Chatting with a Junior Student All Afternoon: How Should We Learn in This Era of AI Frenzy?

I chatted with a junior student for a long time today. He asked me which WeChat official accounts I usually read and how I managed to gain such a deep understanding of AI. To be honest, I don't believe in any specific official account. I only have one core philosophy: **Ditch second-hand knowledge and read the source code directly.**

I have always felt that the most "sexy" thing about the computer world is its "Open Source Spirit." In other disciplines, the good stuff might be hidden away, but here, the top-tier code and the most cutting-edge ideas lie bare on GitHub. The cost of acquiring knowledge is extremely low, which is our greatest dividend. So, stop spending huge amounts of time watching those lengthy video lectures; the efficiency is just too low. In this era, the best way to learn is to pull down top projects from GitHub and ask AI about the parts you don't understand, line by line. Tools like **Cursor** or **Trae** are so friendly to students right now; you can totally treat them as your 24/7 private tutors. Can't understand the code? Throw it to the AI, let it explain the logic and the mathematical principles to you. This is "dimensionality reduction" style learning.

Talking about the specific roadmap, I found that many people are still spinning their wheels in traditional Machine Learning (ML). Although it's taught in junior year compulsory courses, to be honest, the industry doesn't really use that stack much anymore. For ML, I think you only need to understand the four core algorithms: **Linear Regression, Logistic Regression, K-means, and PCA**. Knowing how to use `scikit-learn` and understanding the essence of supervised vs. unsupervised learning is enough. Don't linger there.

The real heavyweight is Deep Learning (DL). I have a very important piece of advice here: **Heavy on Math, Light on Memory.** For example, you must thoroughly understand the mathematical derivation of the **Backpropagation (BP)** algorithm; that is the foundation. But for those cumbersome tensor shape transformation APIs in PyTorch, don't memorize them by rote. That's an engineering detail—easy to get dizzy, easy to forget. The modern way is to write clear comments in the code describing your desired input and output shapes, and leave the code completion to AI. What we need to master are architectural ideas, such as the Attention Mechanism in Transformers. Go read the paper *"Attention is All You Need"*, and understand why it replaced RNNs and LSTMs, rather than obsessing over the specific syntax of a single line of code.

Moving further, we have Large Language Models (LLM) and Agents. Don't be scared off by those fancy terms. Go check out Karpathy's `nanoGPT`; it is the best minimal prototype for understanding LLMs. Want to get your hands dirty? Don't force your laptop to run it. Rent a GPU server on **AutoDL**, learn to use **SSH** for remote connection, and learn to use `nohup` to keep tasks running in the background. By the way, stop using Conda for Python package management. Try `uv` written in Rust; it's blazing fast. And remember to stick to the principle of **"one project, one virtual environment"**—it will save you from countless troubles in the future.

As for Reinforcement Learning (RL), which is on fire right now, it is the key technology that allows models to surpass human data. Although the math is hard, have the patience to gnaw through the papers a few times and learn by combining them with clean codebases like `cleanrl`. You will open the door to a new world. Especially algorithms like **GRPO**, which are used by DeepSeek, are essentially the "prominent learning" of the moment. And if your ambition is to be the one "selling shovels," then the **AI Infra** direction—especially understanding the **PagedAttention** mechanism in `vllm`—is a core competency frantically sought after by major tech giants in both China and the US.

Finally, we talked about **"World Models."** This might be the sixth great realm after ML, DL, RL, LLM, and Agents. Current RL is essentially brute-force search, which is too inefficient. A World Model allows AI to "dream" and "plan" like a human—rehearsing in its mind before acting. This kind of "smart exploration" is the necessary path to Artificial General Intelligence (AGI).

After chatting all this way, thousands of words merge into one sentence: **In this era, don't be a student who just listens to lectures; be an engineer who reads source code.** All the secrets are in the code. Run it, debug it, and pair program with AI. That is the fastest shortcut.