Métodos del Gradiente Estocástico

UNIVERSIDAD DEL VALLE DE GUATEMALA

Wilfredo Gallegos - 20399
Javier Aguilar - 20611

Tabla de Contenido

Objetivo General

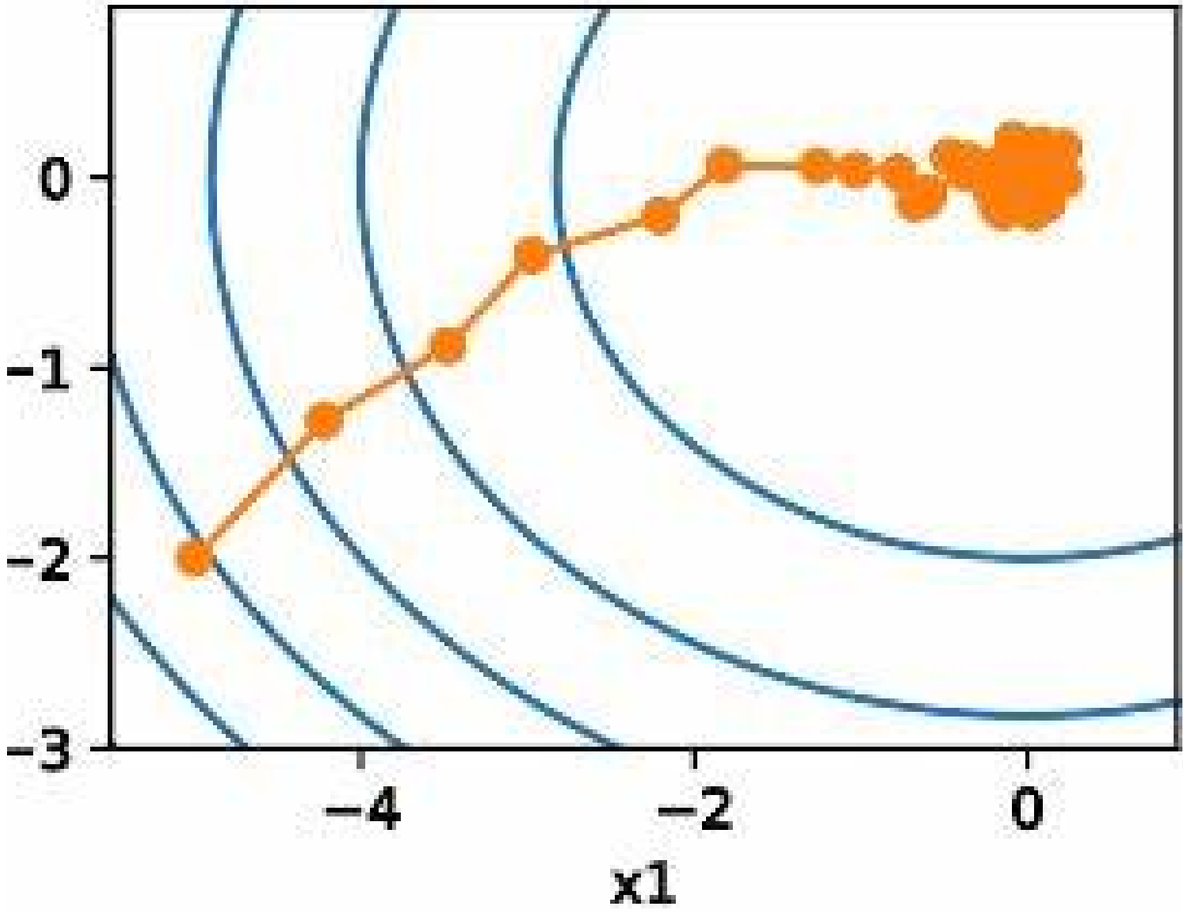
Proporcionar una comprensión integral de los métodos de gradiente estocástico, específicamente momentum, adadelta y adagrad

Objetivos secundarios

1. Comparar en detalle los diferentes métodos (momentum, adadelta, adagrad) en términos de rendimiento, velocidad de convergencia y uso de recursos computacionales.
2. Discutir los desafíos asociados con estos métodos, como el ajuste de parámetros, y cómo estos se pueden superar.

¿Qué es el Gradiente Estocástico?

El Descenso de Gradiente Estocástico (DGE) es un método eficiente para optimizar funciones diferenciables. Reemplaza el gradiente real por una estimación, lo que reduce la carga computacional y permite iteraciones más rápidas en problemas de alta dimensión. Aunque requiere hiperparámetros y es sensible a la escala de las características. El DGE es popular en aprendizaje automático, incluyendo las redes neuronales, debido a su interpretación intuitiva.



Método del Momentum

El método de descenso de gradiente estocástico con Momentum, también conocido como el método de la bola pesada, es una técnica de optimización que se utiliza para acelerar y mejorar la convergencia en el aprendizaje de máquinas. Inspirado en principios físicos, este método considera el vector de peso como una partícula que viaja a través del espacio de parámetros, donde el gradiente de la pérdida actúa como una “fuerza” que acelera la partícula.

En cada iteración, Momentum combina linealmente el gradiente actual y la actualización anterior para determinar la próxima actualización. Esto permite que el algoritmo siga viajando en la misma dirección, evitando oscilaciones y permitiendo una exploración más eficiente del espacio de parámetros.

Además, el método de Momentum es útil para explorar ceros o mínimos de la función de pérdida. Dependiendo del momentum inicial, el algoritmo oscila en todo el espacio de parámetros para encontrar diferentes ceros, lo que puede ayudar a evitar mínimos locales.

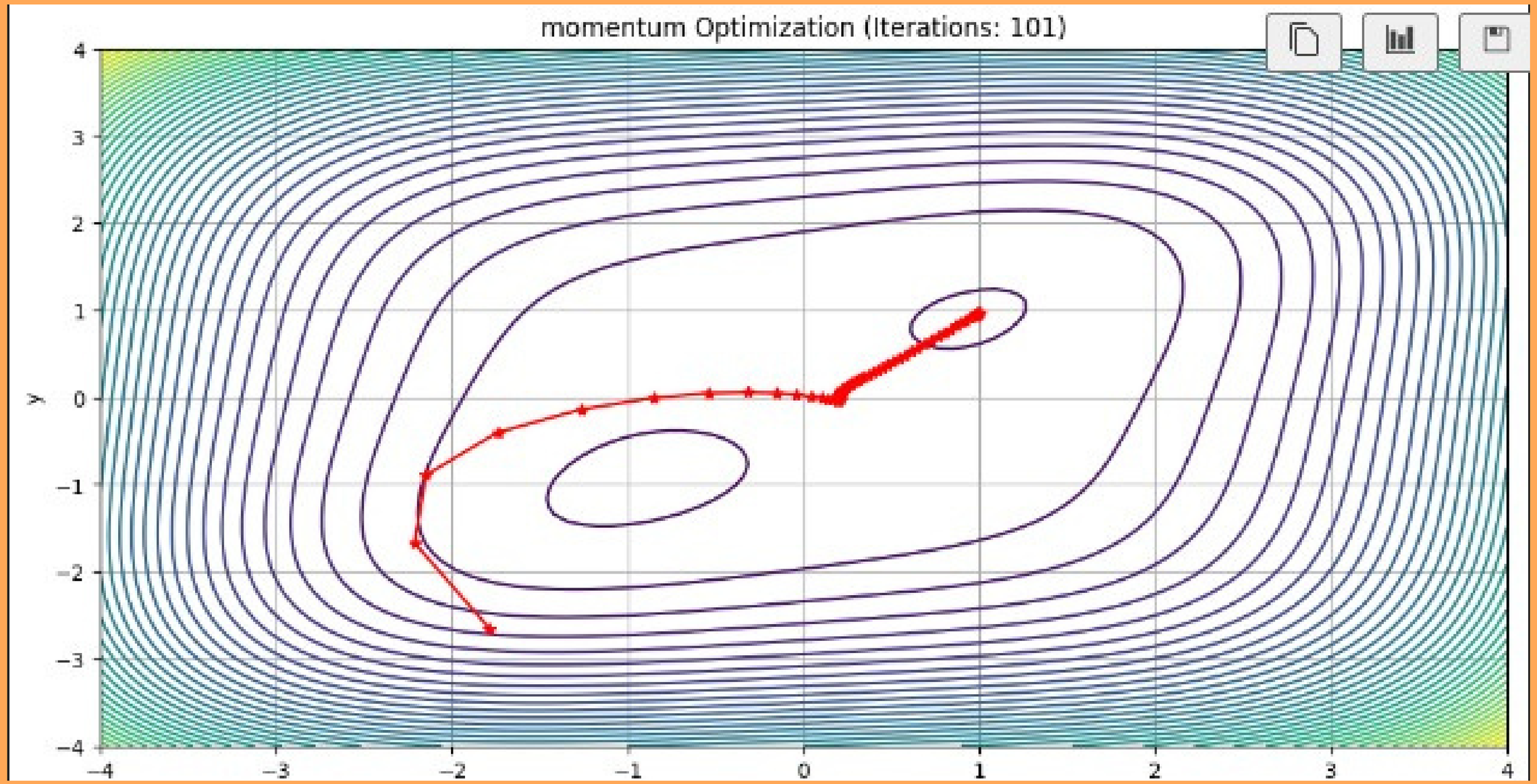
Ecuaciones del método

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

Descripción de las variables

- gamma: Momentum
- n: Tasa de aprendizaje
- J∇(θ): Gradiente
- θ: Vector de variables {x_i}


momentum Optimization (Iterations: 101)
y
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

Método Adadelta

El método Adadelta es una extensión del método de descenso de gradiente estocástico que ajusta la tasa de aprendizaje en función de una ventana móvil de actualizaciones de tamaño acumulado. A diferencia de otros métodos, Adadelta no requiere una tasa de aprendizaje predeterminada, ya que se adapta con el tiempo en función de las actualizaciones del propio algoritmo, permitiendo una convergencia más rápida y una tasa de aprendizaje adaptativa.

Este algoritmo continúa aprendiendo incluso cuando han pasado muchos pasos de actualización, lo que le permite ser robusto a la elección inicial de la tasa de aprendizaje y puede conducir a una convergencia más rápida y estable. Además, busca una buena combinación de los parámetros epsilon y rho para converger de manera rápida y eficiente. Sin embargo, si estos parámetros son demasiado grandes o demasiado pequeños, el algoritmo puede tardar mucho en converger o puede oscilar mucho dentro del espacio del mapa.

Ecuaciones del método

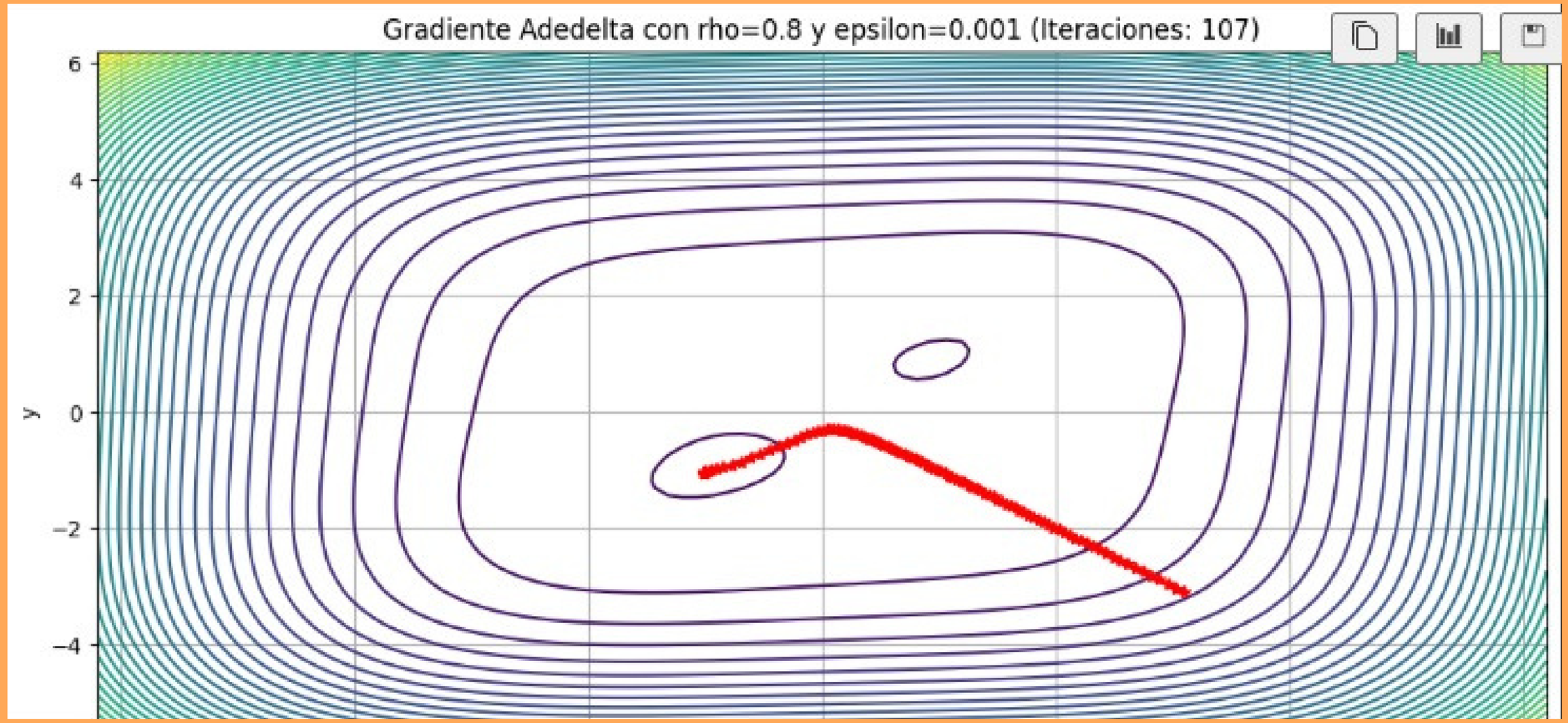
$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2.$$

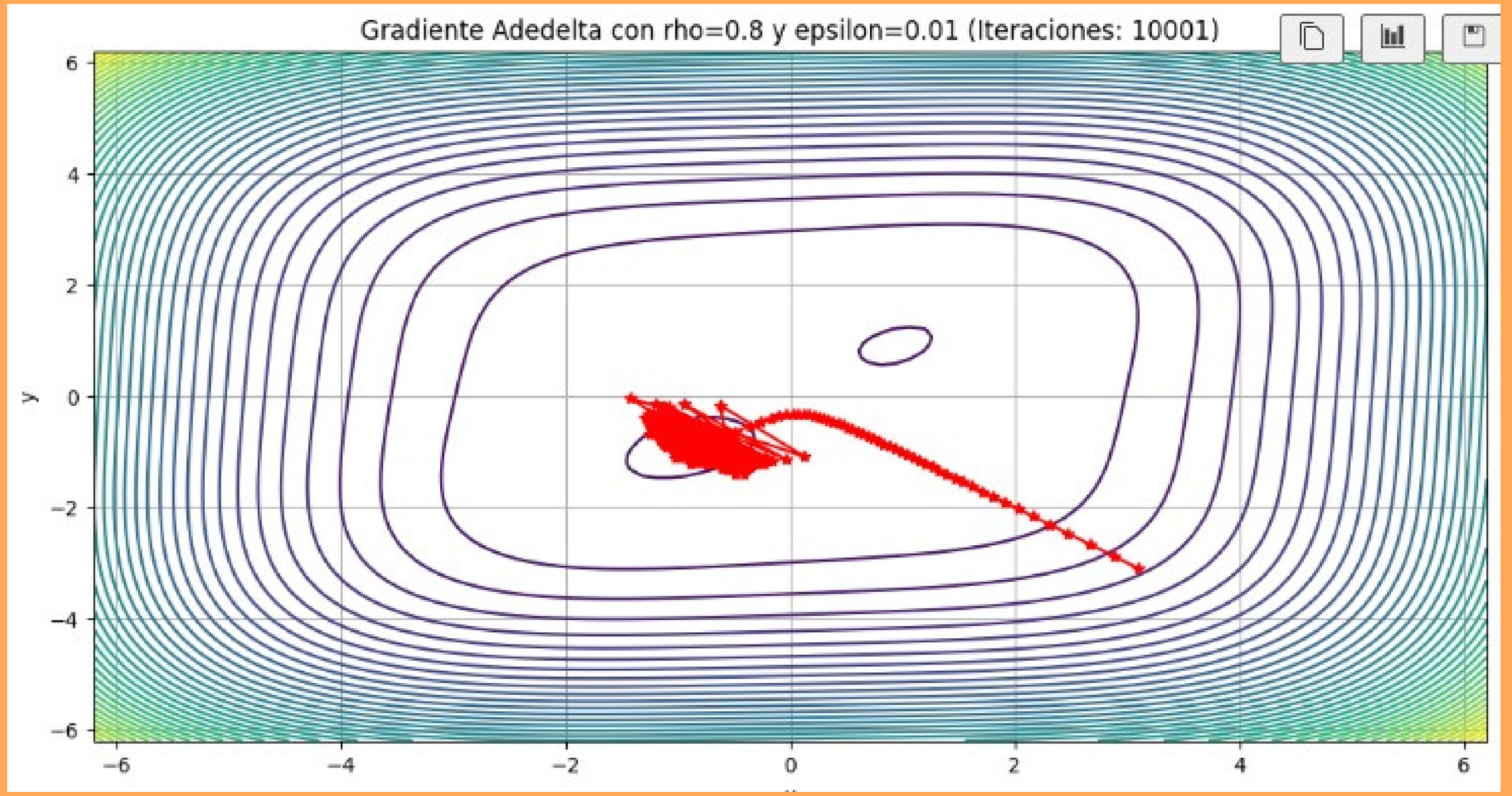
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t.$$

Descripción de las variables

- g_t: es el gradiente en el tiempo t.
- Δθ_t: es el cambio en los parámetros en el tiempo t.
- θ_t es el vector de parámetros en el tiempo t.
- E[g^2]_t: es la acumulación del cuadrado del gradiente en el tiempo t.
- E[Δθ^2]_t: es la acumulación del cuadrado del cambio de parámetros en el tiempo t.
- Gamma: es un factor de decaimiento que controla la importancia relativa de las nuevas y antiguas informaciones en las acumulaciones.


Gradiente Adedelta con rho=0.8 y epsilon=0.001 (Iteraciones: 107)
6
4
2
0
−2
−4
y


Gradiente Adedelta con rho=0.8 y epsilon=0.01 (Iteraciones: 10001)
y
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

Método Adagrad

El método AdaGrad, o algoritmo de gradiente adaptativo, es una variante del método de descenso de gradiente estocástico que ajusta la tasa de aprendizaje para cada parámetro individualmente, basándose en la acumulación de los gradientes de iteraciones pasadas. Esta característica permite que AdaGrad sea más robusto frente a la elección inicial de la tasa de aprendizaje, facilitando una convergencia más rápida y estable.

Este algoritmo aumenta la tasa de aprendizaje para los parámetros más dispersos y la disminuye para los menos dispersos, lo que suele mejorar el rendimiento de la convergencia en entornos donde los datos son dispersos y los parámetros dispersos son más informativos. Esto es especialmente útil en aplicaciones como el procesamiento del lenguaje natural y el reconocimiento de imágenes.

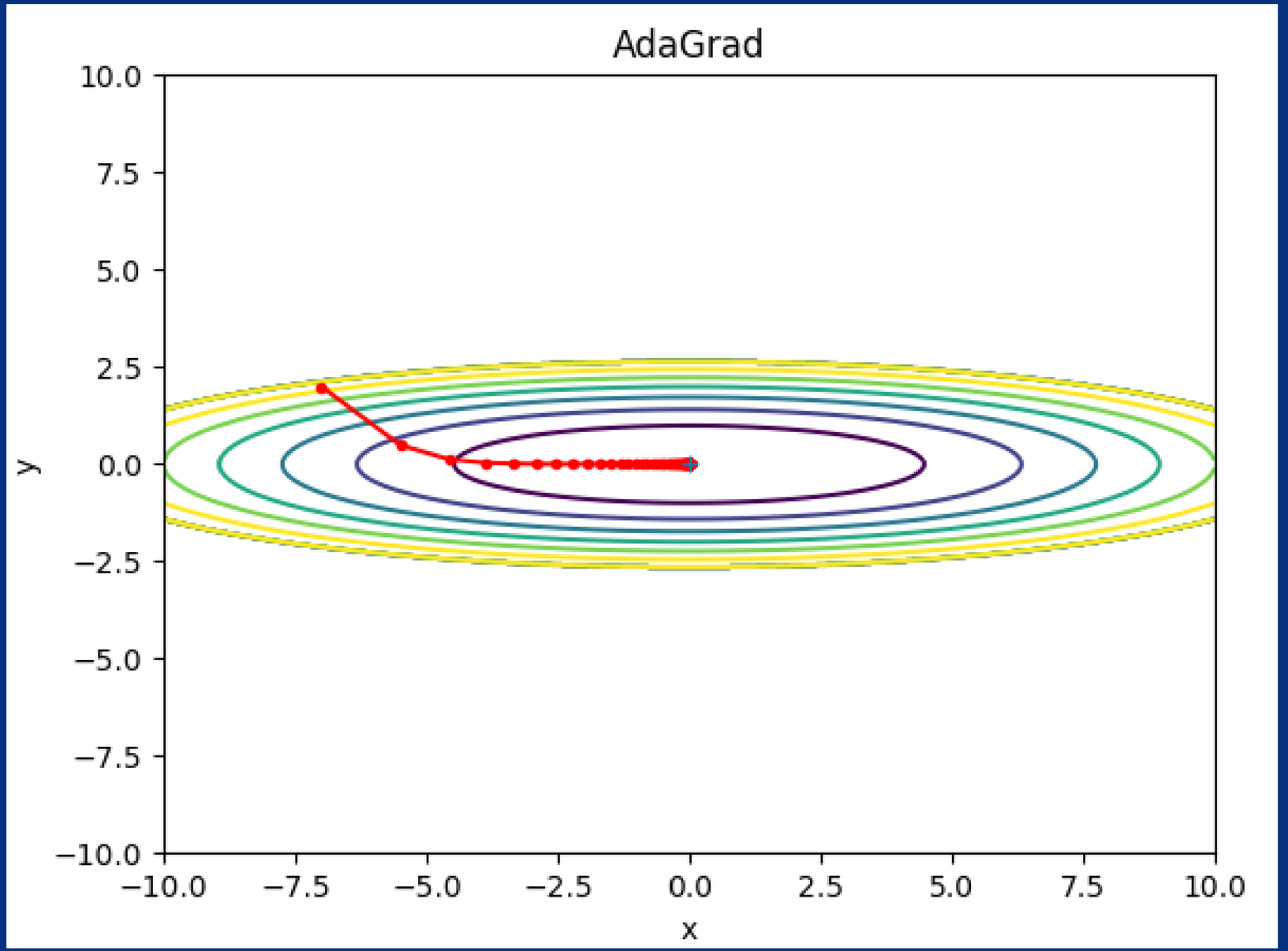
Ecuaciones del método

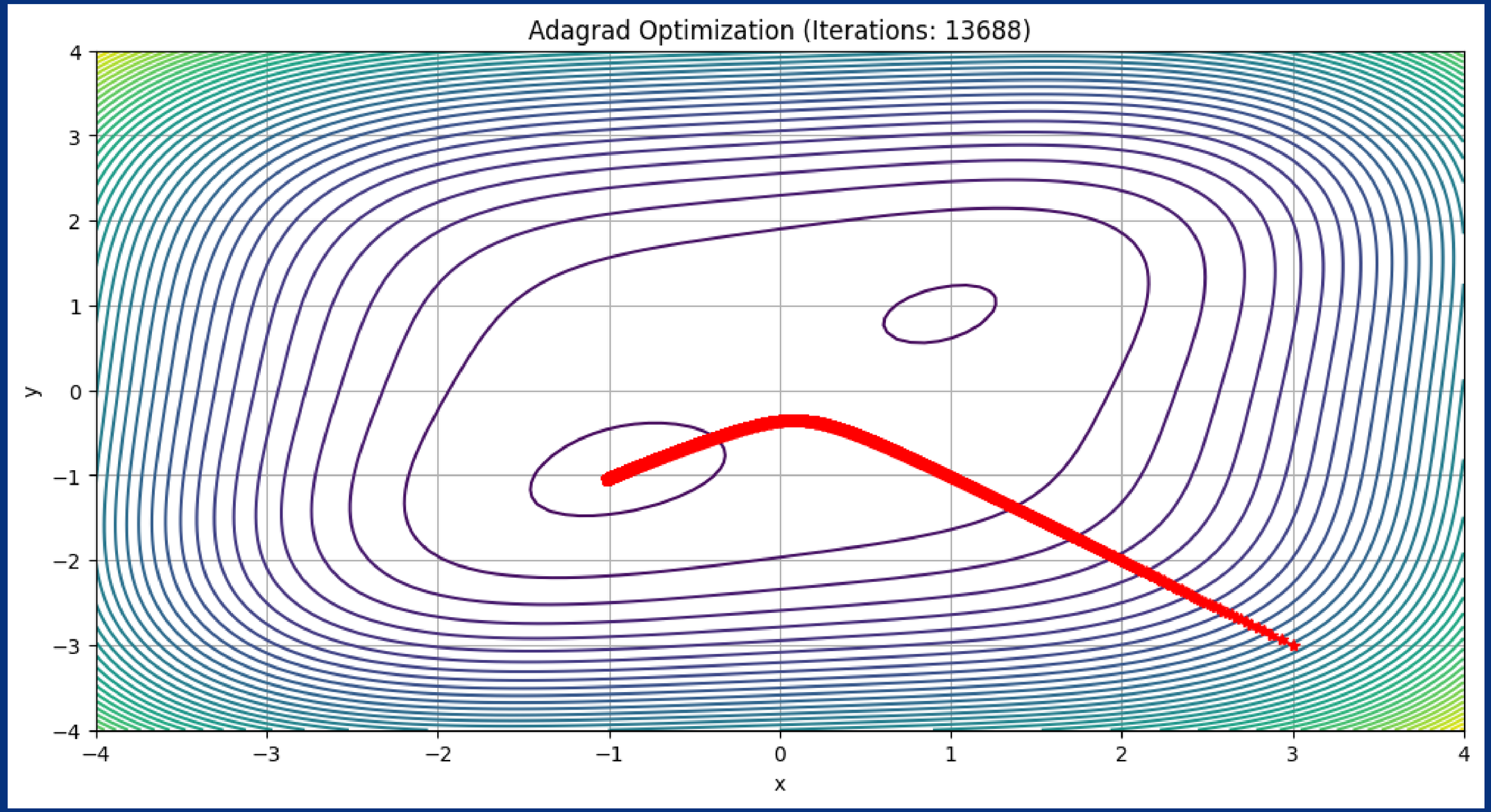
$$g_{t,i} = \nabla_\theta J(\theta_{t,i}).$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t.$$

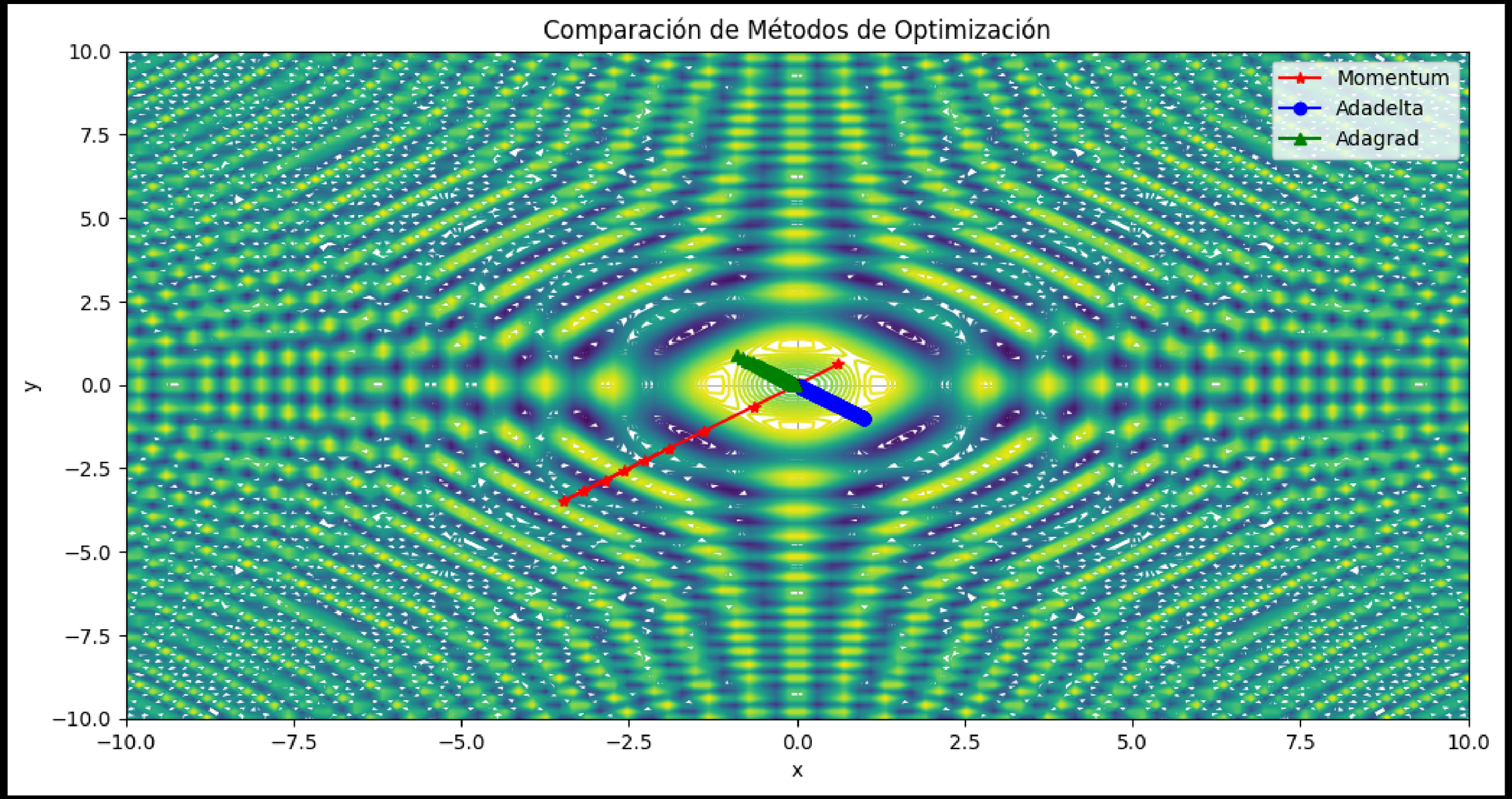
Descripción de las variables

- g_t: es el gradiente en el tiempo t para i variables.
- n: Tasa de aprendizaje
- ∇J(θ): Gradiente
- θ_t: Vector de variables {x_i} en el tiempo t
- G_t: Matriz diagonal con gradiente de los parametros $\underline{\theta}$
- o: producto matriz vector


AdaGrad
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
y
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x


Adagrad Optimization (Iterations: 13688)
4
3
2
1
0
−1
−2
−3
−4
y
−4
−3
−2
−1
0
1
2
3
4
x

Comparación de Métodos


Comparación de Métodos de Optimización
Momentum
Adadelta
Adagrad
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
y
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x

Conclusiones

- Adedelta optimiza los parametros de adagrad, pero depende mas del learning rate inicial
- Al aumentar el Rho y disminuir el Learning_rate hace que Adadelta y Adagrad tarden en converger
- Un Rho más alto en Adagrad conviene para converger más rápido.

Referencias

- Descenso de gradiente estocástico - Wikiwand. (n.d.). Wikiwand. Retrieved November 22, 2023 from https://www.wikiwand.com/es/Descenso_de_gradiente_estocástic
- Fernandez, R. (2018). Descenso de Gradientes Estocástico "SGD" - ▷ Cursos de Programación de 0 a Experto © Garantizados. ▷ Cursos de Programación de 0 A Experto © Garantizados. Retrieved November 22, 2023 from https://unipython.com/descenso-gradientes-estocastico-sgd/
- Descenso de gradiente estocástico: ¡claramente explicado! (2019). ICHI.PRO. https://ichi.pro/es/descenso-de-gradiente-estocastico-claramente-explicado-92162373999921.
- Ruder, S. (2016). Una descripción general de los algoritmos de optimización de descenso de gradiente. *Ruder.io*. https://www.ruder.io/optimizing-gradient-descent/.

¡Muchas Gracias por su atención!

¿PREGUNTAS?

gal20399@uvg.edu.gt

agu20611@uvg.edu.gt

UVG

UNIVERSIDAD DEL VALLE DE GUATEMALA