Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: Fred Torres Cruz
Autor : Ronald Junior Pilco Nuñez

Trabajo Encargado - N° 007

Resolución de Ejercicios de Descenso del Gradiente

Github

Repositorio

Solución del Ejercicio 1

Enunciado

Minimizar la función cuadrática en una dimensión:

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

utilizando el método de **descenso del gradiente**. El punto inicial es $x_0 = 10$ y la tasa de aprendizaje es $\eta = 0.2$. Se pide realizar 5 iteraciones manualmente y anotar los valores de $(k, x_k, g(x_k))$ en una tabla. Además, explicar por qué el resultado tiende a $x = 5$.

Solución Paso a Paso

1. Cálculo del gradiente

La función es $g(x) = (x-5)^2$. El gradiente (derivada) de $g(x)$ con respecto a x es:

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

2. Fórmula de actualización

La actualización en el descenso del gradiente se realiza mediante:

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

Sustituyendo el gradiente y la tasa de aprendizaje $\eta = 0.2$:

$$x_{k+1} = x_k - 0.2 \cdot 2(x_k - 5) = x_k - 0.4(x_k - 5)$$

3. Iteraciones

Comenzamos con $x_0 = 10$. A continuación, realizamos las 5 iteraciones:

- **Iteración 1** ($k = 0$):

$$x_0 = 10, \quad g(x_0) = (10 - 5)^2 = 25$$

$$x_1 = 10 - 0.4(10 - 5) = 10 - 2 = 8$$

- **Iteración 2** ($k = 1$):

$$x_1 = 8, \quad g(x_1) = (8 - 5)^2 = 9$$

$$x_2 = 8 - 0.4(8 - 5) = 8 - 1.2 = 6.8$$

- **Iteración 3** ($k = 2$):

$$x_2 = 6.8, \quad g(x_2) = (6.8 - 5)^2 = 3.24$$

$$x_3 = 6.8 - 0.4(6.8 - 5) = 6.8 - 0.72 = 6.08$$

- **Iteración 4** ($k = 3$):

$$x_3 = 6.08, \quad g(x_3) = (6.08 - 5)^2 = 1.1664$$

$$x_4 = 6.08 - 0.4(6.08 - 5) = 6.08 - 0.432 = 5.648$$

- **Iteración 5** ($k = 4$):

$$x_4 = 5.648, \quad g(x_4) = (5.648 - 5)^2 = 0.4199$$

$$x_5 = 5.648 - 0.4(5.648 - 5) = 5.648 - 0.2592 = 5.3888$$

4. Tabla de Resultados

k	x_k	$g(x_k)$
0	10.0	25.0
1	8.0	9.0
2	6.8	3.24
3	6.08	1.1664
4	5.648	0.4199
5	5.3888	0.1513

Table 1: Resultados de las iteraciones del descenso del gradiente.

5. Explicación del Resultado

El resultado tiende a $x = 5$ porque este es el mínimo de la función $g(x) = (x-5)^2$. El descenso del gradiente busca minimizar la función moviéndose en la dirección opuesta al gradiente (derivada). En este caso, el gradiente es $2(x-5)$, que se anula cuando $x = 5$. Por lo tanto, el algoritmo converge hacia $x = 5$, que es el punto donde la función alcanza su valor mínimo.

Solución del Ejercicio 2

Enunciado

Ajustar una recta $h(x) = \beta_0 + \beta_1 x$ utilizando el método de **descenso del gradiente**. Se tienen los siguientes 5 puntos de entrenamiento:

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

La función de costo que se quiere minimizar es:

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

Se pide realizar al menos 3 iteraciones del descenso del gradiente con una tasa de aprendizaje $\eta = 0.01$, mostrando en cada iteración los nuevos valores de (β_0, β_1) y el costo J.

Solución Paso a Paso

1. Cálculo del gradiente

La función de costo es:

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

El gradiente de J con respecto a β_0 y β_1 es:

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

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

2. Fórmula de actualización

La actualización de los parámetros β_0 y β_1 en cada iteración se realiza mediante:

$$\beta_0^{(k+1)} = \beta_0^{(k)} - \eta \cdot \frac{\partial J}{\partial \beta_0}$$
$$\beta_1^{(k+1)} = \beta_1^{(k)} - \eta \cdot \frac{\partial J}{\partial \beta_1}$$

donde $\eta = 0.01$ es la tasa de aprendizaje.

3. Iteraciones

Comenzamos con valores iniciales $\beta_0^{(0)} = 0$ y $\beta_1^{(0)} = 0$. A continuación, realizamos las 3 iteraciones:

- **Iteración 1** $(k = 0)$:

$$\beta_0^{(0)} = 0, \quad \beta_1^{(0)} = 0$$
$$\frac{\partial J}{\partial \beta_0} = -2\sum_{i=1}^{5} y_i = -2 \cdot 18 = -36$$
$$\frac{\partial J}{\partial \beta_1} = -2\sum_{i=1}^{5} y_i x_i = -2 \cdot 61.9 = -123.8$$
$$\beta_0^{(1)} = 0 - 0.01 \cdot (-36) = 0.36$$
$$\beta_1^{(1)} = 0 - 0.01 \cdot (-123.8) = 1.238$$
$$J(\beta_0^{(1)}, \beta_1^{(1)}) = 3.1494$$

- **Iteración 2** $(k = 1)$:

$$\beta_0^{(1)} = 0.36, \quad \beta_1^{(1)} = 1.238$$
$$\frac{\partial J}{\partial \beta_0} = 4.74$$
$$\frac{\partial J}{\partial \beta_1} = 23.18$$
$$\beta_0^{(2)} = 0.36 - 0.01 \cdot 4.74 = 0.3126$$
$$\beta_1^{(2)} = 1.238 - 0.01 \cdot 23.18 = 1.0062$$
$$J(\beta_0^{(2)}, \beta_1^{(2)}) = 0.8475$$

- **Iteración 3** ($k = 2$):

$$\beta_0^{(2)} = 0.3126, \quad \beta_1^{(2)} = 1.0062$$

$$\frac{\partial J}{\partial \beta_0} = -2.688$$

$$\frac{\partial J}{\partial \beta_1} = -3.74$$

$$\beta_0^{(3)} = 0.3126 - 0.01 \cdot (-2.688) = 0.33948$$

$$\beta_1^{(3)} = 1.0062 - 0.01 \cdot (-3.74) = 1.0436$$

$$J(\beta_0^{(3)}, \beta_1^{(3)}) = 0.7457$$

4. Tabla de Resultados

Iteración k	$\beta_0^{(k)}$	$\beta_1^{(k)}$	Costo $J(\beta_0, \beta_1)$
0	0.0	0.0	18.0
1	0.36	1.238	3.1494
2	0.3126	1.0062	0.8475
3	0.33948	1.0436	0.7457

Table 2: Resultados de las iteraciones del descenso del gradiente.

5. Explicación del Resultado

El costo J disminuye en cada iteración, lo que indica que el descenso del gradiente está funcionando correctamente. Los valores de β_0 y β_1 se ajustan progresivamente para minimizar la función de costo. Si continuáramos con más iteraciones, el costo J seguiría disminuyendo, y los parámetros β_0 y β_1 convergerían a valores óptimos.

Solución del Ejercicio 3

Datos de Entrada

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

Modelo de Clasificación Logística

El modelo de clasificación logística se define como:

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

donde $\mathbf{w} = (w_0, w_1, w_2)$ son los pesos del modelo, y $\mathbf{x} = (1, x_1, x_2)$ es el vector de características con un término de sesgo 1.

Función de Costo Logístico

La función de costo para la clasificación logística es:

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

Gradiente de la Función de Costo

El gradiente de la función de costo con respecto a $\mathbf{w}$ es:

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

Inicialización

Comenzamos con $\mathbf{w}_0 = (0, 0, 0)$ y una tasa de aprendizaje $\eta = 0.1$.

Iteración 1

1. Calculamos $\sigma(\mathbf{w}_0^\top \mathbf{x}_i)$ para cada muestra: - $\sigma(0) = 0.5$ para todas las muestras.

2. Calculamos el gradiente:

$$\nabla J(\mathbf{w}_0) = \frac{1}{4} \sum_{i=1}^{4} (\sigma(0) - y_i)\mathbf{x}_i$$

$$\nabla J(\mathbf{w}_0) = \frac{1}{4} [(0.5 - 0)(1, 0.5, 1.0) + (0.5 - 0)(1, 1.5, 2.0) + (0.5 - 1)(1, 2.0, 2.5) + (0.5 -$$

$$\nabla J(\mathbf{w}_0) = \frac{1}{4} [(0.5, 0.25, 0.5) + (0.5, 0.75, 1.0) + (-0.5, -1.0, -1.25) + (-0.5, -1.5, -1.$$

$$\nabla J(\mathbf{w}_0) = \frac{1}{4}(0, -1.5, -1.5) = (0, -0.375, -0.375)$$

3. Actualizamos los pesos:

$$\mathbf{w}_1 = \mathbf{w}_0 - \eta \nabla J(\mathbf{w}_0) = (0, 0, 0) - 0.1(0, -0.375, -0.375) = (0, 0.0375, 0.0375)$$

4. Calculamos el nuevo valor de la función de costo $J(\mathbf{w}_1)$.

Iteración 2

Repetimos el proceso con $\mathbf{w}_1 = (0, 0.0375, 0.0375)$.

1. Calculamos $\sigma(\mathbf{w}_1^\top \mathbf{x}_i)$ para cada muestra.
2. Calculamos el gradiente $\nabla J(\mathbf{w}_1)$.
3. Actualizamos los pesos:

$$\mathbf{w}_2 = \mathbf{w}_1 - \eta \nabla J(\mathbf{w}_1)$$

4. Calculamos el nuevo valor de la función de costo $J(\mathbf{w}_2)$.

Iteración 3

Repetimos el proceso con $\mathbf{w}_2$.

1. Calculamos $\sigma(\mathbf{w}_2^\top \mathbf{x}_i)$ para cada muestra.
2. Calculamos el gradiente $\nabla J(\mathbf{w}_2)$.
3. Actualizamos los pesos:

$$\mathbf{w}_3 = \mathbf{w}_2 - \eta \nabla J(\mathbf{w}_2)$$

4. Calculamos el nuevo valor de la función de costo $J(\mathbf{w}_3)$.

Solución del Ejercicio 4

Datos de Entrada

Tenemos 1000 observaciones $\{(\mathbf{x}_i, y_i)\}_{i=1}^{1000}$ para un problema de regresión multivariable. Dividimos el conjunto en minibatches de tamaño 50.

Función de Costo

La función de costo es:

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

Gradiente de la Función de Costo

El gradiente de la función de costo con respecto a $\mathbf{w}$ es:

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

Procedimiento de SGD

1. **Inicialización:** Comenzamos con $\mathbf{w}_0 = (0, 0, \ldots, 0)$ y una tasa de aprendizaje $\eta = 0.01$.

2. **Iteración 1:** - Seleccionamos un minibatch de 50 datos. - Calculamos el gradiente sobre el minibatch:

$$\nabla J(\mathbf{w}_0) = -\frac{2}{50}\sum_{i=1}^{50}(y_i - \mathbf{w}_0^\top \mathbf{x}_i)\mathbf{x}_i$$

- Actualizamos los pesos:

$$\mathbf{w}_1 = \mathbf{w}_0 - \eta \nabla J(\mathbf{w}_0)$$

3. **Iteración 2:** - Seleccionamos otro minibatch de 50 datos. - Calculamos el gradiente sobre el nuevo minibatch. - Actualizamos los pesos:

$$\mathbf{w}_2 = \mathbf{w}_1 - \eta \nabla J(\mathbf{w}_1)$$

4. **Iteración 3:** - Repetimos el proceso con otro minibatch. - Actualizamos los pesos:

$$\mathbf{w}_3 = \mathbf{w}_2 - \eta \nabla J(\mathbf{w}_2)$$

Convergencia más rápida con SGD

El SGD converge más rápido en la práctica porque actualiza los pesos con cada minibatch, lo que permite una convergencia más rápida en comparación con el descenso por lotes completos (batch gradient descent), que requiere calcular el gradiente sobre todo el conjunto de datos en cada iteración. Además, el SGD puede escapar de mínimos locales más fácilmente debido a la naturaleza estocástica de las actualizaciones.

Ejemplo de Actualización de Parámetros

Supongamos que después de varias iteraciones, los parámetros $\mathbf{w}$ se actualizan de la siguiente manera:

- Después de la iteración 1: $\mathbf{w}_1 = (0.1, 0.2, \ldots, 0.3)$ - Después de la iteración 2: $\mathbf{w}_2 = (0.15, 0.25, \ldots, 0.35)$ - Después de la iteración 3: $\mathbf{w}_3 = (0.18, 0.28, \ldots, 0.38)$

Estas actualizaciones muestran cómo los parámetros $\mathbf{w}$ se ajustan iterativamente para minimizar la función de costo.