arreglo (a), es decir, el vector (a) es el mismo que (b), por lo que si se imprime notaremos que (a) ha cambiado.
Para evitar que esto suceda, usamos el operador de copia en la estructura de matriz b = a.reshape ((3,2)) cabe Enfatizar tres filas de una matriz o vector y dos columnas de una matriz o vector; por lo tanto, la función es la siguiente:

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

Recuerde que el vector se crea como una copia y el vector (c) se imprime para verificar qué elementos contiene el vector.
Cambie el primer elemento en la posición cero de la columna y la fila del vector (c) a menos noventa y nueve, y luego imprima la matriz.
Cabe destacar que estas operaciones se extenderán por toda la matriz, como se muestra en la siguiente línea:

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

Si multiplicamos la matriz (d) por 2, es decir (d * 2); dado que cada elemento se multiplica por 2, los elementos de la matriz se modifican, por lo que la matriz ha cambiado, y si aumentamos los elementos de la matriz, Lo mismo sucede multiplicado por 2 (d ** 2).
En el aprendizaje automático, los datos que usamos llegan a través de un repositorio desconocido, y la información que llega viene con mucho ruido y basura (las señales dañinas se mezclan con señales útiles para ser transmitidas). Y todas estas interferencias deben eliminarse.

En Python, si el número es incorrecto (número sin asignar, sin valor), se llama NAN. Es por esto que la constante np.NAN existe en Python, es una constante que indica la existencia de un valor basura, que no debe ser parte del cálculo. Para verificar si hay datos basura en el vector, primero cree el vector y coloque un valor incorrecto de la siguiente manera:

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')

Este comando se utiliza para mostrar el valor promedio de los valores que no son NAN. Primero imprima el vector y luego use la función np.isnan para mostrar los vectores que no son NAN, y luego agregue la negación ~, de modo que el valor de False se convierta en Verdadero, porque Python usa el valor verdadero.

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

Esta declaración se usa para filtrar valores que no son NAN y promediar los datos filtrados usando la función np.mean.
Después de analizar estas características, ahora se implementarán en el siguiente ejemplo: Una empresa vende servicios para proporcionar algoritmos de aprendizaje automático a través de HTTP. Como éxito, ha aumentado la

demanda de una mejor infraestructura para satisfacer las solicitudes web entrantes.

Necesita saber cuál es el límite de infraestructura actual, se estima que hay 100.000 solicitudes por hora por hora. Para continuar procesando todas las solicitudes, necesita saber aproximadamente cuándo comprar otros servidores de la nube.
Para responder a estas preguntas planteadas anteriormente en este ejemplo, se ejecutará un programa básico de aprendizaje automático en Python. Primero, obtenga los datos y proceselos. La primera columna es el número de horas que llega cada solicitud y la segunda columna es el número de tareas 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')
```

El rango de datos que se muestra en esta sección es 10. Para averiguar el tamaño de estos datos, puede usar el comando print (data.shape), donde los datos son los datos en el archivo y la forma es el tamaño.
Ahora, dividiremos la matriz (matriz) para crear dos vectores. La primera columna es x (horas) y la segunda columna es y (filas). Para dividir la matriz, use el siguiente comando:

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

Para mostrar los valores en x, y

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

para saber las 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()
```

III. RESULTADOS

Después de crear un vector (a), donde los elementos son: [0 1 2 3 4 5], podemos mostrar la dimensión de la matriz (1) y el número de elementos en el vector (6,). Cuando cambiamos la estructura del vector a tres filas y dos columnas, el resultado del vector es:

```
[[0 1]
 [2 3]
```

[4 5]]

La matriz se muestra de esta forma debido a que en la intruccion anterior se le indica que debe quedar organizada de 3 filas con 2 columnas donde en cada fila van a ir dos elementos.

Al modificar los elementos en la matriz, podemos ver que en la posición cero de la columna y la fila, el segundo número en el vector se ha cambiado por 77.

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

El resultado del vector (a) al imprimir es:

[0 1 77 3 4 5]

Podemos probar que el resultado del vector (a) se ha modificado en el ítem anterior. Para corregir este error, se implementa una copia usando la función c = a.reshape ((3,2)), por lo que Copy () proporciona un vector (a), en el que continúa con el mismo valor 77 de la Conversión, por lo que no ha cambiado.

[[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 tomamos la matriz (d) y multiplicamos sus elementos por 2, eso es (d * 2).
Podemos ver que dado que el vector original (d) con elementos [1,2,3,4,5] está completamente modificado, debido a que cada elemento está al cuadrado, la matriz se cambia por completo, por lo que el vector (d) sigue siendo Existen 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 hay un valor de error en el vector, puede encontrar diferentes funciones. En este ejemplo, primero se crea un vector y se inserta NAN en él.

Además, también se implementan otras funciones, como el cálculo del valor promedio de valores que no son NAN, como se muestra a continuación. Primero cree una matriz que contenga 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]

Al imprimir la función np.isnan (c), se eliminará el valor NAN.
[1. 2. 3. 4.]

La función np.mean se usa para encontrar el valor promedio de datos que no son NAN.

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 poder presenciar cual es el tamaño de los datos mostrados anteriormente se utiliza la función print (data.shape) que nos permite imprimir en pantalla el resultado el cual es:

(743, 2)
Indicando que en la tabla anterior hay 743 filas en 2 columnas.

Si en un momento dado se quiere dividir un arreglo ya sea una matriz o un vector se realiza con 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:

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

se utilizan para conocer las dimensiones de cada uno de los vectores que se le indiquen el resultado en este caso es de: 1 y 1. Dado que son dos vectores y para conocer el tamaño de los vectores se realiza con las funciones:

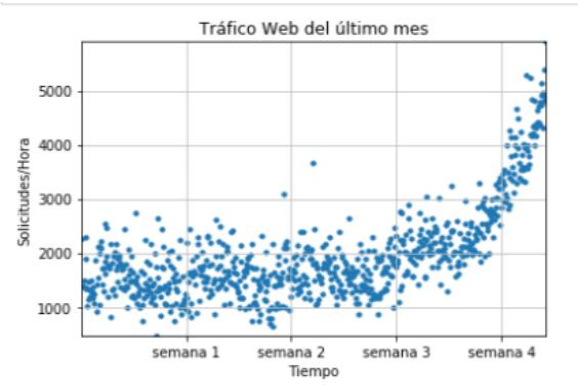
```
print(x.shape, '\n')
print(y.shape)
```

como resultado esto nos muestra que el tamaño de los vectores es de: (743,) y (743,). Estos resultados son iguales debido a que el número de horas en los vectores es igual al número 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 el vestor podemos apreciar el siguiente resultado:
Se da por que me muestra tanto los que están el el eje X como en el Y.

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



IV. CONCLUSIONES}

1. El manejo de los datos en arreglos como matrices o como vectos es mas simple y de una manera mas eficiente y agil cuando se comoprende ls librerias que se requieren para su elavoracion.

2. Luego de ver el funcionamiento de la biblioteca Numpy, se puede concluir que es muy útil en el procesamiento de datos. Más importante aún, cuando los datos que llegan no siempre se pueden usar, porque algunos datos son valores basura, no tienen ningún valor, que es la razón. Por lo tanto, es mejor no usarlos y no considerarlos en los cálculos. Con numpy, puede ver fácilmente los datos NAN y luego eliminarlos del vector.

3. También es muy importante que sea posible graficar los datos que se han obtenido, como el tráfico de la red, que compara las solicitudes que llegan cada hora con el tiempo.

V. INFORMACION ACADEMICA

- **Ana Manuela Gamboa piedrahita**

autora nacida el 26 de julio de 1998 en Pereira, Risaralda,madre Consuelo Piedrahita y padre Milton Gamboa.

estudio de 2003-2015 en la institucion educativa el Dorado graduada el 06 de diciembre de 2015 con el mejor icfes de la jornada de la mañana. Estudio de 2014-2015 un tecnico en desarrollo de software en el sena durante sus ultimos años de escolares.

Estudiante de ingenieria de sistemas y computacion de la universidad tecnologica de pereira (UTP), iniciando el año 2017, ganadora de una beca por parte del programa becas pa' pepas,creado en el 2017, por la alcaldia de pereira.

- **Orfilia castillo Maturana,** nació el 24 julio en Tado Colombia 1998, graduada del bachillerato en el 2015 de la institución educativa el dorado con mención de honor por ser una de las estudiantes más disciplinadas, en el 2015 obtuvo también el título de tecnóloga en programación del software de la institución de educación SENA de la ciudad de Pereira.

Becada por la Alcaldía de Pereira, y actualmente estudiante de ingeniería en sistemas y computación en la universidad tecnológica de Pereira y cursando el sexto semestre, estudiante también de contabilidad y finanzas de manera virtual en la institución a Distancia SENA, y trabaja en la floristería ml detalles de la ciudad de Pereira ubicada en el sector del poblado etapa 1.