

プルーニング後の畳み込みカーネルのパターン解析に基づく可変並列性再構成アーキテクチャにおける計算スキップ

井上 祐[†] 堀 篤史[†] 丸亀 孝生[†]

浅井 哲也[†] Alexandre Schmid^{††} 安藤 洸太[†]

[†] 北海道大学大学院情報科学院
北海道札幌市北区北 14 条西 9 丁目

^{††} スイス連邦工科大学ローザンヌ校

E-mail: [†]{inoue.yuu.a0,hori.atsushi.f9,takao.marukame,asai,ando}@ist.hokudai.ac.jp, ^{††}alexandre.schmid@epfl.ch

あらまし 本研究では、AI 推論向け可変並列性再構成アーキテクチャにおいて、構造的プルーニングの評価指標が畳み込みカーネル構造、推論精度、量子化耐性に与える影響を分析する。対象アーキテクチャは、層の特性に応じて計算データフローを切り替え、所定の重みグループが全ゼロの場合に計算スキップを行う。複数のグループ重要度指標に基づく構造的プルーニング手法を、FP32 精度、エミュレート 8 ビット精度、ゼログループ形成、サイクルレベル速度向上の観点から比較し、ゼロ化グループの空間分布と量子化後マスクの違いから、スキップに適したカーネル形成との関係を明らかにする。

キーワード 再構成可能プロセッサ、ニューラルネットワーク、CGRA、プルーニング

Computation Skipping in a Variable-Parallelism Reconfigurable Architecture Through Pattern Analysis of Pruned Convolutional Kernels

Yu INOUE[†], Atsushi HORI[†], Takao MARUKAME[†],

Tetsuya ASAI[†], Alexandre SCHMID^{††}, and Kota ANDO[†]

[†] Graduate School of Information Science and Technology, Hokkaido University

^{††} Swiss Federal Institute of Technology in Lausanne (EPFL)

E-mail: [†]{inoue.yuu.a0,hori.atsushi.f9,takao.marukame,asai,ando}@ist.hokudai.ac.jp, ^{††}alexandre.schmid@epfl.ch

Abstract This study analyzes the impact of structural pruning criteria on convolutional kernel structures, inference accuracy, and quantization robustness in a variable-parallelism reconfigurable architecture for AI inference. The target architecture dynamically switches computation dataflows according to layer characteristics and incorporates a computation-skipping mechanism that skips operations when all weights within a predefined group are zero. We apply multiple structured pruning methods based on different group importance criteria and compare them in terms of FP32 accuracy, emulated 8-bit accuracy, zero-group formation, and cycle-level speedup. Furthermore, by analyzing the spatial distribution of zero-valued groups and the differences in post-quantization mask patterns, we clarify the relationship between pruning criteria and kernel structures suitable for computation skipping.

Key words Reconfigurable processor, Neural networks, CGRA, Pruning

1. はじめに

近年、プライバシー保護や通信コスト削減の観点から、クラウドに依存せずデバイス上で推論処理を行うエッジ AI への関心が高まっている。エッジデバイスは計算資源や電力供給に制約があるため、ニューラルネットワークの推論処理を高効

率かつ省電力に実行するための専用アクセラレータが求められている。

このような背景のもと、我々は AI 推論向けの可変並列性再構成可能アーキテクチャを検討している [1], [2]。本アーキテクチャは、層ごとの特性に応じてメモリと演算ユニットの接続やデータフローを切り替えることで、多様な層構造に対して

高い計算資源利用率を実現する。さらに、特定の重みグループがすべてゼロである場合に計算をスキップする機構を備えており、モデルのスパース性を活用した高速化が可能である。

一般に、プルーニング手法の評価では、推論精度、スパース率、演算量削減率などが代表的な指標として用いられる。しかし、本アーキテクチャにおける計算スキップは、あらかじめ定義された重みグループ単位で発動するため、これらの指標だけでは実際の高速化効果を十分に表せない。同程度のスパース率であっても、ゼロの配置やグループ単位でのまとまり方によって、スキップ可能な計算量は変化する。

そこで本研究では、可変並列性再構成可能アーキテクチャにおけるグループベース計算スキップに着目し、複数の構造化プルーニング指標が生成する重み構造の違いを分析する。具体的には、各手法について推論精度、ゼログループ形成率、およびサイクルレベルの高速化率を比較するとともに、畳み込みカーネルにおけるゼロ分布やゼロマスクの類似度を解析する。これにより、本アーキテクチャに適したスパース構造の特徴を明らかにし、再構成アーキテクチャに適したモデル設計指針を得ることを目的とする。

2. 関連研究

従来のニューラルネットワークアクセラレータでは、データフローおよびメモリアクセスの最適化によって高効率な推論を実現する研究が数多く行われてきた。例えば、TPU[3]は大規模行列演算を高効率に実行するためのアーキテクチャを採用し、Eyeriss[4]はデータ再利用を最大化するデータフロー設計により、演算エネルギーだけでなくメモリアクセスの削減も重視している。また、ShiDianNao[5]はセンサ近傍での処理を志向し、畳み込み演算を高スループットかつ低消費電力で実行する構成を示した。これらの研究は主として、規則的な密行列・密テンソル計算を前提として、データ移動と計算資源利用の最適化により性能向上を図るものである。これに対し、ニューラルネットワークに内在するスパース性を活用して不要な演算を削減するアーキテクチャも提案されている。EIE[6]は圧縮後の疎な重みを直接処理することで、メモリ容量と演算量の両方を削減する推論エンジンを実現した。Cnvlutin[7]は活性値のゼロに着目し、無効なニューロンに対応する計算を省略することで効率向上を図っている。SCNN[8]は重みと活性値の両方のスパース性を圧縮表現と専用データフローにより活用し、疎な畳み込み演算を高効率に実行する。このように、計算スキップ型アクセラレータでは、単なる演算器の高速化だけでなく、スパースなデータ表現とそれを処理する実行機構の協調設計が重要となる。一方で、モデル圧縮の観点からは、重み数や演算量を削減するためのプルーニング手法が広く研究されてきた。Hanらの手法[9]に代表される非構造化プルーニングは、高いスパース率を達成しやすい一方で、ゼロの分布が不規則になりやすく、一般的なハードウェア上ではインデックス管理や不規則なメモリアクセスが必要となるため、理論上の削減効果を実機性能へ直結させにくい。これに対して、チャンネル単位の削減を行う Channel Pruning[10]や、

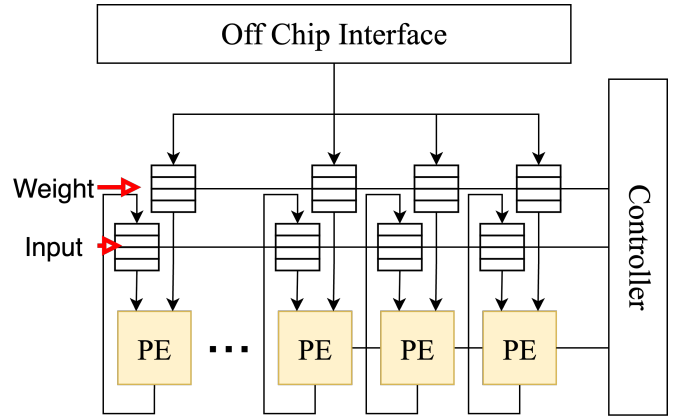


図1 ベースラインアーキテクチャ

フィルタ単位の削減を行う手法[11]、さらには資源効率を考慮した構造的な削減手法[12]など、規則的な単位で重みを削除する構造化プルーニングが提案されている。これらは一般に、ハードウェア実装や実行時間短縮との親和性が高いとされる。しかし、構造化プルーニングが常にハードウェア上で高い効果を示すとは限らない。特に、計算スキップの発動条件が「特定の重みグループがすべてゼロであること」のようにアーキテクチャ固有の規則で定義される場合、重要なのは全体のスパース率そのものではなく、ゼロがその規則に整合した形で形成されるかどうかである。すなわち、同程度のスパース率を持つモデルであっても、ゼロ値の空間分布やグループ単位でのまとまり方が異なれば、実際にスキップ可能な計算量やメモリアクセス量は大きく変わり得る。本研究では、この点に着目し、可変並列性再構成アーキテクチャ[1],[2]を対象として、あらかじめ定義されたグループベース計算スキップ規則との適合性という観点から構造化プルーニング指標を解析する。特に、従来研究で主に議論されてきた精度やスパース率だけでなく、ゼログループ形成のされ方、量子化後の推論性能、さらにカーネル内に現れるゼロパターンの違いに着目して評価する点に特徴がある。これにより、どのようなプルーニング指標が対象アーキテクチャにおける計算スキップに適したカーネル構造を形成するかを明らかにする。

3. 可変並列性再構成アーキテクチャ

3.1 アーキテクチャの概要

対象アーキテクチャは、ニューラルネットワーク推論向けの可変並列性再構成アーキテクチャである[1],[2]。図1にそのベースライン構成を示す。コントローラユニット（CU）は、メモリとPE間の接続およびPE間接続を動的に変更することで、各層の特性に応じた計算データフローを選択する。この再構成性により、固定並列型設計と比べて、多様な層形状およびチャンネル構成に対して高い利用率を実現できる。

本アーキテクチャは、入力経路、出力経路、およびPE間通信を再構成することで、複数の畳み込みデータフローをサポートする。本研究では、カーネル重みを複数回のMAC演算にわたって再利用する Kernel Stationary（KS）データフローが特に

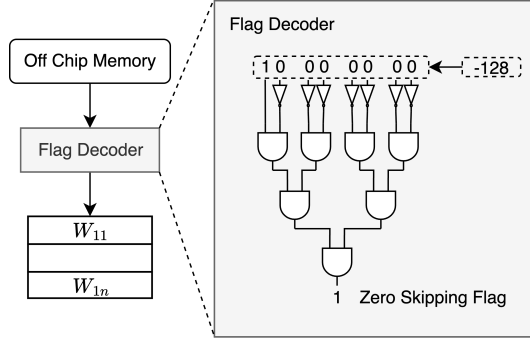


図2 スキップフラグ生成機構 ([1] より引用)

重要であり、これが後述する計算スキップ機構を直接可能にしている。

3.2 グループベースの計算スキップ機構

提案する計算スキップ機構は、KS データフローにおける重み再利用性を利用する。KSCoP (Kernel Stationary Output Channel Parallel) および KSCiP (Kernel Stationary Input Channel Parallel) では、あるカーネル重みが複数の連続する MAC 演算で再利用される。したがって、再利用される重みがゼロであれば、対応する一連の MAC 演算は無効となり、スキップ可能である。

この機構を小さなハードウェアオーバーヘッドで実現するため、本アーキテクチャでは重みメモリと PE アレイの間に小規模なデコーダを配置する (図 2)。スキップフラグは重みデータと同じ記憶領域に埋め込まれるため、スキップ制御専用の追加メモリは不要である。デコーダがゼロ重みグループに対応するスキップフラグを検出すると、CU に通知し、CU はその制御下でカーネルアドレスを更新して次の有効な計算位置へ直接進む。その結果、ゼロ重みをすべての PE へ供給することなく計算スキップが実現され、ゼロ重みグループが連続して現れる場合でも、不要な重み転送と演算を回避できる。

1 サイクルの計算をスキップできるのは、同時に並列 PE へ供給される全重みがゼロである場合に限られる。したがって、ゼロスキップの有効性は、全体スパース率そのものよりも、あらかじめ定義された PE グループ化規則の下でゼロ重みグループがどのように形成されるかに強く依存する。この観察が、後続節におけるプルーニング解析の動機となる。

4. 構造的プルーニング

4.1 グループ定義

対象アーキテクチャでは、計算スキップはあらかじめ定義されたグループ単位で行われる。1 つのグループは、同一カーネル座標にある複数の出力チャンネルの重みをまとめることで構成され、それらの重みが並列 PE へ同時に供給される。したがって、このようなグループ内の全重みがゼロである場合にのみ、1 サイクルの計算をスキップできる。グループサイズは PE アレイの並列度によって決まり、本研究では 64 または 128 といった値を用いる。このグループ化規則の下では、全体スパース率のみではスキップ効率は決まらず、むしろグループ全体をどれだけ効果的にゼロ化できるかが重要となる。図 3 に

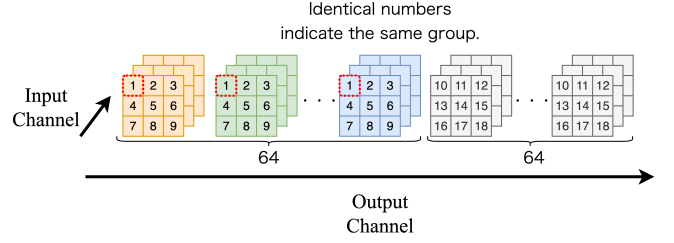


図3 入力チャンネル数 3, 出力チャンネル数 128, PE 数 64 の場合のグループ形成例

グループ形成の例を示す。

4.2 プルーニング手法の概要

本研究では、上記のあらかじめ定義されたグループ定義の下で、指標ベースの構造化プルーニングを調査する。事前学習済み重みから出発し、各グループに対して選択した重要度指標に基づくスコアを与え、重要度の低いグループを削除する。同一の事前学習済みモデルに対して異なる指標を適用しつつ、畳み込み層のスパース率をほぼ一致させることで、指標の選択が推論精度、量子化耐性、および得られるカーネル構造に与える影響を解析する。比較のため、あらかじめ定義されたグループではなく各重み要素を個別に削除する従来の非構造化 magnitude pruning もベースラインとして評価する。

4.3 重要度指標の定義

$G = \{w_1, w_2, \dots, w_n\}$ を 1 つの重みグループとする。学習後グループプルーニングにおいて、本研究では以下の重要度スコアを比較する。

$$S_{\ell_1 \text{sum}}(G) = \sum_{i=1}^n |w_i| \quad (1)$$

$$S_{\max}(G) = \max_{1 \leq i \leq n} |w_i| \quad (2)$$

$$S_{\ell_2 \text{sum}}(G) = \sqrt{\sum_{i=1}^n w_i^2} \quad (3)$$

$$S_{\text{median}}(G) = \text{median}(|w_1|, \dots, |w_n|) \quad (4)$$

$$S_{\text{var}}(G) = \frac{1}{n} \sum_{i=1}^n (w_i - \bar{w})^2 \quad (5)$$

ただし、

$$\bar{w} = \frac{1}{n} \sum_{i=1}^n w_i \quad (6)$$

はグループ内の重みの平均値である。

$S_{\ell_1 \text{sum}}$ はグループ内の重みの絶対値の総和に基づき、グループ全体の寄与の大きさを評価する指標である。 S_{\max} はグループ内の最大絶対値に着目することで、少数の大きな重みによる影響を重視した指標である。 $S_{\ell_2 \text{sum}}$ は ℓ_2 ノルムに基づく指標であり、大きな重みほど強く寄与するため、エネルギー的な観点での重要度を反映する。 S_{median} は絶対値の中央値を用いることで外れ値の影響を抑えつつ、グループ内の典型的な重みの大きさを評価するロバストな指標である。 S_{var} はグループ

表1 各プルーニング手法の定量比較

Method	Conv. Spars. [%]	Zero Grp. [%]	FP32 Top-1 [%]	Emulated 8-bit Top-1 [%]	Speedup [×]
Baseline	0.00	0.00	82.94	81.40	1.00
Mag. pruning (unstr.)	85.00	1.07	82.24	80.21	1.13
Struct. pruning ($S_{\ell_1\text{sum}}$)	86.03	84.99	79.75	77.84	6.02
Struct. pruning ($S_{\ell_2\text{sum}}$)	85.79	84.97	78.13	73.59	6.27
Struct. pruning (S_{max})	86.14	84.99	80.00	77.25	6.11
Struct. pruning (S_{median})	85.86	84.97	79.85	76.75	6.14
Struct. pruning (S_{var})	86.05	84.99	80.58	78.37	6.03

内の重みの分散に基づく指標であり、値が小さいほど各重みが平均値の近くに分布している、すなわちグループ内のばらつきが小さいことを示す。

S_{var} に基づくプルーニングでは、分散の昇順、すなわち分散の小さいグループを優先的に削除し、分散の大きいグループを保持する設定を採用する。分散の小さいグループはグループ内の重みが互いに近い値をとるため、そのようなグループを削除することでネットワーク全体の表現能力への影響を抑えつつスパース化が可能であると考えられる。実際に予備評価においても、この設定は逆順（分散の大きいグループを削除する設定）と比較して高い推論精度を示した。以上の理由から、本研究では分散の昇順に基づくプルーニングを採用する。

5. 評価と解析

5.1 定量的評価

VGG16_BN を CIFAR-100 上で評価する。プルーニング対象は畳み込み層のみとし、グループサイズ（並列度）は 64 に固定する。指標ベース構造化プルーニングでは、各手法間で畳み込み層のスパース率がほぼ一致するように、反復的なプルーニングと再学習を行い、その後 8 ビット QAT ファインチューニングを適用する。

表 1 に、FP32 Top-1 精度、エミュレート 8 ビット Top-1 精度、ゼログループ率、および非プルーニングベースラインに対するサイクルレベル高速化率を示す。構造化手法間では、スパース率とゼログループ率がほぼ一致しており、指標そのものの直接比較が可能である。この条件下では、 S_{var} が FP32 およびエミュレート 8 ビットの両方で最も高い精度を達成している一方、 $S_{\ell_2\text{sum}}$ は最大の高速化率を示すが、精度低下も最も大きい。これに対し、magnitude pruning はほとんどゼロ値グループを生成しないため、高速化は限定的である。

高速化率は、形成されるゼログループ数だけでなく、それらがどこに位置するかにも依存する。3×3 カーネルでは、edge や corner 位置にあるゼログループは、より多くの有効入力画素に適用される位置のゼログループに比べて、削減できる計算量が少ない。また、層ごとのスパース率分布も全体の削減量に影響する。図 4 は、このような空間的・層ごとの差異を可視化したものであり、次節のカーネル構造解析の動機を与える。

5.2 カーネル構造解析

ゼログループ率が類似していても精度や高速化率が異なる理由を調べるため、Jaccard index を用いて得られたカーネル構

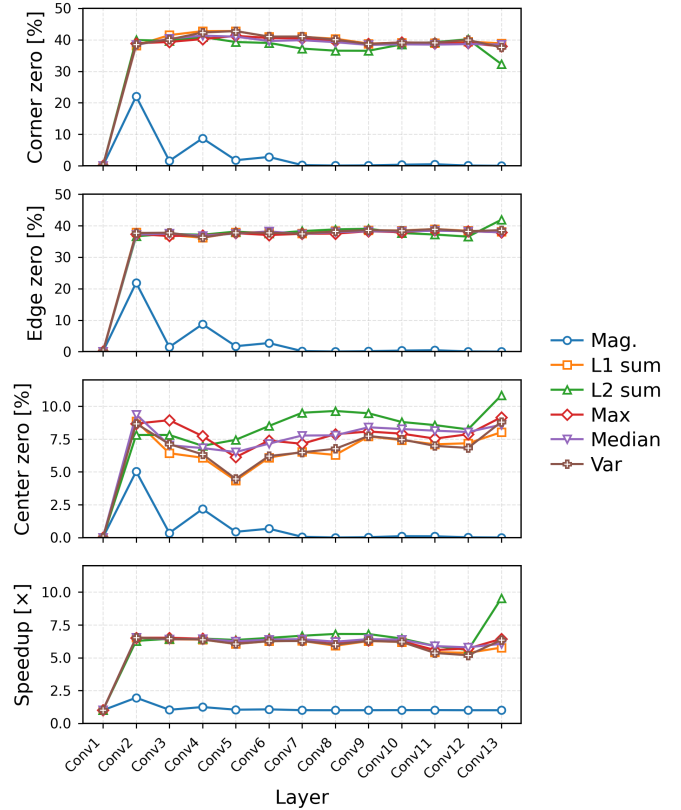


図4 3×3 畳み込みカーネルにおける、各プルーニング手法の層ごとのゼロ値グループの空間分布と対応する高速化率。上3つのプロットは、corner, edge, center 位置におけるゼロ値グループの正規化割合を示し、下段のプロットは対象アーキテクチャにおける層ごとの高速化率を示す。

表2 Jaccard index により測定した S_{var} に対する各手法のプルーニングによるモデル間のカーネル構造類似度

Target	Overall	Corner	Edge	Center
Mag. pruning (unstr.)	0.0126	0.0121	0.0124	0.0170
Struct. pruning ($S_{\ell_1\text{sum}}$)	0.8861	0.9084	0.8864	0.7690
Struct. pruning ($S_{\ell_2\text{sum}}$)	0.7983	0.8188	0.8106	0.6504
Struct. pruning (S_{max})	0.8496	0.8734	0.8521	0.7217
Struct. pruning (S_{median})	0.8144	0.8443	0.8190	0.6544

造を比較する。ここでいうカーネル構造とは、あらかじめ定義されたグループ化規則の下で、各 3×3 畳み込みカーネル内に形成されるゼロ値グループの空間パターンを指す。

具体的には、各畳み込み層の重みテンソルに対して、入力チャネル c_i およびカーネル座標 (k_h, k_w) を固定し、出力チャ

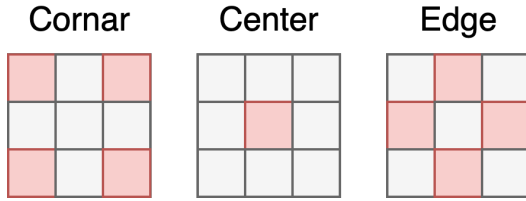


図5 表2で用いる“Corner”, “Edge”, “Center”座標の定義

ネル方向を PE 幅単位のグループに分割する．各グループについて，そのグループ内の全重みが 0 である場合に対応する mask 要素を 1，それ以外を 0 とすることで，二値の zero-row mask を構成する．したがって，各 mask 要素は，対象アーキテクチャにおける計算スキップの判定単位となる重みグループがゼロ化されているかどうかを表す．

表2は，指標ベース構造化プルーニング手法の中で最も高い FP32 およびエミュレート 8 ビット精度を達成した S_{var} に対する類似度をまとめたものである．類似度は，提案プルーニングによって構成された二値の zero-row mask を用いて評価し，手法間で対応するグループ位置を比較している．2 つの手法で得られた mask をそれぞれ A ， B とすると，Jaccard index は

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (7)$$

で定義される．ここで $|A \cap B|$ は両手法で共通して zero-row となった位置数， $|A \cup B|$ は少なくとも一方で zero-row となった位置数を表す．

表2における“Overall”は，各層における Jaccard index を，zero-row mask の要素数を重みとする重み付き平均により集約した値である．各層 ℓ における Jaccard index を J_ℓ ，その mask 要素数を N_ℓ とすると，集約値 \bar{J} は

$$\bar{J} = \frac{\sum_\ell N_\ell J_\ell}{\sum_\ell N_\ell} \quad (8)$$

により求める．

Magnitude pruning は S_{var} に対して極めて低い類似度を示しており，グループを考慮した構造化手法と同じゼログループ構造を形成していないことを示している．一方，他の構造化手法は S_{var} と高い類似性を保っており，Overall の Jaccard 値は 0.7983–0.8861 である．その中では S_{ℓ_1sum} が最も近く， S_{ℓ_2sum} と S_{median} は相対的に類似度が低い．

表2には，図5に示す 3×3 カーネルの corner, edge, center 座標ごとの類似度も示している．ここで“Corner”，“Edge”，“Center”は，それぞれ4つの corner 座標，4つの edge 座標，および center 座標に対応するマスク要素から計算した Jaccard index を表す．各カーネル座標ごとに独立してグループが定義されているため，この内訳により，指標間の構造差がどこより明確に現れるかを把握できる．この比較では，center における類似度が corner や edge より一貫して低く，構造化プルーニング指標間の差異がカーネル中心部の扱いにより強く現れていることが示唆される．この傾向は図4とも整合している．

もう一つの注目すべき傾向として， S_{var} に近いカーネル構造

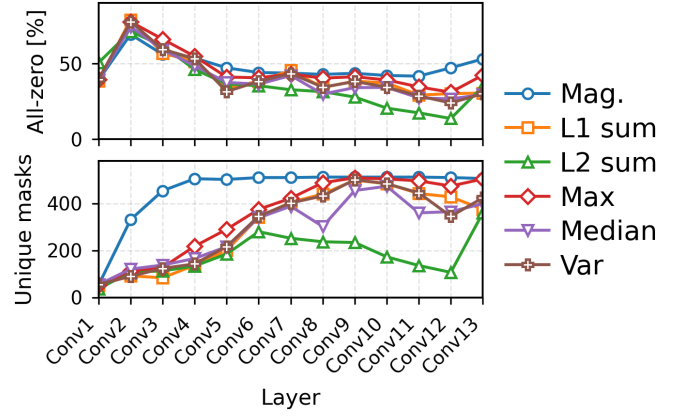


図6 層ごとの全ゼロカーネル率とユニークマスク数．Conv0 は除外

を持つ手法ほど，比較的高い推論精度を維持する傾向が見られる．表1に示すように， S_{var} に高い類似度を持つ S_{ℓ_1sum} は，FP32 およびエミュレート 8 ビット精度も比較的高く保っている一方で， S_{var} との類似度が低い S_{ℓ_2sum} は，より大きな精度低下を示している．この観察は因果関係を直接示すものではないが， S_{var} によって形成されるカーネル構造が，良好な精度特性と関連している可能性を示唆している．

図6では，各構造化プルーニング指標によって生成されるカーネル特性を，層ごとの全ゼロカーネル率と量子化後ユニークマスクパターン数を用いてさらに比較する．

構造化プルーニング手法の中では， S_{var} は，全ゼロカーネル率を比較的低く保ちながら，層を通して比較的高いマスク多様性を維持している．Magnitude pruning モデルは当然ながら最も多くのユニークマスクを示すが，構造化指標の中では， S_{max} ， S_{var} ，および S_{ℓ_1sum} が他の手法より多くのユニークパターンを保持している．一方で， S_{ℓ_2sum} は，全ゼロカーネル数もユニークパターン数も比較的小さい．これらの結果は，ゼログループ率をほぼ一致させた場合でも，得られるカーネル特性がプルーニング指標によって大きく異なることを示している．

この差異は，あらかじめ定義されたグループ化規則がカーネル構造に与える制約から解釈できる．グループは，同一カーネル座標にある複数の出力チャンネルの重みによって構成されるため，構造化プルーニングではそれらの重みをまとめて削除または保持する．その結果，対応するカーネルは同じ粗いマスク構造を共有しやすくなり，残る差異は主として保持されたグループ内部の重み値と，量子化によって追加的に生じるゼロに起因する．

この観点から， S_{var} は特に意味がある．なぜなら，グループ内部の変動が大きいグループを優先的に保持するためである．粗いグループレベルのマスクが共有されていても，各グループ内に保持される重みはより不均一なままであり，その結果，量子化後に得られる二値のゼロ／非ゼロパターンの多様性が大きくなる可能性がある．本研究では，この保持されたパターン多様性が， S_{var} が指標ベース構造化プルーニング手法の中で最も高い精度を達成する理由の一つであると考えられる．

S_{ℓ_1sum} もまた，カーネル構造解析全体を通して S_{var} に近い

傾向を示しつつ、FP32 およびエミュレート 8 ビット精度も比較的近い値を達成している。これは、 S_{var} の優位性が、まったく異なるカーネル構造に由来するのではなく、ゼログループ形成、マスク多様性、全ゼロカーネル抑制のバランスがわずかに優れていることに起因することを示唆している。これに対し、 $S_{\ell_2\text{sum}}$ は全体として多様性が低くなる傾向があり、 S_{max} はいくつかの層で全ゼロカーネルを増やす傾向がある。

総じて、これらの観察結果は、提案する構造化ブルーニング設定の下では、精度がゼログループ形成だけでなく、残存するカーネルパターン間の多様性がどの程度保持されているか、および全ゼロカーネルが過度に生成されていないかにも関係していることを示している。本解析で観測された範囲では、 S_{var} がこれらの要因の間で最も良好なバランスを提供しており、それが高い精度と関係していると考えられる。

6. 総 括

本研究では、可変並列性再構成可能アーキテクチャにおけるグループベース計算スキップに着目し、複数の構造化ブルーニング指標が生成するスパース構造の違いを分析した。評価の結果、同程度のスパース率およびゼログループ率であっても、ゼロの空間分布やカーネル構造の違いにより、サイクルレベルの高速化率に差が生じることを確認した。

特に、分散ベース指標は、推論精度を維持しつつ計算スキップに有利なゼログループ構造を形成しやすく、本アーキテクチャに適したスパース構造を誘導する傾向があることが分かった。また、ゼロマスクの類似度解析を通じて、各ブルーニング指標が異なるカーネル構造を生成していることを定量的に示した。

これらの結果は、本アーキテクチャにおいて効率的な計算削減を実現するためには、単にスパース率を高めるだけでは不十分であり、グループ化規則および計算スキップ条件に適合したスパース構造を形成することが重要であることを示している。したがって、再構成アーキテクチャの性能を最大限に引き出すためには、アーキテクチャの実行特性を考慮したモデル構造設計が不可欠である。

謝 辞

本研究は、JSPS 科研費 JP23K16856, JP23H05489 の助成を受けた研究です。

文 献

- [1] 井上 祐, 堀 篤史, 丸亀 孝生, 浅井 哲也, Schmid A., 安藤 洸太, “可変並列性再構成アーキテクチャの計算削減手法とその最適化ソフトウェア,” リコンフィギャラブルシステム研究会 (RECONF), 富山国際会議場 (富山), 2025 年 12 月 1–3 日.
- [2] Hori A., Inoue Y., Arai F., Marukame T., Asai T., and Ando K., “Design exploration of a reconfigurable architecture with variable parallelism for neural network acceleration,” 2025 IEEE International Conference on Device Technologies for Diversified Applications (DTDA 2025), Sendai, Japan, Oct. 2025.
- [3] N. P. Jouppi, et al., “In-Datacenter Performance Analysis of a Tensor Processing Unit,” arXiv:1704.04760 [cs.AR], 2017.
- [4] Y.-H. Chen, J. Emer, and V. Sze, “Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks,” 2016

- ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), pp. 367–379, 2016.
- [5] Z. Du, et al., “ShiDianNao: Shifting vision processing closer to the sensor,” 2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), pp. 92–104, 2015.
- [6] S. Han, et al., “EIE: Efficient Inference Engine on Compressed Deep Neural Network,” arXiv:1602.01528 [cs.CV], 2016.
- [7] J. Albericio, et al., “Cnvlutin: Ineffectual-Neuron-Free Deep Neural Network Computing,” 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), pp. 1–13, 2016.
- [8] A. Parashar, et al., “SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks,” arXiv:1708.04485 [cs.NE], 2017.
- [9] S. Han, et al., “Learning both Weights and Connections for Efficient Neural Networks,” arXiv:1506.02626 [cs.NE], 2015.
- [10] Y. He, X. Zhang, and J. Sun, “Channel Pruning for Accelerating Very Deep Neural Networks,” arXiv:1707.06168 [cs.CV], 2017.
- [11] H. Li, et al., “Pruning Filters for Efficient ConvNets,” arXiv:1608.08710 [cs.CV], 2017.
- [12] P. Molchanov, et al., “Pruning Convolutional Neural Networks for Resource Efficient Inference,” arXiv:1611.06440 [cs.LG], 2017.