

## Gen AI Immersive: 3-Day Intermediate Course Outline (Detailed)

### Prerequisites:

- Basic understanding of Deep Learning concepts like Neural Networks (structure, types), Activation Functions (ReLU, Sigmoid etc.), Gradient Descent and its variants (SGD, Adam)
- Familiarity with Python programming and libraries like NumPy, Pandas

### Course Objectives:

By the end of this course, participants will be able to:

- **Explain** the role of Sequence Models (RNN, LSTM, Transformers) in Generative AI and their strengths/weaknesses.
- **Utilize** pre-trained word embeddings and vector databases to efficiently represent and retrieve information for text generation.
- **Implement** Retrieval-Augmented Generation (RAG) techniques to combine retrieval and generation for improved outcomes.
- **Apply** LangChain to construct complex and coherent generations by chaining multiple generative models.
- **Master** Prompt Engineering using GPT-4 for crafting effective prompts that guide model outputs towards specific goals.
- **Analyze** real-world case studies showcasing Gen AI applications across diverse domains (Marketing, Content Creation, Customer Service, Code Generation).

### Day 1: Deep Learning Recap and Sequence Models (In-Depth)

- **Recap of Deep Learning Fundamentals** (1 hour)
  - Interactive review with exercises: Participants revisit Neural Network architectures (Perceptrons, MLPs, CNNs), Activation Functions (demonstrating their impact on learning), and Loss Functions (cross-entropy, mean squared error).
  - Code walkthrough: Implement a basic Neural Network in Python using libraries like TensorFlow or PyTorch.
- **Sequence Models for Text Generation** (3 hours)
  - **Recurrent Neural Networks (RNNs):** Detailed explanation with diagrams. Explore Unfolding Time and Backpropagation Through Time (BPTT) with code examples. Discuss limitations of RNNs (vanishing gradients).
  - **Long Short-Term Memory (LSTM):** Introduce the LSTM architecture and its cell structure (forget gate, input gate, output gate) with visualizations. Explain how LSTMs address vanishing gradients. Code a basic LSTM for text generation.
  - **Transformers (Encoder-Decoder Architecture):** Deep dive into the Transformer architecture (Encoder, Decoder, Self-Attention mechanism).

Discuss the benefits of parallelization compared to RNNs. Code a simple Transformer model.

- **Case Study:** Generate different writing styles with LSTMs (e.g., news articles, poems). Participants will be provided with a dataset and code template to train LSTMs for generating text in various styles.
- **Word Embeddings (2 hours)**
  - Explore Word2Vec (CBOW & Skip-Gram models) and GloVe for representing words as vectors. Discuss capturing semantic relationships using cosine similarity.
  - Hands-on Lab: Participants will implement Word2Vec and GloVe algorithms in Python to generate word embeddings for a given vocabulary. They will then analyze word relationships using cosine similarity.

## Day 2: Retrieval-Augmented Generation (RAG) and LangChain (Implementation)

- **Generative AI: The Art of Creation & Expanding the Palette (2 hours)**
  - Discussing different types of Generative AI models (e.g., Generative Adversarial Networks, Variational Autoencoders, Transformers).
  - Highlighting the potential of Gen AI for various creative applications (e.g., generating music, writing poems, designing images).
- **Retrieval-Augmented Generation (RAG) (2 hours)**
  - Explain the core concept of RAG: combining generative and retrieval techniques for text generation. Discuss how to retrieve relevant information to inform the generation process.
  - Code walkthrough: Implement a basic RAG model using libraries like Hugging Face Transformers. Participants will learn how to retrieve relevant passages from a document corpus and use them to condition the generation process.
- **LangChain (2 hours)**
  - Introduce the concept of LangChain: building complex and coherent generations in stages by chaining multiple generative models together. Explore different LangChain architectures.
  - Hands-on Project: Participants will design and implement a LangChain model for a specific task (e.g., generating story summaries). They will chain together multiple models (e.g., one for generating plot points, another for dialogue) to achieve the desired outcome.

## Day 3: Prompt Engineering with GPT-4 (Advanced Techniques)

- **Prompt Engineering for GPT-4 (3.5 hours)**
  - Explain how prompts guide GPT-4 outputs and influence the generation process.
  - Discuss different prompt engineering techniques (e.g., few-shot learning, cloze deletion, informative prompts).
  - Hands-on Workshop: Participants will experiment with crafting effective prompts for GPT-4 using various techniques. They will be provided with a

dataset and explore how different prompts lead to different generation outputs.

- **Gen AI in Marketing & Advertising (0.5 hour)**
  - Explore various applications of Gen AI in marketing and advertising, including:
    - Generating personalized ad copy and product descriptions based on user data.
    - Creating dynamic marketing materials that adapt to different audiences.
  - **Interactive Discussion:** Participants will brainstorm potential applications of Gen AI for their own marketing and advertising needs.
- **Gen AI in Creative Content (0.5 hour)**
  - Showcase the use of Gen AI for creative content generation, such as:
    - Creating different creative text formats (e.g., poems, scripts, musical pieces).
    - Generating story ideas and outlines.
  - **Group Project:** Participants will work in teams to brainstorm and develop a creative writing project using Gen AI techniques (e.g., generating a poem in a specific style).
- **Gen AI in Customer Service (0.5 hour)**
  - Discuss how Gen AI can be used to improve customer service experiences through:
    - Developing chatbots for efficient customer support.
    - Generating personalized responses to customer inquiries.
  - **Case Study Analysis:** Participants will analyze a real-world case study of a company using Gen AI for customer service and discuss the challenges and benefits.
- **Gen AI in Code Generation (0.5 hour)**
  - Explore the potential of Gen AI for automating repetitive coding tasks:
    - Generating code snippets based on natural language descriptions.
    - Completing code based on existing functionality.
  - **Live Coding Demonstration:** Showcase how to use Gen AI models for basic code generation tasks using libraries like OpenAI API.
- **Course Wrap-up & Discussion (30 mins):**
  - Open forum for participants to ask questions and discuss their key takeaways from the course.
  - Provide additional resources and next steps for further exploration of Gen AI.