

Vectores en Python

Daniel Eduardo Macias Estrada

30/1/2021

Vectores

Para crear un vector en Python, utilizamos **numpy.array()**

```
import numpy as np
x = np.array([1,2,3,4,5,6,7,8,9,0])
x
```

```
## array([1, 2, 3, 4, 5, 6, 7, 8, 9, 0])
```

Longitud

Para calcular la longitud (la dimensión), utilizamos la función **len()**

```
len(x)
```

```
## 10
```

Suma y resta de vectores

Sumar y restar dos o más vectores es muy simple, siempre y cuando sean de la misma dimensión

```
x = np.array([0,3,-1,3,5])
y = np.array([1,2,3,-1,0])
x+y
```

```
## array([1, 5, 2, 2, 5])
```

```
x-y
```

```
## array([-1,  1, -4,  4,  5])
```

Producto por un escalar

El producto por un escalar también funciona de forma sencilla

```
x = np.array([1,0,-1,0,2,0,-2])
2*x
```

```
## array([ 2,  0, -2,  0,  4,  0, -4])
```

```
-2*x
```

```
## array([-2,  0,  2,  0, -4,  0,  4])
```

```
5*x
```

```
## array([ 5,  0, -5,  0, 10,  0, -10])
```

Producto escalar

```
def productoEscalar(x,y):
    if(len(x) == len(y)):
        suma = 0
        for i in range(0,len(x)):
            suma = suma + x[i]*y[i]
        return suma
    else:
        return "Error: La dimensión de ambos vectores no son iguales"

x = np.array([0,3,-1,3,5])
y = np.array([1,2,3,-1,0])
productoEscalar(x,y)
```

```
## 0
```

Norma euclídea

Para calcular la norma de un vector, lo que hacemos es utilizar la función **numpy.linalg.norm()**

```
x = np.array([1,2,0,3,-1,1])
np.linalg.norm(x)
```

```
## 4.0
```

```
import math
math.sqrt(productoEscalar(x,x))
```

```
## 4.0
```

Distancia euclídea

Para calcular la distancia entre dos puntos, la definimos como la norma del vector que los une, es decir $d(x, y) = \|x - y\|$. En nuestro caso vamos a utilizar la función **distancia()**, la cual definimos del siguiente modo:

```
def distancia(x,y):
    if(len(x) == len(y)):
        return np.linalg.norm(x-y)
    else:
        return "Error: ambos puntos de pertenecen al mismo plano"

x = np.array([0,3,-1,3,5])
y = np.array([1,2,3,-1,0])
distancia(x,y)
```

```
## 7.681145747868608
```

Ángulo entre dos vectores

```
def angulo(x,y):
    if(len(x) == len(y)):
        angulo = math.acos(productoEscalar(x,y) / np.linalg.norm(x)*np.linalg.norm(y))
        return angulo
    else:
        return "Error: los vectores no son de la misma dimensión"

x = np.array([0,3,-1,3,5])
y = np.array([1,2,3,-1,0])
angulo(x,y) * 180 / math.pi
```

```
## 90.0
```

Proyección ortogonal

```
# Proyección y sobre x
def proyeccion(x,y):
    if(len(x) == len(y)):
        lam = productoEscalar(x,y) / productoEscalar(x,x)
        return lam * x
    else:
        return "Error: Los vectores no son de la misma dimensión"

x = np.array([3,1])
y = np.array([1,2])
proyeccion(x,y)
```

```
## array([1.5, 0.5])
```

Producto vectorial

```
def productoVectorial(x,y):
    if(len(x) == len(y) and len(x)==3):
        return np.array([x[1]*y[2] - x[2]*y[1], x[2]*y[0] - x[0]*y[2], x[0]*y[1] - x[1]*y[0]])
    else:
        return "Error: No se cumplen las condiciones para que se obtenga el producto vectorial"

x = np.array([1,2,3])
y = np.array([0,-1,1])
productoVectorial(x,y)
```

```
## array([ 5, -1, -1])
```

Producto mixto

```
def productoMixto(x,y,z):
    if(len(x) == len(y) and len(x) == len(z) and len(x) == 3):
        return int(np.linalg.det(np.array([x,y,z])))
    else:
        return "Error"

x = np.array([1,2,3])
y = np.array([0,-1,1])
z = np.array([2,0,-3])
productoMixto(x,y,z)
```

```
## 13
```