Descenso del Gradiente

27 de enero de 2025

Definición del Concepto

El descenso del gradiente es un método iterativo para minimizar (o maximizar) una función, muy utilizado en optimización y en el entrenamiento de modelos de aprendizaje automático. Su idea central consiste en moverse en la dirección contraria al gradiente de la función objetivo, pues el gradiente señala la dirección de máximo aumento. Sea $\mathbf{x} \in \mathbb{R}^n$ y $f(\mathbf{x})$ una función diferenciable. Comenzando en un punto inicial $\mathbf{x}_0$, la actualización iterativa se define como:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k),$$

donde η (tasa de aprendizaje) es un parámetro que controla la magnitud del paso. El objetivo es hallar $\mathbf{x}^*$ que minimice $f(\mathbf{x})$. Si η es demasiado grande, el algoritmo puede oscilar o divergir; si es muy pequeño, la convergencia puede ser muy lenta. El descenso del gradiente encuentra aplicaciones extensas en regresión, clasificación y redes neuronales, donde se entrena ajustando pesos y parámetros para minimizar una función de costo. Su simplicidad de implementación y su efectividad en alta dimensión lo convierten en una técnica predominante, aunque puede ser sensible a la elección de la tasa de aprendizaje y a la condición de la matriz Hessiana. Para abordar estas limitaciones, se han desarrollado variantes como el descenso del gradiente estocástico (SGD) y métodos adaptativos (*Adam*, *RMSProp*, *Adagrad*), que seleccionan la magnitud de cada paso en función de la geometría local de la función objetivo o de subselecciones de datos.

Ejemplo Detallado

Consideremos la función:

$$f(x, y) = (x - 1)^2 + (y - 2)^2.$$

Queremos aplicar descenso del gradiente para encontrar su mínimo. A continuación, se explica el procedimiento paso a paso.

1. Cálculo del gradiente

La función objetivo es

$$f(x, y) = (x - 1)^2 + (y - 2)^2.$$

Su gradiente (un vector con las derivadas parciales) es

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = \big(2(x-1),\, 2(y-2)\big).$$

2. Punto inicial y tasa de aprendizaje

Elegimos $\mathbf{x}_0 = (3,\, 1)$ como punto de partida y establecemos la tasa de aprendizaje $\eta = 0{,}2$.

3. Iteración 1

- **Gradiente en $\mathbf{x}_0$**:

$$\nabla f(3, 1) = (2(3-1),\, 2(1-2)) = (4,\, -2).$$

- **Actualización**:

$$\mathbf{x}_1 = \mathbf{x}_0 - \eta\, \nabla f(3, 1) = (3,\, 1) - 0{,}2\,(4,\, -2) = (3 - 0{,}8,\, 1 + 0{,}4) = (2{,}2,\, 1{,}4).$$

- **Valor de la función**:

$$f(3, 1) = (3-1)^2 + (1-2)^2 = 4 + 1 = 5.$$

4. Iteración 2

- **Gradiente en $\mathbf{x}_1$**:

$$\nabla f(2{,}2, 1{,}4) = (2(2{,}2-1),\, 2(1{,}4-2)) = (2{,}4,\, -1{,}2).$$

- **Actualización**:

$$\mathbf{x}_2 = \mathbf{x}_1 - \eta\, \nabla f(2{,}2, 1{,}4) = (2{,}2,\, 1{,}4) - 0{,}2\,(2{,}4,\, -1{,}2) = (2{,}2 - 0{,}48,\, 1{,}4 + 0{,}24) = (1{,}72,\, 1{,}64).$$

- **Valor de la función**:

$$f(2{,}2, 1{,}4) = (1{,}2)^2 + (-0{,}6)^2 = 1{,}44 + 0{,}36 = 1{,}80.$$

5. Iteración 3

- **Gradiente en $\mathbf{x}_2$**:

$$\nabla f(1{,}72, 1{,}64) = \big(2(1{,}72-1),\, 2(1{,}64-2)\big) = (1{,}44,\, -0{,}72).$$

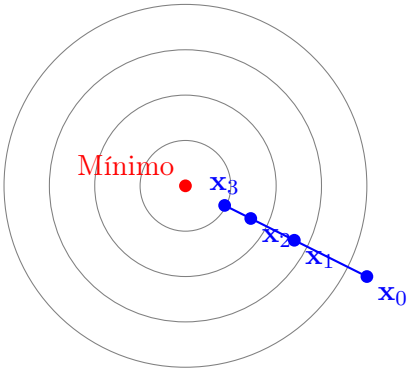
- **Actualización**:

$$\mathbf{x}_3 = \mathbf{x}_2 - \eta\, \nabla f(1{,}72, 1{,}64) = (1{,}72,\, 1{,}64) - 0{,}2\,(1{,}44,\, -0{,}72) = (1{,}72 - 0{,}288,\, 1{,}64 + 0{,}144) = (1{,}432,\, 1{,}7$$

- **Valor de la función**:

$$f(1{,}72, 1{,}64) = (0{,}72)^2 + (-0{,}36)^2 = 0{,}5184 + 0{,}1296 = 0{,}648.$$

Repitiendo estas iteraciones, $\mathbf{x}_k$ se acerca progresivamente al mínimo $(1, 2)$. A continuación, se ilustra el proceso en un diagrama `TikZ` con curvas de nivel y los puntos iterativos:



Ejercicios Propuestos

1. Mínimo de una Función Cuadrática en 1D con Datos Concretos: Minimiza la función

$$g(x) = (x - 5)^2$$

empezando en $x_0 = 10$, con tasa de aprendizaje $\eta = 0{,}2$. Realiza 5 iteraciones manualmente:

$$x_{k+1} = x_k - 0{,}2 \cdot \frac{d}{dx} g(x_k).$$

Anota en una tabla los valores $(k, x_k, g(x_k))$ para cada paso. Explica por qué el resultado tiende a $x = 5$.

2. Tienes los siguientes 5 puntos de entrenamiento:

$$(x_i, y_i) \in \{(1, 2),\ (2, 2{,}8),\ (3, 3{,}6),\ (4, 4{,}5),\ (5, 5{,}1)\}.$$

Ajusta la recta $h(x) = \beta_0 + \beta_1 x$ minimizando

$$J(\beta_0, \beta_1) = \sum_{i=1}^{5} \big(y_i - (\beta_0 + \beta_1 x_i)\big)^2$$

mediante descenso del gradiente. Elige $\eta = 0{,}01$ y realiza al menos 3 iteraciones para actualizar β_0 y β_1. Muestra en cada iteración los nuevos valores de (β_0, β_1) y el costo J.

3. Considera este conjunto de datos con dos características (atributos) x_1, x_2 y etiqueta binaria y:

Muestra	x_1	x_2	y
1	0,5	1,0	0
2	1,5	2,0	0
3	2,0	2,5	1
4	3,0	3,5	1

   Define un modelo de clasificación logística $\sigma(\mathbf{w}^\top \mathbf{x})$ y la función de costo logístico. Inicia con pesos $\mathbf{w}_0 = (0, 0, 0)$ (incluyendo el sesgo como componente adicional). Aplica 3 iteraciones de descenso del gradiente con $\eta = 0{,}1$. Muestra las actualizaciones y cómo disminuye el error de clasificación (o el valor de la función de costo) en cada paso.

4. Supón que dispones de 1000 observaciones $\{(\mathbf{x}_i, y_i)\}_{i=1}^{1000}$ para un problema de regresión multivariable. Divide el conjunto en minibatches de tamaño 50. Emplea la función de costo

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \big(y_i - \mathbf{w}^\top \mathbf{x}_i\big)^2.$$

   Explica cómo aplicarías SGD tomando un minibatch de 50 datos en cada iteración y describe por qué este procedimiento puede converger más rápido en la práctica que el descenso por lotes completos (*batch gradient descent*). Para guiar el cálculo, utiliza una tasa de aprendizaje $\eta = 0{,}01$ y muestra, al menos de forma hipotética, cómo se modificarían los parámetros $\mathbf{w}$ tras varias iteraciones sobre distintos minibatches.

Referencias

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Nocedal, J., & Wright, S. (2006). *Numerical Optimization* (2nd ed.). Springer.