Generative AI and Large Language Models – Complete Exam Notes

1. Generative Artificial Intelligence

Generative AI is a branch of Artificial Intelligence that focuses on generating new content such as text, images, audio, video, and code. Unlike traditional AI systems that only classify or predict outcomes, Generative AI creates original outputs by learning patterns from large datasets. Popular generative models include ChatGPT, image generators, and music generators.

2. Relationship Between AI, ML, DL, and Generative AI

Artificial Intelligence (AI) is the broad field of building intelligent systems. Machine Learning (ML) is a subset of AI where systems learn from data. Deep Learning (DL) is a subset of ML that uses deep neural networks. Generative AI is a specialized part of Deep Learning that generates new content instead of only making predictions.

3. Foundation Models

Foundation models are very large models trained on massive datasets using transformer architectures. They can perform many tasks without task-specific training. Key properties of foundation models are emergence (the ability to perform unseen tasks) and homogenization (same architecture works across many domains).

4. Transformer Architecture

Transformers are deep learning models designed to handle sequential data such as text, images, and audio. They use a mechanism called self-attention to understand relationships between tokens. Transformers allow parallel processing, handle long-range dependencies, and scale efficiently. They have replaced RNNs and CNNs in many tasks.

5. Applications of Transformers

Transformers are widely used in Natural Language Processing tasks such as translation, summarization, question answering, and chatbots. They are also used in Computer Vision for image classification and detection, in audio processing for speech recognition, and in multimodal systems combining text, image, and audio.

6. Large Language Models (LLMs)

Large Language Models are a type of foundation model trained on trillions of tokens. They use transformer architectures and contain billions of parameters. LLMs can understand context, generate human-like text, answer questions, summarize documents, write code, and more.

7. Tokens, Prompts, and Context Window

LLMs process text as tokens, which are words or parts of words. A prompt is the input text given to the model, and the output is called a completion. The context window defines the maximum number of tokens the model can handle at once.

8. Types of LLMs

Base LLMs are trained to predict the next token and do not follow instructions well.
Instruction-tuned LLMs are fine-tuned to follow human instructions and are safer and more helpful.
Specialized LLMs are adapted for specific domains such as healthcare or finance.

9. Prompt Engineering

Prompt engineering is the practice of designing prompts to get better outputs from LLMs. Prompt types include zero-shot (no examples), one-shot (one example), few-shot (multiple examples), and Chain-of-Thought prompting which encourages step-by-step reasoning.

10. Text Generation and Decoding Strategies

LLMs generate text one token at a time using decoding strategies. Greedy decoding selects the most probable token. Temperature controls randomness and creativity. Top-k sampling limits choices to the top k tokens, and top-p sampling selects from tokens with cumulative probability p.

11. Limitations of LLMs

LLMs suffer from knowledge cutoffs, hallucinations (confident but incorrect answers), and weak performance in complex mathematics. They predict text based on probability and do not truly reason.

12. Augmented LLMs and RAG

Augmented LLMs overcome limitations by integrating external data and tools. Retrieval Augmented Generation (RAG) retrieves relevant information from databases or documents and injects it into the prompt, allowing the LLM to generate more accurate and up-to-date answers without retraining.

13. LLM-Powered Applications

LLMs are used in chatbots, enterprise search, customer support, code assistants, and decision systems. Frameworks such as LangChain, LlamaIndex, and OpenAI Assistants help orchestrate these applications.

14. Generative AI Project Lifecycle

The lifecycle includes problem definition, data preparation, model selection, prompting or fine-tuning, evaluation, deployment, and continuous monitoring and improvement.