

Elementos de programación diferenciable

Alejandro Sánchez Yalí

1. ¿Qué es la programación diferenciable?

Un **programa de computadora** consiste en una serie de instrucciones básicas que efectúan una tarea específica. En el ámbito de las ciencias computacionales, estos programas son creados por **programadores**. Sin embargo, existen tareas, especialmente aquellas que involucran patrones intrincados y decisiones complejas, como el reconocimiento de imágenes o la generación de texto, donde escribir un programa tradicional resulta extremadamente difícil, sino imposible.

En contraste, las redes neuronales modernas ofrecen diferentes enfoques. Estas se construyen mediante la combinación de bloques de funciones parametrizadas y se entrenan directamente de los datos usando optimización basada en el **gradiente**. Durante este proceso de entrenamiento, las redes neuronales aprenden simultáneamente la extracción de características y la ejecución de tareas, lo que les permite desarrollar tareas complejas que antes se consideraban inalcanzables para los programas tradicionales. Este nuevo paradigma de programación ha sido denominado como «programación diferenciable» o «software 2.0», términos que han sido popularizados por LeCun (2018) y Karpathy (2017).

Definición 1.1. Programación diferenciable es un paradigma de programación donde los programas (incluyendo flujos de control y estructuras de datos) pueden ser derivados automáticamente, permitiendo la optimización de parámetros basada en gradiente.

1.1. Redes neuronales modernas como programas parametrizados

En programación diferenciable, un programa de computadora clásico se puede definir como la composición de operaciones elementales, formando un **grafo computacional**. La diferencia clave es que los programas (como las redes neuronales) contienen parámetros que pueden ser ajustados a partir de los datos y pueden ser derivados usando la derivación automática (*autodiff*). Típicamente, se asume que los programas definen funciones matemáticamente válidas: la función debe retornar valores idénticos para argumentos idénticos y no debe tener efectos colaterales. Además, la función debe tener derivadas bien definidas, asegurando que se pueda usar la optimización basada en el algoritmo del gradiente. Por lo tanto, la programación diferenciable no es solo el arte de derivar a través de programas, sino también de diseñarlos cuidadosamente.

1.2. ¿Por qué las derivadas son importantes?

El aprendizaje computacional típicamente se reduce a la optimización de una función objetivo determinada, que es la composición de una función de pérdida y una función del modelo (red neuronal). La optimización sin derivadas se denomina **optimización de orden cero**. En este caso, solo se asume que podemos evaluar la función objetivo que deseamos optimizar. Lamentablemente, este método sufre de la maldición de la dimensionalidad, es decir, solo es viable para problemas de baja dimensión, con menos de 10 dimensiones. La optimización basada en derivadas, por otro lado, es mucho más eficiente y puede escalar a millones o miles de millones de parámetros. Los algoritmos que utilizan primeras y segundas derivadas se conocen, respectivamente, como algoritmos de **primer orden** y **segundo orden**.

1.3. ¿Por qué la diferenciación automática es importante?