Aurojit Ghosh

**8/29/2025**

**Report: Week 3**

**Live Lecture 22 – LLM Fine Tuning Process (Practical)**

**In this video some important points covered are**

* **7-step Recipe**
* **Hyperparameters**
* **Simple terms of model**
* **Technical Insight**
* **Lora\_r**
* **Lora\_alpha**
* **Axolotl**
* **Hugging Face Transformers**

There is a 7-step recipe.

1. Collect Raw Data- Gather domain-specific sources.
2. Clean and Preprocess- Remove filter words, fix speaker tags, standardize formats, normalize punctuation
3. Deduplicate and Decontaminate- Remove near duplicate examples and anything overlapping with test sets
4. Annotate Instructions & Outputs- Convert raw input into instruction-output pairs using manual annotation or LLM-based generation
5. Add Diversity & Complexity- Vary task types, tones, lengths, and difficulty to ensure the model learns broadly. Use tools like Augmenttoolkit or Distilabel to generate paraphrases, more complex examples, or new tasks.
6. Evaluate Quality- Run checks (Automated + manual) to catch missing fields, inconsistent tone, or poor outputs. Optionally use LLM as a judge for scoring.
7. Iterate- Fine-tone a small model, find weak points, expand or fix the dataset, and repeat until results stabilize

Some key takeaways are that the model can only learn what your data teaches it. Good fine-tuning does mean it's more data. Most real-world effort is devoted to cleaning, annotation, and quality evaluation, rather than just volume.

Hyperparameters are the instructions you give:

1. How long to train them (training steps)
2. How fast they should learn (learning rate)
3. How big their memory is (context length)
4. How often they practice (batch size)
5. learning-rate

       This controls how fast the model updates its internal understanding during training.

In simple terms: If you're training the model to understand your brand's Instagram voice, should it make big leaps after seeing each example (high learning rate) or minor, careful adjustments( learning rate)?

Technical insights include both high and low learning rates. A too high learning rate results in unstable training and causes the model to forget general knowledge. A too low learning rate results in painfully slow learning and may not capture the time effectively.

num\_train\_epochs controls the duration of training the model on your dataset.

In simple terms: How many times should the model go through your training examples (500 Instagram scripts)

* 1 epoch = one complete pass through the data
* More epochs/steps = more memorization, but also higher risk of overfitting

Technical insight:

* For small datasets, overfitting happens quickly. Two to three epochs are often sufficient.
* With LoRa, training is inexpensive, but excessive training results in the model copying examples rather than generalizing the style.

3. batch-size

How many samples does the model see at once before updating its weights?

In simple terms, is the model learning from one script at a time, or a group of scripts in each mini-lesson?

* A larger batch size results in more stable updates.
* A smaller batch size can lead to noisy learning, but it can be helpful if memory is limited.

Technical Insight:

* LoRA uses minimal memory, so batch sizes of 8, 16, or 32 are commonly used.

4.lora\_r and lora\_alpha (specific to LoRA)

These are LoRA-specific hyperparameters that control the operation of the LoRA adapters.

lora\_r:

* The rank of the low-rate matrix used to inject learnable parameters.
* Think of it as how much capacity you're giving the LoRA module to learn
* Typical values: 4, 8, or 16

lora\_alpha:

* A scaling factor for how strongly Loft updates are applied.
* Usually set to a multiple of lora\_r, like lora\_alpha=16 if lora\_r=8

5.Cutoff\_len / max\_seq\_length/context\_length

This defines the maximum length of your input and output in tokens.

In the Instagram use case:

You might be passing the brand brief, tone, and three example scripts as input.

If your maximum length is too low, parts will get cut off, and the model won't learn properly.

* Typical values: 512, 1024, or 2848

Typical Hyperparameter Values

For LoRA/QLoRA fine-tuning on 24-48 GB VRAM, 500 data samples

Factors influence these values

* Dataset Seize
* Task Complexity
* Model Size
* VRAM/ GPV Constraints
* Response Length

Best Tools for Fine-Tuning LLMs

1. Axolotl- A highly flexible and community standard tool for fine-tuning LLMs using LoRA and QLoRA

Why it's great:

* Built on top of Transformers + PEFT + DeepSpeed
* YAML-Configurable -> no need to write Python code
* Handles both local GPU and multi-GPU training
* Supports popular models, including LLAMA, Mistral, and YI, among others.

Ideal for:

* Anyone who wants complete control over fine-tuning without reinventing scripts
* You want to test LoRA settings, load from Hugging Face, train your own on a GPU, or cloud

Axoloti- a tool designed to streamline post-training for various AI models

Features:

* Multiple model support: Train models like LLamA, Mistral, Mixtral. Pythia, and more

Compatible with Hugging Face transformers causal language models

* Training Methods: Fine-tuning, LoRA, QLoRA, GPTQ, Preference Tuning, Multimodal, Reward Modeling, and Process Reward Modeling
* Easy Configuration: Re-use a single YAML file between dataset preprocessing, training, evaluation, quantization, and inference
* Performance Optimization: Multipacking, Flash Attention Xformers, Flex Attention, Liger Kernel, Cut Cross Entropy, Sequence Parallelism, LoRA optimization, Multi-GPU training, Multinode Training
* Flexible Dataset Handling: Load from local, Hugging Face, and cloud(S3, Azure, GCP, OCI) datasets
* Cloud Ready: We ship Docker Images and also PyPI packages for use on cloud platforms and local hardware

Quick Start

Requirements:

* NVIDIA GPU
* Python 3.11
* PyTorch >= 2.6.0

Axolotl- a famous library that helps in fine-tuning various LLMs like LLama, Falcoln, and npt. It is built on the HF Transformers library and allows you to fine-tune LLMs on your dataset with a simple change to your file.

When seeking help, consider joining the Discord community for support. Check out the Examples director. Read the Debugging Guide.

Installation covers the various ways to install and set up Axolotl for your environment.

1. Requirements

* NVIDIA GPU or AMD GPU
* Python >= 3.11
* PyTorch >= 2.6.0

1. Installation Methods

* PyTorch needs to be installed before installing Axolotl in your local environment
* For Blackwell GPUs, please use Pytorch 2.7.9 and CUDA 12.8

2.1 PyPI Installation (Recommended)

Used no build isolation to detect the installed PyTorch version (if installed) in order not to clobber it, and so that we set the correct version of dependencies that are specific to the PyTorch version or other installed versions.

QuickStart- it's a guide that will walk you through your first model fine-tuning project with Axolotl

1. Quick Example- start by fine-tuning a small language model using Lora, and it uses a 1B Parameter model

Inference and Merging- Covers how to use a trained model for inference, including model loading, interacting testing, merging adapters, and common troubleshooting steps.

**Lab Session- Mastering the Art of Pitch**

**From this session, important points covered are**

* **Essentials in terms of selling**

For mastering the art of pitch, a person needs to know what the goals are in selling, what sales are, and why they are essential. Sales are a reward to the business. Some people want to be builders and designers, and one Example is selling a pen.

To position yourself for selling, there are several key factors to consider. You don't think of the science of sales. You don't think about how much thought goes into sales, how much calculation is involved in sales. There is repetition in sales and refining, which goes into sales. This is usually obvious to people. Once you say things in sales, you live up to them. Scammy sales are the worst sales. To sell something without overcoming objections, sales is the power of making people buy. It involves finding someone facing a problem, providing a solution, removing barriers, and assisting someone in making a purchasing decision. The type of sales you conduct has a lasting impact on your customers. Once you start convincing someone, begin by addressing your own needs before the person's needs. As soon as you put your customers ' needs first, that is when it sounds like a scam or resistance to talking to you.

How to sell?

1. Build trust (come as trustworthy)
2. Understand Requirements
3. Come across as an expert
4. Maintain Authority
5. Show Value by communicating
6. Create Urgency

Come across an expert

1. Fake it till you make it
2. Lie comes out eventually, so coming across as an expert involves being prepared
3. No call is perfect
4. You can only do most things right on most calls to increase the odds of success

To demonstrate value, use only show, not tell, and provide concrete examples. Discussing all that you have to offer is not enough.

To create urgency, discuss having a limited capacity to accept clients. Be prompt so they have reason to show promptness and ask clearly in advance.

Focus on who else is using that product, assume their problems, and start giving solutions.

**Networking Session- Final Capstone Project**

**Important points covered in this session**

* **Chatbills**
* **Fabric pattern as example of pitching**
* **Developing democracies**
* **Accountability Platform**
* **AI Search Background**

Chatbills involve spoken word and scenes. It starts with a problem statement and forms teams. If you have 11 queries for the capstone project, please create a support ticket. If you're trying to build your own custom problem statement, you will need to submit the form in the same project selection form. Draft document that explains what you're creating. Format is shared with the form itself. Pitch your own custom project ideas and start from a full stack, building from the base of LLM to the basics of augmented LLM. It provides agent workflows. If it's a real-world problem statement, there's a greater chance of getting selected.

One person pitched a problem statement about taking a picture of a fabric pattern and transferring it into a paper garment in Photoshop. It's a paid solution and is sent to WhatsApp. Few investors arrive in the future. Regarding high investment, you can chat with video and find the exact timestamp you need.

In many developing democracies, this speaker explained that there is a gap between citizens and their actual elected day-to-day representative. Not everyone is aware of it, and you may not be mindful of what ministers discuss and contribute. There is a disconnect between the actual voters and the representatives they elected. So, the public accountability platform adjusts parliamentary transcripts. This Example is that all Lok Sabha transcripts are uploaded to the Lok Sabha website daily whenever there are sessions. The idea is to pick up each day and, using Rag or AI, determine what the speaker is talking about. You would have 500 or 600 people in parliament. Accountability platform takes all speeches and creates a scoring mechanism. This platform would empower voters or the public.

Another person pitched about the AI search background in product management and marketing. Like giving a brand visibility, a brand name, and what questions or responses. Understanding page structuring and where it is structured. Understanding whether it's optimum for AI Scrapel.

**GitHub Training Video**

**What is Git and why do you need to know it?**

Git is most widely used version control system in the world. Version control system is a system that tracks changes to files over period of time.

You may have resumev1, resumev2, resumev3, and resumefinal saved as 4 different files.

* 1. Go to git-scm.com/download
  2. Go to windows
  3. Press click here to download
  4. Double click on downloads Git-2.44.0-64-bit.exe
  5. Click next to accept terms
  6. Name should be c:\Program Files\Git
  7. Press Next(keep pressing next after next steps are finished)
  8. Select Windows Explorer Integration with Open Git Bash here and Open Git GUI here, GIT Large File Support, Associate.git configuration files with the default text editor, Associate .sh Files to be run with Bash
  9. Select name git
  10. Select override the default branch name for new repositories and leave it main
  11. Select Git from the command line and also from 3rd party software
  12. Selected Use bundled OpenSSH
  13. Use OpenSSL Library
  14. Checkout Windows Style, Commit Unix-style line endings
  15. Use MinTV
  16. Choose Fast Forward or merge
  17. Git Credential Messages
  18. Leave Enable Unchecked
  19. Install
  20. Leaveboxesunchecked and press finish

**To start using GIT**

1. Go to terminal
2. Write git next to C:\Users