

# UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS  
EXACTAS E INGENIERÍA  
DEPARTAMENTO DE  
CIENCIAS  
COMPUTACIONALES

**Seminario de Solución de Problemas de  
Sistemas Basados en Conocimiento**

## Práctica No. 3 Descenso del Gradiente

Nombre: Hurtado González Edgar Arturo  
Código: 212597894

### *Introducción*

El descenso del gradiente es un algoritmo de optimización utilizado comúnmente en el aprendizaje automático y en la optimización de funciones. Su objetivo principal es encontrar el mínimo de una función, es decir, el valor más bajo posible.

Como una función de múltiples variables representada en un paisaje en tres dimensiones, donde el eje  $z$  muestra el valor de la función y los ejes  $x$  e  $y$  representan los parámetros de la función. El descenso del gradiente comienza en algún punto de este paisaje y avanza hacia el mínimo de la función ajustando los parámetros en cada paso.

El proceso implica calcular el gradiente de la función en el punto actual. El gradiente es un vector que indica la dirección y la magnitud del mayor incremento de la función en ese punto. El algoritmo entonces se mueve en la dirección opuesta al gradiente, ya que esa dirección representa la pendiente descendente más pronunciada, acercándose así al mínimo de la función.

El tamaño de cada paso se controla mediante un parámetro llamado tasa de aprendizaje. Una tasa de aprendizaje alta puede hacer que el algoritmo converja más rápidamente, pero también puede hacer que se salte el mínimo global o que oscile alrededor de él. Por otro lado, una tasa de aprendizaje demasiado baja puede hacer que el algoritmo tarde mucho en converger.

El descenso del gradiente puede ser de dos tipos principales: el descenso del gradiente estocástico (SGD, por sus siglas en inglés) y el descenso del gradiente por lotes (Batch Gradient Descent). SGD utiliza un único ejemplo de datos en cada iteración, lo que puede hacerlo más rápido pero más ruidoso, mientras que el descenso del gradiente por lotes considera todos los datos disponibles en cada paso, siendo más preciso, pero potencialmente más lento.

Es fundamental para muchas técnicas de aprendizaje automático, como el entrenamiento de redes neuronales, ya que se utiliza para minimizar la función de pérdida y encontrar los parámetros óptimos que hacen que el modelo se ajuste mejor a los datos de entrenamiento.

### *Variantes del Descenso del Gradiente:*

1. *Mini-Batch Gradient Descent:* Esta variante combina las ventajas del SGD y el Batch Gradient Descent. En lugar de utilizar todos los datos (Batch) o un solo ejemplo (SGD), se dividen los datos en lotes más pequeños y se realiza la optimización utilizando cada uno de estos lotes. Esto equilibra la eficiencia y la estabilidad del algoritmo.
2. *Descenso del gradiente con momento (Gradient Descent with Momentum):* Añade una noción de inercia al descenso del gradiente, lo que ayuda a acelerar el proceso de convergencia en ciertos casos. El "momentum" acumula una fracción del gradiente anterior para determinar la dirección del siguiente paso. Esto ayuda a evitar oscilaciones en direcciones subóptimas y acelera el descenso en dirección al mínimo.
3. *Descenso del gradiente adaptativo (Adaptive Gradient Descent):* Ajusta la tasa de aprendizaje para cada parámetro individualmente según su historial de gradientes. Métodos populares como RMSprop y Adam se basan en este enfoque, adaptando la tasa de aprendizaje a la magnitud de los gradientes para una convergencia más rápida y estable.

### *Problemas y Soluciones:*

1. *Problema de mínimos locales y globales:* Algunas funciones tienen varios mínimos locales y un mínimo global. El descenso del gradiente puede quedar atrapado en mínimos locales si la inicialización no es adecuada. Estrategias como reiniciar el entrenamiento con diferentes inicializaciones o utilizar algoritmos más avanzados pueden ayudar a abordar este problema.

2. *Ajuste de la tasa de aprendizaje*: Encontrar la tasa de aprendizaje óptima puede ser desafiante. Métodos de búsqueda de hiperparámetros y técnicas de programación adaptativa pueden ser útiles para ajustar la tasa de aprendizaje automáticamente durante el entrenamiento.

Otros usos y aplicaciones:

1. *Optimización de hiperparámetros*: El descenso del gradiente se utiliza para optimizar los hiperparámetros de modelos de aprendizaje automático, como la tasa de aprendizaje, la regularización, entre otros, para mejorar el rendimiento del modelo.
2. *Transferencia de aprendizaje*: En la transferencia de aprendizaje, se pueden utilizar gradientes precalculados de modelos previamente entrenados para acelerar y mejorar el entrenamiento en tareas relacionadas.
3. *Redes neuronales profundas*: El descenso del gradiente es fundamental en el entrenamiento de redes neuronales profundas, donde se ajustan millones de parámetros para minimizar la función de pérdida y lograr un buen rendimiento en la tarea deseada.

El descenso del gradiente y sus variantes son elementos clave en la optimización y el entrenamiento de modelos en el aprendizaje automático, siendo fundamental comprender sus conceptos y aplicaciones para mejorar y diseñar modelos más eficaces.

## Desarrollo

El código presentado en esta ocasión implementa el algoritmo de descenso del gradiente para optimizar una función bidimensional y luego visualiza el proceso de optimización en una gráfica tridimensional.

*Definición de la Función y el Gradiente*:  $f(x)$  es la función objetivo, definida como:

$$f(x_1, x_2) = 10 - e^{-(x_1^2 + 3x_2^2)}$$

- `objective_function(x, y)` es la función que devuelve la expresión mencionada anteriormente y es la función que se optimizará utilizando el descenso de gradiente.

*Cálculo de las Derivadas Parciales*: Se definen las funciones `deriv_X(x, y)` y `deriv_Y(x, y)` que calculan las derivadas parciales de la función objetivo con respecto a  $x$  y  $y$  respectivamente.

*Descenso del Gradiente*: La función `gradient_descent` toma la función objetivo, las derivadas parciales, un límite para los valores de  $x$  e  $y$ , el número de iteraciones y la tasa de aprendizaje como entrada. Este algoritmo realiza lo siguiente:

- Inicializa un punto aleatorio dentro de los límites especificados.
- Para el número de iteraciones dado:
  - Calcula las derivadas parciales en el punto actual.

- Actualiza los valores de  $x$  e  $y$ , utilizando el descenso del gradiente.
- Evalúa el punto candidato en la función objetivo y almacena la solución y el puntaje.

*Visualización*: La función `graph` crea una malla bidimensional para  $x$  e  $y$  y calcula los valores de la función objetivo en esa malla. Luego, genera una figura tridimensional que muestra la superficie de la función objetivo. Además, traza el recorrido seguido por el descenso de gradiente sobre esta superficie.

*Parámetros Iniciales y Optimización*: Se establecen los límites para  $x$  e  $y$ , la tasa de aprendizaje y el número de iteraciones. Luego, se realiza la optimización llamando a la función `gradient_descent` con estos parámetros.

*Visualización del Proceso de Optimización*: Se llama a la función `graph` con las soluciones encontradas a lo largo de las iteraciones para mostrar cómo el descenso de gradiente converge hacia un óptimo local de la función objetivo.

La segunda parte del código realiza una regresión lineal utilizando el algoritmo del descenso de gradiente para encontrar los parámetros óptimos (pendiente e intercepto) que mejor se ajusten a los datos de entrada.

*Importar y Preparar los Datos*: Se cargan los datos de un archivo CSV llamado 'Salary\_dataset.csv', que se espera contenga columnas etiquetadas como 'YearsExperience' y 'Salary'. Se extraen los valores de estas columnas como las variables predictoras (YearsExperience) y la variable objetivo (Salary).

*Función de Error (MSE)*: Se define la función `mse(y_true, y_predicted)` para calcular el error cuadrático medio (MSE) entre los valores reales ( $y_{true}$ ) y los valores predichos ( $y_{predicted}$ ).

*Descenso de Gradiente para Regresión Lineal*: La función `linear_regression_gradient_descent` implementa el algoritmo del descenso de gradiente para la regresión lineal. Aquí está el flujo principal del algoritmo:

- **Inicialización de parámetros**: La pendiente (slope) y la intersección (intercept) se inician en 0.
- **Iteraciones**: Se realiza un bucle a lo largo de las iteraciones especificadas.
- **Predicción**: Se calculan los valores predichos utilizando la ecuación de la regresión lineal:  $y = mx + b$  donde  $m$  es la pendiente y  $b$  es la intersección.
- **Cálculo del Error**: Se calcula el error usando la función de error (MSE).
- **Actualización de Parámetros**: Se calculan las derivadas parciales del error con respecto a la pendiente y la intersección, y se utilizan para actualizar estos parámetros a lo largo del gradiente descendente.
- **Detención**: Se establece un umbral de detención para verificar si el cambio en el costo es menor que un valor pequeño (`stopping_threshold`). Si se cumple, el bucle se detiene antes de completar todas las iteraciones.
- Se almacenan los errores y los valores de pendiente e intersección en cada iteración.

*Mostrando los Resultados en una Gráfica:* Se calcula la predicción final utilizando los parámetros encontrados. Luego, se grafican los puntos de datos originales y la línea de regresión obtenida sobre ellos.

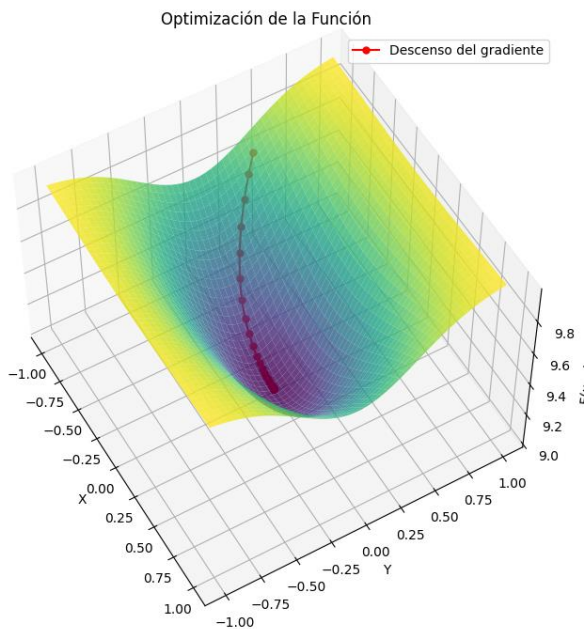
*Visualización del Ajuste de la Regresión:* Se muestra un gráfico que presenta los datos originales dispersos y la línea de regresión lineal ajustada utilizando el descenso de gradiente.

## Conclusión

En conclusión, el código realiza la optimización de una función bidimensional utilizando el método del descenso de gradiente y muestra visualmente el proceso de búsqueda del óptimo sobre la superficie tridimensional de la función objetivo.

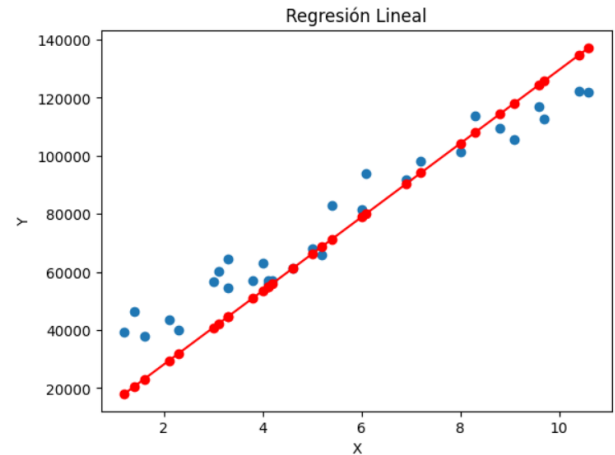
Después también se lleva a cabo la regresión lineal utilizando el descenso de gradiente, encontrando la mejor línea de ajuste para los datos proporcionados y mostrando visualmente la relación entre las variables predictoras y la variable objetivo.

## Resultados



Pendiente: 12680.086193503517

Intersección: 2805.924102789842



## Referencias:

What is gradient descent? | IBM. (s. f.).  
<https://www.ibm.com/mx-es/topics/gradient-descent>

colaboradores de Wikipedia. (2023, 13 noviembre).  
 Descenso del gradiente. Wikipedia, la enciclopedia libre.  
[https://es.wikipedia.org/wiki/Descenso\\_del\\_gradiente](https://es.wikipedia.org/wiki/Descenso_del_gradiente)

Abril, R. R. (2023). El descenso del gradiente. La Máquina Oráculo. <https://lamaquinaoraculo.com/deep-learning/el-descenso-del-gradiente/>