

メカニズムから読み解く Python高速計算テクニック

加藤 公一

本稿について

● ねらい

本稿では、Pythonの数値計算の高速化の手法について、特にC言語などの低級言語との比較で説明していきます。主に題材として機械学習のアルゴリズムを取り上げますが、機械学習に限定せず一般の数値計算をPythonで実装するときに役立つものだと思います。

想定読者としては以下の2通り考えています。

- ① 普段C言語やFORTRANなどの低級言語で数値計算をしているが、Pythonに興味を持ち始めた人
- ② 普段Pythonを使っているが、その中の仕組みに興味を持った人

本誌の主な読者層を考えると、①の方が多そうなので、Pythonをよく知らない人にも配慮しました。また、②の方の人にも配慮して、低級言語の知識がない人でも理解できるように気をつけました。ある程度のPythonの文法の知識があると想定した上で解説しますが、全く知らない人でも追実験をすることで雰囲気はつかめるかと思います。

● 記事の実験環境

本文中で計算速度を測定したベンチマークが幾つか出てきますが、これらの実験はアマゾンAWSが提供するAmazon EC2のt2.microインスタンス上にAnacondaのバージョン3.7をインストールして実行した結果です。t2.microインスタンスとはvCPUを1つ持ち、メモリが1.0Gバイトという比較的安価な仮想サーバのインスタンスです。計算速度は実行環境に大きく依存しますし、同じ環境でもさまざまな外的要因で多少の違いもあります。本稿のベンチマークは、あくまで参考程度の数値ということとらえてください。

● コマンドやプログラムの表記について

LinuxやmacOSで、カレント・ディレクトリのファイル一覧を表示するには、

```
> ls
```

と入力します。このように、本稿ではOSのシェルからの入力は「>」でプロンプトを表すことにします。この場合入力するのは「ls」だけで、「>」は入力しません。

本稿でのPythonのプログラムの動かし方は2つあります。1つはファイルにコードを書いて保存してからそのファイルを指定して実行する方法で、もう1つは対話実行環境(REPL; Read-Eval-Print Loop)によるものです。

```
x.py
```

```
print("!")
```

と書いた場合は、テキスト・ファイルを開いてそこに「print("!")」と書いて、それをファイル名x.pyとして保存せよという意味です。このx.pyがカレント・ディレクトリにあるときはシェルから、

```
> python x.py
```

とすればそれを実行できます。

一方で、REPLを実行するときはシェルから、

```
> python
```

とだけ入力します。するとリスト1のように表示されます。

ここで表示されるバージョンはインストールされているPythonのバージョンにより異なります。ここで表示されている「>>>」がプロンプトで、Pythonのコードを入力待ちしている状態です。

REPLが起動している状態で入力するときは、

```
>>> print("!")
```

のように表記します。これはREPLから「print("!")」と入力せよという意味で、「>>>」は入力しません。

Python環境の準備

Pythonのインストールについてはさまざまな方法があります。普段あまりPythonに慣れ親しんでいない人が本稿のコードを全て試してみたいのであればAnacondaをインストールするのが手っ取り早くて楽だと思います。AnacondaにはPython本体以外にもデータ分析などに必要なライブラリが一通り組み込ま