

PYTHON-БИБЛИОТЕКА ДЛЯ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ SCIPY

Воронко Т.М.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Научный руководитель: Пискун Г.А. – к.т.н, доцент, доцент кафедры ПИКС

Аннотация. Проведен обзор библиотеки Python для решения математических задач *SciPy*. Описаны ее особенности, основные сферы применения, а также набор входящих в нее пакетов. Приведен пример использования данной библиотеки в программе для удаления нежелательного шума из аудиосигнала с помощью преобразования Фурье. Сделан вывод о важности ее использования в программах, содержащих комплексные математические вычисления.

Ключевые слова: *SciPy, Python, NumPy, matplotlib*, математические вычисления

Введение. *Python* – это один из самых популярных языков программирования на сегодняшний день, использующийся для решения огромного количества задач различной сложности и направленности. Одной из главных причин такой популярности, наряду с легким синтаксисом и универсальностью, стало наличие огромного числа дополнительных инструментов, значительно упрощающих и ускоряющих разработку программ: библиотек и фреймворков [1].

Python нередко используется в проектах, требующих проведения различных математических операций и анализа полученных данных. *SciPy* – одна из библиотек *Python*, предоставляющих эти функции.

Основная часть. *SciPy* – это библиотека для языка *Python*, основанная на расширении *NumPy*, но для более глубоких и сложных научных вычислений, анализа данных и построения графиков. *SciPy* в основном написана на *Python* и частично на языках *C*, *C++* и *Fortran*, поэтому отличается высокой производительностью и скоростью работы [2].

SciPy используют специалисты по *Data Science*, *Big Data*, аналитики данных, а также ученые из различных областей наук [2]:

- для сложных математических расчетов, которые тяжело произвести вручную или с помощью калькулятора;

- проведения научных исследований, где требуется использование продвинутой математики;

- глубокого анализа данных, интерполяции и других методов работы с информацией;

- машинного обучения и создания моделей искусственного интеллекта, прогнозирования и построения моделей;

- формирования двумерных и трехмерных графиков, которые можно потом визуализировать (уже при помощи других библиотек, таких как *matplotlib*).

В *SciPy* есть набор пакетов для разных научных вычислений, представленный в таблице 1 [3].

Таблица 1 – Набор пакетов *SciPy*

Название	Описание
<i>cluster</i>	Алгоритмы кластерного анализа
<i>constants</i>	Физические и математические константы
<i>fftpack</i>	Быстрое преобразование Фурье
<i>integrate</i>	Решения интегральных и обычных дифференциальных уравнений
<i>interpolate</i>	Интерполяция и сглаживание сплайнов
<i>io</i>	Ввод и вывод
<i>linalg</i>	Линейная алгебра

Продолжение таблицы 1

Название	Описание
<i>ndimage</i>	<i>N</i> -размерная обработка изображений
<i>odr</i>	Метод ортогональных расстояний
<i>optimize</i>	Оптимизация и численное решение уравнений
<i>signal</i>	Обработка сигналов
<i>sparse</i>	Разреженные матрицы
<i>spatial</i>	Разреженные структуры данных и алгоритмы
<i>special</i>	Специальные функции
<i>stats</i>	Статистические распределения и функции

В качестве примера использования библиотеки создадим программу, которая удаляет нежелательный шум из аудиофайла с помощью преобразования Фурье [4].

Листинг кода для создания сигнала с высокочастотным шумом с помощью библиотеки *NumPy* и *matplotlib* для визуализации:

```
import numpy as np
from matplotlib import pyplot as plt

SAMPLE_RATE = 44100 # Гц
DURATION = 5 #Секунды
def generate_sine_wave(freq, sample_rate, duration):
    x = np.linspace(0, duration, sample_rate*duration, endpoint=False)
    frequencies = x * freq
    # 2pi для преобразования в радианы
    y = np.sin((2 * np.pi) * frequencies)
    return x, y

# Генерируем волну с частотой 2 Гц, которая длится 5 секунд
x, y = generate_sine_wave(2, SAMPLE_RATE, DURATION)
plt.plot(x, y)
plt.show()
```

Графическое представление сгенерированного сигнала изображено на рисунке 1.

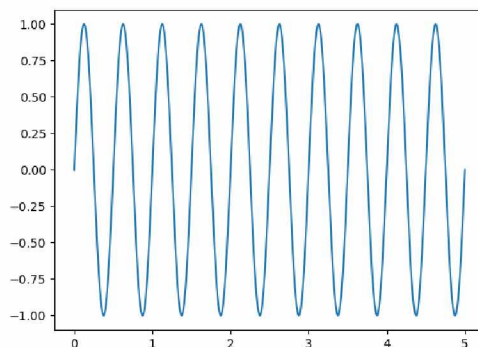


Рисунок 1 – Графическое представление сгенерированного сигнала

Проведя микширование аудиосигналов путем их сложения и нормализации полученного значения, применим преобразование Фурье [5]:

```
_, nice_tone = generate_sine_wave(400, SAMPLE_RATE, DURATION)
_, noise_tone = generate_sine_wave(4000, SAMPLE_RATE, DURATION)
noise_tone = noise_tone * 0.3
mixed_tone = nice_tone + noise_tone
normalized_tone = np.int16((mixed_tone / mixed_tone.max()) * 32767)
```

```
from scipy.fft import fft, fftfreq # Импорт fft-пакета из SciPy
# число точек в normalized_tone
N = SAMPLE_RATE * DURATION
yf = fft(normalized_tone)
xf = fftfreq(N, 1 / SAMPLE_RATE)
```

Результат FFT-преобразования в виде спектра представлен на рисунке 2.

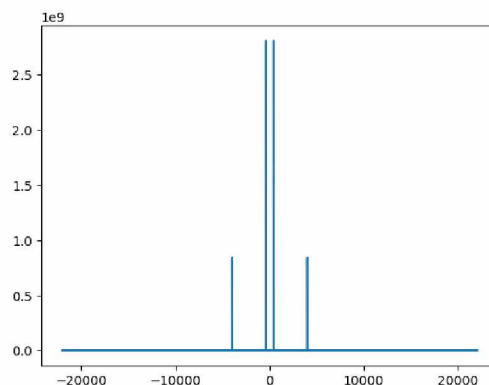


Рисунок 2 – Результат FFT-преобразования

Заключение. Библиотека SciPy позволяет разработчикам на Python упростить и оптимизировать решение самых разнообразных задач, связанных с математическими вычислениями. Набор пакетов данной библиотеки отличается большой вариативностью, что способствует ее использованию в различных областях как научного, так и аналитического исследования, а также в множестве других направлений информационных технологий.

Список литературы

1. Eternalhost // Популярные инструменты Python: библиотеки и фреймворки [Электронный ресурс] – 2022. – Режим доступа: <https://eternalhost.net/blog/razrabotka/python-biblioteki>.
2. SkillFactory // SciPy [Электронный ресурс] – 2023. – Режим доступа <https://blog.skillfactory.ru/glossary/scipy/>.
3. PythonRu // Руководство по SciPy: что это, и как ее использовать [Электронный ресурс] – 2022. – Режим доступа: <https://pythonru.com/biblioteki/scipy-python>.
4. RealPython // Fourier Transforms With scipy.fft: Python Signal Processing [Электронный ресурс] – 2019. – Режим доступа: <https://realpython.com/python-scipy-fft/>.
5. Proglib // Преобразование Фурье для обработки сигнала с помощью Python [Электронный ресурс] – 2020. – Режим доступа: <https://proglib.io/p/preobrazovaniya-fure-dlya-obrabotki-signalov-s-pomoshchyu-python-2020-11-03>.

UDC 004.432

PYTHON LIBRARY FOR PERFORMING MATHEMATICAL TASKS SCIPY

Voronko T.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Piskun G.A – Cand. of Sci., associate professor, associate professor of the department of ICSD

Annotation. A review of the Python library for solving mathematical tasks SciPy is conducted. Its features, main field of application, as well as the set of packages included in it are described. An example of using this library in a program to remove unwanted noise from an audio signal using a Fourier transform is given. A conclusion is drawn about the importance of its use in programs containing complex mathematical calculations.

Keywords: SciPy, Python, NumPy, matplotlib, mathematical calculations