

计算机学习指南

stars 9.8k issues 24 open forks 1.4k license MIT

CSDIY

- 1.必学工具
 - 1.1 VPN
 - 1.2 Vim
 - 1.2.1 为什么学习 Vim
 - 1.2.2 如何学习 Vim
 - 1.2.3 推荐参考资料
 - 1.3 Git
 - 1.3.1 为什么使用Git
 - 1.3.2 如何学习 Git
 - 1.4 GNU Make
 - 1.5 CMake
 - 1.6 Latex
 - 1.7 Docker
 - 1.8 实用工具箱
- 2.好书推荐
- 3.数学基础
- 4.数学进阶
- 5.编程入门
- 6.电子基础
- 7.数据结构与算法
- 8.软件工程
- 9.操作系统
- 10.并行与分布式系统
- 11.计算机系统安全
- 12.计算机网络
- 13.数据库系统
- 14.编译原理
- 15.计算机图形学
- 16.Web开发
- 17.数据科学
- 18.人工智能
- 19.机器学习
- 20.深度学习

1.必学工具

1.1 VPN

1.2 Vim

1.2.1 为什么学习 Vim

在我看来 Vim 编辑器有如下的好处：

- 让你的整个开发过程手指不需要离开键盘，而且光标的移动不需要方向键使得你的手指一直处在打字的最佳位置。
- 方便的文件切换以及面板控制可以让你同时开发多份文件甚至同一个文件的不同位置。
- Vim 的宏操作可以批量化处理重复操作（例如多行 tab，批量加双引号等等）
- Vim 是很多服务器自带的命令行编辑器，当你通过 ssh 连接远程服务器之后，由于没有图形界面，只能在命令行里进行开发（当然现在很多 IDE 如 VS Code 提供了
- ssh 插件可以解决这个问题）。异常丰富的插件生态，让你拥有世界上最花里胡哨的命令行编辑器。

1.2.2 如何学习 Vim

不幸的是 Vim 的学习曲线确实相当陡峭，我花了好几个星期才慢慢适应了用 Vim 进行开发的过程。最开始你会觉得非常不适应，但一旦熬过了初始阶段，相信我，你会爱上 Vim。

Vim 的学习资料浩如烟海，但掌握它最好的方式还是将它用在日常的开发过程中，而不是一上来就去学各种花里胡哨的高级 Vim 技巧。个人推荐的学习路线如下：

先阅读这篇 [tutorial](#)，掌握基本的 Vim 概念和使用方式。用 Vim 自带的 [tutorial](#) 进行练习，安装完 Vim 之后直接在命令行里输入 `vimtutor` 即可进入练习程序。最后就是强迫自己使用 Vim 进行开发，IDE 里可以安装 Vim 插件。等你完全适应 Vim 之后新的世界便向你敞开了大门，你可以按需配置自己的 Vim（修改 `.vimrc` 文件），网上有数不胜数的资源可以借鉴。

1.2.3 推荐参考资料

- Neil, Drew. Practical Vim: Edit Text at the Speed of Thought. N.p., Pragmatic Bookshelf, 2015.
- Neil, Drew. Modern Vim: Craft Your Development Environment with Vim 8 and Neovim. United States, Pragmatic Bookshelf.

1.3 Git

1.3.1 为什么使用Git

Git 是一款分布式的代码版本控制工具，Linux 之父 Linus 嫌弃当时主流的中心式的版本控制工具太难用还要花钱，就自己开发出了 Git 用来维护 Linux 的版本（给大佬跪了）。

Git 的设计非常优雅，但初学者通常因为很难理解其内部逻辑因此会觉得非常难用。对 Git 不熟悉的初学者很容易出现因为误用命令将代码给控制版本控制没了的状况（好吧是我）。

但相信我，和 Vim 一样，Git 是一款你最终掌握之后会感叹“它值得！”的神器。

1.3.2 如何学习 Git

和 Vim 不同，我不建议初学者在一知半解的情况下贸然使用 Git，因为它的内部逻辑并不能熟能生巧，而是需要花时间去理解。我推荐的学习路线如下：

阅读这篇 [Git tutorial](#)，视频的话可以看这个[尚硅谷Git教程](#) 阅读这本开源书籍 Pro Git 的 Chapter1 - Chapter5，是的没错，学 Git 需要读一本书（捂脸）。此时你已经掌握了 Git 的原理和绝大部分用法，接下来就可以在实践中反复巩固 Git 的命令了。但用好它同样是一门哲学，我个人觉得这篇如何写好 Commit Message 的博客非常值得一读。好的此时你已经爱上了 Git，你已经不满足于学会它了，你想自己实现一个 Git！巧了，我当年也

有这样的想法，这篇 tutorial 可以满足你！什么？光实现一个 Git 无法满足你？小伙子/小仙女有前途，巧的是我也喜欢造轮子，这两个 GitHub 项目 [build-your-own-x](#) 和 [project-based-learning](#) 收录了你能想到的各种造轮子教程，比如：自己造个编辑器、自己写个虚拟机、自己写个 docker、自己写个 TCP 等等等等。

1.4 GNU Make

1.5 CMake

1.6 Latex

1.7 Docker

1.8 实用工具箱

2.好书推荐

3.数学基础

4.数学进阶

5.编程入门

6.电子基础

7.数据结构与算法

8.软件工程

9.操作系统

10.并行与分布式系统

11.计算机系统安全

12.计算机网络

13.数据库系统

14.编译原理

15.计算机图形学

16.Web开发

17.数据科学

18.人工智能

19.机器学习

20.深度学习

课程简介

- 所属大学: Stanford
- 先修要求: 机器学习基础 + Python
- 编程语言: Python
- 课程难度: 🌟🌟🌟🌟
- 预计学时: 80 小时 吴恩达在 Coursera 开设的另一门网红课程, 学习者无数, 堪称圣经级的深度学习入门课。深入浅出的讲解, 眼花缭乱的 Project。从最基础的神经网络, 到 CNN, RNN, 再到最近大热的 Transformer。学完这门课, 你将初步掌握深度学习领域必备的知识 and 技能, 并且可以在 Kaggle 中参加自己感兴趣的比赛, 在实践中锻炼自己。

课程资源

- 课程网站: <https://www.coursera.org/specializations/deep-learning>
- 课程视频: <https://www.coursera.org/specializations/deep-learning>, B站有搬运
- 课程教材: 无
- 课程作业: <https://www.coursera.org/specializations/deep-learning>