平成 28 年度　情報・通信工学専攻　コンピュータサイエンスコース　修論発表会

FPGA タブレットを用いた人工知能アプリケーションの高速化

電気通信大学 大学院 情報・通信工学専攻 成見研究室
1531048　佐藤知哉

1　はじめに

近年、スマートフォンなどの携帯型端末の普及や、利用方法が多様化したことにより、アプリケーション自体の機能や性能への需要も上がってきている。この問題を解決する一つの方法として、専用の周辺回路やモジュール、チップを搭載するなどの端末自体のスペックや機能を向上する方法が考えられる。また、研究分野では、Deep Learning という多層で構成されたニューラルネットワークによる機械学習方法や、ゲーム AI（Artificial Intelligence）ではモンテカルロ法による精度の向上などで、人工知能分野に注目が集まっている。Deep Learning では、画像や音声などの大規模なデータを扱うことから、CPU だけでの処理能力では不足していることもあり、GPU(Graphic Processing Unit) を用いた高速化を行うことが多い。また、モンテカルロ法では GPU を利用してより多くのプレイアウトを可能にしている。しかし、GPU は携帯端末に搭載されておらず、広く普及しているスマートフォンやタブレット端末が持つ電力スペックでは、実現性は低い。そこで、本研究では FPGA(Field Programmable Gate Array) をベースに実装した Android タブレット端末を用いることで、人工知能アプリの題材としてオセロアプリの AI と Deep Learning の高速化を目指す。

2　既存研究

2.1　Android アプリからの FPGA の利用による処理の高速化

塩谷の研究 [1] では、Xilinx 社の Programmable SoC を搭載した FPGA 評価ボード ZedBoard [2] をベースにした Android タブレット（以下、FPGA タブレット）の開発を行っている。FPGA チップ上の ARM コアで Android OS を動作させることにより、図 1 のようなタブレット端末として利用できる。また、ダイナミックパーシャルリコンフィギュレーション機能を用いることで、OS の動作中に FPGA の回路記述の一部を再構成することが可能である。これにより、アプリケーションごとに専用回路を用意し、それらを必要に応じて書き換えていくことで、アプリケーション全体の高速化を目指している。

このタブレット端末システムはハードウェア部、ソフトウェア部、ソフトハード部の 3 つの要素で構成されている。本研究では、このタブレット端末を用いて、それぞれの人工知能に対して専用回路を開発し、アプリケーションの高速化を行う。

図 1　FPGA タブレット

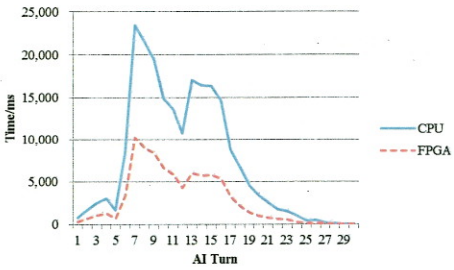

図 2　深さ 5 の測定結果

3　オセロゲームの高速化

Wong らの研究 [3] では、オセロのゲーム AI を FPGA で実装することに成功している。この研究では α-β 法という探索手法を FPGA で実装している。本研究では、できるだけ小規模な回路開発を行うことに焦点を当て、α-β 法よりも簡単な min-max 法によるソフトウェアでの AI の実装と、評価関数の一部である、盤面の着手可能手の列挙処理の専用回路化を行った。その結果を図 2 に示す。探索木の深さが 5 の場合、CPU と比較して約 2.3 倍の高速化に成功した。また、さらなる高速化をめざし、木探索アルゴリズムを α-β 法に置き換えて実装した結果、最大で Min-Max 法の CPU の場合と、α-β 法かつ FPGA を用いた場合の比較で 10.92 倍の高速化となった。

4　Deep Learning による画像分類アプリの高速化

Deep Learning のひとつとして、畳み込みニューラルネットワークと呼ばれる深層学習方法がある。Deep Learning を専用ハードウェアで高速化した例として、Microsoft 社の FPGA による畳み込みニューラルネットワーク [4] が挙げられる。この研究では GPU よりもワットあたりのスループットがよいことを示している。

本研究では、畳み込みニューラルネットワークを Android

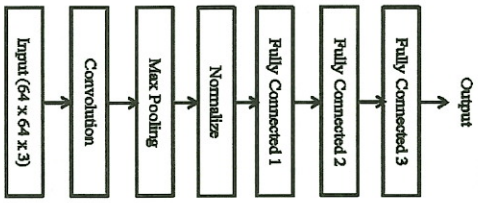

図 3　実装する畳み込みニューラルネットワーク

アプリケーションとして実装し、その一部の処理を専用回路化する。

4.1　実装する畳み込みニューラルネットワーク

実装する畳み込みニューラルネットワークを図 3 に示す。FPGA タブレット上の Android OS が管理するメモリの都合により、小規模なニューラルネットワークにしてある。各層の活性化関数には出力層を除いて正規化線形関数を選択している。出力層にはソフトマックス関数を用いており、出力結果は各カテゴリへの分類結果である。また、多クラス分類問題を想定し、誤差関数には交差エントロピーを用いた。プーリング層では最大プーリングを採用した。

4.2　順伝播の計算時間モデル化

畳み込み層の順伝播に関する処理時間は式 (1) となる。

$$T = T_{CPU} + T_{FPGA} + T_{COM} \tag{1}$$

ここで、T_{CPU},T_{FPGA},T_{COM} はそれぞれ、CPU での処理時間、FPGA での処理時間、CPU-FPGA 間のデータ転送時間を指す。T_{CPU} は畳み込み関数内で FPGA に計算させる箇所をコメントアウトしてそれ以外の処理時間を測定することで求める。具体的には、活性化関数の適用の処理がある。T_{COM} は要素数 N の float 型配列を FPGA に転送する時間を測定することで求める。T_{FPGA} はアルゴリズムから式 (2) のように見積もる。

$$T_{FPGA} = \frac{T_c}{120} \times N_f \times W_{out} \times H_{out}(C \times W_{fil} \times H_{fil} + 1) \tag{2}$$

但し、T_c はクロック周期、N_f は畳み込むフィルタ数、W_{out}, H_{out} は出力されるデータの幅と高さ、C は入力チャネル数、W_{fil}, H_{fil} はフィルタの幅と高さである。

T_{CPU},T_{COM} について FPGA タブレット上で測定した結果と、T_{FPGA} の推定値から、計算モデルによる処理時間は 1.615ms となった。また、CPU のみでの畳み込み層の順伝播の計算時間は 52.367ms であった。このことから FPGA による部分専用回路化を行うことで約 32 倍のスピードアップが期待される。また、活性化関数の正規化線形関数も回路に組み込むと高々 3,072 要素の符号検査で済むため、約 72 倍の高速化が期待できる。ただし、実際の回路の設計に依存する部分が大きいため、この見積もり値よりも効率が悪くなることが予想される。

4.3　専用回路の設計

順伝播の専用回路の作成は Xilinx 社の Vivado HLS[5] を用いた。C++ で書かれた順伝播のプログラムから回路の生成を行った。ツールが吐き出した回路を用いた場合に、専用回路への転送時間、CPU 側での処理時間等を含めた畳み込みニューラルネットワークの順伝播全体でかかる計算時間を表 1 に示すこれより、Vivado HLS で作成した回路を用いた場合、CPU のみの場合と比較して最高で 2.938 倍の高速化が望めることになる。

表 1　順伝播処理全体でかかる計算時間

	min	max
T_{CPU}	0.862ms	
T_{COM}	0.725ms	
回路のレイテンシ	16.236ms	609.01ms
T	17.827ms	610.597ms

5　今後の課題

FPGA タブレットの仕様上、データ転送数が小さければ通信のオーバヘッドにより CPU だけで実行する場合よりも遅くなる可能性がある。この場合に対応するために、PIO を利用してアプリケーションと回路間を直接データのやり取りする方法がある。FPGA チップの Zynq には CPU から直接回路領域へと繋がる ACP バスが存在するため、このバスを利用することで実現できると思われる。

人工知能のような大規模な計算を行う場合の高速化手法として、本研究では FPGA を用いたが、専用回路は規模に比例して複雑になっていく。VivadoHLS のようなソフトウェアから HDL を生成できるツールも存在するが、簡単に HDL を得られる代わりにその性能はそこまで高いものではない。より高い効果を得るためには細かいチューニングが必要である。

参考文献

[1] 塩谷丈史, 成見哲, “FPGA タブレットによるモバイルアクセラレータ”, 情報処理学会研究報告, Vol. 2015-HPC-148, No. 18, 2015

[2] ZedBoard, http://zedboard.org/product/zedboard ,2016 年 6 月 25 日

[3] Wong, C. K., Lo, K. K. ; Leong, P. H. W., *”An FPGA-based Othello endgame solver”*, Proceedings. 2004 IEEE International Conference on Field- Programmable Technology, pp. 81-88, 2004

[4] Kalin Ovtcharov, Olatunji Ruwase, Joo-Young Kim, Jeremy Fowers, Karin Strauss, Eric Chung, *”Accelerating Deep Convolutional Neural Networks Using Specialized Hardware”*, Microsoft Research White Paper, February 23, 2015

[5] Vivado HLS, https://japan.xilinx.com/products/design-tools/vivado/integration/esl-design.html