

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 entrena 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). **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?

Antes de la revolución de la diferenciación automática, investigadores y practicantes necesitaban implementar manualmente el gradiente de las funciones que deseaban optimizar. Calcular gradientes manualmente podía convertirse en algo tedioso para funciones complicadas. Además, cada vez que la función era modificada (por ejemplo, para probar una nueva idea), el gradiente necesitaba ser recalculado. La diferenciación automática representa un cambio radical porque permite a los usuarios enfocarse en la experimentación creativa con funciones para sus tareas específicas.

1.4. Programación diferenciable no es solo aprendizaje computacional

Aunque existe un solapamiento entre el aprendizaje computacional y la programación diferenciable, sus enfoques son diferentes. El aprendizaje computacional estudia las redes neuronales compuestas de múltiples capas, que les permiten aprender **representaciones intermedias** de los datos. Por ejemplo, las redes neuronales convolucionales están diseñadas para el procesamiento de imágenes, mientras que las redes recurrentes están diseñadas para secuencias. Por otro lado, la programación diferenciable estudia las técnicas para diseñar programas complejos y diferenciables. Su uso va más allá del aprendizaje computacional: por ejemplo, en el aprendizaje por refuerzo, la programación probabilística y la computación científica en general.

1.5. Programación diferenciable no es solo diferenciación automatizada

Si bien la diferenciación automatizada es un ingrediente clave de la programación diferenciable, no es el único. La programación diferenciable también está comprometida con el diseño de operaciones que sean diferenciables en principio. De hecho, gran parte de la investigación sobre programación diferenciable se ha dedicado a hacer que las operaciones de la programación computacional clásica sean compatibles

con la programación diferenciable. Cabe destacar que muchas relajaciones diferenciables pueden interpretarse en un marco probabilístico. El tema central de este libro es la interacción entre la optimización, la probabilidad y la diferenciación. La diferenciación es útil para la optimización y, recíprocamente, la optimización puede ser útil para diseñar operadores diferenciables.

2. Fundamentos

En este capítulo revisaremos los conceptos clave de la diferenciación. En particular, enfatizaremos el papel fundamental que juegan las transformaciones lineales.

2.1. Funciones univariadas

2.1.1. Derivadas

Para estudiar funciones, así como sus derivadas, necesitamos capturar sus variaciones infinitesimales alrededor de los puntos tal como se ha definido para la noción de límite.

Definición 2.1 (Límite). Sea $f : \mathbb{R} \mapsto \mathbb{R}$ una función y sean $x_0, c \in \mathbb{R}$. Diremos que c es el límite de f cuando x tiende a x_0 si para todo $\varepsilon > 0$, existe $\delta > 0$ tal que para todo $x \in \mathbb{R}$ que satisface $0 < |x - x_0| < \delta$, se cumple que $|f(x) - c| < \varepsilon$. En este caso escribimos:

$$\lim_{x \rightarrow x_0} f(x) = c.$$

Los límites se preservan bajo operaciones algebraicas básicas. En efecto, si tenemos $f, g : \mathbb{R} \rightarrow \mathbb{R}$ y suponemos que existen los límites

$$\lim_{x \rightarrow x_0} f(x) = c \quad \text{y} \quad \lim_{x \rightarrow x_0} g(x) = d.$$

Entonces, para cualesquiera $a, b \in \mathbb{R}$:

1. **Linealidad:** Si definimos $(af + bg)(x) := af(x) + bg(x)$, entonces

$$\lim_{x \rightarrow x_0} (af + bg)(x) = ac + bd$$

2. **Multiplicación:** Si definimos $(fg)(x) := f(x)g(x)$, entonces

$$\lim_{x \rightarrow x_0} (fg)(x) = cd.$$

Con la noción de límite, podemos definir la clase de funciones que presentan un comportamiento regular, es decir, aquellas funciones donde el límite en cualquier punto coincide con el valor de la función evaluada en ese punto. Esta propiedad fundamental se conoce como **continuidad**.

Definición 2.2 (Funciones continuas). Una función $f : \mathbb{R} \rightarrow \mathbb{R}$ es continua en un punto $x_0 \in \mathbb{R}$ si

$$\lim_{x \rightarrow x_0} f(x) = f(x_0).$$

Además, diremos que f es continua (globalmente) si es continua en todo punto $x_0 \in \mathbb{R}$.

Aunque la noción de continuidad parece una suposición benigna, varias funciones sencillas, como la función escalón de Heavyside, no son continuas y requieren un tratamiento especial.

Observación (Notación de Landau). A lo largo de este texto usaremos la notación o pequeña de Landau. Para dos funciones $f, g : \mathbb{R} \rightarrow \mathbb{R}$, escribiremos $g(x) = o(f(x))$ cuando $x \rightarrow x_0$ si

$$\lim_{x \rightarrow x_0} \frac{|g(x)|}{|f(x)|} = 0.$$

Intuitivamente, esto significa que g es asintóticamente dominada por f cuando $x \rightarrow x_0$. Como caso particular, podemos caracterizar la continuidad de una función f en un punto x_0 mediante esta notación: f es continua en x_0 si y solo si

$$f(x_0 + h) = f(x_0) + o(1) \quad \text{cuando } h \rightarrow 0.$$

Consideremos ahora una función $f : \mathbb{R} \rightarrow \mathbb{R}$. Su valor en un intervalo $[w_0, w_0 + h]$ puede ser aproximado por la secante entre los puntos $(w_0, f(w_0))$ y $(w_0 + h, f(w_0 + h))$ como una función lineal con pendiente $(f(w_0 + h) - f(w_0))/h$. En el límite cuando la variación h tiende a cero alrededor de w_0 , la secante converge a la **recta tangente** de f en w_0 , y su pendiente se define como la derivada de f en w_0 . La siguiente definición formaliza esta intuición.

Definición 2.3 (Derivada). La derivada de una función $f : \mathbb{R} \rightarrow \mathbb{R}$ en un punto $w_0 \in \mathbb{R}$ se define como

$$f'(w_0) := \lim_{h \rightarrow 0} \frac{f(w_0 + h) - f(w_0)}{h},$$

siempre que este límite exista. En tal caso, diremos que f es **diferenciable** en w_0 .

Si f es diferenciable en cualquier $w \in \mathbb{R}$, diremos que es **diferenciable en todas partes**. Si f es diferenciable en un w_0 dado, entonces es necesariamente continua en w_0 como veremos en la siguiente proposición. Sin embargo continuidad no implica diferenciable por es ilustrado por la función de Kink.

Teorema 2.1 (Diferenciabilidad implica continuidad). *Si $f : \mathbb{R} \rightarrow \mathbb{R}$ es diferenciable en $x_0 \in \mathbb{R}$, entonces es continua en $x_0 \in \mathbb{R}$.*

Demostración. Como f es diferenciable en x_0 , existe el límite

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

Por lo tanto, podemos escribir la diferencia como

$$f(x_0 + h) - f(x_0) = f'(x_0)h + r(h),$$

donde $r(h)$ es un término residual que satisface $r(h) = o(h)$ cuando $h \rightarrow 0$. En particular,

$$\lim_{h \rightarrow 0} [f(x_0 + h) - f(x_0)] = \lim_{h \rightarrow 0} [f'(x_0)h + r(h)] = 0,$$

ya que $\lim_{h \rightarrow 0} h = 0$ y $\lim_{h \rightarrow 0} r(h) = 0$. Por lo tanto,

$$\lim_{h \rightarrow 0} f(x_0 + h) = f(x_0),$$

es decir, f es continua en x_0 . □