Métodos Numéricos para Encontrar Mínimos en Funciones Multivariables

D. B. García Alfaro
Universidad Centroamericana
00088023@uca.edu.sv

G. E. Iraheta Guardado
Universidad Centroamericana
00021223@uca.edu.sv

M. A. Montes Varela
Universidad Centroamericana
00042823@uca.edu.sv

***Abstract*—Este trabajo presenta un análisis detallado de dos métodos numéricos fundamentales para la optimización de funciones multivariables: el descenso de gradiente y el método de Newton-Raphson multivariable. Se exploran sus fundamentos matemáticos, condiciones de estabilidad y convergencia, así como sus ventajas y limitaciones.**

A través de un ejemplo práctico de ajuste polinomial a la función seno, se ilustra la aplicación de ambos métodos, incluyendo variantes como el descenso con momento y el uso de amortiguamiento en Newton-Raphson.

El estudio concluye con una comparación entre ambos enfoques, proporcionando criterios para seleccionar el método más adecuado según la naturaleza del problema de optimización.

***Index Terms*—mínimos locales, gradiente, matriz Hessiana, orden de convergencia**

I. Introducción

En el ámbito del Machine Learning y la optimización numérica, un problema recurrente es la minimización de funciones de costo, como el ajuste de pesos en una red neuronal para reducir el error de predicción. Este tipo de problemas suele representarse mediante funciones multivariables, donde el objetivo es encontrar parámetros que conduzcan a un mínimo local suficientemente óptimo. Dada la complejidad analítica de resolver estos problemas de manera exacta, se han desarrollado métodos numéricos iterativos, entre los que destacan el descenso de gradiente y el método de Newton-Raphson multivariable, cada uno con ventajas y limitaciones en términos de eficiencia y convergencia.

El descenso de gradiente (Gradient Descent) es un algoritmo iterativo de primer orden que ajusta los parámetros en la dirección opuesta al gradiente de la función de costo. Su simplicidad lo hace ampliamente utilizado y fácilmente escalable, incluso en problemas de alta dimensionalidad. Sin embargo, su convergencia puede ser lenta, especialmente en funciones no convexas o con variaciones abruptas en su curvatura, donde oscilaciones o estancamientos en regiones planas son comunes.

Por su parte, el método de Newton-Raphson multivariable ofrece una convergencia más rápida al utilizar información de segundo orden a través de la matriz Hessiana. No obstante, su implementación es computacionalmente costosa debido al cálculo y almacenamiento de derivadas segundas, y su eficacia depende críticamente de una inicialización cercana al óptimo, ya que en casos desfavorables puede divergir.

Este documento explora ambos métodos en profundidad: se presenta su fundamento matemático, las condiciones que garantizan estabilidad y convergencia, así como ejemplos ilustrativos de su aplicación. Finalmente, se realiza un análisis comparativo que resalta sus ventajas, desventajas y casos de uso recomendados, proporcionando así un marco de referencia para seleccionar el método más adecuado según el problema de optimización abordado.

II. Descenso de Gradiente

A. Descripción del método

El método de descenso del gradiente es un método para encontrar mínimos locales de una función f, que puede ser una función multivariable. La forma en que se encuentra el mínimo local es empezar en un punto ω_0, evaluar $\nabla f(\omega_0)$, y restar a ω_0 el resultado del gradiente multiplicado por un factor α llamada "tasa de aprendizaje", esto para obtener el siguiente punto ω_1.

Extrapolando, se obtiene la siguiente expresión iterativa:

$$\omega_{k+1} = \omega_k - \alpha \cdot \nabla f(\omega_k)$$

Para entender cómo funciona el método, es necesario recordar lo que es el gradiente. El gradiente de una función multivariable en un punto ω, en notación matemática $\nabla f(\omega)$ es un vector cuya magnitud es la tasa de cambio máxima en ω, y la dirección a la que apunta es hacia donde ocurre el mayor crecimiento en la función; y es por ello que $-\nabla f(\omega)$ es donde ocurre la tasa de mayor decrecimiento, pues es en dirección opuesta a $\nabla f(\omega)$.

Al trabajar con descenso de gradiente, evaluamos ∇f en un punto inicial ω_k para conocer cuál es la dirección de mayor crecimiento, y por ende, la de mayor decrecimiento, y es por eso que ω_k es restado un factor α del gradiente obtenido, para caminar un paso hacia donde la función está decreciendo; llegar a ω_{k+1}, y recalcular la dirección de mayor decrecimiento en dicho punto. Así, iterativamente, se llegará a un mínimo local.

El algoritmo a seguir para el descenso de gradiente se describe por con el siguiente conjunto de pasos:

1) Se inicializa el método en $k = 0$ considerando un punto inicial ω_0.
2) Calcular el gradiente de la función en el punto actual $\nabla f(\omega_k)$.

3) Actualizar la aproximación del mínimo óptimo con la fórmula iterativa del método, considerando la tasa de aprendizaje α asignada

$$\omega_{k+1} = \omega_k - \alpha \cdot \nabla f(\omega_k)$$

4) Verificar alguno de los criterios de convergencia

$$||\nabla f(\omega_k)|| < \varepsilon \quad o \quad ||\omega_{k+1} - \omega_k|| < \varepsilon$$

5) Se incrementa k.

Se repiten los pasos del 2 al 5 hasta haber cumplido con un criterio de convergencia o haber alcanzado un máximo de iteraciones, retornando un ω_k como solución aproximada, es decir, como el vector de parámetros óptimo que minimiza $f(\omega)$.

B. Formulación matemática

Supongamos una función escalar $f(\omega)$ diferenciable. Se sabe que el valor mínimo local ocurre en un punto donde la derivada de la función es igual a cero: $f'(\omega) = 0$.

Pero existen funciones demasiado complejas como para encontrar el mínimo local de forma analítica, por ello, utilizamos un método iterativo.

Lo que se pretende es minimizar la función $f(\omega)$, por lo que se deben tomar pasos en la dirección contraria a la derivada, es decir, en $-\nabla f(\omega)$. Esta idea se entiende mejor con funciones multivariables y su gradiente.

Para una función multivariable $f(\omega)$ donde $\omega \in \mathbb{R}^n$, el gradiente es

$$\nabla f(\omega) = \begin{bmatrix} \frac{\partial f}{\partial \omega_0} \\ \frac{\partial f}{\partial \omega_1} \\ \vdots \\ \frac{\partial f}{\partial \omega_{n-1}} \end{bmatrix}$$

Por lo que el paso de gradiente descendente se convierte en la función iterativa:

$$\omega_{k+1} = \omega_k - \alpha \cdot \nabla f(\omega_k) \tag{1}$$

Siendo α la tasa de aprendizaje que define el tamaño de los pasos en cada iteración hasta la convergencia del método en el punto ω mínimo, parando el algoritmo hasta que el gradiente es lo suficientemente cercano a cero o la diferencia entre las iteraciones presenta cambios muy pequeños, es decir, un error de aproximación menor que una tolerancia.

Como justificación geométrica, sabemos que el gradiente indica la dirección de mayor crecimiento de una función, es decir, que si se pretende minimizar la función, se debe avanzar en la dirección contraria al gradiente, en la de mayor descenso.

C. Análisis de convergencia

El método del descenso de gradiente posee un orden de convergencia que puede variar dependiendo de las propiedades de la función f.

1) Convergencia lineal: Para este caso se asume que la función f es convexa, diferenciable y satisface la condición de Lipschitz, con constante L, es decir:

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|x - y\|^2 \tag{2}$$

Además, se considera un paso α tal que $0 < \alpha < \frac{1}{L}$, y se asume que x^* es un mínimo global de f.

El método iterativo se define por:

$$x_{n+1} = x_n - \alpha \nabla f(x_n) \tag{3}$$

Sustituyendo $y = x_{n+1}$ y $x = x_n$ en (2):

$$y - x = -\alpha \nabla f(x_n) \tag{4}$$

$$\|y - x\|^2 = \alpha^2 \|\nabla f(x_n)\|^2 \tag{5}$$

Además, el producto escalar se evalúa como:

$$\nabla f(x_n)^\top (y - x) = -\alpha \|\nabla f(x_n)\|^2 \tag{6}$$

Sustituyendo (4), (5) y (6) en (2):

$$f(x_{n+1}) \leq f(x_n) - \alpha \|\nabla f(x_n)\|^2 + \frac{L\alpha^2}{2}\|\nabla f(x_n)\|^2 \tag{7}$$

Factorizando:

$$f(x_{n+1}) \leq f(x_n) - \left(\alpha - \frac{L\alpha^2}{2}\right) \|\nabla f(x_n)\|^2 \tag{8}$$

Dado que $\alpha \leq \frac{1}{L}$, se tiene que $\alpha - \frac{L\alpha^2}{2} \geq \frac{\alpha}{2}$, por lo que:

$$f(x_{n+1}) \leq f(x_n) - \frac{\alpha}{2}\|\nabla f(x_n)\|^2 \tag{9}$$

Como f es convexa, se cumple que:

$$f(x_n) - f(x^*) \leq \nabla f(x_n)^\top (x_n - x^*) \tag{10}$$

Aplicando la desigualdad de Cauchy-Schwarz:

$$f(x_n) - f(x^*) \leq \|\nabla f(x_n)\| \cdot \|x_n - x^*\| \tag{11}$$

También, bajo la hipótesis de Lipschitz para el gradiente, se tiene:

$$\|\nabla f(x_n)\|^2 \geq 2L(f(x_n) - f(x^*)) \tag{12}$$

Sustituyendo (12) en (9):

$$f(x_{n+1}) \leq f(x_n) - \alpha L(f(x_n) - f(x^*)) \tag{13}$$

Definiendo el error $\varepsilon_n = f(x_n) - f(x^*)$, se deduce:

$$\varepsilon_{n+1} \leq (1 - \alpha L)\varepsilon_n \tag{14}$$

Por lo tanto, se demuestra convergencia lineal:

$$\boxed{\frac{|\varepsilon_{n+1}|}{|\varepsilon_n|} \leq 1 - \alpha L} \tag{15}$$

2) Convergencia sublineal: Si f no es fuertemente convexa (es decir, no posee un mínimo absoluto estrictamente definido), la rapidez de convergencia disminuye. Utilizando sumas telescópicas a partir de (9), sumando desde $n = 0$ hasta $N - 1$:

$$\sum_{n=0}^{N-1} f(x_n) - f(x_{n+1}) \geq \frac{\alpha}{2} \sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2 \tag{16}$$

Esta suma telescópica se reduce a:

$$f(x_0) - f(x_N) \geq \frac{\alpha}{2} \sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2 \tag{17}$$

Además:

$$\sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2 \leq \frac{2}{\alpha}(f(x_0) - f(x^*)) \tag{18}$$

Y como:

$$N \cdot \min_n \|\nabla f(x_n)\|^2 \leq \sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2 \tag{19}$$

Se concluye:

$$\min_n \|\nabla f(x_n)\|^2 \leq \frac{2}{\alpha N}(f(x_0) - f(x^*)) \tag{20}$$

Sabemos que para funciones convexas:

$$f(x_n) - f(x^*) \leq \|\nabla f(x_n)\| \cdot \|x_n - x^*\| \leq R\|\nabla f(x_n)\| \tag{21}$$

donde $R = \max_n \|x_n - x^*\| \leq \|x_0 - x^*\|$. Elevando ambos lados al cuadrado:

$$(f(x_n) - f(x^*))^2 \leq R^2 \|\nabla f(x_n)\|^2 \tag{22}$$

Usando (20):

$$\min_n (f(x_n) - f(x^*))^2 \leq \frac{2R^2}{\alpha N}(f(x_0) - f(x^*)) \tag{23}$$

Extrayendo la raíz cuadrada:

$$\min_n (f(x_n) - f(x^*)) \leq \frac{R}{\sqrt{\alpha N}} \sqrt{2(f(x_0) - f(x^*))} \tag{24}$$

Esto muestra una cota sublineal de convergencia, decreciendo como:

$$\boxed{O\left(\frac{1}{\sqrt{N}}\right)} \tag{25}$$

□

D. Condiciones de estabilidad

- La función f debe ser diferenciable en todos sus puntos, para así garantizar la obtención de un gradiente para cada punto.
- La función debe cumplir con la condición de Lipschitz, es decir, que la función no tenga cambios bruscos.
- Continuando con la condición de Lipschitz, se puede determinar una tasa de aprendizaje $\alpha \leq \frac{1}{L}$, siendo L la constante de Lipschitz, para garantizar convergencia.
- Si $\alpha > \frac{1}{L}$, entonces no hay convergencia, pues los pasos son demasiado grandes respecto a la suavidad de la función f.

E. Descenso con momento

El Descenso de Gradiente estándar avanza en cada iteración hacia en la dirección del negativo del gradiente, que es la mayor tasa de decrecimiento, hasta llegar a un mínimo donde el gradiente es cero. Sin embargo, este enfoque se ve limitado para funciones con múltiples valles, varios de los cuales son poco profundos: Se puede converger a un mínimo, pero no es el más óptimo.

Una modificación a (1) retoma una idea de la física clásica: una pelota que rueda colina abajo. Conforme avanza la pelota hacia un mínimo local, ésta acumula velocidad, lo cual le permite atravesar valles pocos pronunciados y quedar atrapada en mínimos poco óptimos, para finalmente llegar a un valle lo suficientemente convexo. Este comportamiento se modela mediante la introducción de una velocidad v_t que combina la velocidad anterior con el gradiente actual.

$$v_t = \beta v_{t-1} + (1 - \beta)\nabla f(\omega_t) \tag{26}$$

$$\omega_{t+1} = \omega_t - \alpha v_t \tag{27}$$

donde v_t en (26) hace referencia a la velocidad de la iteración, y se inicializa con $v_0 = 0$; y β es el factor de momento, un valor dentro del intervalo $[0, 1)$ que típicamente se encuentra en alrededor de 0.9. Es por medio de este mecanismo que se acelera la convergencia y se escapa de mínimos locales poco óptimos, especialmente para funciones con mucha curvatura o ruido.

F. Ejemplo ilustrativo

Se tiene un polinomio de grado quinto $p(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_5x^5$ que quiere aproximarse a la función $\sin(x)$ en el intervalo desde -3 a 3. El problema está en conocer cuáles son los valores que deben tomar los coeficientes $a_0, a_1, a_2, \ldots, a_5$ para ese objetivo, por lo que se planteará una función de costo tomando como base el principio de mínimos cuadrados.

$$\theta(a_0, a_1, a_2, \ldots, a_5) = \int_{-3}^{3} (p(x) - \sin(x))^2 \, dx \tag{28}$$

La función θ representa el error cuadrático medio que hay entre la función $\sin(x)$ y el polinomio $p(x)$ en el intervalo $[-3, 3]$; y la forma en la que se minimice el valor de la función depende del valor de los coeficientes a_i de $p(x)$. Ahora, queda

obtener el gradiente de la función de costo para poder realizar la iteración del descenso de gradiente. En este caso, es posible obtener una fórmula analítica del gradiente aprovechando que $p(x)$ puede escribirse como una serie de potencias en notación sigma.

$$\theta(a_0, a_1, a_2, \ldots, a_5) = \int_{-3}^{3} \left(\sum_{k=0}^{5} a_k x^k - \sin(x) \right)^2 dx \quad (29)$$

Derivamos θ en función de los coeficientes del polinomio $p(x)$, que sería derivar θ en función de cada a_i:

$$\frac{\partial \theta}{\partial a_i} = \frac{\partial}{\partial a_i} \int_{-3}^{3} \left(\sum_{k=0}^{5} a_k x^k - \sin(x) \right)^2 dx \quad (30)$$

La ecuación (30) puede expresarse en un orden distinto, realizando la diferenciación antes de la integración, haciendo uso del teorema de Leibniz para diferencia integrales con parámetros. Por lo tanto, se tiene que la derivada de θ respecto a un parámetro a_i es:

$$\int_{-3}^{3} \frac{\partial}{\partial a_i} \left(\sum_{k=0}^{5} a_k x^k - \sin(x) \right)^2 dx \quad (31)$$

De esta forma, se obtiene la derivada parcial de θ respecto a cualquiera de sus parámetros a_i.

$$\frac{\partial \theta}{\partial a_j} = 2 \int_{-3}^{3} x^j \left(p(x) - \sin(x) \right) dx \quad (32)$$

El gradiente de la función de costo, que es el gradiente a utilizar para 1 iterativamente, tiene la forma:

$$\nabla \theta = 2 \begin{bmatrix} \int_{-3}^{3} x^0 \left(p(x) - \sin(x) \right) dx \\ \int_{-3}^{3} x^1 \left(p(x) - \sin(x) \right) dx \\ \int_{-3}^{3} x^2 \left(p(x) - \sin(x) \right) dx \\ \vdots \\ \int_{-3}^{3} x^5 \left(p(x) - \sin(x) \right) dx \end{bmatrix} \quad (33)$$

1) Método estándar: Se optó por utilizar una tasa de aprendizaje de $\alpha = 1 \times 10^{-5}$, además de utilizar métodos numéricos para obtener los resultados de las integrales que componen al gradiente. Después de mil iteraciones, se alcanzó un mínimo local donde los coeficientes a_i no cambiaban mucho, dejando como resultado una aproximación de $\sin(x)$ que minimizó el error cuadrático. La comparación del $p(x)$ con la función $\sin(x)$ se aprecia en Fig. 1. Se partió de un arreglo de coeficientes aleatorio.

En la Fig. 2 se observa el error cuadrático medio en cada iteración, evaluando la función de costo por cada arreglo de valores. Se puede apreciar una forma racional en el decrecimiento del error.

La aproximación del polinomio obtenido usando iteración de descenso de gradiente se muestra a continuación:

$$p(x) = -0.099445 + 0.197967x + 0.001409x^2 + 0.175170x^3 + 0.002015x^4 - 0.023861x^5$$

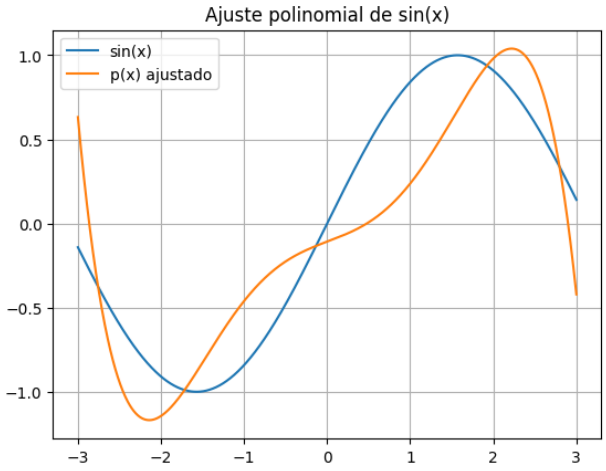

Fig. 1. Ajuste polinomial de $sin(x)$

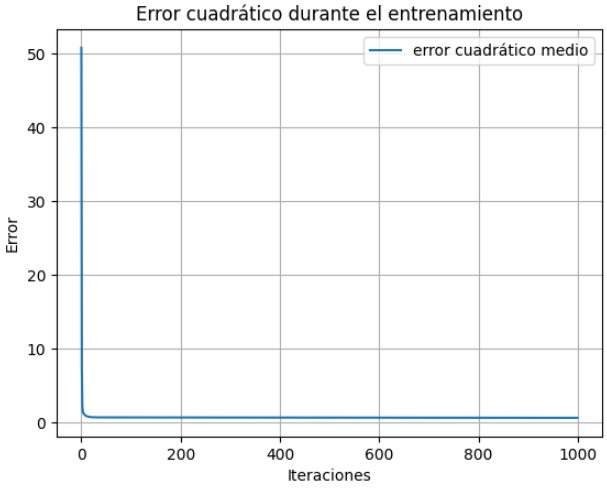

Fig. 2. Evolución del error cuadrático medio

2) Método con momento: Observando que la aproximación del método no es óptima, se puede utilizar la modificación con momento mencionada en la sección II-E para poder llegar a una mejor aproximación. Una consecuencia de este método es que la tasa de aprendizaje cambia, el valor establecido en II-F1 ahora no permite convergencia al ser muy pequeño; y ahora se estableció $\alpha = 1 \times 10^{-3}$ y un factor de momento $\beta = 0.995$. Fig. 3 muestra la aproximación de $p(x)$ después de mil iteraciones, comparando con el resultado de la sección II-F1.

En Fig. 4, la gráfica muestra cómo el error cuadrático medio disminuye y aumenta, una visualización de los mínimos poco pronunciados por los que se pasó para luego converger en un mínimo más óptimo.

El polinomio aproximado resultante con descenso con mo-

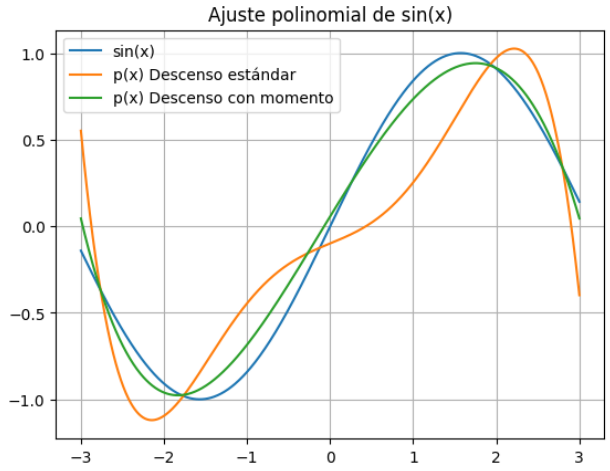

Fig. 3. Ajuste polinomial de $\sin(x)$

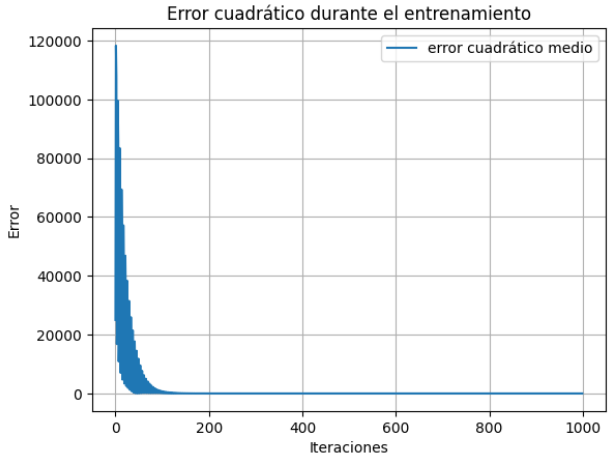

Fig. 4. Evolución del con Descenso de Gradiente con momento

mento es el siguiente:

$$p(x) = 0.056855 + 0.783697x - 0.035206x^2 - 0.072163x^3 + 0.003762^4 - 0.001656x^5$$

G. Ventajas y desventajas del método

1) Ventajas:

- El método de descenso del gradiente es escalable a muchos parámetros, dado que funciona correctamente con un gran número de parámetros. Por ello, es ampliamente utilizado en Machine Learning, por ejemplo. Es eficiente para conjuntos de datos grandes.
- Es un método fácil de implementar con una lógica relativamente sencilla.
- Es generalizable, puede ser aplicado a varias funciones de costo.

2) Desventajas:

- Es demasiado sensible a la tasa de aprendizaje α: si es muy grande puede divergir; si es muy pequeña, puede converger muy lentamente o no alcanzar el mínimo local.
- Puede atascarse en mínimos locales o puntos de silla en funciones no convexas. Se requiere de variaciones para lograr escapar de regiones no óptimas.
- Requiere que la función de costo sea diferenciable respecto a todos los parámetros.

III. Newton-Raphson multivariable

A. Descripción del método

El método de Newton-Raphson es una técnica iterativa usada para encontrar los puntos críticos de una función escalar o vectorial de múltiples variables, hablando específicamente de optimización, se utiliza para encontrar los mínimos locales de una función suave. Utiliza información de segundo orden, es decir, toma en cuenta la curvatura de la función para construir una aproximación cuadrática de la función y determinar cómo avanzar hacia el mínimo.

La estrategia que implementa el método de Newton consiste en modelar localmente la función que se desea minimizar con una parábola multidimensional. Dicha parábola se construye a partir del punto actual usando la expansión de Taylor en segundo orden. Posteriormente, se calcula el mínimo de la parábola y el punto obtenido se usa como la siguiente aproximación del mínimo. El método busca resolver el sistema de ecuaciones:

$$\nabla f(\omega) = 0$$

Lo anterior indica, que se pretende encontrar los puntos en donde la pendiente de la recta tangente a la curva es nula, sabiendo que entre estos puntos podemos encontrar los mínimos locales si la curvatura es **positiva**.

Por ello, Newton utiliza la matriz Hessiana, verificando que sea positiva y definida para asegurar que las aproximaciones se están acercando al mínimo de la función y no hacia otro punto crítico como un máximo o punto de silla. La fórmula iterativa que implementa el método de Newton es

$$\omega_{k+1} = \omega_k - H_f(\omega_k)^{-1} \cdot \nabla f(\omega_k) \tag{34}$$

Dada la fórmula y un punto inicial ω_0, el algoritmo de Newton para minimización realiza los siguientes pasos:

1) Primeramente evalúa el gradiente de la función $\nabla f(\omega_k)$ en el punto actual ω_k, indicando la dirección de mayor pendiente.
2) Luego evalúa la matriz Hessiana $H_f(\omega_k)$, que representa la curvatura, es decir, como cambia el gradiente.
3) Construye una aproximación cuadrática local de la función en torno al punto actual ω_k usando Taylor.
4) Encuentra el mínimo de la función cuadrática aproximada, derivando dicha parábola.
5) Se actualiza el punto a una nueva estimación más cercana al punto real con la fórmula iterativa

$$\omega_{k+1} = \omega_k + \Delta\omega_k \tag{35}$$

Donde $\Delta\omega_k = -H_f(\omega_k) \cdot \nabla f(\omega_k)$.

El procedimiento anterior se repite iterativamente con el objetivo de acercarse progresivamente al mínimo local de la

función. Sin embargo, dado que no se puede iterar indefinidamente en la práctica, es necesario establecer criterios de paro.

1) Criterios de paro: Todos los criterios de paro se basan en verificar que cierto valor se encuentre por debajo de una tolerancia epsilónica ϵ:

- El gradiente:
$$\|\nabla f(\omega_k)\| < \epsilon$$
- Error absoluto:
$$\|\omega_{k+1} - \omega_k\| < \epsilon$$
- Error absoluto de las imágenes:
$$|f(\omega_{k+1}) - f(\omega_k)|$$
- Número máximo de iteraciones:
$$k = k_{max}$$

El método de Newton se vuelve bastante bueno en términos de eficiencia dado que al considerar la curvatura el método puede ajustar el tamaño del paso de forma automática y adaptativa sin depender de una tasa de aprendizaje fija, puede corregir la dirección incluso si el gradiente previamente apunta a una zona menos eficiente, y puede converger en menos iteraciones que métodos de primer orden como Descenso de Gradiente. Esto se debe a que el método cerca de la solución tiene una convergencia **cuadrática**, indicando que el número de cifras correctas se duplica por cada nueva estimación en cada iteración.

B. Formulación matemática

Se tiene una función $f : \mathbb{R}^n \to \mathbb{R}$ de la cual buscamos encontrar el mínimo local, es decir, se busca un punto $\omega^* \in \mathbb{R}^n$ tal que

$$f(\omega^*) \leq f(\omega), \quad \text{para valores cercanos a } \omega^* \tag{36}$$

Para minimizar $f(\omega)$ buscamos los puntos críticos donde el gradiente se anula, por lo que minimizar la función es equivalente a resolver el sistema de ecuaciones no lineal:

$$\nabla f(\omega) = 0 \tag{37}$$

Se aplica el método de Newton-Raphson para resolver el sistema de ecuaciones. Sea $f(\omega)$ dos veces continuamente diferenciable, expandimos en serie de Taylor alrededor del punto actual ω_k:

$$f(\omega) \approx f(\omega_k) + \nabla f(\omega_k)^T(\omega - \omega_k) + \frac{1}{2}(\omega - \omega_k)^T H_f(\omega_k)(\omega - \omega_k) \tag{38}$$

Ahora, para minimizar la aproximación encontrada por medio de la expansión de Taylor de segundo orden, derivamos respecto de ω y se iguala a cero:

$$\nabla f(\omega_k) + H_f(\omega_k)(\omega_{k+1} - \omega_k) = 0 \tag{39}$$

Despejando ω_{k+1} para encontrar una nueva aproximación del mínimo de la función:

$$\omega_{k+1} = \omega_k - H_f(\omega_k)^{-1} \cdot \nabla f(\omega_k) \tag{40}$$

C. Análisis de convergencia

Para poder demostrar el orden de convergencia del método de Newton-Raphson en optimización, tenemos que mantener las hipótesis necesarias:

- La función f debe ser tres veces continua diferenciable, es decir, sus derivadas parciales tienen primera, segunda y terceras derivadas continuas.
- Su gradiente evaluado en el punto final debe ser 0, es decir, $\nabla f(\omega^*) = 0$.
- La matriz Hessiana debe ser definida positiva, es decir, que $\mathbf{x}^T H_f \mathbf{x} > 0$ para todo vector no nulo $\mathbf{x} \in \mathbb{R}^n$.
- ω_k debe de ser lo más cercano posible a ω^*.

Sea $E_k = \omega_k - \omega^*$. La expansión de Taylor del gradiente centrada en ω^* está dada por:

$$\nabla f(\omega_k) = \nabla f(\omega^*) + \nabla^2 f(\omega^*)(\omega_k - \omega^*) + R_1 \tag{41}$$

Donde:

- $\nabla f(\omega^*) = 0$ por la definición del punto mínimo.
- R_1 es término cúbico del error: $R_1 = \frac{1}{2}\nabla f(\xi)(\omega_k - \omega^*)^2$

Por lo que:

$$\nabla f(\omega_k) = \nabla^2 f(\omega^*)E_k + \mathcal{O}(\|E_k\|^2) \tag{42}$$

A su vez, se puede realizar la expansión de la Hessiana en Taylor:

$$\nabla^2 f(\omega_k) = \nabla^2 f(\omega^*) + R_1 \tag{43}$$

Ya que $R_1 = \nabla f(\xi)(\omega_k - \omega^*)$, entonces:

$$\nabla^2 f(\omega_k) = \nabla^2 f(\omega^*) + \mathcal{O}(\|E_k\|) \tag{44}$$

Recordando la iteración del método:

$$\omega_{k+1} = \omega_k - [H_f(\omega_k)]^{-1}\nabla f(\omega_k) \tag{45}$$

Buscando encontrar el error E_{k+1}:

$$E_{k+1} = \omega_{k+1} - \omega^* = \omega_k - \omega^* - [H_f(\omega_k)]^{-1}\nabla f(\omega_k) \tag{46}$$

Reemplazando (42) en (46):

$$E_{k+1} = \omega_k - \omega^* - [H_f(\omega_k)]^{-1}(\nabla^2 f(\omega^*)E_k + \mathcal{O}(\|E_k\|^2)) \tag{47}$$

Por fórmula de la inversión de matrices perturbadas y la evaluación de (44):

$$[H_f(\omega_k)]^{-1} = [\nabla^2 f(\omega_k)]^{-1} = [\nabla^2 f(\omega_k)]^{-1} + \mathcal{O}(\|E_k\|) \tag{48}$$

Sustituyendo en la ecuación anterior:

$$E_{k+1} = E_k - ([\nabla^2 f(\omega_k)]^{-1} + \mathcal{O}(\|E_k\|))(\nabla^2 f(\omega^*)E_k + \mathcal{O}(\|E_k\|^2))$$

Expandiendo:

$$\begin{aligned} E_{k+1} &= E_k - ([\nabla^2 f(\omega_k)]^{-1} \cdot \nabla^2 f(\omega_k) E_k \\ &+ [\nabla^2 f(\omega_k)]^{-1} \cdot \mathcal{O}(\|E_k\|) \\ &+ \mathcal{O}(\|E_k\|) \cdot \nabla^2 f(\omega_k) \\ &+ \mathcal{O}(\|E_k\|) \cdot \mathcal{O}(\|E_k\|^2)) \end{aligned} \tag{49}$$

Quedando:

$$E_{k+1} = \mathcal{O}(\|E_k\|^2) + \mathcal{O}(\|E_k\|^3) \tag{50}$$

Debido a que se toma el orden más bajo, entonces obtenemos que:

$$E_{k+1} = \mathcal{O}(\|E_k\|^2) \tag{51}$$

Demostrando así, que el método de Newton-Raphson multivariable de optimización converge cuadráticamente.

D. Condiciones de estabilidad

1) La función objetivo $f(\omega)$ debe ser suave y continuamente diferenciable hasta segundo orden garantizando que tanto el gradiente $\nabla f(\omega)$ como la Hessiana $H_f(\omega)$ existen y son continuas, lo cual resulta necesario para la validez de la aproximación cuadrática por la serie de Taylor de segundo orden.
2) Si el punto mínimo es ω^*, entonces la matriz Hessiana en dicho punto debe ser positiva definida, es decir, la función tiene curvatura positiva en todas las direcciones, el punto crítico es un mínimo local estricto, asegurando no converger en máximos ni puntos de silla y la dirección de Newton es un descenso verdadero. Esto se puede demostrar verificando los autovalores no nulos de la matriz Hessiana, que todos ellos sean positivos. Si la matriz Hessiana no cumple esta condición, el método puede converger diverge.
3) Newton no es globalmente convergente por sí mismo, requiere que el punto inicial ω_0 se encuentre lo suficientemente cerca del mínimo local verdadero, esto con el fin de garantizar la convergencia del método y la convergencia cuadrática eficiente. Si el punto inicial se encuentra lejos del mínimo, el método puede divergir o converger a otro punto crítico.
4) La Hessiana debe ser invertible, dado que en cada iteración se necesitar resolver el sistema de ecuaciones

$$H_f(\omega_k) \cdot \Delta\omega_k = -\nabla f(\omega_k)$$

Por lo tanto, la matriz Hessiana debe cumplir con que su determinante sea diferente de cero para que exista su inversa y el sistema posea una solución única.

E. Amortiguamiento del paso

El método de Newton-Raphson requiere de que el punto inicial no se encuentre lejos del mínimo para converger; sin embargo en funciones cuya forma no es conocida, especialmente para funciones $\mathbb{R}^n \to \mathbb{R}$, se vuelve muy complicado empezar desde una región cercana a un mínimo; y si la matriz Hessiana es inestable, la divergencia y convergencia es determinada por centésimas en el punto inicial. Entonces, para permitir convergencia, se utiliza un factor de amortiguamiento η en el método.

La fórmula iterativa modificada con amortiguamiento utilizada es la siguiente:

$$\omega_{k+1} = \omega_k - \eta H_f(\omega_k) \cdot \nabla f(\omega_k) \tag{52}$$

donde η es el parámetro de amortiguamiento, típicamente un valor en el intervalo $(0, 1]$. Este factor regula la magnitud del paso en cada iteración mientras conserva la dirección en la que el método desea converger; y se usa en caso de que el método de Newton-Raphson original no converja.

F. Ejemplo ilustrativo

Tendremos el mismo ejemplo hecho en la sección II-F para probar el Descenso de Gradiente: Encontrar los coeficientes $a_0, a_1, \ldots, a_5$ que mejor aproximan un polinomio $p(x)$ de quinto grado a la función $\sin(x)$ en el intervalo $x \in [-3, 3]$. Para ello se planteó la función de costo (28) haciendo uso de mínimos cuadrados; y además se obtuvo (33), el término del gradiente de la función de costo.

Sin embargo, la fórmula iterativa para el método de Newton-Raphson multivariable requiere de la matriz Hessiana de la función de costo. Pero, gracias al gradiente, tenemos un término general para la primera derivada parcial de una constante a_i, así que podemos obtener un término general para las segundas derivadas a_j.

Por medio de (32) se tiene el término general para las componentes del gradiente

$$\frac{\partial \theta}{\partial a_i} = 2 \int_{-3}^{3} x^i (p(x) - \sin(x))\, dx$$

Dado que la matriz Hessiana deriva la primera derivada parcial con respecto a todas las variables por segunda vez, solo queda calcular esta nueva expresión:

$$\frac{\partial}{\partial a_j} \frac{\partial \theta}{\partial a_i} = \frac{\partial^2 \theta}{\partial a_i \partial a_j} = \frac{\partial}{\partial a_j} 2 \int_{-3}^{3} x^i (p(x) - \sin(x))\, dx \tag{53}$$

Nuevamente, utilizando el Teorema de Leibniz, transformamos (53):

$$\frac{\partial^2 \theta}{\partial a_i \partial a_j} = 2 \int_{-3}^{3} \left[\frac{\partial}{\partial a_j} x^i (p(x) - \sin(x)) \right] dx = 2 \int_{-3}^{3} x^i x^j\, dx \tag{54}$$

Entonces, los términos de la matriz Hessiana son descritos por el siguiente término general:

$$H_{ij} = 2 \int_{-3}^{3} x^{i+j}\, dx \tag{55}$$

Evaluar para los coeficientes desde a_0 hasta a_5 devuelve la siguiente matriz Hessiana:

$$H = \begin{bmatrix} 12 & 0 & 36 & 0 & \frac{972}{5} & 0 \\ 0 & 36 & 0 & \frac{972}{5} & 0 & \frac{8748}{7} \\ 36 & 0 & \frac{972}{5} & 0 & \frac{8748}{7} & 0 \\ 0 & \frac{972}{5} & 0 & \frac{8748}{7} & 0 & 8748 \\ \frac{972}{5} & 0 & \frac{8748}{7} & 0 & 8748 & 0 \\ 0 & \frac{8748}{7} & 0 & 8748 & 0 & \frac{708588}{11} \end{bmatrix} \tag{56}$$

Dado que (34) requiere de la inversa de la matriz Hessiana, se tiene que encontrar la inversa de (56)

$$\begin{bmatrix} \frac{75}{256} & 0 & -\frac{175}{1152} & 0 & \frac{35}{2304} & 0 \\ 0 & \frac{1225}{2304} & 0 & -\frac{245}{1152} & 0 & \frac{385}{20736} \\ -\frac{175}{1152} & 0 & \frac{245}{1728} & 0 & -\frac{175}{10368} & 0 \\ 0 & -\frac{245}{1152} & 0 & \frac{175}{1728} & 0 & -\frac{2695}{279936} \\ \frac{35}{2304} & 0 & -\frac{175}{10368} & 0 & \frac{1225}{559872} & 0 \\ 0 & \frac{358}{20736} & 0 & -\frac{2695}{279936} & 0 & \frac{539}{559872} \end{bmatrix} \tag{57}$$

Para este caso, se tuvo la suerte de que (56) resultó ser una matriz constante. De no serlo, tendría que evaluarse para cada término el vector ω_k por cada iteración en (57), lo cual puede ser muy costoso computacionalmente. Sin embargo, ahora hay que multiplicar $H^{-1} \cdot \nabla\theta(\omega_k)$ por cada iteración, una operación costosa, pues es una multiplicación entre matrices.

Ya se tienen todos los elementos necesarios para poder realizar la iteración de Newton-Raphson, partiendo con las valores aleatorios para las contantes a_i. No obstante, el método no converge ya que (34) produce saltos demasiado grandes para este caso, probablemente porque el punto inicial no se encuentra tan cerca de un mínimo local como se esperaba. Es por ello que, para permitir convergencia, se utilizó la fórmula con amortiguamiento (52).

Para reducir la inestabilidad del método, se encontró una factor de amortiguamiento $\eta = 0.04$, sin embargo se requirieron varios intentos para obtener un mínimo local satisfactorio. No obstante, por medio de este método se logró obtener un resultado más óptimo con menos iteraciones (300 iteraciones) que con Descenso de Gradiente; cómo se puede observar en Fig. 5.

El error cuadrático medio también disminuye converge mucho más rápido a cero debido a que el método de Newton-Raphson multivariable tiene una rapidez de convergencia cuadrática. La minimización del error se puede observar en Fig. 6. El error cuadrático medio también se minimizó lo más que pudo con el Descenso de Gradiente como también muestra Fig. 46 en pocas iteraciones, pero el mínimo local alcanzado con Newton-Raphson multivariable se aproxima con menor error.

La aproximación del polinomio obtenido usando iteración de descenso de gradiente se muestra a continuación:

$$\begin{aligned} p(x) = &-0.019534 + 1.043933x - 0.017897x^2 \\ &- 0.186261x^3 + 0.003139x^4 + 0.008928x^5 \end{aligned}$$

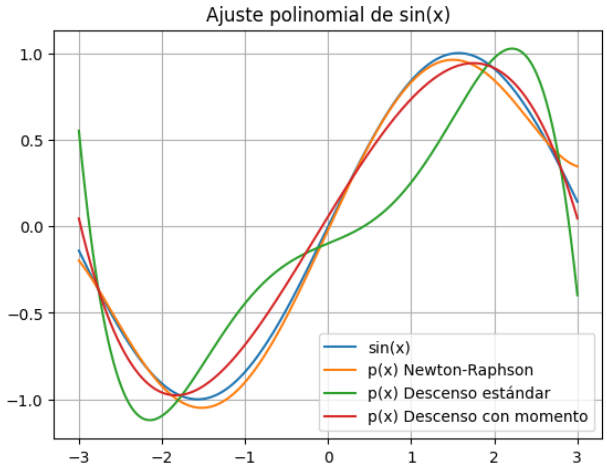

Fig. 5. Ajuste polinomial de $sin(x)$ con Newton-Raphson

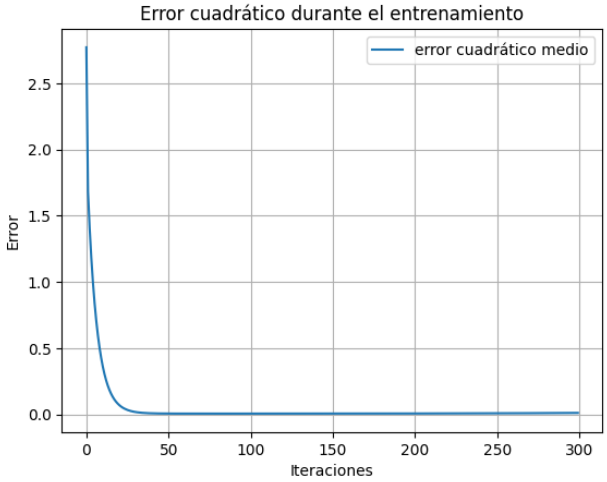

Fig. 6. Evolución del error cuadrático medio usando Newton-Raphson

G. Ventajas y desventajas

1) Ventajas:

- El método de Newton-Raphson tiene convergencia cuadrática si se cumplen las condiciones de estabilidad, por lo cual converge muy rápidamente. La cantidad de cifras correctas y precisión se duplican por cada iteración del método.
- Dado que considera la curvatura de la función mediante la matriz Hessiana, no solo el gradiente, se producen pasos más eficientes en cuánto a precisión de búsqueda.
- El método puro no requiere de una tasa de aprendizaje fija, a diferencial del Descenso de Gradiente; esto dado que se ajusta manualmente adaptándose en cada iteración usando a la matriz Hessiana.
- Al utilizar una expansión por serie de Taylor de segundo orden, se modela mejor el comportamiento real de las funciones no lineales.

2) Desventajas:

- Dado que necesita calcular la matriz Hessiana, computacionalmente puede resultar seriamente costoso y calcular las segundas derivadas analíticamente puede ser complicado o hasta imposible.
- El método puede converger a puntos que no son mínimos dado que busca puntos críticos en donde el gradiente de la función es cero, puede converger a máximos o puntos de silla, no necesariamente a mínimos. Por ello, la condición que dicta que la Hessiana debe ser positiva definida se debe cumplir, para asegurar la convergencia del método en un mínimo.
- Si el punto inicial está muy lejos del mínimo local el método puede divergir, necesita una buena estimación inicial; o bien utilizar un factor de amortiguamiento para asegurar estabilidad, cumpliendo un rol similar a la de la tasa de aprendizaje del Descenso de Gradiente.
- El método no es apto para funciones ruidosas o no suaves, requiere funciones continuas y dos veces diferenciables.

IV. Análisis comparativo de los métodos

Los métodos de Descenso de Gradiente y Newton-Raphson multivariable son herramientas eficaces para la localización de mínimos en funciones multivariables. Cada uno presenta ventajas, limitaciones y contextos de aplicación específicos que los hacen más adecuados según el tipo de problema abordado.

TABLE I
Comparación de los Métodos

Descenso de Gradiente	**Método de Newton-Raphson**
Método de primer orden, solo usa gradiente	Método de segundo orden, usa gradiente y matriz Hessiana
Velocidad de convergencia lineal o sublineal	Velocidad de convergencia cuadrática si se cumplen condiciones de estabilidad
Solo utiliza primeras derivadas, gradiente	Se requiere de segundas derivadas para la matriz Hessiana
Bajo costo computacional	Alto costo computacional al calcular la inversa de la matriz Hessiana por cada iteración
Moderadamente sensible al punto inicial	Altamente sensible, requiere de estar cerca de un mínimo para que haya convergencia.
La estabilidad depende críticamente de la elección de la tasa de aprendizaje α	Depende de la positividad y estabilidad de la matriz Hessiana
Puede atascarse en mínimos no óptimos o puntos de silla	Puede converger a puntos de silla o máximos si la Hessiana no cumple con los criterios de estabilidad
Fácil de implementar, escalable para funciones complejas	Un poco más difícil de implementar, y por la matriz Hessiana, no es fácil escalarla para lidiar con funciones de varias variables

Utilizando como base los ejemplos ilustrativos de las secciones II-F y III-F, se observó que el método de descenso de gradiente estándar tiene problemas para converger en un mínimo óptimo; e incluso cuando lo logra, requiere un número considerable de iteraciones para alcanzar una solución aceptable. No obstante, su implementación es sencilla; y los problemas con los mínimos locales pueden solucionarse con variantes del método, como es el caso del Descenso con Momento mencionado en la sección II-E.

Comparado con el Newton-Raphson multivariable, Descenso de Gradiente no es tan preciso y toma muchas más iteraciones iteraciones alcanzar algo que con el método de Newton solo tomó 300 iteraciones. No obstante, esto solo se pudo tras utilizar el método amortiguado discutido en la sección III-E, ya que el método estándar era muy inestable, y es difícil establecer un punto inicial cerca de un mínimo sin saber cómo se ve la función, especialmente cuando (28) es una función $\mathbb{R}^6 \to \mathbb{R}$. En nuestro ejemplo, el tiempo de ejecución fue menor con Newton, lo cual podría atribuirse a que, en este caso particular, la matriz Hessiana fue constante, simplificando los cálculos.

V. Conclusión

A lo largo de este documento se ha demostrado que tanto el descenso de gradiente como el método de Newton-Raphson multivariable son herramientas poderosas para la optimización de funciones multivariables, cada uno con sus propias fortalezas y debilidades.

El descenso de gradiente destaca por su simplicidad, escalabilidad y facilidad de implementación, aunque su desempeño puede verse limitado por la elección de la tasa de aprendizaje y la presencia de mínimos locales. Por otro lado, el método de Newton-Raphson ofrece una convergencia más rápida gracias al uso de información de segundo orden, pero a costa de una mayor complejidad computacional y sensibilidad a la elección del punto inicial.

La inclusión de técnicas como el momento y el amortiguamiento permite mejorar la estabilidad y eficiencia de ambos métodos. En definitiva, la elección del método óptimo dependerá del tipo de función, la disponibilidad de derivadas y los recursos computacionales, siendo recomendable un enfoque híbrido o adaptativo en problemas complejos.

References

[1] G. Garrigos y R. M. Gower, "Handbook of Convergence Theorems for (Stochastic) Gradient Methods," arXiv preprint arXiv:2301.11235, 2023. [En línea]. Disponible en: https://arxiv.org/abs/2301.11235

[2] M. Gormley, "Lecture 8: Convergence of Gradient Descent," Carnegie Mellon University, 2021. [En línea]. Disponible en: https://www.cs.cmu.edu/~mgormley/courses/10425/slides/lecture8-graddesc.pdf

[3] G. Farina, "Lecture 7: Gradient Descent," Massachusetts Institute of Technology, 2024. [En línea]. Disponible en: https://www.mit.edu/~gfarina/2024/67220s24_L07_gradient_descent/L07.pdf

[4] R. U. Yeh, "Gradient Descent," *Harvey Mudd College*, [En línea. Disponible en: https://pages.hmc.edu/ruye/MachineLearning/lectures/ch2/node7.html.

[5] A. Zhang et al., "Gradient Descent," in *Dive into Deep Learning*, [En línea]. Disponible en: https://d2l.ai/chapter_optimization/gd.html. [Accessed: May-2025].

[6] M. Davenport, "Newton's Method," *Georgia Institute of Technology*, ECE 6270 Lecture Notes, Spring 2021. [En línea]. Disponible en: https://mdav.ece.gatech.edu/ece-6270-spring2021/notes/08-newtons-method.pdf.

[7] G. Farina, "Newton's Method for Optimization," *Massachusetts Institute of Technology*, 6.7220 Lecture 12, Spring 2024. [En línea]. Disponible en: https://www.mit.edu/~gfarina/2024/67220s24_L12_newton/L12.pdf.