人工智能与机器学习基础
LAB0：机器学习开发环境配置

中国科学技术大学

2026 年秋季学期　发布日期与截止日期待定

实验目标与提交要求

本实验帮助大家建立后续实验统一使用的 Python 开发环境。完成后，你应当能够使用 Git 获取课程代码，使用 Conda 管理独立环境，在 Visual Studio Code（VS Code）中选择正确的 Python 解释器，并正常导入 NumPy、scikit-learn 与 PyTorch。

本文分别给出 **Windows 10/11（64 位）**和 **macOS 11 或更高版本（Apple Silicon 或 Intel）**的完整操作。请选择自己的平台，按对应步骤从头完成；不要把两个平台的命令混在同一个终端中执行。

> 预计用时为 **60–90 分钟**。本实验必须单人完成，不要求 GPU，不要求撰写实验报告。最终只需向 OJ 提交验证脚本生成的 **64 位 SHA256 字符串**。截止前可以重复提交。

实验文件目录如下：

```
LAB0/
|-- environment.yml
|-- lab0.pdf
`-- verify_env.py
```

请不要修改 `verify_env.py`。如果程序没有生成 SHA256，应根据错误提示修复环境，而不是删除检查代码。

对作业或实验内容有疑问时，请先搜索并通过课程仓库的 GitHub Issues 提问：

https://github.com/Intelligent114/ALML_public/issues

不要在 Issue 中发布本人学号、邮箱验证码、密码、SHA256 提交值或未公开答案。

1　安装与配置 Git

Git 是一个分布式版本控制系统。本课程使用 Git 发布和更新实验代码。

Windows

打开 Git 官方 Windows 安装页面：

https://git-scm.com/install/windows

下载并运行适合机器架构的 Git for Windows 安装器，保留默认选项即可。也可以在 PowerShell 中使用 Windows Package Manager：

```
winget install --id Git.Git -e --source winget
```

安装完成后关闭并重新打开 PowerShell，运行：

```
git --version
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"
git config --list --show-origin
```

在 PowerShell 中克隆课程仓库：

```
cd $HOME
git clone https://github.com/Intelligent114/ALML_public.git
cd ALML_public
git status
git log -1
```

macOS

打开 Terminal（终端）。macOS 可以通过 Xcode Command Line Tools 安装 Apple 提供的 Git：

```
xcode-select --install
```

在系统弹窗中选择安装，完成后关闭并重新打开 Terminal，再运行：

```
git --version
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"
git config --list --show-origin
```

如果已经使用 Homebrew，也可以运行 `brew install git`；本实验不要求专门安装 Homebrew。在 Terminal 中克隆课程仓库：

```
cd "$HOME"
git clone https://github.com/Intelligent114/ALML_public.git
cd ALML_public
git status
git log -1
```

Windows 与 macOS 后续获取更新的命令都是 `git pull`。如果出现本地修改冲突，请先备份自己的文件，再在 GitHub Issues 中说明平台、文件名和完整错误，不要盲目覆盖。

2 安装 Miniforge 并创建课程环境

Conda 可以为不同项目建立相互隔离的 Python 环境。本课程使用由 conda-forge 维护的 Miniforge：

https://github.com/conda-forge/miniforge/releases/latest

Windows

下载 `Miniforge3-Windows-x86_64.exe` 并运行。建议保留“创建开始菜单快捷方式”，不要勾选把 Miniforge 添加到系统 PATH，以免与其他 Python 冲突。

安装后从开始菜单打开 **Miniforge Prompt**，执行：

```
conda --version
conda init powershell
```

关闭所有 Miniforge Prompt 和 PowerShell 窗口，再重新打开 PowerShell。进入 LAB0 目录并创建课程环境：

```
cd $HOME\ALML_public\LABs\LAB0
conda env create -f environment.yml
conda activate ai3002
python --version
where.exe python
python -c "import sys; print(sys.executable)"
```

macOS

在 Apple 菜单的“关于本机”中查看芯片类型，或者在 Terminal 运行：

```
uname -m
```

根据输出选择安装器：

- `arm64`：下载 Apple Silicon 的 `Miniforge3-MacOSX-arm64.pkg`；
- `x86_64`：下载 Intel 的 `Miniforge3-MacOSX-x86_64.pkg`。

运行签名的 PKG 安装器，并在安装选项中允许初始化当前 Shell。

安装后关闭并重新打开 Terminal，运行：

```
conda --version
```

如果提示找不到 `conda`，执行下面的命令，然后再次关闭并重新打开 Terminal：

```
~/miniforge3/bin/conda init zsh
```

进入 LAB0 目录并创建课程环境：

```
cd "$HOME/ALML_public/LABs/LAB0"
```

```
conda env create -f environment.yml
conda activate ai3002
python --version
which python
python -c "import sys; print(sys.executable)"
```

两个平台都应看到 Python 3.11，解释器路径中应包含 `ai3002`。查看、退出或删除环境的命令在 Windows PowerShell 与 macOS Terminal 中相同：

```
conda info --envs
conda deactivate
conda env remove -n ai3002
```

只有环境已经损坏且普通排错无法解决时才需要删除重建。后续每次做实验前，都应先运行 `conda activate ai3002`。

3 安装 VS Code 并选择正确解释器

VS Code 下载地址为 https://code.visualstudio.com/。安装后，在扩展面板安装 Microsoft 发布的 **Python** 扩展；Python Environments 扩展会协助发现和切换 Conda 环境。

Windows

运行 Windows 安装器时，建议启用“添加到 PATH”和“通过 Code 打开”选项。也可以在 PowerShell 中安装：

```
winget install -e --id Microsoft.VisualStudioCode
```

重新打开 PowerShell，用 VS Code 打开整个课程仓库：

```
code $HOME\ALML_public
```

按 `Ctrl+Shift+P`，选择：

`Python: Select Interpreter` ⟶ `ai3002 (Python 3.11)`

新建 VS Code 集成终端，确认提示符前有 `(ai3002)`，然后运行：

```
where.exe python
python -c "import sys; print(sys.executable)"
```

macOS

下载 macOS 版本，将 Visual Studio Code 拖入 Applications。打开 VS Code，按 `Cmd+Shift+P`；如需在 Terminal 使用 `code`，先执行 `Shell Command: Install 'code' command in PATH`。

在 Terminal 中打开整个课程仓库：

```
code "$HOME/ALML_public"
```

按 Cmd+Shift+P，选择：

Python: Select Interpreter ⟶ ai3002 (Python 3.11)

新建 VS Code 集成终端，确认提示符前有 (ai3002)，然后运行：

```
which python
python -c "import sys; print(sys.executable)"
```

两个平台最常见的问题都不是“库没有安装”，而是 VS Code 选择了另一个 Python。Windows 用 `where.exe python`，macOS 用 `which python`；输出路径必须对应 ai3002 环境。

4 检查机器学习库

environment.yml 会安装 NumPy、scikit-learn 与 PyTorch。LAB0 只检查 CPU 整数计算，不要求 CUDA，也不要求 Apple Metal/MPS。

Windows

在 VS Code 的 PowerShell 集成终端中运行：

```
conda activate ai3002
cd $HOME\ALML_public\LABs\LAB0
python -c "import numpy, sklearn, torch; print('imports ok')"
python -c "import torch; print(torch.__version__)"
```

macOS

在 VS Code 的 zsh 集成终端中运行：

```
conda activate ai3002
cd "$HOME/ALML_public/LABs/LAB0"
python -c "import numpy, sklearn, torch; print('imports ok')"
python -c "import torch; print(torch.__version__)"
```

Windows 没有 NVIDIA GPU 时，`torch.cuda.is_available()` 返回 `False` 是正常的；macOS 通常也会返回 `False`。这不影响实验通过，不要为了 LAB0 额外安装 CUDA Toolkit 或修改 PyTorch 包。

5 生成环境验证 SHA256

验证脚本会检查 Python、Conda、Git 和三个机器学习库，并将确定性的整数结果与本人学号组合后计算 SHA256。

Windows

在 VS Code 的 PowerShell 集成终端中运行：

```
cd $HOME\ALML_public\LABs\LAB0
conda activate ai3002
python verify_env.py --self-test
python verify_env.py --student-id PBXXXXXXXX
```

macOS

在 VS Code 的 zsh 集成终端中运行：

```
cd "$HOME/ALML_public/LABs/LAB0"
conda activate ai3002
python verify_env.py --self-test
python verify_env.py --student-id PBXXXXXXXX
```

两个平台正常输出的末尾都类似：

```
[PASS] Student ID: PBXXXXXXXX
[PASS] Protocol: AI3002-2026F-LAB0-v1

请将下面一行的 64 位字符串提交到 OJ:

0db3...中间省略...7a4c
```

最终 SHA256 只由协议版本、规范化后的学号和固定整数结果组成。软件版本、操作系统、安装路径、GPU 状态和 Git 姓名邮箱均不会上传，也不会影响结果。学号字母会统一转换为大写，最终 SHA256 为小写。

6 提交到 OJ

课程 OJ 地址为 https://oj.temaurinum.moe/。

Windows

使用 Edge、Chrome 或 Firefox 打开 OJ，登录后选择“LAB0：机器学习开发环境配置”。确认页面显示的学号正确，将 PowerShell 中生成的 64 位 SHA256 粘贴到输入框并提交。

macOS

使用 Safari、Chrome 或 Firefox 打开 OJ，登录后选择“LAB0：机器学习开发环境配置”。确认页面显示的学号正确，将 Terminal 中生成的 64 位 SHA256 粘贴到输入框并提交。

OJ 会使用当前登录账户中的学号重新计算期望结果，因此两个平台都应注意：

- 不需要、也不能在 OJ 中另行填写学号；

- 复制其他同学的 SHA256 无法通过；
- 如果注册账户时填错学号，请在 GitHub Issue 中只说明“账户学号需要更正”，不要公开具体学号；
- 截止前可以重复提交，以出现“环境验证通过”为完成标志。

SHA256 是本实验的完成凭证，用于发现常见环境配置错误；它不是可信硬件证明。请独立完成安装过程，因为后续正式实验会直接依赖这个环境。

7 通过 GitHub Issues 提问

所有作业与实验问题统一发布到：

https://github.com/Intelligent114/ALML_public/issues/new/choose

Windows

可以在浏览器中打开上面的地址，或在 PowerShell 中运行：

```
start https://github.com/Intelligent114/ALML_public/issues/new/choose
```

macOS

可以在浏览器中打开上面的地址，或在 Terminal 中运行：

```
open https://github.com/Intelligent114/ALML_public/issues/new/choose
```

提问前先搜索是否已有相同问题。新 Issue 应写明 HW/LAB 编号、Windows 或 macOS、系统版本、CPU 架构、执行的命令、完整错误文本和已经尝试的方法。请使用文字或代码块粘贴错误，避免只发一张无法搜索的截图。

GitHub Issue 是公开页面。不得发布真实学号、个人邮箱、验证码、密码、Cookie、SHA256 提交值、未公开答案或完整个人作业。需要私下处理的账户信息由助教在 Issue 中另行说明安全渠道。

8 常见问题与双平台排错

找不到 git

Windows

关闭并重新打开 PowerShell，运行 `where.exe git`。如果仍找不到，重新运行 Git for Windows 安装器，或执行 `winget install --id Git.Git -e --source winget`。

macOS

在 Terminal 运行 `xcode-select --install`，完成系统弹窗中的安装后重新打开 Terminal，再运行 `which git`。

找不到 conda

Windows

从开始菜单打开 Miniforge Prompt，运行 `conda init powershell`，然后关闭所有终端并重新打开 PowerShell。

macOS

在 Terminal 运行 `/miniforge3/bin/conda init zsh`，然后关闭并重新打开 Terminal。若 Miniforge 安装在其他目录，请在 GitHub Issue 中附上安装路径和 `ls $HOME` 的相关输出，不要公开用户名目录中的私人文件。

VS Code 没有使用 ai3002

Windows

按 `Ctrl+Shift+P` 重新运行 `Python: Select Interpreter`，选择 ai3002；关闭旧集成终端并新建终端，用 `where.exe python` 核对。

macOS

按 `Cmd+Shift+P` 重新运行 `Python: Select Interpreter`，选择 ai3002；关闭旧集成终端并新建终端，用 `which python` 核对。

可以导入 NumPy，但不能导入 sklearn 或 torch

Windows

```
cd $HOME\ALML_public\LABs\LAB0
conda activate ai3002
conda env update -n ai3002 -f environment.yml --prune
```

macOS

```
cd "$HOME/ALML_public/LABs/LAB0"
conda activate ai3002
conda env update -n ai3002 -f environment.yml --prune
```

OJ 提示 SHA256 不匹配

Windows

在 PowerShell 中先运行 `git pull`，再运行 `python verify_env.py --self-test` 和本人学号命令；确认所有检查均为 PASS。

macOS

在 Terminal 中先运行 `git pull`，再运行 `python verify_env.py --self-test` 和本人学号命令；确认所有检查均为 PASS。

如果仍未解决，请按上一节要求创建 GitHub Issue，但不要粘贴最终 SHA256 或本人学号。

9 完成前自检清单

提交前逐项核对；若某一项失败，先回到对应章节排查，再创建 GitHub Issue。

Windows

1. `git --version` 和 `where.exe git` 均能正常输出；
2. `conda info --envs` 中 ai3002 前有星号，`where.exe python` 的第一项属于 ai3002；
3. 在仓库的 `LABs/LAB0` 目录运行库版本检查时，NumPy、scikit-learn 和 PyTorch 均能导入；
4. `python verify_env.py --self-test` 的全部检查均为 PASS；
5. 使用本人学号生成 SHA256，且 OJ 显示“环境验证通过”。

macOS

1. `git --version` 和 `which git` 均能正常输出；
2. `conda info --envs` 中 ai3002 前有星号，`which python` 指向 ai3002；
3. 在仓库的 `LABs/LAB0` 目录运行库版本检查时，NumPy、scikit-learn 和 PyTorch 均能导入；
4. `python verify_env.py --self-test` 的全部检查均为 PASS；
5. 使用本人学号生成 SHA256，且 OJ 显示“环境验证通过”。

完成以上步骤后，请保留本地 ai3002 环境和课程仓库。后续实验只需进入仓库、执行 `git pull`、激活 ai3002，并在 VS Code 中确认解释器仍为该环境。

— End of LAB0 —