16bit 浮動小数点演算を用いたハードウェアベースのニューラルネットワークモデルの実装と検証

1250029　大西　泰雅　（Soft Intelligent System on Chip 研究室）
（指導教員　星野　孝総　教授）

1. 目的

近年、ニューラルネットワークの計算負荷増大に伴い、専用ハードウェアを用いた高速化が求められている。特に、FPGA（Field Programmable Gate Array）は、並列処理能力と柔軟な設計が可能であることから、ニューラルネットワークのリアルタイム推論への適用が期待されている[1]。しかし、FPGA 上での演算はリソース制約が厳しく、特に浮動小数点演算の精度と計算負荷のバランスを取ることが課題となる。本研究では、FPGA 上で 16 ビット浮動小数点演算を用いたニューラルネットワーク実装を行い、特に活性化関数であるシグモイド関数の計算負荷を軽減するために 6 次多項式近似を適用した。本研究の目的は、FPGA 上でニューラルネットワークの効率的な実装手法を確立し、推論精度・計算遅延・リソース使用量の観点から評価することである。

2. 研究内容

本研究では、FPGA 上でのニューラルネットワーク演算を効率化するために、IEEE 754 16 ビット浮動小数点演算を採用し、加算および乗算の基本モジュールを Verilog HDL で設計した。特に、活性化関数として用いられるシグモイド関数の計算負荷を軽減するために、NumPy を用いて最小二乗法により 6 次多項式近似を行い、その係数を用いて FPGA 上で計算を行った。　研究対象として、2 入力 XOR 問題、4 入力 XOR 問題、SIN カーブ近似問題を設定し、それぞれのニューラルネットワークをオフライン学習し、学習済み重みとバイアスを FPGA 上に実装して推論を実行した。FPGA 上の演算結果は GPIO ピンを通じて出力し、ロジックアナライザを用いて計測した。計測結果について推論精度・レイテンシ・リソース消費量を評価した。図 1 におおまかな実験手順を示す。

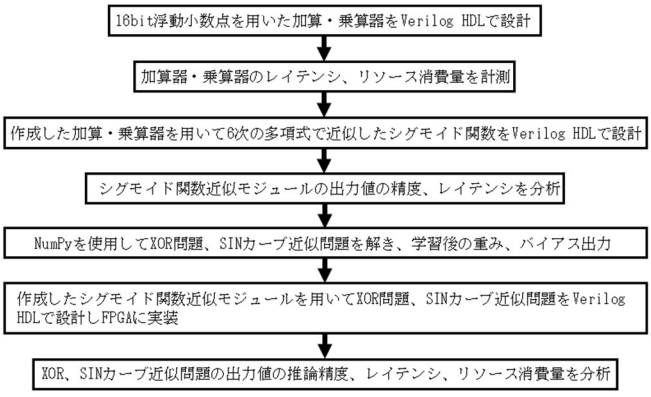

図 1: 本研究の実験手順

3. 結果

実験の結果、加算・乗算モジュールは表 1 のような結果となり、計算に 4 クロック消費していることが分かった。

表 1: 加算・乗算モジュールのリソース消費量とレイテンシ

モジュール	LE（ロジックエレメント）数	Latency（クロック）
加算モジュール	212	4 クロック
乗算モジュール	99	4 クロック

次に、6 次多項式近似を用いたシグモイド関数近似モジュールを評価した結果、図 2 のような結果となり、高い制度での近似結果は得られたが、計算遅延が平均 24.2 クロックと大きい結果となった。2 入力 XOR 問題では期待される出力とほぼ一致する結果が得られ、理論値と比較して高い推論精度が確認された。一方、4 入力 XOR 問題では、(1,1,1,1) 入力パターンにおいて誤差が大きくなる傾向が見られた。これは、NumPy での学習結果段階で出力値を理想的な値に近づけられなかったことが原因である。FPGA 出力結果と理論値との平均二乗誤差（MSE）、平均絶対誤差（MAE）、最大誤差（Max Error）を表 2 に示す。

表 2: FPGA 出力結果と理論値との誤差

problem	MSE	MAE	Max Error
2input XOR	0.005	0.067	0.077
4input XOR	0.061	0.076	0.989

SIN カーブ近似問題では、入力値のスケーリングを適用し、学習済みモデルと FPGA 出力値が概ね一致したが、理想 SIN カーブとの決定係数R^2は 0.866 となり、数値誤差の影響が見られた。結果（点:FPGA 出力結果、線:NumPy での学習結果）を図 3 に示す。

レイテンシ測定、リソース使用量に関しては、2 入力 XOR、4 入力 XOR、SIN カーブ近似へと変わるにつれ、計算遅延は大きくなっていき、LE は FPGA 上限の 15,408LE に対して比較的高い割合を使用していることが分かった。特に、本研究ではシグモイド関数近似モジュール設計を逐次処理で行ったため、計算遅延が発生しやすく、リソース効率のさらなる向上が求められる。

表 3: 実装した 3 問題のリソース消費量とレイテンシ

problem	LE 数	Latency（クロック）
2input XOR	3,191	71.25 クロック
4input XOR	10,172	90.1 クロック
SIN curve	11,899	97.5 クロック

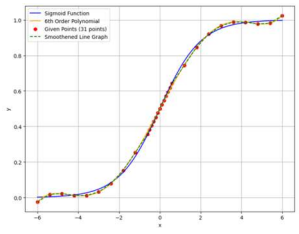

図 2: FPGA に実装したシグモイド近似

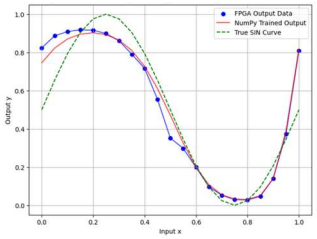

図 3: FPGA に実装した SIN カーブ近似

これらの結果から、FPGA 上でのニューラルネットワーク実装においては、演算精度・リソース使用量・レイテンシのトレードオフを考慮することが重要であることが分かった。今後の改善として、多項式演算を行う箇所でホーナー法や 2 進展開を採用し、計算遅延を削減すること、固定小数点演算の採用や、より高次多項式近似の導入、32 ビット浮動小数点を採用し精度向上の検討が必要である。

参考文献

[1] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, W. Wang, H. Yang, “Going deeper with embedded FPGA platform for convolutional neural network,” Proceedings of the 2016 ACM/SIGDA international symposium on field-programmable gate arrays, pp. 26–35, 2016.