

一、说明

本教程专为中国石油大学（北京）信息科学与工程学院/人工智能学院刘建伟老师组的学生而设计。

该教程的主要目标是引导完全零基础的师弟师妹们初步学习深度学习编程。它仅仅是帮助大家入门（告诉大家需要学习的内容，提供我认为优质的学习资料，给出我个人学习的建议等），后续的学习需要大家自行探索。

作者：zh4men9

邮箱：zh4men9@163.com

二、个人愚见

人生不是做菜，不要等所有食材准备好了才下锅 —— 《饮食男女》

在学习、科研和项目开展等方面，永远不会有完全准备好的时刻，知识储备永远不够充分。关键在于勇敢踏出第一步，只有在实际行动中，才有可能完成任务并取得成功。

以我自己的经验为例，我专注于强化学习研究。最初，我认为必须彻底学习经典书籍和算法，然后才能开始进行科研和实验。我印制了大量书籍和论文，兴致勃勃地开始学习。但很快，我发现自己感到沮丧。书籍众多，公式复杂，我发现很快就忘记了所学，而且没有取得任何小规模的研究成果，这让我感到焦虑。后来，我与刘老师交流，了解到老师以前也有相似的经历，老师建议我直接着手进行科研。于是，我在三天内匆忙阅读了一本优秀的强化学习入门书籍，对强化学习相关的算法和方法有了一定了解。然后，我开始进行科研工作，找到了一篇带有开源代码的论文，开始寻找创新点。接着，我开始进行实验，遇到不懂的地方就及时学习。最终，我撰写了我的第一篇强化学习论文（幸运地取得了成功）。

对于那些没有编程基础的师弟师妹们，无需害怕或感到焦虑。请放心地学习和实践，遇到问题就积极解决，毕竟有前辈们为你们铺平道路。有疑问时，一定要积极向师兄师姐们请教，我们组的师兄师姐们都非常友善，只要你怀着学习的态度，他们一定会乐意提供帮助。

三、学习路线

我将列举出我认为相当重要的内容，每个部分包括以下要点：

- 学习理由
- 学习建议
- 学习资料【请注意：提供的学习资料相对丰富，不需要全部学习，可以根据个人学习进度有选择地学习】

VSCode

- 学习理由
 - i. 我自己使用VSCode，所以如果你在使用过程中遇到问题，可以随时向我咨询。
 - ii. 如果组内的大多数人都使用VSCode，那么后续的师弟师妹们可以向你们请教，形成更好的协作和学习氛围。
 - iii. 我个人认为VSCode比较轻量，插件丰富，使用体验相当不错。
- 学习建议
 - i. 把VScode最基本的操作学会（新建文档，保存，运行，调试，插件安装，SSH连接服务器等）。【SSH连接院内服务器参考刘老师群里文件：服务器开发工具及其使用教程XX版本.docx】
 - ii. 项目实践中学习更多VScode高级操作
- 学习资料
 - i. [B站视频](#)
 - ii. [VSCode官方教程](#)
 - iii. [VSCode官方文档](#)
 - iv. 我录制的[B站视频](#)
 - v. 搜索引擎（推荐Bing, Google）

Miniconda

- 学习理由

Conda是一个开源的包管理器和环境管理器，旨在简化不同软件包和依赖项的安装、管理和版本控制。它允许用户创建和维护多个独立的Python环境，每个环境可以具有不同版本的Python和不同的软件包依赖关系。

Miniconda是Conda的精简版本，它只包括Conda本身和一些基本的依赖项，而不像完整的Anaconda发行版那样包含大量预安装的软件包。Miniconda的主要目的是让用户更自由地构建他们需要的环境，而不会被预装的软件包所限制。

有了Miniconda后，可以方便的创建不同版本的Python虚拟环境，并且不同Python虚拟环境是相互隔离的，互不干扰。这样可以方便的在不同的环境中安装不同版本的Python包，而不会造成版本冲突。
- 学习建议

- i. 安装Miniconda
 - ii. 虚拟环境的常规操作（创建，激活，退出，删除）
 - iii. conda安装包的常规操作（安装，更新，删除）
 - iv. pip安装包的常规操作（安装，更新，删除）
- 学习资料
 - i. [Miniconda安装及使用](#)
 - ii. [Miniconda 安装\(windows10\)](#)
 - iii. 我录制的[B站视频](#)

Python

- 学习理由

深度学习的主要编程语言是Python，因此精通Python非常关键。
- 学习建议

基础知识可以通过资料中的：《Python编程从入门到实践edition2》学习。
该书分为两部分：基础知识（270页）+实践项目（300页）。

建议使用以下学习方法：

 - i. 快速浏览基础知识部分，建立对Python语言的基础语法的初步了解。在这一阶段，不必深入学习，只需获得初步印象即可，重点是快速了解。
 - ii. 重新阅读基础知识部分，这次要在自己的Python环境中实际操作（推荐使用VSCode作为集成开发环境）。通过实际编程来巩固所学内容。
 - iii. 由于基础知识学完后可能会遗忘，不必担心，这是正常现象。在实际项目中，可以随时查询和回顾以重新记忆。
 - iv. 项目部分是给学有余力的同学的建议，可以进一步应用所学的Python知识。
 - v. Python还有许多高级用法，如果对Python感兴趣，可以自行深入学习（可以在百度上寻找相关资料）。基础知识学完后，已经足够应对大部分深度学习实验了（可以使用ChatGPT帮助编写代码）。建议在实际项目中遇到需要的高级用法时再去学习，更有针对性。

要成为Python的熟练使用者需要持之以恒的学习和不断编程实践，这并不是2-3个月就能完成的任务，需要不断地学习和参与开源项目。
- 学习资料
 - i. Python编程从入门到实践edition2
 - ii. [小甲鱼Python教程](#)
 - iii. [莫烦Python教程](#)

Pytorch

- 学习理由

- i. 门槛低
 - ii. 目前大多数深度学习项目用的Pytorch
- 学习建议
 - i. Pytorch的安装（特别是GPU版本的安装）。
 - ii. 学会基本的Pytorch操作（张量操作，自动求导，模型构建，模型训练，模型测试，模型保存，模型加载等）。
 - iii. 跟着简单的项目跑通代码，理解代码的运行逻辑。
- 学习资料
 - i. [Pytorch安装教程](#)
 - ii. [Windows10 1050Ti cuda toolkit11.3 pytorch-gpu 踩坑教程](#)
 - iii. [pytorch-handbook](#)

Tensorflow

- 学习理由
 - i. 开源项目除了基于Pytorch框架，大多数都是Tensorflow框架了。
- 学习建议
 - i. 建议学习Tensorflow 1.x 版本。因为很多经典的开源项目都是基于Tensorflow 1.x的。2.x版本有时间可以学。
 - ii. Tensorflow的安装（特别是GPU版本的安装）。
 - iii. 我建议是能看懂项目中的Tensorflow代码即可，不需要深入学习。
- 学习资料
 - i. [Tensorflow 1.x 教程](#)

Linux

- 学习理由
 - 院内服务器是Ubuntu系统，学习一些基本的Linux（Ubuntu）操作系统的知识，可以更好地使用服务器。同时也防止在使用服务器时出现一些低级错误（比如rm命令乱删除文件情况！！！）。
- 学习建议
 - i. 掌握基本的命令即可（比如，ls, mv, cp, rm, pwd, nvidia-smi, sudo, apt, grep, |, cat, head, tail, vi, vim, man等）
- 学习资料
 - i. [Linux命令大全](#)
 - ii. [Linux命令大全](#)
 - iii. [Linux教程](#)
 - iv. 《鸟哥的Linux私房菜-基础篇》第四版

Markdown

- 学习理由
 - i. Markdown是一种轻量级标记语言，可以用于编写文档，编写README.md文件，编写博客等。
 - ii. Markdown语法简单，学习成本低，使用方便。
 - iii. Markdown语法可以在Github上直接渲染，可以方便地在Github上编写文档。
- 学习建议
 - i. 掌握基本的Markdown语法即可（比如，标题，列表，引用，代码块，链接，图片等）。
- 学习资料
 - i. [Markdown教程](#)

Git

- 学习理由
 - i. Git是一个分布式版本控制系统，可以用于管理代码，管理文档等。
 - ii. Git可以方便地在不同的电脑上同步代码和文档，方便多人协作。
- 学习建议
 - i. 建议使用Github，如果没有梯子可以选择gitee
 - ii. 掌握基本的Git操作即可（比如，git clone, git add, git commit, git push, git pull, git branch, git checkout, git merge等）。
 - iii. 学习VSCode的git插件
- 学习资料
 - i. [廖雪峰Git教程](#)
 - ii. [阮一峰Git教程](#)

English

- 学习理由
 - i. 四六级
 - ii. 深度学习前沿论文均为英文
 - iii. 刘老师要求撰写英文论文
- 学习建议
 - i. 多看英文论文
 - ii. 多写英文论文
- 学习资料
 - 无

四、项目实战

学完上述学习路径后，练习运行开源项目是巩固技能的重要一环。

我在B站上录制了一系列[实战视频](#)，参考学习。

我录制的实战视频是在没有准备的情况下，直接去跑通代码，所以逻辑会比较凌乱。你们可以重点关注我debug的操作，遇到问题后解决问题的思路和方案等。如果你有更好的debug技巧，欢迎分享~

如果你们有经典的开源项目推荐，也欢迎联系我，我将录制跑通视频来分享给大家。

五、其他资料

搜索学术文献：[谷歌学术镜像](#)、[SemanticScholar](#)

论文+代码：[PapersWithCode](#)

实验记录工具：[wandb](#)

读文献工具：[ReadPaper](#)、[知云文献翻译](#)

文献管理软件：[Zotero](#)

竞赛平台：[Kaggle](#)、[阿里云天池](#)

推荐大佬：吴恩达、李沐、李弘毅（B站有他们的课）

本项目GitHub：<https://github.com/zh4men9/DeepLearningZeroToHero>