

NumPyに見る 拡張モジュール作りのポイント

常田 裕士

Pythonで数値計算を行うためのライブラリとして、NumPy⁽¹⁾が使われています。大きなデータの演算を高速に処理できることが特徴です。機械学習、AIの分野で必須と言えるライブラリです。

ここではNumPyの設計をひもとして、ネイティブ実装を使った高速化のために重要なポイントを見ていきます。

● 数値計算ライブラリNumPyとは

NumPyは、Pythonで行列やベクトルを使った数値計算を行うライブラリです。

行列やベクトルの加算、減算、内積、外積といった基本的な計算だけでなく、逆行列や連立1次方程式の求解などの機能があります。

リスト1にベクトルの内積の計算を行う例を示します。

リスト1 NumPyでベクトルの内積の計算を行う例

```
import numpy as np

x = np.array([[1,2],[3,4]]) # 行列
y = np.array([5],[6]) # ベクトル

print(x)
# [[1 2]
#  [3 4]]

print(y)
# [[5]
#  [6]]

print( np.dot(x,y) ) # 行列とベクトルの積を計算
# [[17]
#  [39]]
```

列(N-dimensional array)を表現するデータ構造です。

このndarrayも内部構造は、C言語によるネイティブ実装で作られています。

1：メモリ効率を高めるデータ構造がある

● 数値計算には大規模な多次元配列が必須

NumPyが得意とする数値計算では、多くのケースで巨大な行列やベクトルを扱うため、大規模な多次元配列を効率的に扱えるデータ構造が必要です。

このために、NumPyの基礎となるデータ構造としてndarray⁽²⁾が定義されています。これは、多次元配

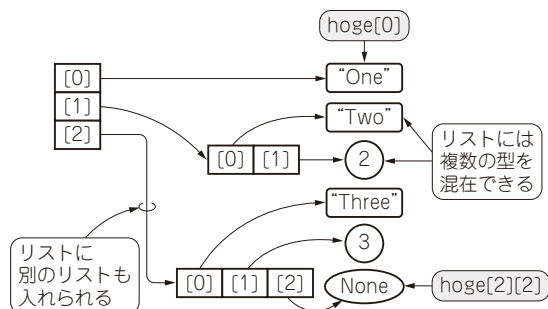
● 効率の悪いPython標準配列のデータ構造

Pythonの言語機能のリストでは、例えば、

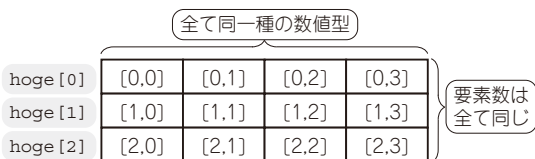
```
[["one"], ["two", 2], ["three", 3, nil]]
```

のように、リストの中に別のリストを持つことができます。またリストの中には文字列と数値のように複数のデータ型を混在させることができます。[図1(a)]。

これは、リストは単なる入れ物であり、中身に対す



(a) Pythonのリストはオブジェクトごとにメモリが取られる



(b) NumPyのndarrayは1つの塊でメモリが確保される

図1 Pythonのリストと比べるとndarrayの多次元配列の方がメモリ使用効率が良い