+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 11/02/2026

Sign Up

DeepInfer Training

3 days (21 hours)

Overview

DeepInfer is DeepSeek's framework dedicated to high-performance inference of its models. With its optimized kernels and mechanisms for parallelism, structured generation, and speculative decoding, it provides the essential building blocks for studying and optimizing the execution of large-scale models.

Our DeepSeek DeepInfer training course will help you understand how the engine works and get started with its offline inference environment, distinguishing its core features from the additional components required for a full-fledged service.

You'll learn how to set up a compatible environment, run a supported model, configure generation parameters, and produce structured outputs. You'll then explore parallelization strategies and key optimization techniques.

By the end of the training, you'll be able to build a reproducible benchmark protocol, compare multiple configurations, and interpret the trade-offs between throughput, latency, and memory consumption to guide your technical decisions.

This training also covers non-regression testing and operational requirements. The workshops will help you identify the engine's limitations, document your results, and design a service architecture tailored to your use cases.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Identify the scope of DeepInfer and the components required for a complete inference service
- Configure a compatible environment and perform offline inference on a suitable infrastructure
- Configure output generation and validate structured outputs according to an expected schema
- Compare the engine's parallelism strategies and acceleration mechanisms
- Measure performance and verify non-regression using a reproducible protocol
- Define the technical and operational requirements for a service architecture built around the engine

Target Audience

- Machine Learning Engineer
- MLOps Engineer
- AI Developer
- AI Solutions Architect

Prerequisites

- Experience with Python and managing virtual environments
- Knowledge of tensors and model execution with PyTorch for inference
- Understanding of language models, tokenization, and generation principles
- Comfort working with the Linux terminal and common Git command-line operations

Technical Prerequisites

- Recommended workstation with 16 GB of RAM, a quad-core processor, and an SSH client to access the remote environment
- Access granted prior to the session to a remote Linux infrastructure equipped with an NVIDIA SM100, with memory and storage scaled for the model and exercises
- Remote environment with Python 3.12 and GCC 13 or higher available for compilation
- NVIDIA stack including CUDA 13 and compatible drivers, with repository dependencies installed in an isolated environment
- Stable internet connection allowing for remote access and necessary downloads, with Git available and model weights prepared before the workshops

Our DeepInfer Training Program

[Day 1 - Morning]

Architecture and Runtime Environment

- Scope of DeepInfer, support for DeepSeek-V4.1-Flash, and the distinction between the offline engine and the full service (raw.githubusercontent.com)

- Language model inference cycle: tokenization, prefill, decode, and key-value memory
- Overview of the Mixture of Experts architecture and identification of computational, memory, and communication constraints
- Setting up the Linux environment, checking dependencies, and locating build scripts and examples
- Hands-on workshop: Verifying the remote environment, setting up an isolated environment, and checking the
availability of the model and accelerators.

[Day 1 - Afternoon]

Initial inferences and structured outputs

- Loading the checkpoint from a local directory and running the offline inference example
- Reading request and response objects, managing tokens, and interpreting execution
- Adjusting sampling parameters, generation lengths, and termination conditions
- Generation constrained by a JSON schema, regular expression, or EBNF grammar, and validation of outputs (github.com)
- Hands-on workshop: Adapt an example to extract information in a specified JSON format and verifying the results using a set of queries.

[Day 2 - Morning]

Parallelism and expert balancing

- Attention distribution using Data Parallelism and Context Parallelism, followed by expert distribution using Expert Parallelism (github.com)
- Reviewing EP, DP, and CP parameters in the examples and verifying their consistency
- Role of FlashMLA, DeepGEMM, and DeepEP in attention, matrix computation, and communications
- Expert balancing with EPLB, redundant experts, and distinction from requests between replicas
- Hands-on workshop: Compare two parallelism configurations compatible with the training cluster and analyze their load and memory usage logs.

[Day 2 - Afternoon] Acceleration

and Reproducibility

- Principles of DSpark, token proposal, and trust-based scheduled verification (raw.githubusercontent.com)
- Adjusting proposal length and interpreting the acceptance rate based on queries
- How Bounded Replay SWA Works and Constraints on Context Suffix Preservation

- The Role of Asynchronous Scheduling in Overlapping CPU Work and Accelerator Execution
- Conditions for batch invariance and deterministic sampling on NVIDIA, with a fixed seed and an adapted configuration
- Hands-on Workshop: Compare an execution with and without speculative decoding, then verify the reproducibility of a run in a deterministic configuration.

[Day 3 - Morning]

Benchmarks and validation of results

- Building a representative measurement set: context sizes, query volumes, and output lengths
- Separate measurement of prefill and decode, warm-up, and data collection without profiler overhead
(raw.githubusercontent.com)
- Reading GPU throughput and steps per second, distinguishing between total accepted counts and tokens actually accepted
- Analysis of trade-offs between throughput, latency, batch size, and memory consumption without excessive extrapolation to full-scale service
- Validation of structured outputs, comparison of perplexity benchmarks, and preparation of non-regression tests
- Hands-on workshop: Run a benchmark campaign, document configurations, and produce a comparative table accompanied by non-regression checks.

[Day 3 - Afternoon]

Service architecture and operational requirements

- Identifying components to add: KV cache system, orchestration of prefill and decode disaggregation, load balancing among replicas
- Positioning of deepseek-recipe for converting protocols and
, without confusing it with an API server (raw.githubusercontent.com)
- Designing the service layer: authentication, quotas, input validation, and data protection
- Definition of observability requirements, memory and latency thresholds, timeouts, and recovery procedures
- Preparation of a deployment package: release branches, capacity, acceptance criteria, and known limitations
- Hands-on workshop: Design a target architecture around DeepInfer and present a validation dossier based on the measurements and tests performed.

Target Audience

This training is intended for both individuals and companies—large or small—that wish to train their teams in a new, advanced IT technology or to acquire specific business knowledge or modern methods.

Assessment upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency level across various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth operation of the training session.

Teaching Methods

Hands-On Training: 60% Hands-On, 40% Theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.

Training organization registered under number 11 75 54743 75. This registration does not constitute state accreditation.
© Ambient IT 2015-2026. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg