

# Métodos de optimización NO lineal

Grupo 19

Catolino Lucas  
Ricarte Matías





# Métodos de optimización no lineal

- Gradiente descendente
- Gradientes conjugados
- ADAM



# Librerías utilizadas

- Numpy
- Scipy: gradientes conjugados (Powell)
- TensorFlow: gradiente descendente, ADAM



# Consideraciones

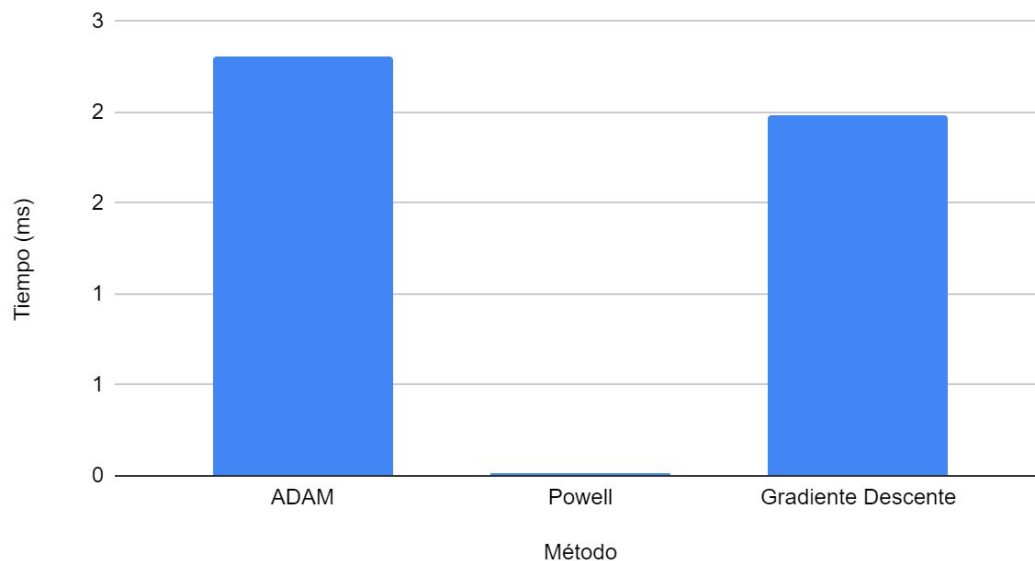
- Condición de corte: error menor a  $10^{-7}$
- Cota máxima de iteraciones: 500
- Tasa de aprendizaje: 0.1



# Tiempos de ejecución

Método	Tiempo
ADAM	(2.0±0.6) ms
Powell	(0.017±0.001) ms
Gradiente Descente	(1.98±0.88) ms

Tiempo por método

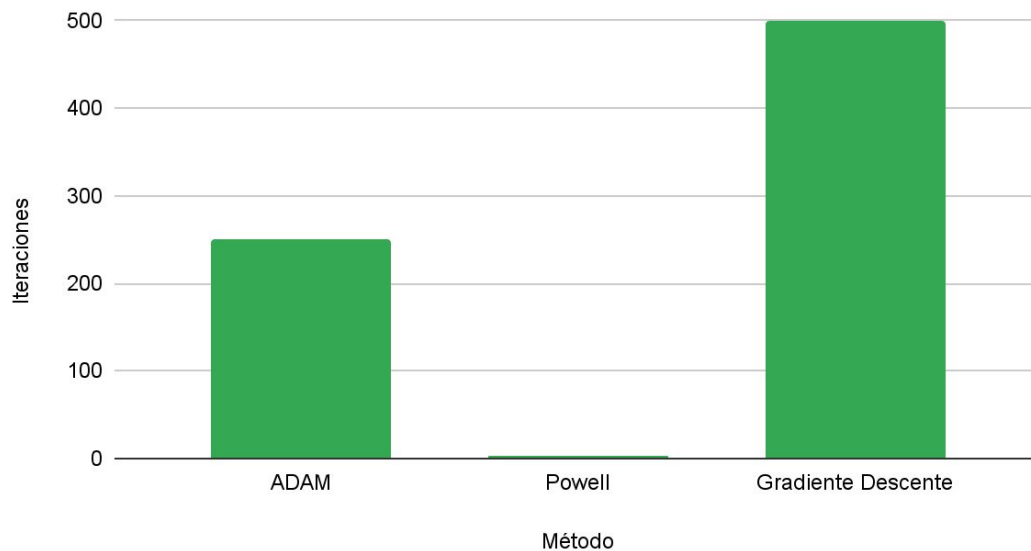




# Iteraciones alcanzadas

Método	Promedio
ADAM	250±9
Powell	3
Gradiente Descente	500

Iteraciones por método

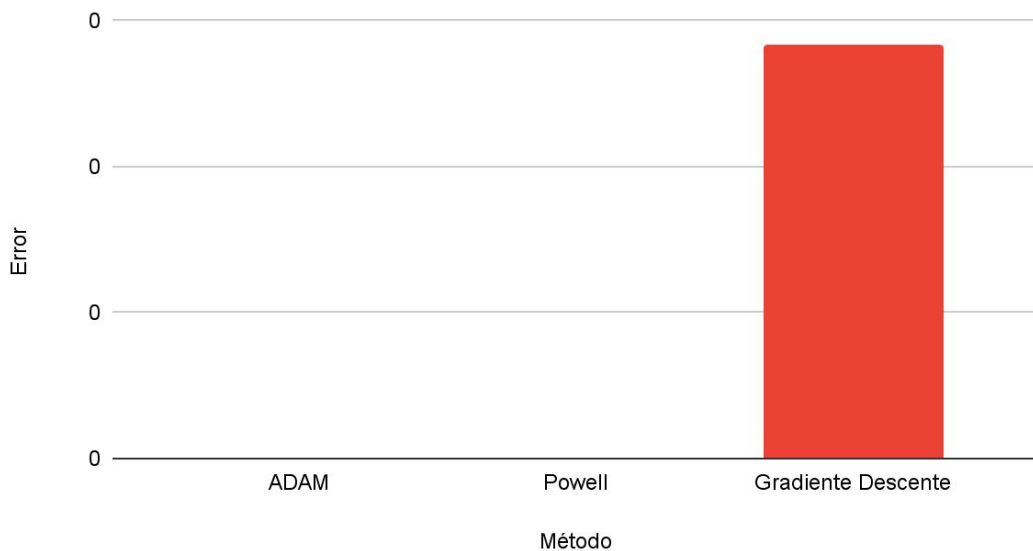




# Error en el óptimo

Método	Error
ADAM	0
Powell	0
Gradiente Descente	0.014

Error por método





# Error del óptimo

## **ADAM**

OPTIMAL ERROR: 9.978922777830449e-08

## **POWELL**

OPTIMAL ERROR: 0.0

## **GRADIENT DESCENT**

OPTIMAL ERROR: 0.001253697886841064





## Argumento del óptimo

ADAM

$$W = \begin{pmatrix} 4.17722391 \\ 9.12699865 \\ 7.6919811 \end{pmatrix}$$

$$w = \begin{pmatrix} -1.56041346 & 0.75923858 & 1.91316528 \\ -1.11005829 & 0.14234894 & 1.95698007 \end{pmatrix}$$

$$w_0 = \begin{pmatrix} 0.37478059 \\ 0.48189252 \end{pmatrix}$$



# Argumento del óptimo

Gradientes conjugados

$$W = \begin{pmatrix} -1.22887684e + 01 \\ 4.65003875e + 02 \\ -8.54624476e + 03 \end{pmatrix}$$

$$w = \begin{pmatrix} 2.62264267e + 00 & 3.20294706e + 01 & 3.14927336e + 01 \\ 4.45033315e + 01 & 3.15160571e + 01 & 3.08187902e + 01 \end{pmatrix}$$

$$w_0 = \begin{pmatrix} 3.13302627e + 01 \\ 3.23177113e + 01 \end{pmatrix}$$



# Argumento del óptimo

Gradiente descendente

$$W = \begin{pmatrix} -1.80828436 \\ -3.05449161 \\ -4.30180749 \end{pmatrix}$$

$$w = \begin{pmatrix} -0.02747256 & 0.60430893 & -1.57987501 \\ 1.63903186 & 0.94041398 & -0.17717408 \end{pmatrix}$$

$$w_0 = \begin{pmatrix} 0.53767958 \\ -0.56461543 \end{pmatrix}$$



# Problemas encontrados

```
GRADIENT_DESCENT (MINIMUM = 0.0000001, MAX_ITER = 50000)
OPTIMAL ARG: <tf.Variable 'Variable:0' shape=(11,) dtype=
array([-1.80828436, -3.05449161, -4.30180749, -0.02747256
        -1.57987501,  1.63903186,  0.94041398, -0.17717408
        -0.56461543])>
OPTIMAL ERROR: tf.Tensor(0.001253697886841064, shape=(),
TIME:  10.478729486465454
ITER:  5000
```

```
POWELL (MINIMUM = 0.0000001, MAX_ITER = 50000)
d:\gits\TP_SIA\Non-Linear-Optimization\optimize.py:24: RuntimeWarning: overflow
encountered in exp
    return np.exp(x) / (1 + np.exp(x))
d:\gits\TP_SIA\Non-Linear-Optimization\optimize.py:24: RuntimeWarning: invalid
value encountered in double_scalars
    return np.exp(x) / (1 + np.exp(x))
OPTIMAL ARG: [ 31.03354077  54.91781667  20.51772731 -895.84643991  91.13231563
 92.0398687 -895.81742202  92.13083551  91.87886842  91.58447211
 92.57882145]
OPTIMAL ERROR: nan
TIME:  0.05100107192993164
ITER:  2
```

# Muchas gracias

Grupo 19

Catolino Lucas  
Ricarte Matías

