6 Week Course Syllabus of Gen AI

AI & ML Overview

- Historical Context
- Fundamental Concepts of AI, ML, Neural Networks, Gen AI
- State of AI/ML In 2024 and trends for the next decade
- Applications of AI at work

LLMs

- The Rise of LLMs

- Training LLMs

□ Providing input Text
□ Optimizing Model Weights
□ Fine-tuning Parameter Values

- Model Architecture of an LLM

- Datasets for LLM

- Learning Models of an LLM

□ Zero-shot Learning
□ Few-shot Learning.
□ Domain Adaptation

- Domain Adaptation Methods

□ Domain specific Pre Training
□ Domain specific Fine Tuning
□ Retrieval Augmented Generation (RAG)
□ Choosing between the 3 models

LLM Real World Use Cases

- Content generation
- Question answering and chatbots
- Content moderation
- Language translation
- Text summarization
- Information retrieval
- Educational tools

Fine Tuning LLMs

- Why to Finetune LLMs

- Types of Fine-Tuning

□ Unsupervised Full Fine-Tuning
□ Contrastive Learning
□ Supervised Fine-Tuning methods.

- Parameter- Efficient Fine Tuning
- Supervised Full Fine-Tuning
- Instruction Fine-Tuning
- Reinforcement learning from Human Feedback (RLHF)

□ Instruction Fine Tuning

- Pretrain Finetuning
- Prompting (GPT-3)
- Instruction tuning (FLAN)

□ Direct Preference Optimization
□ Parameter Efficient Fine Tuning

- Applications of AI at work

Basics of Prompting

- Basics of Prompt Engineering

- Prompting Basics

- Advanced prompting techniques

□ Chain of Thought (CoT)
□ Tree of Thought (ToT)
□ Graph of Thought (GoT)

Retrieval Augmented Generation (RAG)

- Ingestion, Retrieval, Synthesis

- History of RAG

- Ingestion

□ Chunking
□ Embedding
□ Indexing

- Retrieval

□ Query

□ Query conversion
□ Vector comparison
□ Top-K retrieval
□ Data retrieval

- Ingestion

□ Chunking
□ Embedding
□ Indexing

- Synthesis

- RAG Challenges

□ Data ingestion complexity
□ Efficient Embedding
□ Vector Database Considerations
□ Fine-Tuning and Generalization
□ Hybrid Parametric and Non-parametric Memory
□ Knowledge update Mechanisms

- Improving RAG (Ingestion)

□ Better chunking strategies

- Content-based chunking
- Sentence Chunking
- Recursive Chunking

□ Better Indexing Strategies

- Detailed Indexing
- Question-based Indexing
- Optimized Indexing with chunk summaries

- Improving RAG components (Retrieval)

□ Hypothetical Questions and HyDE
□ Context Enrichment
□ Fusion Retrieval or Hybrid Search
□ Reranking & Filtering
□ Query Transformation and Routing

- Improving RAG (Generation)

□ Response Synthesis Approaches
□ Encoder and LLM Fine Tuning

Tools for Building LLM Applications

- Types of LLM applications

□ Custom Model Adaptation
□ RAG-based Applications.

- Types of Tools

□ Input Processing Tools
□ LLM Development Tools
□ Output Tools
□ Application Tools

- RAG

- Data Sources/Pipelines

□ Databricks
□ Airflow
□ Airbyte
□ AWS/GCP/Azure
□ Notion
□ Motherduck

- Vector Databases

□ Pinecone
□ Weavite
□ ChromaDB
□ Faiss
□ PgVector
□ Momento

- LLM Models

□ OpenAI
□ Anthropic
□ Cohere
□ Gemini
□ Hugging face (Source of open models)

- Hosting

□ Streamlit
□ Streamship
□ OctoML
□ Huggingface
□ Modal
□ Replicate

□ Amazon Bedrock

- Orchestration

□ Langchain
□ LlamaIndex
□ Anarchy
□ Fixie
□ LMQL

- Compute/Training Frameworks

□ AWS/GCP/Azure
□ Foundry
□ Lambda
□ Mosaic ML
□ Anyscale
□ Fireworks.ai
□ Training – PyTorch, TensorFlow

- Monitoring

□ Robust Intelligence
□ Gantry
□ Arthur
□ Arize
□ WhyLabs
□ Datadog
□ Helicone

LLM Application Stages (Project Management) (LLMOps)

- Pre-development and planning
- Data preparation and analysis
- Model development and training
- Optimization for deployment
- Deployment and integration
- Post-deployment monitoring and maintenance
- Continuous improvement and compliance

Deployment of LLMs

- Choice between external providers and self-hosting
- System design and scalability
- Monitoring and observability
- Cost management
- Data privacy and security
- Rapid iteration and flexibility

- Infrastructure as code
- Model composition and task composability
- Hardware and resource optimization
- Legal and ethical considerations

Monitoring and Observability

- Basic Monitoring Strategies

□ User- Facing Performance Metrics

- Latency
- Availability
- Error Rates

□ Model Outputs

- Accuracy
- Confidence Sources
- Aggregate Metrics

□ Data Inputs

- Logging Queries
- Traceability

□ Resource Utilization

- Compute Usage
- Memory Usage

□ Data Drift

- Statistical Analysis
- Detection Mechanisms

□ Custom Metrics

- Application-specific KPIs
- Innovation Tracking

- Advanced Monitoring Strategies

□ Real Time Monitoring
□ Data Drift Detection
□ Scalability and Performance
□ Interpretability and Debugging
□ Bias Detection and Fairness
□ Compliance Practices

- Security & Compliance for LLMs

□ Data Security
□ Model Security
□ Infrastructure Security
□ Ethical Considerations
□ GDPR and EU AI Act
□ International Data Protection laws