

NUMPY KUTUBXONASI VA IMKONIYATLARI

Karimova Noila Zafarjon qizi

E-mail: noilakarimova86@gmail.com

Annotatsiya

Mazkur maqolada Python dasturlash tilining asosiy ilmiy hisoblash kutubxonalaridan biri — **NumPy** (Numerical Python) kutubxonasi va uning imkoniyatlari batafsil yoritilgan. NumPy kutubxonasi ko'p o'lchovli massivlar (ndarray) bilan ishlash, matematik, statistik hamda chiziqli algebra amallarini bajarish, tasodifiy sonlar generatsiyasi va ma'lumotlar tahlilini samarali amalga oshirish imkonini beruvchi qudratli vosita sifatida tahlil qilingan. Maqolada kutubxonaning paydo bo'lish tarixi, rivojlanish bosqichlari, tuzilma elementlari, massivlar bilan ishlash usullari hamda matematik funksiyalarning amaliy qo'llanilishi misollar asosida bayon etilgan. NumPy kutubxonasi vektorlashtirish texnologiyasi orqali hisoblash jarayonlarining tezligini oshirishi, C va Fortran tillari bilan integratsiya qobiliyati, shuningdek, u orqali katta hajmdagi ma'lumotlarni optimallashtirishning ilmiy asoslari keltirilgan. Tadqiqot natijalariga ko'ra, NumPy kutubxonasi Python tilining ilmiy va muhandislik sohalaridagi ustuvorligini belgilovchi asosiy komponentlardan biri bo'lib, zamonaviy ilmiy tahlil va modellashtirish tizimlarida ajralmas o'rinni egallaydi.

Kalit so'zlar: NumPy, Python, ndarray, massiv, ilmiy hisoblashlar, vektorlashtirish, ma'lumotlar tahlili, chiziqli algebra, statistik tahlil, sun'iy intellekt, modellashtirish, ilmiy dasturlash, optimallashtirish.

Abstract. This article details one of the primary scientific computing libraries in the Python programming language: the NumPy (Numerical Python) library and its capabilities. The NumPy library was analyzed as a powerful tool that allows working with multidimensional arrays (ndarray), performing mathematical, statistical, and linear algebra operations, and effectively generating random numbers and analyzing data. The article describes the history of the library's emergence, its stages of development, structural elements, methods for working with arrays, and practical applications of mathematical functions, using examples. The scientific foundations of the fact that the NumPy library increases the speed of computational processes through vectorization technology, the ability to integrate with C and Fortran languages, as well as the optimization of large volumes of data through it, are presented. According to the research results, the NumPy library is one of the main components determining the priority of the Python language in the scientific and engineering spheres and occupies an integral place in modern systems of scientific analysis and modeling.

Keywords: NumPy, Python, ndarray, array, scientific calculations, vectorization, data analysis, linear algebra, statistical analysis, artificial intelligence, modeling, scientific programming, optimization.

Аннотация. В данной статье подробно рассматривается одна из основных научных вычислительных библиотек языка программирования Python - библиотека NumPy (Numerical Python) и ее возможности. Библиотека NumPy была проанализирована как мощный инструмент, позволяющий работать с многомерными массивами (ndarray), выполнять математические, статистические и линейные алгебраические операции, эффективно генерировать случайные числа и анализировать данные. В статье на примерах описывается история библиотеки, этапы ее развития, элементы структуры, методы работы с массивами и практическое применение математических функций. Приведены научные основы повышения скорости вычислительных процессов благодаря технологии векторизации библиотеки NumPy, ее способности к интеграции с языками C и Fortran, а также оптимизации больших объемов данных с ее помощью. Согласно результатам исследования, библиотека NumPy является одним из ключевых компонентов, определяющих приоритетность языка Python в научной и инженерной областях, и занимает неотъемлемое место в современных системах научного анализа и моделирования.

Ключевые слова: NumPy, Python, ndarray, массив, научные вычисления, векторизация, анализ данных, линейная алгебра, статистический анализ, искусственный интеллект, моделирование, научное программирование, оптимизация.

Kirish

Zamonaviy ilmiy tadqiqotlar, muhandislik hisob-kitoblari va raqamli texnologiyalar sohasining jadal rivojlanishi natijasida katta hajmdagi ma'lumotlar ustida tezkor va aniq hisoblashlarni amalga oshiruvchi dasturlash vositalariga ehtiyoj ortib bormoqda. Shu nuqtayi nazardan, Python dasturlash tili ilmiy muhitda keng tarqalgan bo'lib, uning eng muhim afzalliklaridan biri — turli sohalar uchun ishlab chiqilgan boy kutubxonalar to'plamidir. Shulardan eng asosiylaridan biri bu — NumPy (Numerical Python) kutubxonasidir.

NumPy kutubxonasi — Python tilida sonli hisoblashlar, chiziqli algebra, statistik tahlil, signallarni qayta ishlash va modellashtirish kabi ko'plab ilmiy yo'nalishlarda asosiy vosita bo'lib xizmat qiluvchi, ochiq manbali dasturiy ta'minot hisoblanadi. NumPy yordamida foydalanuvchi ko'p o'lchovli massivlar (array) ustida yuqori tezlikda amallar bajarishi, murakkab matematik funksiyalarni qo'llashi va katta hajmdagi ma'lumotlarni samarali tarzda saqlashi mumkin. Oddiy Python ro'yxatlari

(list) ma'lumotlar ustida moslashuvchan ishlasa-da, ular sonli hisoblashlarda sust ishlaydi. Sababi, Python interpretatori har bir elementni alohida ob'ekt sifatida qayta ishlaydi, bu esa ko'p resurs talab qiladi. NumPy esa bu muammoni hal qilib, vektorlashtirish tamoyiliga asoslangan massivlar bilan ishlashni taklif etadi. Vektorlashtirish yordamida barcha elementlarga bir vaqtning o'zida amal qo'llaniladi, natijada kodning ishlash samaradorligi o'nlab barobarga oshadi. Bugungi kunda NumPy kutubxonasi sun'iy intellekt, mashinaviy o'rganish, fizika, kimyo, bioinformatika, iqtisodiyot va muhandislik sohalarida keng qo'llanilmoqda. Masalan, neyron tarmoqlar uchun ma'lumotlarni tayyorlash, tajriba natijalarini modellashtirish yoki statistik tahlillarni avtomatlashtirishda aynan NumPy asosiy rolni bajaradi. NumPy kutubxonasi shunchaki matematik vosita emas, balki u butun Python ilmiy ekotizimi (SciPy ekotizimi) uchun poydevor vazifasini o'taydi. Bugungi kunda SciPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, va PyTorch kabi mashhur kutubxonalar o'z funksional imkoniyatlarida NumPy massivlari va algoritmlaridan foydalanadi. Shu sababli, NumPy'ni chuqur o'rganish — Python dasturlash tilida ilmiy tadqiqot olib borishning birinchi qadamidir. NumPy'ning muhim jihatlaridan yana biri — u **C** va **Fortran** tillarida yozilgan yuqori samarali kodlar bilan bevosita integratsiyalashgan. Shuning uchun u oddiy Python kodidan ancha tez ishlaydi, ammo dasturchiga qulay va o'qilishi oson interfeysni saqlab qoladi. Bu esa foydalanuvchiga past darajadagi optimallashtirilgan hisoblashlardan foydalanish imkonini beradi.

NumPy kutubxonasi murakkab matematik operatsiyalarni yuqori tezlikda bajarish imkonini beruvchi optimallashtirilgan arxitekturaga ega. Uning asosida **vektorlashtirilgan hisoblash** g'oyasi yotadi. Bu yondashuvda hisob-kitoblar elementma-element emas, balki butun massiv (array) ustida bir martalik amal sifatida bajariladi. Shu sababli, NumPy kodlari oddiy Python sikllariga nisbatan o'nlab marta tez ishlaydi. NumPy'ning ishlash samaradorligi bir nechta asosiy texnologik tamoyillarga asoslanadi:

1. *C tilida yozilgan yadro (core) mexanizmi* — NumPy'ning asosiy funksional modullari C dasturlash tilida yozilgan bo'lib, ular to'g'ridan-to'g'ri kompyuter xotirasi bilan ishlaydi. Bu Python interpretatori orqali o'tmasdan, past darajadagi hisoblashni amalga oshirish imkonini beradi.
2. *Homogeneous data storage (bir turdagi ma'lumotlar)* — oddiy Python listlaridan farqli o'laroq, NumPy massivlarida barcha elementlar bir xil ma'lumot turiga ega (masalan, float64, int32 va hokazo). Bu esa massiv elementlariga to'g'ridan-to'g'ri murojaat qilishni va xotira sarfini kamaytirishni ta'minlaydi.
3. *Memory-contiguous storage* — NumPy massivlari ketma-ket joylashgan xotira bloklarida saqlanadi. Shu sababli, ularni kompyuter kesh (cache) tizimi samarali qayta ishlay oladi va bu tezlikni oshiradi.

4. *Broadcasting mexanizmi* — turli o'lchovdagi massivlar o'zaro arifmetik amallar bajarish jarayonida avtomatik ravishda moslashtiriladi. Masalan, 3×3 o'lchovdagi matritsaga 1×3 o'lchovli massivni qo'shish mumkin, NumPy bu jarayonni avtomatik ravishda "broadcast" qiladi.

NumPyning ichki tuzilmasi quyidagi asosiy komponentlardan iborat:

- *ndarray obyekti* — bu NumPy'ning asosiy ma'lumot turi bo'lib, ko'p o'lchovli massivni ifodalaydi;
- *ufunc (universal functions)* — massiv elementlari ustida vektorlashtirilgan tarzda ishlaydigan matematik funksiyalar to'plami;
- *random moduli* — statistik modellash va tasodifiy sonlar generatsiyasi uchun vositalar;
- *linalg moduli* — chiziqli algebra amallari (matritsa teskari, determinant, norm va boshqalar) uchun;
- *fft moduli* — Fourier o'zgartirishlari va signal tahlili uchun;
- *polynomial moduli* — ko'phadlar bilan ishlash imkoniyatlarini ta'minlaydi.

Shuningdek, NumPy kutubxonasi *parallel hisoblash* imkoniyatlarini ham qisman qo'llab-quvvatlaydi. Ba'zi amallar (masalan, dot, sum, mean) ichki darajada **BLAS** va **LAPACK** kabi past darajadagi ko'p yadroli matematik kutubxonalar bilan integratsiyalashgan. Bu esa ko'p yadroli protsessorlarda ishlash samaradorligini yanada oshiradi. NumPyning ishlash prinsipi shunchaki matematik amallar bajarishdan iborat emas, u — *ma'lumotlarni strukturaviy tarzda saqlash va ularga optimal tarzda murojaat qilish* mexanizmini ham o'z ichiga oladi. Shu sababli, NumPy faqat matematik vosita emas, balki samarali *ma'lumotlar tuzilmasi (data structure)* sifatida ham qaraladi.

NumPy kutubxonasining markaziy obyekti bu — *ndarray (n-dimensional array)* hisoblanadi. Narray — bu o'lchovlar soni (ndim), shakli (shape) va ma'lumot turi (dtype) bilan aniqlanadigan ko'p o'lchovli massivdir. U sonli ma'lumotlarni ixcham shaklda saqlash va ular ustida vektorlashtirilgan amallar bajarish uchun mo'ljallangan. Har bir ndarray quyidagi asosiy atributlarga ega:

- ndim** — o'lchovlar soni (masalan, 1D, 2D, 3D massivlar);
- shape** — har bir o'lchovdagi elementlar soni (masalan, (3, 4) — 3 qatordan 4 ustunli massiv);
- size** — massivdagi jami elementlar soni;
- dtype** — elementlarning ma'lumot turi (int32, float64, complex128, va hokazo);
- itemsize** — bitta element hajmi (baytlarda);
- nbytes** — massiv tomonidan egallangan umumiy xotira hajmi.

NumPy massivlari odatda quyidagi turlarga bo'linadi:

Bir o'lchovli massivlar (vektorlar) — matematikada vektor sifatida qaraladi. Misol: `np.array([1,2,3,4,5])`

Ikki o'lchovli massivlar (matritsalar) — qator va ustunlardan iborat.

Misol: `np.array([[1,2,3],[4,5,6]])`

Ko'p o'lchovli massivlar (tenzorlar) — 3 yoki undan ortiq o'lchovga ega.

Misol: `np.random.rand(3,4,2)`

Ko'p o'lchovli massivlar sun'iy intellekt va mashinaviy o'rganishda (xususan, neyron tarmoqlarda) keng qo'llaniladi, chunki ular ko'p parametrli ma'lumotlarni ifodalay oladi (masalan, tasvirlar, vaqt qatorlari yoki fazoviy modellar). NumPy massivlarida *indekslash* (elementga murojaat qilish) va *slicing* (massivning bir qismini ajratish) amallari Pythonnikiga o'xshash, ammo yanada qulayroq. NumPy kutubxonasi matematika va statistika sohasidagi deyarli barcha asosiy funksiyalarni o'z ichiga oladi. Quyidagi funksiyalar eng ko'p qo'llaniladi:

Matematik funksiyalar: `sin ()`, `cos ()`, `tan ()`, `exp ()`, `log ()`, `sqrt ()`, `abs ()`, `power ()` va boshqalar;

Statistik funksiyalar: `mean ()`, `median ()`, `std()`, `var()`, `min()`, `max()`, `percentile()`;

Matritsalar uchun: `dot ()`, `matmul()`, `transpose()`, `det()`, `inv()`.

NumPydagi matematik va statistik funksiyalar *vektorlashtirilgan* shaklda ishlaydi, ya'ni butun massiv ustida bir zumda amal bajariladi. Bu hisoblash tezligi nuqtayi nazaridan juda katta ustunlik beradi. NumPy shuningdek murakkab sonlar, trigonometrik tahlil, eksponentsial va logarifmik hisob-kitoblar uchun ham qo'llaniladi. Ilmiy tadqiqotlarda u ko'pincha SciPy, Pandas, Matplotlib bilan birgalikda ishlatiladi. Shu sababli, NumPy Python ilmiy ekotizimining asosi hisoblanadi. NumPy kutubxonasi chiziqli algebra amallarini bajarishda yuqori samaradorlikka ega bo'lgan `numpy.linalg` moduliga ega. Bu modul yordamida **determinant, matritsaning teskari qiymati, xususiy qiymatlar, xususiy vektorlar**, hamda **chiziqli tenglamalar sistemasini yechish** kabi muhim masalalarni yechish mumkin. Chiziqli algebra amallari matematik modellashtirish, fizik jarayonlarni hisoblash, statistik regressiyalar va mashinaviy o'rganish algoritmlarining asosini tashkil etadi. Aynan shuning uchun `numpy.linalg` moduli ilmiy tahlillarda keng qo'llaniladi. NumPyning `numpy.random` moduli tasodifiy sonlar generatsiyasi uchun mo'ljallangan bo'lib, statistik modellashtirish, Monte-Karlo metodlari, tajriba tahlili va mashinaviy o'rganish sohalarida keng qo'llaniladi.

Bu modul quyidagi asosiy imkoniyatlarga ega:

Bir tekis taqsimot: `np.random.rand()`

Normal taqsimot (Gauss): `np.random.randn()`

Butun sonlar: `np.random.randint()`

Binomial, Poisson, Exponential, Beta, Gamma kabi statistik taqsimotlardan namuna olish funksiyalari.

NumPy kutubxonasining eng asosiy ustunligi — bu **vektorlashtirish** (*vectorization*) mexanizmidir. Oddiy Python'da sikllar orqali har bir element ustida alohida amal bajariladi, bu esa ko'p vaqt talab etadi. NumPy esa butun massiv ustida

bir vaqtning o'zida operatsiyalarni bajaradi. NumPy Python ekotizimining “yadrosi” sifatida ko‘plab boshqa ilmiy kutubxonalar uchun asosiy poydevor hisoblanadi. Boshqacha aytganda, Pythondagi deyarli barcha ilmiy va texnik kutubxonalar ichki hisob-kitoblarini NumPy massivlariga tayangan holda amalga oshiradi.

a) SciPy bilan integratsiya

SciPy (*Scientific Python*) – bu NumPyning kengaytmasi bo‘lib, murakkab matematik, statistik, optimallashtirish, signalni qayta ishlash, differensial tenglamalarni yechish kabi amallarni bajaradi. SciPy modullari (`scipy.integrate`, `scipy.optimize`, `scipy.signal`, `scipy.stats` va boshqalar) o‘z hisoblarini to‘liq NumPy `ndarray` massivlari asosida olib boradi.

Masalan, integrallash jarayoni:

```
from scipy import integrate
import numpy as np
```

```
f = lambda x: np.sin(x)
I, _ = integrate.quad(f, 0, np.pi)
print("∫sin(x)dx =", I)
```

Bu yerda `quad()` funksiyasi SciPy’dan olingan bo‘lsa-da, `np.sin(x)` va integrallash jarayoni to‘liq NumPy massivlari orqali amalga oshadi.

b) Pandas bilan integratsiya

Pandas – ma’lumotlar tahlili uchun mo‘ljallangan kutubxona bo‘lib, `DataFrame` obyektlari NumPy massivlariga asoslangan. Har bir ustun aslida `ndarray` sifatida saqlanadi. Shu sababli, Pandas’da bajarilgan ko‘plab statistik va matematik amallar aslida NumPy funksiyalaridan foydalanadi.

Masalan:

```
import pandas as pd
import numpy as np
```

```
Df = pd.DataFrame(np.random.randn(3, 3), columns=['A', 'B', 'C'])
print(df.mean())
```

Bu yerda `mean()` funksiyasi ostida `numpy.mean()` chaqiriladi.

c) Matplotlib va vizualizatsiya

Matplotlib – ma’lumotlarni vizualizatsiya qilish uchun eng mashhur Python kutubxonasidir. U grafik chizish uchun NumPy massivlaridan foydalanadi. Har bir `plt.plot(x, y)` amali aslida `ndarray` ko‘rinishidagi ma’lumotlarni o‘qib, ularni koordinata tekisligida ifodalaydi.

d) TensorFlow, PyTorch, Scikit-learn bilan integratsiya

Mashina o‘rganish (Machine Learning) va sun’iy intellekt sohalarida ishlatiladigan asosiy kutubxonalar — **TensorFlow**, **PyTorch**, **Scikit-learn** — barchasi NumPy’ga

tayanadi. Masalan, TensorFlow'da `tf.Tensor` obyektlarini NumPy massivlariga oson aylantirish mumkin:

```
import tensorflow as tf
import numpy as np
```

```
a = np.array([[1, 2], [3, 4]])
t = tf.convert_to_tensor(a)
print(t)
```

Bu integratsiya ilmiy hisob-kitoblarda yagona ekotizimni yaratadi. NumPy shu tarzda barcha asosiy ilmiy kutubxonalar uchun “*standart hisoblash formati*” sifatida xizmat qiladi. NumPy bugungi kunda ilm-fan, muhandislik, iqtisodiyot, tibbiyot, geologiya, meteorologiya va mashinaviy o'rganish sohalarida keng qo'llanilmoqda. Quyida uning eng muhim amaliy yo'nalishlari keltiriladi.

— **Muhandislik va ilmiy hisoblash:** muhandislik sohasida differensial tenglamalarni sonli yechish, jarayonlarni modellashtirish, tajriba natijalarini qayta ishlashda NumPy asosiy vosita sifatida ishlatiladi. Misol uchun:

issiqlik almashinuvi jarayonlarini modellashtirish;

gaz oqimlari va bosim taqsimotini tahlil qilish;

mexanik tebranishlar va rezonanslarni hisoblash;

barbotajli absorber va kimyoviy reaktor modellari uchun massa almashinuvi hisoblari.

— *Sun'iy intellekt va mashinaviy o'rganish:* NumPy mashinaviy o'rganish algoritmlarining “fundamentini” tashkil etadi. Asosiy algoritmlar — regressiya, klasterlash, gradientli tahlil — barchasi matritsa amallariga asoslangan bo'lib, ular NumPy yordamida bajariladi. NumPy kvant mexanikasi, molekulyar dinamikasi, issiqlik o'tkazish, termodinamik jarayonlar kabi fizik modellarni simulyatsiya qilishda qo'llaniladi. Kimyoviy reaksiya tezligini, energiya potensialini yoki gaz konsentratsiyasini hisoblashda ham NumPy asosiy hisoblash dvigateli bo'lib xizmat qiladi.

— *Tibbiyot va bioinformatika:* Bioinformatika sohasida genetik ketma-ketliklar, oqsillar strukturasini tahlil qilish, rentgen yoki MRI tasvirlarini raqamli qayta ishlashda NumPy algoritmlari qo'llaniladi.

— *Geologiya va meteorologiya:* NumPy geologik qatlamlar tahlili, zilzila kuchlanish modellarini yaratish, iqlim o'zgarishini modellashtirishda ham muhim rol o'ynaydi. Masalan, harorat o'zgarishini vaqt bo'yicha regressiya modeli bilan ifodalash uchun NumPy qo'llaniladi.

Xulosa

NumPy kutubxonasi bugungi kunda Python tilining eng muhim ilmiy komponenti sifatida tan olingan. Uning yuqori samaradorlikka ega vektorlashtirilgan hisoblash yondashuvi, ko'p o'lchovli massivlar bilan ishlash imkoniyatlari, hamda

boshqa kutubxonalar bilan integratsiyasi tufayli u ilmiy tahlil va texnik hisob-kitoblarning ajralmas qismiga aylangan. NumPy orqali foydalanuvchi nafaqat oddiy matematik amallarni, balki chiziqli algebra, statistik tahlil, ehtimollik modellash, signalni qayta ishlash, differensial tenglamalarni yechish kabi murakkab jarayonlarni ham qulay shaklda amalga oshirishi mumkin. Shuningdek, NumPy hisoblash tezligini o'nlab marta oshiradi, kodni soddalashtiradi va tahlilni avtomatlashtiradi, ilmiy ma'lumotlarni vizualizatsiyaga tayyorlaydi, sun'iy intellekt tizimlari uchun asosiy hisoblash bazasini ta'minlaydi. NumPy kutubxonasi o'zining ochiq manbali tabiati va keng jamoatchilik tomonidan qo'llab-quvvatlanishi natijasida tez rivojlanib bormoqda. Shu bois, hozirda u nafaqat Python ekotizimining ajralmas qismi, balki butun ilmiy hisoblash sohasining universal vositasi sifatida e'tirof etiladi.

Foydalanilgan adabiyotlar

1. Bekmuratov Q.A. , Sun'iy intellekt va neyron tarmoqlar. O'quv qo'llanma, Samarqand – 2021.
2. Sadullayeva SH.A, Yusupov D.F., Yusupov F., Sun'iy intellect va neyronto'rli texnologiyalar. O'quv qo'llanma, Urganch – 2021.
3. H.N.Zayniddinov, T.A.Xo'jaqulov, M.P.Atadjanov, "Sun'iy intellekt" fanidan o'quv qo'llanma, Toshkent – 2018.
4. Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). "The NumPy Array: A Structure for Efficient Numerical Computation." *Computing in Science & Engineering*, 13(2), 22–30.
5. McKinney, W. (2017). "Python for Data Analysis." O'Reilly Media.
6. Jones, E., Oliphant, T., Peterson, P. SciPy: Open Source Scientific Tools for Python, 2023.
7. Интеллектуальные информационные системы и технологии: учебное пособие/ Ю.Ю. Громов, О.Г.Иванова, В.В. Алексеев и др. –тамбов: Издво ФГБОУ ВПО «ТГТУ», 2013.-244с.
8. Поталов А.С. Технологии искусственного интеллекта-СПб: СПбГУ ИТМОб 2010-218 с.

Elektron manbalar:

1. NumPy Developers. "NumPy Documentation." <https://numpy.org/doc/stable/>
2. www.lex.uz - O'zbekiston Respublikasi Qonun hujjatlari milliy bazasi
3. <http://www.raii.org/library> - Российская ассоциация искусственного интеллекта
4. <https://www.javatpoint.com/artificial-intelligence-ai>