Challenges and Development of Large Language Models Applied to Intelligent Hardware

Emory Callahan * and Qi Chen

Posted Date: 18 November 2025

doi: 10.20944/preprints202511.1217.v1

Keywords: ChatGPT; LLM; model compression; model quantization; AI chips

 doi:10.20944/preprints202511.1217.v1

Article

Challenges and Development of Large Language Models Applied to Intelligent Hardware

Emory Callahan * and Qi Chen

Department of Computer Science, University of Winnipeg, 515 Portage Avenue, Winnipeg, Manitoba R3B 2E9, Canada
* Correspondence: callahanemory879@gmail.com

Abstract

With the rapid development of artificial intelligence and the increasing popularity of intelligent learning devices, the application of large models in intelligent learning devices is becoming increasingly important. Since this year, large language models represented by ChatGPT have emerged. These models are trained on massive online knowledge bases and possess multi-disciplinary knowledge comprehension capabilities. They can interact through Q&A, generate writing guidance, practice spoken language, and more. Currently, large language models are mainly accessed through API calls to cloud services, which require significant computational resources. This paper discusses the deployment of large language models onto personal intelligent learning devices.

Keywords: ChatGPT; LLM; model compression; model quantization; AI chips

1. Introduction

Large language models possess four key capabilities: understanding, generation, reasoning, and memory. They can handle complex tasks and large-scale data, bringing significant advancements to general-purpose artificial intelligence. Applying large models to intelligent learning devices can endow these devices with high levels of intelligence and support broader application scenarios.

Intelligent learning devices can interact with large models to enable functions such as more accurate English translation, text generation, and open-ended Q&A, offering users improved experiences and convenience. However, the application of large models in intelligent learning devices also poses several challenges, including computing resource consumption, model size, and energy efficiency issues.

Therefore, optimization and innovation are required in both hardware and software to enable efficient large model applications. On the software side, model compression techniques can be used to reduce model parameters and computation while ensuring that precision-optimized algorithms produce acceptable results as much as possible.

On the hardware side, we adopt architectures and algorithms specifically designed for large model applications. These can quickly process and execute AI-related tasks with high parallel computing capabilities and improved memory access speeds. Combined with large model compression techniques and precise computation, this helps reduce data storage and transmission demands, thereby lowering computational complexity and overhead.

2. Related Work

The deployment of large language models (LLMs) in intelligent hardware has gained increasing attention in recent years, particularly in the context of model compression and efficient fine-tuning. Research efforts [1] have proposed modular task decomposition frameworks that leverage LLMs to dynamically coordinate multi-agent systems, highlighting adaptability in real-time environments. Other studies [2] have introduced structural priors and modular adapters to enhance the composability of parameter-efficient fine-tuning, demonstrating effective reuse of model components. Approaches

 doi:10.20944/preprints202511.1217.v1

based on federated distillation [3] emphasize robustness and communication efficiency for distributed fine-tuning, while works in structural regularization [4] address bias mitigation during adaptation. Meanwhile, dynamic structured gating techniques [5] facilitate efficient parameter allocation across heterogeneous tasks, supporting deployment in resource-constrained systems.

From the perspective of data privacy and security, privacy-preserving low-rank tuning methods [6] adopt differential privacy mechanisms to safeguard user information during instruction tuning. Task-aware differential privacy with structural perturbation [7] further improves privacy protection without significantly affecting adaptation performance. Meanwhile, selective semantic masking strategies [8] enable privacy-oriented text generation by restricting the exposure of sensitive components during fine-tuning.

Research on retrieval-augmented generation (RAG) and alignment strategies has also grown rapidly. Two-stage retrieval and cross-segment alignment pipelines [9] have been developed to improve contextual consistency and relevance. In domain-specific generation, context compression combined with structural text representations [10] has shown improved fluency and factual accuracy. Literature-oriented RAG techniques [11] explore efficient knowledge grounding under retrieval constraints, whereas fusion-based RAG architectures [12] enhance the capability of LLMs in complex question answering.

In parallel, modular and structural adaptation has been widely adopted to optimize scalable deployment. Structure-learnable adapter designs [13] provide interpretable and modular mechanisms aligned with mainstream parameter-efficient tuning. Additionally, selective knowledge injection frameworks [14] demonstrate how adapter modules can facilitate context-aware domain adaptation during fine-tuning.

Finally, LLM-enhanced architectures are being increasingly adopted in domain-driven intelligent applications. Multi-scale feature fusion with graph-based representations [15] benefits LLM-assisted text classification by improving global semantic reasoning. Attention-based LLM modeling has been applied to systemic financial risk forecasting [16], clinical risk identification from heterogeneous medical data [17], and enterprise anomaly detection in ETL workflows [18], further demonstrating the practical value of LLM-enabled intelligent systems.

Overall, these studies collectively highlight a growing convergence of structural efficiency, privacy preservation, and domain adaptability in intelligent hardware deployments, reinforcing the potential of LLM-driven solutions in real-world resource-limited environments such as intelligent educational devices.

3. Model Compression and Precision Optimization

3.1. Compression Methods

Large model compression techniques aim to reduce model parameters and computation while maintaining the highest possible performance. Common methods include removing redundant parameters and operations, weight sharing, operator fusion, and quantization to fixed-point numbers.

Removing redundant parameters and operations: This method reduces model size by eliminating unnecessary parameters and connections. It can be implemented using importance-based or sensitivity-based pruning strategies, such as L1 regularization, Taylor expansion, and sensitivity analysis [19].

Operator fusion: This involves combining multiple scattered operations into one to reduce redundant computation. Common examples include convolution plus activation fusion, matrix multiplication plus addition into GEMM, LayerNorm fusion, and Gelu fusion.

Weight sharing: Similar weight parameters are shared to reduce memory requirements. Common approaches include convolution kernel sharing and matrix decomposition.

Quantization: Floating-point parameters are converted into lower-precision fixed-point or binary numbers to reduce memory demand and computational complexity. Typical techniques include symmetric and asymmetric quantization.

 doi:10.20944/preprints202511.1217.v1

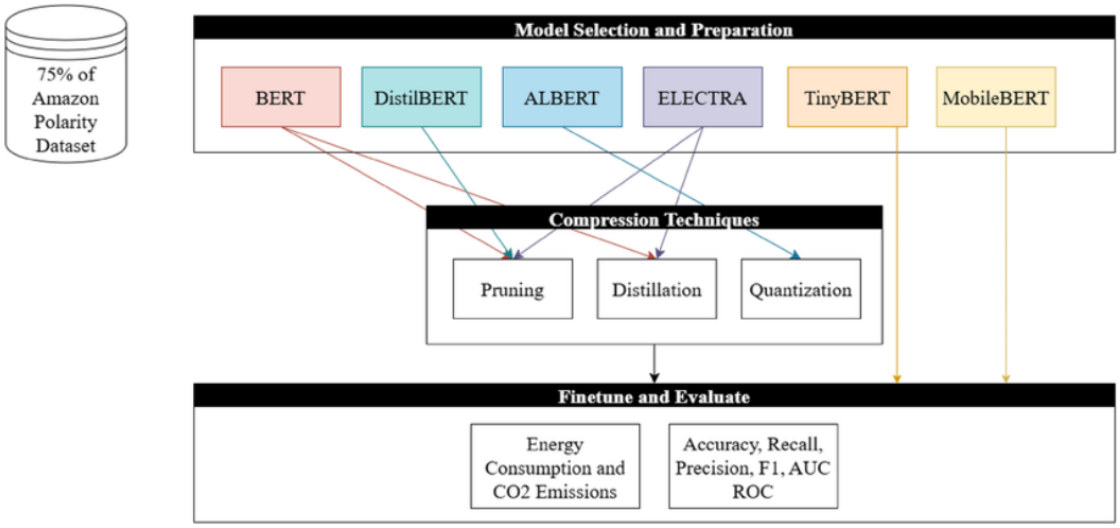

Figure 1. Compression Technique Workflow Diagram.

3.2. Precision Optimization

Model compression can significantly reduce computational demand, but it may also lead to accuracy loss, causing user responses to deviate from expectations. Therefore, it is necessary to incorporate precision optimization techniques during model compression.

3.2.1. Mixed-Precision Quantization

High-bit quantization ensures precision in both memory and computation, while low-bit quantization offers higher efficiency in terms of memory and computation. Since fixed-bit quantization cannot fully balance precision and computation, mixed-precision quantization is needed to further optimize model compression efficiency.

The main idea behind mixed-precision quantization is to distinguish parameters by their sensitivity to precision loss. Generally, some parameters in the model require higher accuracy and must be kept in high-precision, while others can tolerate lower precision and thus be quantized with lower bit widths.

The steps involved in mixed-precision quantization are as follows:

(1) Sensitivity analysis: Divide the model's operators into different groups, where each group corresponds to a quantization bit width [20]. Common analysis methods include: Hessian spectrum, MSE loss, and residual distance.

(2) High-bit precision: Parameters with high precision requirements should retain their original precision and not be quantized [20].

(3) Low-bit precision: Parameters with lower precision requirements are quantized using low-bit widths. Common choices include 2-bit, 4-bit, 8-bit, and 16-bit widths [20].

3.2.2. Quantization Parameter Tuning

Quantization parameter tuning is a fine-tuning technique for quantized parameters. Through repeated testing and evaluation, it fully considers the interaction between model characteristics and quantization strategies. By selecting appropriate quantization methods and parameters, tuning helps optimize model performance while reducing accuracy loss caused by quantization.

The tuning process generally follows these steps:

Understand model characteristics: First, understand the characteristics and limitations of the target device, such as computing power, memory size, and I/O bandwidth. These factors affect inference speed and memory usage, and must be considered during tuning [21].

Choose appropriate quantization methods and parameters: Based on the device characteristics and application scenarios, select appropriate quantization strategies. For instance, group quantization

 doi:10.20944/preprints202511.1217.v1

may be more effective for models with clear sub-structure. Different operators may also require different methods (e.g., Min-Max, KL divergence) [22].

Optimization: During tuning, use optimization algorithms to find the best parameter set. For example, gradient descent can be used to minimize tuning errors [23].

Evaluate performance: After finding the best parameter set, evaluate model performance to verify the tuning result. If performance degradation or other issues are detected, re-tune the parameters to find a better configuration [24-25].

4. AI Chip Acceleration for Large Model Inference

4.1. The Importance of AI Chips in Large Model Inference

Improving inference speed: Large model inference typically requires a large number of matrix operations and complex calculations. AI chips are equipped with high-precision parallel computing capabilities, enabling acceleration of these processes. By leveraging the parallel computing power of AI chips, inference speeds can be significantly enhanced, allowing applications to respond quickly and process large-scale data.

Reducing energy consumption: Large model inference usually consumes significant energy. AI chips can reduce energy use by utilizing specialized hardware design and optimization, achieving higher computational efficiency and energy savings. Compared with traditional general-purpose computing platforms, AI chips can perform the same tasks with lower energy consumption, helping to minimize the energy demand of large-scale model deployments.

Improving data privacy: Currently, large models are often deployed on cloud servers, where personal and sensitive user data must be transmitted to the cloud for inference. This poses potential privacy risks. AI chips enable local inference directly on edge devices, reducing data transmission and exposure, thereby improving the safety and reliability of AI applications.

4.2. Challenges of AI Chip-Accelerated Inference

Hardware design: Efficient large model inference requires AI chips with high computing power, memory capacity, and energy efficiency. Meeting these demands presents many challenges, including improving computing throughput, optimizing memory access and data transfer, and reducing latency. These engineering challenges demand deep research and innovation in chip architecture and hardware optimization.

Software support: In addition to hardware, efficient large model inference also relies on software support, including model compilation, deployment tuning, and runtime environments. These software components must fully exploit AI chip hardware features and the computational graph structure of large models to deliver optimized inference performance. Therefore, software engineering must support the design and tuning of large models to ensure both speed and efficiency.

4.3. Outlook for AI Chip-Accelerated Inference

AI chip development: With the rapid growth of AI technology, AI chip design and fabrication are continually advancing. Future AI chips will be more powerful and capable of meeting increasingly complex large model inference requirements. At the same time, energy efficiency will continue to improve, making inference more feasible and sustainable.

Adaptive inference: Future large model inference will become more intelligent and adaptive. AI chips will be able to adapt to various application scenarios by automatically adjusting computing and memory strategies to deliver optimal performance and energy efficiency. This will allow inference to become more flexible and efficient in diverse application contexts.

Cross-device collaboration: Large model inference will also promote collaboration among devices. By distributing large models across multiple AI chips, different devices can cooperatively execute computations and inference tasks, achieving distributed inference. This is especially useful in applica-

 doi:10.20944/preprints202511.1217.v1

tion scenarios that require high response speed and multi-device coordination, such as autonomous driving systems, where strong computing power and performance are essential.

5. Prospects for Large Model Applications in Intelligent Learning Devices

Large models have great potential for applications in translation pens, learning machines, and other intelligent educational devices. The following are several possible application scenarios:

(1) Real-time translation: Large models can be applied to translation pens and similar devices to implement real-time translation functions. By leveraging large language datasets and language models, large models can better understand and translate content between languages. Users can input the required text or speech into the device, and the large model can quickly translate it into the target language for display on the screen or output via speech.

(2) Learning assistance: Large models can provide strong academic support in learning devices. With access to a vast amount of learning materials and educational resources, large models can help students answer questions, offer explanations and solutions, and even provide personalized learning suggestions based on individual learning conditions. Such applications can improve learning efficiency and performance.

(3) Intelligent Q&A and knowledge retrieval: Large models can enable smart Q&A and knowledge retrieval functions in intelligent devices. Users can ask questions, and the large model can leverage its pre-trained knowledge base to provide accurate and detailed answers. This is widely applicable in fields such as science, medicine, and law, helping users quickly obtain the information they need.

(4) Speech recognition and interaction: Large models can help intelligent devices achieve precise speech recognition and interactive functions. Users can interact with the device via voice input, and the large model can understand commands and questions to generate appropriate responses. This greatly improves usability and user experience.

(5) Content generation and text creation: Large models can enable intelligent devices to generate content and compose text. Users can provide key prompts via the device, and the large model can generate corresponding articles, essays, or reports. This is applicable in areas such as writing, creative work, and content production.

The application prospects of large models in translation pens, learning machines, and other intelligent educational devices are broad. Most of their functions revolve around language models, knowledge Q&A, logical reasoning, and other capabilities, which naturally align with educational scenarios. Moreover, large models and related AI technologies have always been considered crucial foundations for the advancement of AI in education.

6. Conclusions

The combination of large model compression techniques and AI chips can significantly accelerate model inference, making it feasible to deploy large models in translation pens, learning machines, and other educational devices. The application of large models in intelligent learning devices carries immense potential and challenges. Embedding large models in smart hardware helps break away from the limitations of homogeneous competition and builds a new competitive edge.

Learning machines, reading pens, and desk lamps are all carriers of educational content services. While the competitive core remains unchanged, the deployment of large models has shifted from software-driven cloud services to hardware-based local services, further narrowing the gap between content delivery and computing devices [1].

It is believed that in the near future, an AI tutor proficient in all subjects will enter ordinary households and assist children in learning and growing.

References

1. Pan, S. and Wu, D., “Modular Task Decomposition and Dynamic Collaboration in Multi-Agent Systems Driven by Large Language Models,” arXiv preprint arXiv:2511.01149, 2025.

2. Wang, Y., Wu, D., Liu, F., Qiu, Z. and Hu, C., "Structural Priors and Modular Adapters in the Composable Fine-Tuning Algorithm of Large-Scale Models," arXiv preprint arXiv:2511.03981, 2025.
3. Zou, Y., "Federated Distillation with Structural Perturbation for Robust Fine-Tuning of LLMs," Journal of Computer Technology and Software, vol. 3, no. 4, 2024.
4. Liu, H., "Structural Regularization and Bias Mitigation in Low-Rank Fine-Tuning of LLMs," Transactions on Computational and Scientific Methods, vol. 3, no. 2, 2023.
5. Xue, Z., "Dynamic Structured Gating for Parameter-Efficient Alignment of Large Pretrained Models," Transactions on Computational and Scientific Methods, vol. 4, no. 3, 2024.
6. Yao, G., "Privacy-Preserving Low-Rank Instruction Tuning for Large Language Models via DP-LoRA," Journal of Computer Technology and Software, vol. 3, no. 5, 2024.
7. Li, Y., "Task-Aware Differential Privacy and Modular Structural Perturbation for Secure Fine-Tuning of Large Language Models," Transactions on Computational and Scientific Methods, vol. 4, no. 7, 2024.
8. Zhang, R., "Privacy-Oriented Text Generation in LLMs via Selective Fine-Tuning and Semantic Attention Masks," Journal of Computer Technology and Software, vol. 4, no. 8, 2025.
9. Wang, S., "Two-Stage Retrieval and Cross-Segment Alignment for LLM Retrieval-Augmented Generation," Transactions on Computational and Scientific Methods, vol. 4, no. 2, 2024.
10. Xue, P. and Yi, Y., "Integrating Context Compression and Structural Representation in Large Language Models for Financial Text Generation," Journal of Computer Technology and Software, vol. 4, no. 9, 2025.
11. Zheng, J., Chen, Y., Zhou, Z., Peng, C., Deng, H. and Yin, S., "Information-Constrained Retrieval for Scientific Literature via Large Language Model Agents," 2025.
12. Sun, Y., Zhang, R., Meng, R., Lian, L., Wang, H. and Quan, X., "Fusion-based retrieval-augmented generation for complex question answering with LLMs," 2025 8th International Conference on Computer Information Science and Application Technology (CISAT), pp. 116-120, July 2025.
13. Gong, M., Deng, Y., Qi, N., Zou, Y., Xue, Z. and Zi, Y., "Structure-learnable adapter fine-tuning for parameter-efficient large language models," IET Conference Proceedings CP944, vol. 2025, no. 29, pp. 225-230, August 2025.
14. Zheng, H., Zhu, L., Cui, W., Pan, R., Yan, X. and Xing, Y., "Selective Knowledge Injection via Adapter Modules in Large-Scale Language Models," 2025.
15. Song, X., Huang, Y., Guo, J., Liu, Y. and Luan, Y., "Multi-Scale Feature Fusion and Graph Neural Network Integration for Text Classification with Large Language Models," arXiv preprint arXiv:2511.05752, 2025.
16. Xu, Q. R., Xu, W., Su, X., Ma, K., Sun, W. and Qin, Y., "Enhancing Systemic Risk Forecasting with Deep Attention Models in Financial Time Series," 2025.
17. Xie, A. and Chang, W. C., "Deep Learning Approach for Clinical Risk Identification Using Transformer Modeling of Heterogeneous EHR Data," arXiv preprint arXiv:2511.04158, 2025.
18. Chen, X., Gadgil, S. U., Gao, K., Hu, Y. and Nie, C., "Deep Learning Approach to Anomaly Detection in Enterprise ETL Processes with Autoencoders," arXiv preprint arXiv:2511.00462, 2025.
19. Markus Nagel, Marios Fournarakis, Rana Ali Amjad et al., "A White Paper on Neural Network Quantization," [Online]. Available: EB/OL, 15 Jun. 2021. Accessed: 2 Sep. 2023.
20. Zhen Dong, Zhewei Yao, Amir Gholami et al., "HAWQ: Hessian Aware Quantization of Neural Networks with Mixed-Precision," [Online]. Available: EB/OL, 29 Apr. 2019. Accessed: 2 Sep. 2023.
21. Zhen Dong, Zhewei Yao, Daiyaan Arfeen et al., "HAWQ-V2: Hessian Aware Trace-Weighted Quantization of Neural Networks," [Online]. Available: EB/OL, 10 Nov. 2019. Accessed: 2 Sep. 2023.
22. Zhewei Yao, Zhen Dong, Zhangcheng Zheng et al., "HAWQ-V3: Dyadic Neural Network Quantization," [Online]. Available: EB/OL, 23 Jun. 2021. Accessed: 2 Sep. 2023.
23. Itay Hubara, Yury Nahshan, Yair Hanani et al., "Improving Post-Training Neural Quantization: Layer-wise Calibration and Integer Programming," [Online]. Available: EB/OL, 14 Dec. 2020. Accessed: 2 Sep. 2023.
24. Yue Lu, "AI Education Large Model Landing Dictionary," Consumption Daily, 17 Aug. 2023, pp. 1-2.
25. Tim Dettmers, Artidoro Pagnoni, Ari Holtzman et al., "QLoRA: Efficient Finetuning of Quantized LLMs," [Online]. Available: EB/OL, 23 May 2023. Accessed: 2 Sep. 2023.