

Proyecto: Denoising de Imágenes con Técnicas de Optimización

Optimización Matemática

Emilio Frausto Ortiz

Descripción del Proyecto:

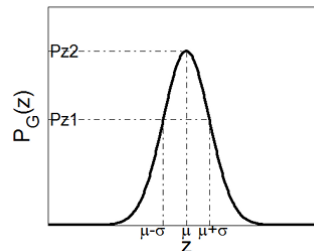
En este proyecto se implementará un algoritmo de Denoising de imágenes basado en la minimización de una función de energía con regularización de norma l^2 . El objetivo es recuperar una imagen limpia a partir de una versión ruidosa, utilizando distintos métodos de optimización para resolver el problema de minimización.

Método Denoising

Nos referimos Denoising al proceso de reducción de ruido en las imágenes. Cabe mencionar que no es posible eliminar el ruido por completo, durante este reporte se explicara por qué. Hay diferentes tipo de ruido, pero en este caso vamos a aplicarle a la imagen un ruido gaussiano.

El ruido gaussiano, es un ruido estadístico que tiene distribución gaussiana. Este tipo de ruido se caracteriza por su curva en forma de campana, que es simétrica al alrededor del valor medio. En imágenes el ruido gaussiano afectara aleatoriamente los pixeles de una imagen para distribuir sus valores de forma uniforme. Su función de densidad de probabilidad viene dada en términos de la media y la varianza de una variable aleatoria en x .

$$P_q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$



Este tipo de ruido también se caracteriza por lo difícil de eliminar por completo. [1][4]

Así se verían las imágenes con diferentes varianzas de ruido gaussiano:

Varianza = 0.9



Varianza = 0.45



Varianza = 0.1



Para tratar de reducir el ruido de las imágenes utilizaremos la siguiente función objetivo:

$$J(u) = (1/2) \|u - f\|^2 + (\lambda/2) \|\nabla u\|^2$$

Donde la imagen ruidosa es f , u es la imagen limpia a estimar (desconocida), ∇u es el gradiente de la imagen, y λ es un parámetro de regularización.

Para minimizar la función objetivo utilizaremos diferentes tipos de Descensos ingresando ∇u . [5]

Descensos Utilizados

-Descenso del Gradiente Simple, caracterizado por ser simple y fácil de implementar, pero lento para grandes conjuntos de datos.

$$X_{k+1} = X_k - \alpha \nabla f(X_k)$$

-Descenso del Gradiente con momentum, caracterizado por esquivar mínimos locales, aunque puede fallar si se ajusta mal.

$$V_{k+1} = \eta V_k + \alpha \nabla f(X_k)$$

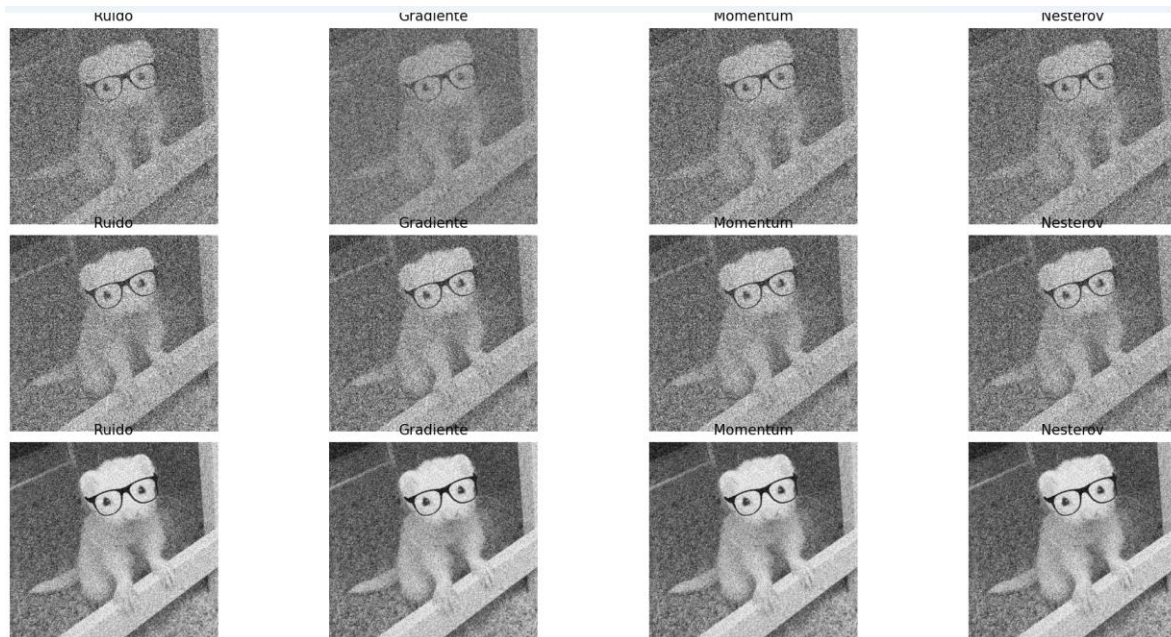
$$X_{k+1} = X_k - V_{k+1}$$

-Descenso del gradiente con aceleración de Nesterov, caracterizado por seguir el gradiente negativo, pero puede rebotar si la función devuelve gradientes ruidosos. [3]

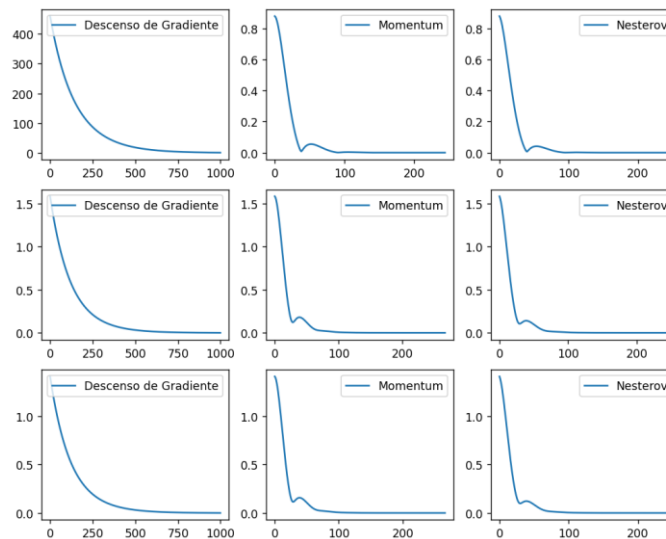
$$V_k = X_k + \beta_k(X_k - X_{k-1})$$

$$X_{k-1} = V_k - \alpha \nabla f(V_k)$$

Estos son los resultados de los descensos aplicados con un lr=0.006, lam = 0.1, eta = 0.9, en un máximo de 1000 iteraciones y un error de 1e-6



Reducción del Error



	Ruido	Iteraciones	Tiempo (s)	Error
Gradiente				
Gradiente Simple	0.9	1000	13.43	0.884959
Momentum	0.9	244	4.34	9.064657e-07
Nesterov	0.9	247	4.56	9.263356e-07
Gradiente Simple	0.45	1000	14.29	0.001099
Momentum	0.45	267	5.46	9.963174e-07
Nesterov	0.45	249	4.84	9.780141e-07

Gradiente Simple	0.1	1000	14.68	0.001167
Momentum	0.1	266	5.46	9.526074e-07
Nesterov	0.1	247	4.83	9.844587

Como podemos observar, en todas se llegó a un error menor a 0, pero la imagen se sigue viendo ruidosa, ¿Estuvo mal la toma del error?

No, no está mal la toma del error, solo que el error es un promedio sobre todos los píxeles ya que el algoritmo que se utilizó es el de MSE. Esto significa que el error está concentrado en zonas muy pequeñas o que visualmente no molestan mucho, como en el caso de las imágenes, donde podemos distinguir la forma del animal, que es lo que se busca.[1]

También es importante remarcar el tiempo que duró cada descenso, ya que podemos observar que el descenso de gradiente simple no llegó al error esperado, pero llegó a un error menor a 0. Al contrario del descenso con aceleración de Nesterov ya que no importó la variación de los parámetros, se mantuvo en 4 segundos y en 247 iteraciones. Y por último el descenso de gradiente con momentum tuvo resultados interesantes ya que fue más rápido con la imagen con mayor ruido pero más lento con menor ruido, y se debe a su característica de ser menos eficiente para problemas bien condicionados.

Conclusiones

La técnica de denoising es sumamente interesante, ya que da una mejor comprensión de las imágenes, en este caso con ruido gaussiano nos mostró que no importa que tanto optimicemos la función objetivo, no se quitará el ruido en su calidad. Eso no significa que este mal, ya que habrá veces donde no necesitamos que la imagen se vea sin ruido, solo necesitamos recalcar sus características y después de sacar sus características podríamos utilizar otras herramientas para terminar de pulir la imagen.

Bibliografía:

- [1] "¿Qué es el ruido gaussiano?", *Statistics Easily*, [En línea]. Disponible en: <https://es.statisticseasily.com/glosario/%C2%BFQu%C3%A9-es-el-ruido-gaussiano%3F/>. [Accedido: 8-may-2025].
- [2] S. López, "Comparación de filtros adaptativos para la reducción de ruido en imágenes digitales", *Computación y Sistemas*, vol. 21, no. 2, pp. 381-390, 2017. [En línea]. Disponible en: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462017000200381. [Accedido: 8-may-2025].
- [3] "Descenso de gradiente con impulso de Nesterov desde cero", *Top Big Data*, [En línea]. Disponible en: <https://topbigdata.es/descenso-de-gradiente-con-impulso-de-nesterov-desde-cero/>. [Accedido: 8-may-2025].
- [4] V. Mazet, "Denoising", *Image Restoration Notebook*, [En línea]. Disponible en: <https://vincmazet.github.io/bip/restoration/denoising.html>. [Accedido: 8-may-2025].
- [5] A. Taleb and T. Maliki, "Image denoising algorithms using norm minimization techniques," in *Proceedings of the International Workshop on Artificial Intelligence for Health (AIH 2021)*, vol. 2904, pp. 1–10, 2021. [En línea]. Disponible en: <https://ceur-ws.org/Vol-2904/17.pdf>. [Accedido: 8-may-2025].