FAST FOURIER TRANSFORM

YASH GUTGUTIA(180001064)

PRITESH PALOD(180001038)

Overview

The Fourier Transform is a mathematical technique that transforms a function of time, $f(x)$, to a function of frequency, $F(\omega)$. The Fourier transform of a function of time is itself a complex-valued function of frequency, whose magnitude represents the amount of that frequency present in the original function, and whose argument is the phase offset of the basic sinusoid in that frequency.

When a signal is discrete and periodic, we don't need the continuous Fourier transform. Instead we use the discrete Fourier transform, or DFT. The FFT is a fast algorithm for computing the DFT. There are many variants of the FFT algorithm developed over time to reduce the computational complexity of the algorithm.

Specifications

The Fourier transform (FT) of the function f(x) is the function F(ω), where:

$F(\omega) = \int f(x) e^{-2\pi i \omega x}.dx$ (from -∞ to ∞).

The Discrete Fourier Transform (DTF) can be written as follows.

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

If we used a computer to calculate the Discrete Fourier Transform of a signal, it would need to perform N (multiplications) x N (additions) = $O(N^2)$ operations.

Due to high computational requirement of brute-force computation of DFT, it was not possible to use that for real-time and online DSP applications for DFTs of larger lengths until 1965, when Cooley and Tukey developed the famous fast Fourier transform (FFT) algorithm. The Fast Fourier Transform (FFT) is an algorithm that determines Discrete Fourier Transform of an input significantly faster than computing it directly. It could be possible to reduce the operation count of DFT from $O(N^2)$ to $O(N\log_2 N)$, for a DFT of length N. During the last 50 years, the innovations in algorithms and architectures have made remarkable progress in the efficiency of computation of the FFT.

The article published by Cooley-Tukey presented an efficient algorithm based on divide-and-conquer approach in order to compute the DFT. Divide-and-conquer approach was applied to the DFT recursively, such that a DFT of any size N = N1*N2 was computed in terms of smaller DFTs of sizes N1 and N2.

Need

Fourier Transform is a mathematical operation that changes the domain (x-axis) of a signal from time to frequency. The discrete Fourier transform (DFT) is the most widely used tool in digital signal processing (DSP) systems. It has indispensable role in many applications such as speech, audio and image processing, signal analysis, communication systems, and many others.

The application of the Fourier Transform isn't limited to digital signal processing. The Fourier Transform can, in fact, speed up the training process of convolutional neural networks.

Goals

The beauty of FFT is its efficient computation of DFT. The direct computation of DFT requires computation of the order of $O(N^2)$ whereas FFT involves only $O(N\log_2 N)$ operations.

Via this project, we will present a brief overview of important advancements in algorithms and applications of FFT.

- Implementation and Analysis of DFT.
- Important advancements in FFT algorithms and its applications.
- Study of Cooley-Tukey FFT algorithm.

Related research papers and References

- Fourier Transforms and the Fast Fourier Transform Algorithm by Paul Heckbert
- 50 Years of FFT Algorithms and Applications G. Ganesh Kumar, Subhendu K. Sahoo & Pramod Kumar Meher
- On the Partial Differential Equations of Mathematical Physics by R. Courant, K. Friedrachs, H. Lewy

- An Algorithm for the machine Calculation of Complex Fourier Series by James W. Cooley and John W. Tukey
- https://www.researchgate.net/publication/333029661_50_Years_of_FFT_Algorithms_and_Applications
- https://towardsdatascience.com/fast-fourier-transform-937926e591cb