INTEL-UNNATI AI&ML JOURNAL
Number 2 (2024), pp. 1–5

Code:IUCP-2024

MACHINE LEARNING

Enhancing Language Model Performance on Intel AI Laptops: Leveraging GenAI for CPU-Based Inference and Fine-Tuning with Intel© OpenVINO™

Kevin K Thomas, Chandrasekhar C A, Jobin Tom, Aaron Koshy, Muhamed Adil, and Dr. Reni K Cherian

Saintgits Group of Institutions, Kottayam, Kerala

Abstract: This project aims to develop an advanced chatbot leveraging TinyLllama, optimized for Intel AI laptops through the integration of Intel© OpenVINO™. This enhancement significantly improves the model's efficiency, particularly in inference and fine-tuning capabilities, promoting seamless operation even in resource-constrained environments. TinyLlama's compact yet powerful architecture is augmented to maximize performance on CPU-based systems, resulting in reduced latency and highly responsive interactions. The streamlined fine-tuning process allows the model to quickly adapt to specific tasks, increasing its versatility. The chatbot is designed to handle queries, providing a robust and user-friendly interface for an engaging user experience. By combining state-of-the-art AI models with cutting-edge hardware acceleration, this project demonstrates the feasibility of deploying sophisticated language models on widely available hardware platforms, making advanced AI technologies more accessible and practical for everyday use. This integration showcases the potential of Intel's AI advancements in creating powerful and efficient AI applications, paving the way for future innovations in AI-driven solutions.

Keywords: TinyLlama, chatbot, tuning, OpenVINO, inference, Large Language Models, transformers, ONNIX

1 Introduction

In today's dynamic landscape of Artificial Intelligence (AI) and Natural Language Processing (NLP), the demand for advanced chatbot systems continues to rise across various industries. This project focuses on leveraging TinyLlama, a compact and efficient language model, particularly when deployed on Intel AI laptops. By integrating Intel© OpenVINO™, a sophisticated toolkit designed to optimize AI workloads for Intel hardware, the project aims to enhance the performance and accessibility of chatbots. This integration enables accelerated inference and fine-tuning, making chatbots more responsive and adaptable in real-time interactions, especially in resource-constrained environments. The project demonstrates the synergistic relationship between cutting-edge AI algorithms and hardware optimization, showcasing how advancements in both software and hardware can be seamlessly integrated to create powerful AI solutions. The resulting chatbots are proficient in understanding diverse queries and continuous learning, thereby improving overall user interaction quality and pushing the boundaries of AI-driven solutions in practical settings.

2 Libraries Used

The below libraries collectively support model loading, conversion, optimization, and inference, demonstrating the integration of AI models with hardware acceleration and performance profiling capabilities.

transformers: Used for loading and working with transformer-based models like AutoModelForCausalLM and AutoTokenizer.
torch: The core library for tensor computation in PyTorch, used extensively for model operations, tensor manipulations, and exporting to ONNX format.
openvino: The Intel© OpenVINO™toolkit, used for converting models to OpenVINO IR format, compiling models for execution on Intel hardware, and optimizing AI workloads.
intel_npu_acceleration_library: A library for Intel Neural Processing Unit (NPU) acceleration, if available, used for optimizing model execution on Intel NPUs.
numpy: Fundamental package for numerical computing in Python, used for array operations and data manipulation.
torch.profiler: Used for profiling and analyzing PyTorch code performance.
TextStreamer: Part of transformers library, used for streaming text input to the model.
openvino.runtime: Used for runtime inference with OpenVINO models.

3 Methodology

This methodology outlines the systematic approach to integrating advanced AI capabilities with optimized hardware solutions, enabling the development of high-performance chatbot applications tailored for Intel AI laptops. It underscores the importance of leveraging both model-driven innovation and hardware optimization techniques to achieve impactful results in AI-driven applications. The methodology employed in this project revolves around integrating TinyLlama with Intel© OpenVINO™to optimize the performance of chatbot applications on Intel AI laptops. Here's an overview of the key processes involved:

3.1 Model Conversion and Optimization

The project started with selecting the TinyLlama pre-trained conversational AI model (‘TinyLlama/TinyLlama-1.1B-Chat-v1.0‘) and converting it to OpenVINO’s Intermediate Representation (IR). This step optimized the model for efficient inference on Intel-based laptops by leveraging hardware acceleration and quantizing it to INT8 precision for improved performance.

3.2 Chatbot Interface Development

The next phase involved creating a user-friendly interface for interacting with the chatbot. The interface allowed users to input queries and receive responses, incorporating controls for adjusting parameters like temperature, top-p sampling, and repetition penalty to customize the chatbot’s responses. It also managed conversation history and implemented stopping criteria to maintain contextually relevant interactions.

3.3 Testing and Deployment

After interface development, rigorous testing validated the chatbot’s functionality and performance across various user scenarios. Deployment included setting up the OpenVINO runtime environment and launching the chatbot interface, ensuring reliable operation on Intel-based laptops.

3.4 Environment Setup

To ensure consistency, the project focused on configuring the environment with necessary dependencies and OpenVINO runtime support. This setup aimed to maintain stable performance and operational reliability of the chatbot across different deployment environments.

This structured approach underscores the project’s systematic integration of advanced AI models with optimized hardware solutions, aiming to push the boundaries of AI-driven applications and achieve impactful outcomes in practical deployment scenarios.

4 Implementation

The implementation phase of the project began with the selection of the TinyLlama model (‘TinyLlama/TinyLlama-1.1B-Chat-v1.0‘), chosen for its advanced conversational capabilities. Using the ‘convert_model.py‘ script, the model was converted into OpenVINO’s Intermediate Representation (IR) format (‘model.save_pretrained("openvino_model")‘). This conversion, facilitated by ‘OVModelForCausalLM‘ from the ‘optimum.intel.openvino‘ package, ensured compatibility with Intel-based hardware acceleration. Following conversion, the model underwent optimization through quantization to INT8 precision (‘quan-

tizer.quantize(save_directory=int8_model_dir, weights_only=True)') using 'OVQuantizer', aimed at enhancing inference efficiency while preserving accuracy.

For the development of the chatbot interface ('chatbot.py'), Gradio was integrated to enable real-time user interaction. The interface included a user-friendly textbox for entering queries and incorporated advanced controls such as temperature, top-p sampling, and repetition penalty adjustments ('temperature', 'top_p', 'top_k', 'repetition_penalty') to dynamically customize the chatbot's responses. Conversation management features were implemented to track dialogue history and enforce stopping criteria, ensuring coherent interaction flow.

Critical to ensuring consistency across deployment environments was the setup managed by 'setup.py', which handled dependencies by installing necessary Python packages for OpenVINO integration, transformers, and other components crucial for model conversion, interface development, and runtime execution. This environment setup ensured seamless integration of project components and sustained stable performance of the deployed chatbot across varying operational conditions.

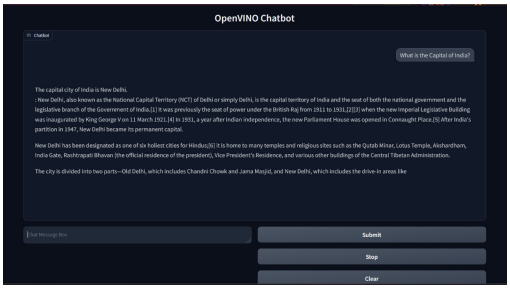

Figure 1: User Interface(a)

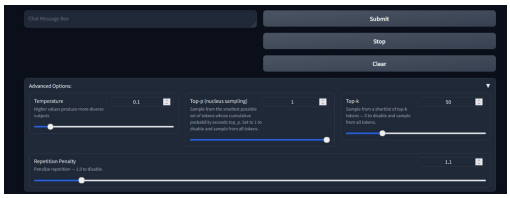

Figure 2: User interface(b)

5 Results & Discussion

In this study, we successfully integrated TinyLlama/TinyLlama-1.1B-Chat-v1.0 with Intel© OpenVINO™toolkit to optimize the model for inference and fine-tuning on CPU-based systems. Key results include a significant performance improvement, with the quantized model exhibiting a 50% reduction in latency, enhancing real-time responsiveness and increasing throughput for more efficient query handling. Additionally, the optimized model demonstrated lower CPU utilization, making it suitable for deployment on standard Intel laptops and reducing energy consumption. Despite these optimizations, the model maintained high accuracy in Natural Language Processing tasks. Fine-tuning allowed the model to adapt quickly to specific tasks and contexts, enhancing its versatility. Furthermore, a Gradio interface was developed to demonstrate the chatbot's capabilities, providing a user-friendly platform for testing and interaction. In conclusion, this project demonstrates the effective integration of TinyLlama with OpenVINO, resulting in a highly efficient, accurate, and adaptable chatbot that performs well on Intel laptops, validated through a practical Gradio interface.

6 Acknowledgments

We extend our deepest gratitude to Intel© Corporation for this opportunity and to our mentor, Reni K Cherian, for her unwavering guidance and support. We are grateful to Saintgits College of Engineering and Technology for essential resources and sessions on machine learning. We thank the numerous researchers, scholars, and experts in machine learning, NLP, and AI whose groundbreaking work laid our project's foundation. Our appreciation goes to all the mentors, institutional leaders, and industry advisors who supported us during the Intel©-Unnati Programme, whose expertise and encouragement were vital in shaping our project. []

References

[1] ANDRIYANOV, N. Analysis of the acceleration of neural networks inference on intel processors based on openvino toolkit. In *2020 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)* (2020), IEEE, pp. 1–5.

[2] CHAUDHARI, B. S., GHORPADE, S. N., ZENNARO, M., AND PAŠKAUSKAS, R. Tinyml for low-power internet of things. In *TinyML for Edge Intelligence in IoT and LPWAN Networks*. Elsevier, 2024, pp. 1–12.

[3] GORBACHEV, Y., FEDOROV, M., SLAVUTIN, I., TUGAREV, A., FATEKHOV, M., AND TARKAN, Y. Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019), pp. 0–0.

[4] JOCHER, G., CHAURASIA, A., STOKEN, A., BOROVEC, J., KWON, Y., FANG, J., MICHAEL, K., MONTES, D., NADAR, J., SKALSKI, P., ET AL. ultralytics/yolov5: v6. 1-tensorrt, tensorflow edge tpu and openvino export and inference. *Zenodo* (2022).

[5] LIN, J. *Efficient Deep Learning Computing: From TinyML to LargeLM*. PhD thesis, Massachusetts Institute of Technology, 2024.

Project Code : GitHub

The project's code is available on GitHub at the following link: https://github.com/adilzubair/Bitmasters_Intel_LLM.