2 値化ニューラルネットワークにおける並列ポップカウンタの効率的 FPGA 実装

Efficient FPGA Implementation of Parallel Popcounters in Binarized Neural Networks

谷川 貴弘 [1]　迫 真太郎 [1]　石浦 菜岐佐 [2]
Takahiro Tanigawa　Sintaro Sako　Nagisa Ishiura

関西学院大学 理工学部 [1]
School of Science and Technology, Kwansei Gakuin Univ.

関西学院大学 工学部 [2]
School of Engineering, Kwansei Gakuin Univ.

1　はじめに

2 値化ニューラルネットワーク (BNN) [1] は, ニューロンの入出力と重みを 2 値に制限したものであり, コンパクトなハードウェア実装を可能にする. BNN のニューロンにおけるポップカウンタを組合せ回路で実装する方法としては全加算器 (3-2 reducer) による Wallace 木 [2] の構成が考えられるが, FPGA 実装では必ずしも LUT の入力数を活かしきれない. 本稿では, reducer の入力を大きくすることにより並列ポップカウンタの LUT 数を削減する手法を提案する.

2　BNN のニューロンと並列ポップカウンタ

BNN におけるニューロンの構成例を図 1 に示す. ニューロンへの入力 +1, −1 をそれぞれ 1, 0 で符号化すると, 乗算は排他的論理和否定 (XNOR) 演算に置き換えられ, ニューロンの出力はその和を閾値と比較することにより決定される. ポップカウンタは与えられた N ビットの和 (の 2 進表現) を求める回路であり, 本稿では計算を 1 サイクルで行う並列ポップカウンタを考える.

ポップカウンタの全加算器による構成例を図 2(a) に示す. N ビットを 3 ビットずつ全加算器で 2 桁の 2 進数にする. 得られた中間和の各桁について同様の操作を繰り返すことにより最終的な結果を求める. 近年の FPGA の LUT の入力数は 3 より大きいので, 全加算器を LUT にマッピングすると LUT の能力が活かしきれない.

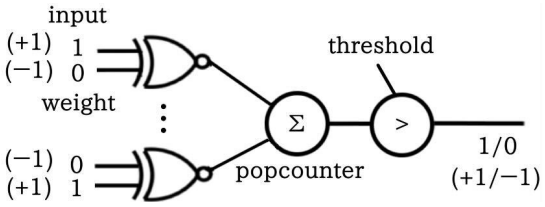

図 1: BNN のニューロンの構成例

3　並列ポップカウンタの効率的な構成

本稿では, 全加算器 (3-2 reducer) よりも大きな入力数の reducer によってポップカウンタを構成する. 6 ビット入力の和を 3 桁の 2 進数で出力する 6-3 reducer で並列ポップカウンタを構成した例を図 2(b) に示す.

ただし, 図 2(c) に示すように, 入力からの段数が 1 つ大きい中間和を同じ reducer の入力にすると, 回路の段数が増えてしまう. そこで図 2(d) のように段数が同じで桁の異なる中間和を reducer の入力とする. 出力数は 4 以上になることがあるが, 回路の段数は抑制できる.

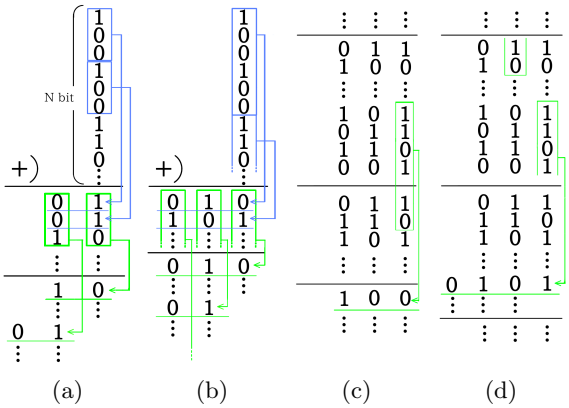

図 2: Reducer によるポップカウントの計算

表 1: 論理合成結果

N		64	128	256	512	1024	2048
LUT 数	単純記述	78	153	319	700	1,444	3,002
	全加算器	67	141	306	634	1,157	2,460
	本稿 1	69	128	265	540	1,106	2,164
	本稿 2	65	135	270	540	1,059	2,082
遅延 (ns)	単純記述	4.50	5.65	6.71	7.78	8.98	9.95
	全加算器	4.78	6.20	7.05	8.39	9.98	11.81
	本稿 1	4.49	5.39	7.08	8.48	10.82	11.80
	本稿 2	4.15	5.14	6.16	7.14	8.07	8.66

Synthesizer: Xilinx Vivado (2020.2), Target: Xilinx Artix-7

4　実装結果

提案手法に基づき, 並列ポップカウンタを Verilog HDL で設計した. 論理合成結果を表 1 に示す. “単純記述” は和の計算を単純に式で記述したもの, “全加算器” は図 2(a) の手法, “本稿 1” は図 2(b) の手法, “本稿 2” は図 2(d) の手法である. “本稿 1” は LUT 数は削減しているが遅延時間が増大している. これに対し “本稿 2” は LUT 数と遅延時間の両方を削減している.

5　むすび

本稿では, 効率的な並列ポップカウンタの FPGA 実装法を提案した. XNOR 演算や閾値との比較までを含めたニューロン全体の効率的な実装が今後の課題である.

参考文献

[1] M. Courbariaux, et al.: “Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1,” *Computer Research Repository*, arXiv:1602.02830 (Mar. 2016).

[2] S. C. Wallace.: “A suggestion for a fast multiplier,” *IEEE Trans. Electronic Computer*, vol. EC-13, issue 1 (Feb. 1964).