Python

Pokročilé matematické funkce

VŠCHT

2019

Numpy

Numpy - některé další funkcionality

Numpy obsahuje celou řadu matematických funkcí.

- trigonometrické funkce
- hyperbolické funkce
- zaokrouhlovací funkce
- sumy, součiny, diference
- exponenciální funkce a logaritmy
- maticové funkce
- diskrétní fourierovo transformaci
- statistické funkce
- funkce pro práci s náhodnými čísly
- funkce pro řešení rovnic a funkce pro práci s polynomy
- a řadu dalších funkcionalit..

Numpy - trigonometrické funkce

Trigonometrické funkce - přehled

- sinus - $sin(x)$
- kosinus - $cos(x)$
- tangens - $tan(x)$
- arkus sinus - $arcsin(x)$
- arkus tangens - $arctan(x)$
- arkus tangens2 (obdoba atan2) - $arctan2(x, quadrant)$
- arkus cosinus - $arccos(x)$
- převod na stupně - $degrees(x)$
- převod na radiány - $radians(x)$
- výpočet délky přepony - $hypot(x)$

Numpy - trigonometrické funkce - příklad

```
import numpy as np

x = np.pi / 4
x_degrees = 45.0
a = 4
b = 3
print("sin({})={}".format(x, np.sin(x)))
print("cos({})={}".format(x, np.cos(x)))
print("tan({})={}".format(x, np.tan(x)))
print("arcsin({})={}".format(x, np.arcsin(x)))
print("arctan({})={}".format(x, np.arctan(x)))
print("arctan2({})={}".format(x, np.arctan2(x, 0)))
print("arccos({})={}".format(x, np.arccos(x)))
print("{} rad={}   ".format(x, np.degrees(x)))
print("{}   ={} rad".format(x, np.radians(x)))
print("for legs a={}, b={},"
      " the hypotenuse c={}".format(a, b, np.hypot(a, b)))
```

Numpy - hyperbolické funkce

Hyperbolické funkce - přehled

- hyperbolický sinus - $sinh(x)$
- hyperbolický kosinus - $cosh(x)$
- hyperbolický tangens - $tanh(x)$
- hyperbolický arkus sinus - $arcsinh(x)$
- hyperbolický arkus tangens - $arctanh(x)$
- hyperbolický arkus cosinus - $arccosh(x)$

Numpy - hyperbolické funkce - příklad

```
import numpy as np

x = np.pi

print("sinh({})={}".format(x, np.sinh(x)))
print("cosh({})={}".format(x, np.cosh(x)))
print("tanh({})={}".format(x, np.tanh(x)))
print("arcsinh({})={}".format(x, np.arcsinh(x)))
print("arctanh({})={}".format(x, np.arctanh(x / 5)))
print("arccosh({})={}".format(x, np.arccosh(x)))
```

Numpy - zaokrouhlovací funkce

Zaokrouhlovací funkce - přehled

- zaokrouhlování na n-desetinných míst - $around(x, decimals = n)$
- zaokrouhlení k nejbližšímu celému číslu - $rint(x)$
- zaokrouhlení směrem k 0 - $fix(x)$
- zaokrouhlení dolů - $floor(x)$
- zaokrouhlení nahoru - $ceil(x)$
- odstranění desetinného rozvoje - $trunc(x)$

Numpy - zaokrouhlovací funkce - příklad

```
import numpy as np

x = [0.45697, 0.8, 1.56789]

print(np.around(x, decimals=2))  # prints [0.46 0.8  1.57]
print(np.rint(x))  # prints [0. 1. 2.]
print(np.fix(x))  # prints [0. 0. 1.]
print(np.floor(x))  # prints [0. 0. 1.]
print(np.ceil(x))  # prints [1. 1. 2.]
print(np.trunc(x))  # prints [0. 0. 1.]
```

Numpy - sumy, součiny a diference

Sumy, součiny a diference - přehled

- součin prvků pole ve směru zvolené osy - $prod(x, axis)$
- součet prvků pole ve směru zvolené osy - $sum(x, axis)$
- kumulativní součin ve směru zvolené osy - $cumprod(x, axis)$
- kumulativní součet ve směru zvolené osy - $cumprod(x, axis)$
- n-tá diference ve směru zvolené osy - $diff(x, n, axis)$
- gradient N-dimensionálního pole - $gradient(x)$
- integrace ve směru zvolené osy lichoběžníkovou metodou - $trapz(x, axis)$
- vektorový součin dvou vektorů - $cross(x, y)$

Numpy - sumy, součiny a diference - příklad

```
import numpy as np

x = [[0, 1], [2, 3]]
y = [[0, 1], [2, 3],[3, 5], [4, 2]]

print(np.prod(x, axis=0))
print(np.prod(x, axis=1))
print(np.sum(x, axis=0))
print(np.cumsum(x, axis=0))
print(np.cumprod(x, axis=0))
print(np.diff(y, 1, axis=1))
print(np.gradient(y))
print(np.trapz(y, axis=0))
print(np.cross([1, 2], [3, 4]))
```

Numpy - exponenciální funkce a logaritmy

Exponenciální funkce a logaritmy - přehled

- exponenciální funkce - $exp(x)$
- výpočet $exp(x) - 1$ - $expm1(x)$
- výpočet 2^x - $exp2(x)$
- přirozený logaritmus - $log(x)$
- dekadický logaritmus - $log10(x)$
- přirozený logaritmus součtu exponenciel $log(exp(x1) + exp(x2)$ - $logaddexp(x1, x2)$
- logaritmus o základu 2 součtu exponenciel $log(2 * *(x1) + 2 * *(x2)$ - $logaddexp(x1, x2)$
- bonus: kardinální sinus - $sinc(x)$

Numpy - exponenciální funkce a logaritmy - příklad

```
import numpy as np

x = 1
y = 3

print(np.exp(x))
print(np.expm1(x))
print(np.exp2(x))
print(np.log(y))
print(np.log10(y))
print(np.logaddexp(x, y))
print(np.sinc(x))
```

Numpy - některé další funkce

Některé další užitečné funkce - přehled

- diskrétní konvoluce dvou vektorů - $convolve(x1, x2)$
- druhá odmocnina - $sqrt(x)$
- třetí odmocnina - $cbrt(x)$
- druhá mocnina - $square(x)$
- absolutní hodnota - $absolute(x)$
- signum - $sign(x)$
- Heavisideova funkce - $heaviside(x, h)$
- maximum (element-wise) vektorů - $fmax(x1, x2)$
- minimum (element-wise) vektorů - $fmin(x1, x2)$
- nahrazení Nan a inf konečnými hodnotami - $nan\_to\_num(x)$
- jednodimenzionální lineární interpolace - $interp(x, xp, fp)$

Numpy - některé další funkce - příklad

```
import numpy as np

x = np.array([1, 2, 4, 0])
y = np.array([-5, 0, 3, 2])
z = np.array([np.inf, 1, np.NaN, 10])

print(np.convolve(x, y))
print(np.sqrt(x))
print(np.cbrt(x))
print(np.square(x))
print(np.absolute(y))
print(np.sign(y))
print(np.heaviside(y, 7))
print(np.fmax(x, y))
print(np.fmin(x, y))
print(np.nan_to_num(z))
```

Numpy - maticové funkce

Maticové - přehled

- vytvoření matice (Matlab styl) - *matrix*("1 2; 3 4")
- matice bez inicializace hodnot - *empty*(*shape*)
- matice nul - *zeros*(*shape*)
- matice jedniček - *ones*(*shape*)
- jednotková matice - *eye*(n)
- absolutní hodnota - *absolute*(x)
- generování matice pomocí matice - *repmat*(x, m, n)
- vygenerování matice s náhodnými hodnotami z rovnoměrného rozdělení - *rand*(*rows*, *cols*)
- vygenerování matice s náhodnými hodnotami z normálního normovaného - *randn*(*rows*, *cols*)

Numpy - maticové funkce - příklad

```
import numpy as np
import numpy.matlib

matrix_shape = (2, 3)
n = 3
numpy_matrix = np.matrix('1 -2; 3 4')

print(np.empty(matrix_shape))
print(np.zeros(matrix_shape))
print(np.ones(matrix_shape))
print(np.eye(n))
print(np.absolute(numpy_matrix))
print(np.matlib.repmat(numpy_matrix, 2, 3))
print(np.random.rand(2, 3))
print(np.random.randn(2, 4))
```

Numpy - DFT

DFT - přehled

DFFT v numpy používá Cooley-Tukey algoritmus.

- 1D DFFT - $fft(x)$
- 2D DFFT - $fft2(x)$
- 1D inverzní DFFT - $ifft(x)$
- 2D inverzní DFFT - $ifft2(x)$
- ND DFFT - $fftn(x)$
- ND inverzní DFFT - $ifftn(x)$

Numpy - statistické funkce

Vybrané statistické funkce - přehled

- medián ve směru osy - $median(x, axis)$
- průměr ve směru osy - $average(x, axis)$
- směrodatná odchylka ve směru osy - $std(x, axis)$
- rozptyl ve směru osy - $var(x, axis)$
- Pearsonův korelační koeficient - $corrcoef(x, y)$
- kovarianční matice - $cov(x)$
- vzájemná korelace - $correlatev(x, y)$
- histogram z dat - $histogram(x, bins = n)$

Numpy - statistické funkce - příklad

```
import numpy as np

x = np.random.randn(1, 100)
y = np.random.randn(1, 100)
print(np.median(x))
print(np.average(x))
print(np.std(x))
print(np.var(x))
print(np.corrcoef(x, y))
print(np.cov(x))
print(np.correlate(x[:, 0], y[:, 0]))
print(np.histogram(x, bins=5))
```

Numpy - náhodná čísla

Práce s náhodnými čísly - přehled

- číslo z normálního rozdelení - *normal*(*loc*, *scale*, *shape*)
- číslo z rovnoměrného rozdělení - *rand*(*shape*)
- náhodná změna prvků v poli - *shuffle*(*x*)
- číslo z binomického rozdělení - *binomial*(*n*, *p*, *shape*)
- číslo z exponenciálního rozdělení - *exponential*([*scale*, *shape*])
- číslo z Poissonova rozdělení - *poisson*(*lam*, *shape*)
- číslo z Weibullova rozdělení - *weibull*(*a*, *shape*)
- číslo z Gumbelovo rozdělení - *gumbel*(*a*, *shape*)
- číslo z mocninného rozdělení - *power*(*a*, *shape*)
- číslo z Paretova rozdělení - *power*(*a*, *shape*)
- nastavení seedu - *seed*([*seed*]) ,nebo *set_state*(*state*)
- kompletní stav náhodného generátoru (tuple) - *get_state*()

Numpy - práce s polynomy

Práce s polynomy - přehled

- hodnota polynomu v bodě - $polyval(x, c)$
- hodnoty kořenů polynomu - $polyroots(x, c)$
- LS fit polynomu na data - $polyfit(x, y, deg)$
- dělení polynomů - $polydiv(c1, c2)$
- násobení polynomů - $polymul(c1, c2)$
- rozdíl polynomů - $polysub(c1, c2)$
- součet polynomů - $polyadd(c1, c2)$
- umocnění polynomu - $polymul(c, pow)$
- vytvoření polynomu z kořenů - $polyfromroots(roots)$