(402) アーキテクチャ

FPGA を用いた CNN のハードウェア実装

Hardware implementation of CNN using FPGA

西森 祐介† 森下 賢幸† 小椋 清孝† 伊藤 信之†
Yusuke Nishimori† Takayuki Morishita† Kiyotaka Komoku† Nobuyuki Itoh†
†岡山県立大学大学院 情報系工学研究科

1 研究背景・研究目的

CNN（Convolutional Neural Network，畳み込みニューラルネットワーク）は画像認識において高い性能を示すが，大規模なデータを扱うと回路規模が大きくなり認識や学習に時間がかかる．ここで，CNN を FPGA 上でハードウェア化することにより，パイプライン処理や並列処理が可能となり，処理時間が短縮され高速化できる可能性がある．そこで本研究では，CNN の認識回路と，学習回路を設計し高速化を図った．開発環境は Xilinx 社の ISE Design Suite 14.7 を使用し，設計言語には Verilog HDL を用いた．

2 CNN 認識回路の設計

Xilinx Virtex-5 XC5V-LX220 を対象として実装設計した．初期ターゲットとした CNN 認識回路を図 1 に示す．全結合層はシナプスとシグモイド関数で構成されたニューロンであり，その回路化を行った[1]．CNN の中間層と出力層にはニューロンを用い，入力層は畳み込み層とプーリング層を回路化した．これらを用いて中間層 3 ニューロン，出力層 1 ニューロンの CNN 全体をハードウェアに実装した．

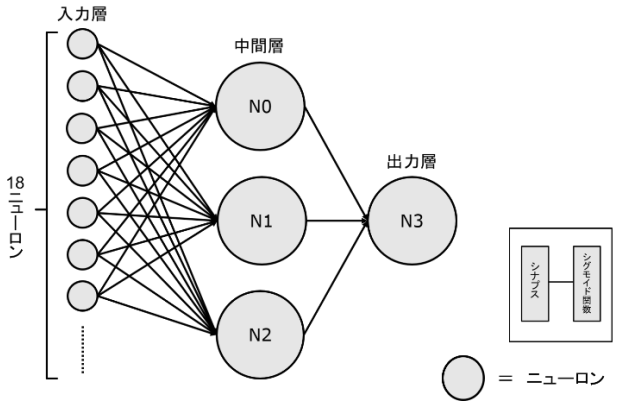

図 1 設計した CNN 認識回路

中間層と出力層のニューロンの構成要素であるシナプスとシグモイド関数はルックアップテーブル方式を用いて設計した．これにより，あらかじめメモリに保存した値をカウンタを用いて順に出力するように設計した．シグモイド関数は中間値も求められるように線形補間多項式を回路設計した．入力層の構成要素である畳み込み層は，画像とフィルタを重ね合わせ，重なり合う画素どうしの積を求め，フィルタ全体でその積の合計値を求める．畳み込み層の出力データがプーリング層の入力データとなる．プーリング演算では最大プーリングと呼ばれる計算手法を用いる．これは入力画像にフィルタをかけ，その中の最大値を出力する．畳み込み演算の入力データは 11×11，フィルタは 3×3 を用いた．プーリング演算の入力データは 9×9，フィルタは 3×3 を用いた．

3 出力層の重み学習回路の設計

出力層の重み学習回路のブロック図を図 2 に示す．

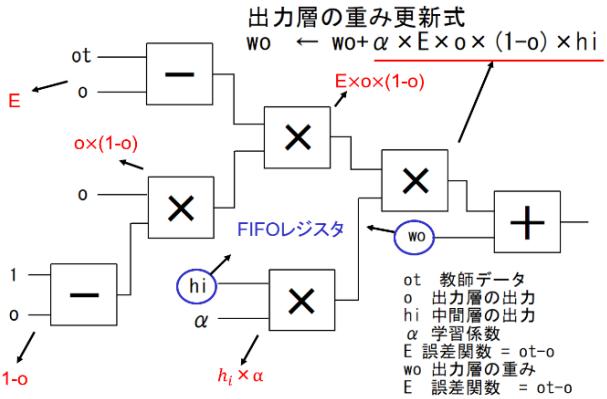

図 2 出力層の重み学習更新式

出力層の重み学習更新式では，計算順序が正しくなるように，四則演算器を用いて学習式を計算する．図 3 の左下の減算器を例とすると，片方の入力ポートに 1 を入力し，もう片方の入力ポートに出力層の出力 o を入力することにより，1-o を計算して出力するような仕組みである．この図 3 の入力ポートh_iには中間層の出力 3 個の値と閾値の計算に用いる値 -1 の合計 4 つのデータが入力される．よって，CNN 認識回路のh_i(中間層の出力)をそのまま渡す．しかし，このままだと認識回路から学習回路へと送られるデータの学習計算と，このh_iに値が入力されるタイミングにかなりの時間差が生じてしまい，正しく

計算処理が行われない．従って，この処理の時間差をなくす改善案として，FIFO(first in first out)レジスタを利用することが考えられる．これは，データ処理において，先に入ったものを先に処理して出し，後に入ってきたものは先に入ったものより後から処理して出すレジスタである．入力ポートw_oについても同様に FIFO レジスタを使用する．

4　シミュレーション結果

CNN 認識回路，出力層の重み学習回路の論理シミュレーションを行い，正しく動作するか確認した．畳み込み層への入力値は，図 1 に示す上側の畳み込み層を畳み込み層 1，下側の畳み込み層を畳み込み層 2 とする．畳み込み層 1 には 11×11 の中央の列の値を全て 1 とし，その他の値を全て 0 とする．フィルタに関しても同じく中央の列を全て 1 としその他の値を 0 とする．畳み込み層 2 には，入力データは畳み込み層 1 と同じものを用い，フィルタを中央の行を全て 1 としその他の値を全て 0 にしたものを用いた．シミュレーション環境は ISim を用い，「Behavioral」を選択してシミュレーションを行った．シミュレーション結果を図 3 に示す．

	シナプス出力結果		シグモイド関数出力結果	
ニューロン名	理論値	シミュレーション値	理論値	シミュレーション値
N0(中間層)	54892	54892	4000	4000
N1(中間層)	9e28a	9e28a	4001	4001
N2(中間層)	a5c2f	a5c2f	4001	4001
N3(出力層)	634406bd	634406bd	4000	4000

図 3 CNN 認識回路シミュレーション結果

中間層と出力層のニューロンを構成するシナプスとシグモイド関数の出力結果と理論値が一致した．よって正しく動作しているといえる．同様に，出力層重み学習回路が正しく動作していることを確認した．

5　回路規模･遅延時間

設計した CNN の各構成要素それぞれの bit 精度を図 4 に示し，回路規模及び遅延時間を図 5 に示す．現段階では全体の回路規模及び遅延時間は測定できていないため，設計した各構成要素ごとの回路規模及び遅延時間のみ示す．図 5 の評価対象の※で示すものはクロック数を 30mHz とし，※※で示すものは 38mHz としてシミュレーションを行った．回路規模の評価方法は Design Summary の「Number of Slice Registers」，「Number of Slice LUTs」の数値とし，遅延時間は「Post-Route」を選択してシミュレーションを行った．

畳み込み処理回路	4bit
最大プーリング処理回路	4bit
中間層シナプス回路	64bit
中間層シグモイド関数回路	16bit
出力層シナプス回路	64bit
出力層シグモイド関数回路	16bit
出力層重み学習回路	32bit

図 4 各構成要素の bit 精度

図 4 の数値は各回路の出力データの bit 精度である．各回路は固定少数点数で演算を行う．シナプスの演算ビット精度が他と比較して大きい値となったが、計算には 24bit あれば十分なため，最適化を検討している．

評価対象	遅延時間	LUTs	Register
畳み込み層	127.0ns	4536	0
プーリング層	17.6ns	333	0
中間層シナプス※	685.7ns	1240	264
出力層シナプス※※	297.5ns	1201	288
シグモイド関数※	389.3ns	236	196

図 5 各構成要素の回路規模と遅延時間

畳み込み処理，プーリング処理に比べてシナプス，シグモイド関数の計算時間が長くなった．回路規模は積和演算を並列処理しているシナプスが大きくなった．

6　結論

中間層数 3，出力層数 1 の CNN 認識回路を回路設計し，シミュレーションを行い正しく動作していることを確認した．設計した CNN の各構成要素の回路規模と遅延時間を求めた．なお，CNN の中間層の重み学習回路についても現在設計中であり，完成後にこの CNN 全体の速度評価を行う予定である．今後の課題は，回路の最適化，回路規模の評価，手書き数字認識などの大規模 CNN へ対応させることである．

参考文献

[1]小高知宏，“機械学習と深層学習”，オーム社 (2016).