

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
E.P. de Ingeniería Estadística e Informática

Docente: Ing. Torres Cruz Fred
Presentado por: Quispe Ito Luz Leidy

GRADIENTE

El **gradiente** de una función representa la dirección y la magnitud del cambio más rápido de dicha función. En una función de una sola variable $f(x)$, el gradiente corresponde simplemente a su derivada:

$$\nabla f(x) = f'(x)$$

En una función de varias variables, por ejemplo $f(x, y)$, el gradiente es un vector que contiene todas las derivadas parciales:

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

El gradiente apunta en la dirección de **mayor incremento** de la función. Por ello, para **minimizar** una función se sigue la dirección contraria al gradiente.

Método del Descenso del Gradiente

El **descenso del gradiente** es un algoritmo iterativo utilizado para encontrar el mínimo de una función diferenciable. En cada iteración, el valor de la variable se actualiza según la regla:

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

donde:

- x_i : valor actual de la variable,
- η : tasa de aprendizaje,
- $f'(x_i)$: derivada o pendiente de la función en x_i .

El proceso se repite hasta que la derivada $f'(x)$ se aproxima a cero, lo cual indica que se ha alcanzado un punto mínimo (local o global).

Ejemplo aplicado

Para la función:

$$f(x) = x^2 \Rightarrow f'(x) = 2x$$

La actualización se realiza como:

$$x_{i+1} = x_i - n(2x_i)$$

Si el valor inicial es $x_0 = 3$ y la tasa de aprendizaje $n = 0,01$, los valores de x se van acercando gradualmente al punto mínimo en $x = 0$, donde la pendiente $f'(x) = 0$.

Este proceso se denomina **gradiente descendente**, ya que cada paso desciende la función hacia su valor mínimo.

1. Implementación en R para - Gradiente de una función

En esta parte se describe el código R utilizado para implementar el método del gradiente.

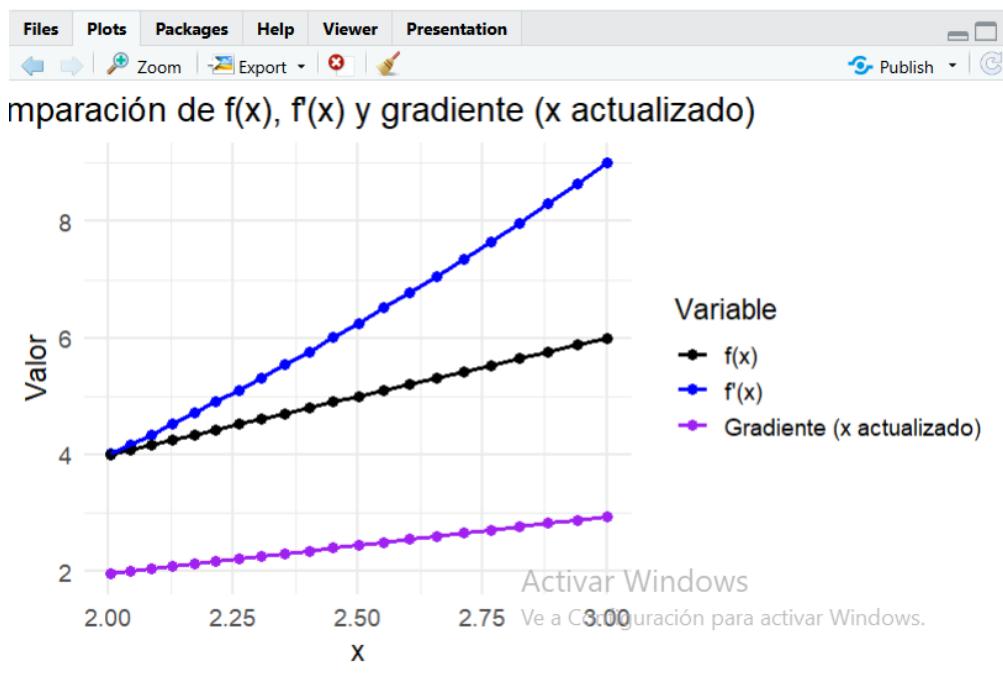
```

1 # GRADIENTE
2 n <- 0.01           # Tasa de aprendizaje
3 x0 <- 3             # Valor inicial
4 iter <- 21           # Numero de iteraciones
5
6 # DEFINICION DE FUNCIONES
7 f <- function(x) x^2          # Funcion objetivo
8 f_deriv <- function(x) 2 * x    # Derivada
9
10 # INICIALIZACION DE VECTORES
11 x <- numeric(iter)
12 fx <- numeric(iter)
13 fpx <- numeric(iter)
14 grad <- numeric(iter)
15
16 x[1] <- x0
17
18 # CALCULO ITERATIVO
19 for (i in 1:iter) {
20   fx[i] <- f(x[i])            # f(x)
21   fpx[i] <- f_deriv(x[i])      # f'(x)
22   grad[i] <- x[i] - n * fpx[i] # x actualizado
23   if (i < iter) {
24     x[i + 1] <- grad[i]
25   }
}
```

```
26 }
27
28 # TABLA DE RESULTADOS
29 tabla <- data.frame(
30   Iteracion = 1:iter,
31   xo = x,
32   fx = fx,
33   fpx = fpx,
34   grad = grad
35 )
36 print(tabla)
37
38 library(ggplot2)
39 library(tidyr)
40
41 # DATOS PARA GRAFICO
42 datos_long <- tabla |>
43   pivot_longer(cols = c(fx, fpx, grad),
44                 names_to = "Variable",
45                 values_to = "Valor")
46
47 # GRAFICO
48 ggplot(datos_long, aes(x = xo, y = Valor, color = Variable)) +
49   geom_point(size = 2) +
50   geom_line(linewidth = 1) +
51   scale_color_manual(values = c("black", "blue", "purple"),
52                      labels = c("f(x)", "f'(x)", "Gradiente (x actualizado")
53                      )) +
54   labs(title = "Comparacion de f(x), f'(x) y gradiente (x actualizado)",
55        x = "x",
56        y = "Valor",
57        color = "Variable") +
58   theme_minimal(base_size = 14) +
59   theme(plot.title = element_text(hjust = 0.5))
```

Listing 1: Implementación del metodo del gradiente en R

Iteración	x ₀	f(x)	f'(x)	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
13	2.354150	5.542023	4.708300	2.307067
14	2.307067	5.322559	4.614134	2.260926
15	2.260926	5.111786	4.521852	2.215707
16	2.215707	4.909359	4.431415	2.171393
17	2.171393	4.714948	4.342786	2.127965
18	2.127965	4.528236	4.255931	2.085406
19	2.085406	4.348918	4.170812	2.043698
20	2.043698	4.176701	4.087396	2.002824
21	2.002824	4.011304	4.005648	1.962767



2. Implementación en R para - Gradiente de dos función

```
1 # GRADIENTE (2 VARIABLES)
2 library(ggplot2)
3 library(plotly)
4
5 # Parametros
6 n <- 0.1                      # Tasa de aprendizaje
7 x0 <- 3                         # Valor inicial de x
8 y0 <- -2                        # Valor inicial de y
9 iter <- 25                       # Numero de iteraciones
10
11 # Definicion de funciones
12 f <- function(x, y) x^2 + y^2          # Funcion objetivo
13 grad_f <- function(x, y) c(2*x, 2*y)      # Gradiente f (x, y)
14
15 # Inicializacion
16 x <- numeric(iter)
17 y <- numeric(iter)
18 fx <- numeric(iter)
19
20 x[1] <- x0
21 y[1] <- y0
22 fx[1] <- f(x[1], y[1])
23
24 # Iteraciones del gradiente
25 for (i in 1:(iter - 1)) {
26   grad <- grad_f(x[i], y[i])
27   x[i + 1] <- x[i] - n * grad[1]
28   y[i + 1] <- y[i] - n * grad[2]
29   fx[i + 1] <- f(x[i + 1], y[i + 1])
30 }
31
32 # Tabla de resultados
33 tabla <- data.frame(
34   Iter = 1:iter,
35   x = x,
36   y = y,
37   fxy = fx
38 )
39 print(tabla)
40
41 # Superficie 3D con trayectoria
42 x_seq <- seq(-4, 4, length.out = 100)
43 y_seq <- seq(-4, 4, length.out = 100)
44 z <- outer(x_seq, y_seq, f)
45
46 fig <- plot_ly() |>
47   add_surface(x = ~x_seq, y = ~y_seq, z = ~z, opacity = 0.7, colorscale =
48     "Viridis") |>
49   add_trace(x = x, y = y, z = fx, type = "scatter3d", mode = "lines+
50     markers",
51       line = list(color = "red", width = 5),
52       marker = list(size = 3, color = "red"),
53       name = "Trayectoria del gradiente") |>
```

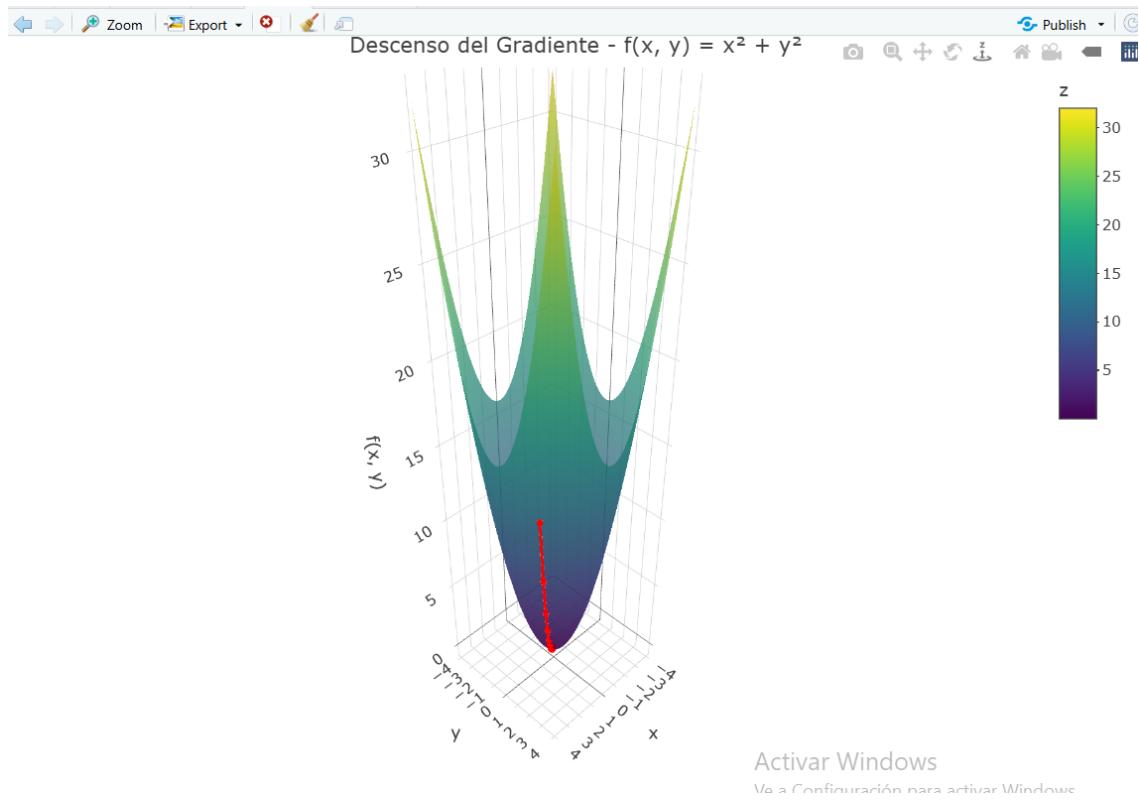
```

52 layout(
53   title = "Descenso del Gradiente - f(x, y) = x + y",
54   scene = list(
55     xaxis = list(title = "x"),
56     yaxis = list(title = "y"),
57     zaxis = list(title = "f(x, y)")
58   )
59 )
60
61 fig

```

Listing 2: Implementación del metodo del gradiente en R

Iter	x	y	fxy
1	3.00000000	2.000000000	1.300000e+01
2	2.40000000	1.600000000	8.320000e+00
3	1.92000000	1.280000000	5.324800e+00
4	1.53600000	1.024000000	3.407872e+00
5	1.22880000	0.819200000	2.181038e+00
6	0.98304000	0.655360000	1.395864e+00
7	0.78643200	0.524288000	8.933532e-01
8	0.62914560	0.419430400	5.717460e-01
9	0.50331648	0.335544320	3.659175e-01
10	0.40265318	0.268435456	2.341872e-01
11	0.32212255	0.214748365	1.498798e-01
12	0.25769804	0.171798692	9.592307e-02
13	0.20615843	0.137438953	6.139076e-02
14	0.16492674	0.109951163	3.929009e-02
15	0.13194140	0.087960930	2.514566e-02
16	0.10555312	0.070368744	1.609322e-02
17	0.08444249	0.056294995	1.029966e-02
18	0.06755399	0.045035996	6.591783e-03
19	0.05404320	0.036028797	4.218741e-03
20	0.04323456	0.028823038	2.699994e-03
21	0.03458765	0.023058430	1.727996e-03
22	0.02767012	0.018446744	1.105918e-03
23	0.02213609	0.014757395	7.077873e-04
24	0.01770887	0.011805916	4.529839e-04
25	0.01416710	0.009444733	2.899097e-04



Activar Windows

[Vea la Configuración para activar Windows](#)