第5章 行列やベクトルの表現に使われるndarrayを扱う関数を作る

# 画像拡大/縮小の拡張モジュールを作る

常田 裕士

リスト1 numpy.get_include()でインクルートするパスを設定

```
from distutils.core import *
import numpy

print( numpy.get_include() )

setup(name='MyModule',
      version='1.0',
      ext_modules=[Extension('mymodule',
                             ['mymodule.c'])],
      include_dirs = [numpy.get_include()]
     )
```

リスト2 Py_tp_execのスロットの処理

```
static PyObject* mymodule_exec(PyObject *module)
{
        import_array();
        return NULL;
}

static PyModuleDef_Slot MyModuleSlots[] =
{
        {Py_mod_exec, mymodule_exec},
        {0, NULL}
};
```

ネイティブ拡張モジュール作りの例としてNumPyのデータ形式のndarrayを扱う関数を作成します．

NumPyでは行列やベクトルの表現にndarrayを使って，高速な処理を実現しています．

第2章で解説した通り，Pythonによる実装ではndarrayの要素ごとへのアクセスがボトルネックになりがちですが，この部分をネイティブ実装とすることで性能を改善できます．

本章で作るNumPyのデータを扱うネイティブ拡張モジュールも，第4章のMy拡張モジュールと同様の方法で作成できます．

## 事前準備

### ● setup.pyの編集

setup.pyでNumPyのヘッダ・ファイルをコンパイルのオプションに設定します．

メソッドnumpy.get_include()でインクルードのパスが取得できます(**リスト1**)．

NumPyの機能を使うには，NumPyが用意している初期化関数のimport_array()を呼ぶ必要があります．第4章の例と同じように，Py_mod_execのスロットの処理でこれを実行します(**リスト2**)．

## 高速に画像を拡大縮小するモジュールを実装

NumPyで使っているndarrayは，NumPyのソースコードのnumpy/ndarraytypes.hで定義されています．これをインクルードして，引数で渡されるオブジェクトをPyArrayObjectにキャストして処理を行うことができます．

ここでは例として，画像データが格納されているndarrayの単純な拡大縮小の処理を行ってみます．

### ● ndarrayのデータを作る

**リスト3**にソースコードを示します．この中のPyArray_SimpleNewで，新たにndarrayのデータを作ることができます．PyArray_Newに対するマクロになっていて，少ないパラメータの指定で扱えます．PyArray_Newの詳細はNumPyのAPIリファレンスを参照してください．

元の画像から取得した配列のサイズを基に，縦横($X$，$Y$)の要素数を指定した倍率をかけた，画像の入れ物を作ります．

### ● 元画像のデータをコピー

次にndarrayのメモリから直接値を取得し，格納処理をします．

ここでは，画素ごとにメモリのコピーを行いますが，

- dimensions，stridesで表されるサイズ
- データ型
- 配列の格納順序がC形式かFortran形式
- 値のエンディアンはどちらか

など，メモリにどのような形式で値が格納されている