Transformer モデルの量子化における活性分布の整形による精度維持に関する研究

TR23009 若山 浩之　　指導教授：藤吉 弘亘

1. はじめに

Transformer モデルは, 自然言語処理や画像認識分野の幅広いタスクにて卓越した性能を発揮している. 本モデルは高精度化に伴いモデルサイズが増大しており, 小規模なハードウェア環境で動作させることが困難である. その解決法として, 枝刈り, 蒸留, 量子化等によるモデル圧縮が行われている. 本研究では, モデル内部の数値を低ビットで表現する量子化に着目し, モデル圧縮を図る. Transformer モデルの主要な計算量を占める Linear 層において, アクティベーションを 8bit, 重みを 1.58bit に量子化することで, モデル全体の圧縮を図る.

Transformer モデルの MLP ブロックで使用する 2 層の Linear 層には, 非線形変換として GELU 関数を用いる. GELU 関数の出力は小さな正の値を多く含むロングテール分布の特性を持つ. この特性が, 量子化時に数値の丸め誤差を増大させる要因となる [1]. 本研究では, この課題を解決するために, GELU 関数の出力分布を整形する Activation Shaping Connection (ASC) を提案する. 提案手法により, 高精度を維持しながらモデルを圧縮が可能になる.

2. 量子化

量子化とは, 高ビットの数値表現を低ビットに変換し, モデル圧縮する技術である. 通常, モデルの中間特徴量であるアクティベーションや重みは 32bit 浮動小数点で表現されるが, これを 8bit や 4bit に圧縮することでモデル圧縮を図る. しかし, 量子化による丸め誤差が累積し, 精度低下を招く可能性がある. そのため, 高精度を維持するための工夫が必要である.

Transformer モデルの主要な計算量を占める Linear 層の量子化は, モデル圧縮において重要となる. BitNet[2] では Transformer の Linear 層を量子化してモデルを圧縮する手法が提案されている. Linear 層に量子化を適用した BitLinear 層の計算方法を図 1 に示す.

Layer Norm は, アクティベーションの平均と分散を計算して標準化を行う. この操作により, アクティベーション分布が一定の範囲内に収まり, 量子化による誤差を抑制できる.

Layer Norm を通した, アクティベーションを, 8bit に量子化する. その際, 入力値を特定の範囲内に収まるよう, スケーリングと整数化を段階的に行う. まず, アクティベーション行列 $x \in \mathbb{R}^{N \times D}$ の全要素の絶対値に対する最大値を $\gamma = \max(|x|)$ と定義し, これを基準に x の各要素を $[-1, 1]$ の範囲にスケーリングする. 次に, スケーリングした値を $[-128, 127]$ の範囲に再スケーリングし, 最後に整数化を行う. この一連の操作を式 (1) に示す.

$$x_{\text{quant}} = \text{round}\left(\text{Clip}\left(\frac{x \cdot 128}{\gamma}, -128, 127\right)\right) \tag{1}$$

ここで, Clip 関数は, 値を $[-128, 127]$ の範囲に収める役割を持つ.

重みは 1.58bit を量子化する. ここで, 1.58bit とは, 重みが 3 値 $\{-1, 0, 1\}$ に量子化される際の情報量を示し, $-\log_2(1/3) = \log_2 3$ に基づくエントロピーで算出される. この量子化では, 重み行列 $W \in \mathbb{R}^{N \times D}$ を特定の範囲に収め, スケーリング後に 3 値 $\{-1, 0, 1\}$ に変換する. まず, 重み行列 W の全要素に対して, 絶対値の平均値 $\beta = \text{mean}(|W|)$ を基準にスケーリングを行い, RoundClip 関数を用いて 3 値化を行う. この操作を式 (2) に示す.

$$w_{\text{quant}} = \text{RoundClip}\left(\frac{w}{\beta}, -1, 1\right) \tag{2}$$

ここで, RoundClip 関数は, 値を {-1, 0, 1} の最も近い値に丸める役割を持つ.

逆量子化では, 量子化したアクティベーションと重みを元のスケールに復元する処理を行う. 量子化により出力がスケーリングした値となるため, 補正が必要である. この処理を式 (3) に示す.

$$y = w_{\text{quant}} \cdot x_{\text{quant}} \cdot \frac{\beta\gamma}{128} \tag{3}$$

ここで, γ はアクティベーションのスケーリング係数, 128 は量子化による定数スケール, β は重みのスケーリング係数を表す. これらの係数を組み合わせることで, 量子化した出力を補正し, 元のスケールに再調整する. この逆量子化処理により, スケーリングの影響を最小限に抑えた出力を得る.

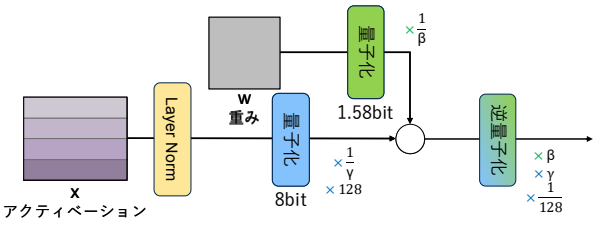

図 1: BitLinear 層の計算方法

3. GELU 関数によるロングテール分布の課題点

本研究では, BitNet で提案された Linear 層の量子化手法を, 画像認識タスクに特化した Vision Transformer (ViT) に適用する. ViT は計算量の多くが Linear 層に集中しているため, Linear 層を BitLinear 層へ置き換えることでモデル圧縮を図る. また, GELU 関数によるロングテール分布が量子化時に与える影響を分析し, その課題を解決するための手法を提案する.

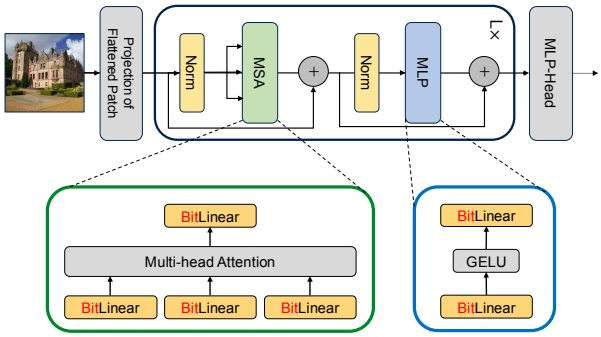

図 2: BitLinear 層を使用した Vison Transformer

BitLinear 層を使用した ViT のモデル構造を図 2 に示す. ViT の MLP ブロックでは, 非線形活性化関数として GELU 関数が採用されている. この GELU 関数の出力はロングテール分布を持つ特性があり, アクティベーションを $[-128, 127]$ の範囲で量子化する際, 負の値に対応する $[-128, 0]$ の範囲がほとんど利用されない課題が生じる. 量子化では数値の表現範囲が非効率に利用されることで, 使用可能な数値の幅が無駄に使用され, その結果, モデルの表現力が低下する.

図 3 に示すガウス分布とロングテール分布の丸め誤差に関するシミュレーション結果から, 両分布における平均二乗誤差の違いを示す. Linear 層および BitLinear 層の丸め誤差を比較した結果, ガウス分布では平均二乗誤差が 0.02 であるのに対し, ロングテール分布では 1.33 と, ロングテール分布では丸め誤差がさらに大きくなる. これらの結果は, ロングテール分布が量子化に伴う丸め誤差の累積を引き起こす原因となることを示唆している.

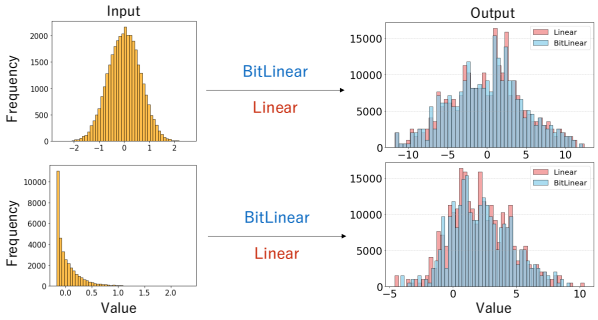

図 3: 分布形状による丸め誤差シミレーション

4. 提案手法

本研究では, GELU 関数によるロングテール分布の課題を解決するために, アクティベーション分布を整形する新たなコネクションである Activation Shaping Connection (ASC) を提案する. ASC は, GELU 関数により生じるロングテール分布を整形することで, 量子化時の丸め誤差の累積を抑制し, 精度を維持することを目指す.

4.1 Activation Shaping Connection

ASC は, アクティベーションの分布を整形することで, 丸め誤差の累積を抑制し, 学習の安定性を向上させるためのコネクションである. 本研究では, ViT の MLP ブロックで使用される GELU 関数のロングテール分布を, 正負の値がより均等に分布する形状に整形するように ASC を導入する. ASC を導入した MLP ブロックの構造を図 4 に示す. この整形処理では, BitLinear 層を用いて変換し, 分布を整形する. ASC の処理を式 (4) に示す.

$$\mathbf{y} = \mathrm{GELU}(\mathrm{BitLinear}(\mathbf{x})) \odot \mathrm{BitLinear}(\mathbf{x}) \qquad (4)$$

ここで, $\mathbf{x}$ はアクティベーション, $\odot$ は要素積を示す. 具体的には, BitLinear 層の後に GELU 関数を適用する通常の処理に加え, 新たに BitLinear 層を通過するコネクションを導入する. この導入したコネクションの出力と, 通常の処理の出力間で要素積を計算することで, 整形された出力 $\mathbf{y}$ を得る. この結果, 整形した $\mathbf{y}$ を MLP ブロックの 2 層目の BitLinear 層に入力する. これにより, GELU 関数のロングテール分布に起因する課題が解決され, 丸め誤差の累積を抑制できる.

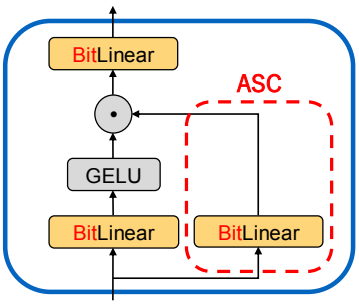

図 4: ASC を導入した MLP ブロック

4.2 ガウス分布整形の Loss の導入

MLP ブロックの 2 層目の BitLinear 層への入力分布は, ASC により得られる出力分布に依存する. そのため, ASC の出力分布が極端に偏ると, 意図した分布整形が機能しなくなるため, 適切な対称性を持たせる必要がある. そこで, Loss 関数にガウス分布への近似を促す制約を追加する. 具体的には, ASC により出力される BitLinear 層の分布をガウス分布に近似させるため, λ_{gauss} を追加して分布の整形を行う. Loss 関数の全体式を式 (5) に示す.

$$\lambda_{\mathrm{all}} = \lambda_{\mathrm{label}} + \alpha \cdot \lambda_{\mathrm{gauss}} \qquad (5)$$

ここで, λ_{label} は従来のラベルに基づく損失, λ_{gauss} は出力分布をガウス分布に近似させるための損失, α は両損失項の重み付けバランスを調整するハイパーパラメータである. λ_{gauss} は, 各層の ASC におけるアクティベーションを標準化し, 標準化したアクティベーションとガウス分布との間の KL ダイバージェンスを計算することで, アクティベーションをガウス分布に近似させる.

5. 評価実験

本章では, 提案手法の有効性を検証するために, ImageNet-1k データセットを用いて評価実験を実施する.

5.1 Activation Shaping Connection の有効性

提案手法である ASC の有効性を評価するため, 従来の MLP ブロックを使用した BitNet と, ASC を導入したモデルを比較する. 表 1 に示すように, ASC の導入により BitNet と比較して精度が 7.22 ポイント向上した. この改善は, GELU 関数のロングテール分布を適切に整形し, 丸め誤差の累積を抑制したことが主な要因であると考える. また, 量子化なしの ViT と比較して, Linear 層の圧縮率が 63.59 %に達した. 一方で, ガウス分布に近似する Loss のハイパーパラメータ α を 1.0 に設定すると精度が低下した. しかし, $\alpha = 0.0009$ に設定した場合は, Loss を導入しない場合より精度が向上した. これは, ガウス分布の近似より分布整形が安定し, 丸め誤差の抑制および学習全体の安定性が向上したと考える.

表 1: ASC を用いた評価結果

モデル	量子化	ASC	α	サイズ [MB]	精度 [%]
ViT	-	-	-	2084.33	64.68
BitNet	√	-	-	531.14	49.38
	√	√	0	758.76	56.60
	√	√	1.0	758.76	48.93
	√	√	0.9	758.76	53.56
	√	√	0.09	758.76	55.98
	√	√	0.009	758.76	56.61
	√	√	0.0009	758.76	**56.97**

5.2 ASC によるアクティベーション分布の可視化

ASC の有効性を検証するため, アクティベーション分布の変化を可視化する. 図 5 に示すように, ASC の各ルートで得られる分布形状を比較した結果, アクティベーションが最終的に正負の値がより均等に分布する形状に整形されることを確認した. この結果, ASC は GELU 関数によるロングテール分布の課題を抑制し, 次の BitLinear 層での丸め誤差の累積を抑制できることを示す.

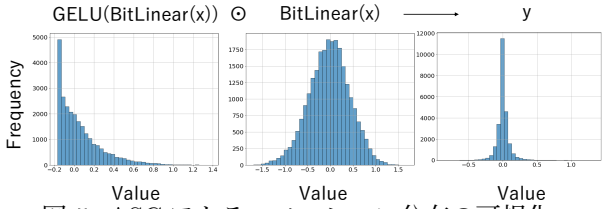

図 5: ASC アクティベーション分布の可視化

6. おわりに

本研究では, 小規模ハードウェア環境で動作可能な Transformer モデルを実現するため, GELU 関数のロングテール分布を整形し, 丸め誤差を抑制する ASC を導入した量子化手法を提案した. 評価実験の結果, 提案手法が精度向上とモデル圧縮の両立に有効であることを確認した. また, 小規模なハードウェア環境への適用可能性が示され, 実用化への期待が高まる結果となった. 今後は, より複雑なモデルや多様なタスクへの適用可能性についてさらなる検証を進める予定である.

参考文献

[1] L. Yijiang, *et al.*, “Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers” , WACV ,2023.

[2] M. Shuming, *et al.*, “The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits” , arXiv:2402.17764 ,2024.

研究業績

[1] 若山浩之 等, “CNN と ViT を組み合わせたモデルのノイズへの頑健性”, 画像の認識・理解シンポジウム, 2023.