

科学技術英語 英文読解課題

Ec5 24 番 平田 蓮

付録 A 発展的な NumPy

この付録では、配列計算のための NumPy ライブラリをより深く掘り下げていきます。これには ndarray 型の内部の詳細や、より高度な配列操作とそのアルゴリズムが含まれます。

この付録は雑多な内容を含んでおり続けて読む必要はありません。

A.1 ndarray オブジェクトの内部構造

NumPy の ndarray は、同じ型のデータのブロックを (連続か区切られているかにかかわらず) 多次元配列オブジェクトとして提供します。データ型 (dtype) は、データが浮動小数点、整数、真偽値、あるいはこれまで見てきた他の型のどれであるかを決定します。

ndarray を柔軟にしている理由の一つに、すべての配列オブジェクトがデータをブロック区切りでみていることがあります。例えば配列に対する参照 `arr[:, :2, ::-1]` がデータを一切コピーしていないことを不思議に思うかもしれません。その理由は、ndarray は単なるデータとその型の集まりではなく、配列がメモリ内をさまざまなステップサイズで移動できるようにする「ストライド」情報を持っているからです。より正確に言うと、ndarray の内部は以下のような構成になっています。

- RAM やメモリマップ内にあるデータの集まりへのポインタ
- 配列内の一定個数のデータの型
- 配列の形を表すタプル
- 各次元に沿って要素を一つ進めるために遷移するバイト数 (整数値) のタプル

図 1 は、ndarray の内部構造を示す模式図です。

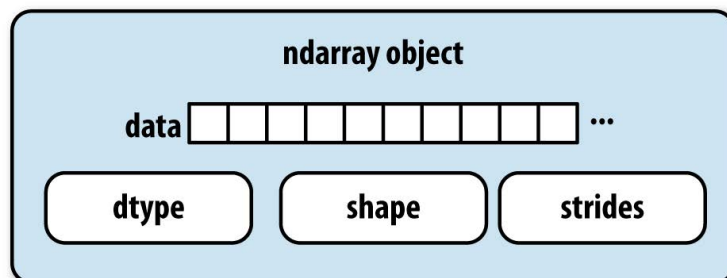


図 1 NumPy の ndarray オブジェクト

例えば、 10×5 の配列は $(10, 5)$ の形を持っています。

```
1 np.ones((10, 5)).shape # (10, 5)
```

float64(8 バイト) 型の典型的な $3 \times 4 \times 5$ 配列は、 $(160, 40, 8)$ のストライドを持っています。(一般的に特定の軸のストライドが大きいほど、その軸に沿って計算を行うコストが高くなるので、ストライドについて知っておくと便利です。)

```
1 np.ones((3, 4, 5), dtype=np.float64).strides # (160, 40, 8)
```

典型的な NumPy ユーザーが配列のストライドに興味を持つことは稀ですが、ストライドは 0 配列をコピー、構築するための重要な要素です。ストライドには負の値を設定することもでき、配列をメモリ上で逆向きに移動させることができます (例えば、`obj[::-1]` や `obj[:, ::-1]` のようなスライスがこれに該当します)。