Uzbek Scholar Journal
Volume- 47, December- 2025
www.uzbekscholar.com

NUMPY LIBRARY AND ITS FEATURES

Tajimamatov Isroil Nurmamatovich
Fergana State University, Department of Applied Mathematics and
Senior Lecturer of the Department of Informatics
Email: israiltojimamatov@gmail.com

Shavkatova Dilyoraxon Iqbol kizi
Student at Fergana State University
Email: dildoradhavkatova@gmail.com

Abstract
This article discusses the theoretical and practical capabilities of the NumPy library, which is the main tool for performing numerical calculations and data analysis in the Python programming language. The introduction explains the role of NumPy in scientific computing, artificial intelligence, and big data processes worldwide, and the relevance of the topic. The literature review analyzes the results obtained in the works of foreign scientists such as T. Oliphant, S. van der Walt, and CR Harris on ndarray arrays, the array-programming paradigm, and the formation of the NumPy ecosystem. The results and discussion section scientifically reveals the possibilities of ndarray arrays, vectorization and broadcasting mechanisms, statistical and linear algebra functions, random number generation, working with files, and integration with other libraries. Also, conclusions and practical recommendations are developed on the prospects for the application of NumPy in geodesy, GIS, remote sensing, economic and social research, and education.

Keywords: Python, NumPy, ndarray arrays, vectorization, broadcasting, statistical functions, linear algebra, random numbers, scientific computing, data analysis.

INTRODUCTION

Against the backdrop of the digital transformation process, the expansion of the concept of "big data" and the rapid development of artificial intelligence technologies, the need for a convenient programming environment for scientific computing and data analysis has increased dramatically. Over the past decade, the Python programming language and the ecosystem of libraries formed around it have taken a leading role in satisfying this need. In particular, the NumPy library is recognized as the foundation of this ecosystem.[1]

NumPy (Numerical Python) is a high-performance library for working with numerical arrays and matrices, which, unlike the simple list structure of the Python language, provides compact memory management, vectorized operations, and fast operations on multidimensional data. NumPy is not only a standalone library, but also a common “array layer” for the entire scientific Python ecosystem. Popular libraries such as Pandas, SciPy, Matplotlib, scikit-learn, TensorFlow, PyTorch perform their internal calculations using NumPy arrays.[2]

[1]https://www.nature.com/articles/s41586-020-2649-2
[2]https://arxiv.org/abs/2006.10256

NumPy is used as a core computing tool in hundreds of areas around the world, including satellite image processing, physical modeling, bioinformatics, financial risk assessment, and working with machine learning model parameters. For example, the work of CR Harris and co-authors shows how the array paradigm has been used in complex scientific projects such as gravitational wave detection and black hole image processing using the NumPy library.[3]

In the higher education system of Uzbekistan, there is a growing need to teach students modern programming tools in such disciplines as "Artificial Intelligence and Artificial Neural Networks", "Data Analysis", "GIS and Remote Sensing". In this process, in-depth mastery of NumPy will allow students to work independently on scientific and practical projects, effectively deal with large amounts of data, and also understand the internal mechanisms of artificial intelligence models.

In this context, the purpose of this article is to analyze the NumPy library and its capabilities theoretically and practically, to provide an overview of its development stages and role in the scientific ecosystem based on scientific literature, and to develop proposals for its use in the educational and research process.

To achieve the goal, the following tasks were set:

- Covering the basic concepts and architecture of the NumPy library (in particular, ndarray arrays);
- analysis of the content of scientific research conducted on NumPy by foreign scientists;
- scientifically and theoretically justify the advantages of vectorization, broadcasting, statistics, and linear algebra functions;
- Provide results and discussion based on examples of NumPy's applications in areas such as data analysis, artificial intelligence, and GIS;
- Develop recommendations for NumPy teaching methodology for higher education in Uzbekistan.

LITERATURE ANALYSIS

A number of fundamental developments have been made by scientists around the world on the NumPy library, which allow us to evaluate NumPy not only as a technical tool, but also as a conceptual paradigm that unifies the entire scientific Python ecosystem.

First of all, the “Guide to NumPy” manual written by Travis E. Oliphant occupies a special place in the history of the formation and development of the NumPy library. In this work, the author explains in detail the need for the emergence of NumPy, the initial limitations of the Python language for numerical calculations, and how these problems were solved through the ndarray structure. Oliphant's approach analyzes the optimization of operations on arrays in C, the concept of universal functions (ufunc), broadcasting rules, and modules such as linear algebra, Fourier transforms, and random number generation as a whole system.[4]

The second important source is the article “The NumPy Array: A Structure for Efficient Numerical Computation” by S. van der Walt, SC Colbert, G. Varoquaux, which systematically describes the internal structure of the NumPy array, memory management, and basic principles for improving performance. The authors show that in order to achieve C-level speed in Python, a high-level language, three main approaches are needed: vectorization, minimizing data copying, and reducing the number of operations. The article also analyzes the interoperability of NumPy arrays with other libraries.[5]

[3]https://www.nature.com/articles/s41586-020-2649-2
[4]https://web.mit.edu/dvp/Public/numpybook.pdf
[5]https://arxiv.org/abs/1102.1523

The third important study is the article “Array Programming with NumPy” by CR Harris et al. In this work, NumPy is interpreted not only as an array library, but also as a central implementation of the array-programming paradigm. The authors show various examples of organizing, visualizing, and analyzing data using NumPy arrays in many fields, such as physics, astronomy, geophysics, biology, and finance. At the same time, the article also highlights the role of NumPy as an interface that provides interoperability with GPUs, distributed computing, and other array libraries.[6]

Additionally, various online courses, university textbooks, and blog articles highlight NumPy's advantages in scientific computing - its speed, efficient memory usage, rich feature set, and open source community support.[7]

The literature review shows that NumPy is a common foundation for almost all scientific libraries working in Python today, and its theoretical foundations are quite well developed. At the same time, there is a need for a systematic analysis of NumPy's capabilities in specific disciplines - for example, geodesy, cartography, GIS, remote sensing, economic modeling. This article aims to partially fill this gap.

RESULTS AND DISCUSSION

Analysis of the NumPy library shows that it is not just a simple "array library", but is the basis of the array-programming paradigm, a conceptual approach that has fundamentally changed the methodology of scientific computing. Below, the main capabilities of NumPy are analyzed in detail from a scientific and methodological point of view.

First, the ndarray arrays that make up NumPy's central object provide a high level of abstraction for the Python programmer. Unlike traditional Python lists, ndarrays consist of elements of the same type, whose memory area is arranged sequentially. This provides the cache-friendly features inherent in the C language and dramatically increases the speed of arithmetic operations. As a result, for example, when performing a simple addition operation on a 10 million-element vector, NumPy gives the result in a few milliseconds using vectorized operators, while pure Python code performing the same operation with for loops takes seconds. This difference is important not only for practical convenience, but also in terms of saving computing resources when working with large amounts of scientific data.

Secondly, the concept of vectorization forms the methodological basis of NumPy. Just as it is natural in mathematics to express operations on vectors and matrices through formulas, NumPy also uses notations in this spirit. For example, writing the sum of two vectors as z = x + y actually performs element-by-element addition over millions of elements. Here, the user does not manually control the iteration process - he trusts NumPy's internal implementation. This approach frees the programmer from low-level technical details and allows him to focus on the essence of the problem - the mathematical model and algorithm.

Third, the broadcasting mechanism in NumPy simplifies operations on arrays of different sizes. For example, there is no need to create additional loops or temporary arrays to add the same 1D vector to each row of a 2D matrix or to multiply certain columns by scalars. According to the rules of broadcasting, a small-sized array is logically “fitted” to a large-sized array and operations are performed element-by-element. This mechanism not only makes the code compact, but also reduces additional memory consumption. In scientific research - for example, when applying transformations

[6]https://arxiv.org/abs/2006.10256
[7]https://www.cloudthat.com/resources/blog/empowering-scientific-computing-in-python-with-numpy

corresponding to each pixel in satellite images - effective use of broadcasting principles can significantly reduce computational time.

Fourth, the NumPy library's set of statistical functions allows for quick data analysis at the initial stage. Calculation of indicators such as the mean, median, quartiles, variance, standard deviation, minimum-maximum, correlation is performed with one or two lines of code. It is also possible to analyze separately by rows or columns using the axis parameter. For example, if we depict annual yield data for different districts in the Fergana region in the form of a 2D array, calculations on axis=0 will provide statistical indicators in terms of districts, and on axis=1 - in terms of years. This is very convenient for rapid exploratory analysis (EDA) in agricultural, economic or demographic research.

Fifth, the set of functions provided for linear algebra makes NumPy a classic mathematical modeling tool. With the help of functions such as inverse matrix, determinant, eigenvalues and eigenvectors, solving systems of linear equations in the numpy.linalg module, an “internal engine” of models is built that allows you to solve physical processes, balance equations in economic models, calculate regression coefficients, and solve differential equations by discretization. For example, in the analytical solution of multivariate regression models, it is necessary to calculate the inverse matrix of the X^TX matrix; NumPy organizes this process not only in terms of code simplicity, but also in terms of computational speed.

Sixth, NumPy also has extensive support for random number generation. NumPy's random number module (new interfaces to numpy.random in the current generation) is essential for modeling based on Monte Carlo methods, statistical simulations, initializing neural network weights, and augmenting data by adding random noise. The role of this module becomes even more obvious when we consider that the random selection of initial weights in training artificial intelligence models has a significant impact on model convergence.

Seventh, NumPy also has sufficient capabilities for working with files. Writing and reading arrays to disk in a compact form in .npy and .npz formats, as well as loading data from text .txt and .csv files, facilitates the exchange of information necessary for industry, geoinformatics, and statistical analysis. For example, geodetic measurement results can be temporarily stored in .csv files, loaded into NumPy arrays, and after performing the necessary coordinate transformations, filtering, and smoothing operations, the results can be exported to external systems. This process helps to automate a chain of multi-stage geodetic calculations.

Eighth, one of the most important aspects of NumPy is its integration with other libraries. The DataFrame structure in the Pandas library stores internal data in NumPy arrays; in Matplotlib graphs, the coordinates of axes and plotted points are obtained from NumPy objects; the optimization, signal processing, and interpolation modules in SciPy also work with ndarrays. In machine learning, NumPy arrays serve as the main input data passed to the Neural Network frameworks - TensorFlow and PyTorch. Thanks to this integration, a student can quickly work with dozens of libraries by mastering one basic abstraction - ndarray.

Ninth, it is worth noting the role of NumPy in the process of programming education. In traditional programming courses, students are often limited to syntax, data types, and standard algorithms (sorting, searching). Laboratory and practical exercises based on NumPy allow them to deal with real scientific problems - for example, analyzing experimental results, solving differential equations numerically, 3D visualization of surfaces, calculating indices (NDVI, NDBI, etc.) from satellite images.

This is important for increasing student motivation and connecting theoretical mathematics and programming with real life.

Tenth, the open-source and community-based development model of the NumPy library makes it a stable and continuously updated tool. NumPy is hosted on GitHub; researchers, engineers, and educators from around the world actively contribute to its code base, adding new features, fixing bugs, and improving documentation. This process not only continuously improves the quality of the library, but also allows students to participate in open source projects.

As for the use of NumPy in the conditions of Uzbekistan, first of all, there are great opportunities in the fields of geodesy, cartography and GIS. Remote sensing data - for example, each channel of Landsat, Sentinel, MODIS images can be loaded into the NumPy environment as a multidimensional array, and operations such as calculating spectral indices, dynamic analysis over time, filtering, image standardization and normalization can be performed on them. Also, some statistical analyses on the land cadastre - the area share of land types, crop yields, land reclamation indicators - can be quickly calculated with the help of NumPy and subsequently exported to GIS platforms. This increases the analytical potential of research institutes, organizations working on cadastre and land resources.

In order to improve the methodology of teaching NumPy, the following suggestions can be put forward. First, a special section “NumPy Fundamentals” should be included in the curriculum of subjects such as “Artificial Intelligence and Artificial Neural Networks”, “Data Analysis”, “GIS and Remote Sensing”. It is advisable to organize at least 2–3 laboratory works in this section, including ndarray arrays, vectorization, broadcasting, indexing and linear algebra operations. Second, to enhance students' independent learning, small project works based on NumPy can be assigned - for example, tasks on working with a real dataset (population statistics, meteorological data, economic indicators, geophysical measurements). Third, it is advisable for scientific supervisors to develop modern programming skills by requiring NumPy-based calculations in dissertations written for master's and doctoral students.

The above scientific analysis and discussion show that the NumPy library is a strategically important tool for modern scientific computing and data analysis. It not only extends the capabilities of the Python language, but also forms a common "language" for many disciplines and fields - a paradigm for working with arrays. In this regard, in-depth mastery of NumPy should be considered a necessary competency for today's students, researchers, and practitioners.

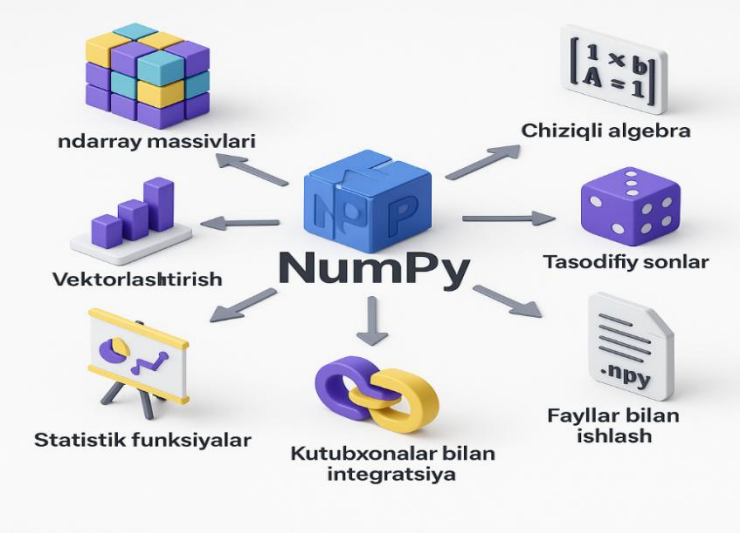

Figure 1. NumPy training methodology

CONCLUSION

The analysis showed that the NumPy library is a central tool for performing numerical calculations and data analysis in the Python programming language. Thanks to ndarray arrays, vectorized operations, broadcasting mechanism, rich statistical and linear algebra functions, NumPy significantly accelerates work with large volumes of data and allows you to write code concisely and clearly. As a result, students, researchers and practitioners can focus on the essence of the problem - mathematical models and algorithms, without being distracted by low-level technical details.

According to the results of the literature review, the works of Oliphant, van der Walt, Harris and other foreign scientists interpret NumPy not only as a simple library, but also as the foundation of the scientific Python ecosystem, the main implementation of the array-programming paradigm. NumPy has become a major computing platform in artificial intelligence, data science, physics, astronomy, geophysics, bioinformatics, economic modeling and many other fields, working in seamless integration with libraries such as Pandas, SciPy, Matplotlib, scikit-learn, TensorFlow, PyTorch.

NumPy is also of great importance in the conditions of Uzbekistan: in geodesy, cartography, GIS and remote sensing, NumPy creates great opportunities for processing satellite image channels as arrays, for rapid analysis of statistical and economic indicators, for building models necessary for land cadastre and resource monitoring. Therefore, it is advisable to teach the basics of NumPy as a mandatory component in higher education institutions within the framework of such disciplines as “Artificial Intelligence”, “Data Analysis”, “GIS and Remote Sensing”, and to organize laboratory and independent work based on this library.

In general, the NumPy library is a strategically important tool for training competitive specialists in the context of modern digital technologies, increasing the efficiency of scientific research, and forming a human resource base capable of working with large amounts of data, and its in-depth mastery is a requirement today.

REFERENCES

1. Kawaguchi K. Deep Learning without Poor Local Minima // Advances in Neural Information Processing Systems (NeurIPS). - 2016. - No. 29.
2. Choromanska A., Henaff M., Mathieu M., Ben Arous G., LeCun Y. The Loss Surfaces of Multilayer Networks // Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS). – 2015. – P. 192–204.
3. Dauphin YN, Pascanu R., Gulcehre Ç., Cho K., Ganguli S., Bengio Y. Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization // Advances in Neural Information Processing Systems (NeurIPS). – 2014. – P. 2933–2941.
4. Nguyen Q., Hein M. The Loss Surface of Deep and Wide Neural Networks // Proceedings of the 34th International Conference on Machine Learning (ICML). – 2017. – P. 2603–2612.
5. Goodfellow I., Bengio Y., Courville A. Deep Learning. – Cambridge, MA: MIT Press, 2016. – 775 p. (Chapter 8: Optimization for Training Deep Models - discussion of local minima and saddle points).
6. Choromanska A., LeCun Y., Ben Arous G. Open Problem: The Landscape of the Loss Surfaces of Multilayer Networks // Conference on Learning Theory (COLT). – 2015. – P. 1756–1760.
7. Goodfellow I. Notes on Optimization and Loss Surfaces in Deep Learning // Online Abstracts and Reviews, 2016 – (Discussions on local minima and saddle points).

8. Charpiat G. Deep Learning in Practice: Optimization Landscape and Overparameterization. – Online lecture notes, Université Paris-Saclay, 2017. – (Loss landscape, local minima, saddle points, overparameterization).