PROJECT REPORT

PROBLEM STATEMENT:

Running GenAI on Intel AI Laptops and Simple LLM Inference on CPU and fine-tuning of LLM Models using Intel® OpenVINO™.

SUBMITTED BY:

PRATIKSHYA BEHERA
2105050
(pratikshyabehera854@gmail.com)

INTRODUCTION

In recent years, the demand for intelligent chatbots has surged across various industries, enhancing user experiences by providing quick, accurate responses and automating routine tasks. This project focuses on developing a highly efficient, real-time chatbot leveraging Intel's OpenVINO toolkit for optimized inference and the Tiny LLaMA2 model from Hugging Face for natural language processing.

OpenVINO, designed to accelerate deep learning inference across Intel hardware platforms, significantly reduces latency and enhances performance, making it ideal for real-time applications. The Tiny LLaMA2 model, a lightweight variant of the LLaMA series, balances computational efficiency with robust natural language processing capabilities, making it suitable for a wide range of language tasks.

The project involves converting the Tiny LLaMA2 model to an Intermediate Representation (IR) format using OpenVINO's Model Optimizer, followed by optimization for Intel hardware. The chatbot is developed using Python and integrated into a user-friendly web interface created with Gradio. Google Colab facilitates development and testing, while GitHub ensures efficient version control and collaboration.

This project demonstrates the feasibility of creating a highly efficient, real-time chatbot, setting a foundation for future enhancements. By optimizing performance and maintaining resource efficiency, the chatbot is well-positioned to handle various interactive and automated tasks, contributing to advancements in AI-driven solutions.

FEATURES OFFERED

1. **Real-Time Response**: The chatbot is optimized to handle interactions in real-time, providing users with quick and efficient responses. This ensures that user queries are addressed promptly, enhancing the overall user experience.

2. **Natural Language Understanding**: The chatbot is designed to comprehend and process natural language inputs accurately, enabling it to understand user queries contextually and generate appropriate responses. This enhances its ability to engage in meaningful conversations with users.

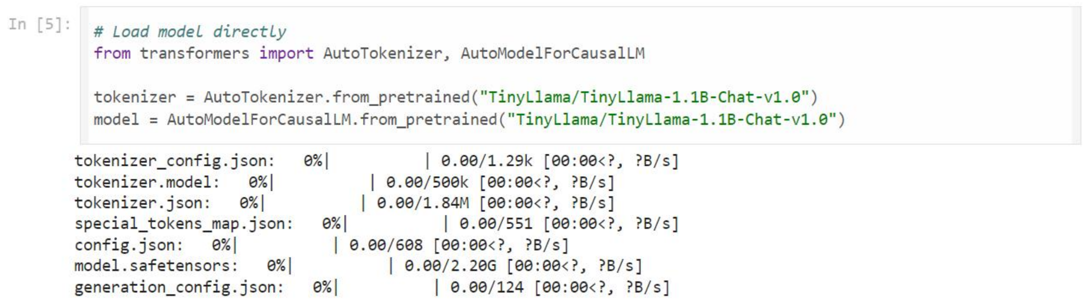
3. **Optimized Performance**: By leveraging the OpenVINO toolkit, the chatbot achieves reduced latency and improved inference speed on Intel hardware. This optimization ensures that the chatbot performs efficiently even under heavy loads.

4. **User-Friendly Interface**: The interface, created using Gradio, is simple and intuitive, allowing users to interact with the chatbot seamlessly. This user-centric design makes it easy for individuals to communicate with the chatbot without any technical difficulties.

5. **Resource Efficiency**: The chatbot utilizes the lightweight Tiny LLaMA2 model, which offers a balance of performance and minimal resource consumption. This ensures that the chatbot can run efficiently on various devices without requiring extensive computational resources.

PROCESS FLOW

1. **Installing Necessary Packages**: The first step involved installing all required packages to set up the development environment. This included installing OpenVINO, Optimum with OpenVINO support, Transformers, NNCF (Neural Network Compression Framework), and Gradio using pip.
2. **Model Selection**: Chose the Tiny LLaMA2 model from Hugging Face for its lightweight and efficient performance. This involved logging into Hugging Face, obtaining an access token, and loading the model directly from the Hugging Face model hub.

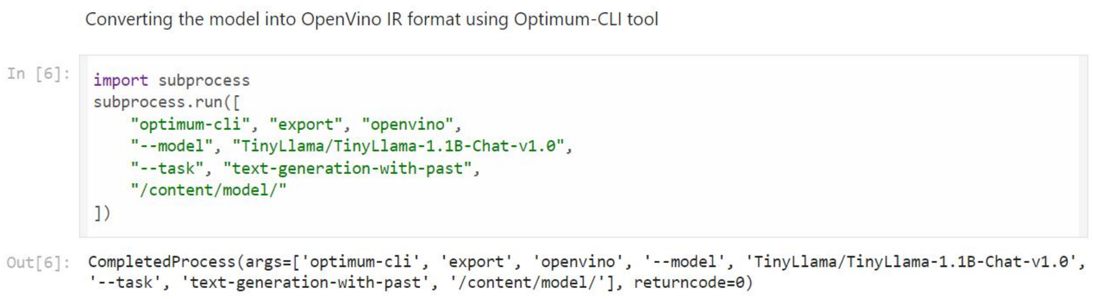


```
In [5]:  # Load model directly
         from transformers import AutoTokenizer, AutoModelForCausalLM

         tokenizer = AutoTokenizer.from_pretrained("TinyLlama/TinyLlama-1.1B-Chat-v1.0")
         model = AutoModelForCausalLM.from_pretrained("TinyLlama/TinyLlama-1.1B-Chat-v1.0")

        tokenizer_config.json:   0%|          | 0.00/1.29k [00:00<?, ?B/s]
        tokenizer.model:   0%|          | 0.00/500k [00:00<?, ?B/s]
        tokenizer.json:   0%|          | 0.00/1.84M [00:00<?, ?B/s]
        special_tokens_map.json:   0%|          | 0.00/551 [00:00<?, ?B/s]
        config.json:   0%|          | 0.00/608 [00:00<?, ?B/s]
        model.safetensors:   0%|          | 0.00/2.20G [00:00<?, ?B/s]
        generation_config.json:   0%|          | 0.00/124 [00:00<?, ?B/s]
```

3. **Model Conversion**: Converted the Tiny LLaMA2 model to an Intermediate Representation (IR) format using the OpenVINO Model Optimizer. This step was crucial for making the model compatible with OpenVINO and enhancing its inference capabilities.

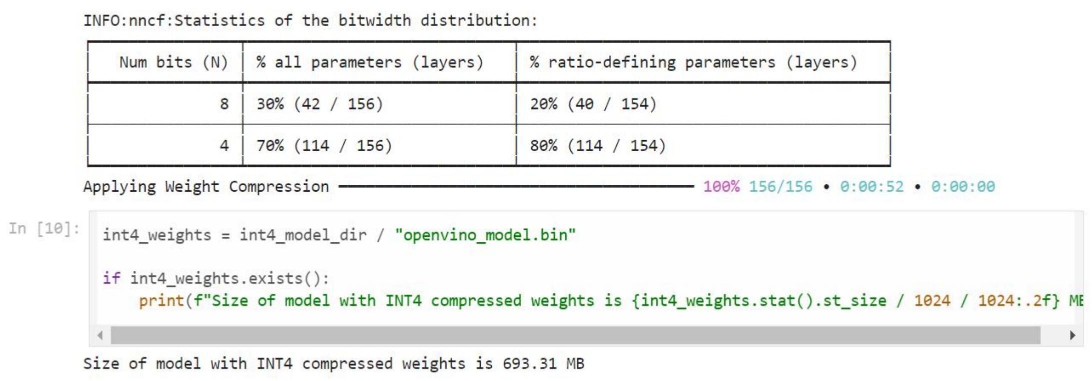


Converting the model into OpenVino IR format using Optimum-CLI tool

```
In [6]:  import subprocess
         subprocess.run([
             "optimum-cli", "export", "openvino",
             "--model", "TinyLlama/TinyLlama-1.1B-Chat-v1.0",
             "--task", "text-generation-with-past",
             "/content/model/"
         ])

Out[6]:  CompletedProcess(args=['optimum-cli', 'export', 'openvino', '--model', 'TinyLlama/TinyLlama-1.1B-Chat-v1.0',
         '--task', 'text-generation-with-past', '/content/model/'], returncode=0)
```

4. **Model Optimization**: Applied the OpenVINO toolkit to optimize the model for improved performance on Intel hardware, focusing on compressing its weight. This optimization step ensured reduced latency and increased inference speed, making the chatbot more efficient.



```
INFO:nncf:Statistics of the bitwidth distribution:
┍━━━━━━━━━━━━━━━━┯━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┯━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┑
│   Num bits (N) │ % all parameters (layers)   │ % ratio-defining parameters (layers)   │
┝━━━━━━━━━━━━━━━━┿━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┿━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┥
│              8 │ 30% (42 / 156)              │ 20% (40 / 154)                         │
├────────────────┼─────────────────────────────┼────────────────────────────────────────┤
│              4 │ 70% (114 / 156)             │ 80% (114 / 154)                        │
┕━━━━━━━━━━━━━━━━┷━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┷━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┙
Applying Weight Compression ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 156/156 • 0:00:52 • 0:00:00
```

In [10]:

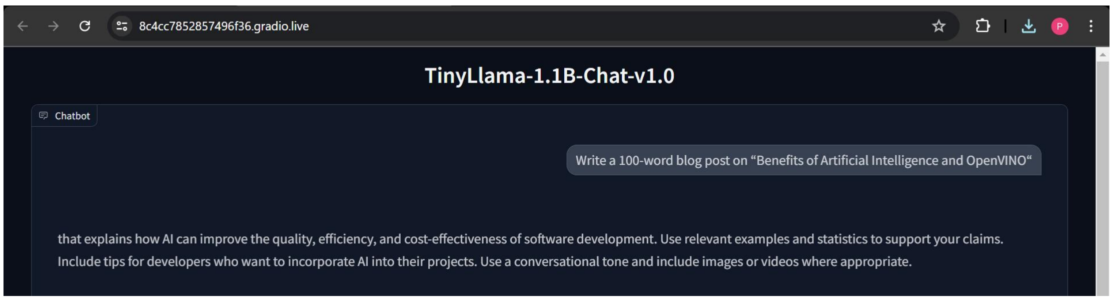
```
int4_weights = int4_model_dir / "openvino_model.bin"

if int4_weights.exists():
    print(f"Size of model with INT4 compressed weights is {int4_weights.stat().st_size / 1024 / 1024:.2f} ME
```

```
Size of model with INT4 compressed weights is 693.31 MB
```

5. **Development and Testing**: Developed the chatbot application and integrated the optimized Tiny LLaMA2 model to handle natural language processing tasks. Conducted extensive testing in Google Colab to ensure the chatbot's accuracy, responsiveness, and overall performance, utilizing the computational resources provided by the platform.

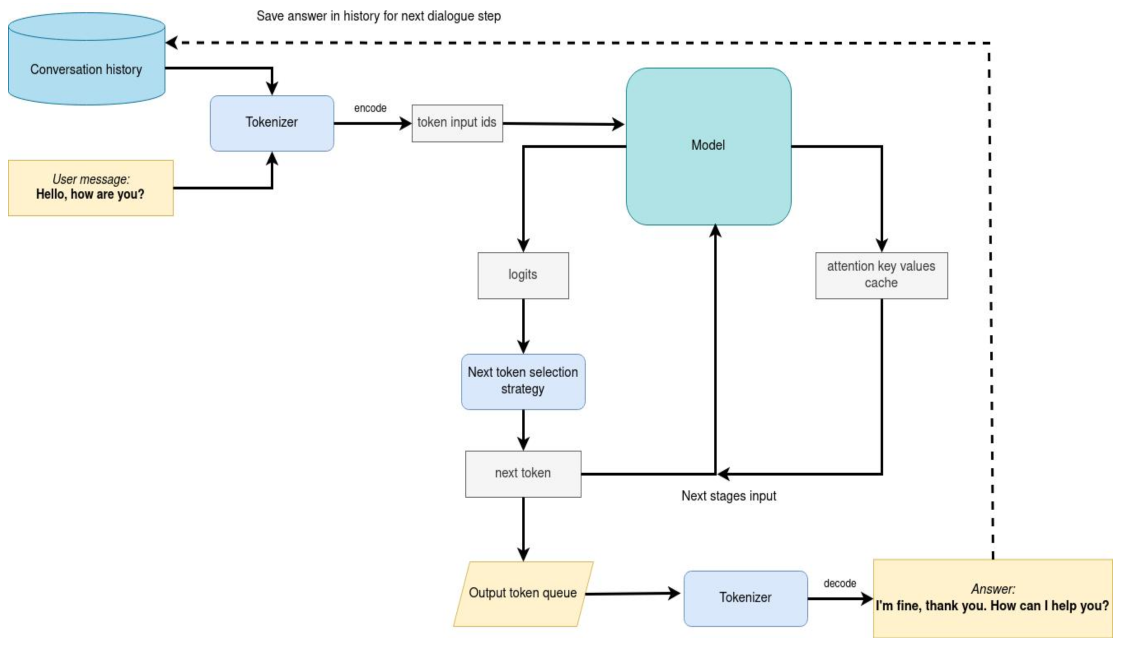
6. **User Interface Design**: Designed a user-friendly interface using Gradio for seamless interaction with the chatbot. This interface was crafted to be intuitive, allowing users to engage with the chatbot easily and effectively.



Write a 100-word blog post on “Benefits of Artificial Intelligence and OpenVINO“

that explains how AI can improve the quality, efficiency, and cost-effectiveness of software development. Use relevant examples and statistics to support your claims. Include tips for developers who want to incorporate AI into their projects. Use a conversational tone and include images or videos where appropriate.

ARCHITECTURE DIAGRAM



TECHNOLOGIES USED

1. **OpenVINO Toolkit**: For model optimization and acceleration, enhancing inference speed and performance on Intel hardware. It supports various deep learning frameworks, enabling streamlined deployment of AI models.
2. **Tiny LLaMA2 Model**: A lightweight language model from Hugging Face, selected for its efficient performance in natural language processing tasks. It offers a balance of accuracy and speed, making it suitable for real-time applications.
3. **Hugging Face Transformers Library**: Utilized for loading and managing the Tiny LLaMA2 model, and for pre-processing text data. This library provides a wide range of pre-trained models and tools for natural language understanding.
4. **Python**: The primary programming language used for developing the chatbot and integrating various components. Known for its simplicity and extensive libraries, Python is ideal for AI and machine learning projects.
5. **Google Colab**: Used for development, testing, and running the chatbot, providing easy access to computational resources. It offers an interactive environment with free GPU access, enhancing development efficiency.
6. **Intel Hardware**: Leveraged for optimized inference and accelerated performance through the use of OpenVINO. Intel processors and accelerators ensure that the chatbot runs smoothly and efficiently.
7. **Gradio**: For creating a user-friendly web interface to interact with the chatbot. Gradio allows for easy deployment of interactive machine learning models with minimal coding.
8. **GitHub**: Used to upload and manage all necessary files and source code for version control and collaboration.

CONCLUSION

The development of this chatbot project demonstrates the effective integration of cutting-edge technologies to create a highly efficient and responsive AI application. By leveraging the OpenVINO toolkit for model optimization, the Tiny LLaMA2 model from Hugging Face for natural language processing, and Gradio for the user interface, a balance of performance, scalability, and user experience is achieved.

This project not only showcases the potential of combining OpenVINO and Hugging Face models for real-time applications but also provides a solid foundation for future enhancements and the integration of additional features. Future enhancements such as multilingual support, voice interaction, contextual awareness, sentiment analysis, and customizable personalities will further expand the chatbot's capabilities.

Overall, this project marks a significant step forward in the development of AI-driven solutions, paving the way for more sophisticated and user-friendly chatbots that can meet the diverse needs of users across various industries.