**Mastering GPTs: From Scratch to Real-World Applications**

*Anish Mahapatra*

***Main Prerequisites****: Intermediate Python, understanding of the structure of GPTs from the Book – Build a Large Language Model (From Scratch)*

***Main Skills Learned***: How to build a large language model (LLM) from scratch, how to build an AI application, how to finetune your GPT for a particular use-case

1. **Tell us about yourself**

Anish Mahapatra is a Senior Data Scientist at a Fortune 200 company with over 7 years of experience in building end-to-end applications. He possesses extensive expertise in artificial intelligence (AI), machine learning (ML), Python, PySpark, large language models (LLMs), and cloud technologies.

Anish has made significant contributions to Data Science, Data Engineering, and AI by creating content, courses, and blogs for platforms such as UpGrad, Towards Data Science, and KnowledgeHut. Holding a master’s degree in data science, he is passionate about emerging technologies and has garnered over 150k views on Medium and other technology blogs.

1. **Give us a brief description of the series**

This series of live projects is designed to take learners on a comprehensive journey through the building, adapting, and fine-tuning GPT models using Python. Starting with the foundational project of building a custom GPT model locally, learners will gain an in-depth understanding of the GPT architecture and essential skills in data preprocessing and tokenization. The series progresses to developing a GPT model capable of interacting with PDF documents, enabling learners to extract and process text data from PDFs and create interactive dialogue systems based on the extracted content. Each project builds upon the previous one, ensuring a cohesive learning experience that introduces more complex concepts and techniques gradually.

In the final project, learners will fine-tune a pre-trained GPT model for specific tasks such as question-answering using the Stanford Question Answering Dataset (SQuAD). This hands-on approach will reinforce theoretical concepts and provide practical experience in real-world applications of GPT models.

By the end of the series, learners will have mastered the art of building and fine-tuning large language models. This structured progression from foundational knowledge to advanced application ensures that learners gain both confidence and competence in working with GPT models.

1. **Who are the projects for and what will they learn?**

These projects are designed for intermediate to advanced learners who have a foundational understanding of machine learning and deep learning principles. They are ideal for data scientists, AI enthusiasts, and developers looking to deepen their knowledge of GPT models and their practical applications.

**Prerequisites**

* **Machine Learning Basics**: Learners should have a good understanding of machine learning concepts and techniques
* **Deep Learning Foundations**: Familiarity with the structure and working of neural networks is helpful
* **Programming Skills**: Proficiency in Python programming is necessary
* **Framework Knowledge**: Prior experience with PyTorch is helpful, as the projects will use this framework
* **Data Handling**: Basic skills in data preprocessing and manipulation using libraries such as NumPy and pandas

**What will learners get out of these live projects?**

* **Implementing GPT models from scratch**:   
  Gain hands-on experience in building GPT models from the ground up, understanding the intricacies of their architecture
* **Data preprocessing and tokenization**:   
  Learn essential techniques for preparing and tokenizing text data, crucial to training effective language models
* **Training and evaluating models**:   
  Develop skills in setting up training loops, monitoring performance, and evaluating model outputs to ensure high-quality results
* **Extracting and processing text from PDFs**:   
  Master the ability to extract textual information from PDF documents and preprocess it, expanding the practical applications of GPT models
* **Developing interactive applications**:   
  Build interactive systems that leverage GPT models for real-time text generation and dialogue, to enhance user engagement and functionality
* **Fine-tuning pre-trained models**:   
  Learn how to adapt pre-trained GPT models to specific tasks, improving their performance on domain-specific applications
* **Build domain-specific applications**:   
  Apply GPT models to specialized fields, to create tailored solutions that address unique challenges and opportunities in various industries

**Languages and Libraries that will be used**

* **Programming Languages**
  + Python
* **Libraries and Frameworks**
  + PyTorch
  + NumPy
  + Pandas
  + Transformers (Hugging Face)

These projects provide a structured and incremental learning path, ensuring that learners understand the theoretical aspects of GPT models and also gain the practical experience needed to implement and adapt these models for real-world applications.

Whether you’re a data scientist, developer, or AI enthusiast, these projects will provide you with the hands-on experience and confidence to implement advanced AI solutions. Follow along with me to transform your understanding of real-world end-to-end AI with these exciting and practical projects!

1. **Brief Project Outlines**

**Project #1: Build Your Custom LLM Locally (4-5 hours)**

This project guides learners through the process of building a custom GPT model locally. It focuses on understanding the foundational architecture of GPT models and training a small-scale version on a local machine. Learners will gain hands-on experience in setting up the environment, preparing data, building the model architecture, training the model, and generating text with the trained model. This project is ideal for those looking to dive deep into the mechanics of GPT models and gain practical skills in implementing and training these models from scratch.

* **Milestone 1**: Introduction to GPT and code Implementation

Set up the environment and understand the basics of GPT architecture

* **Milestone 2**: Data Preparation and Tokenization   
  Prepare a small text dataset and implement tokenization using Byte-Pair Encoding (BPE)
* **Milestone 3**: Building the Model Architecture   
  Code the transformer architecture from scratch
  + Implement the attention mechanism
  + Construct feed-forward layers
* **Milestone 4**: Training the Model
  + Set up the training loop and optimizer
  + Train the model on the prepared dataset, monitoring its performance
* **Milestone 5**: Generating Text with the Trained Model   
  Implement a text generation function and test the model by generating new text samples

This project will equip learners with the skills to understand, implement, and train a GPT model locally, providing a strong foundation for more advanced projects.

**Project #2: Build a GPT to Chat with a PDF (3-4 hours)**

This project enables learners to build a GPT model capable of extracting information from PDFs and engaging in interactive dialogues based on the extracted content. By integrating PDF processing techniques with GPT models, learners will create a system that can understand and answer questions about the content of PDF documents. This project provides practical experience in data extraction, model training, and developing interactive applications, making it a valuable learning opportunity for learners interested in natural language processing and AI-driven interactions.

* **Milestone 1**: Introduction to PDF Processing

Set up the environment and learn how to extract text from PDFs

* **Milestone 2**: Data Extraction and Preprocessing

Extract text from sample PDFs and preprocess the data for model training.

* Handle text extraction from various PDF formats
* Clean and prepare the extracted text for training
* **Milestone 3**: Building and Training the GPT Model

Adapt the GPT architecture for the extracted text and train the model on this data.

* Implement the GPT model using PyTorch
* Train the model on the extracted PDF text data
* **Milestone 4**: Implementing the Chat Interface   
  Develop a chat interface that allows users to interact with the GPT model based on PDF content
* **Milestone 5**: Testing and Evaluation

Test the chat interface with various PDFs and evaluate the model’s performance in answering questions

This project will equip learners with the skills to develop interactive AI applications that can process and understand document content, opening new possibilities for automating information retrieval and enhancing user interactions.

**Project #3: Fine-tune Your GPT for Q&A with SQuAD Data (4-5 hours)**

This project focuses on fine-tuning a pre-trained GPT model for question-answering tasks using the Stanford Question Answering Dataset (SQuAD). Learners will explore the techniques of fine-tuning large language models on domain-specific datasets to achieve high performance on targeted tasks. By the end of the project, learners will have a functional Q&A system capable of answering questions based on the content of a given text. This project is ideal for those looking to enhance their skills in model adaptation and fine-tuning for specific applications in natural language processing.

* **Milestone 1**: Introduction to Fine-tuning and SQuAD Dataset

Understand the concept of fine-tuning and get familiar with the SQuAD dataset

* **Milestone 2**: Preparing the Dataset

Download and preprocess the SQuAD dataset for model training

* **Milestone 3**: Fine-tuning the GPT Model

Set up the training environment and fine-tune the GPT model on the SQuAD dataset

* + Implement the training loop for fine-tuning
  + Monitor and adjust model parameters during training
* **Milestone 4**: Implementing the Q&A System

Develop a system to handle user queries and generate answers using the fine-tuned model.

* Build an interface for users to input questions
* Integrate the fine-tuned model to provide answers
* **Milestone 5**: Evaluation and Optimization

Evaluate the model’s performance on the SQuAD validation set and optimize for better accuracy

This project will provide learners with the skills to adapt and fine-tune large language models for specific tasks, enhancing their ability to create highly accurate and functional AI applications in the field of natural language processing.

1. **Recommended Reading**

To supplement your learning and enhance your understanding of the concepts covered in these projects, we recommend the following resources:

**Books and Chapters**

* ***Build a Large Language Model (From Scratch) - Manning***
* Chapters 1, 2, 3, 4, 5, 6, 7, Appendix A, B
* ***Generative AI in Action - Manning***
* Chapter 1: Understanding Generative AI Basics
* Chapter 3: Working Through an API - Generating Text

- **Online Articles and Tutorials**

PyTorch Documentation and Tutorials

* ***PyTorch Documentation****:* [*https://pytorch.org/docs/stable/index.html*](https://pytorch.org/docs/stable/index.html)
* ***PyTorch Tutorials****:* [*https://pytorch.org/tutorials/*](https://pytorch.org/tutorials/)

Essential resources for mastering PyTorch, including detailed explanations and hands-on examples.

- **Hugging Face Blog and Tutorials**

* ***Hugging Face Blog****:* [*https://huggingface.co/blog/*](https://huggingface.co/blog/)
* ***Hugging Face Tutorials****:* [*https://huggingface.co/transformers/tutorials/*](https://huggingface.co/transformers/tutorials/)
* Valuable insights on working with transformers and large language models, including model training and fine-tuning.

These resources will give you a deeper understanding of the theoretical and practical aspects of building and fine-tuning large language models, enhancing your learning experience with the series of live projects and equipping you with the knowledge to tackle advanced AI projects.

1. **Competition**

It’s important to understand the competitive landscape to ensure the uniqueness and value of these live projects. Here are some similar projects and courses available:

* **Minimal GPT Implementation**:   
  This project offers a minimal and efficient implementation of the GPT model in PyTorch, focusing on educational purposes and small-scale applications.   
  *GitHub Repository:* [*https://github.com/iVishalr/GPT*](https://github.com/iVishalr/GPT)
* **LitGPT**:   
  Trains a minimal version of a GPT language model using PyTorch Lightning.   
  *GitHub Repository:* [*https://github.com/tomogwen/LitGPT*](https://github.com/tomogwen/LitGPT)
* **MicroGPT**:   
  A lightweight implementation of the GPT model, designed to be easy to modify and suitable for small-scale applications.   
  *GitHub Repository:* [*https://github.com/LeeSinLiang/microGPT*](https://github.com/LeeSinLiang/microGPT)
* **GPT-1**:   
  An implementation of the GPT-1 model, that includes code for preprocessing training data and the pre-training / fine-tuning process.   
  *GitHub Repository:* [*https://github.com/akshat0123/GPT-1*](https://github.com/akshat0123/GPT-1)

While these projects do not directly compete with the overall series of live projects, they may serve as valuable references for implementing GPT models using PyTorch and highlight the practical applications of generative pre-trained transformers in educational settings.