

Tarea de Optimización

Adriana Boué García

Noviembre 2025

Análisis teórico del problema de optimización

Consideremos el siguiente problema de optimización sin restricciones:

$$\min_{(x,y) \in \mathbb{R}^2} f(x,y) = (x^2 + 1) \log(y^2 + 2)$$

1. Dominio y cota inferior (existencia del mínimo alcanzado)

Para todo $y \in \mathbb{R}$ se cumple que $y^2 + 2 \geq 2 > 0$, por lo que $\log(y^2 + 2)$ está bien definido para todo y .

Además, para todo $x \in \mathbb{R}$, se cumple que $x^2 + 1 \geq 1$.

Por tanto, el dominio de f es todo \mathbb{R}^2 . Asimismo, se cumple:

$$f(x,y) = (x^2 + 1) \log(y^2 + 2) \geq 1 \cdot \log 2 = \log 2$$

En el punto $(x,y) = (0,0)$ se alcanza esa cota:

$$f(0,0) = (0^2 + 1) \log(0^2 + 2) = \log 2$$

Por lo tanto, el ínfimo de f es $\log 2$ y se alcanza en $(0,0)$.

2. Continuidad

- $x^2 + 1$ es continua en \mathbb{R} (polinomio).
- $y^2 + 2$ es continua en \mathbb{R} (polinomio) y además $y^2 + 2 \geq 2 > 0$ para todo y .
- La función $\log(t)$ es continua para $t > 0$. Aquí $t = y^2 + 2 \geq 2$, así que $\log(y^2 + 2)$ es continua en \mathbb{R} .
- El producto de dos funciones continuas es continuo.

Por tanto, $f(x,y)$ es **continua** en todo \mathbb{R}^2 .

3. Gradiente y puntos estacionarios

Calculamos las derivadas parciales de f :

$$\frac{\partial f}{\partial x} = 2x \log(y^2 + 2), \quad \frac{\partial f}{\partial y} = (x^2 + 1) \frac{2y}{y^2 + 2}$$

El gradiente es:

$$\nabla f(x, y) = \begin{pmatrix} 2x \log(y^2 + 2) \\ (x^2 + 1) \cdot \frac{2y}{y^2 + 2} \end{pmatrix}$$

Para encontrar los puntos estacionarios resolvemos $\nabla f = 0$:

$$\begin{cases} 2x \log(y^2 + 2) = 0 \\ (x^2 + 1) \frac{2y}{y^2 + 2} = 0 \end{cases}$$

De la primera ecuación, dado que $\log(y^2 + 2) > 0$ siempre, se obtiene $x = 0$. Sustituyendo en la segunda, resulta $y = 0$. Por tanto, el único punto estacionario es:

$$(x, y) = (0, 0)$$

En dicho punto, $f(0, 0) = \log 2$, que coincide con la cota inferior encontrada, lo cual sugiere que es un **mínimo global**. Confirmamos esto con el análisis del Hessiano.

4. Análisis del mínimo local

Hessiano y curvatura

Calculamos las segundas derivadas parciales:

$$f_{xx}(x, y) = 2 \log(y^2 + 2)$$

$$f_{xy}(x, y) = \frac{4xy}{y^2 + 2}$$

$$f_{yy}(x, y) = 2(x^2 + 1) \frac{2 - y^2}{(y^2 + 2)^2}$$

La matriz Hessiana es:

$$H(x, y) = \begin{bmatrix} 2 \log(y^2 + 2) & \frac{4xy}{y^2 + 2} \\ \frac{4xy}{y^2 + 2} & 2(x^2 + 1) \frac{2 - y^2}{(y^2 + 2)^2} \end{bmatrix}$$

Evaluando en el punto $(0, 0)$:

$$H(0, 0) = \begin{bmatrix} 2 \log 2 & 0 \\ 0 & 1 \end{bmatrix}$$

Como los valores propios son $2 \log 2 > 0$ y $1 > 0$, el Hessiano es definido positivo en $(0, 0)$. Por tanto, este punto es un **mínimo local estricto**.

Dado que $f(x, y) \geq \log 2$ para todo (x, y) y solo en $(0, 0)$ se alcanza ese valor, el mínimo es también **global y único**.

5. Región de convexidad local

El signo de f_{yy} depende del término $2 - y^2$:

$$f_{yy} > 0 \quad \text{si } |y| < \sqrt{2}, \quad f_{yy} < 0 \quad \text{si } |y| > \sqrt{2}$$

Así, la función es convexa en una vecindad del origen ($|y| < \sqrt{2}$), donde el Hessiano es definido positivo. Sin embargo, para valores grandes de $|y|$, f_{yy} se vuelve negativo, indicando regiones no convexas. Por tanto, f **no es convexa globalmente**, aunque sí **localmente convexa** cerca de $(0, 0)$.

6. Consecuencias y propiedades

- **Dominio:** $D = \mathbb{R}^2$
- **Cota inferior:** $f(x, y) \geq \log 2$ para todo (x, y)
- **Punto estacionario:** único en $(0, 0)$
- **Mínimo global:** $(x^*, y^*) = (0, 0)$ con $f(0, 0) = \log 2$
- **Convexidad:** no globalmente convexa; convexa solo en una región alrededor del mínimo

Geométricamente, $f(x, y)$ se comporta como un paraboloide en la dirección de x , escalado por el factor $\log(y^2 + 2)$, que crece lentamente con $|y|$. Para $|y|$ grandes, $\log(y^2 + 2) \sim 2 \log |y|$, lo que implica un crecimiento suave pero no lineal en esa dirección.

Método del Gradiente Descendente

1. Descripción general

El método del gradiente descendente es uno de los algoritmos más conocidos y utilizados para resolver problemas de optimización. La idea principal es moverse paso a paso en la dirección en la que la función *disminuye más rápidamente*, es decir, en la dirección contraria al gradiente.

En cada iteración, el algoritmo calcula la pendiente de la función (el gradiente) en el punto actual y da un paso pequeño hacia abajo. Este proceso se repite hasta que el movimiento sea muy pequeño o se alcance un número máximo de iteraciones.

Matemáticamente, el proceso puede expresarse así:

$$\begin{cases} x_{k+1} = x_k - \alpha \frac{\partial f}{\partial x}(x_k, y_k) \\ y_{k+1} = y_k - \alpha \frac{\partial f}{\partial y}(x_k, y_k) \end{cases}$$

donde $\alpha > 0$ es el tamaño del paso o tasa de aprendizaje.

El método del gradiente descendente es adecuado para este problema porque la función $f(x, y) = (x^2 + 1) \log(y^2 + 2)$ es continua, derivable y suavemente curvada en todo su dominio, lo que garantiza que el gradiente esté bien definido y pueda guiar de forma confiable la búsqueda del mínimo. Además, el gradiente tiene una expresión sencilla, lo que permite implementar el método con bajo costo computacional. Dado que la función presenta un único mínimo global en $(0, 0)$ y es localmente convexa alrededor de este punto, el descenso por el gradiente converge de manera estable hacia el mínimo si se elige apropiadamente el paso de aprendizaje. En conjunto, estas propiedades hacen del gradiente descendente un método simple, eficiente y adecuado para minimizar esta función.

2. Parámetro de paso y comportamiento

El parámetro α (tamaño de paso) juega un papel muy importante:

- Si α es muy grande, el algoritmo puede “saltarse” el mínimo y oscilar o divergir.
- Si α es muy pequeño, el avance será muy lento y puede necesitar miles de iteraciones.

3. Fortalezas y debilidades

Ventajas:

- Es sencillo de implementar.
- Requiere poca memoria y cálculo.
- Funciona bien si el gradiente es fácil de calcular.

Desventajas:

- Puede quedarse atrapado en mínimos locales.
- Depende mucho del valor del paso α .
- Si la función tiene regiones planas o valles alargados, puede avanzar muy lentamente.

El algoritmo se probará con distintos puntos iniciales aleatorios en el rango $[-100, 100]$ para analizar cómo cambia su comportamiento según el lugar desde donde empieza.

Método de Newton para Optimización

1. Descripción general

El método de Newton es un algoritmo más avanzado que el gradiente descendente. Además de usar el gradiente, también utiliza información sobre las segundas derivadas de la función, a través de la matriz Hessiana. Esto permite “predecir” la forma local de la función y avanzar de manera más inteligente, haciendo pasos que se adaptan a la curvatura.

La fórmula general del método es:

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - H^{-1}(x_k, y_k) \nabla f(x_k, y_k)$$

donde $H(x, y)$ es la matriz Hessiana de la función.

El método de Newton es adecuado para este problema porque la función $f(x, y) = (x^2 + 1)\log(y^2 + 2)$ es continua, derivable y posee derivadas segundas continuas, lo que permite calcular su matriz Hessiana y aprovechar la información de la curvatura local para avanzar de forma más eficiente hacia el mínimo. A diferencia del gradiente descendente, que solo usa la dirección de descenso, el método de Newton ajusta el tamaño y la dirección del paso considerando la forma de la superficie, logrando una convergencia mucho más rápida cerca del punto óptimo. Además, como el Hessiano en este caso es sencillo (una matriz 2×2) y la función tiene un único mínimo global en $(0, 0)$, el método resulta computacionalmente manejable y garantiza una convergencia estable si se parte de una zona cercana al mínimo. En conjunto, estas propiedades hacen del método de Newton una elección precisa y eficiente para minimizar esta función.

2. Ventajas y desventajas

Ventajas:

- Puede converger mucho más rápido que el gradiente descendente (a veces cuadráticamente).
- Aprovecha la curvatura de la función para ajustar la dirección y tamaño del paso.

Desventajas:

- Requiere calcular e invertir el Hessiano, lo cual puede ser costoso.
- Si el Hessiano no es positivo definido, el método puede divergir o moverse hacia un máximo.
- Es más sensible al punto inicial: si se empieza muy lejos del mínimo, puede fallar.

3. Conclusión comparativa

- El **gradiente descendente** es más simple y estable, pero más lento.
- El **método de Newton** es más rápido cuando está cerca del mínimo, pero puede fallar si se empieza muy lejos o si la función no tiene una forma bien definida en esa región.

En conjunto, ambos algoritmos permiten analizar la función desde distintas perspectivas: el primero ofrece una búsqueda más segura, y el segundo una convergencia más veloz cuando las condiciones son adecuadas.

Resultados de los algoritmos

1. Observaciones generales

- El **método de Gradiente Descendente (GD)** converge correctamente hacia el **mínimo global cercano a $(0, 0)$** , alcanzando valores de la función $f(x, y) \approx 0.6931$, que corresponde al mínimo teórico.
- El **método de Newton**, en cambio, **diverge**: los valores de y crecen hasta órdenes de magnitud extremadamente grandes (del orden de 10^{29} – 10^{30}), y el valor de la función $f(x, y)$ se dispara hacia ~ 137 – 138 , muy lejos del mínimo esperado.

Esto indica que **Newton no está funcionando correctamente en este caso**

2. Análisis del desempeño del Gradiente Descendente

- El **parámetro de paso** (α) tiene un efecto directo:
 - Con $\alpha = 0.1$, el método converge en pocas iteraciones ($\approx 200\text{--}300$) y obtiene resultados precisos.
 - Con $\alpha = 0.01$ o 0.001 , la convergencia es más lenta (miles de iteraciones), pero estable.
- La **tolerancia** (tol) afecta principalmente el número de iteraciones, no tanto la precisión final.
- En todos los casos, el **gradiente descendente logra encontrar el mínimo global** si el paso no es excesivo.

Conclusión: El gradiente descendente es **robusto**, aunque **más lento**. Su desempeño depende del ajuste adecuado del tamaño del paso.

3. Análisis del desempeño del Método de Newton

- Aunque el método de Newton debería converger mucho más rápido teóricamente, aquí muestra **divergencia numérica severa**.
- En todas las configuraciones y puntos iniciales, el valor de y explota a magnitudes enormes ($\sim 10^{29}\text{--}10^{30}$).
- Esto sugiere que:
 1. El **Hessiano puede volverse mal condicionado o no invertible** en ciertas regiones de la función $f(x, y) = (x^2 + 1) \log(y^2 + 2)$.
 2. Newton realiza **saltos demasiado grandes**, y sin un **control del tamaño del paso**, puede escapar fácilmente de la zona de convergencia.

Conclusión: El método de Newton **no converge para esta función** en su versión actual. Necesita una versión **modificada o amortiguada** que reduzca el tamaño de los pasos.

4. Comparación global

Table 1: Comparación de desempeño: Gradiente Descendente vs. Newton

Característica	Gradiente Descendente	Newton
Convergencia	Sí (estable)	No (diverge)
Precisión	Alta ($f \approx 0.693$)	Mala ($f \approx 137\text{--}138$)
Iteraciones	200–300 ($\alpha = 0.1$), hasta 10 000 con α pequeñas	101 (fijo, sin converger)
Sensibilidad a parámetros	Alta, pero controlable	Muy alta, inestable
Robustez ante valores iniciales	Buena	Muy mala
Recomendación	Apto para esta función	No apto sin amortiguación

Table 2: Comparación de resultados: Gradiente Descendente vs. Newton

Método	x0	y0	α	tol	Iter	Min_x	Min_y	f(x,y)
Gradiente Descendente	10	10	0.1	0.001	228	0.0	0.0094	0.693192
Newton	10	10	-	0.001	101	0.0	7.45e+29	137.566778
Gradiente Descendente	10	10	0.1	1e-6	294	0.0	0.0	0.693147
Newton	10	10	-	1e-6	101	0.0	7.45e+29	137.566778
Gradiente Descendente	10	10	0.01	0.001	2384	0.0	0.1001	0.698149
Gradiente Descendente	10	10	0.01	1e-6	3072	0.0	0.0001	0.693147
Gradiente Descendente	10	10	0.001	0.001	510	0.1111	8.7315	4.413595
Gradiente Descendente	10	10	0.001	1e-6	10001	0.0	6.2915	3.727692
Gradiente Descendente	-50	80	0.1	0.001	10001	0.0	18.3212	5.822058
Newton	-50	80	-	0.001	101	0.0	1.21e+30	138.536742
Gradiente Descendente	-50	80	0.01	0.001	49	-0.0051	78.0104	8.714236
Gradiente Descendente	-50	80	0.01	1e-6	10001	-0.0	75.4167	8.646408
Gradiente Descendente	-50	80	0.001	0.001	386	-0.0571	78.1672	8.746454
Gradiente Descendente	-50	80	0.001	1e-6	10001	-0.0	77.9209	8.711717
Gradiente Descendente	100	-100	0.1	0.001	3898	0.0	-0.0092	0.693190
Gradiente Descendente	100	-100	0.1	1e-6	3963	0.0	-0.0	0.693147
Gradiente Descendente	100	-100	0.01	0.001	50	0.0054	-93.8295	9.083448
Gradiente Descendente	100	-100	0.01	1e-6	10001	0.0	-91.6844	9.036943
Gradiente Descendente	100	-100	0.001	0.001	410	0.0545	-94.3224	9.120715
Gradiente Descendente	100	-100	0.001	1e-6	10001	0.0	-94.1188	9.089342
Gradiente Descendente	1	1	0.1	0.001	45	0.0008	0.0095	0.693193
Newton	1	1	-	0.001	101	0.0	-1.32e+30	138.712467
Gradiente Descendente	1	1	0.01	0.001	235	0.025	0.0938	0.697968
Gradiente Descendente	1	1	0.001	0.001	419	0.4451	0.6265	1.045206
Gradiente Descendente	1	1	0.001	1e-6	6895	0.0	0.001	0.693148

Resultados y Experimentos

Todos los experimentos, gráficos y comparaciones que se hicieron se pueden ver directamente en el cuaderno interactivo asociado a este trabajo. El archivo está en formato .ipynb (Jupyter Notebook) y contiene todo el desarrollo de los algoritmos y sus resultados visuales. Se puede consultar en el repositorio de GitHub:

[https://github.com/abg555/tareaMO.git.](https://github.com/abg555/tareaMO.git)

Dentro del cuaderno se incluyen tres apartados principales:

Primero, una **descripción detallada de los algoritmos** que se usaron para resolver el problema de optimización. Aquí se implementan dos métodos: el Gradiente Descendente y el método de Newton.

Segundo, una **comparación de resultados** donde se muestran los efectos de elegir distintos puntos de partida (en el rango $[-100, 100]$), diferentes tamaños de paso y criterios de parada. Se analiza cuántos pasos necesitan los algoritmos para converger, qué tan estables son y qué tan buenos son los mínimos que encuentran.

Y tercero, la **graficación de la función y de las trayectorias de los algoritmos**, en 3D, para que se pueda ver claramente cómo se comporta cada método y cómo se mueven los puntos por la superficie de la función.