

Complete Generative AI Course: From Basics to Expert Level

Duration: 10 days (08 hours/day)

Chapter 01: Open Source Models with Hugging Face

- Learn how to easily build AI applications using open source models and Hugging Face tools.
- Find and filter open source models on Hugging Face Hub based on task, rankings, and memory requirements.
- Write just a few lines of code using the transformers library to perform text, audio, image, and multimodal tasks.
- Easily share your AI apps with a user-friendly interface or via API and run them on the cloud using Gradio and Hugging Face Spaces.

Chapter 02: Quantization Fundamentals with Hugging Face

- Learn how to quantize any open source model.
- Learn how to compress models with the Hugging Face Transformers library and the Quantolibrary.
- Learn about linear quantization, a simple yet effective method for compressing models.
- Practice quantizing open source multimodal and language models.

Prerequisite: A basic understanding of machine learning concepts using PyTorch.

Chapter 03: Preprocessing Unstructured Data for LLM Applications

- Improve your RAG system to retrieve diverse data types
- Learn to extract and normalize content from a wide variety of document types, such as PDFs, PowerPoints, Word, and HTML files, tables, and images to expand the information accessible to your LLM.
- Enrich your content with metadata, enhancing retrieval augmented generation (RAG) results and supporting more nuanced search capabilities.
- Explore document image analysis techniques like layout detection and vision and table transformers, and learn how to apply these methods to preprocess PDFs, images, and tables.

Chapter 04: Prompt Engineering with Llama 2 & 3

- Learn best practices for prompting and selecting among Meta Llama 2 & 3 models.
- Learn best practices specific to prompting Llama 2 & 3 models.
- Interact with Meta Llama 2 Chat, Code Llama, and Llama Guard models.
- See how you can build safe, responsible AI applications using the Llama Guard model.

Chapter 05: Building Applications with Vector Databases

- Learn to build six applications powered by vector databases: semantic search, retrieval augmented generation (RAG), anomaly detection, hybrid search, image similarity search, and recommender systems, each using a different dataset.
- Learn to create six exciting applications of vector databases and implement them using Pinecone.
- Build a hybrid search app that combines both text and images for improved multimodal search results.
- Learn how to build an app that measures and ranks facial similarity.

Chapter 06: LangChain for LLM Application Development

- The framework to take LLMs out of the box. Learn to use LangChain to call LLMs into new environments, and use memories, chains, and agents to take on new and complex tasks.
- Learn LangChain directly from the creator of the framework, Harrison Chase
- Apply LLMs to proprietary data to build personal assistants and specialized chatbots
- Use agents, chained calls, and memories to expand your use of LLMs

Prerequisite: Basic Python

Chapter 07: LangChain: Chat with Your Data

- Create a chatbot to interface with your private data and documents using LangChain.
- Learn from LangChain creator, Harrison Chase
- Utilize 80+ loaders for diverse data sources in LangChain
- Create a chatbot to interact with your own documents and data

Prerequisite: Basic Python

Chapter 08: Knowledge Graphs for RAG

- Learn how to build and use knowledge graph systems to improve your retrieval augmented generation applications.
- Use Neo4j's query language Cypher to manage and retrieve data stored in knowledge graphs.
- Write knowledge graph queries that find and format text data to provide more relevant context to LLMs for Retrieval Augmented Generation.
- Build a question-answering system using Neo4j and LangChain to chat with a knowledge graph of structured text documents.
- Use Mistral's API to call user-defined Python functions for tasks like web searches or retrieving text from databases, enhancing the LLM's ability to find relevant information to answer user queries.

Prerequisite recommendation: Familiarity with LangChain

Chapter 09: Large Language Models with Semantic Search

- Learn to use LLMs to enhance search and summarize results.
- Enhance keyword search using Cohere Rerank
- Use embeddings to leverage dense retrieval, a powerful NLP tool
- Evaluate your effectiveness for further optimization

Prerequisite: Basic Python

Chapter 10: Reinforcement Learning from Human Feedback

- A conceptual and hands-on introduction to tuning and evaluating large language models (LLMs) using Reinforcement Learning from Human Feedback.
- Get a conceptual understanding of Reinforcement Learning from Human Feedback (RLHF), as well as the datasets needed for this technique
- Fine-tune the Llama 2 model using RLHF with the open source Google Cloud PipelineComponents Library
- Evaluate tuned model performance against the base model with evaluation methods

Prerequisite: Intermediate Python

Chapter 11: Finetuning Large Language Models

- Learn to finetune an LLM in minutes and specialize it to use your own data
- Master LLM finetuning basics
- Differentiate finetuning from prompt engineering and know when to use each
- Gain hands-on experience with real datasets for your projects

Prerequisite: Basic Python

Chapter 12: JavaScript RAG Web Apps with LlamaIndex

- Build a full-stack web application that uses RAG capabilities to chat with your data.
- Learn how to build a RAG application in JavaScript, and use an intelligent agent that discerns and selects from multiple data sources to answer your queries.
- Build a full-stack web app with an interactive frontend component that interacts and chats with your data.
- Learn how to persist your data, enable chatting with your data and make streaming responses possible, all implemented using the create-llama command-line tool

Prerequisite: Basic JavaScript

Chapter 13: LLM Ops

- Learn LLM Ops best practices as you design and automate the steps to tune an LLM for a specific task and deploy it as a callable API. In the course, you'll tune an LLM to act as a question-answering coding expert. You can apply the methods learned here to tune your own LLM for other use cases.
- Adapt an open source pipeline that applies supervised fine-tuning on an LLM to better answer user questions.
- Learn best practices, including versioning your data and your models

- Learn responsible AI by outputting safety scores on sub-categories of harmful content.

Prerequisite: Basic Python