1P-05

Genetic Algorithm に基づく Binary Weight 可変構造 DNN の高速学習ハードウェア

高木 俊平† 渡部 直弘† 大塚 卓哉†† 北澤 仁志†

†東京農工大学 ††日本電信電話株式会社 NTT 先端集積デバイス研究所

1 はじめに

近年，Deep Neural Network (DNN) が画像認識やデータ解析に用いられ，好成績を残している．DNN のノード間の weight(重み) をバイナリ値とした BinaryConnect[1] が提案され，ハードウェア実装した際に乗算器が不要となるため組み込み機器向け等の小型省電力な回路の実現が期待できる．しかしバイナリ値 DNN の学習ではバックプロパゲーション法 (BP) は必ずしも有効ではない．我々は Genetic Algorithm (GA) による DNN の最適化 (以下では GADNN と呼ぶ) を試みているが，DNN を GA で最適化する場合には個体毎の評価値算出のため多量のフォワード計算が必要となる [2]．そこで本研究では GADNN のフォワード計算を高速に実行するための FPGA を用いたハードウェアを提案する．提案ハードウェアは DNN の weight と構造を変化させる機能を有する．パイプライン型アーキテクチャによりスループット 1 data/clock のハードウェアを実現した．

2 フォワード計算による DNN 最適化

本研究で対象とする DNN はノード同士が全接続された 3 層以上の DNN である．GA においては DNN の weight 値とネットワーク構造を遺伝子として扱う．weight 値はバイナリの± 1 に加えて，weight を間引くことと同義である 0 を含めた 3 値である．図 1 に示すように，ネットワーク構造は予め定めた上限ノード数からノードを drop する (削除する) ことで変化させる．なおノードの drop は入力層にも適用する．GADNN では各世代で生成された個体ごとに学習データを用いてフォワード計算を行い DNN を評価する．本稿では GADNN にハードウェアを用いる手法を取り扱うが，Particle Swarm Optimization (PSO) など，他のフォワード計算による評価値の算出が必要な最適化手法にも提案ハードウェアは有効である．また BP を用いないので，AND や OR などの論理演算も DNN 中に取り入れることが出来る．

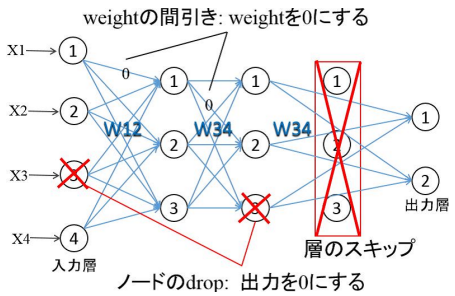

図 1 DNN の weight と構造の最適化

FPGA-based Learning Hardware of Binary-Weight Variable-Structure DNN based on Genetic Algorithm

† Shunpei Takaki, Naohiro Watanabe , Hitoshi Kitazawa (Tokyo University of Agriculture and Technology)

†† Takuya Otsuka (NTT Device Technology Labs, NTT Corporation)

3 ハードウェアアーキテクチャ

3.1 全体構成

提案するハードウェアは以下の 3 つの特徴を持つ．

(1) スループット 1 data/clock のパイプライン型．

(2)weight 値が± 1,0 のため乗算器が不要．

(3)DNN の構造と weight 値を変更可能．

図 2 にハードウェアのモジュール構成を示す．ホスト PC と FPGA ボードは Gigabit Ethernet インターフェース (GbE) を用いて通信を行う．FPGA ボードの制御と状態の監視には Command レジスタと Status レジスタを用いる．学習データ，教師信号，weight は GbE を通してメモリに格納される．ハードウェア中の演算は全て 16-bit 固定小数点数で行う．

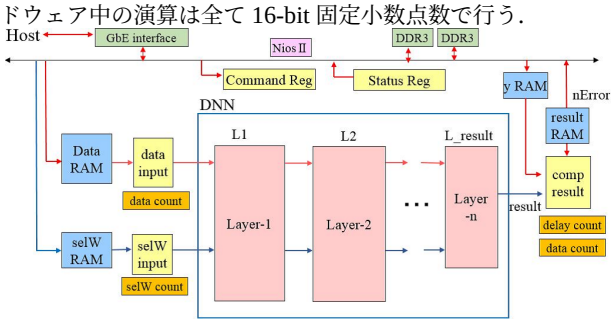

図 2 ハードウェアのモジュール構成

3.2 メモリ

図 3 にメモリの構成を示す．selWRAM には GA の 1 世代で評価する個体数分の weight を格納する．dataRAM と yRAM には学習データとそれに対応する教師信号が学習データの個数分格納される．

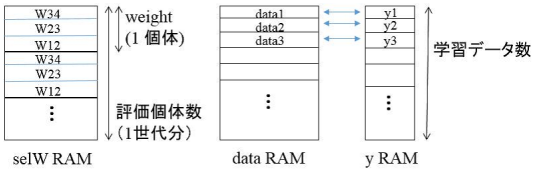

図 3 各メモリのデータ内容

3.3 動作フロー

図 4(a) に GA における提案ハードウェアの動作フローチャートを示す．学習データと教師データは GA を開始した際にハードウェアに転送される．ホスト PC で weight が生成されるとハードウェアに転送され，動作を開始する．weight は DNN 回路に送り込まれ，DNN のフォワード計算によりエラー率を算出する．ハードウェアが受け取った weight を全て評価し終えホストに結果を返すと，ホストで新しい weight が生成され再度ハードウェアへ転送される．また，ハードウェアはホストから命令を受け取り動作を開始すると，図 4(b) に示す状態遷移図に従って動作する．なお，新しい weight の生成のための計算ステップ数は DNN の評価に比べて極めて小さいため，ARM 等のコアプロセッサで可能であるが，実験段

階ではアルゴリズムが確定していないためホストで実行した．

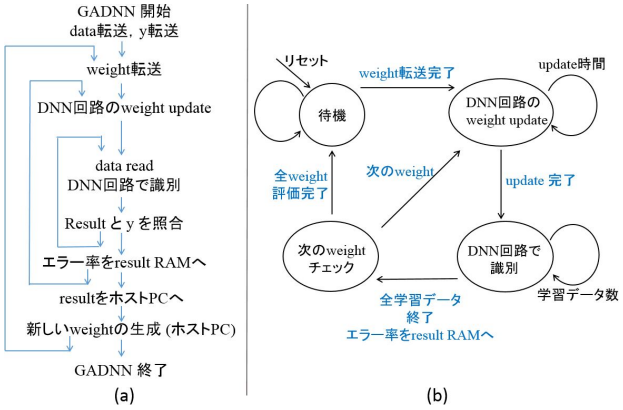

図 4　動作フローチャートと状態遷移図

3.4　パイプライン型 可変構造 DNN 回路

図 5(a) に DNN の 1 層分の構造，図 5(b) に対応するハードウェア回路を示す．この回路は式 (1) に示す積和演算を行う．ここで m, n は第 l 層と第 $l+1$ 層のノード数である．なお weight w はバイナリ値である．

$$y_j = \sum_{i=1}^{m} w_{ij} \cdot x_i \qquad (j = 1, \cdots, n) \tag{1}$$

1 層分の回路は Processing Element (PE) のアレイ構造であり，1 個の PE が 1 個の weight を扱う．PE アレイには chainIn 信号，selW 信号と第 l 層へのデータが入力され，第 $l+1$ 層のノード値を出力する．l 層の入力データは，入力層の場合 DNN への学習データであり，それ以降の層では前の層から出力されたノード値である．DNN 回路へは 1 クロック毎に 1 つの学習データを入力する．

chainIn 信号はシフトレジスタにより隣接 PE へ送られ，PE は weight 値に応じて入力データを chainIn に加減算することで式 (1) の積和演算を求める．適切なタイミングで入力データに対し積和演算するために，入力層以外のデータ入力回路には chainIn のレイテンシに応じた遅延用 FIFO を設ける．入力層に FIFO が不要であるのは，データが時系列信号の場合すべての入力回路に同一値を入力すればタイミングが合い，並列入力データの場合でもデータを予めずらしておけば良いからである．

selW 信号はシフトレジスタにより更新する．selW 信号には 4 ビットの weight 値が乗せられ，PE は weight の値に応じて加算，減算，チェイン出力をゼロにする (ノードを drop) 等の分岐処理を行う．

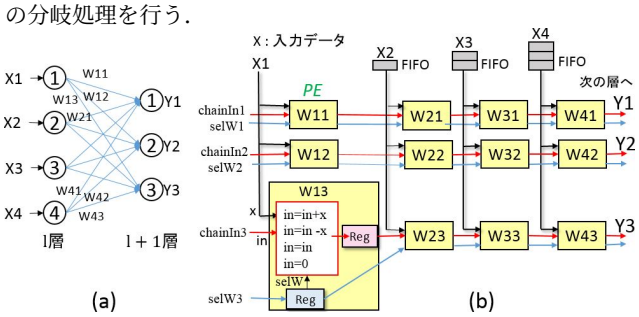

図 5　パイプライン型の積和演算回路

図 6 に DNN 構造を変化させる例を示す．weight 値による PE の制御によって構造を変えることが可能になる．

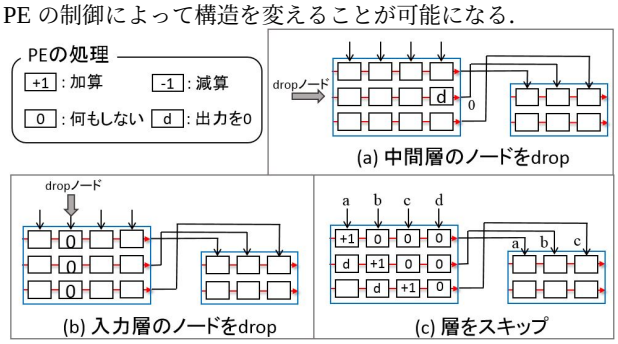

図 6　PE の weight 値による DNN 構造の変化

DNN の最終層には図 7 の最大値選択用回路を用いる．最終層のノードから最大値を選択し，ノード番号を識別結果として出力する．識別結果は教師信号と比較して正解 or 不正解の判断をするが，DNN 回路へのデータ入力から出力までのレイテンシは構造に関わらず一定のため，DNN 回路のレイテンシのタイミングで入力データに対応する教師信号と比較する．

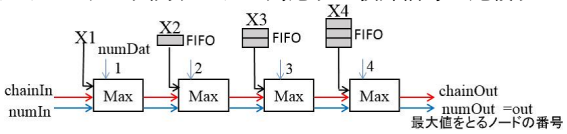

図 7　DNN 最終層の最大値選択回路

4　実装結果

提案するアーキテクチャを Altera Stratix V 5SGXEA7 ボードに実装した．

4.1　資源使用量

提案ハードウェアは PE に乗算器を用いないため，実装する DNN のサイズは主にレジスタ数によって制限される．表 1 にネットワークサイズとレジスタ，メモリ，ALM の各資源使用量と動作周波数の関係を示す．なお動作周波数はハードウェアから GbE を除いたときの値である．

表 1　資源使用量と動作周波数

ネットワーク	Reg.	M20K	ALMs	動作周波数
128-32-2 (3 層)	104,389	2097	39,106 (17%)	389 MHz
128-64-64-2 (4 層)	274,452	2097	107,357 (46%)	260 MHz
128-64-64-64-64-2 (6 層)	436,237	2344	188,408 (80%)	212 MHz

4.2　実行速度

提案ハードウェアはスループット 1 data/clock のため，実行速度は動作周波数と DNN 回路のレイテンシによって決まる．表 1 の 128-64-64-64-64-2(6 層) のネットワークにおいて，学習データを 8000 個，1 世代の評価個体数を 300 個とした場合，実行速度は約 83 世代/sec (25000 個体/sec) となる．

5　まとめ

FGPA を用いて，GA による DNN 最適化を高速に行うハードウェアを提案した．

参考文献

[1] M. Courbariaux, et al.,”BinaryConnect: Training Deep Neural Networks with binary weights during propagations,” Advances in Neural Information Processing Systems, 2015.

[2] 渡部ら ,”Forward 計算と Genetic Algorithm による Binary Weight 可変構造 DNN の最適化”, 情処全大 79 回,2017.