Apuntes Semana 4 - 13/3/2025

Kenneth Chacón Rivas 2020054320

***Abstract*— Estos apuntes corresponden a la segunda clase de la Semana 4 del curso de Inteligencia Artificial, impartida el día 13 de marzo de 2025. Se abordan conceptos fundamentales de regresión lineal, incluyendo la construcción del modelo, la función de pérdida y su minimización mediante descenso del gradiente. Además, se repasaron reglas básicas de derivación, el cálculo de derivadas parciales y la actualización de parámetros. Finalmente, conceptos claves como tasa de aprendizaje, epochs, batch y técnicas de optimización.**

I. Introducción

Durante esta clase se profundizó en los fundamentos de la regresión lineal, un modelo ampliamente utilizado en inteligencia artificial para predecir valores continuos. Se presentó el enfoque matemático detrás del modelo, el papel de la función de pérdida y la importancia de su minimización. Asimismo, se introdujo el algoritmo de descenso del gradiente y su implementación mediante derivadas parciales. También se repasaron conceptos esenciales que permiten comprender y aplicar correctamente el proceso de entrenamiento, como la tasa de aprendizaje, las epochs, el uso de batches y las variantes del descenso del gradiente.

II. Noticias de la semana

- **Manus:** Es un agente de inteligencia artificial que realiza procesos cognitivos y ejecuta acciones de forma autónoma. Destaca por su capacidad de generar resultados concretos en diversas tareas tanto profesionales como cotidianas, minimizando la intervención humana.
- **Archon:** Es un agente de inteligencia artificial diseñado para generar otros agentes mediante técnicas avanzadas de programación y bases de conocimiento estructuradas, permitiendo así la automatización eficiente de procesos complejos.
- **OpenAI:** OpenAI lanzó herramientas que facilitan la creación de agentes de inteligencia artificial autónomos. Estas incluyen una nueva API (Responses) para búsqueda web, análisis de documentos y tareas computacionales, así como un SDK que permite gestionar múltiples agentes simultáneamente, ampliando la automatización de procesos complejos.

III. Regresión lineal

La regresión lineal es un método estadístico en el cual se tiene un conjunto de datos etiquetados:

$$\{(x_i, y_i)\}_{i=1}^{N}$$

Cada dato posee una etiqueta o variable dependiente y_i que es un valor real, y un vector de variables independientes x_i que influye sobre esta etiqueta. Se utiliza para realizar predicciones de valores continuos, estimando una relación lineal entre una variable dependiente y una o varias variables independientes.

Para realizar una regresión lineal, se debe construir un modelo definido como:

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

Donde:

- w vector de pesos (mide la influencia de la variable independiente sobre la dependiente)
- b es el sesgo (indica el valor esperado cuando la variable independiente es cero)
- x es el vector de variables independiente o de entrada.

Ejemplo de clase

A continuación, se presenta un ejemplo sencillo para ilustrar esta idea:

Horas de estudio (x)	**Calificación (y)**
1	50
2	55
3	65
4	70
5	75

Ejemplo de horas de estudio y calificaciones obtenidas.

El modelo de regresión lineal correspondiente a estos datos sería:

$$y = 5x + 45$$

Se observa que por cada hora adicional de estudio, la calificación incrementa en promedio 5 puntos, siendo 45 la calificación esperada sin horas de estudio.

IV. Minimización de la función de pérdida

A. *Diferencia entre Loss function y Cost function*

- **Loss function (función de pérdida):** mide el error de predicción para un único ejemplo o dato individual.
- **Cost function (función de costo):** representa el promedio de las funciones de pérdida para todo el conjunto de datos. Es decir, es una medida global del desempeño del modelo.

B. *Error Cuadrático Medio (MSE)*

Se busca minimizar la siguiente función:

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

Con el objetivo de lograr mayor precisión, minimizando el error que tiene el modelo en todos los samples. Un valor alto de L indica que el modelo se equivocó muchas veces en las etiquetas que debía predecir.

C. *Funciones convexas y no convexas*

Una función **convexa** es aquella que posee un único punto mínimo global. Por otro lado, una función **no convexa** puede presentar múltiples mínimos locales, además del mínimo global.

V. Descenso del gradiente

El descenso del gradiente es un algoritmo de optimización que utiliza derivadas de la función L, para encontrar el mínimo de una función. En cada paso, la derivada indica hacia dónde ajustar los parámetros para disminuir el error. Esto se repite hasta llegar al menor valor posible.

Importante solo se puede aplicar este algoritmo siempre y cuando la función que representa al modelo ($f_{w,b}$) y la función de costo sean derivables.

VI. Repaso de derivadas

- **Derivada de una constante:**

$$f(x) = k \quad \Rightarrow \quad f'(x) = 0$$

donde k es constante.

- **Derivada de la función identidad:**

$$f(x) = x \quad \Rightarrow \quad f'(x) = 1$$

- **Derivada de una constante por una función:**

$$f(x) = kx \quad \Rightarrow \quad f'(x) = k$$

- **Derivada de una potencia:**

$$f(x) = x^n \quad \Rightarrow \quad f'(x) = n \cdot x^{n-1}$$

- **Derivada de una suma de funciones:**

$$f(x) = u(x) + v(x) \quad \Rightarrow \quad f'(x) = u'(x) + v'(x)$$

- **Derivada del producto de funciones:**

$$f(x) = u(x) \cdot v(x)$$

$$\Downarrow$$

$$f'(x) = u'(x) \cdot v(x) + u(x) \cdot v'(x)$$

Derivadas parciales:

Las derivadas parciales se utilizan cuando una función depende de dos o más variables. Para calcular la derivada parcial con respecto a una variable, se considera constante a las demás.

Ejemplo:

$$f(x, y) = 4x + 5y \quad \Rightarrow \quad \frac{\partial f}{\partial x} = 4, \quad \frac{\partial f}{\partial y} = 5$$

Cálculo de derivadas parciales de L:

$$L = \frac{1}{N} \sum_{i=1}^{N} (wx_i + b - y_i)^2$$

Derivada de L con respecto a w

$$\frac{\partial L}{\partial w} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial w} (wx_i + b - y_i)^2$$

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

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

Derivada de L con respecto a b

$$\frac{\partial L}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial b} (wx_i + b - y_i)^2$$

$$= \frac{1}{N} \sum_{i=1}^{N} 2(wx_i + b - y_i) \cdot 1$$

$$= \frac{2}{N} \sum_{i=1}^{N} (wx_i + b - y_i)$$

VII. Actualización de parámetros

Con las derivadas parciales de L respecto a w y b ya calculadas, estas se utilizan directamente en las siguientes fórmulas para actualizar los parámetros:

$$w = w - \alpha \frac{\partial L}{\partial w}$$

$$b = b - \alpha \frac{\partial L}{\partial b}$$

donde α es un hiperparámetro que controla la magnitud del ajuste aplicado a los parámetros.

Estas expresiones permiten ajustar los valores de w y b con base en los resultados de las derivadas anteriores, lo que nos lleva a reducir el valor de L progresivamente.

VIII. Elección de la tasa de aprendizaje

Dado que α es un hiperparámetro, somos nosotros quienes debemos decidir qué valor asignarle. La elección de esta tasa de aprendizaje tiene un impacto directo en el proceso de optimización.

α **muy pequeño:** Los ajustes de los parámetros serán muy leves, lo que implica que el modelo deberá realizar muchas iteraciones para acercarse al valor óptimo.

α **muy grande:** Los pasos serán tan amplios que puede que el modelo no logre optimizar los parámetros correctamente, e incluso se aleje del mínimo deseado.

Ideal: Lo más recomendable es buscar un valor adecuado de α probando distintas opciones. Una técnica útil consiste en comenzar con un α relativamente alto y reducirlo gradualmente a medida que avanza el entrenamiento. Esto permite que el modelo se optimice rápidamente en las primeras etapas y, luego, ajuste con mayor precisión conforme se acerca al mínimo.

IX. Conceptos importantes

Epoch:

Una **epoch** es una iteración completa sobre todos los datos del conjunto de entrenamiento. Es decir, se utilizan todos los ejemplos (samples) una vez para actualizar los parámetros del modelo.

Las epochs son un hiperparámetro que podemos definir. Ejemplo: Si tenemos 10,000 samples y elegimos 5 epochs, el modelo verá esos 10,000 datos cinco veces es decir 50,000 datos.

Batch:

Un **batch** es un subconjunto del total de datos de entrenamiento que se toman para calcular el gradiente.

El tamaño del batch también es un hiperparámetro que podemos definir. Ejemplo: Si tenemos 10,000 samples y usamos un batch de 1,000, necesitaremos completar 10 batches para procesar todos los datos, es decir, para completar una epoch.

Existen tres técnicas principales para aplicar descenso del gradiente, que se diferencian según la cantidad de datos que se usan para calcular cada actualización:

1. Batch Gradient Descent: Utiliza todos los datos de entrenamiento para calcular el gradiente y actualizar los parámetros. Es preciso, pero puede ser lento cuando hay muchos datos.

2. Stochastic Gradient Descent (SGD): Actualiza los parámetros utilizando solo un dato (sample) a la vez. Es más rápido, pero introduce más variación en el proceso de optimización.

3. Mini-Batch Gradient Descent: Combina las dos anteriores. Usa pequeños grupos de datos (batches) para actualizar los parámetros. Requiere un hiperparametro referente a que tamaño de batch se va utilizar.

X. Conclusión

La clase permitió entender cómo la regresión lineal puede ser utilizada como herramienta predictiva al ajustar sus parámetros mediante descenso del gradiente. Se evidenció la importancia de calcular correctamente las derivadas parciales de la función de pérdida y cómo estas influyen directamente en la actualización de los parámetros del modelo. Además, se analizó cómo la elección adecuada de hiperparámetros como la tasa de aprendizaje, las epochs y el tamaño de batch afecta el proceso de optimización. Estos conceptos son fundamentales para el diseño y entrenamiento de modelos de inteligencia artificial efectivos.