



Training 4 Programmers

To discuss this course and customizations:

Call: +1 434-509-5680 or Email: sales@training4programmers.com

Generative AI and LLMs for Python Programmers

Duration

5 days

Description

This comprehensive course provides a deep dive into Generative AI and Large Language Models (LLMs). It covers the evolution of text generation techniques, the architecture of Transformer models, and practical skills in text generation, prompt engineering, and generative configuration. The course guides you through the entire Generative AI project lifecycle, from planning to deployment. It also explores advanced techniques in pre-training LLMs, domain adaptation, fine-tuning, and model evaluation. You'll learn about Parameter-Efficient Fine-Tuning (PEFT) and techniques for aligning models with human values. The course concludes with a discussion on the ethical considerations, biases, privacy, and security concerns in Generative AI, and how to develop responsible AI practices.

Objectives

- Understand the fundamentals of Generative AI and Large Language Models (LLMs).
- Learn about the evolution of text generation techniques, from N-grams and RNNs to Transformer models.
- Master the architecture and mechanisms of Transformer models, including attention mechanisms and positional encoding.
- Gain practical skills in text generation with Transformers, prompt engineering, and generative configuration.
- Navigate the Generative AI project lifecycle, from planning and scoping to deployment and monitoring.
- Explore advanced techniques in pre-training LLMs, domain adaptation, fine-tuning, and model evaluation.
- Learn about Parameter-Efficient Fine-Tuning (PEFT) and techniques for aligning models with human values.
- Understand the ethical considerations, biases, privacy, and security concerns in Generative AI and develop responsible AI practices.

Prerequisites

No generative AI experience is required. Experience with Python is required.



Training 4 Programmers

To discuss this course and customizations:

Call: +1 434-509-5680 or Email: sales@training4programmers.com

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Student will need permission to install Docker Desktop, Python, Visual Studio Code, and Visual Studio Code Extensions on their computers.

Outline

- Introduction to Generative AI & LLMs
 - Overview of Generative AI
 - Introduction to Large Language Models (LLMs)
 - Historical Perspective on Text Generation
 - Use Cases and Tasks for LLMs
- Text Generation before Transformers
 - N-grams and Statistical Language Models
 - Recurrent Neural Networks (RNNs)
 - Long Short-Term Memory (LSTM) Networks
 - Limitations of Pre-Transformer Models
- Transformer Architecture
 - Introduction to Transformer Models
 - Attention Mechanism
 - Encoder-Decoder Architecture
 - Self-Attention and Multi-Head Attention
 - Positional Encoding
- Generating Text with Transformers
 - Text Generation Techniques
 - Beam Search, Sampling, and Top-k/Top-p Sampling
 - Practical Examples of Text Generation
- Prompting and Prompt Engineering
 - Introduction to Prompt Engineering
 - Designing Effective Prompts
 - Techniques for Prompt Optimization
 - Examples and Best Practices



Training 4 Programmers

To discuss this course and customizations:

Call: +1 434-509-5680 or Email: sales@training4programmers.com

- Generative Configuration
 - Model Hyperparameters
 - Training Configurations
 - Inference Configurations
 - Fine-Tuning Configurations
- Generative AI Project Lifecycle
 - Project Planning and Scoping
 - Data Collection and Preprocessing
 - Model Selection and Training
 - Evaluation and Iteration
 - Deployment and Monitoring
- Pre-training Large Language Models
 - Pre-training Objectives
 - Datasets for Pre-training
 - Computational Challenges
 - Scaling Laws and Compute-Optimal Models
- Domain Adaptation and Fine-Tuning
 - Domain Adaptation Techniques
 - Instruction Fine-Tuning
 - Fine-Tuning on a Single Task
 - Multi-Task Instruction Fine-Tuning
- Model Evaluation and Benchmarks
 - Evaluation Metrics for LLMs
 - Standard Benchmarks
 - Evaluating Model Performance
 - Parameter-Efficient Fine-Tuning (PEFT)
- Parameter Efficient Fine-Tuning (PEFT)
 - Introduction to PEFT
 - PEFT Techniques 1: LoRA (Low-Rank Adaptation)
 - PEFT Techniques 2: Soft Prompts
- Aligning Models with Human Values
 - Introduction to Model Alignment
 - Reinforcement Learning from Human Feedback (RLHF)
 - Obtaining Feedback from Humans
 - Reward Model and Fine-Tuning with Reinforcement Learning
 - Addressing Reward Hacking



Training 4 Programmers

To discuss this course and customizations:

Call: +1 434-509-5680 or Email: sales@training4programmers.com

- Scaling Human Feedback
- Model Optimizations for Deployment
 - Model Compression Techniques
 - Quantization and Pruning
 - Optimizing Inference Performance
 - Deployment Strategies
- Generative AI Project Lifecycle Cheat Sheet
 - Quick Reference Guide for Project Lifecycle
 - Key Steps and Best Practices
 - Common Pitfalls and Solutions
- Using the LLM in Applications
 - Integrating LLMs into Applications
 - Interacting with External Applications
 - Helping LLMs Reason and Plan with Chain-of-Thought
- Advanced Techniques and Applications
 - Program-Aided Language Models (PAL)
 - ReAct: Combining Reasoning and Action
 - LLM Application Architectures
- Responsible AI
 - Ethical Considerations in Generative AI
 - Bias and Fairness in LLMs
 - Privacy and Security Concerns
 - Developing Responsible AI Practices
- Conclusion
 - Recap of Key Concepts
 - Q&A Session
 - Next Steps and Future Trends