



Optimización numérica mediante gradiente descendente: Estudio de convergencia y regularización en problemas convexos y no convexos

Tobio Santiago¹, De Notta Luca²

Departamento de Ingeniería, Universidad de San Andrés, Victoria, Buenos Aires, Argentina.

E-mails: ¹stobio@udea.edu.ar, ²ldenotta@udea.edu.ar

Abstract: En este trabajo se evalúa la performance de diferentes métodos de optimización numérica aplicados a dos problemas: la función de Rosenbrock y regresión lineal del dataset California Housing. Para el primer caso, se analiza el método de gradiente descendente variando la tasa de aprendizaje y las condiciones iniciales, demostrando la importancia de estos hiperparámetros en la convergencia. En el segundo problema, se comparan tres enfoques: cuadrados mínimos con pseudoinversa, gradiente descendente y gradiente descendente con momentum, incluyendo una variante con regularización L2. Los resultados muestran que momentum acelera significativamente la convergencia, reduciendo el número de iteraciones en un 82% con $\beta = 0.95$. La regularización L2 no impacta significativamente en el condicionamiento del problema. Este estudio proporciona insights sobre la elección óptima de hiperparámetros y el balance entre velocidad de convergencia y estabilidad numérica en problemas de optimización.

Keywords: Optimización numérica, Gradiente descendente, Función de Rosenbrock, Regresión lineal, Momentum, Regularización L2

1 Introducción

La optimización numérica constituye una herramienta fundamental en diversos campos de la ciencia e ingeniería, siendo particularmente relevante en el contexto del aprendizaje automático y el análisis de datos. Entre los diversos métodos de optimización, el gradiente descendente y sus variantes han emergido como técnicas predominantes debido a su simplicidad conceptual y eficacia computacional.

En este trabajo, analizamos el comportamiento y rendimiento de diferentes métodos de optimización basados en gradiente, centrándonos en dos casos de estudio: la función de Rosenbrock, una función no convexa comúnmente utilizada como benchmark en problemas de optimización, y un problema de regresión lineal aplicado al dataset California Housing. El primer caso nos permite estudiar el comportamiento de los métodos en un entorno controlado y bien caracterizado, mientras que el segundo provee un contexto práctico de aplicación real.

Específicamente, para la función de Rosenbrock, evaluamos el método de gradiente descendente bajo diferentes tasas de aprendizaje y condiciones iniciales, contrastándolo con el método de Newton. En el caso de la regresión lineal, comparamos la eficacia del gradiente descendente con la solución analítica mediante pseudoinversa, explorando además el impacto de la regularización L2 en los coeficientes del modelo.

La metodología empleada se centra en el análisis experimental, donde evaluamos cada método bajo diferentes hiperparámetros y condiciones, utilizando métricas como el error cuadrático medio y la velocidad de convergencia. Complementamos este análisis con visualizaciones que permiten comprender mejor el comportamiento de cada método.

Este documento está organizado de la siguiente manera: en la Sección 2 presentamos el marco teórico necesario para comprender los métodos estudiados. La Sección 3 detalla el desarrollo experimental realizado. Los resultados obtenidos y su análisis se presentan en la Sección 4. Finalmente, en la Sección 5 exponemos las conclusiones del trabajo.

2 Marco Teórico

2.1 Funciones convexas

Una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es convexa si su dominio es un conjunto convexo y para cualquier par de puntos x, y en su dominio y cualquier $t \in [0, 1]$ se cumple:

$$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y) \quad (1)$$

Intuitivamente, esta definición implica que el segmento que une dos puntos cualesquiera del gráfico de f está siempre por encima de este. Una propiedad fundamental de las funciones convexas es que todo mínimo local es también un mínimo global. Las funciones convexas poseen características que las hacen especialmente útiles en optimización:

- **Condición de primer orden:** Si f es convexa y diferenciable, para todo x, y en el dominio vale que:

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad (2)$$

Geométricamente, esta condición implica que el hiperplano tangente a la función en cualquier punto x está siempre por debajo del gráfico de la función.

- **Condición de segundo orden:** Si f es dos veces diferenciable, entonces es convexa si y solo si su matriz Hessiana es semidefinida positiva:

$$H(x) = \nabla^2 f(x) \geq 0 \quad (3)$$

la matriz Hessiana semidefinida positiva significa que la curvatura de la función es no negativa en todas las direcciones. En otras palabras, la función se curva hacia arriba o mantiene una pendiente constante, pero nunca se curva hacia abajo.

La convexidad es una propiedad deseable en optimización porque garantiza que los métodos de descenso convergerán al mínimo global de la función. Esto simplifica significativamente el proceso

de optimización, ya que no existe el riesgo de quedar atrapado en mínimos locales. Los algoritmos de descenso convergerán al mínimo global de la función.

2.2 Método de gradiente descendente

El método de gradiente descendente es un algoritmo iterativo de primer orden para encontrar el mínimo de una función diferenciable. La idea central es que el gradiente de una función apunta en la dirección de máximo crecimiento, por lo que el negativo del gradiente indica la dirección de máximo descenso. El algoritmo actualiza iterativamente la estimación del mínimo mediante:

$$x_{k+1} = x_k - \eta \nabla f(x_k) \quad (4)$$

donde $\eta > 0$ es la tasa de aprendizaje o paso, y $\nabla f(x_k)$ es el gradiente de f evaluado en x_k . La elección de η es crucial para la convergencia del método

- Un η demasiado pequeño resultará en una convergencia innecesariamente lenta.
- Un η demasiado grande puede causar oscilaciones o divergencia.

Para funciones convexas y diferenciables con gradiente Lipschitz continuo (es decir, existe $L > 0$ tal que $|\nabla f(x) - \nabla f(y)| \leq L|x - y|$), el método converge si:

$$0 < \eta < \frac{2}{L} \quad (5)$$

Los criterios de convergencia más comunes son:

- Norma del gradiente: $|\nabla f(x_k)| < \epsilon$
- Diferencia relativa: $|x_{k+1} - x_k| < \epsilon$
- Diferencia en función objetivo: $|f(x_{k+1}) - f(x_k)| < \epsilon$
- Número máximo de iteraciones: $k > k_{max}$

La norma del gradiente es particularmente significativa como criterio de corte porque, en un punto crítico (mínimo, máximo o punto silla), el gradiente es cero. Por lo tanto, cuando $|\nabla f(x_k)|$ es suficientemente pequeño, podemos estar razonablemente seguros de estar cerca de un punto crítico. Además, para funciones convexas, este punto crítico será necesariamente el mínimo global.

2.3 Gradiente Descendente Estocástico

El Gradiente Descendente Estocástico (GDE) es una variante del gradiente descendente tradicional que approxima el gradiente utilizando subconjuntos aleatorios de los datos, lo que lo hace especialmente útil para grandes conjuntos de datos. En lugar de calcular el gradiente sobre todos los datos en cada iteración, el GDE utiliza una muestra aleatoria (mini-batch) en cada paso:

$$x_{k+1} = x_k - \eta \nabla f_B(x_k) \quad (6)$$

donde f_B representa la función de pérdida calculada sobre el mini-batch B . Este enfoque tiene varias ventajas:

- Menor costo computacional por iteración
- Capacidad de escapar de mínimos locales debido al ruido estocástico
- Mejor generalización en problemas de aprendizaje automático

2.4 Momentum

El método de momentum es una modificación del gradiente descendente que incorpora información sobre actualizaciones anteriores para mejorar la convergencia. La idea es acumular un "momento" en la dirección de optimización consistente:

$$v_{k+1} = \beta v_k + \nabla f(x_k) \quad x_{k+1} = x_k - \eta v_{k+1} \quad (7)$$

donde $\beta \in [0, 1]$ es el coeficiente de momentum que determina cuánto influyen las actualizaciones anteriores en el paso actual. Este método ofrece varias ventajas:

- Acelera la convergencia en direcciones de poco cambio
- Reduce oscilaciones en direcciones de alta curvatura
- Ayuda a superar mínimos locales y puntos silla

La intuición física detrás del momentum es similar a una pelota rodando por una superficie: acumula velocidad en las pendientes y mantiene su momento incluso en regiones planas, lo que ayuda a superar pequeñas irregularidades en la superficie de la función objetivo.

2.5 Método de Newton-Raphson

El método de Newton-Raphson utiliza información de segundo orden para optimizar funciones, lo que resulta en una convergencia más rápida que el gradiente descendente cuando se está cerca del óptimo. El método approxima la función objetivo por su expansión de Taylor de segundo orden:

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2} \Delta x^T H(x) \Delta x \quad (8)$$

donde $H(x)$ es la matriz Hessiana de f en x . Para encontrar el mínimo de esta aproximación cuadrática, derivamos respecto a Δx e igualamos a cero:

$$\nabla f(x) + H(x) \Delta x = 0 \quad (9)$$

Por lo tanto, el paso de actualización en el método de Newton-Raphson es:

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k) \quad (10)$$

Este método tiene convergencia cuadrática cuando está cerca del óptimo y la matriz Hessiana es definida positiva

2.6 Cuadrados Mínimos

El método de cuadrados mínimos busca encontrar los parámetros que minimizan la suma de los errores cuadráticos entre las predicciones de un modelo y los valores observados. Para un sistema sobredeterminado $Xw \approx y$, donde $X \in \mathbb{R}^{m \times n}$ con $m > n$, buscamos minimizar:

$$\min_w \|Xw - y\|_2^2 \quad (11)$$

Esta función objetivo es convexa y diferenciable. Su gradiente es:

$$\nabla_w \|Xw - y\|_2^2 = 2X^T(Xw - y) \quad (12)$$

Igualando el gradiente a cero, obtenemos el sistema de ecuaciones normales:

$$X^T X w = X^T y \quad (13)$$

La solución analítica viene dada por:

$$w = (X^T X)^{-1} X^T y = X^+ y \quad (14)$$

donde X^+ es la pseudoinversa de Moore-Penrose de X . Esta puede calcularse mediante la descomposición en valores singulares (SVD) de X :

$$X = U \Sigma V^T \implies X^+ = V \Sigma^+ U^T \quad (15)$$

donde Σ^+ se obtiene de Σ tomando el recíproco de los valores singulares no nulos y transponiéndola. Este enfoque es numéricamente más estable que calcular $(X^T X)^{-1} X^T$ directamente. La pseudoinversa tiene las siguientes propiedades importantes:

- Existe incluso cuando X no es cuadrada o es singular
- Proporciona la solución de norma mínima cuando el sistema es indeterminado
- Minimiza $|Xw - y|_2$ entre todas las posibles soluciones

2.7 Regularización L2 (Ridge Regression)

En problemas de regresión, especialmente cuando se tienen muchas variables predictoras o estas están altamente correlacionadas, el modelo puede sobreajustar los datos de entrenamiento, produciendo coeficientes de gran magnitud que generalizan pobremente a nuevos datos. La regularización L2 aborda este problema añadiendo un término de penalización a la función objetivo:

$$\text{ECM}_\lambda(w) = \frac{1}{n} |y - Xw|_2^2 + \lambda |w|_2^2 \quad (16)$$

donde $\lambda > 0$ es el parámetro de regularización que controla el equilibrio entre el ajuste a los datos y la complejidad del modelo. El término $|w|_2^2$ penaliza coeficientes grandes, favoreciendo soluciones más estables. Para encontrar la solución analítica, calculamos el gradiente e igualamos a cero:

$$\nabla_w \text{ECM}_\lambda = \frac{2}{n} X^T(Xw - y) + 2\lambda w = 0 \quad (17)$$

Despejando, obtenemos la solución de Ridge Regression:

$$w_\lambda = (X^T X + n\lambda I)^{-1} X^T y \quad (18)$$

El parámetro λ cumple varios roles importantes:

- Cuando $\lambda = 0$, recuperamos la solución de mínimos cuadrados ordinaria
- Cuando $\lambda \rightarrow \infty$, los coeficientes tienden a cero
- Para $\lambda > 0$, la matriz $(X^T X + n\lambda I)$ es siempre invertible, incluso si $X^T X$ es singular

La elección de λ es crucial:

- Valores pequeños de λ permiten que el modelo se ajuste mejor a los datos pero con riesgo de sobreajuste
- Valores grandes de λ producen modelos más simples y estables, pero pueden llevar a subajuste
- Un λ óptimo se puede encontrar mediante validación cruzada

Una interpretación geométrica útil es que la regularización L2 restringe la norma euclídea de los coeficientes, efectivamente limitando la complejidad del modelo. Esto se puede visualizar como la intersección de la solución de mínimos cuadrados con una bola de radio determinado por λ en el espacio de parámetros. En términos de valores singulares, si $X = U\Sigma V^T$ es la SVD de X , entonces la solución regularizada puede escribirse como:

$$w_\lambda = V(\Sigma^T \Sigma + n\lambda I)^{-1} \Sigma^T U^T y \quad (19)$$

Esta forma revela que la regularización L2 reduce la influencia de los componentes asociados a valores singulares pequeños, que son los principales responsables de la inestabilidad en la solución no regularizada.

3 Desarrollo Experimental

3.1 Optimización con Gradiente Descendente

En esta sección, llevaremos a cabo un estudio comparativo de métodos de optimización numérica aplicados a la función de Rosenbrock, una función no convexa comúnmente utilizada como benchmark en problemas de optimización. En particular,

analizaremos el desempeño del método de gradiente descendente y del método de Newton para encontrar el mínimo global de esta función, que se encuentra en el punto $(x, y) = (a, a^2)$.

El objetivo principal de este experimento es evaluar y contrastar la performance de ambos métodos bajo diferentes condiciones, prestando especial atención a la influencia de hiperparámetros como la tasa de aprendizaje (η) en el caso del gradiente descendente, y estudiando cómo las condiciones iniciales afectan la convergencia de ambos métodos. A su vez, nos interesarán determinar el número de iteraciones requeridas por cada técnica para alcanzar el mínimo global con una tolerancia prefijada sobre la norma del gradiente. Para llevar a cabo este análisis, comenzaremos estudiando las propiedades analíticas de la función de Rosenbrock, su gradiente y su matriz Hessiana, lo que nos permitirá demostrar la no convexidad del problema. Posteriormente, realizaremos experimentos numéricos variando los hiperparámetros y condiciones iniciales para evaluar el comportamiento y la robustez de cada método.

3.1.1 Función Rosenbrock

La función de Rosenbrock, también conocida como función banana debido a su forma característica, está definida como:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2, \quad (20)$$

donde a y b son parámetros que definen la forma del valle y la pendiente de la función respectivamente. Sin embargo, en este trabajo vamos a analizar la performance del gradiente descendente para hallar el mínimo global de esta función tomando $a = 1$, $b = 100$

3.1.2 Experimentos numéricos

Los experimentos numéricos se llevaron a cabo utilizando Python con las siguientes librerías: NumPy para operaciones matemáticas y matriciales, Pandas para el manejo y análisis de datos, y Matplotlib para la visualización de resultados. Las implementaciones de los métodos de gradiente descendente y Newton-Raphson fueron desarrolladas como funciones propias, verificadas contra implementaciones de referencia.

En primer lugar, se realizó un estudio de la sensibilidad del método de gradiente descendente a las condiciones iniciales. Para ello, se seleccionó un conjunto de seis puntos iniciales distribuidos en diferentes regiones del dominio de la función: $(1, 1.5)$, $(1.5, 1)$, $(2, 3)$, $(2, 1)$, $(1, 2)$ y $(3, 3)$. Estos puntos fueron elegidos para evaluar el comportamiento del método tanto cerca como lejos del mínimo global conocido en $(1, 1)$. Para cada punto inicial, se ejecutó el algoritmo con una tasa de aprendizaje fija de $\eta = 10^{-3}$ y se registró tanto la condición de corte alcanzada como la distancia euclídea al mínimo global.

Subsecuentemente, se llevó a cabo un análisis de la influencia de la tasa de aprendizaje en la convergencia del método. Se consideró un rango logarítmico de tasas de aprendizaje $\eta \in [10^{-6}, 10^{-2}]$ con 15 valores equiespaciados en escala logarítmica. Para este experimento, se fijó el punto inicial en $(2, 2)$ para aislar el efecto de la tasa de aprendizaje en la convergencia. Para cada η , se registró si el método convergió, la condición de corte alcanzada y la distancia al óptimo global.

Finalmente, para contrastar con el método de gradiente descendente, se evaluó el método de Newton-Raphson utilizando el mismo conjunto de puntos iniciales del primer experimento. En este caso, se registró para cada punto inicial si el método convergió al mínimo global (considerando una tolerancia de 10^{-4} en la distancia euclídea), su condición de corte y el punto de convergencia alcanzado. Este enfoque permitió calcular la tasa de éxito del método de Newton-Raphson en términos del porcentaje de puntos iniciales que llevaron a una convergencia exitosa al mínimo global.

Para todos los experimentos, se utilizó como criterio de convergencia que la norma del gradiente fuera menor que una tolerancia prefijada $\epsilon = 10^{-6}$, o que se alcanzara un número máximo de 10000 iteraciones. Adicionalmente, se implementaron salvaguardas para detectar y manejar casos de divergencia numérica o overflow en las evaluaciones del gradiente y el Hessiano.

3.2 Cuadrados mínimos mediante descenso por gradiente

En este experimento, se utilizó el conjunto de datos California Housing para realizar regresión lineal con distintos enfoques. Los datos fueron divididos en conjuntos de entrenamiento y prueba utilizando dos métodos: muestreo estratificado y muestreo simple aleatorio. A continuación, se detallan las etapas principales de preprocessamiento y preparación de los datos

División de los datos: El dataset original se dividió en conjuntos de entrenamiento y prueba, asegurando una proporción del 80% para entrenamiento y el 20% restante para prueba. Dado que la variable objetivo (el precio de las viviendas) tiene una distribución sesgada, se implementaron dos estrategias para la división:

- **Muestreo Simple Aleatorio:** División sin aplicar técnicas de balanceo.
- **Muestreo Estratificado:** Creación de quintiles con la variable objetivo para preservar su distribución en ambos conjuntos.

La técnica de estratificación fue implementada mediante la función `pd.qcut` de Pandas, generando etiquetas categóricas que fueron utilizadas como criterio para el muestreo estratificado. Este preprocessamiento garantiza que los modelos sean comparables y que los resultados obtenidos sean válidos y representativos

3.2.1 Solución Analítica (Pseudoinversa)

El enfoque inicial consistió en calcular los coeficientes óptimos mediante la solución analítica usando la pseudoinversa, definida como $w = (X^T X)^{-1} X^T y$. Se utilizaron los conjuntos escalados de entrenamiento y prueba para evaluar la calidad del ajuste en términos del error cuadrático medio (ECM):

$$ECM = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (21)$$

3.2.2 Evaluación del Gradiente Descendente

El método de gradiente descendente (GD) fue evaluado utilizando diferentes tasas de aprendizaje η . La tasa óptima $\eta_{opt} = 1/\sigma_1^2$, donde σ_1 es el valor singular más grande de X_{train} , se seleccionó como punto de partida. También se probaron tasas menores: $\eta_{opt}/2$, $\eta_{opt}/10$, y $\eta_{opt}/100$

Para cada η , se registraron:

- La evolución del ECM durante las iteraciones.
- El número total de iteraciones necesarias para converger.
- El ECM final en los conjuntos de entrenamiento y prueba.

3.2.3 Evaluación en Conjunto de Entrenamiento

Se realizaron dos experimentos principales utilizando el conjunto de entrenamiento:

- **Pseudoinversa:** Se calcularon las predicciones \hat{y} usando los coeficientes obtenidos mediante pseudoinversa. Los resultados se graficaron comparando \hat{y} con los valores reales y .
- **Gradiente Descendente (GD):** Se generaron las predicciones \hat{y} utilizando los coeficientes optimizados por GD con una tasa de aprendizaje η . El gráfico compara \hat{y} con los valores reales, de manera análoga a la pseudoinversa.

Ambos métodos se representaron mediante gráficos de dispersión, donde la línea punteada roja indica la predicción perfecta ($\hat{y} = y$).

3.2.4 Evaluación en Conjunto de Prueba

Se repitieron los experimentos anteriores en el conjunto de prueba, evaluando:

- Predicciones usando los coeficientes calculados por la pseudoinversa.
- Predicciones usando los coeficientes optimizados por GD.

Se generaron gráficos de dispersión similares para comparar visualmente las predicciones \hat{y} con los valores reales y .

3.2.5 Comparación entre GD Estándar y Momentum

Se evaluó la influencia del momentum en el gradiente descendente ajustando el hiperparámetro β . Para ello:

- Se probaron tres valores de β : 0.9, 0.95 y 0.99.
- Para cada β , se registró la evolución del error cuadrático medio (ECM) durante las actualizaciones de pesos.
- El ECM obtenido mediante la pseudoinversa se incluyó como referencia.

Los resultados se graficaron en escala logarítmica para mostrar la disminución del error con cada actualización.

3.2.6 Tasa de Aprendizaje Óptima

La tasa de aprendizaje $\eta_{opt} = 1/\sigma_1^2$ fue calculada a partir del valor singular máximo σ_1 de X_{train} . Se comparó su rendimiento con tasas menores, dividiendo η_{opt} entre 2, 10 y 100.

3.2.7 Efecto de la Regularización L2 (Ridge Regression)

El algoritmo de gradiente descendente actualiza los coeficientes w en cada iteración utilizando la regla:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta \nabla \text{ECM}_\lambda(\mathbf{w}_k) \quad (22)$$

Donde:

- η es la tasa de aprendizaje.
- $\nabla \text{ECM}_\lambda(\mathbf{w}_k)$ es el gradiente calculado en la iteración actual k .

El gradiente de la función de error cuadrático medio regularizado es 17

Para evaluar el impacto de la regularización L2, se realizaron los siguientes experimentos:

1. **Comparación de Soluciones Exactas y GD:** Se calcularon los coeficientes w utilizando dos métodos: solución exacta (mediante descomposición SVD) y gradiente descendente (GD), ambos con y sin regularización. Se probaron cinco valores de λ : 0, 10^{-4} , 10^{-2} , 1 y 10. Los siguientes criterios se analizaron para cada combinación:

- Error Cuadrático Medio (ECM) en entrenamiento y prueba.
- Norma $\|\mathbf{w}\|$, para evaluar el efecto de "shrinkage".
- Convergencia del error en GD (visualizado en escala logarítmica).

2. **Representación de los Coeficientes del Modelo:** El efecto de λ en los coeficientes del modelo se estudió utilizando la solución exacta. Se graficaron las magnitudes de los coeficientes en función de λ (escala logarítmica), mostrando cómo aumentos en λ reducen las magnitudes de los coeficientes (*regularización fuerte*) hasta aproximarlos a 0.

3. **Convergencia de Gradiente Descendente:** Se analizó la convergencia del error cuadrático medio (ECM) para:

- Gradiente descendente sin regularización.
- Gradiente descendente con regularización y $\lambda = 0.01 \times \sigma_1$, donde σ_1 es el valor singular más grande de X_{train} .

4. **Evaluación de Diferentes Tasas de Aprendizaje para Regularización:** Se analizaron las tasas de aprendizaje η en el contexto de gradiente descendente con regularización Ridge, utilizando el valor de $\lambda =$

$0.01 \times \sigma_1$, donde σ_1 es el valor singular más grande de X_{train} . Las tasas probadas fueron:

$$\eta_{\text{opt}}, \frac{\eta_{\text{opt}}}{2}, \frac{\eta_{\text{opt}}}{5}, \frac{\eta_{\text{opt}}}{10}$$

Para cada η , se evaluaron:

- La evolución del error cuadrático medio (ECM) durante las iteraciones.
- El número total de iteraciones necesarias para converger.
- El ECM final alcanzado.

La Figura 12 muestra las curvas de convergencia del ECM para cada η , comparadas con:

- Gradiente descendente sin regularización.
- Soluciones exactas mediante SVD (con y sin regularización).

5. *Evaluación del Error en Test vs λ :* Se exploró el efecto del factor de regularización λ sobre el error cuadrático medio en test (RMSE). Se evaluaron 50 valores de λ entre 10^{-4} y 10^4 en escala logarítmica, calculando:

- El RMSE con regularización para cada λ .
- El RMSE sin regularización como línea base.

4 Resultados

4.1 Optimización con Gradiente Descendente

En esta sección presentamos un análisis detallado de nuestros experimentos con el método de gradiente descendente aplicado a la función de Rosenbrock.

4.1.1 Demostración de la no-convexidad de la función Rosenbrock

El gradiente de la función de Rosenbrock viene dado por:

$$\nabla f(x, y) = \begin{bmatrix} -2(a - x) - 4bx(y - x^2) \\ 2b(y - x^2) \end{bmatrix} \quad (23)$$

Y su matriz Hessiana asociada viene dada por:

$$H(x, y) = \begin{bmatrix} 2 - 4b(y - x^2) + 8bx^2 & -4bx \\ -4bx & 2b \end{bmatrix} \quad (24)$$

La no convexidad de la función se puede verificar analizando el Hessiano. Para que una función sea convexa, su matriz Hessiana debe ser semidefinida positiva para todo punto (x, y) .

$$\begin{vmatrix} 2 - 4b(y - x^2) + 8bx^2 - \lambda & -4bx \\ -4bx & 2b - \lambda \end{vmatrix} = 0 \quad (25)$$

Esta ecuación característica tiene solución:

$$\lambda_{1,2} = \frac{A \pm \sqrt{A^2 - 4B}}{2} \quad (26)$$

donde:

- $A = 2 - 4b(y - x^2) + 8bx^2 + 2b$
- $B = 4b - 8b^2(y - x^2) + 16b^2x^2 - 16b^2x^2$

Para ciertos valores de (x, y) , por ejemplo $(0, 0)$, el autovalor mas chico es negativo, lo que demuestra que la función no es convexa.

4.1.2 Visualización de la función Rosenbrock

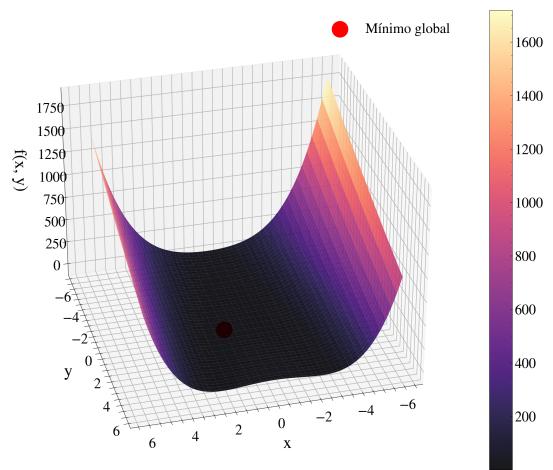


Fig. 1: visualización de la función Rosenbrock

La función de Rosenbrock, representada en la (1), muestra su característica forma de "valle" o "banana". El mínimo global se encuentra en el punto $(1,1)$, marcado con un punto rojo. La forma peculiar de esta función la hace particularmente desafiante para los métodos de optimización, ya que combina regiones de alta curvatura con un valle largo y estrecho donde el gradiente es relativamente pequeño.

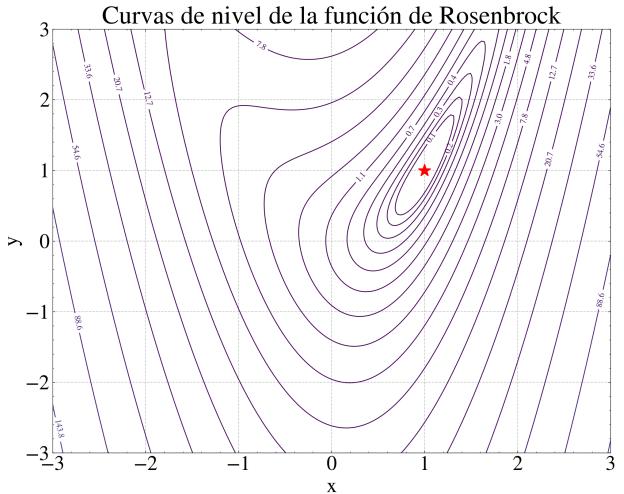


Fig. 2: Curvas de nivel de la función Rosenbrock

4.1.3 Análisis del impacto del Learning Rate (η)

La 2 presenta las curvas de nivel de la función, permitiendo una mejor visualización de la topología del problema. Como se evidencia en 1, el comportamiento del método de gradiente descendente es altamente sensible a la elección de la tasa de aprendizaje η . Para valores muy pequeños ($\eta \approx 10^{-6}$), el método converge de manera estable pero requiere un gran número de iteraciones (>3000). Al aumentar η , observamos una reducción significativa en el número de iteraciones necesarias, alcanzando un mínimo de 703 iteraciones con $\eta = 5.18e-05$.

Sin embargo, tasas de aprendizaje superiores a $3.73e-04$ resultan en overflow del gradiente, confirmando lo establecido en el marco teórico sobre la necesidad de mantener η por debajo de cierto umbral para garantizar la convergencia. Es interesante notar que la distancia al óptimo no decrece monótonamente con η , alcanzando su valor

Learning Rate	Condición de Corte	Distancia al Óptimo
1.00e-06	por step en iter 3217	1.89974600
1.93e-06	por step en iter 5958	1.60941760
3.73e-06	por step en iter 3970	1.50934226
7.20e-06	por step en iter 2517	1.42719101
1.39e-05	por step en iter 1549	1.42311766
2.68e-05	por step en iter 945	1.40939566
5.18e-05	por step en iter 703	1.36595066
1.00e-04	por step en iter 2926	0.96742828
1.93e-04	por step en iter 2776	1.74389240
3.73e-04	por step en iter 3931	0.49268111
7.20e-04	Overflow en gradiente	nan
1.39e-03	Overflow en gradiente	nan
2.68e-03	Overflow en gradiente	nan
5.18e-03	Overflow en gradiente	nan
1.00e-02	Overflow en gradiente	nan

Table 1 Comparación de tasas de aprendizaje para el método de gradiente descendente

mínimo (0.49268111) con $\eta = 3.73e-04$, justo antes de entrar en la región de inestabilidad.

El fenómeno de overflow en el gradiente, observado tanto para tasas de aprendizaje altas ($\eta > 3.73 \times 10^{-4}$) se debe a la estructura particular de la función de Rosenbrock. El término $b(y - x^2)$ con $b = 100$ puede crecer muy rápidamente cuando $|y - x^2|$ es grande, llevando a valores del gradiente que exceden la capacidad de representación numérica de punto flotante. Este comportamiento es especialmente probable en regiones donde x o y toman valores grandes en magnitud, ya que el término cuadrático x^2 amplifica cualquier desviación.

4.1.4 Sensibilidad a condiciones iniciales

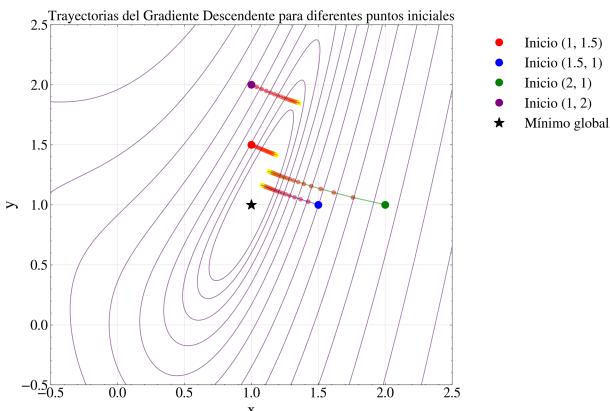


Fig. 3: trayectorias del gradiente descendiendo

La tabla (2) revela la influencia crítica del punto inicial en la convergencia del método. El punto inicial $(1.5, 1)$, más cercano al mínimo global, logra convergencia en solo 5 iteraciones con una distancia final de 0.07907436 . En contraste, puntos más alejados como $(2, 3)$ requieren más de 5000 iteraciones, aunque logran una distancia al óptimo similar (≈ 0.288). El caso extremo $(3, 3)$ resulta en overflow, sugiriendo que existen regiones desde las cuales el método no puede recuperarse.

Punto Inicial	Condición de Corte	Distancia GT
[1, 1.5]	Convergencia por step en iter 1301	0.28894949
(1.5, 1)	Convergencia por step en iter 5	0.07907436
(2, 3)	Convergencia por step en iter 5051	0.28868699
(2, 1)	Convergencia por step en iter 3852	0.28847594
(1, 2)	Convergencia por step en iter 3068	0.28870617
(3, 3)	Error numérico: Overflow en gradiente	inf

Table 2 Comparación del desempeño del método para diferentes puntos iniciales.

4.1.5 Comparación con el método de Newton

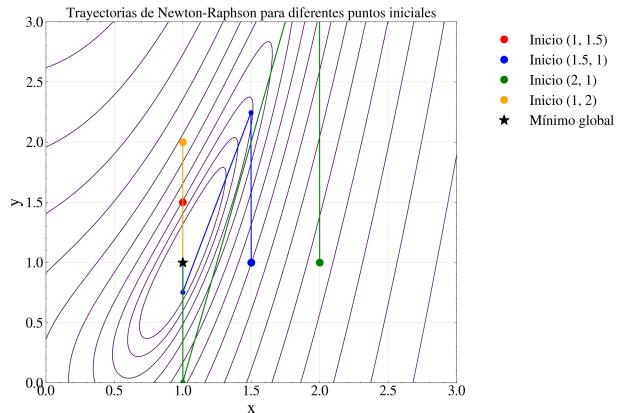


Fig. 4: trayectorias del método de Newton

Las trayectorias visualizadas en la 3 muestran cómo el gradiente descendente sigue un camino serpenteante a través del valle de la función, característico de su naturaleza de primer orden. En contraste, 4 y 3 evidencian la superior eficiencia del método de Newton-Raphson, que logra convergencia en 5 o menos iteraciones para todos los puntos iniciales exitosos, con distancias al óptimo del orden de 10^{-16} .

Esta diferencia en rendimiento confirma la ventaja teórica del método de Newton-Raphson al utilizar información de segundo orden (matriz Hessiana), permitiendo una convergencia cuadrática cerca del óptimo. Sin embargo, es importante notar que este método requiere el cálculo y la inversión del Hessiano en cada iteración, lo que puede ser computacionalmente costoso para problemas de mayor dimensión.

Punto Inicial	Condición de Corte	Distancia al GT
(1, 1.5)	Convergencia gradiente en iter 1	3.1401e-16
(1.5, 1)	Convergencia step en iter 4	6.3863e-12
(2, 1)	Convergencia step en iter 4	0.0000e+00
(1, 2)	Convergencia gradiente en iter 1	0.0000e+00
(3, 3)	Convergencia gradiente en iter 5	2.2204e-16

Table 3 Análisis de convergencia del método de Newton

4.1.6 Deducción del orden de convergencia del método de Newton

El método de Newton para optimización tiene convergencia cuadrática local. A continuación, demostraremos formalmente esta propiedad. Sea x^* el punto óptimo y definamos el error en la iteración n como:

$$e_n = x_n - x^* \quad (27)$$

En el punto óptimo x^* , sabemos que el gradiente es nulo y la matriz Hessiana es definida positiva:

$$\nabla f(x^*) = 0 \quad (28)$$

Realizando una expansión de Taylor del gradiente alrededor de x^* :

$$\nabla f(x_n) = \nabla f(x^*) + H(x^*)(x_n - x^*) + \frac{1}{2} \nabla^2 f(\xi)(x_n - x^*)^2 \quad (29)$$

donde ξ está entre x_n y x^* . Como $\nabla f(x^*) = 0$:

$$\nabla f(x_n) = H(x^*)(x_n - x^*) + \frac{1}{2} \nabla^2 f(\xi)(x_n - x^*)^2 \quad (30)$$

La matriz Hessiana también puede expandirse alrededor de x^* :

$$H(x_n) = H(x^*) + \nabla^3 f(\eta)(x_n - x^*) \quad (31)$$

donde η está entre x_n y x^* . Por la iteración de Newton:

$$x_{n+1} = x_n - H^{-1}(x_n)\nabla f(x_n) \quad (32)$$

Restando x^* de ambos lados:

$$e_{n+1} = x_{n+1} - x^* = e_n - H^{-1}(x_n)\nabla f(x_n) \quad (33)$$

Sustituyendo las expansiones:

$$e_{n+1} = e_n - [H(x^*) + \nabla^3 f(\eta)e_n]^{-1}[H(x^*)e_n + \frac{1}{2}\nabla^3 f(\xi)e_n^2] \quad (34)$$

Usando la fórmula para la inversa de una matriz perturbada:

$$\begin{aligned} [H(x^*) + \nabla^3 f(\eta)e_n]^{-1} &= H^{-1}(x^*) \\ &- H^{-1}(x^*)\nabla^3 f(\eta)e_n H^{-1}(x^*) + O(||e_n||^2) \end{aligned} \quad (35)$$

Sustituyendo y simplificando algebraicamente:

$$e_{n+1} = Ce_n^2 + O(||e_n||^3) \quad (36)$$

donde C es una constante que depende de $H(x^*)$ y $\nabla^3 f$. Por lo tanto, existe una constante K tal que:

$$||e_{n+1}|| \leq K||e_n||^2 \quad (37)$$

cuando x_n está suficientemente cerca de x^* . Esta desigualdad demuestra la convergencia cuadrática local del método de Newton.

Esta tasa de convergencia implica que el número de dígitos correctos aproximadamente se duplica en cada iteración, lo que explica la rápida convergencia del método una vez que se encuentra suficientemente cerca del óptimo. Es importante notar que la convergencia es local, requiriendo un punto inicial suficientemente cercano al óptimo para garantizar este comportamiento.

4.2 Cuadrados mínimos mediante descenso por gradiente

En esta sección se presentan los resultados de la aplicación del método de descenso por gradiente al problema de cuadrados mínimos. En particular, se analiza el impacto de la tasa de aprendizaje, la estratificación de los datos, la convergencia del algoritmo y el impacto de la regularización L2 en la performance del modelo.

4.2.1 Demostración de la convexidad del Error Cuadrático Medio

El Error Cuadrático Medio (ECM) para un modelo lineal está dado por:

$$ECM(w) = \frac{1}{n}\|y - Xw\|^2 \quad (38)$$

donde $X \in \mathbb{R}^{n \times (d+1)}$, $y \in \mathbb{R}^n$ y $w \in \mathbb{R}^{d+1}$.

Demostraremos la convexidad del ECM verificando las condiciones de primer y segundo orden. Si bien una condición es suficiente, verificar ambas nos brinda una comprensión más profunda de la función.

Condición de primer orden: Para dos puntos arbitrarios w_1, w_2 , debemos verificar:

$$ECM(w_2) \geq ECM(w_1) + \nabla ECM(w_1)^T(w_2 - w_1) \quad (39)$$

El gradiente del ECM es:

$$\nabla ECM(w) = \frac{2}{n}X^T(Xw - y) \quad (40)$$

Sustituyendo en la desigualdad y desarrollando:

$$\frac{1}{n}\|y - Xw_2\|_2^2 \geq \frac{1}{n}\|y - Xw_1\|_2^2 + \frac{2}{n}(Xw_1 - y)^T X(w_2 - w_1) \quad (41)$$

Esta desigualdad se cumple debido a la convexidad de la norma euclíadiana al cuadrado.

Condición de segundo orden: La matriz Hessiana del ECM es:

$$H(w) = \nabla^2 ECM(w) = \frac{2}{n}X^T X \quad (42)$$

Para todo vector $v \neq 0$:

$$v^T H(w)v = \frac{2}{n}v^T X^T X v = \frac{2}{n}\|Xv\|_2^2 \geq 0 \quad (43)$$

4.2.2 Análisis de la estratificación

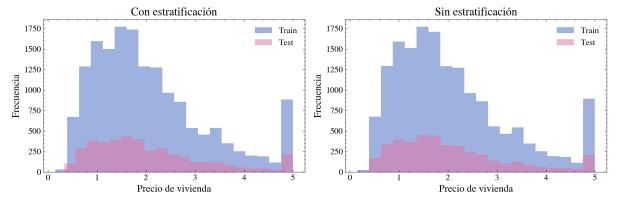


Fig. 5: estratificación de los datos

La Figura 5 presenta una comparación de la distribución de los precios de vivienda entre los conjuntos de entrenamiento y prueba, bajo dos escenarios: con y sin estratificación.

La estratificación es una técnica de muestreo que busca mantener la misma distribución de la variable objetivo en ambos conjuntos. En los histogramas podemos observar que ambos métodos logran distribuciones similares, con una forma aproximadamente log-normal característica de los precios de vivienda.

Comparando los coeficientes obtenidos:

Conjunto	Con estratificación
Train	0.2174, 0.3814, 0.2202, 0.0966, 0.0848
Test	0.2734, 0.3587, 0.1964, 0.0922, 0.0789

Table 4 Comparación de coeficientes con estratificación

Conjunto	Sin estratificación
Train	0.2171, 0.3803, 0.2212, 0.0974, 0.0838
Test	0.2189, 0.3849, 0.2216, 0.0925, 0.0818

Table 5 Comparación de coeficientes sin estratificación

La estratificación produce diferencias más marcadas entre los coeficientes de entrenamiento y prueba, particularmente en el primer coeficiente (0.2174 vs 0.2734). En contraste, sin estratificación los coeficientes mantienen valores más consistentes entre ambos conjuntos, con diferencias máximas del orden de 0.004.

Este resultado sugiere que, para este conjunto de datos particular, la estratificación podría no ser necesaria dado que el muestreo aleatorio simple ya logra preservar adecuadamente la distribución de los precios entre los conjuntos de entrenamiento y prueba.

4.2.3 Análisis del modelo de regresión lineal

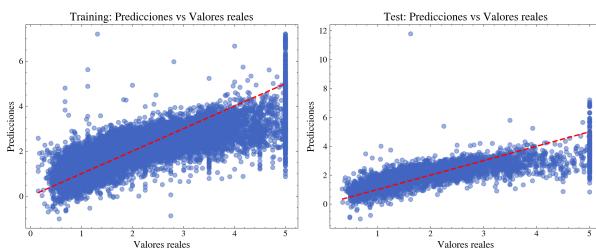


Fig. 6: predicciones del modelo lineal

La Figura 6 muestra la relación entre las predicciones del modelo y los valores reales, tanto para el conjunto de entrenamiento como de prueba. El análisis revela varios aspectos importantes:

Ajuste del modelo:: La línea roja punteada representa la relación ideal donde predicciones = valores reales. Observamos que:

El modelo tiende a sobreestimar los valores bajos (puntos por encima de la línea) y subestimar los valores altos (puntos por debajo). Además, la dispersión de los puntos aumenta con el valor real, indicando heterocedasticidad en las predicciones, es decir, la variabilidad de los errores no es constante a lo largo de todos los niveles de la variable dependiente. El comportamiento es similar en ambos conjuntos, lo que sugiere que el modelo generaliza razonablemente bien.

Este análisis sugiere que el modelo captura las tendencias generales en los datos, pero podría beneficiarse de técnicas adicionales para manejar la heterocedasticidad observada y mejorar la precisión en los extremos de la distribución.

4.2.4 Análisis de convergencia del gradiente descendente

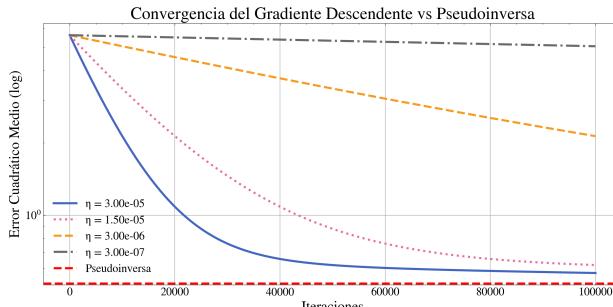


Fig. 7: Comparación convergencia del gradiente descendente y la pseudoinversa

La Figura 7 muestra un análisis comparativo entre el método de gradiente descendente con diferentes tasas de aprendizaje y la solución mediante pseudoinversa. A partir del gráfico observamos que la pseudoinversa obtiene el error mínimo en una única iteración, mientras que el gradiente descendente requiere numerosas iteraciones para aproximarse a este valor, siendo altamente sensible a la elección de la tasa de aprendizaje.

La elección de la tasa de aprendizaje $\eta = 1/\sigma_1$ tiene un fundamento teórico importante. La matriz $X^T X$ que aparece en el gradiente del ECM tiene como autovalores los cuadrados de los valores singulares de X . El valor singular más grande σ_1 determina la dirección de máxima curvatura de la función objetivo. Por lo tanto, $1/\sigma_1$ asegura que el paso del gradiente sea inversamente proporcional a esta curvatura máxima, evitando oscilaciones y divergencia.

Método	MSE Train	MSE Test	Iteraciones
Pseudoinversa	0.519372	0.551437	1
GD ($\eta=3.00\text{e-}05$)	0.574015	0.588683	100000
GD ($\eta=1.50\text{e-}05$)	0.619370	0.635283	100000
GD ($\eta=3.00\text{e-}06$)	2.134062	2.142178	100000
GD ($\eta=3.00\text{e-}07$)	5.053412	5.018803	100000

Table 6 Comparación de métodos de optimización

Los resultados experimentales confirman esta intuición teórica: con $\eta = 3 \times 10^{-5}$ (cerca a $1/\sigma_1$) se obtiene la convergencia más rápida entre todas las tasas probadas, alcanzando un MSE de 0.574015 en entrenamiento y 0.588683 en test. Tasas de aprendizaje menores resultan en una convergencia excesivamente lenta, como se evidencia con $\eta = 3 \times 10^{-7}$ donde el error permanece alto incluso después de 100000 iteraciones.

Es destacable que la pseudoinversa no solo converge en una única iteración, sino que también alcanza el menor error tanto en entrenamiento (0.519372) como en test (0.551437). Este resultado es esperable ya que proporciona la solución exacta al problema de mínimos cuadrados, mientras que el gradiente descendente solo puede aproximarla iterativamente.

4.2.5 Comparación entre Pseudoinversa y Gradiente Descendente

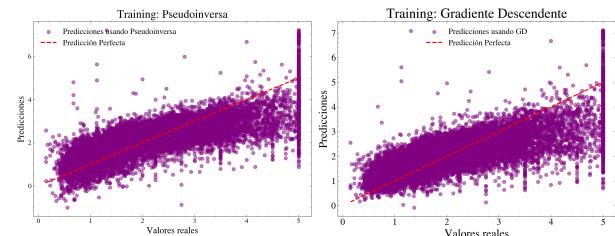


Fig. 8: Comparación de predicciones en el conjunto de entrenamiento

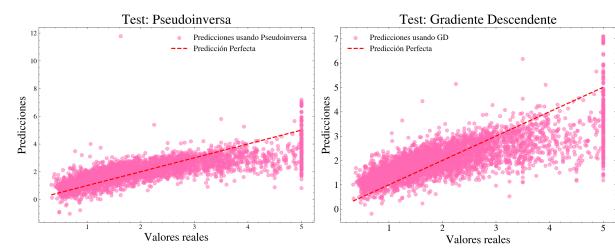


Fig. 9: Comparación de predicciones en el conjunto de prueba

Las Figuras 8 y 9 presentan una comparación directa entre las predicciones obtenidas mediante la pseudoinversa y el método de gradiente descendente, tanto en el conjunto de entrenamiento como en el de prueba.

En el conjunto de entrenamiento, ambos métodos muestran patrones similares de dispersión alrededor de la línea de predicción perfecta (línea roja punteada). Sin embargo, se observan algunas diferencias significativas. La pseudoinversa muestra una dispersión más uniforme y compacta alrededor de la línea ideal, especialmente en la región central de los valores (entre 2 y 4). Por su parte, el gradiente descendente exhibe una dispersión ligeramente mayor, particularmente notable en los valores extremos.

Esta diferencia se hace más evidente en el conjunto de prueba. Las predicciones de la pseudoinversa mantienen una dispersión más controlada, mientras que el gradiente descendente tiende a mostrar mayor variabilidad, especialmente para valores altos de precio (>4). Además, se observa que ambos métodos tienden a subestimar los

valores altos y sobreestimar los bajos, un comportamiento típico en problemas de regresión lineal.

4.2.6 Análisis del método de momentum

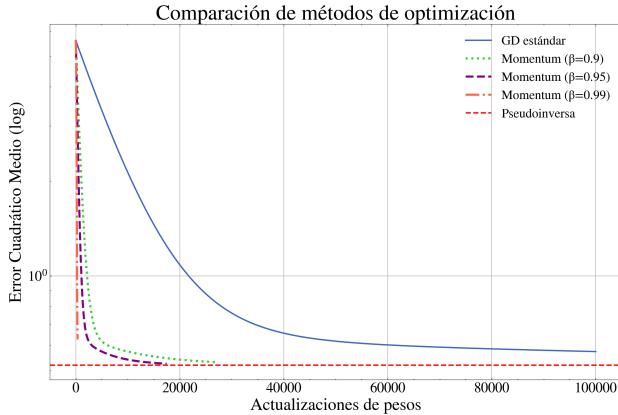


Fig. 10: Comparación de métodos de optimización con momentum

La Figura 10 presenta una comparación entre el gradiente descendente estándar, tres variantes de momentum con diferentes valores de β , y la solución por pseudo inversa.

Método	MSE Train	MSE Test	Iteraciones
Pseudo inversa	0.519372	0.551437	1
GD estándar	0.519372	0.551437	100000
Momentum ($\beta=0.9$)	0.530718	0.553414	27331
Momentum ($\beta=0.95$)	0.525219	0.551002	17476
Momentum ($\beta=0.99$)	0.628089	0.644228	397

Table 7 Comparación del desempeño de los métodos

Dado que el ECM es una función convexa, como se demostró anteriormente, no existe el riesgo de quedar atrapado en mínimos locales. Por lo tanto, el uso de SGD como estrategia para escapar de estos no tiene sentido en este contexto, aunque podría ser beneficioso para reducir el costo computacional en conjuntos de datos muy grandes. Por lo que no fue considerado en este análisis.

El método de momentum demuestra ser notablemente superior al gradiente descendente estándar en términos de velocidad de convergencia. Con $\beta = 0.95$, momentum alcanza un error similar al GD estándar en apenas 17476 iteraciones, una reducción de aproximadamente 82%. Esta aceleración se debe a que momentum acumula información de gradientes pasados, permitiendo avanzar más rápidamente en direcciones consistentes y reduciendo oscilaciones en regiones con alta curvatura.

Sin embargo, observamos que con $\beta = 0.99$, el método se detiene prematuramente después de solo 397 iteraciones, con un error significativamente mayor. Esta interrupción temprana probablemente se debe a problemas de overflow numérico: un β tan cercano a 1 implica que el término de momento crece muy rápidamente, pudiendo exceder los límites de representación numérica. Además, valores altos de β pueden resultar en pasos demasiado grandes que dificultan alcanzar la tolerancia establecida de 10^{-6} en la norma del gradiente.

La elección de $\beta = 0.95$ parece ser óptima para este problema, logrando un balance entre velocidad de convergencia y estabilidad numérica. Este valor permite una acumulación suficiente de momento para acelerar significativamente la convergencia, mientras mantiene el algoritmo numéricamente estable.

4.2.7 Análisis del efecto de la regularización

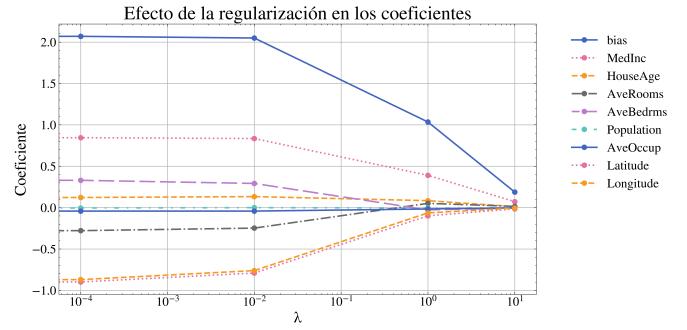


Fig. 11: Efecto de la regularización en los coeficientes del modelo

La Figura 11 ilustra el comportamiento de los coeficientes del modelo bajo diferentes valores del parámetro de regularización λ . Para valores pequeños ($\lambda < 10^{-2}$), los coeficientes mantienen sus magnitudes iniciales, indicando poco impacto de la regularización. Se observan tres comportamientos distintivos: los coeficientes de 'AveOccup' y 'Latitude' muestran las reducciones más pronunciadas, partiendo de valores altos (2.0 y 0.8 respectivamente) y convergiendo a cero cuando $\lambda > 1$. El coeficiente de 'HouseAge' exhibe un comportamiento único, convergiendo a cero desde valores negativos. El resto de los coeficientes muestra una tendencia más gradual hacia cero. El efecto de "shrinkage" característico de la regularización L2 se hace evidente para $\lambda > 1$, donde todos los coeficientes se aproximan a cero, resultando en un modelo más simple pero potencialmente menos expresivo. Esta reducción en la magnitud ayuda a prevenir el sobreajuste, aunque podría incrementar el sesgo del modelo.

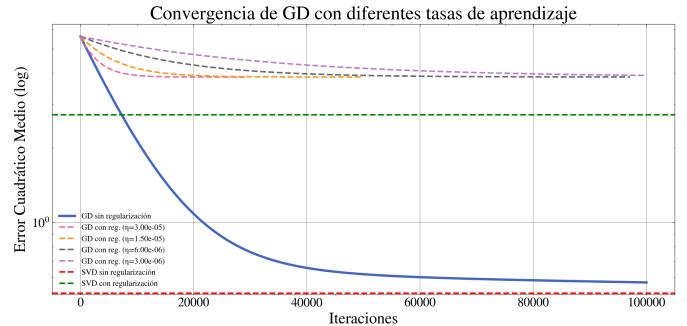


Fig. 12: Convergencia del gradiente descendente con regularización L2

La Figura 12 muestra el comportamiento del gradiente descendente aplicado al problema regularizado con diferentes valores del parámetro λ . Al comparar con los resultados sin regularización, observamos varios aspectos interesantes:

η	MSE final	Iteraciones	Mensaje
3.00e-05	3.852473	29018	Convergencia en iter 29017
1.50e-05	3.852772	49738	Convergencia en iter 49737
6.00e-06	3.854853	97083	Convergencia en iter 97082
3.00e-06	3.911792	100000	Máx iteraciones alcanzado

Table 8 Resultados con regularización L2

La regularización L2 modifica significativamente la superficie de error. Mientras que el gradiente descendente sin regularización alcanzaba un MSE de aproximadamente 0.57 en entrenamiento, con

regularización el error aumenta a 3.85. Este incremento es esperable ya que la regularización introduce un término de penalización adicional en la función objetivo.

La tasa de convergencia también se ve afectada. El modelo regularizado converge más rápidamente con la tasa de aprendizaje óptima ($\eta = 3.00e-5$), requiriendo solo 29018 iteraciones comparado con las 100000 del modelo sin regularizar. Esto se debe a que el término de regularización $\lambda\|w\|_2^2$ mejora el condicionamiento del problema, haciendo la superficie de error más "suave".

Un resultado notable es que valores más pequeños de η resultan en convergencia más lenta o incluso falta de convergencia dentro del límite de iteraciones, como se observa para $\eta = 3.00e-6$. Esto contrasta con el caso no regularizado donde tasas de aprendizaje más pequeñas eventualmente convergían, aunque lentamente.

La línea roja punteada en el gráfico representa la solución exacta mediante pseudoinversa regularizada $(X^T X + \lambda I)^{-1} X^T y$. Observamos que el gradiente descendente se aproxima a este valor óptimo, confirmando la correcta implementación de la regularización.

Este comportamiento ilustra el propósito principal de la regularización L2: si bien aumenta el error de entrenamiento, mejora la estabilidad numérica del problema y potencialmente la generalización del modelo, como sugiere la teoría estadística del ridge regression.

4.2.8 Análisis del factor de regularización óptimo

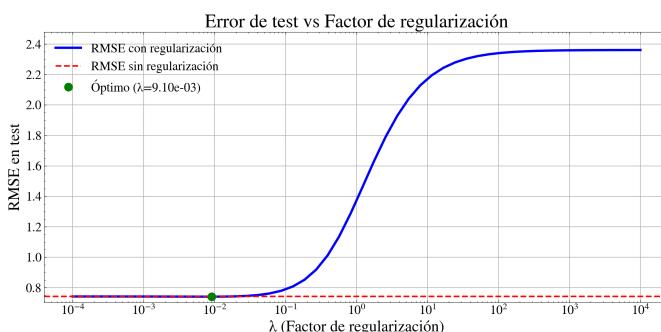


Fig. 13: Evolución del error de test en función de λ

La Figura 13 muestra la evolución del error de test (RMSE) en función del parámetro de regularización λ en escala logarítmica.

Modelo	RMSE Test
Sin regularización	0.7426
Con regularización óptima ($\lambda = 9.10e-3$)	0.7415

Table 9 Comparación de RMSE en test

El gráfico revela un comportamiento característico de la regularización L2: para valores muy pequeños de $\lambda (< 10^{-3})$, el error se mantiene prácticamente igual al modelo sin regularizar, mientras que para valores grandes ($> 10^1$), el error crece significativamente hasta saturar en aproximadamente 2.4.

El punto óptimo se encuentra en $\lambda = 9.10e-3$, donde se alcanza un RMSE de 0.7415, ligeramente mejor que el modelo sin regularizar (0.7426). Esta mejora marginal sugiere que el problema original no sufría de sobreajuste significativo, lo que explica por qué la regularización no produce una mejora sustancial en la generalización.

Es notable la existencia de una región relativamente amplia ($10^{-4} < \lambda < 10^{-1}$) donde el error se mantiene estable y cercano al óptimo, lo que indica cierta robustez del modelo frente a la elección de λ en este rango. Sin embargo, más allá de $\lambda > 1$, el error aumenta dramáticamente debido a un subajuste excesivo causado por la fuerte penalización de los coeficientes.

5 Conclusiones

En este trabajo estudiamos y comparamos diferentes métodos numéricos de optimización aplicados a dos tipos de problemas: uno no convexo (función de Rosenbrock) y otro convexo (regresión lineal con cuadrados mínimos).

Para la función de Rosenbrock, los experimentos revelaron la importancia crítica de la elección de la tasa de aprendizaje y el punto inicial en el método de gradiente descendente. Con $\eta = 3.73e-4$ se obtuvo el mejor compromiso entre velocidad de convergencia y estabilidad, aunque tasas mayores resultaron en overflow numérico. El método de Newton-Raphson demostró convergencia cuadrática, alcanzando precisiones del orden de 10^{-16} en solo 5 iteraciones cuando el punto inicial estaba suficientemente cerca del óptimo.

En el problema de regresión lineal, la comparación entre métodos reveló resultados significativos. La pseudoinversa proporcionó la solución exacta en una única iteración, mientras que el gradiente descendente requirió 100000 iteraciones para aproximarse al mismo error. La adición de momentum mejoró notablemente la convergencia, reduciendo el número de iteraciones en un 82% con $\beta = 0.95$. Sin embargo, valores de β cercanos a 1 resultaron en inestabilidad numérica, evidenciando la importancia del balance entre aceleración y estabilidad.

La regularización L2 tuvo un impacto casi nulo en el rendimiento del modelo. Con $\lambda = 9.10e-3$ se logró una mínima mejora en el error de test (RMSE 0.7415 vs 0.7426), sugiriendo que el problema original no sufría de sobreajuste significativo. El análisis de los coeficientes bajo diferentes valores de λ mostró el efecto de "shrinkage" característico de ridge regression, con una transición suave desde el modelo sin regularizar hasta el sobreajuste extremo para $\lambda > 1$.

Estos resultados proporcionan insights valiosos sobre el comportamiento de diferentes métodos de optimización y la importancia de una cuidadosa selección de hiperparámetros. La elección del método óptimo depende no solo de la naturaleza del problema (convexo vs no convexo) sino también de consideraciones prácticas como el costo computacional y la estabilidad numérica requerida.