

Knihovna NumPy

NumPy (zkratka pro *Numerical Python*) je základní knihovna pro vědecké výpočty v jazyce Python. Je to de facto standard pro manipulaci s daty v oblasti datové vědy, strojového učení a AI.

Proč používat NumPy?

Ačkoliv Python nabízí vestavěné seznamy (lists), pro výpočty v AI jsou nevhodné. NumPy přináší:

- **Rychlost:** Operace jsou implementovány v jazycích C a Fortran, což je řádově rychlejší než standardní Python.
- **Vektorizace:** Umožňuje provádět operace nad celými poli najednou bez nutnosti psát cykly `for`.
- **Paměťová efektivita:** Pole NumPy zabírají v paměti mnohem méně místa než seznamy objektů.

Základní objekt: ndarray

Srdcem knihovny je objekt **ndarray** (n-dimensional array). Jde o homogenní pole, což znamená, že všechny prvky musí být stejného datového typu (nejčastěji `float` nebo `int`).

Ukázka vytvoření polí

```
import numpy as np

# Vektor (1D pole)
v = np.array([1, 2, 3])

# Matice (2D pole)
m = np.array([[1, 2], [3, 4]])

# Tenzor (3D pole - např. 2x2x2)
t = np.zeros((2, 2, 2))
```

Klíčové operace pro AI

Broadcasting (Šíření)

NumPy dokáže pracovat s poli různých tvarů během aritmetických operací. Například přičtení skaláru ke každému prvku matice proběhne automaticky bez cyklů.

Lineární algebra (submodul linalg)

Pro potřeby AI obsahuje NumPy specializovaný modul `numpy.linalg`.

Funkce	Význam
<code>np.dot(a, b)</code>	Skalární součin nebo násobení matic.
<code>np.transpose(a)</code>	Transpozice matice.
<code>np.linalg.inv(a)</code>	Výpočet inverzní matice.
<code>np.linalg.eig(a)</code>	Výpočet vlastních čísel a vektorů.

Praktický příklad: Výpočet v neuronu

V lineární algebře pro AI jsme zmínili aktivaci neuronu jako $z = w \cdot x + b$. V NumPy to vypadá takto:

```
inputs = np.array([1.0, 2.0, 3.0])
weights = np.array([0.2, 0.8, -0.5])
bias = 2.0

# Výpočet jedním příkazem
output = np.dot(inputs, weights) + bias
```

Integrace v ekosystému

NumPy tvoří základ, na kterém staví téměř všechny ostatní knihovny pro AI:

- **Pandas:** Pro analýzu tabulkových dat.
- **Scikit-learn:** Pro klasické algoritmy strojového učení.
- **SciPy:** Pro pokročilé vědecké výpočty.
- **TensorFlow/PyTorch:** Používají tenzory, které jsou přímo inspirovány architekturou NumPy polí.

Tip: Většina metod v AI očekává data ve formátu NumPy polí. Umět efektivně „tvarovat“ pole (pomocí `.reshape()`) je jednou z nejdůležitějších dovedností datového inženýra.

Související témata

- [Lineární algebra pro AI](#)
- [Jazyk Python](#)
- [Knihovna Pandas](#)

From:

<https://www.serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:

https://www.serviceit.cz/doku.php?id=it:python_numpy

Last update: **2026/01/21 09:13**

