



SCHOOL OF DATA  
SCIENCE & ENGINEERING  
**数据科学与工程学院**

## 《开源软件设计与开发》课程总结

院 系： 数据科学与工程学院计算机技术

学 号： 51195100042 姓 名： 刘中祺

2019 年 12 月 31 日

# 目录

1. 开源理解 .....	3
1.1 开源软件与开源社区.....	3
1.2 开源社区的起源.....	4
1.3 开源与人工智能.....	6
1.4 开源人工智能框架 .....	7
2. 开源贡献 .....	9
2.1 Dota AutoChess Community .....	9
2.2 FastNLP Community.....	10
3. 课程反馈 .....	12
4. 参考文献 .....	12

# 1. 开源理解

## 1.1 开源软件与开源社区

开源软件，即开放源码软件。开源软件是与源代码一起分发的软件，可以由用户读取或修改。普遍开源软件应符合自由分发、源代码必须包含在程序中、任何人都必须能够修改源代码、可以重新分发源代码的修改版本等标准。并且此软件的使用，修改和分发也不受许可证的限制。

开源社区又称开放源代码社区，一般由拥有共同兴趣爱好的人所组成，根据相应的开源软件许可证协议公布软件源代码的网络平台，同时也为网络成员提供一个自由学习交流的空间。由于开放源码软件主要被散布在全世界的编程者所开发，开源社区就成了他们沟通交流的必要途径，因此开源社区在推动开源软件发展的过程中起着巨大的作用。



开源社区普遍认为开源软件应符合以下标准：

- 该计划必须自由分发
- 源代码必须包含在程序中
- 任何人都必须能够修改源代码
- 可以重新分发源代码的修改版本

开源软件许可证不得要求排除或干扰其他软件的运行。与以不可更改的编译格式分发的传统软件不同，开源软件以编译和非编译格式提供，允许开放代码修改。在传统的软件许可

中，此特权将保留给版权所有者。并非所有软件开发人员都赞成使用开源软件，但许多软件开发人员已经接受了它，因为它可以更快地修复软件问题，最终可能会带来更高质量的应用程序。

## 1.2 开源社区的起源

### （1）私有软件

1976 年 2 月 3 日，Bill Gates 发表了著名的《Open Letter to Hobbyists》(致电脑业余爱好者的一封信)，在这封公开信中，Bill Gates 明确提出软件应该具版权。这封公开信奠定了软件产品版权的理论基础，也促使私有软件成为软件产业发展的主流。版权理念的提出，必然导致源代码被彻底封闭起来。



私有软件就是闭源软件。客观地讲，Bill Gates 的“Copy Right”理念以及私有软件的兴起，对于推动软件产业的兴旺功不可没。但是，Microsoft 等私有软件厂商对于软件市场的垄断，引发了用户和广大软件开发者的强烈不满，兴起了反对私有软件理念的运动，开源软件就是这种运动的重要组成部分。

### （2）自由软件

1983 年，MIT 的 Richard Stallman 开始倡导自由软件运动。1985 年，Stallman 等人创建了自由软件基金会。Stallman 提出了与“Copy Right”理念针锋相对的“Copy Left”理念，其表现形式是 GPL，即公共许可证。

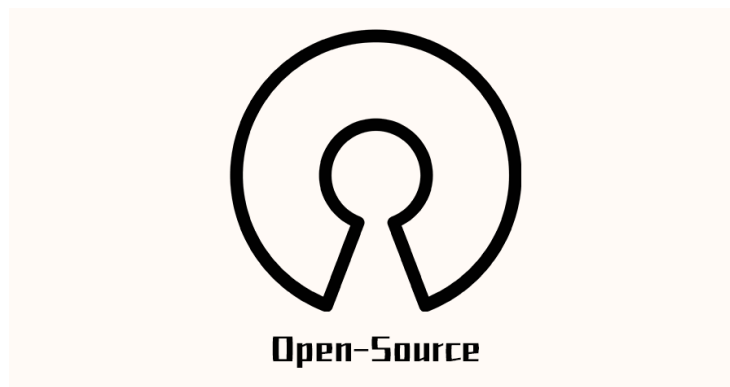
1991 年，Linus Torvalds 开发出 Linux，并接受“Copy Left”理念，使得自由软件运动有了自己可以与 Microsoft 的 Windows 相抗衡的操作系统。自由软件运动初战成功。但是，自由软件运动关于自由的追求，毕竟和现实的商业氛围格格不入，带着过于理想化的色彩。这种反商业的信条，让一些本来也反对私有软件的人士对自由软件敬而远之。正是在这种背景下，一部分原有自由软件运动人士，如 Eric Raymond，开始尝试将理想的自由软件与现实的

商业氛围进行某种衔接，这就是本文所涉及的开源软件运动。



### （3）开源软件

1998 年 2 月，Eric Raymond 等人正式创立“Open Source Software”这一名称，并组建了开放源代码创始组织。OSI 曾给出了开源软件的标准定义。按照 OSI 的定义，开源软件是指允许任何人使用、拷贝、修改、分发的软件。



开源软件就是源代码开放且可以自由复制的软件。开源软件运动超越自由软件运动的地方在于，它既抓住了私有软件的 Achilles Heel，即封闭源代码，又可以实现与商业氛围的结合。

### （4）开源社区

由于以 Microsoft 为代表的私有软件企业占据着软件产业的主流地位，开源软件要想生存和发展，就必须在组织形式上进行创新。这一创新是由 Eric Raymond 在 1998 年提出的，他在被堪称开源软件的《The Cathedral and the Bazaar》中正式提出了“开源社区”模式。开源社区是开源软件开发最重要的组织形式，是由所有参与开发和改进可以软件的爱好者所组成的社区。

开源社区这种组织有着不同于软件开发企业的本质特征，这种本质特征就在于，软件开发企业内部的各个员工是基于利益关系而组织在一起的，而开源社区内部的各个成员是基于兴趣而走到一起的。从这个角度来说，软件开发企业是一种利益组织，而开源社区是一种兴趣组织。

### （5）开放社区

对于开源社区这种软件开发组织，西方学者进行了更加深入的研究，他们提出了更为深刻的理解，这就是“开放社区”。如果仅仅凭借兴趣，开源社区是难以发展壮大起来的，开源社区要想真正发展壮大到可以和巨型软件开发企业进行有效的竞争，就必须在兴趣的基础上，形成共同的文化，也就是要将开源社区从兴趣组织发展为文化组织。

## 1.3 开源与人工智能

深度学习是新一代人工智能的核心支撑。在人工智能的技术体系中，深度学习框架处于硬件层和应用层之间，其作用相当于个人电脑时代的 Windows 和移动时代的 Android/iOS，堪称智能时代的操作系统。以深度学习框架为核心的深度学习平台，是人工智能技术研发和产业化的关键基础设施。

不过，如果要独立研发深度学习框架，搭建完整的深度学习平台，需要强大的技术支撑。迄今为止，全球范围内能够做出功能完备深度学习平台的企业，只有 Google、Facebook 等少数几家。而它们推出的深度学习开源平台，也确实在全球人工智能领域占据很大份额，不少国内企业同样使用。



图17 基于开源软件的人工智能典型解决方案 PMLCAFF 产品经理社区

研究和发展技术领先、功能完备、自主可控的深度学习框架和平台，对于推动我国人工智能的技术创新、产业发展和人才培养，实现人工智能技术和产业的自主可控，加速各行各业智能化升级，具有至关重要的作用。

在 GitHub 上的热门开源项目中，人工智能类占了很大比例。开源开放是人工智能领域发展的全球主要趋势之一。人工智能的快速发展，推动开源开放生态越发繁荣；而开源开放

同时也推动人工智能快速普及，两者相互促进。

## 1.4 开源人工智能框架

### （1）TensorFlow

TensorFlow 是一个开源软件库，最初由 Google Brain Team 的研究人员和工程师开发。TensorFlow 使用数据流图进行数值计算。图中的节点表示数学运算，边表示它们之间通信的多维数据数组（张量）。其架构灵活，您可以使用单个 API 将计算部署到桌面、服务器或移动设备中的一个或多个 CPU 或 GPU。



TensorFlow 提供了多种 API。最低级别的 API—TensorFlow Core—提供了完整的编程控制。高级 API 则建立在 TensorFlow Core 的顶部。这些更高级别的 API 通常比 TensorFlow Core 更容易学习和使用。此外，更高级别的 API 使得重复性的任务在不同的用户之间变得更加容易、更一致。一个高级 API 就像 `tf.estimator`，可以帮助您管理数据集、评估器、训练和推理。

### （2）Caffe

Caffe 是一种清晰而高效的深度学习框架。Caffe 最初由杨庆佳在加州大学伯克利分校读博期间发起，后来由伯克利 AI 研究公司（BAIR）和社区贡献者联合开发。它主要专注于用于计算机视觉应用的卷积神经网络。对于计算机视觉相关的任务来说，Caffe 是一个不错且较为流行的选择，您可以在 Caffe Model Zoo 上注册，下载很多已经成功建模的模型，直接用于开发。





### (3) OpenNN

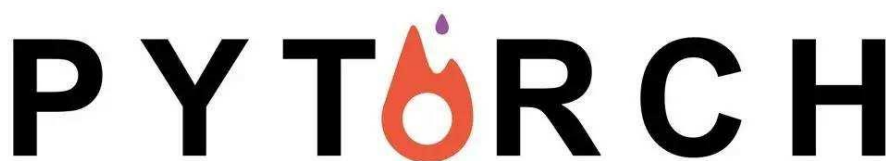
Opennn (开放神经网络图书馆)以前被称为 Flood，它是以 R. Lopez 在 2008 年泰罗尼亚技术大学的博士论文《在工程变分问题的神经网络》为基础开发的。Opennn 使用一组函数实现了数据挖掘，并且，可以使用一个 API 将这些函数嵌入到其他软件工具中，使软件工具和预测分析任务之间进行交互。

Opennn 的主要优点就是它的高性能。由于采用 c++开发，因此它有更好的内存管理和更高的处理速度，并利用 OpenMP 和 GPU 加速度 (CUDA) 实现 CPU 并行化。Opennn 包中含有单元测试、许多示例和大量文档。为神经网络算法和应用的研究开发提供了一个有效的框架。神经网络设计是一个基于 OpenNN 的专业预测分析工具，这就意味着神经网络设计的神经引擎是基于 OpenNN 建立的。OpenNN 旨在从数据集和数学模型中进行学习。



### (5) Torch

Torch 是一个开源机器学习库、科学计算框架和基于 Lua 编程语言的脚本语言。Torch 供 Facebook 人工智能研究小组、IBM、Yandex 和 Idiap 研究所使用。现在，它已经扩展到 Android 和 iOS 系统上，研究人员也使用 Torch 来构建硬件实现数据流。



Pytorch 是一个 Python 的开源机器学习库，用于自然语言处理等应用，主要由 Facebook 的人工智能研究小组开发，Uber 的概率编程软件"Pyro"就是在 Pytorch 上创建的。



## 2. 开源贡献

### 2.1 Dota AutoChess Community

Dota 自走棋这款游戏是基于目前世界上具有相当高知名度的老牌 MOBA 游戏 Dota2 的一款新兴游戏，由国人巨鸟多多工作室开发，在 2018-2019 年再一次掀起了新的世界级游戏热潮。



刀塔自走棋将策略对战棋牌玩法与 Dota 自定义地图结合，支持 1-8 人游戏。刀塔自走棋将参加游戏的 8 位玩家划分在 8 块不同的棋盘上，通过每回合抓取棋池中的棋子，挑选搭配出不同的棋子组合，并与玩家和 AI(野怪)对抗，获取金币，购入棋子，不断强化自己的组合来获得胜利。

#### (1) 小黑盒自走棋社区

Steam，由美国传来的世界最大的游戏平台，世界上绝大多数游戏，包括 Dota2，都在 Steam 平台上发布，玩家们在平台上消费购买正版游戏，无论是 PC 端还是主机端，各类游戏应有尽有。

小黑盒，由国内企业出品，专为 Steam 游戏玩家定制，集新闻资讯、游戏资料库、玩家社区、战绩查询为一体的手机 APP，小黑盒与 Steam 联通，用户可以在 APP 上获取大量游戏相关的资讯、攻略，也可以在玩家社区内与不同玩家进行互动分享。



自走棋风靡之后，全球数十万的玩家沸腾了，国内玩家自然也在小黑盒上聚集起来，自走棋社区也成为首页即可看到的榜上有名的社区。每天都有数以百计千计的玩家在社区内发布各类自走棋相关文章，包括版本更新内容、游戏攻略、阵容搭配、新颖思路、上分指南、同人故事、新版本猜想、对开发者的建议等等。即便是游戏的开发者也常常混迹与自走棋社

区，毕竟这是开发者了解玩家想法的一大重要途径。

包括我在内的众多的玩家在自走棋社区内不断的分享交流游戏的见解，并为开发者提供新的版本思路，我相信虽然没有很多公开信息能证明，但开发者在自走棋社区内必然了解到了非常多有用的信息与玩家意见，以及创新性的思路。

## （2）GitHub 自走棋

Dota 自走棋的游戏源码是公开的，随意都可以访问。自走棋由 lua 语言开发，是一个小巧的轻量级游戏脚本语言。为了提高自己的游戏技巧和了解游戏内部代码机制，我分析源代码，分析游戏本质，棋子的移动，技能的释放等等，摸清楚这些就掌握了别人所没有掌握到的知识。

在代码分析中，关于游戏中的热点技能“末日”，经过我们的分析发现，“末日”这个技能确实是选择最高等级的棋子，具体机制是，施放技能时，算法会遍历敌对阵容的棋子列表，当新遍历的棋子等级大于等于之前的最高等级时，技能选择的目标则发生变化，这个顺序并非随机产生的，而是根据用户对棋子的上阵下阵即时改变的，先上阵的棋子将排列在列表的前方，后上阵的则在后方。

发现了这个技能的机制后，我在 github 和小黑盒中立即做出了分享，并且获得了大量的好评与回应。是自走棋游戏社区的首次发现，“末日”这个技能可以说是游戏中最为头疼的几个技能之一，为玩家们游戏对局具有很大的意义。

有一次在分析棋子技能时，关于棋子“神谕者”的技能“命运敕令”。自走棋是由国人开发，代码中的变量名，尤其是技能的变量名通常以拼音命名，“命运敕令”的技能变量命名为 mingyunsheling，技能中“敕”字的拼音为 chi，但很明显，变量名中的 she 为开发人员因拼写错误而诞生。

我把这个位置记下来，并且在 github 上给开发者提了一个 issue。过了几天，这个 issue 有了回应，原来拼错的变量名依然是错的，但是在那行代码的上方多了一行注释：“这个地方当时开发的时候拼错了，应该是 chi，但是后面牵扯太多代码了不好弄就懒得改了【捂脸】”

## （3）与开发者的直接交流

我认识一个刀塔自走棋的大主播，通过他我们能与自走棋的开发者巨鸟多多工作室直接的交流，很多时候我们会跨越社区，直接为开发者做出建议。在一次大的版本更新后的抽棋机制变得并不合理，开发者为了加快对局节奏，缩小了棋子池。针对这一点我与几个程序员好友以及主播，给开发者直接做了建议，也就有了现在合理而受欢迎的“D++”抽棋机制。

## 2.2 FastNLP Community

FastNLP 是一款轻量级的自然语言处理（NLP）工具包，目标是快速实现 NLP 任务以及构建复杂模型。

FastNLP 由复旦大学自然语言处理领域专家邱锡鹏老师团队开发，我有幸加入了

FastNLP 开发者交流群，进入了 FastNLP community。在交流群内与诸多使用者与开发者交流讨论经验，并一起为框架的进一步发展做努力。

Description:

Add the **YellowFin** optimizer, called YFOptimizer.

YellowFin is an auto-tuning optimizer based on momentum SGD which requires no manual specification of learning rate and momentum. It measures the objective landscape on-the-fly and tunes momentum as well as learning rate using a local quadratic approximation.

We can use YFOptimizer just like Adam and SGD.

For example:

```
from fastNLP.core.optimizer import YFOptimizer

trainer = Trainer(model=model, n_epochs=100, optimizer=YFOptimizer(lr=0.01),
train_data=train_data, dev_data=dev_data, loss=CrossEntropyLoss(), metrics=AccuracyMetric() )
```

paper: <https://arxiv.org/abs/1706.03471>

code: [https://github.com/JianGoForIt/YellowFin\\_Pytorch](https://github.com/JianGoForIt/YellowFin_Pytorch)

Main reason: Add a fancy optimizer, which performs well in many tasks, though has its shortcomings. The biggest advantage is that you can use the default parameters directly, no need to adjust LR manually. (Note: there are also some people say they get a bad performance.)

在调用 FastNLP 进行深度学习实验时，调整模型结构过程中希望使用 YellowFin 优化器，YellowFin 是由斯坦福大学提出的 SGD 动量自调节器算法，所以在代码中加入了 YellowFin 的 API，提出了 PR 并且非常荣幸能够成功被接纳。



**Changes approved**

[Show all reviewers](#)

1 approving review by reviewers with write access. [Learn more.](#)



1 approval



3 pending reviewers



**All checks have passed**

[Show all checks](#)

1 successful check



**This branch has conflicts that must be resolved**

Only those with [write access](#) to this repository can merge pull requests.

**Conflicting files**

fastNLP/core/optimizer.py

### 3. 课程反馈

#### （1）开源软件的选择

课程里所提供的有导师的开源软件多是数据库方向、应用方向等，对于半数的数据挖掘与机器学习方向的同学，由于研究方向的不同，很难调动起兴趣。

#### （2）Slack 交流工具的使用

Slack 一般在成体系的大公司和社区内使用较广，但是对于学校内部，尤其是大家的参与度不够高的情况下，Slack 的使用多少有点邯郸学步的感觉，甚至限制了同学们的热情。

#### （3）课堂内容的传授

课堂内容大多数是开源相关的理论课，较为枯燥，可以考虑加入更多实践性的内容，或者互动交流性的内容。

### 4. 参考文献

- [1] 张盖伦. 开源开放是人工智能发展主要趋势之一[N]. 科技日报. 2019-10-17(003)
- [2] 袁萌. 开源社区的力量[J]. 开放世界系统. 2006(8):1
- [3] 陈伟. 开源社区在中国[J]. 软件世界. 2006:34-38