

编程环境配置指南

核心步骤：

1. 安装 **Miniconda** (使用清华源以加速下载)
 2. 配置 **Conda** 源 (永久更换为国内镜像源)
 3. 创建并激活 **Conda** 虚拟环境 (可跳过)
 4. 安装推荐的技术栈
-

步骤 1: 安装 Miniconda

Miniconda 是一个轻量级的 Conda 安装程序，只包含 Conda、Python 和一些必需的包。

1. 访问清华大学开源软件镜像站：

- 在浏览器中打开: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/>

2. 下载适合您系统的安装包：

- **Windows:** 下载最新的 **Miniconda3-py311_xxxx-Windows-x86_64.exe** 文件。(py311 代表默认安装 Python 3.11)
- **macOS:**
 - **Apple Silicon (M1/M2/M3):** 下载 **Miniconda3-py311_xxxx-MacOSX-arm64.sh**。
 - **Intel Mac:** 下载 **Miniconda3-py311_xxxx-MacOSX-x86_64.sh**。
- **Linux:** 下载 **Miniconda3-py311_xxxx-Linux-x86_64.sh**。

3. 运行安装程序：

- **Windows:**
 - 双击下载的 **.exe** 文件。
 - 在安装过程中，推荐选择 **"Just Me"**。
 - **最重要的一步:** 当看到 "Advanced Installation Options" 时，**强烈建议勾选 "Add Miniconda3 to my PATH environment variable"**。虽然安装程序会提示不推荐，但这会让后续在 CMD 或 PowerShell 中使用 **conda** 命令变得非常方便。
- **macOS / Linux:**
 - 打开终端 (Terminal)。
 - 进入你下载文件所在的目录 (通常是 **~/Downloads**)。

```
cd ~/Downloads
```

- 运行安装脚本 (请将文件名替换为您下载的实际文件名):

```
bash Miniconda3-py311_xxxx-Linux-x86_64.sh
```

- 按照屏幕提示操作。一直按回车键阅读许可协议，然后输入 **yes** 同意。
- 当询问安装位置时，直接按回车使用默认路径即可。
- 当最后询问 “Do you wish the installer to initialize Miniconda3 by running conda init?” 时，输入 **yes** 并按回车。

4. 重启终端:

- 关闭所有已打开的终端窗口，然后重新打开一个新的终端。
- 你会看到命令行提示符前面出现了 **(base)** 字样，这表示 Miniconda 安装成功，并且你正处于默认的 **base** 环境中。

步骤 2: 配置 Conda 源为国内镜像

这一步将 Conda 的默认下载地址指向清华大学的镜像服务器，可以极大地加速后续所有包的下载速度。

- 打开终端 (Windows 用户可以使用 Anaconda Prompt, 或者配置好 PATH 后的 CMD/PowerShell)。
- 依次执行以下命令：

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/r
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/msys2
conda config --set show_channel_urls yes
```

- **配置 PyTorch 和 Conda-Forge 的源 (非常重要!):**

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
```

现在，你的 Conda 已经配置完成，可以享受飞速下载了。

可以跳过的部分:

步骤 3: 创建并激活 Conda 虚拟环境

为每个项目创建一个独立的环境是一个非常好的习惯，可以避免不同项目之间的包版本冲突。

1. 创建环境:

- 在终端中运行以下命令。我们将环境命名为 **pnn_test**，并指定使用 **python=3.11**。

```
conda create --name pnn_test python=3.11
```

- Conda 会列出将要安装的包，输入 `y` 并按回车。

2. 激活环境:

- 创建完成后，根据提示运行激活命令：

```
conda activate pnn_test
```

- 你会看到命令行提示符从 `(base)` 变成了 `(pnn_test)`，表示你已成功进入新环境。之后的所有安装操作都将在这个环境中进行。

步骤 4: 安装推荐的技术栈

现在，我们在激活的 `pnn_test` 环境中（如果跳过了第3步，可以不必激活环境），安装必要工具。

推荐使用单个命令一次性安装核心库，Conda 会自动处理复杂的依赖关系。

- 对于配备 NVIDIA GPU 的用户 (CUDA):
 - 首先，确认你的 NVIDIA 驱动版本支持 CUDA 12.9。
 - 然后运行以下命令：

```
conda install pytorch=2.3.0 torchvision=0.18.0 torchaudio=2.3.0
pytorch-cuda=12.9 numpy=1.26 scipy=1.13 pandas=2.2 matplotlib=3.8
seaborn=0.13 jupyterlab=4.1 tqdm wandb -c pytorch -c nvidia -c conda-
forge
```

- 说明: `-c` 参数指定了搜索包的渠道（channel），优先从 pytorch 和 nvidia 的官方渠道下载以确保兼容性。我们之前已经通过配置文件添加了这些源的清华镜像，所以下载速度会很快。

- 对于没有 NVIDIA GPU 或使用 Apple Silicon Mac 的用户 (CPU-only):
 - 运行以下命令：

```
conda install pytorch=2.3.0 torchvision=0.18.0 torchaudio=2.3.0
cpuonly numpy=1.26 scipy=1.13 pandas=2.2 matplotlib=3.8 seaborn=0.13
jupyterlab=4.1 tqdm wandb -c pytorch -c conda-forge
```

- 说明: `cpuonly` 告诉 Conda 安装一个不依赖 CUDA 的 PyTorch 版本。

安装完成后，你的开发环境就完全准备好了！

每天开始工作时: 打开终端，运行 `conda activate pnn_test` 进入你的项目环境（如果跳过了第3步，可以不必激活环境）。

