1. Gpt_2 Finetuned

This focuses on fine-tuning a GPT-2 model for generating answers to medical exam questions.

- **Loading and Preprocessing the Data**:
    - The data is loaded from a JSON file and is transformed into a Pandas DataFrame. The key fields include questions, options (A, B, C, D), the correct option, and relevant metadata like the topic name and subject.
    - The dataset is processed into a format compatible with Hugging Face's Dataset format.
- **Tokenization and Model Fine-Tuning**:
    - Dataset is loaded into the GPT-2 tokenizer and prepares the data for training by creating a tokenized dataset.
    - A tokenization function is implemented to format the input as a question with multiple options, followed by the expected answer.
    - The fine-tuning process uses Trainer and TrainingArguments from the Hugging Face Transformers library.
- **Model Training and Evaluation**:
    - The fine-tuned model is used to generate responses for multiple-choice medical questions.
    - After training, the notebook uses the fine-tuned model to predict answers based on newly presented questions.

2. RAG_pipeline_using_Fine_tuned_GPT_LLM.ipynb

Retrieval-Augmented Generation (RAG) pipeline using a fine-tuned GPT-2 model to answer questions based on retrieved document chunks. Key components include:

- **Document Embedding and FAISS Retrieval**:
    - Document Embedding: The document chunks are embedded into dense vector representations using a DistilBERT.
    - Question Embedding: Similarly, the user's input question is embedded into a dense vector representation using the same pretrained model.
    - The FAISS index is used to retrieve top-k most relevant document chunks based on the similarity of user questions with the chunks.
    - FAISS search is performed after embedding the user's input using a DistilBERT model.
- **Answer Generation**:
    - The fine-tuned GPT-2 model generates answers based on the context provided by the retrieved chunks.
    - For each chunk, a question is posed to GPT-2, and the generated answers are retrieved.
- **Answer Selection**:
    - The generated answers are compared against the available multiple-choice options using cosine similarity between the embeddings of the options and the generated answers.
    - The final answer is predicted by selecting the most frequent answer across the generated responses.

GPU Optimization

Implemented techniques like
Batch Processing - The document embeddings and question embeddings were generated in batches, minimizing the number of GPU kernel calls and reducing memory fragmentation.
Mixed Precision Computation - The model was trained using mixed precision, which reduced memory usage and increased the training speed by allowing the model to run with 16-bit floats (half precision) where applicable, instead of 32-bit floats (full precision).