1.程算课的 C/CPP 环境——已配；
2.Anaconda——已安装；
3.python 虚拟环境——已创建；

```
(base) duqiu@duqiudembp ~ % conda create -n ML python=3.12 numpy matplotlib pand
as jupyter notebook
Channels:
 - defaults
Platform: osx-arm64
Collecting package metadata (repodata.json): done
Solving environment: done
```

4.CUDA 安装：
因为 PC 没有 Nvidia GPU，安装不了 CUDA；

1. M 系列芯片不支持 CUDA 的本质原因

- **硬件层面**：CUDA 是 NVIDIA 专属的并行计算框架，仅支持 NVIDIA GPU 的硬件指令集（如 CUDA Cores）。而 M 系列芯片的 GPU 是 Apple 自研的**Apple Silicon GPU**（基于 ARM 架构，集成于 SoC 中），其硬件架构与 NVIDIA GPU 完全不同，无法执行 CUDA 的底层指令。
- **软件层面**：CUDA Toolkit 的驱动和运行时库仅针对 NVIDIA GPU 编译，没有适配 Apple Silicon 的版本（NVIDIA 也未为 M 系列芯片开发 CUDA 支持）。

5.pytorch 安装：

```
(ML) (base) duqiu@duqiudembp Project % /opt/anaconda3/envs/ML/bin/python /Users/duqiu/Project/test
.py
2.6.0
True
True
(ML) (base) duqiu@duqiudembp Project %
```

6.常用包安装——已安装
7.虚拟环境必要性：
（1）依赖隔离：避免不同项目间库版本冲突；
（2）环境纯净：不污染系统全局 Python，便于团队合作；
（3）版本管理：支持多 python 版本；
8.无 conda 激活方式：
首先得到路径：

```
Last login: Fri Jul 18 16:17:03 on ttys009
(base) duqiu@duqiudembp ~ % conda activate ML
(ML) duqiu@duqiudembp ~ % echo $CONDA_PREFIX
/opt/anaconda3/envs/ML
(ML) duqiu@duqiudembp ~ %
```

（1）直接调用环境内工具：

```bash
# 运行 Python 脚本（替换为实际脚本名）
/opt/anaconda3/envs/ML/bin/python your_script.py

# 安装包（示例：安装 torch）
/opt/anaconda3/envs/ML/bin/pip install torch
```

（2）临时激活环境：

```bash
# 激活（仅需一次，后续命令直接使用环境内工具）
export PATH=/opt/anaconda3/envs/ML/bin:$PATH
export CONDA_PREFIX=/opt/anaconda3/envs/ML

# 验证：应输出环境内 Python 路径
which python  # 输出：/opt/anaconda3/envs/ML/bin/python

# 使用完毕后恢复（关闭终端或执行）
unset PATH CONDA_PREFIX
```

（3）创建一键激活脚本：

```
# 创建脚本（自动适配你的路径，无需修改）
echo 'export PATH=/opt/anaconda3/envs/ML/bin:$PATH' > ~/activate_ml.sh
echo 'export CONDA_PREFIX=/opt/anaconda3/envs/ML' >> ~/activate_ml.sh
echo 'echo "ML 环境已激活"' >> ~/activate_ml.sh

# 赋予执行权限
chmod +x ~/activate_ml.sh

# 激活环境（每次使用前执行，无需 conda 指令）
source ~/activate_ml.sh

# 验证：`python --version` 应显示环境内版本
```

9.配环境需要编译部分：

需要编译部分：

（1）硬件加速库：如 PyTorch 的 CUDA 版本

（2）C/C++拓展库：numpy 的 BLAS/LAPACK 优化，scikit-learn 的高效算法模块；

无需编译部分：
（1）纯 Python 库：flask，requests 等
（2）预编译二进制包：PyTorch 的 MPS 版本