

Preliminary for AI Research

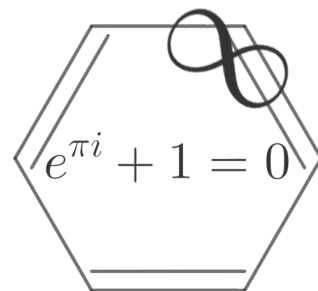
From Classroom to Research

【新手友好】AI 科研常用知识

实践中常用的编程、深度学习、机器学习、数学

作者：Xi Wang (Hytidel)

版本：2025.09.08



From Classroom to Research.

前言

本课程旨在介绍从本科生的课堂学习到 AI 科研的衔接知识，所涉及的知识大多是各 AI 领域的科研中通用的。此外，本课程还作为笔者的后续课程（敬请期待）的先修课，因此会补充介绍计算机视觉 (Computer Vision, CV) 和计算机图形学 (Computer Graphics, CG) 中需用到的知识。

本课程的主要内容如下：**5**章介绍了 AI 科研中常用的数学知识（应少尽少），它本应被放在第 1 章，但笔者知道这样会直接劝退绝大部分人，故将其放在最后一章，但前面的章节仍会引用到**5**章的知识，请读者需要时自行查阅。后续的章节也大致是难度逐渐上升的顺序安排。具体地，**1**章介绍了常用的编程知识，主要是 Python 和常用的库；**2**章介绍了常用的可视化和绘图方法，让你能亲手画出论文中的图；**3**章着重介绍和补充 AI 科研实践中常用的深度学习知识，尤其是 *PyTorch* 框架的使用，并附有代码实现；**4**章介绍了一些常用的机器学习知识，它们会在某些领域的科研中出现。此外，每章还配有简单的实验或例题作为示例和习题。

值得一提的是，本课程基本只列出了每个板块在实践（即论文、代码）中常用的基础知识 (preliminary)。作为一门主打“广度”的课，不可避免地会损失一些“深度”。若希望深入学习某个板块，我们也附上了优质的学习资源的链接，请读者自行学习。

虽然本课程的定位是“喂饭”教程，但仍假设读者有本科的基础的数学课（高等数学/数学分析、线性代数/高等代数、概率论与数理统计）、计算机课（C 语言的语法、Python 的语法、面向对象基础）和深度学习（如动手学深度学习¹）的基础。对于过于基础的知识，我们也附上了优质的学习资源的链接，请读者自行学习。

本课程的配套讲解视频为²（建议关注、一键三连），GitHub 仓库为³（建议 star）。

¹<https://courses.d2l.ai/zh-v2/>

²<https://space.bilibili.com/382329676/lists/6307119?type=season>

³https://github.com/Hyidel/preliminary_for_ai_research

目录

第一章 编程基础	1
1.1 环境	1
1.1.1 查询显卡信息	1
1.1.1.1 CUDA	1
1.1.1.2 nvidia-smi	1
1.1.1.3 nvcc -V	2
1.1.2 配置环境	3
1.1.3 找到 conda 环境的路径	3
1.2 Python 基础	5
1.2.1 if __name__ == "__main__"	5
1.2.2 工作路径	5
1.2.2.1 展示工作路径	5
1.2.2.2 切换工作路径	5
1.2.3 Jupyter Notebook	5
1.2.3.1 切换工作路径	5
1.2.3.2 注意事项	5
1.2.4 Python 基本语法	5
1.2.5 locals()、globals()、SimpleNamespace	5
1.2.6 断点、调试	5
1.2.7 generator、yield	5
1.2.8 文件读写	5
1.2.9 面向对象	5
1.2.9.1 Python 面向对象基础语法	5
1.2.9.2 __call__()	5
1.2.9.3 静态变量	5
1.2.9.4 类方法	5
1.2.9.5 抽象类、抽象方法	5
1.2.9.6 MethodType	5
1.2.10 arg、kwargs	5
1.3 常用 Python 库	5
1.3.1 数据类型	6
1.3.1.1 typing 库	6
1.3.1.2 更多数据类型	6
1.3.2 pathlib 库	6
1.3.3 itertools 库	6
1.3.4 运行	6
1.3.4.1 命令行参数	6
1.3.4.2 指定显卡	6
1.3.4.3 Hydra 库	6
1.3.4.4 Fire 库	6
1.3.5 数据处理	6

1.3.5.1 NumPy 库	6
1.3.5.2 Pandas 库	6
1.3.5.3 SciPy 库	6
1.3.6 常用文件类型的读写	6
1.3.6.1 .npd	6
1.3.6.2 .yaml	6
1.3.6.3 .pkl	6
1.3.6.4 .json	6
1.3.6.5 .csv、.tsv	6
1.3.7 进度条	6
1.3.7.1 Tqdm 库	6
1.3.8 并行	6
1.3.8.1 Multiprocessing 库	6
1.3.8.2 concurrent.futures	6
1.3.9 前端	6
1.3.9.1 Gradio 库	6
 第二章 可视化与绘图基础	 7
 第三章 深度学习基础	 8
 第四章 机器学习基础	 9
 第五章 数学基础	 10

第一章 编程基础

1.1 环境

1.1.1 查询显卡信息

1.1.1.1 CUDA

Compute Unified Device Architecture (CUDA) 是 NVIDIA 提出的并行计算平台和编程模型，可让开发者用 C/C++/Python 等语言操作 GPU。GPU 内有很多个 CUDA 核心，计算高度并行。

1.1.1.2 nvidia-smi

在终端（cmd, powershell, bash, shell 等）运行

```
1 nvidia-smi
```

输出如下：

```
1 (no_pain_diffusion) PS C:\Users\Hytidel> nvidia-smi
2 Mon Sep 8 09:07:24 2025
3 +-----+
4 | NVIDIA-SMI 566.07                Driver Version: 566.07      CUDA Version: 12.7   |
5 |-----+-----+
6 | GPU Name               Driver-Model | Bus-Id      Disp.A | Volatile Uncorr. ECC |
7 | Fan Temp   Perf        Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
8 |                               |                       |              MIG M. |
9 |=====+=====+
10 |  0 NVIDIA GeForce RTX 4060 ... WDDM | 00000000:01:00:0 Off |            N/A |
11 | N/A  50C    P8             1W /  80W |   72MiB /  8188MiB |    0%      Default |
12 |                               |                       |              N/A |
13 +-----+-----+
14
15 +-----+
16 | Processes:                                |
17 | GPU   GI   CI          PID    Type   Process name                  GPU Memory |
18 |      ID   ID                                 Usage            |
19 +-----+-----+
20 |    0   N/A  N/A       12392    C+G    D:\TencentQQ\QQ.exe           N/A       |
21 +-----+-----+
```

上方为 head 信息和 GPU 概览表，其中的重要参数：

1. Driver Version: 566.07 表示 NVIDIA 驱动器 (Driver) 版本为 566.07；
2. CUDA Version: 12.7 表示该驱动最高支持的 CUDA Runtime 版本；
3. GPU 表示显卡的编号 (0-indexed, 即编号从 0 开始)。下方对应的 0 表示本机的 0 号卡；
4. Name 表示显卡的名称，一般为显卡的型号。下方对应的 NVIDIA GeForce RTX 4060 ... 表示 0 卡是 RTX 4060，后面的 ... 是未完整显示的显卡名称；
5. (不重要) Driver-Model 表示驱动运行的模式。常见模式：¹(1) Windows Display Driver Model (WDDM)¹：

¹https://en.wikipedia.org/wiki/Windows_Display_Driver_Model

面向图形和显示，GPU 同时用于图形渲染和计算；（2）Tesla Compute Cluster (TCC)²：面向计算的模式，GPU 完全用于计算；

6. Pwr:Usage/Cap 表示当前/最大功率。下方对应的 1W / 80W 表示当前功率 1 W，最大 80 W；
7. Memory-Usage 表示显存 (Video RAM, VRAM) 的使用情况。下方对应的 72MiB / 8188MiB 表示总显存为 8188 MiB（约 8 GiB，称 0 卡是 8G RTX 4060），当前已用 72 MiB；
8. GPU-Util 表示 GPU 核心的利用率。下方对应的 0% 表示利用率为 0%，即空闲。

注 区分 MiB/MB 与 GiB/GB：

1. 1 MiB = 2^{20} Bytes, 1 GiB = 1024 MiB；
2. 1 MB = 10^6 Bytes, 1 GB = 1000 MB

下方的 Processes 表示使用 GPU 的进程表，其中的重要参数：

1. GPU 表示进程占用的 GPU 的编号；
2. PID 和 Process name 分别表示占用 GPU 的进程编号和进程名；
3. Type 表示进程使用 GPU 的形式：（1）C: Compute，计算任务；（2）G: Graphics，图形任务（如桌面显示、图像和视频的渲染等）；（3）C+G：同时有计算和图形任务；
4. GPU Memory Usage 表示进程占用的 VRAM。WDDM 模式下，进程级的 VRAM 占用一般显示 N/A，但不代表进程不占用 VRAM。这是因为 WDDM 将 VRAM 虚拟化，则驱动不会将进程级的 VRAM 占用情况暴露给 nvidia-smi。故 WDDM 模式下，一般只通过 GPU 概览表中的 Memory-Usage 或任务管理器查看总 VRAM 占用情况。

注意事项：

1. 当代（2025）主流的 CUDA 版本是 11.3 / 11.6 / 11.8 / 12.1 / 12.2+，一般驱动版本 525.xx+ 即可兼容；
2. 此处的 CUDA Version: 12.7 不是当前的 CUDA 版本，而是指该驱动跑不超过 12.7 版本的 CUDA 都没问题，因为 CUDA 理论上向下兼容。然而，实践中最好还匹配工程项目的驱动版本和 CUDA 版本，尤其是 CUDA 版本，否则有时会出现奇怪的问题。当然，不代表匹配了就不会出现奇怪的问题。另一方面，CUDA Version: 12.7 有时也能强行跑 12.7+ 版本的 CUDA，但不保证不会出现奇怪的问题。总之，大部分情况下 CUDA 版本与项目使用的 CUDA 版本匹配即可；
3. 某些多年前经典论文的官方实现可能采用较低版本的 CUDA，有时用更高版本的 CUDA 会出现奇怪的问题。解决方法：（1）设置多个版本的 CUDA，请自行查阅教程（不保证不会出现奇怪的问题）；（2）经典的论文一般都有很多民间实现和优化，在 GitHub 上查找年份较新的民间的仓库，一般可兼容当代（2025）主流的 CUDA 版本；
4. 口语化的“把显卡跑满”有两层意思：（1）在不显存超限（俗称“爆显存”，即 CUDA out of Memory，简称 Coom）的前提下，VRAM 占用尽量接近最大 VRAM。一般通过调大 batch_size 实现；（2）让 GPU 利用率尽量接近 100%。GPU 利用率低常是因为数据加载慢、并行度低、任务太小。解决：① 批处理/数据并行，如将多次小计算合并为一次大计算；② 异步并行：加载数据与 GPU 计算同时进行，避免 GPU 等待 CPU 加载数据；③ 避免频繁在 CPU 与 GPU 间拷贝数据；④ 提高 batch_size，充分发挥 GPU 能力；⑤ 混合精度训练，加速训练，并减小 VRAM 占用，进而可进一步提高 batch_size；⑥ 算子融合：将多个运算融合为一个算子。

1.1.1.3 nvcc -V

NVIDIA CUDA Compiler (nvcc) 用于编译 .cu 文件。在终端运行

```
1 nvcc -V
```

输出

```
1 (no_pain_diffusion) PS C:\Users\Hytidel> nvcc -V
```

²https://docs.nvidia.com/gameworks/content/developertools/desktop/tesla_compute_cluster.htm

```

2 nvcc: NVIDIA (R) Cuda compiler driver
3 Copyright (c) 2005-2022 NVIDIA Corporation
4 Built on Tue_Mar__8_18:36:24_Pacific_Standard_Time_2022
5 Cuda compilation tools, release 11.6, V11.6.124
6 Build cuda_11.6.r11.6/compiler.31057947_0

```

中 Cuda compilation tools, release 11.6, V11.6.124 表示 CUDA Toolkit 编译器版本号（俗称 *CUDA* 版本）为 11.6，小版本号（基本用不到）为 11.6.124。因 CUDA 版本 11.6 不超过驱动支持的最高版本 12.7，故理论上可正常使用。

1.1.2 配置环境

在终端运行

```

1 conda create -n no_pain_diffusion python=3.10 -y
2
3 conda activate no_pain_diffusion

```

创建一个 Python 3.10（或更高版本）的 conda 环境，并激活。

运行

```

1 python --version

```

输出 Python 版本，我的版本是 3.10.18。

```

1 (no_pain_diffusion) PS C:\Users\Hytidel> python --version
2 Python 3.10.18

```

安装 torch 2.1.x（或更高版本）和对应版本的 torchvision。版本可参考³。建议安装 CUDA 11.8 对应的 torch 和 torchvision，即下方的 URL 结尾的 cu118。虽然我的 CUDA 版本是 11.6，但仍能按 11.8 安装。

```

1 pip install torch==2.3.1 torchvision==0.18.1 --index-url https://download.pytorch.org/whl/cu118

```

检查 CUDA 是否可用。

```

1 import torch
2
3 print(
4     torch.cuda.is_available()
5 )

```

输出 True 即为可用。

安装 requirements.txt 中的其它依赖。

```

1 pip install -r requirements.txt

```

1.1.3 找到 conda 环境的路径

在终端运行：

```

1 conda env list

```

³<https://pytorch.org/get-started/previous-versions/>

结果:

```
1 (no_pain_diffusion) PS C:\Users\Hytidel> conda env list
2 # conda environments:
3 #
4 base                D:\anaconda3
5 d2l-gpu              E:\JupyterNotebookEnv\envs\d2l-gpu
6 no_pain_diffusion * E:\JupyterNotebookEnv\envs\no_pain_diffusion
```

其中的星号表示当前激活的 conda 环境,⁴是 conda 环境 no_pain_diffusion 的安装路径。

在路径后接 /Lib/site-packages 即为该 conda 环境安装的库的目录,如我的是⁵。

在扩散模型的科研中,常需在 `diffusers` 库的代码的基础上修改。`diffusers` 库的代码可在上述路径后接 /diffusers 找到。

⁴E:/JupyterNotebookEnv/envs/no_pain_diffusion

⁵E:/JupyterNotebookEnv/envs/no_pain_diffusion/Lib/site-packages

1.2 Python 基础

1.2.1 `if __name__ == "__main__"`

1.2.2 工作路径

1.2.2.1 展示工作路径

1.2.2.2 切换工作路径

1.2.3 Jupyter Notebook

1.2.3.1 切换工作路径

1.2.3.2 注意事项

1.2.4 Python 基本语法

1.2.5 `locals()`、`globals()`、`SimpleNamespace`

1.2.6 断点、调试

1.2.7 `generator`、`yield`

1.2.8 文件读写

1.2.9 面向对象

1.2.9.1 Python 面向对象基础语法

1.2.9.2 `__call__()`

1.2.9.3 静态变量

1.2.9.4 类方法

1.2.9.5 抽象类、抽象方法

1.2.9.6 `MethodType`

1.2.10 `arg`、`kwargs`

1.3 常用 Python 库

本节介绍 AI 科研中常用的部分 Python 库。可视化和绘图相关的库见2章，深度学习相关库见3章，机器学习相关库见4章，数学相关库（如数值计算等）见5章。

1.3.1 数据类型

1.3.1.1 typing 库

1.3.1.2 更多数据类型

1.3.2 pathlib 库

1.3.3 itertools 库

1.3.4 运行

1.3.4.1 命令行参数

1.3.4.2 指定显卡

1.3.4.3 Hydra 库

1.3.4.4 Fire 库

1.3.5 数据处理

1.3.5.1 NumPy 库

1.3.5.2 Pandas 库

1.3.5.3 SciPy 库

1.3.6 常用文件类型的读写

1.3.6.1 .npd

1.3.6.2 .yaml

1.3.6.3 .pkl

1.3.6.4 .json

1.3.6.5 .csv、.tsv

1.3.7 进度条

1.3.7.1 Tqdm 库

1.3.8 并行

1.3.8.1 Multiprocessing 库

1.3.8.2 concurrent.futures

1.3.9 前端

1.3.9.1 Gradio 库

第二章 可视化与绘图基础

第三章 深度学习基础

第四章 机器学习基础

第五章 数学基础