Лабораторная работа 7

Луняк Николай

19 апреля 2021 г.

Оглавление

Список иллюстраций

Листинги

Глава 1

Запуск `chap07.ipynb`

Запускаем, проверяем, все работает.

Глава 2

Быстрое преобразование Фурье

2.1 Теория

Нам предстоит реализовать алгоритм самостоятельно, взяв за основу «Danielson-Lanczos lemma».

$$DFT(y)[n] = DFT(e)[n] + e^{-\frac{2\pi in}{N}} \cdot DFT(o)[n]$$

где

$$\begin{aligned} DFT(y)[n] &- n\text{-ый элемент DFT от } y \\ e &- \text{массив, содержащий элементы } y[2k-2] \ (k \in \mathbb{N}) \\ o &- \text{массив, содержащий элементы } y[2k-1] \ (k \in \mathbb{N}) \end{aligned}$$

Идея алгоритам заключается в том, что мы на каждом шаге делим данные пополам, вызывая алгоритм рекуррунтно для каждой половины, а затем пользуемся леммой выше. Деление на четные и нечетные элементы занимает $\Theta(n)$ времени и памяти, а «глубина» рекурсии есть $\Theta(\log n)$.

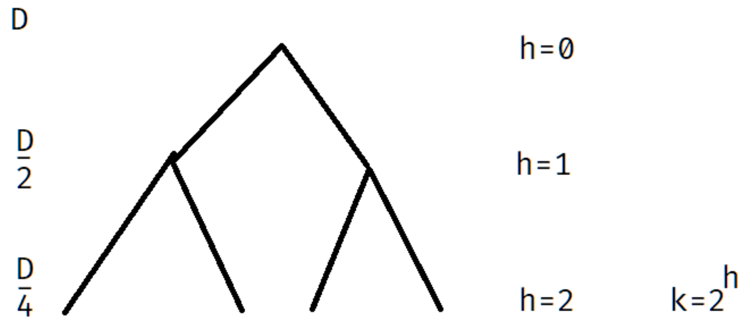

Рис. 2.1: Визуализация (описывать графику в латехе больно, я люблю WYSIWYG)

На рисунке 2.1 k означает число вершин на уровке h, а D - сколько данных каждая такая вершина обрабатывает. Таким образом, если мы все сложим, то общее число действий, которые нужно совершить:

$$\begin{aligned} S &= \sum_{h=0}^{\log_2 D} 2^h \cdot \frac{D}{2^h} \\ &= \sum_{h=0}^{\log_2 D} D \\ &= D \cdot \sum_{h=0}^{\log_2 D} 1 \\ &= D \cdot (\log_2 D + 1) \end{aligned}$$

И тогда

$$\Theta(S) = \Theta(D \cdot (\log_2 D + 1)) = \Theta(D \cdot \log D)$$

Стоит еще обратить внимание на то, что из формулировки леммы неясно, какую форму должен иметь результат, ведь e и o содержат в 2 раза меньше элементов, чем y. Также не оговаривается, как мы должны поступать в случаях, когда на некотором шаге рекурсии в очередном y нечетное число элементов (если у нас $len(y) \neq 2^m$ $(m \in \mathbb{N})$).

В силу периодичности DFT будем использовать `numpy.tile()`, чтобы «растянуть» e и o до размеров y. При этом возможно остающийся один непарный элемент в конце будем игнорировать. Нетрудно видеть, что результаты вызова $DFT(e)$ и $DFT(o)$ будут всегда либо массивами с четным числом элементов, либо с одним единственным элементом (база).

2.2 Реализация

Для начала подгрузим накопившийся код с импортами на «все случаи жизни».

```
from thinkdsp import Signal, Sinusoid, SquareSignal, TriangleSignal, SawtoothSignal, ParabolicSignal
from thinkdsp import normalize, unbias, PI2, decorate
from thinkdsp import Chirp
from thinkdsp import read_wave
from thinkdsp import Spectrum, Wave, UncorrelatedGaussianNoise, Spectrogram
from thinkdsp import Noise

import numpy as np
import pandas as pd

from matplotlib import pyplot

import thinkstats2

from scipy.stats import linregress

import scipy
import scipy.fftpack

loglog = dict(xscale='log', yscale='log')

PI2 = np.pi * 2
```

Листинг 2.1: Импорты

Теперь посмотрим «референсный» `fft` и сравним его со медленной реализацией из книжки.

```
y = [-0.5, 0.1, 0.7, -0.1]
np.fft.fft(y)
```

Листинг 2.2: Точно верный ответ

И в логе видим: `array([ 0.2+0.j , -1.2-0.2j, 0.2+0.j , -1.2+0.2j])`.

```
def dft(ys):
    N = len(ys)
```

```
    ts = np.arange(N) / N
    freqs = np.arange(N)
    args = np.outer(ts, freqs)
    M = np.exp(1j * PI2 * args)
    amps = M.conj().transpose().dot(ys)
    return amps

dft(y)
```

Листинг 2.3: Реализация из книжки

И в логе видим: `array([ 0.2+0.00000000e+00j, -1.2-2.00000000e-01j, 0.2+1.95943488e-16j, -1.2+2.00000000e-01j])`.

```
def fft(y):
    N = len(y)
    half = N // 2

    if half == 0:
        return y

    e = np.zeros((half))
    o = np.zeros((half))

    for it in range(half):
        e[it] = y[it * 2]
        o[it] = y[it * 2 + 1]

    # len() may be less than half
    # (if half is odd)
    dft_e = fft(e)
    dft_o = fft(o)
    M = len(dft_e)

    dft_e_tiled = np.tile(dft_e, 2)
    dft_o_tiled = np.tile(dft_o, 2)

    dft = np.zeros((M * 2), dtype=np.complex)

    for it in range(M * 2):
        dft[it] = dft_e_tiled[it] + np.exp(-PI2 * 1j * it / (
    M*2)) * dft_o_tiled[it]

    return dft

fft(y)
```

Листинг 2.4: Наш код

И в логе видим: `array([ 0.2+0.j , -1.2-0.2j, 0.2+0.j , -1.2+0.2j])`. Это вполне согласуется с предыдущими результатами, что свидетельствует о верности.