



Open Source Generative AI

- 4-Day Class
- Hands-on labs

Course Overview

Learn how to write practical AI applications via hands-on labs. You will design and develop Transformer models, ensuring data security in your work. The course covers AI transformer architectures, Python programming, hardware requirements, training techniques, and AI tasks like classification and regression. It includes hands-on exercises with open-source LLM frameworks, advanced topics like fine-tuning and quantization, and offers AI certification from Alta3 Research. Ideal for Python Developers, DevSecOps Engineers, and Managers or Directors, the course requires basic Python skills and provides access to a GPU-accelerated server for practical experience.

Review this course online at <https://www.alta3.com/courses/ai-gpu>

Who Should Attend

- Project Managers
- Architects
- Developers
- Data Acquisition Specialists

What You'll Learn










- Understand AI architecture, specifically the Transformer model.
- Describe the role of tokenization and word embeddings in AI processing.
- Train and optimize a Transformer model using PyTorch.
- Master advanced prompt engineering for model control.
- Install and use AI frameworks like Llama-2.
- Explore model quantization and fine-tuning.
- Understand different AI interaction modes (chat vs. instruct).
- Maximize AI Model Performance
- Compare hardware acceleration options (CPU vs. GPU).
- Apply strategies to maximize AI model performance.
- Write a real world AI Web Application

Outline






Deep Learning Intro

- 🗣️ Lecture: What is Intelligence?
- 🗣️ Lecture: Generative AI Unveiled
- 🗣️ Lecture: The Transformer Model
- 🗣️ Lecture: Feed Forward Neural Networks
- 📖 Lecture + Lab: Tokenization
- 📖 Lecture + Lab: Word embeddings
- 📖 Lecture + Lab: Positional Encoding




Build a Transformer Model from Scratch

-  Lecture: PyTorch
-  Lecture + Lab: Construct a Tensor from a Dataset
-  Lecture + Lab: Orchestrate Tensors in Blocks and Batches
-  Lecture + Lab: Initialize PyTorch Generator Function
-  Lecture + Lab: Train the Transformer Model
-  Lecture + Lab: Apply Positional Encoding and Self-Attention
-  Lecture + Lab: Attach the Feed Forward Neural Network
-  Lecture + Lab: Build the Decoder Block
-  Lecture + Lab: Transformer Model as Code













Prompt Engineering

-  Lecture: Introduction to Prompt Engineering
-  Lecture + Lab: Getting Started with Gemini
-  Lecture + Lab: Developing Basic Prompts
-  Lecture + Lab: Intermediate Prompts: Define Task/Inputs/Outputs/Constraints/Style
-  Lecture + Lab: Advanced Prompts: Chaining, Set Role, Feedback, Examples

Hardware requirements

-  Lecture: GPUs role in AI performance (CPU vs GPU)
-  Lecture: Current GPUs and cost vs value
-  Lecture: Tensorcore vs older GPU architectures

Pre-trained LLM

-  Lecture: A History of Neural Network Architectures
-  Lecture: Introduction to the LLaMa.cpp Interface
-  Lecture: Preparing A100 for Server Operations
-  Lecture + Lab: Operate LLaMa2 Models with LLaMa.cpp
-  Lecture + Lab: Selecting Quantization Level to Meet Performance and Perplexity Requirements
-  Lecture: Running the llama.cpp Package
-  Lecture + Lab: Llama interactive mode
-  Lecture + Lab: Persistent Context with Llama
-  Lecture + Lab: Constraining Output with Grammars
-  Lecture + Lab: Deploy Llama API Server
-  Lecture + Lab: Develop LLaMa Client Application
-  Lecture + Lab: Write a Real-World AI Application using the Llama API

Fine Tuning

-  Lecture + Lab: Using PyTorch to fine tune models
-  Lecture + Lab: Advanced Prompt Engineering Techniques

Testing and Pushing Limits

-  Lecture + Lab: Maximizing Model Limits

Prerequisites

- Python - PCEP Certification or Equivalent Experience
- Familiarity with Linux