July 12, 2023

Contents

1 Setup

Importiamo le librerie qui usate

```
import numpy as np
from scipy import stats
```

2 Algebra lineare con numpy

Per trasporre si usa il metodo `transpose` degli array o l'attributo `T`

```
X = np.arange(12).reshape((3, 4))
X.transpose()
X.T           # usa il metodo swapaxes che è più generale
```

Per il prodotto tra matrici si usa `np.dot` o con la chiocciola:

```
np.dot(X.T, X)
X.T @ X
# anche X.T.dot(X)
```

Altre funzioni utili sono riportate in tabella 1

Funzione	Descrizione
`np.diag`	Return the diagonal (or off-diagonal) elements of a square matrix as a 1D array, or conv
`np.trace`	Compute the sum of the diagonal elements
`np.linalg.det`	Compute the matrix determinant
`np.linalg.eig`	Compute the eigenvalues and eigenvectors of a square matrix
`np.linalg.inv`	Compute the inverse of a square matrix
`np.linalg.pinv`	Compute the Moore-Penrose pseudo-inverse inverse of a square matrix
`np.linalg.qr`	Compute the QR decomposition
`np.linalg.svd`	Compute the singular value decomposition (SVD)
`np.linalg.solve`	Solve the linear system $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$, where $\mathbf{A}$ is a square matrix
`np.linalg.lstsq`	Compute the least-squares solution to $\mathbf{y} = \mathbf{Xb}$

Table 1: Funzioni per algebra lineare