# 確率的丸めを用いた言語モデルの量子化を意識した学習

趙 開顔[1] 田原 司睦[2] 小林 健一[2] 本田 巧[2] 山崎 雅文[2] 鶴岡 慶雅[1]
[1] 東京大学大学院 情報理工学系研究科 [2] 富士通株式会社 人工知能研究所
[1]{zhaokaiyan1006, yoshimasa-tsuruoka}@g.ecc.u-tokyo.ac.jp
[2]{tabaru, kenichi, honda.takumi, m.yamazaki}@fujitsu.com

## 概要

近年、大規模言語モデル (LLMs) の各パラメータの重みを二値や三値に量子化することで、推論時のメモリ使用量を大幅に削減できることが報告されている。しかし、これらのモデルの学習には依然として多くのメモリが必要である。その理由の一つは、これらのモデルを学習する際に、Straight-Through Estimator (STE) に必要な、(量子化されていない) 高精度の重み行列を保持する必要があるからである。そこで本研究では、学習時のメモリ使用量を削減するため、バックプロパゲーションにおいて STE を用いずに、量子化された低精度の重み行列を直接更新することを試みる。具体的には、確率的丸めを利用することで、低ビットの重みを用いる際に生じる情報の損失を防ぐ。LLaMA 構造の言語モデルを用いた実験の結果、低精度の重みのみでの学習が可能であることが明らかになった。

## 1 はじめに

大規模言語モデル (LLMs) は、機械翻訳 [1]、要約 [2]、推論 [3]、質問応答 [4] など、幅広い自然言語処理（NLP）タスクにおいて有望な解決策として注目されている。しかし、現在の LLM の規模がさらに拡大する中で、それらを学習するためには膨大なデータセットと大量の計算資源が必要になるという課題がある [5]。

量子化は、高精度のパラメータ行列を低精度形式に変換する手法であり、リソース効率の高い LLM を実現するための効果的なアプローチとして注目されている [6]。従来の量子化手法は、学習後量子化 (Post-Training Quantization, PTQ) と量子化を意識した学習 (Quantization-Aware Training, QAT) の 2 つに分類される。PTQ は、すでに事前学習された LLM の重み行列のビット精度を削減する手法である [7, 8]。一方、QAT はトレーニング中に量子化を組み込むことで、学習プロセス全体を通じて低ビット精度にモデルを適応させる [9]。

最近、QAT 手法である BitNet [10, 11] は、フル精度（FP32）のトランスフォーマーを二値 (binary) または三値 (ternary) モデルに量子化する試みを行った。この手法では、重み行列の値を{-1, 0, 1}に制約しながら、フル精度モデルと同程度性能を維持している。図 1 (a) に、BitNet のような従来の QAT 手法の学習プロセスを示す。量子化された重みと入力を基に損失を計算した後、バックプロパゲーションを通じてその損失が高精度の元の重みに伝播され、Straight-Through Estimator (STE) [12] を用いて重みが更新される。この高精度の重みは、各学習ステップで再度量子化される必要がある。この反復プロセスは、量子化が微分不可能であることから [13]、特別な勾配蓄積手法を必要とする。その結果、高精度の重み行列を学習全体を通じて保持する必要があり、これが大きなメモリ消費につながる。

特に、QAT を LLM に拡張する場合、実用的な利用には大きな課題が伴う。なぜなら、LLM のような大規模なモデルでは、重み行列を保持するために膨大なメモリが必要になるからである。例えば、1B パラメータ規模の LLM の重みを FP32 形式で保持するには約 4GB のメモリが必要だが、三値の重みを使用すればそのメモリ使用量を 0.2GB まで削減できる。このようなメモリの制約は QAT の実用性を大きく制限する要因となっている。

これらの課題に対処するため、我々は、量子化された重みを直接利用するアプローチ（以後、DQT (Direct Quantized Training) と呼ぶ）を試みる。これは、学習プロセス全体で低精度の重みのみを維持する、修正された QAT 手法である。DQT は、バックプロパゲーション中に量子化された低精度の重みを直接更新することで、STE への依存を排除することができる（図 1 (b) 参照）。具体的には、私たちは確率的丸め [14] を使用し、すべての重み行列を学習全

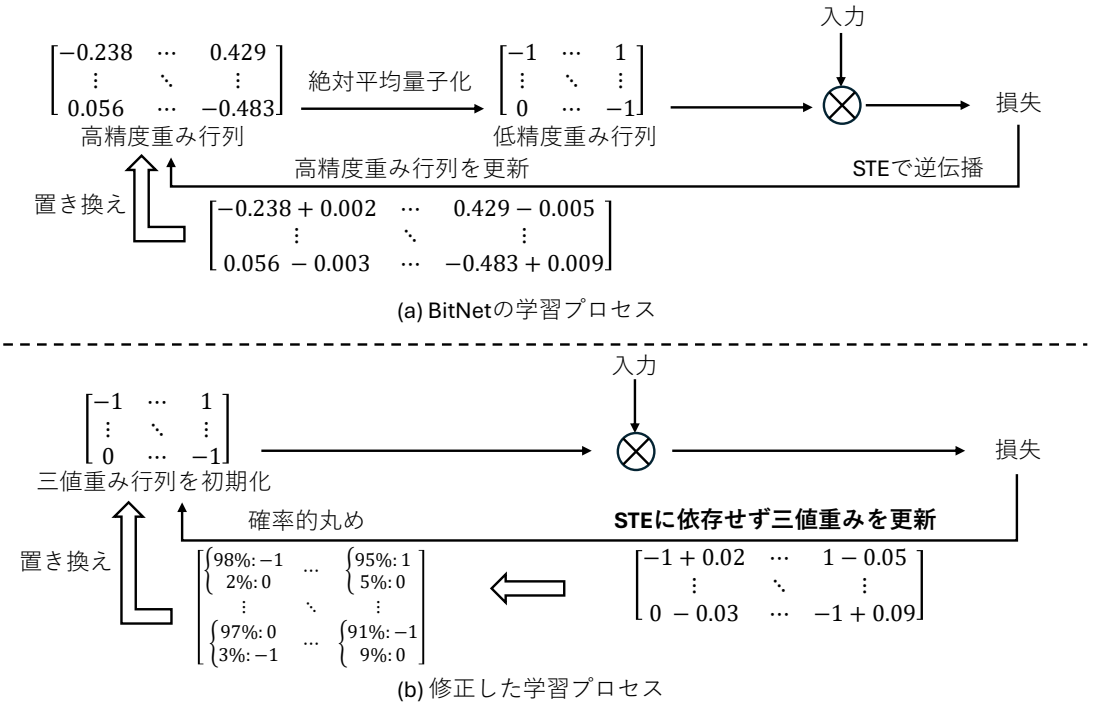


**図 1**　BitNet と私たちの修正した学習トプロセスの比較。上：BitNet の学習プロセスでは、元の高精度の重みが逆伝播プロセスで STE を使って更新される。下：私たちは低精度の重みを直接確率的丸めで更新し、各学習ステップで重み行列を量子化する必要がなくなり、重み行列を常に三値のままで保持している。

体を通じて $n$ ビット精度で固定する。この丸め技法は、値がその代表的な精度からどれだけ離れているかに基づいて、値を最も近い表現可能な精度に確率的に丸めるもので、バックプロパゲーション後も重み行列の低精度形式を保持し、低ビット重みのみを使用することによって生じる情報の損失を最小限に抑える。DQT を用いることで、モデルは各学習ステップで高精度の重み行列を量子化する必要がなくなり、理論的に QAT プロセスが加速され、メモリ使用量が削減される。さらに、DQT の軽量なメモリ依存性は、計算資源が制約された場面においても量子化の適用が可能になる。

DQT で学習した LLaMA 構造のモデルの実験結果は、次のことを示している：(1) DQT で訓練されたモデルは、重み行列が三値に制約されていても学習が可能であること、(2) 8 ビットの DQT を使用することで、モデルは BitNet b1.58 と競争力のある性能 (損失が 5%劣化) を達成できること、(3) DQT で訓練されたモデルは、推論時に三値の重みのみを使用し、BitNet と近い性能を発揮することが可能であること。私たちは、DQT が従来の QAT が抱える計算上の課題に対処するための新たな視点を提供する可能性があると考えている。

## 2　関連研究

最近の LLM に対する量子化手法は、学習後量子化 (PTQ) と量子化を意識した学習 (QAT) の 2 つに分類できる。

**学習後量子化 (PTQ)**　は、パラメータが事前に学習された後に、高精度のパラメータを低精度のものに変換する手法である。モデルの性能を維持しつつこの変換を達成するために、小規模のキャリブレーションデータが一般的に利用されている [15, 16, 8]。他にも、キャリブレーションデータを必要としない方法が存在している [17]。PTQ の課題は、圧縮効率とパフォーマンス低下の最小化とのバランスを取ることにあり、これには計算上のトレードオフが伴い、一般的な用途においては非自明な課題となる。さらに、学習された高精度の表現と制約されたビット幅との間にギャップがあるため、PTQ の性能は QAT の性能に一貫して劣っている [18, 19]。

**量子化を意識した学習 (QAT)**　は学習プロセス中にモデルパラメータの量子化を組み込むことで、高精度のパラメータ学習と量子化のギャップを埋める。LLM への QAT の最初の適用は Liu ら [19] によるもので、彼らはデータフリーの蒸留手法を提案

し、モデルを 4 ビットに量子化した。Xu ら [20] は蒸留手法を拡張し、学習可能なベクトルを導入して二値量子化を実現した。最近では、BitNet [10, 11] が提案され、重みの値を{-1, 0, 1}に制約し、スクラッチから QAT を実現した。しかし、量子化プロセスは微分不可能であるため、学習中に STE [12] のような特別な勾配近似手法が一般的に使用される。この方法は効果的ではあるが、学習プロセス中に高精度の重みが常に保持されるため、学習が遅くなり、計算メモリの使用量が増加する。これらの非効率性は、例えば数十億パラメータ規模のモデルに QAT を適用する際に特に顕著となり、QAT の実用化を妨げる大きな障害となる。これに対処するため、本研究では、従来の QAT 手法と比較してメモリ使用量を大幅に削減し、より効率的で実用的な QAT 手法の可能性を探る。

# 3 手法

## 3.1 確率的丸め

確率的丸め (Stochastic Rounding) は Neumann ら [14] に由来し、元々は数値計算におけるバイアスを軽減するために使用されていたが、最近では深層学習モデルにも応用されている [21]。これは、値とその最も近い表現可能な精度との距離に基づいて値を確率的に丸める手法である。高精度の値 $x$ が与えられた場合、確率的丸めは以下の式で定義される [21, 22]：

$$\mathrm{SR}(x) = \begin{cases} \lfloor x \rfloor, \text{ with } p = \lceil x \rceil - x \\ \lceil x \rceil, \text{ otherwise} \end{cases}, \tag{1}$$

ここで、$p$ は $x$ を floor($x$) または ceil($x$) に変換する確率を表している。この方法により、高精度の値を自然に低精度の値に量子化することができる。

## 3.2 確率的丸めに基づいた学習プロセス

次に、DQT の詳細について説明する。図 1(b) に示されているように、私たちは高精度の重み行列ではなく、低精度の重み行列から学習を始める。この初期化は、ランダムに生成された重み行列 $W$ に対して [11] 絶対平均量子化 (AbsMean Quantization) を利用することで実現する。重み行列 $W$ の絶対平均値は次のように表される：

$$\mathrm{AbsMean}(W) = \frac{1}{k} \sum_{i=1}^{k} |w_i|, \tag{2}$$

ここで、$w_i$ は重み行列 $W$ の $i$ 番目の要素を表す。$n$ ビット量子化の場合、量子化の範囲を制約するために、$Q_n = -2^{n-1}$ および $Q_p = 2^{n-1} - 1$ が定義される。スケーリング係数 $s$ は次のように定義される：

$$s = \frac{Q_p}{\mathrm{AbsMean}(W)}. \tag{3}$$

最後に、量子化された重み $\tilde{W}$ は次の式で表すことができる：

$$\tilde{W} = \mathrm{Clip}[\mathrm{Round}(W \cdot s), Q_n, Q_p]/s, \tag{4}$$

ここで、Clip() 関数はすべての値が範囲 $[Q_n, Q_p]$ 内に収まるようにし、Round() 関数は最も近い整数を返す。この方法により、量子化された $\tilde{W}$ を $n$ ビットに制約することができる。一方、入力と活性化関数については、BitNet [10, 11] で導入された設定に従い、8 ビットに量子化する。

各学習ステップで、$\tilde{W}$ と入力に基づいて言語モデリングのクロスエントロピー損失を計算した後、まず optimizer によって更新用の高精度重み行列 $W'$ を計算させる。従来の QAT では、STE を適用し、$W'$ が元の高精度の $W$ を置き換え、その後、次の学習ステップで再び式 (2) から式 (4) の量子化プロセスを経る。しかし、私たちの手法では、このプロセスを簡略化し、$W'$ に直接式 (1) を用いて確率的丸めを適用する。

$$\widetilde{W} = \mathrm{SR}(W'), \tag{5}$$

これにより、高精度の重みを保持する必要がなく、$n$ ビットを維持できるため、STE を使わず、式 (2) から式 (4) のプロセスを省略できる。DQT では、$\tilde{W}$ を直接 $\widetilde{W}$ に置き換えて、次の学習ステップに進む。したがって、学習全体を通じて重み行列が常に $n$ ビットに制約されることが保証される。これが従来の QAT 手法との最大の違いである。

# 4 実験

## 4.1 実装の詳細

**データセット** 英語版 Wikipedia データセット (20231101.en)[1] を使用して、ベースラインと DQT のモデルを事前学習させる。訓練データの最大長は 512 に設定している。512 より長いテキストは個別に分割し、512 未満のテキストにはパディングを適用する。その結果、元の 640 万の文から 1400 万の文を含むデータセットを得ることができた。

1) https://huggingface.co/datasets/wikimedia/wikipedia

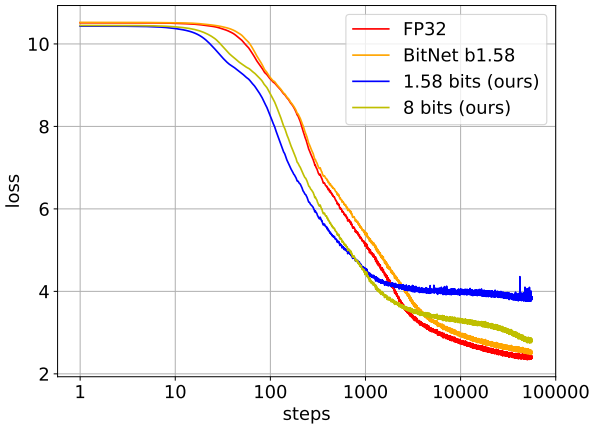


**図 2**　DQT と他のベースラインとの比較。横軸は学習ステップを表しており、各モデルに最適な学習率を個別に選択している。

**ベースライン**　DQT と比較するために、2 種類のベースラインを選択した。1 つ目は再現した FP32 モデルである。2 つ目のベースラインは再現した BitNet b1.58 で、三値重みを使用する QAT 手法である。BitNet と公平に比較するため、推論時に DQT モデルも三値重みのみを使用するように制約している。

ハイパーパラメータやハードウェアなど、その他の実装に関する詳細は付録 A.1 に記載している。

## 4.2　実験結果

図 2 は、DQT のモデルと再現した LLaMA、BitNet b1.58 の学習損失を示している。図 2 において、青線は DQT の三値実装を示しており、収束可能であることを示している。DQT の三値モデルと再現した FP32 モデルを比較すると、パフォーマンスにギャップが見られる。これは予想通りで、DQT モデルは学習中に 1.58 ビットの情報のみで動作するため、FP32 モデルと比較してその能力が制限されている。しかし、ビット幅が長いモデル（例えば 8 ビット）を使用し、重みパラメータを 8 ビットに量子化し、これらの 8 ビット値のみを学習中に維持する場合、FP32 モデルに近いパフォーマンスを示している。

BitNet との比較について、DQT の三値モデルは学習中に高精度の重みを使用しないため、パフォーマンスは低くなる。しかし、プロセス全体で三値の重みのみを維持するため、BitNet よりも必要とするメモリが少ない。DQT で 8 ビット量子化を有効にした場合と比較しても、BitNet は高精度の情報が必要であるため、メモリの使用量は DQT よりも多い。

| Methods | Loss | PPL |
|---|---|---|
| FP32 (reproduced) | 5.39 | 41.93 |
| BitNet b1.58 (reproduced) | 5.52 | 45.83 |
| DQT 1.58 bits (ours) | 6.20 | 73.41 |
| DQT 8 bits (ours) | 5.80 | 55.75 |
| DQT 8 bits (ternary Inf.)† | 5.93 | 60.98 |

**表 1**　WikiText-2 での評価結果。損失と困惑度スコアを報告している。† は、8 ビットで学習され、三値重みで推論を行ったモデルを示している。

## 4.3　三値重みでの推論

私たちの提案する DQT では、STE を使用していないため、順伝播および逆伝播は直接 $n$ ビットの重み行列に対して行われる。これは、大きなビット幅で訓練された DQT モデルが推論時にもそのビット幅のままであることを意味している。BitNet b1.58 との公正な比較を行うため、私たちはモデルを三値推論を行うように適応させる。これを実現するために、順伝播時に三値の重みを使用し、逆伝播時には STE を通じて $n$ ビットの重みを維持する。

BitNet b1.58 と私たちの DQT モデルの推論性能の違いを示すため、WikiText-2 [23] [2] のテストセットを使用して、それぞれの損失と困惑度スコアを報告する。その結果を表 1 に示す。WikiText-2 のテキスト長は 512 に設定し、これは訓練セットと一致している。表 1 に示されているように、DQT 8 ビットモデルと BitNet の WikiText-2 における評価損失の違いは最小限で、約 5%の劣化にとどまっている。三値推論を適用した場合、8 ビット推論に比べてわずかに性能が低下するが、依然として近い結果を示しており、三値推論においてもロバストであることが示されている。

# 5　終わりに

本研究では、低精度重み行列を確率的丸めで直接更新することで、メモリ使用量を削減する修正された QAT 手法である Direct Quantized Training（DQT）を試みた。実験により、DQT が高精度の重みを保持しなくても学習を可能であることを示した。さらに、ビット幅を 8 ビットに拡張した場合、DQT モデルは学習と推論の両方で FP32 モデルおよび BitNet b1.58 と近いパフォーマンスであることを確認した。

2）https://huggingface.co/datasets/Salesforce/wikitext/viewer/wikitext-2-v1/test

# 参考文献


[1] Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. A paradigm shift in machine translation: Boosting translation performance of large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[2] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization. **Transactions of the Association for Computational Linguistics**, Vol. 12, pp. 39–57, 2024.

[3] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In **Proceedings of the 36th International Conference on Neural Information Processing Systems**, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[4] OpenAI. Gpt-4 technical report. 2023.

[5] Jiangfei Duan, Shuo Zhang, Zerui Wang, Lijuan Jiang, Wenwen Qu, Qinghao Hu, Guoteng Wang, Qizhen Weng, Hang Yan, Xingcheng Zhang, Xipeng Qiu, Dahua Lin, Yonggang Wen, Xin Jin, Tianwei Zhang, and Peng Sun. Efficient training of large language models on distributed infrastructures: A survey, 2024.

[6] Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting llm quantization. **arXiv preprint arXiv:2405.18137**, 2024.

[7] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 32. Curran Associates, Inc., 2019.

[8] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training compression for generative pretrained transformers. In **The Eleventh International Conference on Learning Representations**, 2023.

[9] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, June 2018.

[10] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models, 2023.

[11] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024.

[12] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. **arXiv preprint arXiv:1308.3432**, 2013.

[13] Shangyu Chen, Wenya Wang, and Sinno Jialin Pan. Metaquant: Learning to quantize by learning to penetrate non-differentiable quantization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 32. Curran Associates, Inc., 2019.

[14] John Von Neumann and Herman Heine Goldstine. Numerical inverting of matrices of high order. 1947.

[15] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In **Proceedings of the 37th International Conference on Machine Learning**, ICML'20. JMLR.org, 2020.

[16] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. In **International Conference on Learning Representations**, 2021.

[17] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**, pp. 13169–13178, 2020.

[18] Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models, 2024.

[19] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 467–484, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[20] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[21] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In **Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37**, ICML'15, p. 1737–1746. JMLR.org, 2015.

[22] Zhenyu Zhang, Ajay Jaiswal, Lu Yin, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. Q-galore: Quantized galore with int4 projection and layer-adaptive low-rank gradients, 2024.

[23] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[24] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 2023.

# A 付録

## A.1 実装の詳細

実験には、NVIDIA A100 GPU で fake simulation 量子化を使用して学習を行った。モデルの構造は 12 層の LLaMA [24] に基づき、隠れ層のサイズと中間層のサイズをそれぞれ 768、2048 に設定した。アテンションヘッドの数は 12 に設定し、最大位置エンベディングは最大入力長に従い、512 に設定した。モデルとベースラインの総パラメータ数は約 130M である。ベースラインでは AdamW を最適化アルゴリズムとして使用し、DQT モデルでは確率的丸めを組み込んだ修正した AdamW を使用した。コサインスケジューラを 2000 ステップのウォームアップとともに適用した。すべてのモデルは、4 つの A100 80G GPU で 1 エポックの学習を行い、バッチサイズは各 GPU で 64 に設定し、勾配の累積は行わなかった。学習率は、各モデルについて{1e-5, 1e-4, 5e-4, 1e-3}でグリッドサーチを行った。具体的には、BitNet と FP32 モデルには 1e-4、DQT 1.58 ビットには 1e-3、DQT 8 ビットには 5e-4 を使用している。トークナイザについては、公開されている事前学習済みのもの [3] をそのまま使用し、学習中に更新は行わなかった。すべてのモデルは FP32 で混合精度を有効にして学習した。

3） https://huggingface.co/1bitLLM/bitnet_b1_58-large