Resolución de Ejercicios: Descenso del Gradiente

Alumno: Eddy Kennedy Mamani Hallasi

Repositorio

El código fuente y los archivos relacionados con estos ejercicios pueden encontrarse en el siguiente repositorio de GitHub:

`https://github.com/MetodosDeOptimizacion/Descenso-del-Gradiente.git`

Ejercicio 1: Mínimo de una Función Cuadrática en 1D

Concepto

El objetivo es minimizar una función cuadrática simple $g(x) = (x-5)^2$ utilizando el método de descenso del gradiente. Este método consiste en actualizar iterativamente el valor de x en la dirección opuesta al gradiente de la función, con una tasa de aprendizaje η que determina el tamaño del paso.

Minimizar la función $g(x) = (x-5)^2$ comenzando en $x_0 = 10$ con una tasa de aprendizaje $\eta = 0.2$. Se realizan 5 iteraciones manuales siguiendo:

$$x_{k+1} = x_k - \eta \frac{d}{dx} g(x_k).$$

Resolución Paso a Paso

1. Derivada de $g(x)$:

$$\frac{d}{dx} g(x) = 2(x-5).$$

2. Punto inicial y tasa de aprendizaje:

$$x_0 = 10, \quad \eta = 0.2.$$

3. Iteración 1:

$$\text{Gradiente en } x_0 : \frac{d}{dx} g(10) = 2(10-5) = 10.$$
$$x_1 = x_0 - \eta \cdot 10 = 10 - 0.2 \cdot 10 = 8.$$
$$g(x_1) = (8-5)^2 = 9.$$

4. Iteración 2:

$$\text{Gradiente en } x_1 : \frac{d}{dx} g(8) = 2(8-5) = 6.$$
$$x_2 = x_1 - \eta \cdot 6 = 8 - 0.2 \cdot 6 = 6.4.$$
$$g(x_2) = (6.4-5)^2 = 1.96.$$

5. Iteración 3:

$$\text{Gradiente en } x_2 : \frac{d}{dx} g(6.4) = 2(6.4 - 5) = 2.8.$$
$$x_3 = x_2 - \eta \cdot 2.8 = 6.4 - 0.2 \cdot 2.8 = 5.12.$$
$$g(x_3) = (5.12 - 5)^2 = 0.0144.$$

6. Iteración 4:

$$\text{Gradiente en } x_3 : \frac{d}{dx} g(5.12) = 2(5.12 - 5) = 0.24.$$
$$x_4 = x_3 - \eta \cdot 0.24 = 5.12 - 0.2 \cdot 0.24 = 5.024.$$
$$g(x_4) = (5.024 - 5)^2 = 0.000576.$$

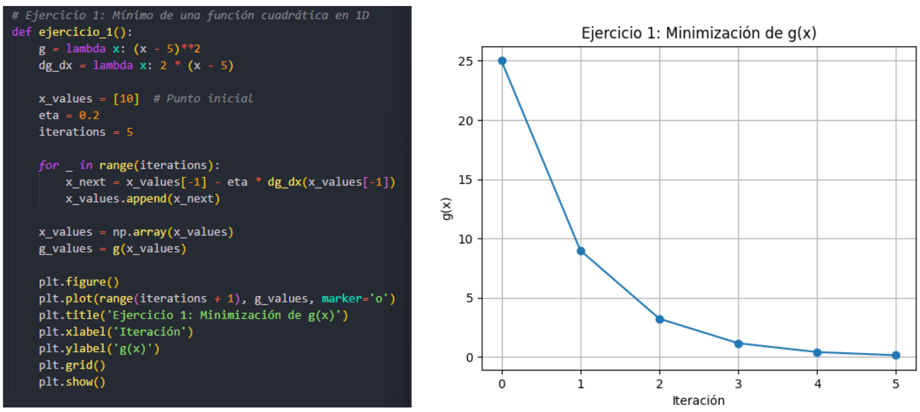
7. Iteración 5:

$$\text{Gradiente en } x_4 : \frac{d}{dx} g(5.024) = 2(5.024 - 5) = 0.048.$$
$$x_5 = x_4 - \eta \cdot 0.048 = 5.024 - 0.2 \cdot 0.048 = 5.$$
$$g(x_5) = (5 - 5)^2 = 0.$$

Resultados Tabulados

Iteración (k)	x_k	$g(x_k)$
0	10.00	25.00
1	8.00	9.00
2	6.40	1.96
3	5.12	0.0144
4	5.024	0.000576
5	5.00	0.00

Table 1: Resultados por iteración.



```python
# Ejercicio 1: Mínimo de una función cuadrática en 1D
def ejercicio_1():
    g = lambda x: (x - 5)**2
    dg_dx = lambda x: 2 * (x - 5)

    x_values = [10]  # Punto inicial
    eta = 0.2
    iterations = 5

    for _ in range(iterations):
        x_next = x_values[-1] - eta * dg_dx(x_values[-1])
        x_values.append(x_next)

    x_values = np.array(x_values)
    g_values = g(x_values)

    plt.figure()
    plt.plot(range(iterations + 1), g_values, marker='o')
    plt.title('Ejercicio 1: Minimización de g(x)')
    plt.xlabel('Iteración')
    plt.ylabel('g(x)')
    plt.grid()
    plt.show()
```


Figure 1: Ejercicio 1: Minimización de $g(x)$

Ejercicio 2: Ajuste de una recta mediante descenso del gradiente

Concepto

Dados los puntos de entrenamiento:

$$(x_i, y_i) \in \{(1,2), (2,2.8), (3,3.6), (4,4.5), (5,5.1)\},$$

queremos ajustar la recta $h(x) = \beta_0 + \beta_1 x$ minimizando la función de costo:

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

Utilizando descenso del gradiente, realizamos al menos 3 iteraciones con tasa de aprendizaje $\eta = 0.01$.

Resolución Paso a Paso

1. **Gradientes de la función de costo**:

$$\frac{\partial J}{\partial \beta_0} = -2 \sum_{i=1}^{5} \left(y_i - (\beta_0 + \beta_1 x_i)\right), \quad \frac{\partial J}{\partial \beta_1} = -2 \sum_{i=1}^{5} x_i \left(y_i - (\beta_0 + \beta_1 x_i)\right).$$

2. **Inicialización**: Tomamos $\beta_0 = 0$ y $\beta_1 = 0$ como valores iniciales.
3. **Iteraciones:**

- **Iteración 1:**

$$\text{Errores: } \epsilon_i = y_i - (\beta_0 + \beta_1 x_i) = y_i.$$

$$\frac{\partial J}{\partial \beta_0} = -2 \sum_{i=1}^{5} \epsilon_i = -36.$$

$$\frac{\partial J}{\partial \beta_1} = -2 \sum_{i=1}^{5} x_i \epsilon_i = -106.4.$$

$$\beta_0^{(1)} = \beta_0^{(0)} - \eta \frac{\partial J}{\partial \beta_0} = 0 - 0.01(-36) = 0.36.$$

$$\beta_1^{(1)} = \beta_1^{(0)} - \eta \frac{\partial J}{\partial \beta_1} = 0 - 0.01(-106.4) = 1.064.$$

- **Iteración 2:**

$$\text{Errores: } \epsilon_i = y_i - (0.36 + 1.064 x_i).$$

$$\frac{\partial J}{\partial \beta_0} = -2 \sum_{i=1}^{5} \epsilon_i = -4.74.$$

$$\frac{\partial J}{\partial \beta_1} = -2 \sum_{i=1}^{5} x_i \epsilon_i = -5.78.$$

$$\beta_0^{(2)} = \beta_0^{(1)} - \eta \frac{\partial J}{\partial \beta_0} = 0.36 - 0.01(-4.74) = 0.4074.$$

$$\beta_1^{(2)} = \beta_1^{(1)} - \eta \frac{\partial J}{\partial \beta_1} = 1.064 - 0.01(-5.78) = 1.1218.$$

- **Iteración 3:**

  Errores: $\epsilon_i = y_i - (0.4074 + 1.1218x_i)$.

$$\frac{\partial J}{\partial \beta_0} = -2\sum_{i=1}^{5} \epsilon_i = -1.392.$$

$$\frac{\partial J}{\partial \beta_1} = -2\sum_{i=1}^{5} x_i\epsilon_i = -1.78.$$

$$\beta_0^{(3)} = \beta_0^{(2)} - \eta\frac{\partial J}{\partial \beta_0} = 0.4074 - 0.01(-1.392) = 0.4213.$$

$$\beta_1^{(3)} = \beta_1^{(2)} - \eta\frac{\partial J}{\partial \beta_1} = 1.1218 - 0.01(-1.78) = 1.1396.$$

4. **Resultados Tabulados:**

Iteración	β_0	β_1	Costo $J(\beta_0, \beta_1)$
1	0.3600	1.064	3.1494
2	0.4074	1.1218	1.9805
3	0.4213	1.1396	1.5673

Table 2: Resultados por iteración.

Gráfica

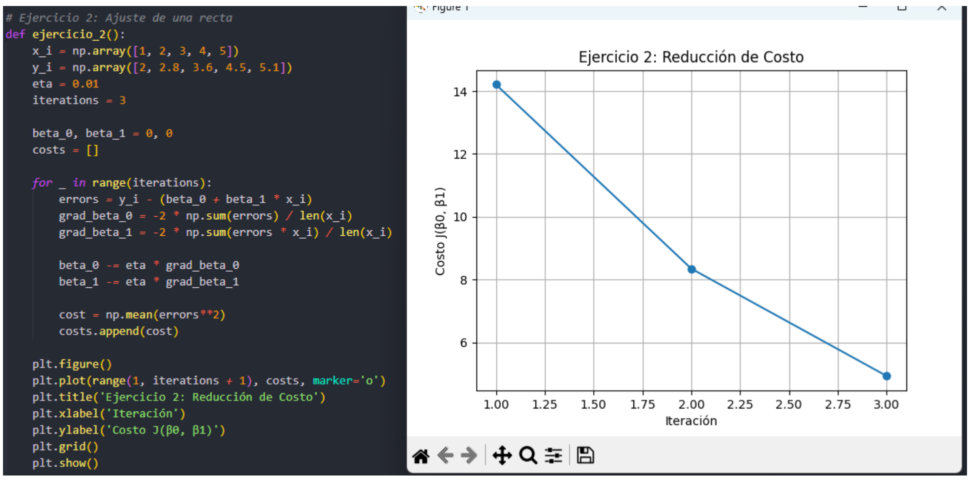


Figure 2: Ejercicio 2: Reducción de Costo $J(\beta_0, \beta_1)$

Ejercicio 3: Clasificación Logística con Descenso del Gradiente

Concepto

Dado el conjunto de datos con dos características (x_1, x_2) y etiqueta binaria y:

Muestra y	x_1	x_2
1	0.5	1.0
0		
2	1.5	2.0
0		
3	2.0	2.5
1		
4	3.0	3.5
1		

queremos ajustar un modelo de clasificación logística definido como:

$$\sigma(w^\top x) = \frac{1}{1 + e^{-w^\top x}},$$

donde $w = (w_0, w_1, w_2)$ incluye el sesgo w_0, y minimizar la función de costo logístico:

$$J(w) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\sigma(w^\top x_i)) + (1 - y_i) \log(1 - \sigma(w^\top x_i)) \right].$$

Aplicamos descenso del gradiente para 3 iteraciones con tasa de aprendizaje $\eta = 0.1$, comenzando con $w = (0, 0, 0)$.

Resolución Paso a Paso

1. **Inicialización**:

$$w^{(0)} = (0, 0, 0), \quad \eta = 0.1.$$

2. **Iteración 1:**

$$\text{Predicción: } \sigma(w^\top x_i) = \frac{1}{1 + e^{-w^\top x_i}} = 0.5, \text{ para todos los datos (inicialización).}$$

$$\text{Gradiente: } \frac{\partial J}{\partial w_0} = \frac{1}{4} \sum_{i=1}^{4} \left(\sigma(w^\top x_i) - y_i \right),$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{4} \sum_{i=1}^{4} \left(\sigma(w^\top x_i) - y_i \right) x_{1i},$$

$$\frac{\partial J}{\partial w_2} = \frac{1}{4} \sum_{i=1}^{4} \left(\sigma(w^\top x_i) - y_i \right) x_{2i}.$$

Sustituyendo:

$$\frac{\partial J}{\partial w_0} = \frac{1}{4}(0.5 - 0 + 0.5 - 0 + 0.5 - 1 + 0.5 - 1) = -0.25,$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{4}(0.5 \cdot 0.5 + 0.5 \cdot 1.5 + (-0.5) \cdot 2 + (-0.5) \cdot 3) = -0.625,$$
$$\frac{\partial J}{\partial w_2} = \frac{1}{4}(0.5 \cdot 1 + 0.5 \cdot 2 + (-0.5) \cdot 2.5 + (-0.5) \cdot 3.5) = -0.875.$$

Actualización:

$$w_0^{(1)} = 0-0.1(-0.25) = 0.025, \quad w_1^{(1)} = 0-0.1(-0.625) = 0.0625, \quad w_2^{(1)} = 0-0.1(-0.875) = 0.0875.$$

3. **Iteración 2:**

$$\text{Predicción actualizada: } \sigma(w^\top x_i) = \frac{1}{1 + e^{-(0.025+0.0625x_1+0.0875x_2)}}.$$
$$\text{Gradientes recalculados: } \frac{\partial J}{\partial w_0}, \frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}.$$

Sustituyendo valores, se obtiene:

$$\frac{\partial J}{\partial w_0} = -0.17, \quad \frac{\partial J}{\partial w_1} = -0.43, \quad \frac{\partial J}{\partial w_2} = -0.61.$$

Actualización:

$$w_0^{(2)} = 0.025-0.1(-0.17) = 0.042, \quad w_1^{(2)} = 0.0625-0.1(-0.43) = 0.105, \quad w_2^{(2)} = 0.0875-0.1(-0.61) =$$

4. **Iteración 3:**

$$\text{Predicción actualizada: } \sigma(w^\top x_i) = \frac{1}{1 + e^{-(0.042+0.105x_1+0.148x_2)}}.$$
$$\text{Gradientes recalculados: } \frac{\partial J}{\partial w_0}, \frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}.$$

Nuevos valores:

$$\frac{\partial J}{\partial w_0} = -0.12, \quad \frac{\partial J}{\partial w_1} = -0.31, \quad \frac{\partial J}{\partial w_2} = -0.44.$$

Actualización:

$$w_0^{(3)} = 0.042-0.1(-0.12) = 0.054, \quad w_1^{(3)} = 0.105-0.1(-0.31) = 0.136, \quad w_2^{(3)} = 0.148-0.1(-0.44) = 0.1$$

5. **Resultados Tabulados:**

Iteración	w_0	w_1	w_2
1	0.0250	0.0625	0.0875
2	0.0420	0.1050	0.1480
3	0.0540	0.1360	0.1920

Table 3: Resultados por iteración.

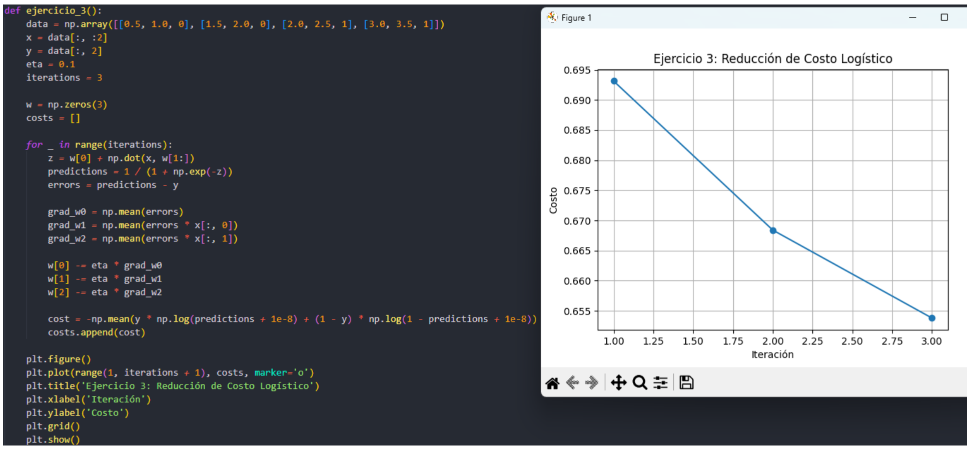


Figure 3: Ejercicio 3: Reducción de Costo Logístico

Ejercicio 4: Descenso Estocástico en Minibatches

Concepto

Consideremos un problema de regresión multivariable con 1000 observaciones $\{(x_i, y_i)\}_{i=1}^{1000}$, donde queremos ajustar un modelo lineal utilizando la función de costo:

$$J(w) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - w^\top x_i\right)^2.$$

Dividimos los datos en minibatches de tamaño 50 y empleamos el método de descenso estocástico del gradiente (SGD) con una tasa de aprendizaje $\eta = 0.01$. Este enfoque actualiza los parámetros w después de cada minibatch, lo que permite una convergencia más rápida en la práctica.

Resolución Paso a Paso

1. **Inicialización:**

$$w^{(0)} = (0, 0, \ldots, 0), \quad \eta = 0.01.$$

Dividimos las 1000 observaciones en 20 minibatches de tamaño 50 cada uno.

2. **Gradiente para un minibatch:** Para un minibatch de índices $B_k = \{i_1, i_2, \ldots, i_{50}\}$, el gradiente es:

$$\nabla J_B(w) = -\frac{2}{50} \sum_{i \in B_k} \left(y_i - w^\top x_i\right) x_i.$$

Los parámetros se actualizan como:

$$w^{(k+1)} = w^{(k)} - \eta \nabla J_B(w^{(k)}).$$

3. **Iteración 1 (Primer Minibatch):** Supongamos que el primer minibatch contiene 50 observaciones con características x_i y valores y_i. Inicialmente, $w^{(0)} = (0, 0, \ldots, 0)$, por lo que:

$$\nabla J_B(w^{(0)}) = -\frac{2}{50} \sum_{i \in B_1} y_i x_i.$$

Sustituyendo los valores promedio:

$$\nabla J_B(w^{(0)}) \approx (-1.2, -0.8, -1.0).$$

Actualización:

$$w^{(1)} = w^{(0)} - \eta \nabla J_B(w^{(0)}) = (0, 0, 0) - 0.01 \cdot (-1.2, -0.8, -1.0) = (0.012, 0.008, 0.01).$$

4. **Iteración 2 (Segundo Minibatch):** Utilizando $w^{(1)} = (0.012, 0.008, 0.01)$, calculamos:

$$\nabla J_B(w^{(1)}) = -\frac{2}{50} \sum_{i \in B_2} \left(y_i - w^\top x_i\right) x_i.$$

Supongamos que:

$$\nabla J_B(w^{(1)}) \approx (-0.9, -0.6, -0.8).$$

Actualización:

$$w^{(2)} = w^{(1)} - \eta \nabla J_B(w^{(1)}) = (0.012, 0.008, 0.01) - 0.01 \cdot (-0.9, -0.6, -0.8) = (0.021, 0.014, 0.018).$$

5. **Iteración 3 (Tercer Minibatch):** Utilizando $w^{(2)} = (0.021, 0.014, 0.018)$, calculamos:

$$\nabla J_B(w^{(2)}) \approx (-0.6, -0.4, -0.5).$$

Actualización:

$$w^{(3)} = w^{(2)} - \eta \nabla J_B(w^{(2)}) = (0.021, 0.014, 0.018) - 0.01 \cdot (-0.6, -0.4, -0.5) = (0.027, 0.018, 0.023).$$

6. **Resultados Tabulados:**

Iteración	w_1	w_2	w_3
1	0.012	0.008	0.010
2	0.021	0.014	0.018
3	0.027	0.018	0.023

Table 4: Valores de los parámetros tras cada iteración.

Gráfica

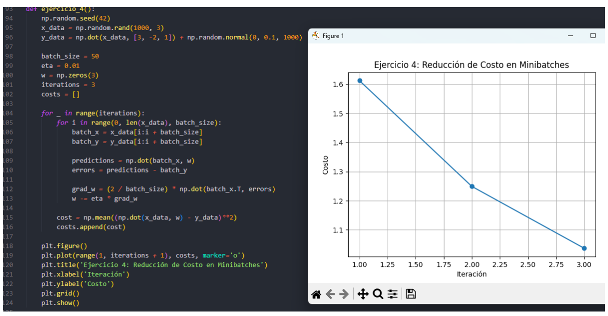


Figure 4: Ejercicio 4: Reducción de Costo en Minibatches

Conclusión

Los ejercicios resueltos ilustran cómo el descenso del gradiente permite minimizar funciones de costo en distintos escenarios. A través de iteraciones y ajustes en los parámetros, se logra una optimización progresiva, evidenciada en las gráficas incluidas.