Macroentrenamiento en Inteligencia Artificial MeIA 2025

Módulo 1 - Inducción

Proyecto Final

Gómez Meza Jorge Ángel

Profesores:
Dr. Miguel Ángel Álvarez Carmona
Dr. Angel Ramón Aranda Campos
Dr. Ángel Díaz Pacheco
Grupo 1

Fecha de Entrega: 8 de junio 2025

1. Objetivo

El objetivo central de este proyecto es implementar desde cero el método de optimización del descenso del gradiente para resolver una tarea de regresión lineal. El modelo se entrenará y evaluará utilizando un subconjunto del conocido conjunto de datos Iris

2. Fundamentos del Modelo y Conceptos Clave

Antes de detallar el algoritmo, es importante entender la estructura del modelo y los conceptos para su implementación.

El modelo de regresión lineal busca predecir una variable dependiente (y) a partir de un conjunto de variables independientes (x_j). Como nuestro problema utiliza tres características, la forma funcional del modelo es:

$$\hat{y} = (w_1 \cdot x_1) + (w_2 \cdot x_2) + (w_3 \cdot x_3) + b$$

Donde:

- $\hat{y}$ es la predicción (longitud del pétalo).
- x_1, x_2, x_3 son las características de entrada: sepal_width, petal_width y sepal_length.
- w_1, w_2, w_3 son los **pesos** del modelo.
- b es el término de **sesgo** o intercepto

<u>Interpretación de Pesos (Wj) y Sesgo (b)</u>

Tenemos múltiples pesos, pero un solo sesgo.

- **Pesos (w_1, w_2, w_3):** Existe un peso por cada característica de entrada. Cada peso actúa indicando qué tan importante es esa característica específica para la predicción final. Un peso grande (positivo o negativo) significa que la característica tiene un gran impacto, mientras que un peso cercano a cero indica poco impacto.
- **Sesgo (b):** Existe un solo sesgo para todo el modelo. Este término, también llamado intercepto, representa el valor base de la predicción si todas las características de entrada fueran cero. Geométricamente, su función es permitir que el modelo se desplace verticalmente para ajustarse mejor al conjunto de datos.

Justificación de la Gráfica 3D: La "Línea de Ajuste" como un Hiperplano

En las instrucciones del proyecto se nos solicita graficar la "línea ajustada", aunque el término "línea" debería ser correcto, su representación gráfica depende del número de variables.

- Con una variable independiente, el modelo es una línea y se grafica en 2D.
- Con dos variables independientes, el modelo es un plano y se debe graficar en 3D.
- Con tres o más variables, como en este caso, el modelo es un hiperplano.

Para visualizar nuestro modelo de tres variables de entrada, la forma más directa es representar un plano en 3D, donde dos ejes corresponden a dos de las características, el tercer eje a la variable de salida, y el plano en sí representa las predicciones del modelo. Por esta razón, una gráfica 3D es la visualización más adecuada.

3. Explicación Paso a Paso del Algoritmo y Justificación

A continuación, se detalla el algoritmo de descenso del gradiente.

Paso 1: Inicialización de Parámetros

- "Sean b^0 y w_j^0 para ∀ j = 1, ..., m los valores iniciales. Estos pueden ser todos ceros (o valores aleatorios)".

- **Justificación:** El algoritmo comienza estableciendo los pesos y el sesgo en un punto de partida conocido. En nuestra implementación, se inicializan todos los parámetros en cero, lo que cumple directamente con esta instrucción.

Paso 2: Proceso Iterativo de Optimización

- "Repetir para k=0,...K" y "Calcular gradientes" y "Actualizar parámetros".

- **Justificación:** El "núcleo" por así decirlo del algoritmo es un bucle que se repite un número determinado de veces. En cada iteración, se realizan los siguientes sub-pasos para acercar el modelo a la solución óptima:

    1. **Cálculo de Predicciones:** Se calcula el valor predicho $\hat{y}$ para cada muestra utilizando la fórmula del modelo lineal: $\hat{y} = \sum_{j=1}^{m} w_j x_j + b$

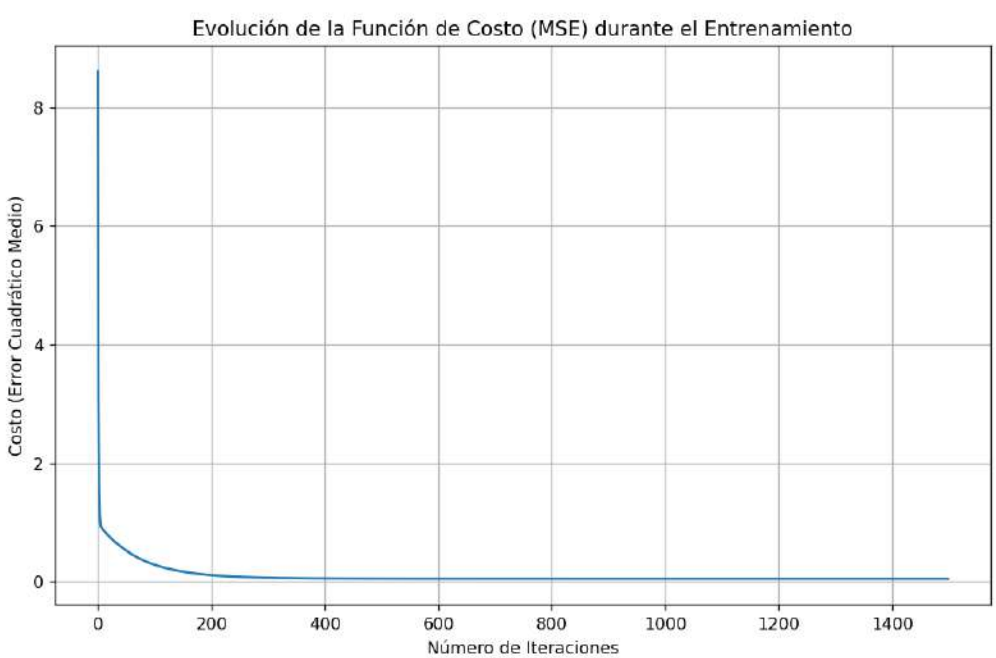
    2. **Cálculo del Costo:** Se evalúa qué tan erróneas son las predicciones actuales utilizando la función de costo del Error Cuadrático Medio (MSE): $f(W, b) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}^{(i)} - \hat{y}^{(i)})^2$. Este valor nos indica el rendimiento del modelo en la iteración actual.

3. **Cálculo de Gradientes:** Se calculan las derivadas parciales de la función de costo con respecto a cada peso (w_j) y al sesgo (b). Estas derivadas nos indican la dirección en la que el error aumenta más rápidamente.
4. **Actualización de Parámetros:** Se actualizan los parámetros moviéndolos en la dirección *opuesta* al gradiente, controlado por una tasa de aprendizaje (α): $b^{k+1} := b^k - \alpha\partial_{b^k}f$ y $w_j^{k+1} := w_j^k - \alpha\partial_{w_j^k}f$.
5. <u>Paso 3: Convergencia</u>

- "verificamos convergencia".
- **Justificación:** El proceso iterativo se detiene después de un número fijo de iteraciones o cuando el costo deja de disminuir significativamente. En ese punto, se considera que el modelo ha convergido y los parámetros encontrados son los óptimos.

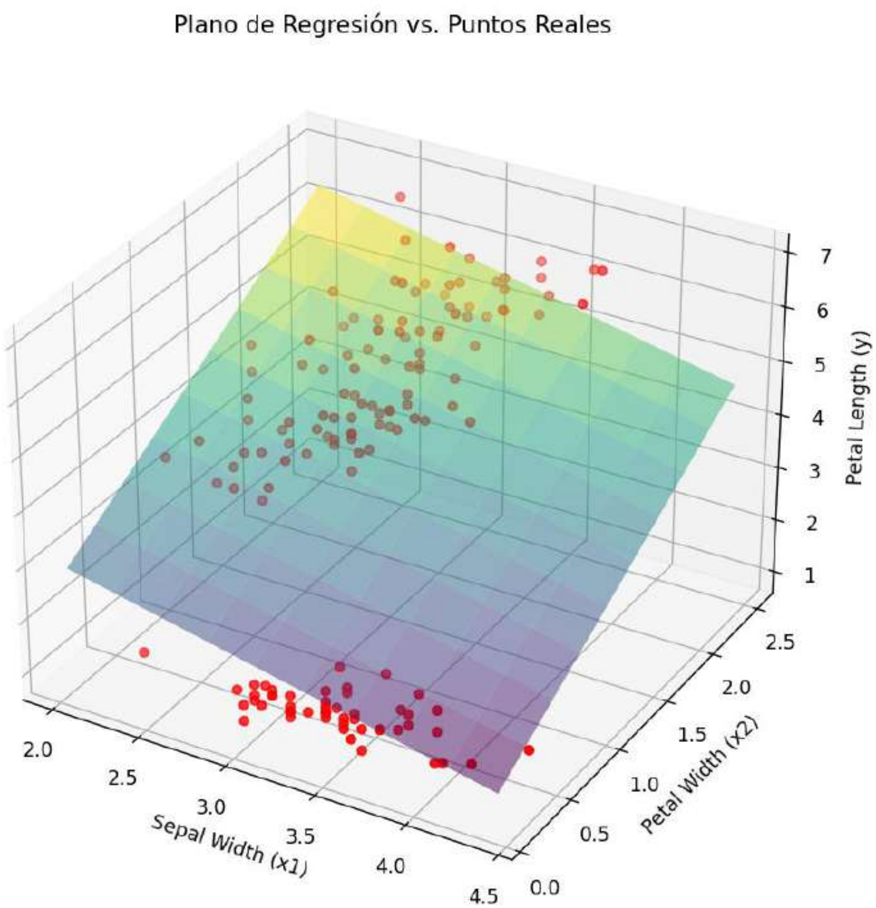
4. Resultados y Análisis

Gráfica 1: Evolución de la Función de Costo



- **Interpretación:** La curva muestra una rápida caída del error (costo) durante las primeras iteraciones, lo que indica un aprendizaje eficiente. Posteriormente, la curva se aplana, lo que significa que el modelo ha convergido a un punto de error mínimo. Esto confirma que el algoritmo funcionó correctamente.

Gráfica 2: Visualización del Ajuste del Modelo



Interpretación:

- Los Puntos Rojos (Datos Reales): Cada punto rojo representa una "flor" real del conjunto de datos. Su posición en el espacio está determinada por sus valores verdaderos de Sepal Width (eje x), Petal Width (eje y) y Petal Length (eje z). Estos puntos son la "verdad fundamental" que el modelo intenta predecir.
- El Plano Translúcido (El Modelo de Regresión): Esta superficie representa las predicciones del modelo. Para cualquier combinación de Sepal Width y Petal Width, la altura del plano en ese punto es el valor de Petal Length que el modelo predeciría. Es la representación visual de la ecuación matemática que el algoritmo de descenso del gradiente ha encontrado.

En conclusión, esta gráfica demuestra visualmente que el modelo ha aprendido una relación lineal coherente a partir de los datos. Muestra cómo el modelo generaliza una tendencia a partir de múltiples puntos de datos.

Análisis Comparativo de Resultados

```
PS C:\Users\jorgo> & C:/Users/jorgo/anaconda3/python.exe "c:/Facultad de Ingeniería UNAM/MelAI 2025/Gradiente_descendente.py"
--- Evolución de la Función de Costo ---

--- Gráfica del Ajuste del Modelo ---

--- Comparación de Resultados con scikit-learn ---
            Parámetro  Implementación Propia  scikit-learn
 Peso 1 (sepal_width)              -0.696557     -0.646012
 Peso 2 (petal_width)               1.424160      1.446793
Peso 3 (sepal_length)               0.734229      0.729138
         Sesgo (bias)              -0.109437     -0.262711
PS C:\Users\jorgo>
```

Finalmente, para validar nuestra implementación, se comparan los parámetros obtenidos con los del modelo LinearRegression de scikit-learn.

Interpretación de Resultados: Los pesos calculados por nuestra implementación (Implementación Propia) son similares a los calculados por la librería estándar (scikit-learn). Por ejemplo, el Peso 2 (petal_width) es 1.42 vs 1.44, y el Peso 3 (sepal_length) es 0.73 vs 0.72. Esta gran similitud valida que nuestro algoritmo es correcto y ha encontrado una solución casi idéntica a la del modelo de referencia.

5. Conclusión

El proyecto se completó con éxito. Se implementó correctamente el algoritmo de descenso del gradiente desde cero, logrando ajustar un modelo de regresión lineal a los datos del Iris. La validación a través de la curva de costo, la visualización del ajuste y la comparación con scikit-learn muestran que los objetivos del proyecto se cumplieron.