UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)

Ciencia de Datos

Visualización de la Información

TAREA-3

18.02.2022

1. Introducción

La serie de Fourier aproxima una señal o función contínua mediante una serie infinita de sinusoides.

$$f(t) = a_0 + \sum_{k=1}^{\infty} [a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)] \tag{1}$$

donde los coeficientes de la ecuación se calculan como:

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega_0 t)\, \delta t \tag{2}$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega_0 t)\, \delta t \tag{3}$$

El análisis de Fourier transmite una función como un agregado de componentes periódicos y extrae esas señales de los componentes. Cuando tanto la función como su transformada se intercambian con las partes discretas, entonces se expresa como Transformada de Fourier. La Transformada rápida de Fourier (FFT) trabaja principalmente con algoritmos computacionales para aumentar la velocidad de ejecución. Algoritmos de filtrado, multiplicación, procesamiento de imágenes son algunas de sus aplicaciones.

2. Actividad 1

Utilice el módulo `scipy.fft` de Python para la transformación rápida de Fourier de la siguiente función:

$$f(t) = 3\sin(7{,}2\pi t + 2) + 1 \tag{4}$$

Grafique la función y visualícela de manera que se pueda ver bien toda la función y sus cambios. Puede usar `t=np.linspace(0,T,N)` donde $T = 10$ y $N = 256$

2.1. Consideraciones

Uno de los puntos más importantes para medir en la Transformada rápida de Fourier es que solo podemos aplicarlo a datos en los que la marca de tiempo es uniforme. El módulo `scipy.fft` convierte el dominio de tiempo dado en el dominio de frecuencia. La FFT de longitud N secuencia `x[n]` se calcula mediante la función `fft()`. Por ejemplo:

```
from scipy.fftpack import fft
```

El `numpy.fft` funciona de forma similar al módulo `scipy.fft`. El `scipy.fft` exporta algunas características del `numpy.fft`. El `numpy.fft` se considera más rápido cuando se trata de matrices 2D. La implementación es la misma.

3. Actividad 2

Dado el siguiente código:

```
import matplotlib.pyplot as plt
import numpy as np
from math import pi

[x,y] = np.meshgrid(np.linspace(0,10,64),
np.linspace(0,10,64))

f = np.sin(2*pi*(0.2*x + 0.7*y))+np.random.uniform(0, 1, x.shape)
plt.pcolor(x,y,f)
```

Obtenga la Transformada directa e inversa en 2D así como sus gráficas y visualícelas de tal forma que se puedan ver bien todas las funciones y sus cambios.