

高位合成ツールを使ってディープ・ラーニングをFPGAに!

ディープ・ラーニング・フレームワークChainerとXilinx社SDSoCを連携 ディープ・ニューラル・ネットワークをFPGAで動かす俺々フレームワークを作る

中原 啓貴 Hiroki Nakahara

ディープ・ラーニングをFPGAに実装するには、ディープ・ニューラル・ネットワークの学習とFPGA実装の両方が必要であり、敷居が高い状態でした。しかし近年、ディープ・ラーニングのフレームワークの充実と高位合成ツールの普及により、FPGAへの実装の敷居は大幅に下がってきています。ここではディープ・ラーニング・フレームワークであるChainerを使って、畳み込みディープ・ニューラル・ネットワーク(CNN)を設計し、Zynq搭載評価ボードに実装する方法を説明します。

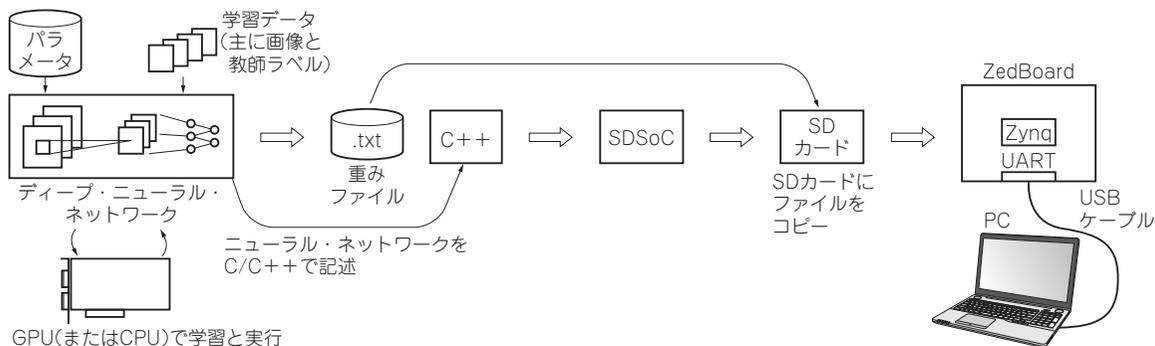


図1 一般的なFPGA向けディープ・ラーニングのフロー

現在は、多種多様なディープ・ニューラル・ネットワークのフレームワークが提供されている。多くはGPUによる学習の高速化や、NumPy形式による学習結果の出力をサポートしている。これらのフレームワークからC/C++を組み合わせて高位合成を使う、またはRTLコードと組み合わせて論理合成してFPGAにポーティングできる

1 ディープ・ラーニングをFPGAにポーティングするために

- アルゴリズム進化が激しいのでC/C++のままFPGAに実装できると良い

筆者はこれまで、ディープ・ラーニングをFPGAにポーティングする方法を紹介してきました^{(1)~(4)}。

前回の記事⁽²⁾で、Chainerで学習した重み係数をPythonのNumPy形式で保存して読み込む方法を説明しましたが、実はTensorFlowやCaffe、Theanoなどのディープ・ラーニング・フレームワークでは、NumPy形式での読み書きが可能です。つまりこの方法は、主流のいろいろなディープ・ラーニング・フレームワークで流用することができます。この記事を参考にして、自作のオリジナル・ディープ・ラーニングFPGAフレームワーク(俺々フレームワーク)を作成することも可能ですので、ぜひチャレンジしてみてください(筆者は現在、2値化CNNフレームワークの開発をしている)。

図1に一般的なFPGA向けフレームワークのフローを示します。まず、学習データとディープ・ニューラル・ネットワークのモデルを用意し、ディープ・ラー

ニング・フレームワークを用いて学習を行います。これまでは全て手で開発していましたが、Pythonベースのフレームワークが多数出てきており、GPUもサポートしていることから、敷居が大幅に下がりました。ディープ・ラーニング・フレームワークの比較を表1に示します。どれもNumPy形式の学習済みデータを出力できるので、自分の好みに合ったフレームワークを選択できます。筆者は新しい論文の成果を再現するために、さまざまなフレームワークを使っています。

学習が済んだモデルはファイルに出力し、高位合成用のC/C++に読み込ませます。HDLでディープ・ニューラル・ネットワークを設計してもよいのですが、設計時間に対するディープ・ラーニングの進化の速度を考えると、現時点では割に合わない状況です。今年も新しいネットワーク・アルゴリズムがどんどん出てきているので、高位合成ツールを使って即開発・即実現が求められると考えています。

- Xilinx社SDSoCを使った事例を紹介

前回の記事ではAltera SDK for OpenCL向けのC++記述を使いましたが、今回はXilinx社のSDSoCを使って実現します。ディープ・ニューラル・ネット