

Updated on 10/01/2026

Sign Up

Advanced PyTorch Training

3 days (21 hours)

Overview

PyTorch provides advanced tools to optimize the training and inference of deep learning models, efficiently utilize GPUs, and distribute computations across infrastructure suited for large-scale models.

Our Advanced PyTorch Training: Optimization and Industrialization will help you transition from a functional model to high-performance training that meets production requirements.

You'll learn to profile CPU and GPU workloads, analyze memory usage, and identify bottlenecks. You'll use `torch.compile` and `TorchInductor` to accelerate your models and diagnose graph breaks and recompilations.

You'll implement Mixed Precision Training and optimize your data pipelines before distributing your training across multiple GPUs using DDP and FSDP.

The training also covers quantization, pruning, and inference optimization. Finally, you'll learn how to export your models using `torch.export` and ONNX and integrate their monitoring into an MLOps workflow.

By the end of this course, you'll be able to profile, accelerate, distribute, and deploy PyTorch models while managing the trade-offs between performance, memory, accuracy, and cost.

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

Objectives

- Profile and diagnose CPU and GPU performance
- Accelerate models using `torch.compile` and TorchInductor
- Optimize memory, DataLoaders, and mixed-precision training
- Distribute training with DDP and FSDP
- Optimize inference with quantization and pruning
- Export and deploy models using `torch.export` and ONNX

Target Audience

- Machine Learning Engineers
- AI Engineers
- Experienced Data Scientists
- MLOps Engineers

Prerequisites

- Proficiency in Python and PyTorch
- Experience in designing and training neural networks
- Proficiency with Tensors, Autograd, Dataset, and DataLoader
- Knowledge of deep learning principles

Software Requirements

- At least 8 GB of RAM; 16 GB recommended
- Access to a CUDA-compatible GPU or a GPU cloud environment
- Python environment with Jupyter Notebook or VS Code
- Access to multiple GPUs or infrastructure suitable for the DDP and FSDP workshops

Agenda for our Advanced PyTorch Training Course

[Day 1 - Morning]

Profiling and diagnosing PyTorch performance

- Analyzing CPU and GPU performance with PyTorch Profiler
- Identify bottlenecks related to computation, data transfers, and I/O
- Analyze GPU memory usage and allocation
- Diagnose issues with gradients, synchronization, and computationally expensive operations
- Use hooks and debugging tools to inspect model execution
- Hands-on workshop: profile a pre-trained model, identify its main bottlenecks, and propose an optimization plan.

[Day 1 - Afternoon]

Accelerate models with torch.compile and TorchInductor

- Understand how torch.compile works and how PyTorch graphs are compiled
- Discover the role of TorchInductor in optimizing execution
- Identify and diagnose graph breaks and recompilations
- Compare different compilation modes based on performance constraints
- Measure compilation gains during training and inference
- Hands-on workshop: Compile a model, analyze graph breaks, and measure the gains achieved compared to eager execution.

[Day 2 - Morning]

Optimizing training, memory, and data pipelines

- Implement Mixed Precision Training to reduce memory consumption and speed up computations
- Understand numerical formats and balance precision, stability, and performance
- Optimize Dataset and DataLoader: workers, prefetching, pinned memory, and transfers to the GPU
- Reduce unnecessary copies, synchronizations, and memory allocations
- Diagnose gradient issues and stabilize training for complex models
- Hands-on workshop: Optimizing a training pipeline by combining mixed precision, DataLoader, and memory management.

[Day 2 - Afternoon]

Distributing training with DDP and FSDP

- Understand multi-GPU training architectures and communication costs
- Distributing training with DistributedDataParallel (DDP)
- Understanding parameter, gradient, and optimizer state sharding with FSDP
- Choose between DDP and FSDP based on model size, available memory, and infrastructure
- Diagnosing synchronization, memory, and performance issues in distributed training
- Hands-on workshop: Adapting a PyTorch training job for multi-GPU, then comparing a DDP and FSDP strategies.

[Day 3 - Morning]

Optimizing models and inference

- Leveraging advanced pre-trained models with PyTorch and Hugging Face
- Applying quantization techniques to reduce memory usage and speed up inference
- Use pruning to reduce model complexity

- Optimize batches, memory, and latency based on inference constraints
- Evaluate trade-offs between accuracy, throughput, latency, memory, and cost
- Hands-on workshop: Quantify and optimize a pre-trained model, then compare its performance before and after optimization.

[Day 3 - Afternoon]

Export, monitor, and deploy PyTorch models

- Export models using `torch.export` to prepare them for execution outside the training environment
- Export and validate models using ONNX and verify the consistency of results
- Prepare models for integration into a production inference pipeline
- Version control for models, configurations, metrics, and experiments
- Integrate monitoring, reproducibility, and MLOps practices into the PyTorch lifecycle
- Hands-on workshop: Optimize and export a model, validate its inference, and then build a workflow for production deployment and performance monitoring.

Target Audience

This training is intended for both individuals and companies—large or small—seeking to train their teams in a new advanced IT technology or to acquire specific domain knowledge or modern methodologies.

Pre-training Assessment

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 in 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 running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. 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.