SAPCAD: Fine-Tuning Environment Setup Guide (GPU Cluster)

Objective

This guide walks through the process of setting up a GPU-compatible Python environment on the TU Clausthal HPC cluster for LLM fine-tuning and experimentation as part of the SAPCAD project.

Directory Setup

1. Connect to the cluster via SSH:

```
ssh <username>@cloud-243.rz.tu-clausthal.de
```

2. Create and navigate to your working directory in scratch:

```
mkdir -p /scratch/<username>
cd /scratch/<username>
```

Install Miniconda

Download and install Miniconda locally inside the scratch directory:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

bash Miniconda3-latest-Linux-x86_64.sh -b -p /scratch/<username>/miniconda3
```

Shell Configuration

Switch to bash shell if necessary:

```
bash
```

Activate Conda with:

```
export PATH=/scratch/<username>/miniconda3/bin:$PATH
```

Optional (to make it permanent):

```
echo 'export PATH=/scratch/<username>/miniconda3/bin:$PATH' >> ~/.bashrc
```

Create and Activate Conda Environment

Create and activate a clean Conda environment:

```
conda create -n sapcad python=3.10 -y
conda activate sapcad
```

Install Required Packages

Install LLM and fine-tuning dependencies:

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
pip install transformers datasets peft accelerate
```

Notes and Best Practices

- Keep backend and training environments separate to avoid dependency conflicts.
- Avoid using sudo. Install only in user space (/scratch/<username>).
- Use bash shell for compatibility with Conda and Python tooling.
- Monitor GPU usage and clean up large temporary files in scratch.

Purpose of This Environment

This environment is intended exclusively for training and fine-tuning language models (e.g., LLaMA, Mistral) with domain-specific prompts. It is separate from the FastAPI-based backend environment used for user interaction.