Problem Statement

(PS-04): Introduction to GenAI and Simple LLM Inference on CPU and finetuning of LLM Model to create a Custom Chatbot

Unique Idea Brief (Solution)

Large Language Models, or LLMs, are a type of Generative Artificial Intelligence (GenAI), and this research investigates their potential for commercial use. It looks into whether LLM inference can be performed on CPUs that are easily accessible and shows how to optimize an LLM to build a personalized chatbot.

An overview of GenAI principles and LLM capabilities is provided in the report. After that, it looks at methods for effective LLM inference on CPU architectures, emphasizing the benefits of CPU-based solutions in business contexts. Lastly, the project shows how to refine a pre-trained LLM using a particular dataset to create a unique chatbot. The possible advantages and factors to take into account while implementing GenAI and customized chatbots in industrial settings are covered in the report's conclusion.

Features Offered

1. Environment and Dependency Management

- **Isolated Environment:**
  Utilize Conda to create an isolated environment ensuring consistency and avoiding conflicts.
- **Comprehensive Dependency Installation:**
  Install necessary dependencies using requirements.txt for easy setup.

2. Data Handling

- **Dataset Integration:**
  Seamlessly integrate the Alpaca dataset for training and evaluation purposes.

3. Model Fine-Tuning

- **Pre-trained Model Utilization:**
  Leverage the TinyLLama-1.1B-Chat-v1.0 model, tailored for conversational AI.
- **Customizable Training Parameters:**
  Define and adjust training parameters like learning rate, batch size, gradient accumulation, and epochs to optimize model performance.

Features Offered

4. Training and Evaluation

- **Periodic Evaluation:**
  Conduct evaluations at regular intervals to monitor and improve training performance.
- **Checkpointing:**
  Save model checkpoints periodically to resume training from the last saved state in case of interruptions.

5. Enhanced Optimization Techniques

- **Learning Rate and Weight Decay:**
  Fine-tune learning rate and weight decay parameters to prevent overfitting and enhance generalization.
- **Gradient Accumulation:**
  Use gradient accumulation to handle larger batch sizes effectively within memory constraints.

6. Robustness and Reproducibility

- **Random Seed Setting:**
  Ensure reproducibility by setting a random seed.
- **Best Model Loading:**
  Automatically load the best model at the end of training based on evaluation metrics.

Features Offered

7. Chatbot Implementation

- **Configurable Chatbot Pipeline:**
  Build a chatbot using the fine-tuned model with a configurable pipeline.
- **Interactive Query Handling:**
  Generate responses to user queries, enabling interactive conversational capabilities.

Process flow

1. Environment Setup

- **Create Conda Environment:**
  Create and activate with Python 3.10.
- **Install Dependencies:**
  Install intel-extension-for-transformers.

2. Repository and Data Setup

- **Clone Repository:**
  Clone fine-tuning GitHub repo.
- **Navigate to Directory:**
  Move to the fine-tuning directory.
- **Install Requirements:**
  Install from requirements.txt.

3. Authentication and Data Download

- **Hugging Face Login:**
  Authenticate using Hugging Face CLI.
- **Download Dataset:**
  Fetch Alpaca dataset.

Process flow

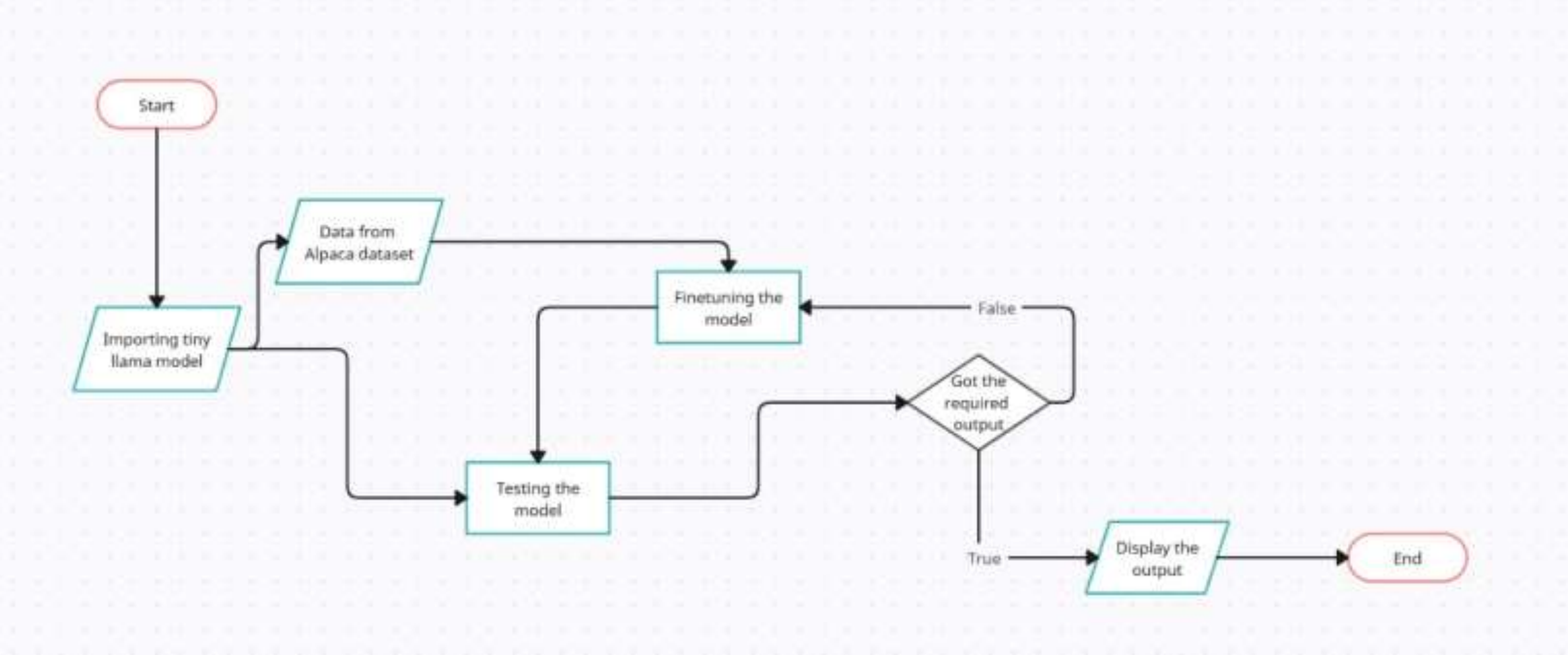
4. Fine-Tuning the Model

- **Define Model and Training Args:**
  Set model, data, and training arguments.
- **Fine-Tune Model:**
  Start the fine-tuning process.

5. Build Chatbot

- **Configure Chatbot:**
  Set pipeline configuration.
- **Build and Test Chatbot:**
  Generate responses to queries.

Architecture Diagram



Technologies used

Intel Extension for Transformers

TinyLlama-1.1B-Chat-v1.0

stanford_alpaca

Team members and contribution:

TEAM	CONTRIBUTION
Joshua Sunny Ninan	Chatbot Inference, Project Report
Edwin K Mathew	Chatbot Inference, Project Report
Rohith N.S	Fine-tuning, Presentation, GitHub
Avin Joy	Fine-tuning, Presentation

In addition to the specified roles, all team members actively participated and contributed equally to every aspect of the project, ensuring a balanced and collaborative effort.

Conclusion

This project explored the potential of Generative AI (GenAI), specifically Large Language Models (LLMs), for industrial applications. A successful simple LLM inference was achieved on a CPU, showcasing the potential for LLM deployment in resource-constrained environments. Additionally, we gained a practical understanding of the fine-tuning process, culminating in the development of a custom chatbot.

The findings suggest that CPU-based LLM inference, coupled with fine-tuning techniques, can bridge the gap between cutting-edge GenAI technology and practical industrial applications. This approach offers a cost-effective and accessible solution for leveraging the power of LLMs in various industrial settings.