Tarea N°1 análisis de algoritmos

Integrantes:

Diego Silva Madariaga/20.965.500-4

Christian Pérez Flores /21.074.048-1

Sección:

412

Profesor:

Mladen Williams Nadinic Cruz

Índice.

1.Introducción.

Esta tarea trata sobre usar la estrategia de dividir y vencer para comparar los dos métodos para calcular la CCS: uno basado en una implementación iterativa tradicional, y otro que utiliza la Transformada Rápida de Fourier (FFT) junto con su inversa (IFFT)

Aunque el contexto del problema pertenece al procesamiento de señales, el enfoque está orientado al estudio de algoritmos desde la perspectiva del análisis de su eficiencia y diseño. Es por esto que el objetivo principal es ilustrar cómo la estrategia de dividir y vencer permite optimizar el rendimiento de un algoritmo que, en su versión iterativa, posee una complejidad mayor.

En el procesamiento digital de señales, la convolución es una operación esencial para diversas aplicaciones como el filtrado de señales, detección de patrones y procesamiento de audio. Sin embargo, su implementación directa puede llegar a ser ineficiente cuando se trata de señales de gran tamaño, Es por esto que la Suma de Convolución Circular (CCS) surge como una alternativa computacionalmente viable, especialmente cuando se combina con técnicas de transformada rápida.

2.Desarrollo

2.1 Explicación del problema

La suma de convolución (CS) es una operación fundamental en el procesamiento de señales, equivalente a la integral de convolución en el caso continuo. Sin embargo, esta operación se define sobre un rango infinito (de -∞ a ∞), lo que la hace impracticable para su implementación discreta en computadores. Para resolver esta dificultad, se utiliza una versión acotada y computacionalmente viable llamada suma de convolución circular (CCS) de N puntos.

$$N = N_1 + N_2 - 1,$$

donde N_1 y N_2 son los largos de las señales x[n] y h[n], respectivamente.

Para poder aplicar la CCS, ambas señales deben tener largo N, lo cual se logra rellenándolas con ceros (zero-padding). Esto permite obtener una salida y[n] también de largo N.

$$y[n] = H^{**T} \cdot x \quad \text{(Ecuación 1)}$$

donde HT es la transpuesta de la matriz de tipo Toeplitz construida a partir de h[n], y x es la señal de entrada como vector columna (sin zero-padding adicional).

También se puede resolver la CCS de forma más eficiente mediante la transformada discreta de Fourier (DFT). Si X[k] y H[k] son las DFT de x[n] y h[n].

$$Y[k] = X[k] \cdot H[k] \quad \text{(Ecuación 2)}$$

Y la señal y[n] se obtiene aplicando la inversa de la DFT (IDFT) a Y[k].

Para mejorar la eficiencia del cálculo, se utiliza la transformada rápida de Fourier (FFT) y su inversa (IFFT), que son implementaciones recursivas basadas en la estrategia de dividir y conquistar, válidas cuando N es una potencia de 2, es decir, N = 2^m.

2.2 Análisis del problema

La primera implementación recorre explícitamente los índices involucrados mediante ciclos for, lo que resulta en un algoritmo de complejidad temporal O(N**2).

La segunda implementación aplica la Transformada Rápida de Fourier (FFT) y su inversa (IFFT), ambas programadas recursivamente siguiendo la estrategia Dividir y Vencer. Esta técnica divide la señal original en dos partes (pares e impares), resuelve la FFT para cada una de ellas, y luego combina los resultados. El tiempo de ejecución asociado a esta técnica está dado por la ecuación de recurrencia:

$$T(N)=2T(N/2)+O(N)$$

Que resuelta mediante el Teorema Maestro corresponde a un orden de complejidad:

$$T(N)=O(N\ logN)$$

Lo cual representa una mejora significativa frente al enfoque iterativo tradicional.

2.3 Código usado

Para el desarrollo de la tarea se utilizó el lenguaje Python por su facilidad para manejar operaciones matemáticas y visualización de datos. También se usaron las bibliotecas numpy para operaciones vectoriales y números complejos, matplotlib para la generación de gráficas, cmath para funciones complejas, y scipy.linalg para la construcción de la matriz Toeplitz.

Se implementaron tres funciones principales(el código usado se adjunta con este informe y el link a un Google collab en las referencias)

- CCS iterativa con matriz Toeplitz

En este método se calcula el largo de la señal de salida como $N = N_1 + N_2 - 1$, donde N_1 y N_2 son las longitudes de las señales de entrada x[n] y h[n]. Luego, ambas señales se rellenan con ceros hasta tener largo N. A partir de la señal h[n], se construye una matriz de tipo Toeplitz circular, Esta matriz representa la operación de convolución circular.

Finalmente, se calcula la salida y[n] mediante la multiplicación matricial entre la transpuesta de la matriz Toeplitz y el vector columna de la señal x[n]. Este método es computacionalmente costoso.

- CCS utilizando FFT e IFFT (estrategia dividir y conquistar)

Este segundo método utiliza la estrategia dividir y conquistar para calcular la convolución circular de manera más eficiente. Primero, se expande el tamaño de ambas señales al siguiente número potencia de 2 igual o mayor a N, lo que es necesario para aplicar la FFT correctamente, luego se implementa lo siguiente.

Se implementa la FFT de manera recursiva. La señal se divide en dos partes: una con los elementos en posiciones pares y otra con los impares. Este proceso se repite hasta llegar a señales de tamaño 1. Luego se combinan los resultados usando la fórmula de la FFT, Una vez obtenidas las transformadas de x[n] y h[n], se realiza una multiplicación punto a punto entre ellas.

Después, se aplica la IFFT (también implementada de forma recursiva y con la misma lógica de dividir y combinar), para obtener la señal de salida y[n] en el dominio del tiempo, Finalmente se toma solo la parte real de la señal obtenida.

Este enfoque tiene una complejidad de orden N log N, lo que lo hace mucho más eficiente que el método iterativo cuando se trabaja con señales largas.

- Comparación de tiempos de ejecución

Para evaluar la eficiencia de ambos métodos, se generaron señales aleatorias de longitud creciente, con valores entre 1 y -1 para $2 \leq N \leq 2^{**}13$.

Para cada tamaño de señal se midió el tiempo de ejecución del método iterativo y del método con FFT/IFFT usando un cronómetro computacional. Los tiempos obtenidos se almacenaron y se graficaron para visualizar cómo aumenta el tiempo de ejecución.

3.Analisis de los resultados

La gráfica presentada compara los tiempos de ejecución de los dos métodos desarrollados para calcular la Suma de Convolución Circular, el método iterativo basado en la multiplicación con la matriz Toeplitz transpuesta (según la ecuación 1) y el método FFT basado en la estrategia dividir y conquistar. Se utilizaron señales aleatorias con valores entre 1 y -1 para 2 ≤ N ≤ 2**13.

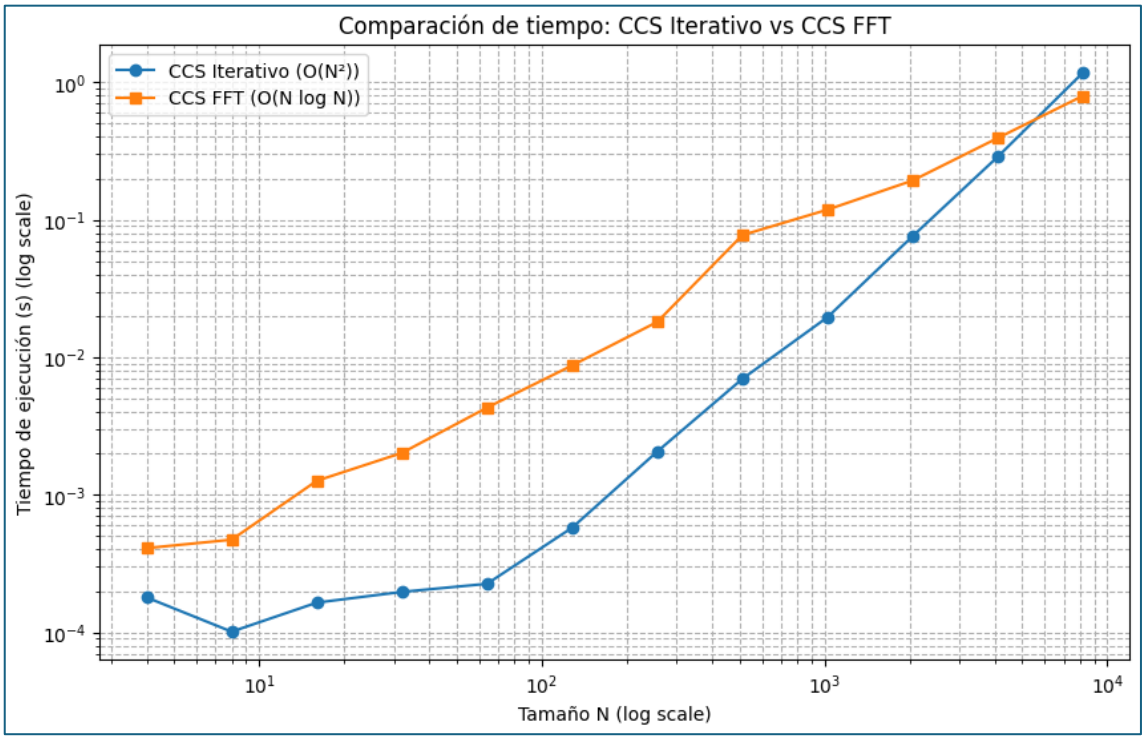

Gráfico 1. comparación de tiempos

Los tiempos de ejecución para cada algoritmo fueron graficados en escala logarítmica, la gráfica final muestra los dos comportamientos diferenciados.

El algoritmo iterativo, presenta un crecimiento exponencial respecto al tamaño N, lo cual es consistente con su complejidad O(N**2).

El algoritmo FFT recursivo, basado en la estrategia de Dividir y Vencer, presenta una curva de crecimiento mucho más suave, acorde O(N logN).

Debido al uso de señales aleatorias, los tiempos pueden variar levemente en cada ejecución, lo que justifica pequeñas diferencias entre gráficas generadas en distintas corridas del código. No obstante, la tendencia de comportamiento se mantiene constante, demostrando así la efectividad de las estrategia de dividir y vencer.

4.Conclusiones

Se puede concluir que la estrategia de dividir y vencer permite reducir significativamente la complejidad computacional de ciertos problemas, como se demuestra al comparar la CCS tradicional frente a su versión mediante FFT.

Pues la implementación iterativa sirve como base conceptual, pero resulta ineficiente para tamaños grandes, mientras que el uso de FFT, correctamente implementado de forma recursiva, mejora el rendimiento gracias a la descomposición eficiente del problema, validando el enfoque teórico del curso de *Análisis de Algoritmos*.

5.Referencias

- A Anand Kumar. Digital Signal Processing. PHI Learning, 2013

  Link al código, se debe ingresar con un correo UTEM

- https://colab.research.google.com/drive/1vkk7gUvEywoU4rnRVcnDnHgf33CAZpg6?usp=sharing