C-013

エッジ推論高速化のためのモデル構造最適化と協調動作支援ツールの開発

Development of a Model Structure Optimization and Cooperative Execution Support Tool for Accelerating Edge Inference

野村美月 ‡　　中西知嘉子 ‡
Mitsuki Nomura　Chikako Nakanishi

1. はじめに

エッジデバイスには計算資源の制約があり，リアルタイム処理のためには高速かつ軽量なモデルが求められる．こうした背景から，SoC FPGA のようなハードウェアを活用した推論処理が注目されている．

本研究では，ディープラーニングモデルのネットワーク構造を解析し，ハードウェア実装を考慮した層の融合や削除によってモデルの簡素化を図ることで，FPGA 上で効率的に推論を実行するための支援ツールの開発を目的とする．

2. 使用ツール

2.1 Ceras [1]

Ceras(C++ edge rapid ai simulator)とは，弊研究室で開発された，機械学習モデルの推論処理を実行するための C++言語によるイブラリである．標準 C++ライブラリのみで動作するため，環境構築が不要で幅広いプラットフォームで利用可能である．弊研究室では主に，SoC FPGA を用いた推論処理の高速化に関する研究に Ceras を活用しており，論理回路との協調動作が容易にできることも特徴の一つである．また，Ceras を使用する際には，機械学習モデルを Ceras に対応したフォーマット形式(以下，Ceras 形式とする)へ変換する必要がある．Ceras で推論を行うには，学習済みモデルを一度 ONNX 形式へ変換し，ONNX 形式から Ceras 形式に変換するという手順を取る．Ceras 形式は，ネットワーク構造や重みといった基本情報に加えて，ハードウェア上で効率的に推論処理を実行するための補助情報（例：レイヤ間データのメモリ配置，量子化情報など）も含んでいる．これにより，Ceras 推論エンジンは，ハードウェアの効率の良い動作と，ソフトウェアとハードウェアの協調動作を意識した実行処理を実現することが可能となる．

2.2 ONNX(Open Neural Network eXchange)

ONNX は，異なる機械学習フレームワーク間でモデルを共有・転送可能にするオープンフォーマットである．これにより，学習済みモデルを共通形式で扱えるようになり，学習環境と推論環境の分離や柔軟なデプロイが可能となる．

3. 提案手法：Ceras 形式モデルへの変換と FPGA 協調実行支援

本章では，学習済みの深層学習モデルを Ceras 形式へ変換し，FPGA と CPU の協調実行を効率化するために導入した各種支援機構について述べる．本手法は，PyTorch や Keras などの多様なフレームワークで構築された学習済みモデルを出発点とし，ONNX 形式への変換を経て，最終的に Ceras 形式へと変換することで，推論の高速化とメモリ効率の最適化を図るものである．

本研究では，ResNet を対象として検証を行った．ResNet は，残差接続により深いネットワークでも学習が安定する画像認識向け CNN であり，広く使われている代表的なモデルである．

Ceras 形式への変換において，本研究では以下の 3 つの主要な支援機構を導入している：

① 層の融合変換機構

FPGA 上の演算回路に最適化された形で，Conv2D と ReLU，BatchNorm，Add などを統合することで，推論処理の効率を高める機構．

② 共有メモリのブロック割り当て機構

SoC 内の共有メモリをブロック単位で管理し，層の生存期間に基づいてメモリを静的に割り当てることで，データの重複保持や必要なデータを誤って消してしまうことを防ぎつつ，メモリ使用量を削減する機構．

③ 実行先の指定機構

各層が FPGA で実行されるか，CPU で実行されるかを明示的に指定し，データ転送の無駄を回避することで，協調動作の効率を高める機構．

これら 3 つの機構は，すべてモデル変換時に静的に適用されるため，推論実行時に追加的な制御を行う必要がなく，一貫性のある高効率な協調推論実行が可能となる．

3.1 層の融合変換機構

Ceras 形式へのモデル変換においては，FPGA アクセラレータ上での推論実行を効率化するために，複数の演算層をあらかじめ融合（fuse）する処理を適用する．

(1) Conv2D + ReLU の融合

Conv2D 層の後段に位置する ReLU 層については，FPGA の Conv2D 回路にアクティベーション処理（ReLU）を組み込むことが可能である．そのため，Conv2D 層のモデル情報に「Conv2D の計算後に ReLU を適用する」という属性を付加することで，ReLU 処理を内部的に吸収する．この処理に伴い，元の ReLU 層はモデルから削除され，Conv2D 層の出力は ReLU 層の出力先に再接続される．これにより，演算ユニット数の削減と処理の高速化が実現される．

(2) Conv2D + BatchNorm の融合

BatchNorm 層は，推論時にはスケーリングおよびバイアス加算の処理に置き換え可能であり，Conv2D の重み・バイアスに吸収することで統合可能である．

本手法では，PyTorch や ONNX が提供する fuse 処理（Conv-BN 融合）を用いて，モデル中の Conv2D + BatchNorm を一つの Conv2D 層に変換する．この処理は Ceras 形式への変換に先立つ前処理段階で行われ，Ceras モデルには既に統合済みの重み・バイアスが取り込まれる．

(3) Add + ReLU の融合

Add 層においても，加算後にアクティベーション関数を適用する回路構成が可能である．本手法では，Add 層の演

算回路に ReLU 処理を組み込む機構を設け，モデル情報上では「Add の演算後に ReLU を実行する」という属性を付加する．その後，ReLU 層を削除し，Add 層がその出力を引き継ぐ構成に変換される．

これらの融合処理は，推論時の演算効率向上とメモリ転送量削減の両面において有効であり，Ceras 形式におけるモデル軽量化および高速化の基盤となる

3.2 共有メモリのブロック割り当て機構

Ceras では，FPGA と CPU の協調推論を実現するために，SoC 内に配置された共有メモリを，FPGA アクセラレータと CPU の間の中間データの受け渡し領域として活用している．この共有メモリは，限られた資源であるため，効率的な管理が不可欠である．

本手法では，この共有メモリを固定サイズの複数のブロックに分割し，各演算層が使用するメモリブロックを Ceras 形式のモデル変換時に静的に割り当てる．これにより，推論実行中における動的なメモリ確保や解放を行う必要がなくなり，実行時のオーバーヘッドを回避できる．

3.2.1 ライフタイムに基づく割り当て戦略

メモリブロックの割り当ては，各層の生存期間（ライフタイム）を解析し，各層の生存期間を考慮することで，必要なデータが上書きされることなく，安全にメモリブロックを再利用できるようにする．ここでの生存期間とは，ある層の出力が生成された時点から，それを最後に使用する層までの期間を指す．

- 層の出力ごとに生存区間を計算し，ブロックの重複使用が可能かどうかを判定．
- 干渉する生存区間をグラフで表現し，隣接するノードが同一ブロックを持たないようにすることで，カラーリングアルゴリズムを適用．
- 必要なブロック数を最小限に抑えながら，データ消失や上書きを防止する割り当てを決定．

この割り当て結果は Ceras 形式のモデルに付加され，推論時はその情報に従ってデータを読み書きすることで，不要なメモリコピーや領域の再確保を一切行わない高効率な実行が可能となる．

3.2.2 FPGA 回路と共有メモリの連携

FPGA 上の複数の演算モジュールが連続して実行される場合には，出力されたデータブロックをそのまま次の層の入力ブロックとして利用する構成が取られる．これにより，出力結果を CPU に転送して再配置する必要がなく，データ転送コストの削減とスループットの向上が実現される．

さらに，出力が複数の層で参照される場合にも，ブロックを保持したまま共有利用が可能であり，データの一時保存のために CPU 側へコピーする必要がない．

このように，層のライフタイム情報に基づいた静的な共有メモリのブロック割り当て戦略は，Ceras における FPGA 協調実行において極めて重要であり，回路資源とメモリ資源の両方を最適化する基盤技術である．

3.3 実行先の指定機構

深層学習モデルには，すべての演算が FPGA で実行可能とは限らず，一部の層（例：Resize，Shape，Constant 演算など）は回路化されておらず，CPU での実行が必要となる．本手法では，Ceras 形式への変換時に，各層ごとに「CPU で実行されるか，FPGA で実行されるか」という実行先の情報を明示的に付加する機構を導入している．

CPU 実行層における共有メモリ活用

回路化されていない層が CPU で実行される場合でも，その出力が次の FPGA 実行層の入力として使用されるケースが多い．この場合，一度 CPU のローカルメモリで計算された出力を，再度共有メモリに転送するのは効率が悪い．

そこで本手法では，CPU 実行層に対しても共有メモリブロックの静的割り当てを許可し，必要に応じて，CPU が直接共有メモリに出力を書き込む構成を採用している．この情報も Ceras モデル中に含まれており，実行時にはこの情報に従って処理される．これにより，CPU → FPGA 間の不要なデータ転送を回避し，実行効率を向上，すべての層の入出力データ管理を統一的に共有メモリ上で行うことが可能などのメリットが得られる．

このように，各層の実行先を明示的にモデル内で管理することにより，FPGA・CPU 間の協調実行において処理の流れが明確化され，実行制御の一貫性が保たれる．また，データコピーの有無やメモリブロックの管理方針も，この情報に基づいて静的に最適化されるため，推論中の追加制御や負荷が発生しない．

このように，本機構は，FPGA に実装されていない層を柔軟に扱いつつ，データ転送の最小化と処理効率の最大化を両立する上で重要な要素である．

4. まとめ・今後の展望

本章では，Ceras 形式へのモデル変換において，FPGA と CPU の協調推論を効率化するための 3 つの機構を提案した．

- 層の融合変換機構では，Conv2D + ReLU，Conv2D + BatchNorm，Add + ReLU などの構成を事前に融合することで，演算ユニット数の削減と処理の高速化を実現した．
- 共有メモリのブロック割り当て機構では，層の生存期間に基づく静的なメモリ割り当てを行い，必要なデータが上書きされることなく，限られた共有メモリを安全かつ効率的に再利用可能とした．
- 実行先の指定機構では，回路化されていない層を CPU で実行する際にも共有メモリを活用し，CPU と FPGA 間の不要なデータ転送を削減した．

これらの機構はすべて，Ceras 形式への変換時に静的に適用されるため，推論実行中に追加の制御を必要とせず，高効率な協調動作を実現する．

今後の展望として，次の課題と取り組みが挙げられる．

- 対象モデルの拡張：大規模・複雑モデルへの適用と評価．
- 実機による推論速度評価：Ceras を実装した実際の FPGA プラットフォーム上での推論時間・レイテンシ・スループットなどの定量的な性能評価を通じて，提案手法の実効性を検証する．

本研究の成果により，限られた計算資源を有効活用しながら，高精度かつ高速なエッジ AI 推論が可能となり，今後の FPGA ベースの AI 実行環境における基盤技術としての応用が期待される．

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)

† 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