

# Redes Neuronales y Aprendizaje Profundo para Visión Artificial

Dr. Ariel H. Curiale

Cuatrimestre: 2do de 2019

---

## PRÁCTICA 0: INTRODUCCIÓN A PYTHON/NUMPY/SCIPY/MATPLOTLIB

---

1. Resuelva el siguiente sistema de ecuaciones de la forma más simple que pueda

$$\begin{aligned}x + z &= -2 \\2x - y + z &= 1 \\-3x + 2y - 2z &= -1\end{aligned}$$

2. Calcular el histograma, la media ( $\mu$ ) y la desviación estándar ( $\sigma$ ) de:

```
data= np.random.gamma(3,2,1000)
```

Verificar que sean las correctas.

3. Definir una función para encontrar las raíces de la ecuación de segundo grado

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

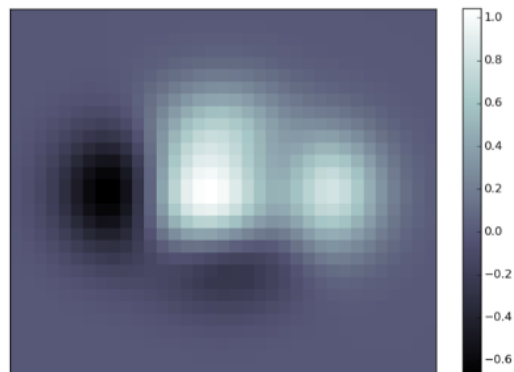
El parámetro  $a$  es obligatorio,  $b$  y  $c$  son opcionales con un valor por defecto de 0. La función debe devolver un array con las raíces.

4. Crear una función que dados los coeficientes de un polinomio de segundo grado grafique la parábola ( $x \in [-4.5, 4]$ ). Además, marcar las raíces en el gráfico, por ejemplo con puntos rojos y un texto indicando su valor con 3 decimales utilizando latex ( $x_i = 1.22$ ).
5. Defina una clase `Lineal` que recibe parámetros  $a$ ,  $b$  y un método que resuelve la ecuación  $ax + b$ . Definir el método para pueda ser invocado como una función.
6. Herede de la clase `Lineal` una subclase `Exponencial` que resuelva la ecuación  $ax^b$ .
7. Crear un módulo `circunferencia.py` donde se define la constante `PI` y la función `area(r)` que calcula el área para una circunferencia de radio  $r$ . Importar el módulo con el alias `circle`. Verificar el valor de `PI` y calcular el área. Luego importar el modulo utilizando la sintaxis `from .... import ...` y verificar nuevamente la constante y calcular el área. ¿Son los mismos objeto?
8. Crear un paquete llamado `geometria` el cual contenga al módulo `circunferencia` creado anteriormente y otro análogo que se llame `rectangulo`, también con la función `area` que permite calcular su área.

9. Generar la estructura de paquetes, módulos y archivos que necesite para que las siguientes instrucciones funcione correctamente:

- `import p0_lib`
- `from p0_lib import rectangulo`
- `from p0_lib.circunferencia import PI, area`
- `from p0_lib.ellipse import , area`
- `from p0_lib.rectangulo import area as area_rect`

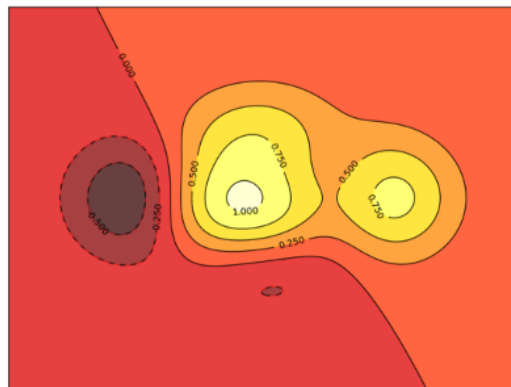
10. A partir de la definición de la función y el código de abajo intente obtener un gráfico como el siguiente teniendo en cuenta el colormap (bone), la interpolación de la imagen y el origen:



```
def f(x, y):
    return (1 - x / 2 + x ** 5 + y ** 3) * np.exp(-x ** 2 - y ** 2)

n = 10
x = np.linspace(-3, 3, 4 * n)
y = np.linspace(-3, 3, 3 * n)
X, Y = np.meshgrid(x, y)
plt.imshow(f(X, Y))
```

11. A partir de la definición de la función y el código de abajo obtener un gráfico como el siguiente teniendo en cuenta el colormap (hot):

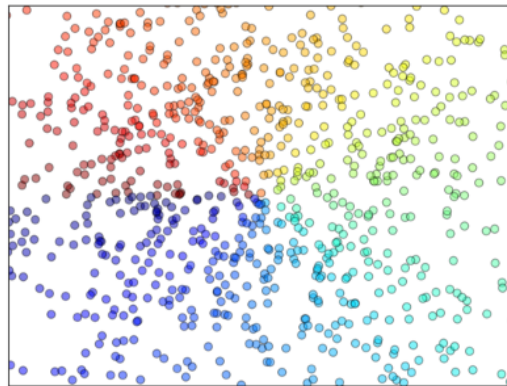


```
def f(x, y):
    return (1 - x / 2 + x ** 5 + y ** 3) * np.exp(-x ** 2 - y ** 2)

n = 256
x = np.linspace(-3, 3, n)
y = np.linspace(-3, 3, n)
X, Y = np.meshgrid(x, y)

plt.contourf(X, Y, f(X, Y), 8, alpha=.75, cmap='jet')
C = plt.contour(X, Y, f(X, Y), 8, colors='black', linewidth=.5)
```

12. A partir del código de abajo intente obtener un gráfico como el siguiente teniendo en cuenta el tamaño de los puntos, color y transparencia:



```
n = 1024
X = np.random.normal(0, 1, n)
Y = np.random.normal(0, 1, n)

plt.scatter(X, Y)
```