

Mastering LLM Fine-Tuning

From Systems to Services Equip your team to build, fine-tune, deploy, and maintain open-source Qwen models in production-quality environments - 2 days

FA-0309 · 2 days

Agentic & Generative AI · AI for Leaders & Business · DevOps, Cloud & Infrastructure

Introduction

Universities can't afford to sit on the sidelines when it comes to AI research—and that means getting serious about open-source LLMs like Qwen. This 2-day intensive is designed for your Linux operations teams who need to get their hands dirty with the real infrastructure work: setting up GPU servers, handling distributed training, fine-tuning performance, and keeping model endpoints running smoothly in production.

We skip the academic fluff and dive straight into what actually matters when you're trying to keep these systems running day after day. Your team will walk away knowing how to build infrastructure that won't buckle under pressure—systems that are built to scale and locked down tight. Because when researchers are counting on your setup to power their breakthroughs, "good enough" just isn't good enough.

Learning outcomes

By course end, participants will be able to:

- Architect and deploy GPU-based Linux servers optimized for Qwen fine-tuning and inference
- Configure and run full-parameter, LoRA, and Q-LoRA fine-tuning pipelines
- Handle large dataset preparation, gradient checkpointing, and mixed-precision training
- Use distributed frameworks (DeepSpeed, FSDP, Hivemind) efficiently
- Deploy models to inference endpoints and monitor service health
- Implement Python scripts/notebooks for orchestration, automation, logging and scheduling

Prerequisites

Participants must have:

- Expertise in Linux server operations, virtualization, containerization (Docker/VMs)
- Basic Python skills: scripting, environment tools, pip/conda
- Familiarity with Git, SSH, networking fundamentals
- Access to GPU-enabled servers (single or multi-node), with CUDA support

Training outline

1. 1. Qwen Foundations & Infrastructure

- Overview of Qwen model family (Qwen-1.5 → Qwen-3), architectures, versions, licensing
- Hardware sizing: VRAM, RAM, disk considerations (16+ GB VRAM, 32+ GB RAM, 100+ GB storage)
- Linux environment setup: drivers, CUDA, Docker, Python, package management, GPU monitoring

2. 2. Model Retrieval & Environment Setup

- Using HF/Alibaba model hubs and official Qwen GitHub repo
- Python environment: installing Transformers, PEFT, DeepSpeed, FSDP, accelerate, bitsandbytes, unsloth

3. 3. Fine-Tuning Approaches

- Theory and comparative pros/cons: full-parameter tuning, LoRA, Q-LoRA
- Gradient checkpointing mechanics (ensuring compatibility with transformers 4.32)
- Mixed precision (bf16/fp16) logic and server implications

4. 4. Pipeline Walkthrough: Example Runs

- Single GPU fine-tuning using LoRA and Q-LoRA (shell scripts + Python orchestration)
- Multi-GPU multi-node with DeepSpeed ZeRO (levels 2 & 3), FSDP
- Distributed approaches: Hivemind + TorchTune for multi-machine setups

5. 5. Dataset & Training Management

- Preparing JSONL/chat-ML datasets for SFT

- Tokenization, prompt formatting, chain-of-thought strategies
- Data pipelines in Python: preprocessing, sharding, augmentation

6. 6. Fine-Tuning in Action

- Live walkthrough: fine-tuning Qwen3-32B-instruct on a niche dataset (e.g., medical reasoning)
- Monitoring resource usage, diagnosing OOMs, adjusting batch/seq lengths
- Best practices: checkpoints, logging, hyperparameter tracking

7. 7. Optimization & Quantization

- Post-training quantization: AWQ, GPTQ, GGUF, auto-gptq
- Merging LoRA adapters and tokenizer alignment to create fused ready-to-deploy models

8. 8. Deployment & Service Setup

- Exporting models for vLLM, FastAPI, Triton, or TensorRT endpoints
- Building Dockerized inference services with GPU support
- Orchestration: systemd, Docker Compose / Kubernetes pod configs
- Logging, monitoring, autoscaling and GPU utilization dashboards

9. 9. Production & Maintenance

- Versioning models and rolling out new deployments
- Automatic retraining triggers: cron jobs, data drifts, performance QC
- Data privacy, access control, licensing compliance
- Team workflows: division between ops and code, CI/CD integration

10. 10. Capstone & Q&A

- Wrap-up: review best practices (hardware, frameworks, automation, monitoring)
- Capstone: define a fine-tuning + deployment plan for a university use case
- Instructor-led live troubleshooting of real scenarios

This intensive course equips your Linux-savvy team with a robust systems-level command over Qwen-based LLM pipelines—fine-tuning, optimizing, deploying, and maintaining models at production scale. Leveraging proven industry frameworks and backed by an instructor with 30 years' experience, your university will build a self-sufficient AI infrastructure using real-world workflows.

