Apuntes de Clase Inteligencia Artificial IC-6200 Semana 3 | Jueves 23 de feb del 2024

Maynor Martínez Hernández
Profesor: Steven Pacheco
Tecnológico de Costa Rica

I. Repaso de Regresión Lineal

A. *Definición y Visualización*

La regresión lineal se centra en la colección de pares ordenados (x_i, y_i), donde x es un vector de tamaño d. Se construye un modelo de la forma

$$f_{(w,b)}(x) = wx + b \tag{1}$$

que representa una recta. Aquí, w es un vector de tamaño d que representa los pesos asociados a cada característica de x, y b es la intersección con el eje y, también conocido como el término de bias.

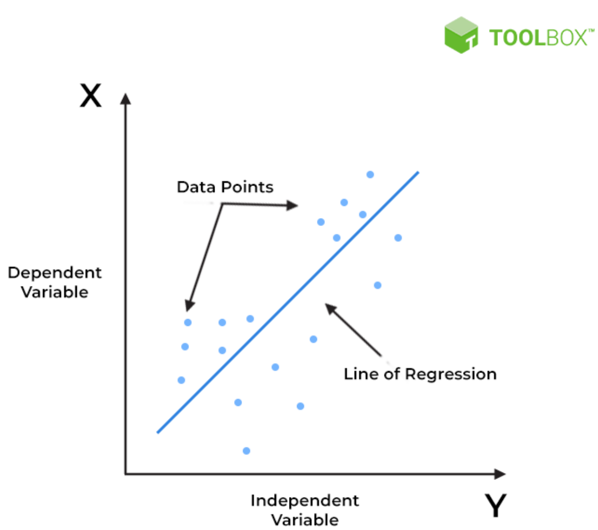

Fig. 1. Visualización de la regresión lineal con puntos de datos y la línea de regresión.

B. *Mean squared error (MSE)*

El MSE o se define como la media del cuadrado de las diferencias entre los valores observados y_i y los valores predichos por el modelo $f_{(w,b)}(x_i)$, para todos los puntos de datos i:

$$L = \frac{1}{n}\sum_{i=1}^{n}(f_{(w,b)}(x_i) - y_i)^2 \tag{2}$$

C. *Loos Functino*

La función de pérdida $(f_{(w,b)}(x_i) - y_i)^2$ representa la diferencia entre la predicción del modelo y la salida real, elevada al cuadrado. Este enfoque se utiliza tanto para la función de pérdida como para la función de costo, términos que a menudo se usan de manera intercambiable en este contexto.

Minimizar el MSE: El objetivo es minimizar el MSE para que las predicciones del modelo sean lo más precisas posible. Dado que el modelo está parametrizado por w y b, la precisión de la predicción dependerá de estos parámetros.

D. *Funciones Convexas y No Convexas*

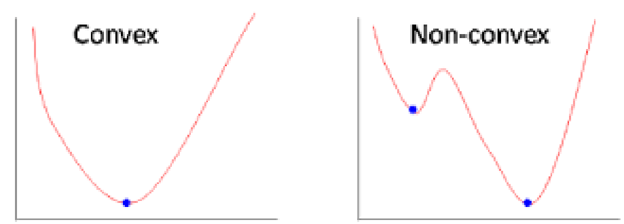

Fig. 2. Ejemplo de funciones convexas y no convexas.

Para optimizar el modelo, es crucial utilizar un método eficaz que pueda lidiar con posibles no convexidades en la función de costo. Aquí es donde el método del descenso del gradiente se vuelve relevante.

E. *Descenso del Gradiente*

El descenso del gradiente es una técnica de optimización que ajusta iterativamente los parámetros w y b para minimizar la función de costo.

MSE vs MAE: Se elige el MSE sobre otras medidas, como el error absoluto medio (MAE), debido a su diferenciabilidad, lo que facilita la aplicación de técnicas de optimización como el descenso del gradiente.

II. Resolución de Derivada

En el aprendizaje mediante regresión lineal, se ajustan los parámetros w y b para minimizar la función de pérdida L. Esto requiere el cálculo de los gradientes respecto a w y b. Para ello se necesita resolver la derivada y ajustar el erorr. A continuación, se presenta la derivada de la función de pérdida L con respecto a w y b.

A. Derivada Respecto a w

Dada la función de pérdida

$$L = \frac{1}{n}\sum_{i=1}^{n}(f_{(w,b)}(x_i) - y_i)^2, \tag{3}$$

donde $f_{(w,b)}(x_i) = wx_i + b$, la derivada parcial de L con respecto a w se calcula como sigue:

$$\frac{\partial L}{\partial w} = \frac{\partial}{\partial w}\left(\frac{1}{n}\sum_{i=1}^{n}(wx_i + b - y_i)^2\right) \tag{4}$$

$$= \frac{2}{n}\sum_{i=1}^{n}(wx_i + b - y_i) \cdot \frac{\partial}{\partial w}(wx_i + b - y_i) \tag{5}$$

$$= \frac{2}{n}\sum_{i=1}^{n}(wx_i + b - y_i) \cdot x_i. \tag{6}$$

B. Derivada Respecto a b

Al igual que con w, la derivación de la función de pérdida L con respecto al sesgo b es crucial para entender cómo ajustar b para minimizar el error. La derivada parcial de L con respecto a b se calcula de la siguiente manera:

Dada la función de pérdida

$$L = \frac{1}{n}\sum_{i=1}^{n}(f_{(w,b)}(x_i) - y_i)^2, \tag{7}$$

donde $f_{(w,b)}(x_i) = wx_i + b$, la derivada parcial de L con respecto a b se obtiene como:

$$\frac{\partial L}{\partial b} = \frac{\partial}{\partial b}\left(\frac{1}{n}\sum_{i=1}^{n}(wx_i + b - y_i)^2\right) \tag{8}$$

$$= \frac{2}{n}\sum_{i=1}^{n}(wx_i + b - y_i) \cdot \frac{\partial}{\partial b}(wx_i + b - y_i) \tag{9}$$

$$= \frac{2}{n}\sum_{i=1}^{n}(wx_i + b - y_i), \tag{10}$$

pues $\frac{\partial}{\partial b}(wx_i + b - y_i) = 1$ al ser b un término independiente de x_i.

Esta expresión nos proporciona el gradiente de la función de pérdida con respecto a b, que se utiliza en el algoritmo de descenso del gradiente para actualizar b en cada iteración mediante la regla de actualización:

C. Resultados de la Derivada y el Hiperparámetro α

Tras obtener las derivadas parciales de la función de pérdida con respecto a los parámetros w y b, se utilizan en el algoritmo de descenso del gradiente para actualizar dichos parámetros. Las actualizaciones se realizan según las siguientes reglas:

$$w := w - \alpha\frac{\partial L}{\partial w}, \tag{11}$$

$$b := b - \alpha\frac{\partial L}{\partial b}, \tag{12}$$

Nótese que α representa la tasa de aprendizaje (Learning Rate), un hiperparámetro que determina el tamaño del paso en las actualizaciones de w y b.

Este es seleccionado por el Machine Learning Engineering y no es aprendido por el modelo durante el entrenamiento. La selección de α se realiza generalmente mediante experimentación y ajuste fino.

Además, en el contexto de funciones no convexas, un α muy pequeño podría llevar al algoritmo a converger a un mínimo local en lugar del mínimo global. La selección óptima de α se busca iterativamente, equilibrando entre la eficiencia de la convergencia y la precisión de la solución encontrada.

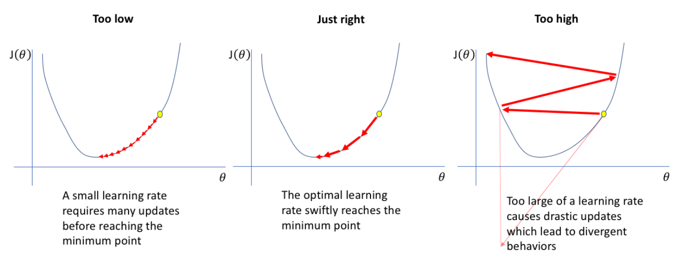

Fig. 3. Impacto de diferentes valores de α en la convergencia del algoritmo de descenso del gradiente.

III. Código Python para Regresión Lineal

Este código implementa las funciones básicas para realizar regresión lineal, incluyendo la definición del modelo, el cálculo de la derivada de la función de pérdida, y la actualización de los parámetros del modelo.

A. Función del Modelo

```
def modelo(x, w, b):
    # Calcula la regresion lineal basada en los parametros w y b con respecto a x.
    # Parametros:
    # x (array-like): Vector de caracteristicas.
    # w (array-like): Vector de pesos.
    # b (float): Bias (tambien conocido como sesgo).
    # Retorna:
    # array-like: Resultado de la regresion lineal.

    return w*x + b
```

B. Función de Pérdida - MSE

```
def MSE(y_pred, y):
    # Calcula el error cuadratico medio (MSE) entre y_pred y y.
    return (y_pred - y)**2
```

C. Actualización de Parámetros

```
# Inicializacion de parametros
w = np.random.default_rng().random()
b = np.random.default_rng().random()

# Tasa de aprendizaje
alfa = 0.0005

# Actualizacion de parametros mediante descenso del gradiente
```

```python
for epoch in range(epochs):
    derivadas_w = []
    derivadas_b = []
    for (x, y) in samples:
        y_pred = modelo(x, w, b)
        derivadaW = 2*x*(y_pred - y)
        derivadaB = 2*(y_pred - y)
        derivadas_w.append(derivadaW)
        derivadas_b.append(derivadaB)

    promedio_derivada_w = np.mean(derivadas_w)
    w = w - alfa*promedio_derivada_w
    promedio_derivada_b = np.mean(derivadas_b)
    b = b - alfa*promedio_derivada_b
```

D. *Observaciones*

Nótese, como hay varios puntos clave sobre la implementación y comprensión de la regresión lineal.

- **Ejemplo Visual:** Dicho código es un resmen del código mostrado por el profesor en la clase, por lo que se comprimen muchas funcionesy datos.
- **Selección de α:** El valor de α, la tasa de aprendizaje, fue elegido experimentalmente. En este caso, se utilizó un valor de 0.0005.
- **Inicialización de datos:** Se inició seleccionando un conjunto de samples para entrenar el modelo. Se asignaron valores aleatorios para los parámetros w y b.
- **Código para Graficar:** Se utilizó de matplotlib para visualizar los samples junto con la línea de regresión.
- **Cálculos de Errores:** Se calcularon los errores para cada sample utilizando la función de pérdida MSE (Mean Squared Error).
- **Actualización de Parámetros w y b:** Al finalizar el recorrido de todos los samples, se promediaron las derivadas y se actualizaron los parámetros. Este proceso se repite por cada época, ajustando w y b para minimizar el error.
- **Epochs:** Cada ciclo completo de actualización de parámetros se conoce como una época. Después de cada época, se visualizan los resultados para observar la convergencia del modelo hacia la solución óptima.

References

[1] "Understanding Machine Learning," YouTube, 2023. [Video]. Disponible en: https://www.youtube.com/watch?v=IHZwWFHWa-w.

[2] "Deep Learning Basics," YouTube, 2023. [Video]. Disponible en: https://www.youtube.com/watch?v=sDv4f4s2SB8.

[3] J. Jim, "Mean Squared Error (MSE) Explained," Statistics by Jim, 2023. [Online]. Disponible en: https://statisticsbyjim.com/regression/mean-squared-error-mse/.