

June 7, 2023

Contents

1	Setup	1
2	Algebra lineare con numpy	1

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

Table 1: Funzioni per algebra lineare