

## Memoria práctica 1 de Aprendizaje Automático



# UNIVERSIDAD DE GRANADA

Antonio Manuel Fresneda Rodríguez

# Índice

Ejercicio sobre la búsqueda iterativa de óptimos.....	3
Apartado 1.....	3
Apartado 2.....	3
Apartado 3.....	4
Apartado 4.....	5
Ejercicio sobre regresión lineal.....	6
Apartado 1.....	6
Apartado2.....	9
Apartado 3.....	10
Apartado 4.....	10

# Ejercicio sobre la búsqueda iterativa de óptimos

## Apartado 1

El algoritmo del gradiente descendiente está implementado en el script de python codigo.py en la función GD.

## Apartado 2

A)

El gradiente se define como:

$$\nabla f(x, y) = \left( \frac{\partial E}{\partial x}, \frac{\partial E}{\partial y} \right)$$

Dada la función:

$$E(u, v) = (e^{v-2} u^3 - 4 v^3 e^{-u})^2$$

Sus derivadas parciales son:

$$\frac{\partial E}{\partial u} = 2(e^{v-2} u^3 - 4 v^3 e^{-u})(4 v^3 e^{-u} + 3 e^{v-2} u^2)$$

$$\frac{\partial E}{\partial v} = 2(u^3 e^{v-2} - 12 e^{-u} v^2)(u^3 e^{v-2} - 4 e^{-u} v^3)$$

Finalmente, el gradiente de la función  $E(u, v)$  es:

$$\nabla E(u, v) = [2(e^{v-2} u^3 - 4 v^3 e^{-u})(4 v^3 e^{-u} + 3 e^{v-2} u^2), 2(u^3 e^{v-2} - 12 e^{-u} v^2)(u^3 e^{v-2} - 4 e^{-u} v^3)]$$

B)

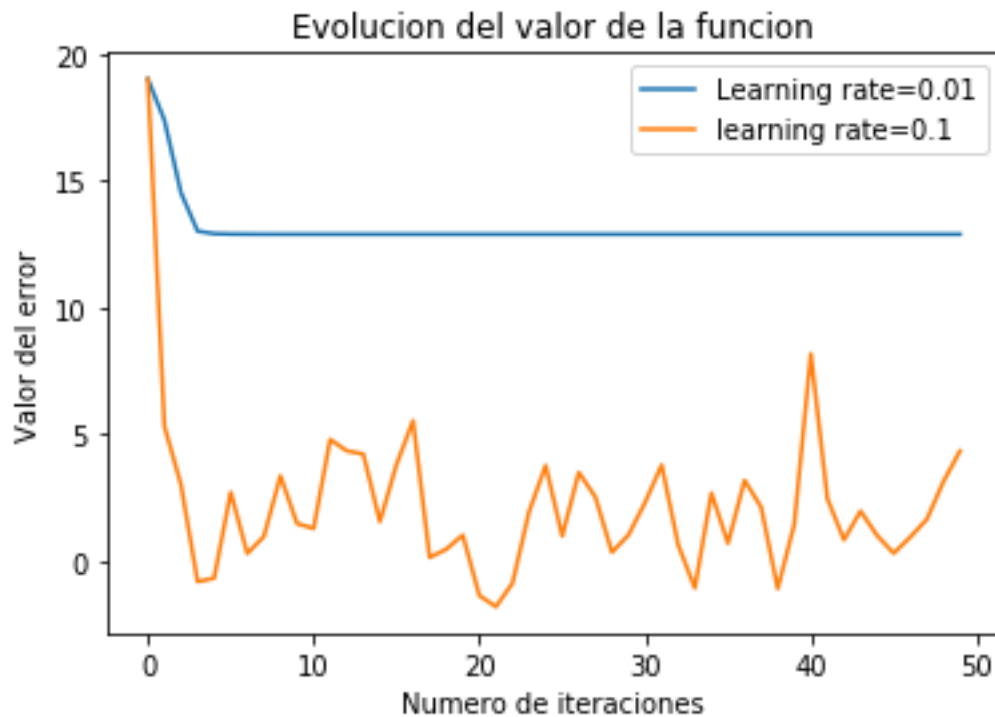
Con una tasa de aprendizaje de 0.1, el algoritmo no encontró un mínimo, debido al tiempo que tardaba en terminar, menor a  $10^{-14}$ , el mínimo que encontró tras 99 millones de iteraciones fue alrededor de  $10^{-9}$ .

Con una tasa de aprendizaje de 0.05, el algoritmo encontró un mínimo menor a  $10^{-14}$  tras 37 iteraciones.

C)

Ese mínimo se encuentra en las coordenadas  $(u,v)=(1.1195439,0.65398806)$ , cuya imagen es  $E(1.1195439,0.65398806)=2.09620956e^{-14}$ .

### Apartado 3



Usando el learning rate de 0.01 es tan pequeño que el algoritmo no es capaz de sortear un punto en la función, por eso vemos que a partir de un punto no el valor no avanza . En cambio si cambiamos a otro learning rate “sorteamos” ese punto, pero caemos en una serie de puntos que nos hacen oscilar de forma continuada.

Coordenada de inicio	Numero de iteraciones	Coordenada del mínimo	Valor del mínimo
(2.1,-2.1)	49	(2.2438049,-2.2379258)	-1.82007854
(3,-3)	49	(2.7309356,-2.7132791)	-0.3812495
(1.5,1.5)	49	(1.7779244-1.0320568)	18.04207858
(1,-1)	49	(1.7715243-1.0379605)	-0.3812495

## Apartado 4

La dificultad del uso de estos métodos es:

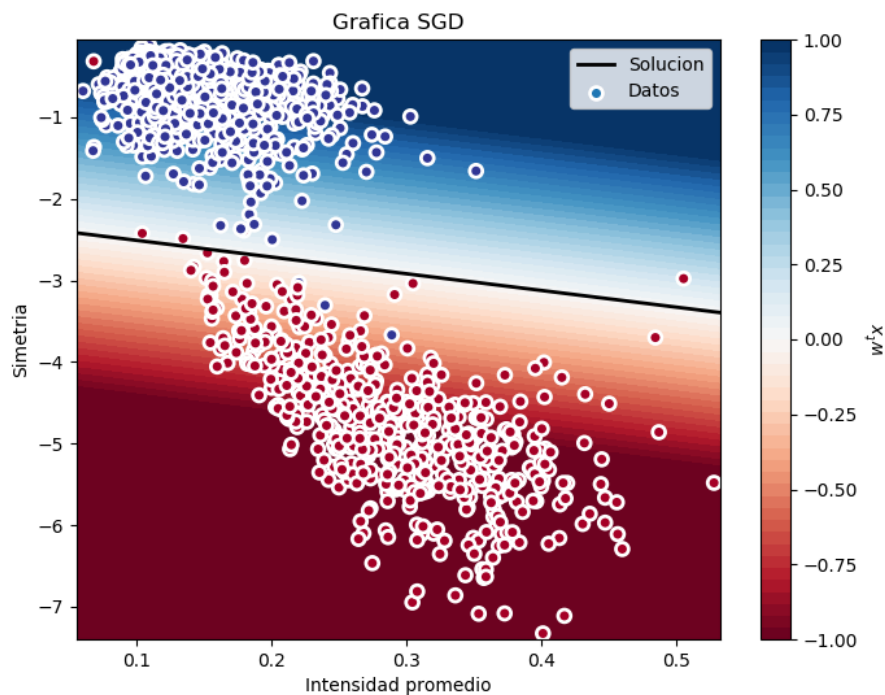
1. Establecer la tasa de aprendizaje: Como hemos visto en el primer ejercicio, la diferencia entre usar 0.1 y usar 0.5 son millones de iteraciones en el algoritmo.
2. La forma de la función: En el caso de que la función sea convexa, el mínimo se encuentra fácilmente puesto que el algoritmo puede converger mas rápidamente (esto no es seguro, puesto que cabe la posibilidad de que si la función tiene un mínimo local, puede quedarse estancado en el mismo). Pero en el caso de que no sea así, el algoritmo puede quedarse en algún mínimo local y no avanzar hacia el mínimo global.
3. También influye el numero de iteraciones del algoritmo.

# Ejercicio sobre regresión lineal

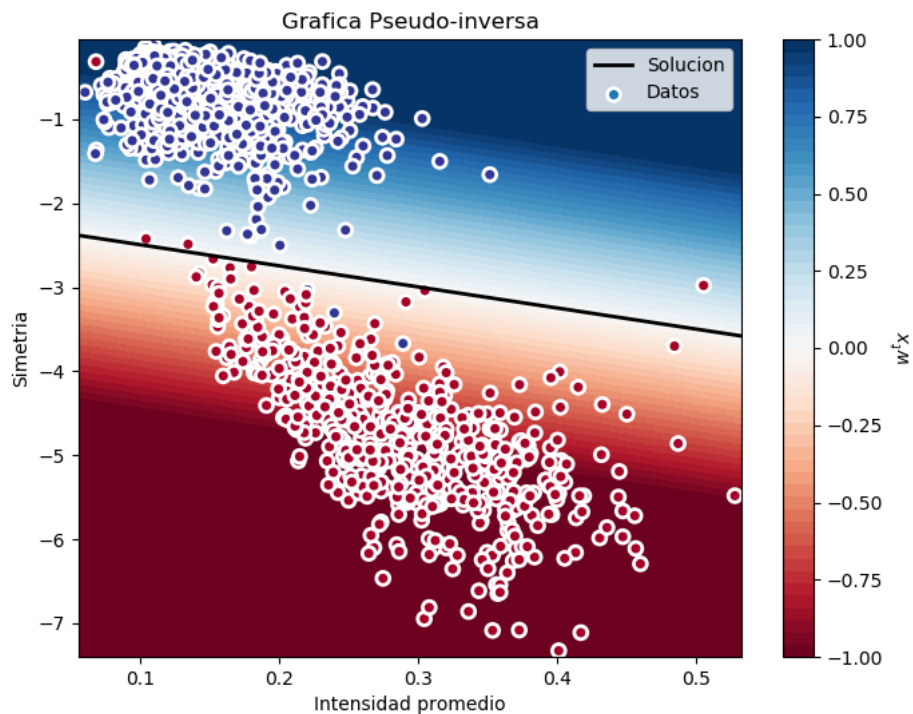
## Apartado 1

Este apartado está implementado en el script de python `codigo.py`. El gradiente descendiente estocástico está implementado en la función `SGD`. El método de la pseudo-inversa está implementado en la función `PIA`.

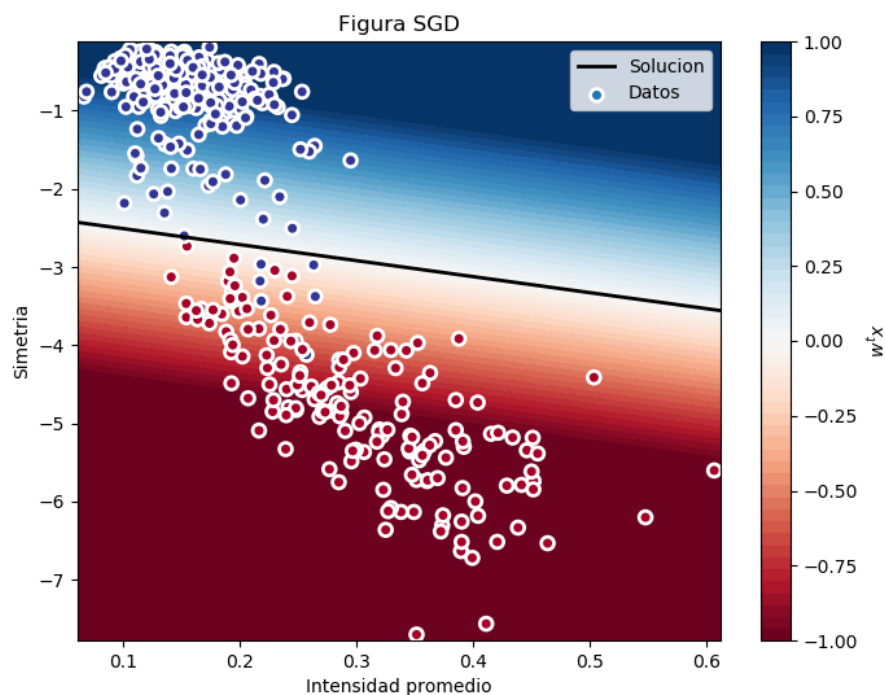
Aquí podemos ver la gráfica de como se reparte el conjunto de datos del train y la recta que vemos es la recta estimada con el SGD.



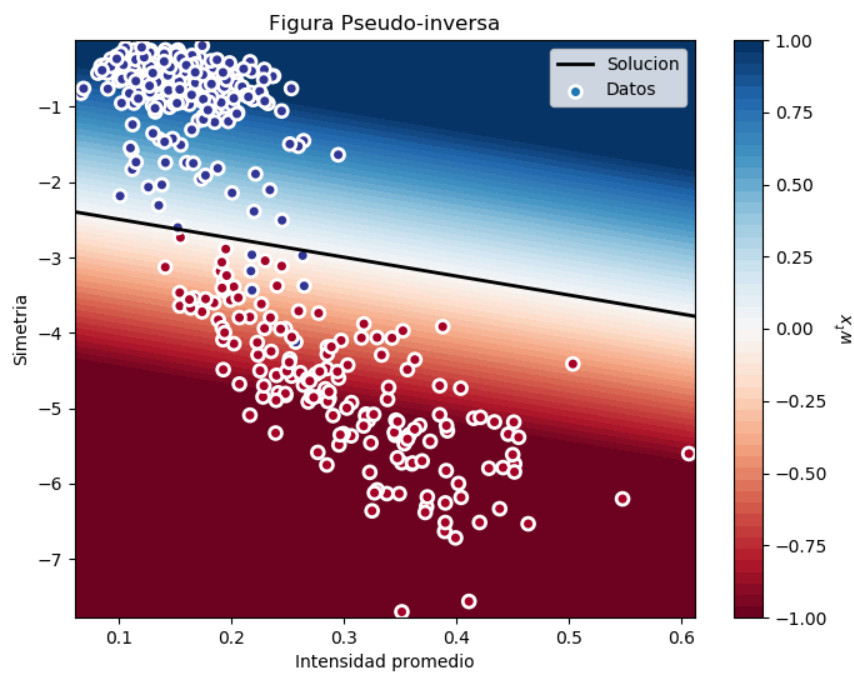
La siguiente gráfica representa como se reparte el conjunto de datos del train y la recta que vemos es la estimada con el método de la pseudo-inversa.



La siguiente gráfica vemos como se reparte el conjunto de datos del test y la recta estimada con el método del SGD.



La siguiente gráfica vemos como se reparte el conjunto de datos del test y la recta estimada con el método de la pseudo-inversa.



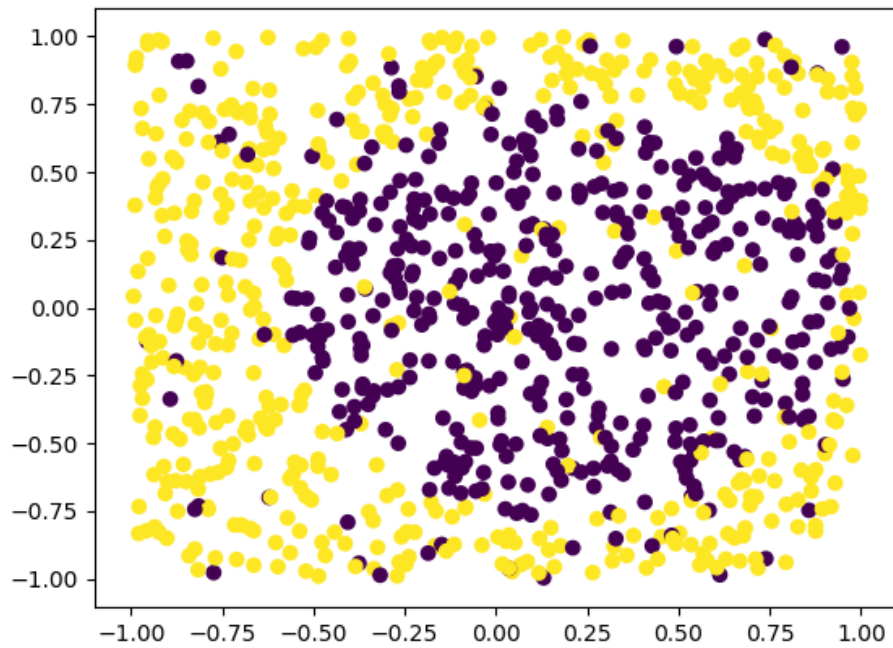
El error medio obtenido en cada uno de estos a sido:

1. 0,51 % para el train (tanto en el método SGD como con el de la pseudo-inversa).
2. 1,6 % para el test (tanto en el método SGD como en el de la pseudo-inversa).



## Apartado2

La distribucion de los puntos es la siguiente:

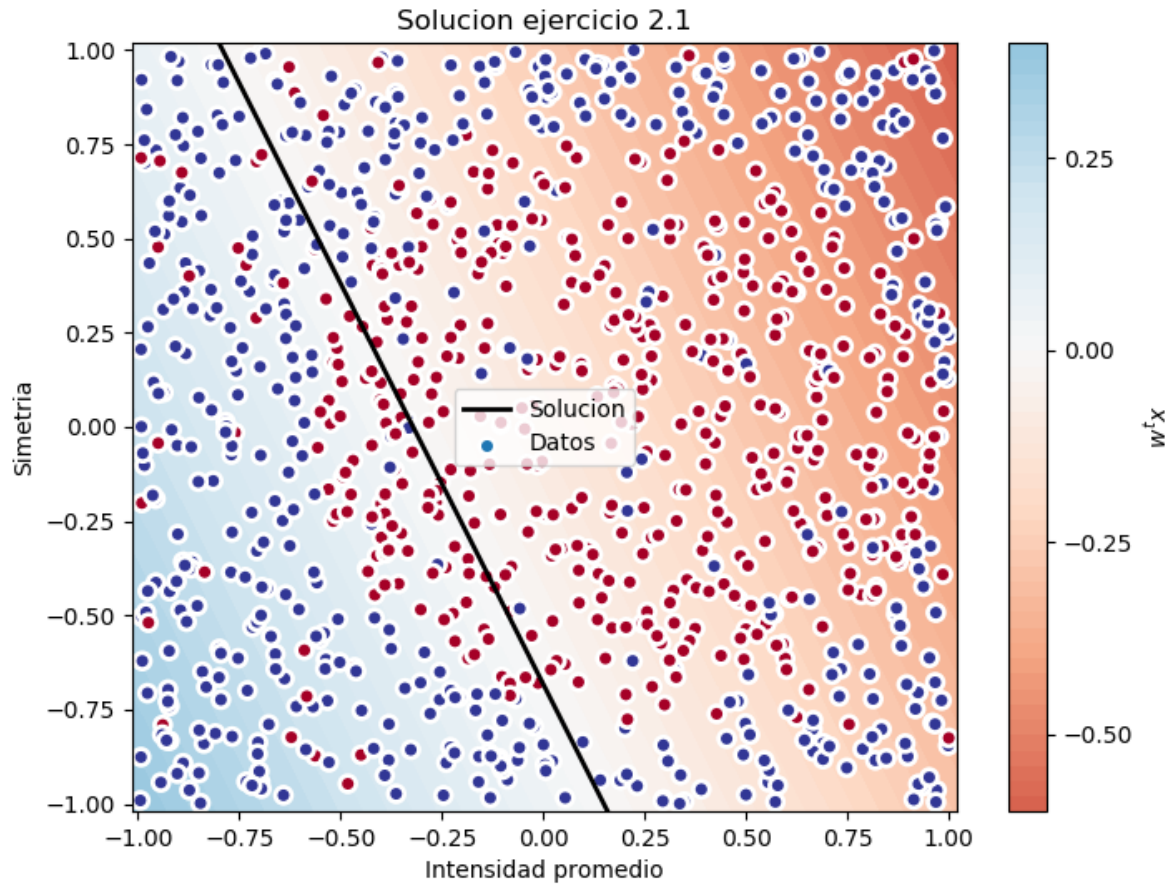


Usando esta distribución, y usando el SGD, la tasa de error ha sido: 38%.

## Apartado 3

El error medio en el train de 1000 muestra generadas con esta distribución ha sido: 41.3%.  
El error medio de la muestra test (1 muestra por cada muestra de train) ha sido: 41.5%.

Esta gráfica ilustrará los resultados.



## Apartado 4

Como hemos visto en el apartado 2 de este ejercicio, la distribución que generamos no es linealmente divisible (se ve claramente que la muestra se divide en dos circunferencias). Esto obviamente provoca que el error se dispare.