C-014

C++AI 推論ライブラリに向けたモデル構造最適化ツールの開発
Model Structure Optimization Software for C++ AI Inference Library

伊原 和美† 中西 知嘉子†
Kazumi Ihara Chikako Nakanishi

1. はじめに

1.1 研究背景と目的

近年エッジ AI と呼ばれる AI 技術が様々な場所で活用されている。エッジ AI とは、エッジ端末とよばれる端末上で推論処理を行う技術のことを指す。

エッジ AI のメリットとして、利用時に通信を行わないこと、端末が小型かつ安価であることから、リアルタイムなレスポンスを要求される処理に対応できること、様々な場所で利用可能であることがあげられる。

また、デメリットとして、エッジ端末の処理性能が一般的にクラウド AI で用いられるサーバに劣り、複雑な処理を高速に行うことは難しい点があげられる。

そこで、本研究では、エッジ AI の精度を保ちつつ、計算量を削減する手法について検討する。

1.2 関連研究

エッジ AI の精度を保ちつつ、計算量を削減する手法として、次のような手法がある。

まず、FPGA および Ceras(C++AI 推論ライブラリ)[1]を用いて、特定の層を回路化することにより、計算時間を削減する手法がある。Ceras(C++ Edge Rapid AI Simulator)とは、C++言語で記述された、AI 推論ライブラリである。C++標準ライブラリのみで動作することから、環境構築が容易であるメリットがあり、FPGA の研究開発に使用されている。

Ceras および Ultra96V2 を使用し、YOLOv5 の畳み込み層および活性化関数を回路化する手法では、回路化前の CPU のみで実行した YOLOv5 の推論時間が 226113.18ms であったのに対し、回路化後の推論時間は 9575.22ms であった。[2]この結果から、計算量の大きい層を回路化する手法は、計算時間の削減において有効な手法であるといえる。

回路化するにしたがって、回路化された層の実行時間は大きく削減されたが、回路部分以外の実行時間には削減の余地が大きく残されており、この部分を削減することで、回路化を行う手法がより有効なものとなると考えた。

また、Ceras の動作には、機械学習モデルを Ceras フォーマットへ変換する必要がある。Ceras フォーマットを扱うことのできるモデルの最適化ツール等がなく、回路の実装を行う際には、設計者が自身の設計した回路に合わせてモデルを手作業で変更する必要がある。

2. 提案手法

そこで、本研究では、C++AI 推論ライブラリ Ceras に向けたモデル構造最適化ツールの開発を行い、モデル構造の冗長な部分の置換および研究開発作業の支援を行う。

2.1 概要

本研究で開発を行うツール（以下、本ツールとする）は、主に次の 3 つの機能を提供する。1 つ目が「モデル構造最適化機能」、2 つ目が「モジュール構造置換機能」、3 つ目が「モデル構造可視化機能」である。ソフトウェアの動作画面を図 1、図 2 に示す。

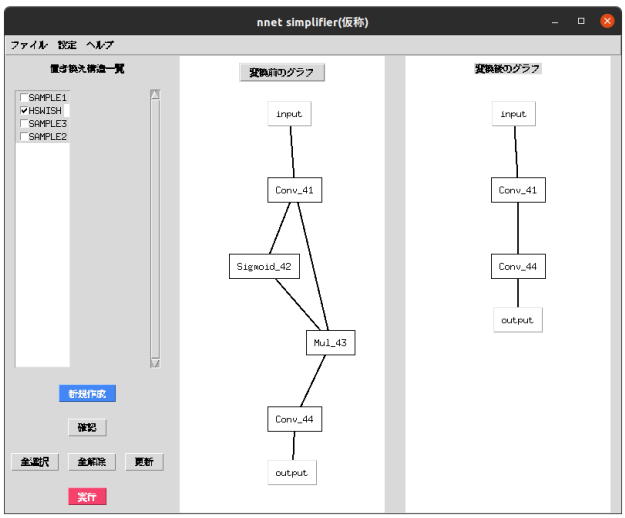

図 1　メイン画面

図 1 は、本ツールを起動した際に表示されるメイン画面である。

メイン画面の主な動作フローは次の通りである。

まず、図 1 上部のメニュー欄から、Ceras フォーマットのモデルを読み込む。読み込まれたモデルの構造は、画面中央に表示される。

次に、図 1 の左側のチェックボックスリストから、モデル構造の変更方法のオプションを設定する。ここで、どのような層構造の変換を行うかについては、2.2 節で詳細を述べる。

そして、図 1 左側下部の「実行」ボタンをクリックすると、ユーザが選択したオプションに基づいて、モデルの最適化が実行される。最適化実行後のモデルの構造が、図 1 右側に表示される。ここで、ユーザの意図した通りの最適化が実施されているかどうかを確認することができる。

† 大阪工業大学大学院　システムアーキテクチャ研究室
Osaka Institute of Technology System Architecture Laboratory

また、図 1 左下の新規作成ボタンをクリックすると、図 2 に示すような置換構造設定画面が表示される。

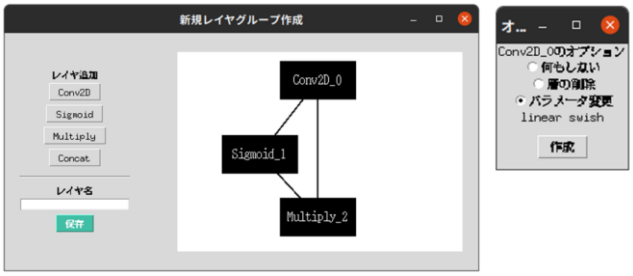

図 2　置換構造設定画面

この画面からは、ユーザが任意の最適化設定を作成することができる。

2.2 層構造の最適化

本ツールでは、次の 4 つの最適化設定を用意している。

まず 1 つ目が、Conv2D 層とアクティベーション層の融合である。Conv2D 層にアクティベーション層の情報を追加し、モデルからアクティベーション層を削除することによって、構造の簡易化を実現する。

2 つ目が、Conv2D 層と BatchNormalization 層の融合である。推論時、BatchNormalization 層は、単に定数倍と定数値の加算が行われる。よって、BatchNormalization 層のパラメータを用いて Conv2D 層のカーネルおよびバイアスの値を変更し、BatchNormalization 層をモデルから削除することで、計算量を削減することができる。

3 つ目が、Conv2D 層と Sigmoid 層、Mul 層が連なる構造の融合である。構造変換の様子を図 3 に示す。

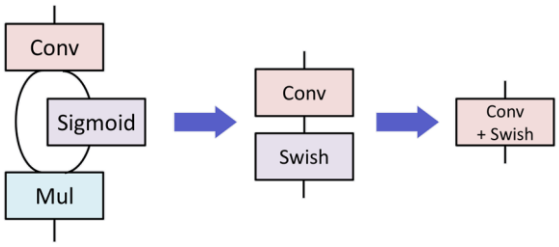

図 3　Sigmoid,Mul 層の Swish 化

図 3 の左側に示す Conv2D 層と Sigmoid 層、Mul 層が連なる構造は、Conv2D 層と Swish のアクティベーション層が連なる構造と等価な出力が行われる。そこで、Sigmoid と Mul 層を削除する。さらに、Swish 関数よりも計算負荷の少ない Hard-Swish[3]を活性化関数として使用するよう Conv2D 層のパラメータを追加することで、計算時間の削減をはかる。

4 つ目が、RepVGG[4]の訓練時に使用される層構造の発見と、融合によるパラメータの変更である。構造変換の様子を図 4 に示す。

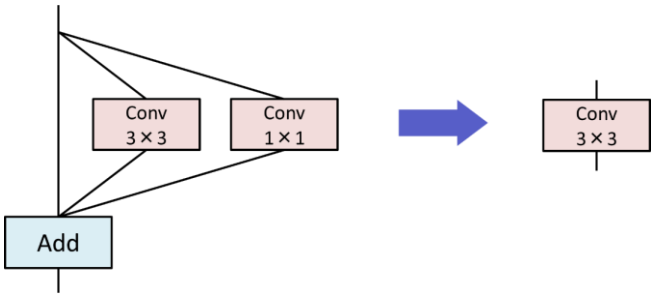

図 4　RepVGG の訓練時モデルの構造変換

図 4 の左側に示している、3×3 の Conv2D 層、1×1 の Conv2D 層、skip-connection が 1 つの Add 層に接続される構造は、1 つの 3×3 の Conv2D 層に変換することができる。これらの構造は等価であり、このような変換を行うことで、よりモデル構造を簡潔にすることができる。

これらの最適化は、ユーザが自由に適用の有無を変更することができること、また、ユーザが作成した回路に合わせて任意の構造を指定できることから、当初の目的である、モデル構造の最適化および FPGA、Ceras を活用した研究開発活動の支援を行えるものと考える。

3. まとめ・今後の展望

本研究では、C++AI 推論ライブラリ Ceras へ向けた、モデル構造の最適化および研究開発活動を支援するツールの開発を行った。

今後は、本ツールを足掛かりとして、さまざまなモデル構造の比較検討および柔軟な構造変更手法の検討に取り組もうと考えている。また、本ツールの目的である、FPGA を用いた AI の推論時間の削減を目的とした研究開発活動の支援機能を充実させるために、Ceras との連携および改良を検討している。

謝辞

本研究を行うにあたり、多大なご支援・ご協力を賜りました大阪工業大学情報科学部システムアーキテクチャ研究室の皆様に対しまして、感謝の念を示します。

参考文献

[1] 西岡駿, 中西知嘉子, "機械学習ライブラリの C 言語化の実現", 電子情報通信学会ソサイエティ大会(2021)
[2] 三枝渉, 中西知嘉子, “物体検出モデル「YOLOv5」のエッジデバイスへの実装の検討”, 電子情報通信学会総合大会(2023)
[3] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam, "Searching for MobileNetV3", ICCV(2019)
[4] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, Jian Sun, “RepVGG: Making VGG-style ConvNets Great Again”, CVPR(2021)