УДК 004.032.26
DOI: 10.26102/2310-6018/2026.55.4.005

Quantization of outlier free quantizable language models

S.A. Khan[1], A.S.M.H. Kabir[2]✉, R.A. Lukmanov[1]

[1]Innopolis University, Innopolis, Republic of Tatarstan, the Russian Federation
[2]Moscow Institute of Physics and Technology, Moscow, the Russian Federation

Abstract. As deep learning models including the LLMs become a part of our daily lives, they continue to require more and more computational cost. The heavy models need a lot of processing power to train and even to make inferences. However, we can reduce this cost by compression techniques such as quantization. Standard quantization of some transformer models comes at the risk of presence of outliers that result in inaccurate results. In this study, we develop a hybrid model which involves using clipped softmax in attention heads of the model during training to mitigate outliers and then applying activations aware weights only quantization on trained model which helps in reducing quantization error by scaling the weights before quantization. We show that our approach results in better handling of outliers, hinted by reduced kurtosis in clipped softmax trained quantized models as compared to vanilla trained quantized models. Overall, our hybrid method not only achieves the best final model performance but does so by effectively suppressing outliers by a factor of 5–7x across key metrics, making the model far more robust to the quantization process.

Keywords: quantization, outlier, perplexity, attention, softmax, kurtosis.

Acknowledgement: This work was supported by the Academy of Sciences of the Republic of Tatarstan under grant agreement No. 254/2024-PD.

For citation: Khan S.A., Kabir A.S.M.H., Lukmanov R.A. Quantization of outlier free quantizable language models. *Modeling, Optimization and Information Technology.* 2026;14(4). https://moitvivt.ru/ru/journal/article?id=2082 DOI: 10.26102/2310-6018/2026.55.4.005

Квантование языковых моделей без выбросов

С.А. Кхан[1], А.С.М.Х. Кабир[2]✉, Р.А. Лукманов[1]

[1]Университет Иннополис, Иннополис, Республика Татарстан, Российская Федерация
[2]Московский физико-технический институт, Москва, Российская Федерация

Резюме. По мере того, как модели глубокого обучения, включая большие языковые модели (LLM), становятся частью нашей повседневной жизни, они требуют все больших вычислительных ресурсов. Тяжелые модели нуждаются в значительной вычислительной мощности как для обучения, так и для выполнения выводов. Однако эту нагрузку можно снизить с помощью методов сжатия, таких как квантование. Стандартное квантование некоторых моделей трансформеров сопряжено с риском появления выбросов, что приводит к неточным результатам. В данном исследовании разрабатывается гибридная модель, которая включает использование усеченного софтмакса в модулях внимания модели во время обучения для смягчения влияния выбросов, а затем применение квантования только весов с учетом активаций на обученной модели. Это помогает снизить ошибку квантования за счет масштабирования весов перед квантованием. Показано, что предлагаемый подход позволяет лучше справляться с выбросами, о чем свидетельствует уменьшение куртоза у моделей с квантованием, обученных с усеченным софтмаксом, по сравнению с моделями, обученными стандартным способом. В целом, гибридная методика не только обеспечивает наилучшую итоговую производительность модели, но и эффективно подавляет выбросы в 5–7 раз по ключевым метрикам, делая модель значительно более устойчивой к процессу квантования.

Ключевые слова: квантование, выброс, перплексия, внимание, софтмакс, куртозис.

Благодарности: Данная работа была поддержана Академией наук Республики Татарстан в рамках гранта № 254/2024-PD.

Для цитирования: Кхан С.А., Кабир А.С.М.Х., Лукманов Р.А. Квантование языковых моделей без выбросов. *Моделирование, оптимизация и информационные технологии*. 2026;14(4). (На англ.). URL: https://moitvivt.ru/ru/journal/article?id=2082 DOI: 10.26102/2310-6018/2026.55.4.005

Introduction

Language models and other deep learning models have gradually become integral to many aspects of modern life. However, deploying these models comes at a high computational and energy cost, making it challenging to run them efficiently on resource-constrained devices [1]. To address this, certain model compression techniques have been developed that make large models smaller and faster by representing weights and activations with lower precision without significant loss in accuracy. In the course of this paper, we focus on one of such techniques which is quantization. Quantization is used to reduce size of the model by representing its weights and activations with lower numerical precision. It also enables the model to perform inference faster [2]. However, quantizing transformer models like BERT or LLaMA often leads to significant accuracy degradation particularly in attention layers. This is largely due to the presence of outliers - unusually large values especially in intermediate hidden states. When quantizing, these large values force scaling factors to accommodate them, causing the rest of the data (which is small in magnitude) to become indistinguishable under low-precision formats, leading to severe information loss [3]. To solve this problem, it is necessary to quantize the model in a way that it does not have outliers. There already exist some methods such as SmoothQuant [4] and Quantizable Transformers [5] techniques. In this paper, we focus on a hybrid model trained using outlier free training technique introduced under Quantizable Transformers approach and quantized using activations aware quantization. We explore the effectiveness of this hybrid approach evaluate how well it maintains the accuracy while solving the outlier problem in transformer models.

With the advancements in the field of large language models (LLMs) and deep learning architectures based on transformers, the application and integration of such models in several areas have expanded significantly. From natural language processing (NLP) tasks such as machine translation, text summarization, and question answering to broader applications in code generation, autonomous systems, and scientific computing, language models are becoming essential tools in modern AI systems. However, it’s not easy to adopt these models as there is a high cost of computation involved in training and deployment of these model. These computations require large scale clusters of GPUs or TPUs leading to utilization of more energy and compute power [1].

As the model sizes increase and models become more complex, it becomes more important to focus on carbon footprint and efficiency of the models. To cater these problems, we need to look for some techniques that optimize model training and inference while making sure that accuracy of model is not degraded by a significant difference. These techniques reduces the compute and storage costs of deep learning models. One of the notable techniques is compression. Compressions works by reducing computational requirements, memory footprint and the number of parameters, while maintaining the accuracy of the models. Among the most effective methods for compression are quantization, pruning, knowledge distillation and can include neural architecture search as well [6].

Quantization refers to the process of reducing the numerical precision of model weights and sometimes activations, often replacing floating-point representations with lower-bit integer representations. This significantly reduces the memory bandwidth requirements and increases inference speed, making the deployment of models more feasible on resource- constrained

hardware such as edge devices, mobile phones, and embedded systems. There are multiple quantization techniques available as Post Training Quantization (PTQ) and Quantization Aware Training (QAT). In QAT, there are further granularities such as per- channel quantization [7], per-layer quantization [8], per-vector quantization [9] and per-tensor quantization [10]. However, it is not always convenient to quantize transformer-based models, especially in cases where outlier values appear in the activations, as this can lead to reduced accuracy. Different approaches like Quantizable Transformers and smoothquant have been developed to address this issue aiming to reduce the degradation due to outliers presence.

Pruning, another compression technique, involves the systematic removal of less significant parameters in a neural network. It works by reducing connections or neurons, making the model size smaller and maintaining the accuracy. Pruning, if combined with quantization, can make the deep learning models more efficient and eventually more suitable for real world deployment.

Knowledge distillation is just another technique in which a large, complex model transfers its knowledge to a smaller, more efficient model. The large, complex model is usually referred to as the teacher model while the smaller, more efficient model is referred to as student model. The student model can reach similar performance with fewer parameters by using the soft predictions from the teacher model, making it a good option for deployment in environment with limited resources.

In this paper, our focus remains on quantization specifically in the context of transformer-based models such as BERT. Despite the benefits that quantization offers, it remains challenging to use it for transformers because the outliers might be present in activations causing instability in low-bit quantization. This instability eventually leads to lower accuracy. To address this problem of outliers, techniques like Quantizable Transformers, OutEffHop and SmoothQuant have been proposed. These methods either normalize outliers before quantization or make use of smoothing transformations to distribute their influence across layers, making the model less stable when using lower-precision computation. Quantizing transformer models requires careful management of activation outliers, particularly within the Feed-Forward Network (FFN) sublayers. The Quantizable Transformers approach demonstrates that many attention heads (especially in BERT and LLaMA) produce excessively large FFN outputs that severely degrade quantized performance. Applying softmax clipping substantially reduces the outliers.

Dataset. The dataset used is a combination of wiki-40b [11] and bookcorpus [12] dataset. Further details about the dataset can be found in Table 1.

Table 1 – Records for Wiki-40B and BookCorpus datasets
Таблица 1 – Записи для наборов данных Wiki-40B и BookCorpus

Dataset	Training Set	Test Set	Validation Set
Wiki-40B	2,926,536	162,274	163,597
BookCorpus	74,004,228	–	–

For training the model, concatenated training sets of Wiki-40B and BookCorpus were used. And for the evaluation, the Wiki-40B validation set was used

Materials and methods

We conduct an experimental study by training the BERT-base-uncased model using both a standard (vanilla) training approach and Quantizable Transformers' Outlier Free training technique. Once trained, we apply quantization strategies to these models, including uniform quantization and activation-aware quantization [13], a well-known quantization framework

optimized for large language models. The work flow is shown in Figure 1. Our goal is to compare the effectiveness of hybrid approach where we train the model using clipped softmax approach and quantize it with activations aware quantization with other techniques in terms of model accuracy, inference speed, and computational efficiency. The results provide insights into the trade-offs between different quantization methods and their impact on transformer-based architectures.

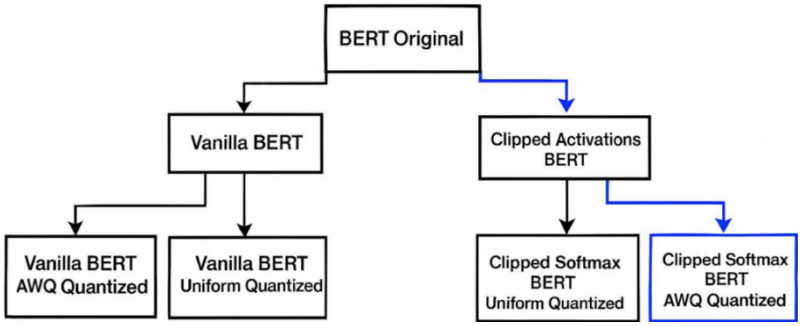


Figure 1 – Training and quantization workflow
Рисунок 1 – Процесс обучения и квантования

Training the model. The BERT-base-uncased model [14] was trained using the Masked Language Model training method with sequences truncated or padded to a maximum length of 128 tokens. The learning rate was 0.0001 and the training was run for 370,000 steps. The parameters of BERT base uncased are mentioned in Table 2.

Table 2 – Parameter counts for BERT base uncased model
Таблица 2 – Количество параметров модели BERT base uncased

Component	Parameter Count
Embeddings	23,837,184
Encoder	85,054,464
Head	24,094,068
Total (pre-training)	132,985,716
Total (encoder)	108,891,648

Vanilla approach. For the vanilla approach, no architectural changes were made.

Clipped softmax apparoch. For clipped softmax training, the softmax function was clipped with gamma (γ) value of −0.025 and while zeta (ζ) value 1.0 as required in the following equation of clipped softmax.

$$\text{clipped\_softmax}(x; \zeta, \gamma) := \text{clip}(\zeta - \gamma) \cdot \text{softmax}(x) + \gamma, 0, 1. \quad (1)$$

We also incorporated additional attention head dropout to encourage quantization robustness. These architectural changes are designed to suppress outlier activations.

Quantizing the model. Both the vanilla trained model and the outlier-free trained model were later on quantized using the post-training quantization techniques.

Uniform quantization. The models were quantized to int-8 using the post-training quantization technique where uniform quantization was performed. For weights, symmetric quantization was performed, defined by:

$$q_w = round\left(\frac{w}{s}\right), \text{ and } s = round\left(\frac{max(|w|)}{2^{b-1}-1}\right), \quad (2)$$

where w – full-precision weight, q_w – quantized integer weight, s – symmetric scale factor, b – bitwidth (e.g., 8 for int-8).

While for activations asymmetric quantization was performed, defined by:

$$q_a = round\left(\frac{a - a_{min}}{s}\right), \text{ and } s = round\left(\frac{a_{max} - a_{min}}{2^b - 1}\right), \quad (3)$$

where a – full-precision activation, q_a – quantized integer activation, a_{min}, a_{max} – observed min and max activation values, s – asymmetric scale factor, b – bitwidth (e.g., 8 for uint-8).

Table 3 mentions some details of the uniform quantization processing.

Table 3 – Timing and processor details for uniform quantization method
Таблица 3 – Время выполнения и характеристики процессора для метода равномерной квантизации

Quantization Method	Dataset Prep Time, minutes	Evaluation Time, minutes	GPU Used
Outlier Free Quantized	74.37	51.13	A100 – 40 GB
Vanilla Quantized	73.65	31.65	A100 – 40 GB

Activation aware quantization. The vanilla trained model and the outlier-free trained model were also quantized to int-8 using the activations-aware quantization technique, but in this approach the model was not fully quantized to 8 bits. The weights were quantized to 8-bits while the activations were kept in 16-bits. Table 4 mentions some details of the activation-aware quantization processing.

Table 4 – Timing and processor details for activation-aware quantization method
Таблица 4 – Время выполнения и характеристики процессора для метода квантования с учетом активации

Quantization Method	Dataset Prep Time	Evaluation Time	GPU Used
Outlier Free Quantized	88.75 minutes	136.25 minutes	T4 – 16 GB
Vanilla Quantized	65.46 minutes	141.33 minutes	T4 – 16 GB

Results and analysis

The models trained using the outlier-free approach seem to perform well, with perplexity around 6. Table 5 shows GPU that was used to train the models, the training time, the resulting perplexity and footprint of the models.

Table 5 – Timing and processor details for AWQ quantization method
Таблица 5 – Время выполнения и характеристики процессора для метода квантования AWQ

Model	GPU Used	Training Time	Perplexity	Model Footprint
Outlier Free Trained Model	A100 – 40 GB	1564.65 minutes	6.03	219,036,788
Vanilla Trained Model	A100 – 40 GB	1505.30 minutes	6.47	219,036,788

To assess the impact of outlier suppression strategies on quantization robustness, we track two diagnostic metrics: Max FFN Output ∞-Norm and Max FFN Input+Output ∞-Norm. The ∞-Norm for a vector x can be defined as:

$$\|x\|_\infty = \max_{i = 1.....n} |x_i|. \quad (4)$$

And eventually Max FFN Output ∞-Norm for evaluation is given by:

$$\max_{i \in \text{evaluation batches}} \|FFN_{out}^{(i)}\|_\infty. \quad (5)$$

And Max FFN Input+Output ∞-Norm is given by:

$$\max_{i \in \text{evaluation batches}} ||FFN^{i}_{out} + \text{residual}^{(i)}||_{\infty}. \quad (6)$$

These metrics capture the largest activation magnitudes in the feed-forward network (FFN) output and the residual + FFN combination, respectively. High values in these norms signal the presence of outliers, that can introduce a significant quantization error.

Similarly, we track the kurtosis and provide maximum and average kurtosis offering insights into how tail-heavy or spiky the activation distributions are. The kurtosis for a given layer of a batch can be defined as:

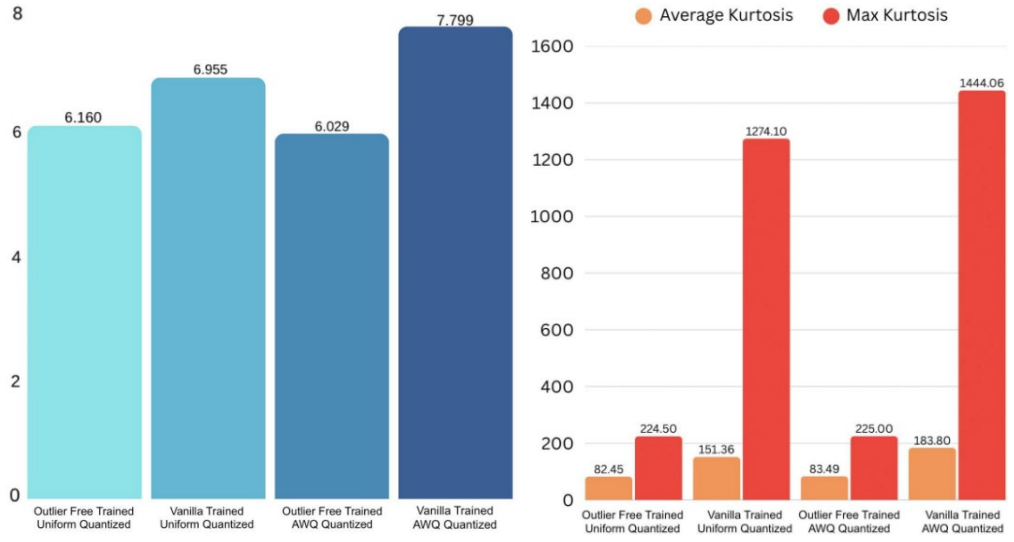
$$\text{Kurtosis}(x) = \frac{E[(x-\mu)^4]}{\alpha^4}, \quad (7)$$

where $\mu = E[x]$ is the mean, $\alpha^2 = E[(x-\mu)^2]$ is the variance.

In the outlier-free trained models, these values are much lower as compared to vanilla trained models indicating outlier suppression. Also, the perplexity highly improved when the outlier-free model was fully quantized to int-8 using the uniform quantization technique. At the same time, the Kurtosis for outlier-free quantized model is quite low indicating better handling of extreme outliers which was well handled by our hybrid approach.

The quantization using the activations-aware technique resulted in worse perplexity of the outlier-free quantized model while slightly improving the perplexity of the vanilla trained quantized model.

A comparison of results in Figure 2 clearly shows that the hybrid approach of training a model with clipped softmax and original attention head, and then quantizing using the activations aware weights quantization leads to a better perplexity then the other approaches defined. We also see a significant decrease in max kurtosis signifying reduction in outliers. We do see a higher Max kurtosis in Vanilla BERT quantized by AWQ, and Vanilla BERT quantized by uniform quantization technique indicating presence of outliers. Figure 3, 4 and 5 show the perplexity, max kurtosis and average kutosis over validation steps respectively. Table 6 and 7 presents the evaluation statistics for uniform quantized models and AWQ quantized models.



a) *b*)

Figure 2 – Quantization metrics comparison between AWQ and Uniform quantization: *a* – perplexity comparison; *b* – average kurtosis

Рисунок 2 – Сравнение метрик квантования между методами AWQ и равномерного квантования: *а* – сравнение перплексии; *б* – средний эксцесс

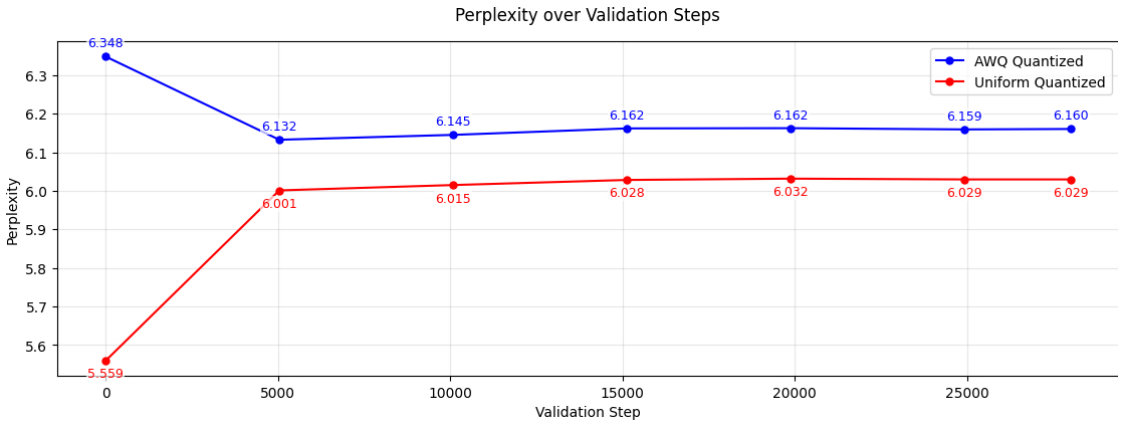


Figure 3 – Perplexity comparison between AWQ and uniform quantization models
Рисунок 3 – Сравнение перплексии между моделями с квантованием AWQ и равномерным квантованием

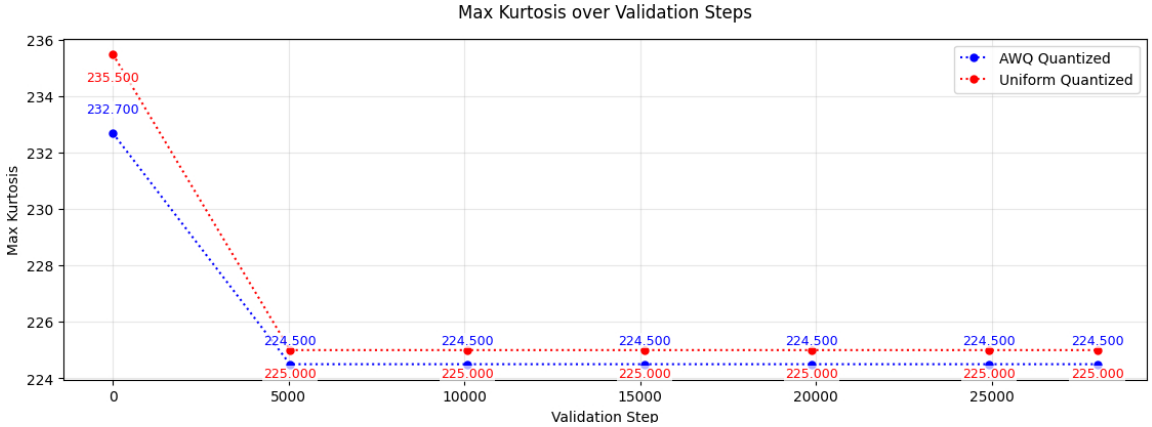


Figure 4 – Maximum kurtosis values during model validation
Рисунок 4 – Максимальные значения куртоза при валидации модели

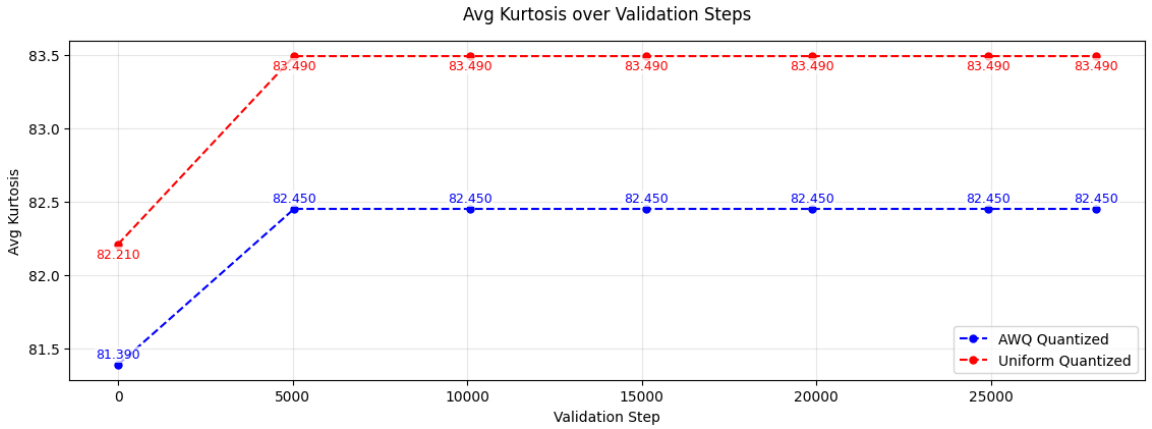


Figure 5 – Average kurtosis measurements, indicating the overall distribution characteristics of the quantized models
Рисунок 5 – Средние значения куртоза, показывающие общие характеристики распределения квантованных моделей

Table 6 – Evaluation statistics for uniform quantized models
Таблица 6 – Статистика оценки моделей с равномерным квантованием

Quantized Model – Uniform Quantization	Perplexity	Avg. Kurtosis	Max. Kurtosis	Max FFN Output Inf Norm	Max FFN Input + Output Inf Norm	Max LN (FFN i + o) Inf Norm
Outlier Free Model	6.160	82.45	224.5	26.3	48.9	18.6
Vanilla Model	6.955	151.36	1274.1	173.9	186.7	23.6

Table 7 – Evaluation statistics for AWQ quantized models
Таблица 7 – Статистика оценки моделей с квантованием AWQ

Quantized Model – AWQ	Perplexity	Avg. Kurtosis	Max. Kurtosis	Max FFN Output Inf Norm	Max FFN Input + Output Inf Norm	Max LN (FFN i + o) Inf Norm
Outlier Free Model	6.029	83.49	225.0	26.3	49.0	18.7
Vanilla Model	7.799	183.8	1444.06	181.178	194.165	23.56

Conclusion

In this work, we developed a strategy comprising of two steps to counter the challenges associated with the quantization of transformer-based language models. We found that handling the outliers in the activation during the model training is critical for performing quantization after the model is trained. Using a clipped softmax function regularized the model leading to noticeable reduction in outliers. This is shown by lower kurtosis and infinity norms.

Other than that, we also showed that Activation-Aware Weight Quantization (AWQ) works better than uniform quantization for the outlier free model, achieving a lower perplexity of 6.029. This shows that the two steps work well together: training the model to reduce outliers makes it easier for AWQ to keep important weights, which helps maintain performance. Overall, our results give a clear and practical guideline for researchers and engineers, suggesting that preparing models for quantization during training is more effective than applying quantization techniques only after training.

REFERENCES / СПИСОК ИСТОЧНИКОВ

1. Li P., Yang J., Islam M.A., Ren Sh. *Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models.* arXiv. URL: https://arxiv.org/abs/2304.03271 [Accessed 18th August 2025].
2. Gholami A., Kim S., Dong Zh., et al. *A Survey of Quantization Methods for Efficient Neural Network Inference.* arXiv. URL: https://arxiv.org/abs/2103.13630 [Accessed 18th August 2025].
3. Dettmers T., Lewis M., Belkada Y., Zettlemoyer L. *LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale.* arXiv. URL: https://arxiv.org/abs/2208.07339 [Accessed 18th August 2025].
4. Xiao G., Lin J., Seznec M., et al. *SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models*. arXiv. URL: https://arxiv.org/abs/2211.10438 [Accessed 18th August 2025].
5. Bondarenko Y., Nagel M., Blankevoort T. *Quantizable Transformers: Removing Outliers by Helping Attention Heads Do Nothing.* arXiv. URL: https://arxiv.org/abs/2306.12929 [Accessed 18th August 2025].

6. Khan S.A., Shulepina S., Shulepin D., Lukmanov R.A. Review of Algorithmic Solutions for Deployment of Neural Networks on Lite Devices. *Computer Research and Modeling.* 2024;16(7):1601–1619. https://doi.org/10.20537/2076-7633-2024-16-7-1601-1619
Кхан С.А., Шулепина С., Шулепин Д., Лукманов Р.А. Обзор алгоритмических решений для развертывания нейронных сетей на легких устройствах. *Компьютерные исследования и моделирование.* 2024;16(7):1601–1619. (На англ.). https://doi.org/10.20537/2076-7633-2024-16-7-1601-1619
7. Krishnamoorthi R. *Quantizing Deep Convolutional Networks for Efficient Inference: A Whitepaper.* arXiv. URL: https://arxiv.org/abs/1806.08342 [Accessed 24th August 2025].
8. Dumitru R.-G., Yadav V., Maheshwary R., et al. *Layer-wise Quantization: A Pragmatic and Effective Method for Quantizing LLMs Beyond Integer Bit-levels.* arXiv. URL: https://arxiv.org/abs/2406.17415 [Accessed 24th August 2025].
9. Dai S., Venkatesan R., Ren H., et al. *VS-Quant: Per-Vector Scaled Quantization for Accurate Low-Precision Neural Network Inference.* arXiv. URL: https://arxiv.org/abs/2102.04503 [Accessed 24th August 2025].
10. Nagel M., van Baalen M., Blankevoort T., Welling M. *Data-Free Quantization through Weight Equalization and Bias Correction.* arXiv. URL: https://arxiv.org/abs/1906.04721 [Accessed 28th August 2025].
11. Guo M., Dai Z., Vrandečić D., Al-Rfou R. Wiki-40B: Multilingual Language Model Dataset. In: *Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, 11–16 May 2020, Marseille, France.* European Language Resources Association; 2020. P. 2440–2452.
12. Zhu Y., Kiros R., Zemel R., et al. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 07–13 December 2015, Santiago, Chile.* IEEE; 2015. P. 19–27. https://doi.org/10.1109/ICCV.2015.11
13. Lin J., Tang J., Tang H., et al. *AWQ: Activation-Aware Weight Quantization for LLM Compression and Acceleration*. arXiv. URL: https://arxiv.org/abs/2306.00978 [Accessed 24th August 2025].
14. Devlin J., Chang M.-W., Lee K., Toutanova K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv. URL: http://arxiv.org/abs/1810.04805 [Accessed 24th August 2025].

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Кхан Самеед Ахмед, аспирант, факультет компьютерных наук и инженерии, Университет Иннополис, Иннополис, Республика Татарстан, Российская Федерация.
e-mail: sameedkhandurrani@gmail.com

Sameed Ahmed Khan, Postgraduate, Faculty of Computer Science and Engineering, Innopolis Univeristy, Innopolis, Republic of Tatarstan, the Russian Federation.

Кабир А. С. М. Хумаюн, аспирант, кафедра интеллектуальных информационных систем и технологий, Московский физико-технический институт, Москва, Российская Федерация.
e-mail: humaun.kabir@phystech.edu

A. S. M. Humaun Kabir, Postgraduate, Department of Intelligent Information Systems and Technologies, Moscow Institute of Physics and Technology, Moscow, the Russian Federation.

Лукманов Рустам Абубакирович, доцент, Университет Иннополис, Республика Татарстан, Российская Федерация.
e-mail: r.lukmanov@innopolis.ru

Rustam A. Lukmanov, Associate Professor, Innopolis Univeristy, Innopolis, Republic of Tatarstan, the Russian Federation.

Статья поступила в редакцию 09.02.2026; одобрена после рецензирования 18.03.2026; принята к публикации 10.04.2026.

The article was submitted 09.02.2026; approved after reviewing 18.03.2026; принята к accepted for publication 10.04.2026.