

計算ライブラリ CuPy で PC の Max 数十倍

AIで注目! GPU超高速Pythonの研究

ご購入はこちら

苅田 成樹

NVIDIA社のGPU (Graphics Processing Unit) 並列コンピューティング・アーキテクチャ CUDAに対応する多次元配列ライブラリにCuPy⁽¹⁾があります。GPUを利用することにより高い演算性能が得られます。Pythonでは定番の数値計算ライブラリNumPyと高い互換性を持ちます。

NumPyの機能は膨大なもので、全てが移植されているわけではありませんが、基本的な多次元配列の要素ごとの演算、線形代数の演算やソートといった、CUDAが得意とするたくさんの処理が実装されています。

本稿では、公式リポジトリの例題⁽²⁾を元に、CuPyの実力やどんなことができるのかを探ってみます。

3.9GHz動作のCore i7とNVIDIAのGPUボード GeForce GTX 1080搭載PCで比べると2~70倍ほど高速化できました(表1)。重たいディープ・ラーニングの計算などもずいぶん高速化できそうです。

性能評価

● 準備1…Python環境の構築

Python環境は、Pythonのディストリビューション Anacondaを使って構築します。以下のウェブ・ページから、各OS対応のインストーラを入手できます。

<https://www.anaconda.com/download>

今回はPython 3.6版を使います。Linuxの場合は、図1の手順で実験用の仮想環境を作ります。

デフォルト設定では、\$HOME/anaconda3以下にインストールされます。環境構築後は、最後の1行を実行すれば今回の環境を呼び出せます。

● 準備2…CuPyライブラリのインストール

次にCuPyライブラリをインストールします。手順を図2に示します。今回は執筆時の最新版のv2.0.0を使っています。

● 準備3…公式リポジトリの例題

CuPyによってどれくらい性能が向上するかを示す最良の例として、公式リポジトリの例題があります。

gitを使ったダウンロード方法を図3に示します。

● 実行

examplesディレクトリには、NumPyと実行速度を比較するコードがいくつかあります(図4)。各ディレクトリにあるPythonコードを実行するだけです。

3.90GHz動作のCore i7-4770KとNVIDIA社のGPUボード GeForce GTX 1080を搭載するPCによる実行結果を表1に示します。GPUの初回実行はコンパイル時間が含まれるので2回目の結果を掲載しています。全体で2~76倍もの高速化ができたことが確認できます。

このようにNumPyが得意な線形代数などの配列操作がでてくる(≠forループを使わない)、機械学習・確率統計のコードは簡単にCuPyで高速化できます。

記述テクニック

この節では、実際にCuPyを使った簡単な関数を書いていきます。そして公式のexamplesにおける高速化も紹介します。

表1 Python科学計算をGPUで行うとMax何十倍もPCより高速にできる
比較対象のPCは3.9GHz動作Core i7搭載。公式リポジトリにある例題の実行結果

名称	内容	時間[s]		性能比[倍]
		NumPy (CPU)	CuPy (GPU)	
gmm	混合ガウス分布による確率モデルの学習	1.87	0.949	1.97
kmeans	K平均法によるクラスタリング	9.98	1.14	8.75
cg	共役勾配法による最適化	32.2	13	2.47
finance	金融で有名な Black-Scholes 方程式の計算	8.66	0.114	75.96