Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente: Ing. Fred Torres Cruz
Estudiante: Luis Angel Quenaya Loza
Código: 241411

Trabajo N°7: Gradiente de una Función

1. Concepto del Gradiente

El gradiente de una función representa la dirección y la rapidez con que cambia su valor respecto a sus variables. En otras palabras, muestra hacia dónde y cuánto aumenta o disminuye una función.

Si la función depende de una sola variable, el gradiente se reduce a su derivada. Para una función $f(x)$, el método del **descenso del gradiente** busca el punto donde la función alcanza su valor mínimo, actualizando los valores de x según:

$$x_{i+1} = x_i - \eta f'(x_i)$$

donde:

- x_i: valor actual de la variable.
- η: tasa de aprendizaje, controla el tamaño del paso.
- $f'(x_i)$: derivada de la función en x_i.

Este proceso se repite hasta que los cambios sean insignificantes y la función se acerque a su mínimo global.

2. Implementación en R (una variable)

El siguiente código en `R` aplica el método del gradiente descendente a la función $f(x) = x^2$, cuya derivada es $f'(x) = 2x$. Se observa cómo el valor de x va disminuyendo gradualmente hasta aproximarse al punto mínimo en $x = 0$.

```
#-----------------------------------------------------
# M todo del gradiente descendente para f(x) = x^2
#-----------------------------------------------------

n <- 0.01  # tasa de aprendizaje
f <- function(x) x^2
f_deriv <- function(x) 2 * x

```

```
x0 <- 3     # valor inicial
iter <- 21 # n mero de iteraciones

x <- numeric(iter)
fx <- numeric(iter)
fpx <- numeric(iter)
grad <- numeric(iter)

x[1] <- x0

for (i in 1:iter) {
  fx[i] <- f(x[i])
  fpx[i] <- f_deriv(x[i])
  grad[i] <- x[i] - n * fpx[i]
  if (i < iter) {
    x[i + 1] <- grad[i]
  }
}

tabla <- data.frame(x = x, f_x = fx, f_deriv = fpx, grad = grad)
print(tabla)
```

	xo	fx	fpx	grad
1	3.000000	9.000000	6.000000	2.940000
2	2.940000	8.643600	5.880000	2.881200
3	2.881200	8.301313	5.762400	2.823576
4	2.823576	7.972581	5.647152	2.767104
5	2.767104	7.656867	5.534209	2.711762
6	2.711762	7.353655	5.423525	2.657527
7	2.657527	7.062451	5.315054	2.604377
8	2.604377	6.782777	5.208753	2.552289
9	2.552289	6.514179	5.104578	2.501243
10	2.501243	6.256218	5.002487	2.451218
11	2.451218	6.008472	4.902437	2.402194
12	2.402194	5.770536	4.804388	2.354150

Figura 1: Evolución del valor de x durante el descenso del gradiente para $f(x) = x^2$.

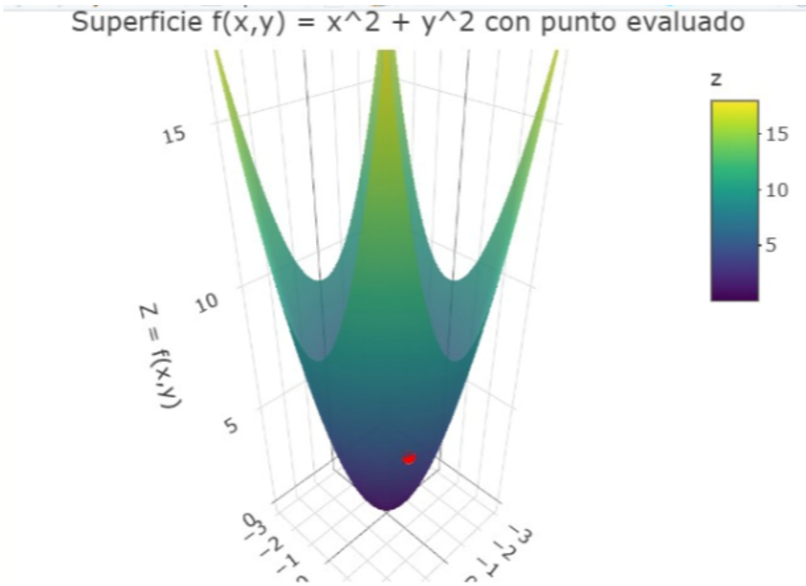

Figura 2: Comportamiento gráfico de la función $f(x)$ y su gradiente.

3. Gradiente de una Función de Dos Variables

Cuando una función depende de dos variables, el gradiente se convierte en un vector formado por las derivadas parciales respecto a cada variable. Este vector apunta en la dirección de máximo crecimiento de la función y se define como:

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

El método del **descenso del gradiente** busca el punto mínimo moviéndose en la dirección opuesta al gradiente. Las ecuaciones de actualización son:

$$x_{i+1} = x_i - \eta \frac{\partial f}{\partial x}(x_i, y_i)$$
$$y_{i+1} = y_i - \eta \frac{\partial f}{\partial y}(x_i, y_i)$$

donde η controla la magnitud del paso en cada iteración.

4. Ejemplo en R (dos variables)

Consideremos la función:

$$f(x, y) = 3x^2y^3 + 6xy^2$$

Sus derivadas parciales son:

$$\frac{\partial f}{\partial x} = 6xy^3 + 6y^2, \quad \frac{\partial f}{\partial y} = 9x^2y^2 + 12xy$$

El siguiente código implementa el método del gradiente descendente para esta función, mostrando cómo las coordenadas (x, y) se aproximan al punto de mínimo en cada iteración.

```
#------------------------------------------------------------
# Gradiente descendente para f(x,y) = 3x^2y^3 + 6xy^2
#------------------------------------------------------------

f <- function(x, y) 3*x^2*y^3 + 6*x*y^2
fx <- function(x, y) 6*x*y^3 + 6*y^2
fy <- function(x, y) 9*x^2*y^2 + 12*x*y

n <- 0.01   # tasa de aprendizaje
iter <- 20  # n mero de iteraciones

x <- numeric(iter)
y <- numeric(iter)
z <- numeric(iter)

x[1] <- 2
y[1] <- 1

for (i in 1:iter) {
  z[i] <- f(x[i], y[i])
  dfx <- fx(x[i], y[i])
  dfy <- fy(x[i], y[i])
  if (i < iter) {
    x[i + 1] <- x[i] - n * dfx
    y[i + 1] <- y[i] - n * dfy
  }
}

tabla <- data.frame(iter = 1:iter, x = x, y = y, fxy = z)
print(tabla)
```

	iter	x	y	fxy
1	1	2.000000	1.000000000	2.400000e
2	2	1.820000	0.400000000	2.383181e
3	3	1.803411	0.264941440	9.409830
4	4	1.797187	0.187059422	4.407377
5	5	1.794382	0.136546147	2.253276
6	6	1.792989	0.101741280	1.215156
7	7	1.792255	0.076855790	6.789380
8	8	1.791852	0.058618735	3.888264
9	9	1.791624	0.045021475	2.266778
10	10	1.791492	0.034756524	1.338917
11	11	1.791415	0.026935663	7.986500

Figura 3: Trayectoria del descenso del gradiente en $f(x, y)$.

5. Conclusión

El método del gradiente descendente es una herramienta esencial para encontrar mínimos de funciones. A través de pasos sucesivos, el algoritmo ajusta los valores de las variables para acercarse progresivamente al punto de menor valor.

Este enfoque no solo es fundamental en análisis numérico, sino que también constituye la base de muchos algoritmos de **aprendizaje automático**, donde se optimizan funciones de error o de costo. Comprender su funcionamiento permite interpretar de manera más profunda cómo los modelos mejoran su rendimiento mediante ajustes iterativos.