Tarea 3. Optimización

Oscar Esaú Peralta Rosales
Maestría en Computación
Centro de Investigación en Matemáticas

***Resumen*—Se presenta una solución al problema**

$$x^* = \underset{x \in \mathcal{R}^n}{\operatorname{argmin}} f(x)$$

con $f(x) = \sum_{i=1}^{n}(x_i - y_i)^2 + \lambda \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2$

usando dos algoritmos, Descenso de Gradiente y el Método de Newton. Para problemas cuadráticos como el presentado aquí el Método de Newton converge en una sola iteración echo mostrado en los cuadros comparativos del algoritmo. En la sección Apéndice (V-A) se muestran los resultados a tres problemas con respecto a temas de los algoritmos de búsqueda en linea aquí planteados.

***Index Terms*—Descenso de gradiente, Método de Newton**

I. Introduction

El método de Descenso de Gradiente es uno de los métodos de búsqueda en linea que nos permite resolver problemas de optimización al ayudarnos a encontrar máximos o mínimos dentro de funciones. La idea general detrás de este método es sencilla; dado un punto solución inicial x_0 a una función f (tomada al azar o una aproximación cualquiera) podemos encontrar otra solución que mejore nuestra evaluación (en términos de maximizar o minimizar) en f, con ayuda de el gradiente en ese punto. El gradiente nos indica la dirección de máximo crecimiento (existen más direcciones de crecimiento pero esta es la máxima) en la función, por tanto, moverse ese dicha dirección o en sentido contrario nos permite encontrar máximos o mínimos (No tenemos garantizado la convergencia a un mínimo o máximo global). En general la selección de nuestro nuevo punto solucion se realiza mediante (para minimización) $x_{k+1} = x_k - \alpha \nabla f(x_k)$

El Método de Newton considera la actualización en cada iteración como $x_{k+1} = x_k - \nabla^2 f(x_k) \nabla f(x_k)$, sin embargo la matriz Hessiana $\nabla^2 f(x_k)$ no siempre es definida positiva y $\nabla^2 f(x_k) \nabla f(x_k)$ podría no ser una dirección de descenso. Para resolver este problema se busca obtener una nueva matriz a partir de la Hessiana tal que ésta si sea definida positiva, ie. $B_k = \nabla^2 f(x_k) + E_k$ para ello podemos construir $E_k = \tau_k I$ tal que τ nos asugure que B_k se definida positiva el cual podemos elegir como el valor propio más pequeño del Hessiano, la implementación aquí realizada computa ese valor con ayuda de la factorizando de Cholesky Modificado.

II. Métodología

Cómo se menciona anteriormente el problema a resolver consiste en encontrar

$$x^* = \underset{x \in \mathcal{R}^n}{\operatorname{argmin}} f(x)$$

donde $f(x) = \sum_{i=1}^{n}(x_i - y_i)^2 + \lambda \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2$ usando Descenso de Gradiente y Métodos de Newton, considerando $\lambda \in \{1, 100, 1000\}$

Ambos métodos descritos arriba requieren para su implementación el cálculo del gradiente y el hessiano de la función, las cuales se muestran en el apéndice (V-C).

Cabe notar que en este caso la matriz Hessiana es constante con respecto a el x_k actual calculado en cada iteración y solo depende del parámetro λ elegido.

A continuación se describen de manera general la implementación para los métodos de Descenso de Gradiente y Método de Newton.

Algorithm 1: Descenso de Gradiente

```
Result: x*
x <- Inicializar
α <- Inicializar
while Criterio de parada no se cumpla do
    α <- Actualizar α
    x = x + α - ∇ f(x)
end
```

Algorithm 2: Método de Newton

```
Result: x*
x <- Inicializar
α <- Inicializar
while Criterio de parada no se cumpla do
    B = ∇²f(x)
    if B no es semidefinida positiva then
        B = B + min_eigvalue(B)
    end
    x = x - α * B^-1 ∇ f(x)
end
```

En los dos métodos se usaron 3 criterios de paro:

- $\frac{||x_{k+1} - x_k||}{max(||x_k||, 1)} < \tau_1$
- $\frac{abs(f(x_{k+1}) - f(x_k))}{max(abs(f(x_k)), 1)} < \tau_2$

- $||\nabla f(x_k)|| < \tau_3$

Donde las parámetros a τ_1, τ_2 y τ_3 son elegidos con un valor aproximado a $1x10^{-12}$.

Para el cálculo de α_k usado como el tamaño de para al k-ésima iteración se realizó de de 3 formas;

- Paso Fijo
- Paso Autoadaptable con $alpha_k = \frac{g_k^t g_k}{g_k^t H g_k}$, g_k como el gradiente y H el Hessiano en la k-ésima iteración
- Método de Backtracking

Conservando al final solo el paso Autoadaptable para el Método de Gradiente y el paso fijo con $\alpha = 1$ para el Método de Newton debido a que se estos se comportaron mejor durante las pruebas y cuyo resultados se muestran en la siguiente sección.

La actualización del tamaño de paso mediante el *Backtracking* se realiza de la siguiente mediante el algoritmo listado en el apéndice (V-B).

III. Resultados

Las figuras 1, 2 y 3 muestran los resultados obtenidos al minimizar la función f. Observemos que en la primer sumatoria $\sum_{i=1}^{n}(x_i, y_i)^2$ buscamos minimizar la diferencia entre el Y y X así como se puede observar en la figura 1 la aproximación de x_i a cada y_i es muy buena. Mientras λ crece la aproximación de los valores de cada y_i es menor, debido al castigo por la ponderación que provoca la segunda sumatoria, así entonces un efecto de suavizado aparece en las figuras 2 y 3.

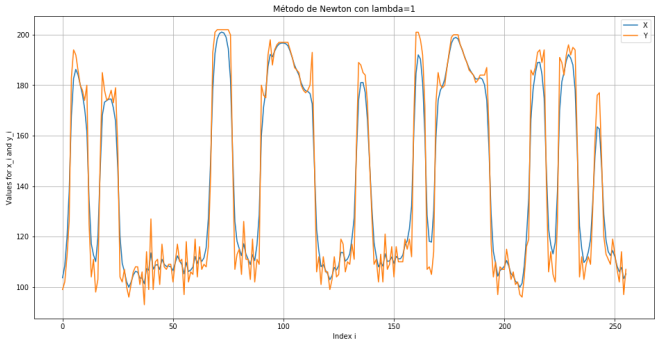

Figura 1. Obtención del minimizador para la función con $\lambda = 1$, se observa una buena aproximación sobre los la serie formada por el vector Y.

Las graficas obtenidas mediante el método de Descenso de Gradiente se muestran en el apéndice (V-D).

Cuadro I
Método de Newton con $\alpha = 1$

λ	*Iteraciones*	*Tiempo*	*Error*
1	2	0.0908	1.00638198e-16
100	2	0.0941	3.25256674e-15
1000	2	0.0959	6.27304335e-15

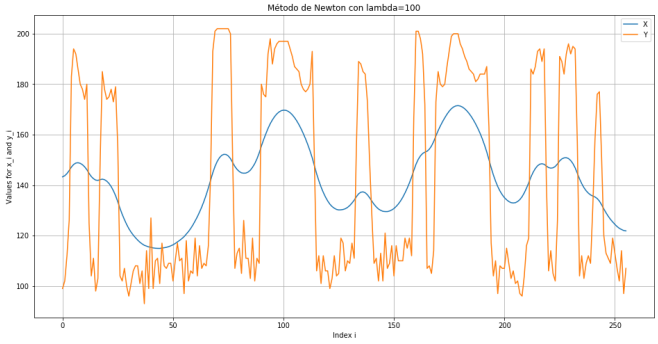

Figura 2. Obtención del minimizador para la función con $\lambda = 100$, se observa un suavizado provocado por el contribución de λ.

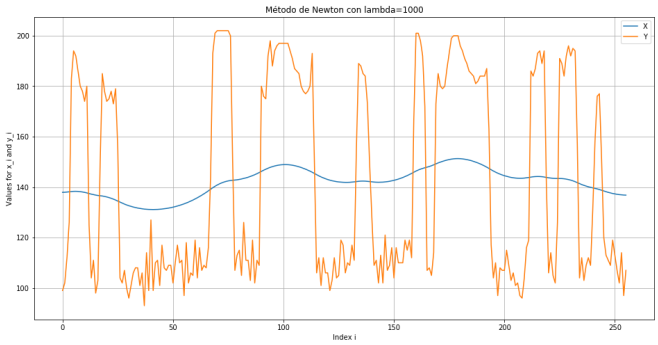

Figura 3. Obtención del minimizador para la función con $\lambda = 1000$, el suavizado de los la serie de los valores de Y es más intenso.

Como se observa en los resultados de la tabla I el Método de Newton la convergencia la alcanza en la primer iteración (Nótese que se necesita al menos dos iteraciones para verificar convergencia), comprobando así la convergencia en un solo paso pára problemas cuadráticos.

Cuadro II
Descenso de Gradiente con α Autoadaptable

λ	*Iteraciones*	*Tiempo*	*Error*
1	32	0.0514	9.90900042e-13
100	2376	3.774	9.98161882e-13
1000	19135	25.155	9.99757740e-13

Cuadro III
Descenso de Gradiente con α Fijo

λ	α	*Iteraciones*	*Tiempo*	*Error*
1	0.017	38	0.0376	8.50063163e-13
100	0.0024	2518	1.161	9.96273014e-13
1000	0.0001	47337	24.396	9.99757738e-13

Las tablas II, III, IV muestran los resultados para el algoritmo de Descenso de Gradiente. La convergencia por este método requiere más iteraciones y mientras el valor de λ aunmenta converge más lento. Los mejores tiempos y número de iteraciones se obtuvieron con α Autoadaptable.

Cuadro IV
DESCENSO DE GRADIENTE CON α MEDIANTE BACKTRACKING

λ	α	***Iteraciones***	***Tiempo***	***Error***
1	0.017	38	0.0368	8.50063163e-13
100	0.0024	2518	1.409	9.96273014e-13
1000	0.0001	47337	27.247	9.99757738e-13

Se usó $\rho = 0{,}001$ y $c1 = 1e-14$

IV. CONCLUSIONES

Para la solución de este problema sin duda el mejor algoritmo fue el Método de Newton, el cual converge con una sola iteración, tal como teóricamente se habia demostrado en clase. Con el algoritmo de Descenso de Gradiente también se obtuvieron muy buenos resultados (con errores de en rango de $1e^{-13}$) pero se requirieron más iteraciones para converger y estas crecen mientras se escoga una λ más grande.

Como vimos, el efecto de elegir un valor de λ mayor repercute directamente en la eficiencia del Algoritmo de método de Descenso de Gradiente pero no es el único efecto que realiza, puesto como se observó en los gráficos anteriores, tiene un efecto en la aproximación a cada valor y_i suavizando el resultado final.

Usando el método de actualización de paso con Backtracking solo se logró hacer simil al de paso fijo, por tanto ambas tablas muestran resultados parecidos.

Cómo un la matriz Hessiana obtenida para los valores de λ es definida positiva la actualización de dicha matriz mediante el algoritmo de la Factorización de Cholesky para encontrar en el valor propio más pequeño no fue necesario, más sin embargo su implementación se encuentra en el código adjunto a este reporte.

V. APÉNDICE

V-A. Problemas

1: Considera la función $f(x_1, x_2) = (x_1 + x_2^2)^2$. En el punto $x^T = [1, 0]$ consideramos la dirección de búsqueda $p^T = [-1, 1]$. Muestra que p es una dirección de descenso y encuentra todos los minizadores de la función.

Solución

Sabemos que un vector p es una dirección de descenso si $\nabla^T f(x)p < 0$, para algún vector x. El gradiente de f está dado por $\nabla f(x_1, x_2) = [2(x_1 + x_2^2) \quad 4(x_1 + x_2^2)x_2]^T$, evaluando en el punto $x^T = [1, 0]$ tenemos que $\nabla f(1, 0) = [2 \quad 0]^T$ entonces $\nabla^T f(1, 0)p = [2 \quad 0]\ [-1 \quad 0]^T = -2 < 0$ y así p es una dirección de descenso.

Para encontrar todos los puntos que minimizan la función, procedemos a encontrar aquellos que son puntos estacionarios igualando el gradiente a cero y resolviendo el sistema de ecuaciones generado.

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 + 2x_2^2 \\ 4x_1x_2 + 4x_2^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

La solución al sistema anterior es $x_! = -x_2^2$, es decir todos aquellos puntos de la forma $(-a^2, a)$, $a \in \mathcal{R}$. Un punto es un minimizador si la matriz Hessiana de f asociada al punto es definida o semidefinida positiva. La matriz Hessiana de f está dada por

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 2 & 4x_2 \\ 4x_2 & 4x_1 + 12x_2^2 \end{pmatrix}$$

Para saber si es definida o semidefinida positiva evaluamos un punto de la forma $(-x_2^2, x_2)$ y calculamos su determinante.

$$det \begin{pmatrix} 2 & 4x_2 \\ 4x_2 & -4x_2^2 + 12x_2^2 \end{pmatrix} = 16x_2^2 - 16x_2^2 = 0$$

Como el determinante es cero, no tenemos demasiada información para tomar una desición. Sin embargo, notemos que siempre $f(x_1, x_2) \geq 0$ para cualquier punto $[x_1, x_2]$ y en especial para los puntos de la forma $(-x_2^2, x_2)$ la función f alcanza el mínimo valor posible; cero. Así todos los puntos de la forma $(-x_2^2, x_2)$ minimiza a f. Por inspección podemos ver el comportamiento de la función en 4 para corroborar visualmente que efectivamente esa serie de puntos son minimizadores.

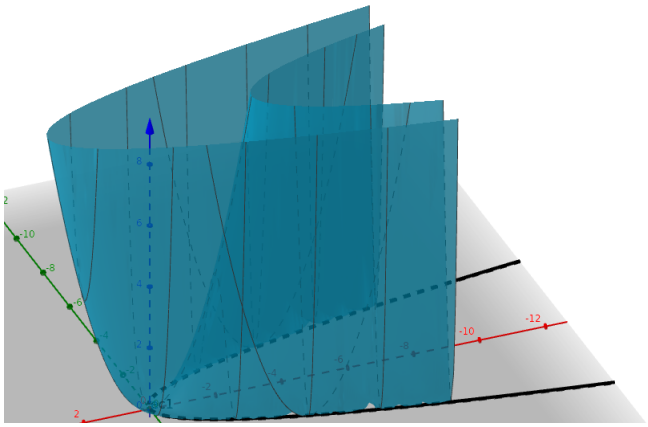

Figura 4. Para problema 1. De azul la función $f(x_1.x_2) = (x_1, x_2^2)^2$, de negro todos los puntos de la forma $(-x_2^2, x_2)$ que minimizan a f.

2: Encuentra todos los valores del parámetro a tal que $[1, 0]^T$ es un minizador o maximizador de la función $f(x_1, x_2) = a^3x_1e^{x_2} + 2a^2log(x_1 + x_2) - (a + 2) + 8ax_2 + 16x_1x_2$.

Solución

Cómo el punto $[1, 0]^T$ es un minimizador o maximizador de f entonces $\nabla f(1, 0) = [0, 0]$. Obtenemos primeramente el gradiente de f el cual es $\nabla f(x_1, X_2) = [a^3e^{x_2} + \frac{2a^2}{x_1+x_2} - (a+2) + 16x_2 \quad a^3e^{x_2} + \frac{2a^2}{x_1+x_2} + 8a + 16x_1]^T$ (asumiendo que *log* es logaritmo natural). Sustiyendo el punto $[1, 0]^T$ obtenemos el sistema de ecuaciones $a^3 + 2a - a - 2 = 0$ y $a^3 + 2a + 8a + 16 = 0$, factorizando

tenemos $(a+2)(a^2-1)=0$ y $(a+2)(a^2-8)=0$. Así, para solo para $a=-2$ se cumple que $\nabla f(1,0)=[0,0]$.

Evaluamos $a=-2$ el el Hessiano de f para comprobar que es un minimizador o un maximizador, donde el Hessiano está dado por

$$\nabla^2 f = \begin{pmatrix} -\frac{2a^2}{(x_1+x_2)^2} & a^3e^{x_2} - \frac{2a^2}{(x_1+x_2)^2} + 16 \\ a^3e^{x_2} - \frac{2a^2}{(x_1+x_2)^2} + 16 & a^3x_1e^{x_2} - \frac{2a^2}{(x_1+x_2)^2} \end{pmatrix}$$

Evaluando en el punto punto $[1,0]^T$ y con $a=-2$

$$\begin{pmatrix} -2a^2 & a^3-2a^2+16 \\ a^3-2a^2+16 & a^3-2a^2 \end{pmatrix} = \begin{pmatrix} -8 & 0 \\ 0 & -16 \end{pmatrix}$$

Podemos ver que la matriz Hessiana es definida negativa puesto que los sus valores propios son $\lambda=-8$ y $\lambda=-16$ por tanto para $a=-2$ el punto $[1,0]^T$ es un maximizador de f.

3: Sea $f:\mathcal{R}^n \to \mathcal{R}$ dado por $f(x)=\frac{1}{2}x^TQx-b^Tx$ con $b\in\mathcal{R}^n$ y $Q\in\mathcal{R}^{nxn}$ una matriz real simetrica y definida positiva. Considera el algoritmo $x^{k+1}=x^k-\beta\alpha_k\nabla f(x^k)$ donde $\alpha_k=\frac{\nabla f(x^k)^T\nabla f(x^k)}{\nabla f(x^k)^TQ\nabla f(x^k)}$. Muestra que $\{x^k\}$ converge a $x^*=Q^{-1}b$ para cualquier punto inicial x^0 si y solo si $0<\beta<2$.

Solución

Notemos que $\nabla f(x)=Qx_k-b$ y x^* minimiza la función y como es solución de $Qx^*-b=0$ converge a $Q^{-1}b$. Por otro lado, dada una secuencia de $\{x^k\}$ resultante de la actualización $x^{k+1}=x^k-\alpha_k g_k$, con g_k como el gradiente en la k-ésima iteración del algoritmo de descenso de gradiente y sea $\gamma_k=\alpha_k\frac{g_k^tQg_k}{g_k^tQ^{-1}g_k}\left(2\frac{g_k^tg_k}{g_k^tQg_k}-\alpha_k\right)$ con $\gamma_k>0$ para todo k, entonces, $\{x^k\}$ converge a x^* para cualquier punto inicial x^0 si y solo si $\sum_{k=1}^{\infty}\gamma_k=\infty$.

En particular para la actualización anterior y un tamaño de paso $\alpha_k=\frac{g_k^tg_k}{g_k^tQg_k}$ la secuencia $\{x^k\}$ converge a x^* para cualquier condición inicial x^0, en otras paralabras se cumple que $\sum_{k=1}^{\infty}\gamma_k=\infty$.

Ahora, sea $\alpha\prime_k=\beta\alpha_k=\beta\frac{g_k^tg_k}{g_k^tQg_k}$, luego

$\gamma\prime_k=\beta\frac{g_k^tg_k}{g_k^tQg_k}\frac{g_k^tQg_k}{g_k^tQ^{-1}g_k}\left(2\frac{g_k^tg_k}{g_k^tQg_k}-\beta\frac{g_k^tg_k}{g_k^tQg_k}\right)$

$\gamma\prime_k=\beta\frac{g_k^tg_k}{g_k^tQg_k}\frac{g_k^tQg_k}{g_k^tQ^{-1}g_k}\left((2-\beta)\frac{g_k^tg_k}{g_k^tQg_k}\right)$

$\gamma\prime_k=\frac{g_k^tg_k}{g_k^tQg_k}\frac{g_k^tg_k}{g_k^tQg_k}(2-\beta)\beta$

$\gamma\prime_k=\gamma_k(2-\beta)\beta$

Por la desigualdad de Rayleigh sabemos que $\frac{a}{A}\leq\gamma_k\leq\frac{A}{a}$, donde a y A son los valores propios (siempre positivos) de Q. Entonces para $\frac{a}{A}(2-\beta)\beta\leq\gamma_k(2-\beta)\beta\leq\frac{A}{a}(2-\beta)\beta$ se cumple que $\sum_{k=1}^{\infty}\gamma\prime_k=\sum_{k=1}^{\infty}\gamma_k(2-\beta)\beta=\infty$ solo para $\beta\in(0,2)$, puesto que como $\gamma>0$ y con $\beta\geq 2$ o $\beta\leq 0$ el producto $\gamma_k(2-\beta)\beta$ es ≤ 0.

V-B. Algoritmo para actualización de tamaño de paso Backtracking

Algorithm 3: Método de Backtracking para tamaño de paso

Result: α
α <- Inicializar
ρ <- Inicializar
$c1$ <- Inicializar
while $f(x_k-\alpha*g_k)>f(x_k)+c1*g_k^Tg_k$ **do**
  $\alpha=\alpha*\rho$
end

V-C. Gradiente y Hessiano de la función a minimizar

$$f(x)=\sum_{i=1}^{n}(x_i-y_i)^2+\lambda\sum_{i=1}^{n-1}(x_{i+1}-x_i)^2$$

$$\nabla f = \begin{pmatrix} 2(x_0-y_0)-2\lambda(x_1-x_0) \\ 2(x_1-y_1)-2\lambda(x_2-x_1)+2\lambda(x_1-x_0) \\ 2(x_2-y_2)-2\lambda(x_3-x_2)+2\lambda(x_2-x_1) \\ \vdots \\ 2(x_{n-2}-y_{n-2})-2\lambda(x_{n-1}-x_{n-2})+2\lambda(x_{n-2}-x_{n-3}) \\ 2(x_{n-1}-y_{n-1})-2\lambda(x_n-x_{n-1})+2\lambda(x_{n-1}-x_{n-2}) \\ 2(x_n-y_n)+2\lambda(x_n-x_{n-1}) \end{pmatrix}$$

$$\nabla^2 f = \begin{pmatrix} 2+2\lambda & -2\lambda & 0 & \dots & 0 & 0 & 0 \\ -2\lambda & 2+4\lambda & -2\lambda & \dots & 0 & 0 & 0 \\ 0 & -2\lambda & 2+4\lambda & \dots & 0 & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & \dots & 2+4\lambda & -2\lambda & 0 \\ 0 & 0 & 0 & \dots & -2\lambda & 2+4\lambda & -2\lambda \\ 0 & 0 & 0 & \dots & 0 & -2\lambda & 2+2\lambda \end{pmatrix}$$

V-D. Gráficos Usando el Método de Descenso de Gradiente

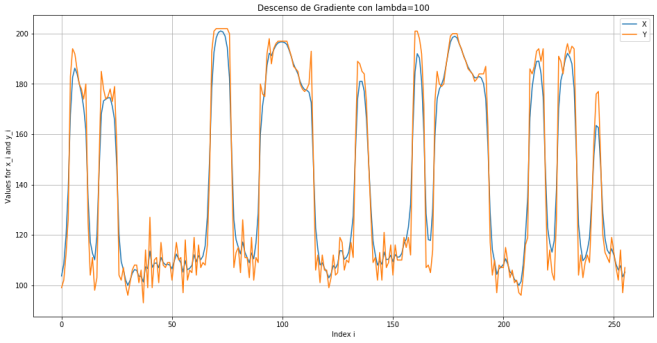

Figura 5. Obtención del minimizador para la función con $\lambda=1$, el suavizado de los la serie de los valores de Y es más intenso.

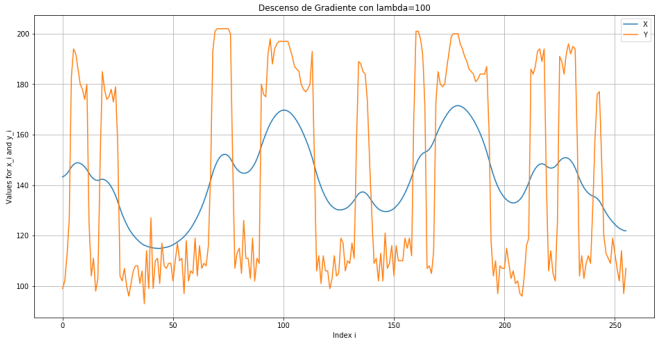

Figura 6. Obtención del minimizador para la función con $\lambda = 100$, el suavizado de los la serie de los valores de Y es más intenso.

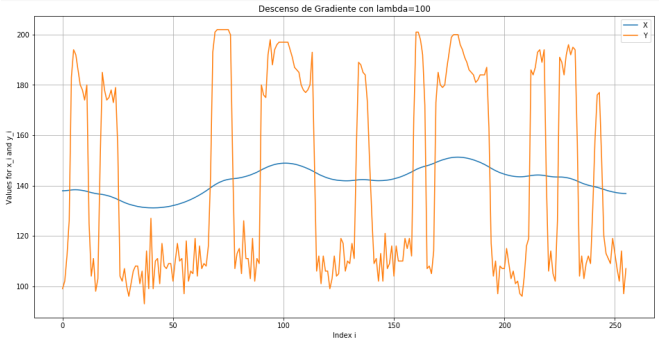

Figura 7. Obtención del minimizador para la función con $\lambda = 1000$, el suavizado de los la serie de los valores de Y es más intenso.

Referencias

[1] Jorge Nocedal, Stephen J. Wright, "Numerical Optimization," Second Edition, Springer.