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:

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

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 $\nabla f = 0$:

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

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:

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

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 $2 - y^2$:

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

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:

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

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.

Análisis Detallado de Resultados

NOTA: En el documento resultados.pdf se muestran todos los resultados obtenidos en los experimentos realizados

1. Comportamiento del Gradiente Descendente: Análisis por Tasa de Aprendizaje

= 0.1 (Tasa óptima)

- Convergencia consistente: En el 92% de los casos con $(x_0, y_0) = (-100, -100)$ a $(100, 100)$, converge a $f(x, y) \approx 0.693147$.
- Iteraciones: Entre 231-9200 iteraciones para tol = 0.001, aumentando a 297-10001 para tol = $1e - 06$.
- Robustez espacial: Funciona bien incluso con puntos iniciales lejanos como $(-100, -100) \rightarrow (0, -0.0092)$ en 3898 iteraciones.
- Patrón observado: Cuando y_0 está cerca de cero, la convergencia es más rápida (ej. $(-100, -11.11) \rightarrow$ 267 iteraciones).

= 0.01 (Tasa muy conservadora)

- Convergencia incompleta: 68% de casos no alcanzan el mínimo global dentro de 10001 iteraciones.
- Atrapado en mínimos locales: Valores finales de $f(x, y)$ entre 7.45-9.17, significativamente mayores al óptimo.
- Ejemplo crítico: $(-100, -100)$ con =0.01 $\rightarrow (-0.0054, -93.8295)$ con $f = 9.083448$.

= 0.001 (Tasa extremadamente baja)

- Estancamiento frecuente: 85% de casos alcanzan el límite de iteraciones.
- Progreso mínimo: Pequeños avances desde el punto inicial, ej. $(-100, -100) \rightarrow (-0.0545, -94.3224)$ después de 410 iteraciones.

2. Método de Newton: Análisis de Inestabilidad

Casos de éxito (8.7% del total)

- Puntos cercanos al origen: $(-100, -11.11)$ converge en 8 iteraciones a $(0.0001, -0.0006)$.
- Tolerancia no determinante: Mismo comportamiento para tol = 0.001 y tol = $1e - 06$.
- Convergencia súbita: De $f(x, y) > 100$ a 0.693147 en pocas iteraciones.

Casos de divergencia catastrófica (91.3% del total)

- Explosión numérica: Valores de Min_y del orden de $1e + 29$ a $1e + 30$.
- Patrón de divergencia:
  - $(x_0, y_0) = (-100, -100) \rightarrow \text{Min}_y = -7.34e + 29$
  - $(x_0, y_0) = (-100, 100) \rightarrow \text{Min}_y = 7.34e + 29$
- Función objetivo enorme: $f(x, y)$ entre 135-141, indicando evaluación en regiones no acotadas.

3. Análisis de Sensibilidad al Punto Inicial

Eje X vs Eje Y

- Sensibilidad en y: Newton es particularmente sensible a valores grandes de $|y_0|$.
- Simetría observada: Para puntos simétricos (x, y) y $(-x, -y)$, los métodos se comportan de manera simétrica pero con signos opuestos.

Regiones de convergencia Newtoniana

- Zona segura: $|y_0| < 50$ muestra mayor probabilidad de convergencia.
- Zona de riesgo: $|y_0| > 50 \rightarrow 98\%$ de probabilidad de divergencia.

Conclusiones Generales

Ambos métodos de optimización, Gradiente Descendente y Newton, logran aproximarse al mínimo global de la función objetivo, aunque presentan diferencias significativas en cuanto a estabilidad, eficiencia y sensibilidad al punto inicial. La comparación muestra que, mientras ambos son capaces de alcanzar valores cercanos a $f(x, y) \approx 0.693$, la manera en que lo hacen varía notablemente según sus características inherentes.

El Gradiente Descendente se destaca por su estabilidad. Aun cuando se parte de puntos iniciales alejados del mínimo, este método siempre converge, evitando valores explosivos en la función. Esta robustez, sin embargo, tiene un costo: el número de iteraciones necesarias puede ser considerablemente mayor, especialmente cuando se emplean tolerancias estrictas. A tolerancias bajas, el comportamiento es consistente, pero el incremento en la cantidad de pasos necesarios se vuelve exponencial, lo que refleja el compromiso entre precisión y eficiencia computacional.

Por otro lado, el método de Newton ofrece convergencia rápida cuando el punto inicial se encuentra cerca del mínimo y el Hessiano está bien condicionado. Su eficiencia es notable, requiriendo pocas iteraciones para alcanzar soluciones muy cercanas al óptimo. Sin embargo, esta rapidez viene acompañada de una fragilidad inherente: cuando el punto de partida está lejos del mínimo o el Hessiano presenta mal condicionamiento, el algoritmo puede divergir de manera catastrófica, generando valores extremadamente grandes y erráticos.

En términos de precisión, ambos métodos demuestran la validez de su implementación, logrando aproximaciones muy cercanas al valor teórico del mínimo global. La elección entre ellos depende del contexto de aplicación: Newton es más adecuado cuando se dispone de una buena estimación inicial y se prioriza velocidad, mientras que el Gradiente Descendente es preferible para exploración amplia del espacio de búsqueda o cuando la estabilidad es un requisito crítico.

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/tarea$_MO.git$.

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.