AFR の構造化枝刈りへの適用におけるスコア集約手法の改良に関する研究

TR24006 小林　亮太　　指導教授：藤吉 弘亘

1. はじめに

大規模言語モデル（LLM）は高い性能を示す一方，数十億のパラメータによる膨大な計算コストとメモリ使用量が実用化の障壁となっている．モデルサイズを軽量化するための代表的なアプローチとして冗長なパラメータ（重み）を削除する枝刈りがある．

枝刈りは非構造枝刈りと構造化枝刈りに大別される．非構造枝刈りは重み単位で独立に削除するため高い精度の維持を達成するが，削除位置が不規則なため推論の高速化には寄与しない．一方，構造化枝刈りはニューロン単位で重みを削除するため高速化が容易だが，削除の自由度が低いため性能劣化を招くという課題がある．

本研究では，非構造枝刈り手法である Adaptive Feature Retention (AFR) を構造化枝刈りに適用することで，重み単位の重要度（枝刈りスコア）の評価を活かして，モデルサイズの削減と推論の高速化を目指す．AFR を構造化枝刈りに適用する際には，重み単位の枝刈りスコアをニューロン単位に集約する必要がある．このとき，単純平均による集約では「符号情報の喪失」と「外れ値の影響」という 2 つの問題が生じる．本研究では，これらに対処するための改善手法を提案し，LLM として Llama-3-8B を用いた評価実験により提案手法の有効性を実証する．

2. Adaptive Feature Retention (AFR)

AFR [1] は，事前学習済みモデルに対する非構造枝刈り手法である．AFR は，ReFer [2] と SNIP [3] という 2 つの枝刈りスコアを標準化した後に加算することで，事前学習で獲得した特徴空間の維持と下流タスクへの適応を両立する．重み θ_n に対する枝刈りスコア $S_{AFR}(\theta_n)$ は式 (1) のように定義される．

$$S_{AFR}(\theta_n) = \mathcal{Z}\left(\left|\frac{\partial \sum_{l=1}^{L} F_{svd}^{l}}{\partial \theta_n}\theta_n\right|\right) + \mathcal{Z}\left(\left|\frac{\partial \mathcal{L}}{\partial \theta_n}\theta_n\right|\right) \quad (1)$$

ここで，$\mathcal{Z}(\cdot)$ は標準化を表し，第 1 項は ReFer，第 2 項は SNIP に相当する．F_{SVD}^{l} は，レイヤー l の出力する特徴量に対する特異値分解による特異値の平均である．ReFer は特徴表現の維持に焦点を当てているため，下流タスクへの学習の際に必要な重みを削除する可能性がある．

SNIP は，目的関数の勾配と重みの積を枝刈りスコアとする非構造枝刈り手法である．事前学習済みモデルでは多くの重みの勾配が小さく，$\frac{\partial \mathcal{L}}{\partial \theta_n} \approx 0$ となり，SNIP のみでは適切な枝刈りスコアの評価が困難である．

3. AFR の構造化における課題

本研究では，AFR の重み単位の枝刈りスコアの評価を構造化枝刈りに適用して，推論の高速化を行う．本章では，その際に生じる問題点を明確にする．

3.1 単純平均による構造化

非構造手法である AFR を構造化手法に適用するために図 1 に示すような単純平均による集約を考える．まず，重み行列の各要素に対して枝刈りスコアを算出し，式 (2) に示すようにニューロン単位で重みの枝刈りスコアを集約して平均スコア $\bar{S}_j$ を求める．

$$\bar{S}_j = \frac{1}{m}\sum_{i=1}^{m} |S_{ij}| \quad (2)$$

ここで，S_{ij} はニューロン j の i 番目の重みに対する枝刈りスコア，m はニューロンあたりの重み数である．最後に，ニューロン単位の平均スコアから低い順に枝刈り率に応じた数のニューロンを削除する．

3.2 単純平均による構造化の問題点

単純平均による枝刈りを LLM モデルに適用して，自然言語のデータセットで評価した場合，表 1 に示すように大幅な性能低下が生じた．この原因に関して，以下の 2 つの問題が存在すると考える．

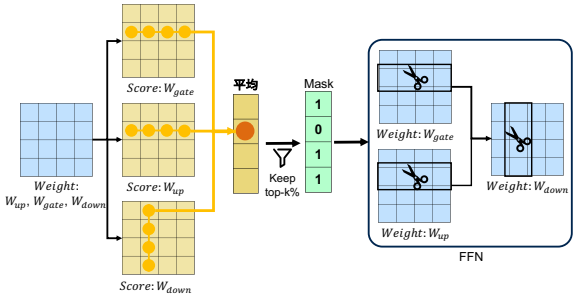

図 1: 構造化枝刈りのための枝刈りスコア集約の処理

表 1: 予備実験結果（accuracy %）

枝刈率	手法	WinoG	HellaS	ARC-e	ARC-c	MMLU	平均
0%	Llama-3-8B	72.61	79.16	77.74	53.33	62.13	68.99
20%	AFR（非構造）	71.59	73.79	76.94	48.38	58.85	65.91
	AFR（単純平均）	59.35	53.63	43.48	29.35	30.14	43.19
50%	AFR（非構造）	60.62	50.64	55.35	32.94	35.07	46.92
	AFR（単純平均）	52.25	29.29	29.92	26.02	23.07	32.11

問題 1: 最適化方向の一貫性情報の喪失

ReFer と SNIP の枝刈りスコアは勾配と重みの積として計算され，その符号は最適化過程における重みの挙動を表現する．勾配降下法による重みの更新則 $\theta \leftarrow \theta - \eta\frac{\partial \mathcal{L}}{\partial \theta}$ において，枝刈りスコアの符号は重みの最適化方向を示す指標となる．正の場合は重みの絶対値が減少する方向に，負の場合は増加する方向に働く．構造化枝刈りでは，ニューロン単位で削除を行うため，ニューロン内の重みの最適化方向の協調性が重要となる．全ての重みが同一方向に最適化される場合，そのニューロンは構造的に一貫した役割を持つ．一方，最適化方向が混在している場合，ニューロン内で相殺効果が生じており，全体としての寄与は限定的である．しかし，式 (2) では重み単位で絶対値を取るため，ニューロン内の最適化方向の一貫性という構造的特性が評価できず，ニューロン単位の重要度が適切に評価されない．

問題 2: 外れ値の影響

図 2 は，ある層における ReFer の重み単位の枝刈りスコア分布である．図 2 より，大部分の枝刈りスコアは比較的狭い範囲に分布しているが，両端に極端に大きな値や小さな値を持つ外れ値が少数存在していることがわかる．実際，全体の枝刈りスコアのレンジは，-2,884〜9,803 であり，その幅は 12,687 と大きな値である．ここで極値の 2%を削除するとレンジの幅は 48.799 となる．単純平均では，外れ値の有無によりニューロン間の平均スコアの大小関係が変化して，ニューロン単位の枝刈りスコアが適切に評価されないという問題がある．

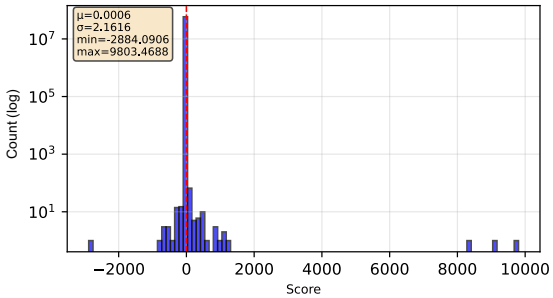

図 2: AFR の枝刈りスコアの分布例

4. 提案手法

前章で述べた 2 つの問題に対処するため，新たな枝刈りスコアの集約手法を提案する．

4.1 絶対値枝刈りスコア集約

問題 1 に対処するため，集約後の絶対値処理を提案する．ニューロンに含まれる各重みの枝刈りスコアを符号付きのまま平均し，平均化した後に絶対値を取る．これにより，符号が揃っているニューロンは平均後も大きな値を保ち，符号が混在しているニューロンは平均により小さくな

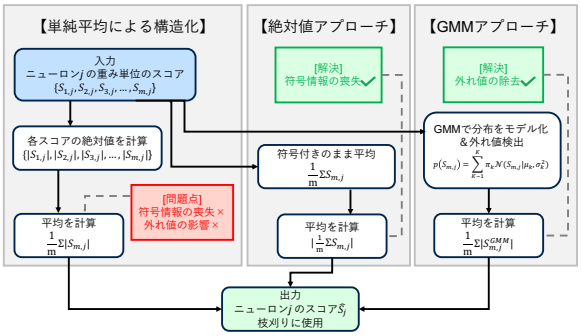

図 3: 提案手法の概要

る．このように，ニューロン内の符号の一貫性を評価でき，構造化枝刈りに適した枝刈りスコアの評価が可能となる．

4.2 GMM を用いた外れ値処理

問題 2 の外れ値の影響を軽減するため，Gaussian Mixture Model（GMM）を用いた外れ値処理手法を提案する．ニューロン j の枝刈りスコア集合 $\{S_{1j}, S_{2j}, \ldots, S_{mj}\}$ に対し，GMM により枝刈りスコア分布をモデル化する．各枝刈りスコア S_{ij} の確率密度 $p(S_{ij})$ を以下のように定義する．

$$p(S_{ij}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(S_{ij}|\mu_k, \sigma_k^2) \tag{3}$$

ここで，π_k, μ_k, σ_k^2 はそれぞれ混合比，平均，分散であり，EM アルゴリズムで推定する．成分数 K は BIC により $K \in \{1,2,3,4,5\}$ から最適な値を選択する．推定されたモデルにより各枝刈りスコアの密度を評価し，密度が下位 2%に該当するものを低密度と判定する．次に，枝刈りスコアを昇順にソートし，分布の両端から連続して低密度が続く範囲を外れ値として検出し，その境界の枝刈りスコアで置換する．これにより，極端な値の影響を抑制しつつ，データ数を保ったまま安定した集約が可能となる．

4.3 提案手法の枝刈りスコア

4.1 節と 4.2 節で提案する手法を組み合わせた手法も提案する．これは，符号付きの重み単位の枝刈りスコアに対して GMM 処理を適用した後，平均を取り，最後に絶対値を取る処理となる．具体的には，GMM 処理後の枝刈りスコア S'_{ij} を用いて以下のように計算する．

$$\bar{S}_j^{abs+GMM} = \left| \frac{1}{m} \sum_{i=1}^{m} S'_{ij} \right| \tag{4}$$

この手法により，符号の一貫性評価と外れ値の影響抑制の両方を実現し，安定した枝刈りスコアの評価が期待される．

5. 評価実験

提案手法の有効性を検証するため，Llama-3-8B を対象とした評価実験を実施した．

5.1 実験概要

Llama-3-8B の各ブロックにおける FFN に対して構造化枝刈りを適用する．枝刈り率は既存研究で広く採用されている 20% と 50% とする．性能評価には，WinoGrande, HellaSwag, ARC-easy/ARC-challenge, MMLU の 5 つのベンチマークデータセットを使用し，accuracy を評価指標とする．比較手法として，非構造 AFR，構造化 AFR（単純平均，絶対値のみ，GMM のみ，絶対値+GMM），および既存手法（LLM-Pruner，LoRAP，CFSP）を用いる．

5.2 実験結果

表 2 に提案手法の評価結果を示す．単純平均と比較して提案手法である絶対値のみの場合は 20% 枝刈りで 13.04 ポイント，GMM のみの場合は 16.31 ポイントの精度向上を示した．絶対値+GMM の場合が最も高い性能を示し，20% 枝刈りで 18.06 ポイント，50% 枝刈りで 11.33 ポイントの精度向上を達成した．これは，符号情報の損失と外れ値の影響の両方が解決されたことによる相乗効果である．

5.3 既存手法との比較

提案手法（絶対値+GMM）を既存の構造化枝刈り手法と比較する．表 3 に既存手法との比較結果を示す．提案手法は 20% 枝刈りで 61.25% を達成し，他の既存手法を上回った．50% 枝刈りでも 43.44% を達成し，LLM-Pruner（34.87%）と比較して 8.57 ポイント向上した．

表 2: 提案手法の精度評価（accuracy %）

枝刈率	手法	WinoG	HellaS	ARC-e	ARC-c	MMLU	平均
0%	Llama-3-8B	72.61	79.16	77.74	53.33	62.13	68.99
20%	非構造枝刈り						
	AFR	71.59	73.79	76.94	48.38	58.85	65.91
	構造化枝刈り						
	AFR（単純平均）	59.35	53.63	43.48	29.35	30.14	43.19
	AFR（絶対値のみ）	67.40	67.90	70.20	42.83	32.81	56.23
	AFR（GMM のみ）	68.57	**70.70**	70.58	44.80	42.86	59.50
	AFR（絶対値+GMM）	**68.90**	69.70	**72.81**	**46.16**	**49.32**	**61.25**
50%	非構造枝刈り						
	AFR	60.62	50.64	55.35	32.94	35.07	46.92
	構造化枝刈り						
	AFR（単純平均）	52.25	29.29	29.92	26.02	23.07	32.11
	AFR（絶対値のみ）	52.96	37.88	43.18	25.94	23.04	36.60
	AFR（GMM のみ）	**58.41**	**48.59**	48.36	29.78	27.78	42.58
	AFR（絶対値+GMM）	56.75	47.33	**51.89**	**29.95**	**28.27**	**43.44**

表 3: 既存手法との比較（accuracy %）

枝刈率	手法	WinoG	HellaS	ARC-e	ARC-c	MMLU	平均
0%	Llama-3-8B	72.61	79.16	77.74	53.33	62.13	68.99
20%	提案手法	68.90	69.70	**72.81**	**46.16**	49.32	**61.25**
	LLM-Pruner	69.85	69.02	63.59	40.53	48.36	58.26
	LoRAP	**71.19**	**70.48**	69.61	45.48	44.57	60.27
	CFSP	68.67	68.06	67.63	42.92	**50.71**	59.60
50%	提案手法	56.75	**47.33**	**51.89**	**29.95**	**28.27**	**43.44**
	LLM-Pruner	52.49	35.50	37.88	25.51	22.95	34.87
	LoRAP	**57.30**	40.19	42.60	26.79	26.85	38.75
	CFSP	57.06	43.13	48.53	28.50	26.61	40.77

5.4 リソース削減効果の評価

表 4 に各枝刈り率におけるリソース削減効果を示す．50% 枝刈りでパラメータ数及び VRAM 使用量 35.1% 削減，推論速度 1.57 倍の高速化を達成した．構造化枝刈りにより実用的な高速化が実現された．

表 4: リソース削減効果の評価

枝刈率	パラメータ数		VRAM [GB]		推論速度 [ms/sample]	
0%	8.03B	—	16.06	—	22.98	1.0x
20%	6.90B	14.0%↓	13.08	14.0%↓	20.89	1.10x
50%	5.21B	35.1%↓	10.42	35.1%↓	14.64	1.57x

6. おわりに

本研究では，非構造枝刈り手法を構造化枝刈りに適用することにより，精度の維持と推論の高速化を両立する手法を提案した．

単純平均による枝刈りスコアの評価では，符号情報の損失と外れ値の影響により性能が著しく劣化することを発見した．これらに対処するため，絶対値アプローチと GMM による外れ値除去を組み合わせた手法を提案した．Llama-3-8B を対象とした評価実験により，提案手法は単純平均手法と比較して大幅な性能向上を達成し，既存の構造化枝刈り手法と比較して同等以上の性能を示した．また，50% 枝刈りでパラメータ数及び VRAM 使用量 35.1% 削減，推論速度 1.57 倍の高速化を達成した．

今後の課題として，Attention モジュールへの拡張，集約方法の最適化，層ごとの適応的枝刈り率設定が挙げられる．

参考文献

[1] 新田常顧 等, “事前学習済みモデルの知識維持と下流タスク適応を両立した Single-shot Foresight Pruning”, 画像の認識・理解シンポジウム, 2025

[2] 新田常顧 等, “事前学習モデルの特徴表現を維持した Single-shot Foresight Pruning”, 画像の認識・理解シンポジウム, 2024

[3] N. Lee, *et al.*, “SNIP: Single-shot Network Pruning based on Connection Sensitivity”, International Conference on Learning Representation (ICLR) , 2019.

研究業績

[1] 小林亮太，平川翼，山下隆義，藤吉弘亘, “非構造的枝刈り手法の構造化手法への変換による LLM の軽量化”, 東海支部連合大会, 2025.