**Introducing Generative AI with AWS Project**

***Project Title: Building Domain Expert Model***

***Name: Mukhtarr SK Manjang***

***Date: 05/09/2024***

**Introduction**

***Project Overview***: This model will be trained on a dataset of domain-specific knowledge (medical dataset). The model can then be used to create chat applications, internal knowledge applications, or text content generation for company collateral.

***Project Objective***: My task in this project is to train (fine-tune) a large language model. The model is trained to become a domain expert on a medical dataset, It’s capable of generating informative, accurate, and contextually relevant text responses.

**Dataset Description**

The dataset focuses on genomic profiling in myeloid neoplasms and acute leukemia, which is critical for diagnostic evaluation, risk assessment, and therapeutic decision-making

**The Model Architecture (Llama2 7B)**

The Llama2 7B model is2 7B model is based on the transformer architecture, which is widely used in natural language processing. Here are some key features of built on the transformer architecture, its architecture:

1. **Transformer Blocks**: The model consists of multiple transformer blocks that include layers for self-attention which is highly effective for natural and feed-forward neural networks. This structure allows the model to process and understand complex language patterns.
2. **7 Billion language processing tasks. Here are some key features of Parameters**: As indicated by "7B," the model has approximately 7 billion parameters, which enables it to capture its architecture
3. A wide range of linguistic nuances and generate coherent text.
4. **Multi-Head Attention**: It employs multi-head attention mechanisms, allowing it to attend to different parts of the input sequence simultaneously, a transformer architecture that includes both encoder and decoder components, allowing thus enhancing its contextual understanding.
5. **Layer Normalization and Residual Connections**: Each transformer block includes layer normalization and residual connections, which help stabilize training and improve for efficient processing of sequential data the flow of gradients.
6. **7 Billion Parameters**: The model contains The model is pre-trained on a large corpus of text data and can be fine-tuned on specific tasks or domains, making it versatile for various applications in text generation and understanding.

This architecture enables Llama2 to perform effectively across a range of natural language processing tasks, showcasing its capabilities in generating high-quality text. approximately 7 billion parameters, enabling it to learn and generate complex language patterns and nuances.

1. **Self-Attention Mechanism**: Llama2 uses multi-head self-attention, which allows the model to weigh the importance of different words in a sentence, capturing contextual relationships effectively.
2. **Layer Stacking**: The architecture consists of multiple stacked transformer layers, which enhances the model's ability to learn hierarchical representations of language.
3. **Pre-training and Fine-tuning**: It is pre-trained on a large corpus of text and can be fine-tuned for specific tasks, improving its performance in generating domain-specific content.

**Evaluation Process**

1. **Deployment of the Meta Llama 2 7B Model**: The model was deployed using Amazon SageMaker to evaluate its text generation capabilities.
2. **Setup and Authentication**: AWS services were authenticated by using the SageMaker Python SDK to ensure smooth deployment.
3. **Model Inference**: The Llama 2 7B model was evaluated by providing text input and generating output predictions, assessing its performance in text sequence generation.
4. **Output Analysis**: The text generation results were reviewed to evaluate the model's relevance and capability in the domain it was tested on.
5. **Endpoint Deletion**: After evaluation, the deployed model and its associated resources were deleted to prevent unnecessary costs.

**Evaluation Outputs summary**

The model generated a sequence of text based on the provided input prompt, predicting the next words in the sequence. The outputs were evaluated based on their relevance and coherence within the domain specified in the prompt. The generated text without fine-tuning was likely found to be limited in providing domain-specific insights, suggesting a need for fine-tuning to improve performance.

**Fine-Tune process**

1. ***Model Setup***: The fine-tuning process uses the Meta Llama 2 7B model on AWS SageMaker, leveraging pre-trained weights.

2. ***Training Data***: The model is trained on domain-specific data to enhance its performance in generating relevant text sequences for a specific context.

3. ***Training Process***: Multiple epochs are executed, with loss and perplexity metrics monitored to track model improvements. Example: After the first epoch, the loss was reduced to 2.5145, and the perplexity was 12.3604.

4. ***Evaluation***: The fine-tuned model is evaluated using perplexity to assess its ability to predict text sequences.

5. ***Model Saving***: The best-performing model based on evaluation metrics is saved for future use.

6. ***Resource Cleanup***: After evaluation, the model and endpoints are deleted to avoid unnecessary costs.

**The Fine-Tune model Output summary**

***Training Performance***: After fine-tuning, the model achieved:

**A loss of 2.5145** and a perplexity **of 12.3604** during the evaluation after epoch 0.

For epoch 1, the training perplexity was further reduced to **8.2131**, with a corresponding loss of **2.1057**.

*The fine-tuned model could generate domain-specific and relevant text content based on the input prompts provided. The results indicated that fine-tuning improved the model’s performance in producing more coherent and relevant text predictions within the targeted domain*.

***Conclusion***

Fine-tuning significantly enhanced the relevance and quality of text generation, making the model better suited for specific applications. However, the fine-tuned model can be further tested and potentially deployed for real-world applications, especially in domains like healthcare, where accurate and informative text generation is critical. The scalability of the medical dataset needs to be improved by indulging in more relevant research in the subject to minimize any potential risk and biases.

This project demonstrates the power of model fine-tuning in improving the performance of pre-trained models for specialized tasks.

*Reference: Udacity AI(project documentation format)*