

《计算科学导论》课程总结报告

|  |  |
| --- | --- |
| 姓 名 | 纪淑琴 |
| 学 号 | 2407010103 |
| 专业班级 | 计科2401 |
| 学 院 | 计算机科学与技术学院 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程认识  30% | 问题思考  30% | 格式规范  20% | IT工具  20% | 总分 | 评阅教师 |
|  |  |  |  |  |  |

2024年11月24日

# 1 引言

作为计算机专业的学生，我们怀着对代码世界的向往进入大学，满心都是成为技术大神、攻克实践项目的宏大愿望。当时，听到 “计算科学导论” 这门课程，我的内心深处是不解——我们为什么不多花时间去学习编程、开发，却要来学习这看似乏味无聊的计算科学导论？

最开始，我以为它不过是些老生常谈的“概念”“意义”“历史”，相较于亲手敲下一行行代码、看着程序成功运行的成就感，导论课本的知识显得无味而抽象。然而，随着课程的逐渐进行，我才发现我错得离谱。如今课程已经结束，下文我将会回顾这段学习历程，梳理我的所学所思所悟。

# 2 对计算科学导论这门课程的认识、体会

总的来说，《计算科学导论》具有极大的深度和广度，是我们计算机学生的奠基课程。也许我们还不能完全理解导论课上讲到的知识，但是导论课程让我们对计算机专业的历史、发展、前景有了宏观的认识，它包罗万象，无所不有，就像一座沉默的宝库，等待着我们在未来成长的道路上一点点去熟悉、挖掘。计算科学导论从科学哲学的思想方法讲起，循序渐进到计算科学的意义、内容与方法。二进制代码不再是一串串冰冷的字符，而是信息存储、传输与处理的基石；算法原理也不再是晦涩难懂的规则，而是优化程序、提高效率的途径。这门课串联起看似孤立的专业知识，让我意识到计算机硬件与软件息息相关，进而决定整个系统性能。

计算科学导论展现了计算机行业多学科融合的特点，为我们介绍了大量典型实例。计算科学导论课程不仅包括计算机领域知识，还包括数学、物理、哲学、逻辑学等方面的相关内容，比如丢番图方程的解、计算的正确性问题、集成电路、分布式网络协议等等。其中最令我印象深刻的是“哲学家共餐问题”。假设有五位哲学家围坐在一张圆桌旁，每位哲学家左右手边各放有一支筷子，每位哲学家需要同时拿起其左边和右边的筷子才能开吃。由于筷子被相邻两位哲学家共用，很容易出现僵持局面，使程序陷入 “死锁”。例如，五位哲学家同一时刻伸手去拿左边筷子，这时每人都手握一支筷子，并且都在等右边的邻座放下筷子。程序就此卡顿，哲学家们都吃不上饭。这一问题还有待我们进一步学习和研究。此外，还有“停机问题”“荷兰国旗问题”等贴近生活实际、解决具体难题的典型实例，让我们对抽象的概念有了更加具体深入的理解。

## 2.1 计算科学与计算机科学有何区别？

在研究内容上，计算科学属于理学门类，是一个融合了数学、物理、计算机科学等多个学科的领域，主要使用先进的计算能力和计算技术来理解和解决复杂的问题，注重理论研究。计算机科学属于工学门类，研究[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA/140338?fromModule=lemma_inlink" \t "_blank)及其周围各种现象和[规律](https://baike.baidu.com/item/%E8%A7%84%E5%BE%8B/3311038?fromModule=lemma_inlink" \t "_blank)，比如研究计算机系统结构、程序软件、人工智能以及计算本身的性质和问题的[学科](https://baike.baidu.com/item/%E5%AD%A6%E7%A7%91/2634099?fromModule=lemma_inlink" \t "_blank)[1]。也就是说，计算机科学既研究抽象的算法、语法，也研究具体的编程语言、软件和硬件等，更注重实际应用。

计算科学强调的是“问题求解”的过程，特别是对于计算过程特别复杂、需要大量计算的问题，重点在于建立合适的计算模型，将实际问题转化成计算机能够理解的形式，从而使用计算机进行进一步数值计算或者模拟。简单来说，就是利用各种计算技术，通过数学建模、物理原理应用等方法，在不同领域中解决数值计算方面的问题。比如，我读到有研究者利用云计算、量子遗传算法等构建蛋白质折叠空间结构预测框架，建立最小蛋白质分子能量优化函数作优化求解[2]。

计算机科学更强调的是计算机系统的组成部分及其相互关系。硬件上包括研究计算机的设计原理和性能优化，包括内存、硬盘、运算器、控制器等；软件上则研究程序开发、系统架构等内容。比如，有研究者针对控制器放置问题MCP提出了一种融合遗传算法和k-medoid聚类算法的启发式SDN MCP机制，提供有效的低延迟的控制器部署方案[3]。

## 2.2 AI会不会产生意识？

在《银翼杀手》等科幻电影中，复制人拥有与人类别无二致的自我意识。这不仅让人思考，在AI技术日新月异的今天，AI是否真的会像影视作品中的人造生命一样，产生属于自己的意识？下文我将结合计算机的数学起源探究这一问题。

从课程中我们可以学习到，丢番图方程的求解问题为计算机科学中的可计算性理论奠定了基础。但很多丢番图方程是极其复杂甚至无解的。这表明在数学领域，问题的求解存在边界。同理，基于数学逻辑构建的计算机，其运算能力也有边界。

无理数和超越数的发现，说明数学中存在无法用简单的分数或有限的代数方程来表示的数。对于AI来说，它们处理数字是基于二进制表示和预设的计算规则，虽然可以近似表示无理数和超越数，但无法真正理解这些数抽象的数学意义。这表明 AI的认知局限于它所学到的模式，而不是对抽象概念的真正理解。

哥德尔不完备定理中提到，一个包含了算术的任意数学系统，不可能同时满足完备性和一致性。而计算机就是一种基于二进制数字运算的命题演算系统。计算机中给出的运算公理是可计算的，具备一致性。因而我们很容易得出，计算机系统是不完备的。固然，我们可以扩充计算机的运算公理，但是只要公理是有限集，计算机系统仍然是不完备的。然而，对于计算机来说，无限的公理集合代表着无限的存储空间。显然，计算机不可能实现无限的公理集合，因而计算机始终具有不完备性。同时，图灵提出的可计算性理论定义了哪些问题可解，哪些问题不可解。某种意义上，这也为计算机的能力范围划定了边界。AI的计算过程本质上也是在这个可计算的范围内。

此外，从感性的角度来看，AI的构建只是基于大量的数据和预先设计的算法，这种学习是对已有模式的归纳和模仿，缺少内在的自我认知。意识的产生涉及到自我认知、情感经历等目前还难以用算法完全描述的复杂内容。而很明显AI缺乏产生主观体验和情感的生理基础。重要的是，AI缺少自我质疑的能力，它无法突破预设的思维定势。

综上所述，现有技术下，AI是不可能产生意识的。

## 2.3 那些专业建议2025届高考考生选择？

以下是一些建议2025届高考考生选择的专业：

* **计算机科学与技术**：它是信息技术的基础学科，包括计算机硬件、软件、网络、算法等多方面知识。2025届考生可从事算法、软件开发等工作，就业范围广泛，薪资水平较高。
* **软件工程**：更加注重软件设计与开发，培养学生软件项目分析、实现、测试和维护的能力。随着信息技术的不断发展，软件的应用场景越来越广泛。从移动应用到企业级系统，从智能设备到云计算平台，都需要大量的软件工程专业人才来进行开发和维护，就业前景广阔，薪资待遇也相对较高。
* **电子信息**：涉及电子技术、信号处理、通信原理、计算机技术等多个领域，旨在培养能够从事各类电子设备和信息系统的研究、设计、制造、应用和开发的高级工程技术人才。该专业的就业面广，可从事电子设备研发、通信系统设计、信号处理等工作。
* **电气工程及其自动化**：主要研究电能的产生、传输、变换、控制和利用，培养能够从事电气工程领域的系统运行、自动控制、电力电子技术、信息处理、试验分析等工作的高级工程技术人才。毕业生可在电力系统、电气设备制造、工业自动化等领域就业，如进入国家电网、发电厂、电力设计院等单位工作。
* **数学与应用数学**：数学是自然科学的基础，该专业培养学生具备扎实的数学理论基础和较强的逻辑思维能力，为进一步从事数学研究或其他相关领域的工作打下坚实的基础。毕业生可在教育、科研、金融、数据分析等领域就业，如从事数学教学、数学建模、金融分析、数据挖掘等工作。AI 领域的许多算法和模型都基于深厚的数学原理，如概率论、线性代数、微积分等。数学专业能够培养学生的逻辑思维和问题解决能力，为理解和应用 AI 算法提供坚实基础，毕业生可从事 AI 算法研究、数据分析等工作，是 AI 领域不可或缺的人才。

# 3 进一步的思考

以下是对于分组演讲主题“半监督学习”的进一步思考。

* 有哪些对"未标记样本"进行合理预测的方法？

1. 自训练（Self-training）

自训练是一种简单而常用的半监督学习方法。自训练方法在利用小型标记数据集和大型未标记观察值进行预测任务方面更为有效。模型使用学习的分类器的置信度分数来识别低密度区域中的决策边界，而无需对数据分布做出额外的假设。自我训练的核心原则包括迭代地将伪标签分配给置信度分数高于某个阈值的未标记样本，丰富标记的数据集并重新训练分类器[4]。它首先使用"有标记样本"训练一个分类器，然后将该分类器应用于"无标记样本"，并将得到的置信度较高的样本添加为"有标记样本"，然后重新训练分类器[5]。

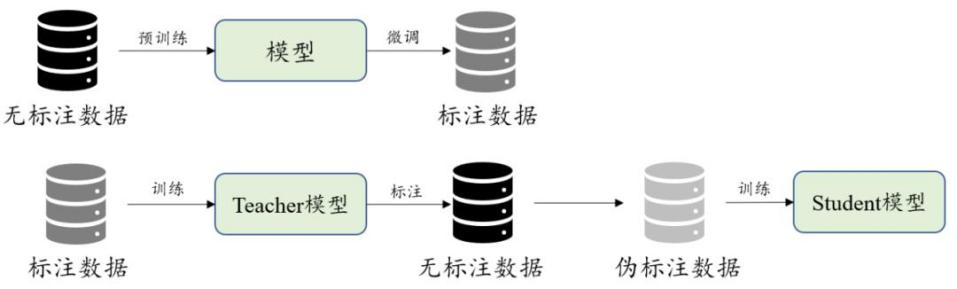


图1 自训练

1. 协同训练（Co-training）

协同训练是一种基于多视角学习的半监督学习方法。它使用多个独立的分类器，每个分类器使用不同的特征子集进行训练。然后，每个分类器将对"无标记样本"进行预测，并将预测结果作为"有标记样本"添加到对方的训练集中，然后重新训练分类器[6]。

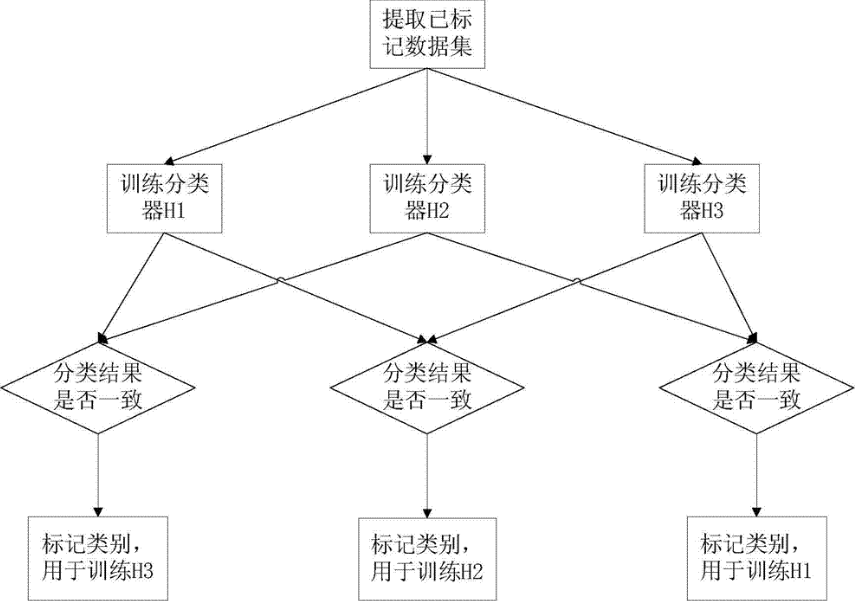


图2 协同训练的流程

1. 分歧校正（Divergence-based correction）

分歧校正是一种基于数据分布的半监督学习方法。它使用“有标记样本”和“无标记样本”的分布差异来提供进行训练，引导机器更好地利用“无标记样本”[6]。常见的方法包括自编码器、生成对抗网络（GAN）、伪标签等。

* 协同训练是否有可行的改进方法？

Blum和T.Mitchell的经典论文"Combining labeled and unlabeled data with co-training."中详细陈述[7]，协同训练假设特征能够分成两个独立分布的数据集，每个子特征集训练一个分类器。每个分类器把未标记的样本分出来给另一个分类器，选择出置信度高的进行标记，重复以上过程。

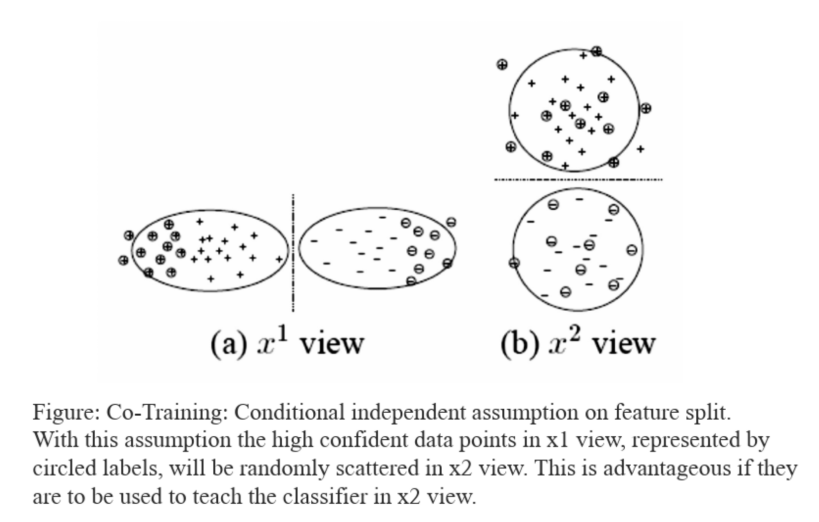


图3 协同训练

这一算法的缺点是，多数问题中，可获得的数据集不够大，而且随机划分视图的策略可靠性较低。

目前，Kuniaki Saito、Yoshitaka Ushiku、Tatsuya Harada在文章《Asymmetric Tri-training for Unsupervised Domain Adaptation》提出了一种非对称三训练方法进行无监督域适应，其中研究者将伪标签分配给未标记的样本，并训练神经网络，就好像它们是真正的标签一样。研究者不对称地使用三个网络，不对称是指两个网络用于标记未标记的目标样本，一个网络由伪标记样本训练以获得目标判别表示。这一方法被证明在用于域适应的基准数字识别数据集上实现了最先进的性能[8]。

此外，有研究表明，Tri-training 通过使用未标记的数据进行分类，可以有效提高分类器的泛化能力，但容易对未标记的数据进行错误标记，从而形成训练噪声。因此，有研究者提出一种基于密度峰值聚类的三训练算法，使用密度峰值聚类算法来获取训练数据的类聚类中心和局部密度，并将类聚类中心的截断距离内的样本标识为空间结构较好的样本，并将这些样本标记为核心数据，并使用核心数据更新分类器，可以减少迭代过程中的训练噪声，从而提高分类器的性能。实验结果表明，与标准 Tri-training 算法及其改进算法相比，DPC-TT 算法具有更好的分类性能[9]。

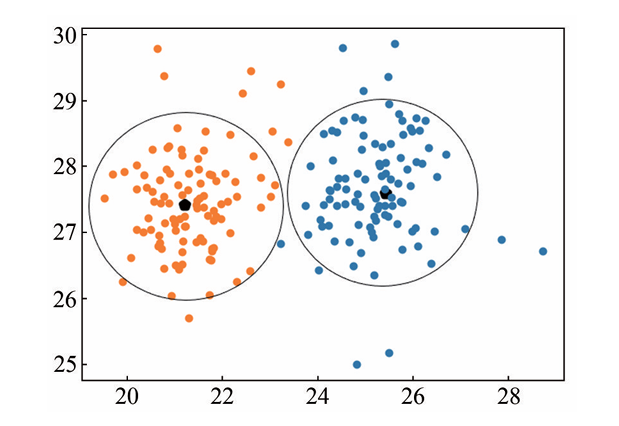


图4 DPC-TT算法噪声过滤示意图

* 半监督学习的优缺点是什么？

与其他方法相比，监督学习需要大量的标注数据来训练模型，无监督学习的精度相对较低，而半监督学习则可以利用未标注数据来提高模型性能，在机器学习中有着独特的优势。

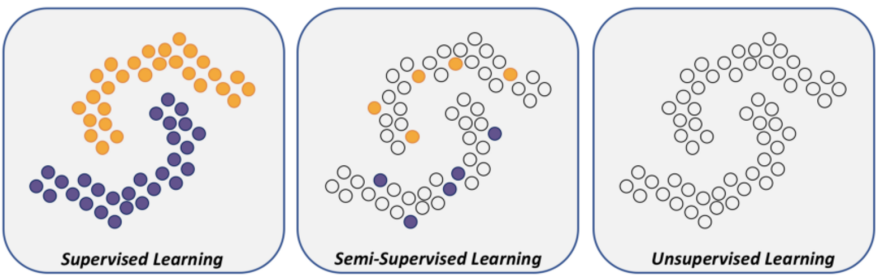


图5 机器学习的三种方法

半监督学习的优点：

1. 利用未标记样本的信息：半监督学习能够充分利用大量未标记样本的信息，相对于仅使用标记样本的监督学习，能够提供更多的数据用于模型训练，从而提高模型的性能和泛化能力。
2. 减少标记成本：相对于监督学习，半监督学习可以通过标记少量的样本来达到较好的性能，从而减少了标记样本的成本和工作量。
3. 扩展应用范围：在一些领域中，例如医学、图像处理等，获取标记样本非常困难，但未标记样本比较容易获得。半监督学习可以在这些场景下扩展应用范围，使得模型能够充分利用未标记样本的信息。

半监督学习的缺点：

1. 假设限制：半监督学习依赖于一定的假设，例如未标记样本和标记样本在数据分布上是相似的。当这些假设不成立时，半监督学习的性能可能会降低。
2. 错误传播：在半监督学习中，未标记样本的预测结果错误可能会传播到后续的训练中，导致模型性能的下降。如果预测结果的质量较低，可能会对模型产生负面影响。
3. 异常值敏感：未标记样本可能存在异常值，如果不加以处理，可能会影响模型的训练效果。

# 4 总结

计算科学导论的课程已经结束，但是我们的计算机旅程才刚刚开始。总结这段导论课程，我收获颇丰，对本专业的历史、发展趋势、行业前景和挑战有了较为全面的认识。在学习中，我对课程中某些深入性的概念并不能做到完全理解，但是在孙老师的教导下，我逐渐学习到了带有结论性的观点、方法和认识，它们都是计算科学领域多年的经验积累，随着我后续学习的不断深入，我会不断地更新对这些知识的领悟。

导论课程引导着我从科学哲学的角度去看待和学习计算科学。第一、二章的学习中，我学习了计算机的数学起源和物理起源，了解到科学哲学可以为计算科学提供理论基础与认知框架。我意识到，计算科学并非单纯撰写代码、操作计算机硬件，而是一门融合数学、逻辑学、工程学等多领域知识，旨在运用有效算法、借助计算设备高效处理复杂问题的交叉学科。导论课程以高屋建瓴的角度指导了我的学习方向。

除了计算科学领域的学习，我还在导论课程上广泛了解了与计算机科学相关的学科基本知识，对本专业的历史变革、主流研究方向有了全面的了解。比如在第三章的学习中，我了解了数字逻辑与集成电路、机器指令与汇编语言、计算机图形学、计算机网络等方面的知识。其中我对计算机图形学产生了浓厚的兴趣。计算机图形学利用计算机来研究图形的表示、生成、处理和显示，在可视化、工业建模等领域有着极其重要的作用，包括图像处理、模式识别、计算几何等方面的知识。课后，通过《计算机图形学基础教程》[10]这本著作，我初步了解了计算机图形学的基础内容。我又通过《浅谈计算机图形学与图形图像处理技术》[11]这篇文章，初步学习了图形系统的组成和功能。

此外，孙老师还带领我们详细分析了2024级培养方案。培养方案包括了培养目标、毕业要求、核心课程等等，尤其是能力支撑层次示意图让我直观地感受到了每一门课程对我的能力提升作用。希望通过大学四年的努力，我将会成为一名高级工程技术人才。此外，我还深入进行了职业规划。从全面深刻的自我分析，到精准详细的环境分析和定位，我眼前的职业发展路径更加清晰明朗，引导我逐步实现自己的梦想。

导论课程带给我的不仅仅是理论知识，还有科研的综合能力。例如，导论课程上孙老师提出的具有思辨性的问题，比如思考“数学、逻辑与新闻之间的联系”“什么是艺术”让我对所学的知识进行批判性思考，提出自己独到的见解。课堂上我积极思考，互相交流进步，这种思考训练有助于培养我们独立思考、敢于质疑的科学精神。同时，导论课程有着形式新颖、成效显著的考核方式，分组演讲、职业规划、总结报告相结合的方式，既锻炼了我的团队协作能力、书面表达能力、反思总结能力，又让我深化对课程的理解，提高了逻辑思维能力。导论课程为我后续的学习、研究打下了深厚的基础。

# 5 附录

## Github

个人网址：<https://github.com/Augendwind>

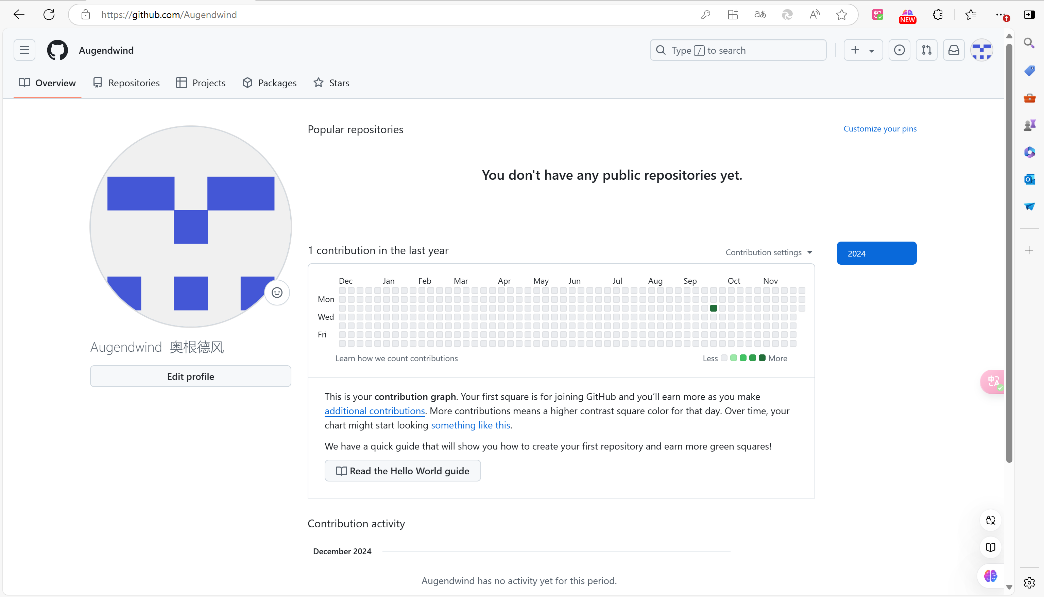


图6 Github个人网站

## 力扣

个人网址：<https://leetcode.cn/u/condescending-yalow9rk/>

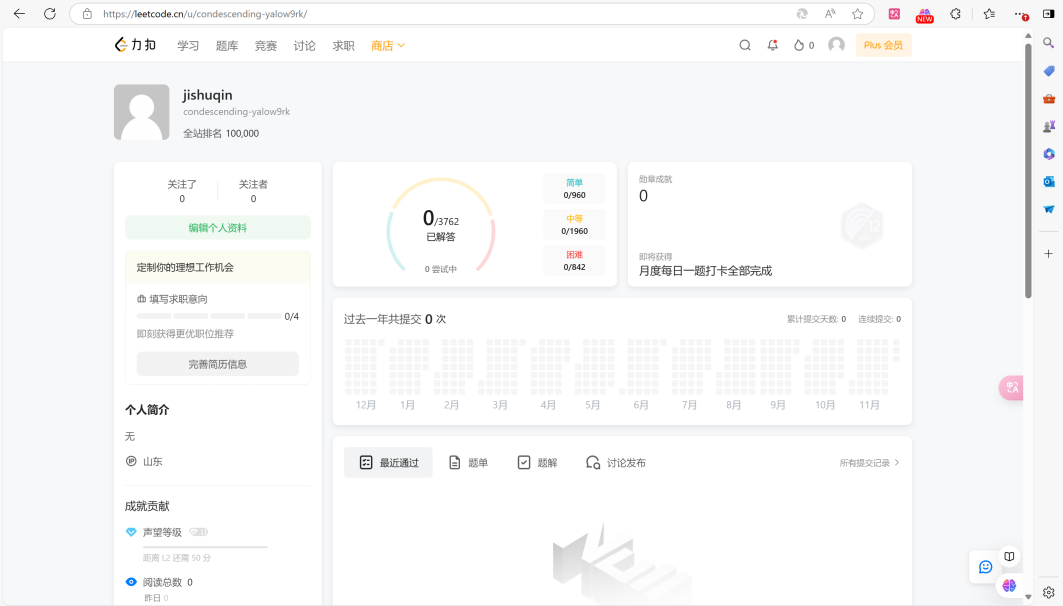


图7 力扣个人网站

## 菜鸟教程

网址：<https://www.runoob.com/>

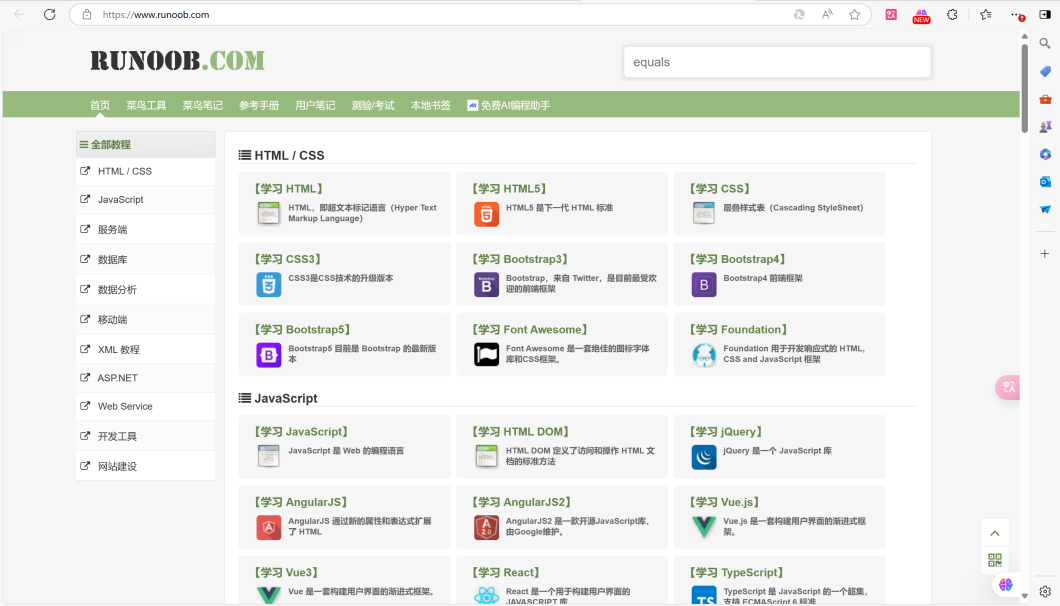


图8 菜鸟教程

## CSDN

个人网址：<https://blog.csdn.net/2404_87627565?type=blog>

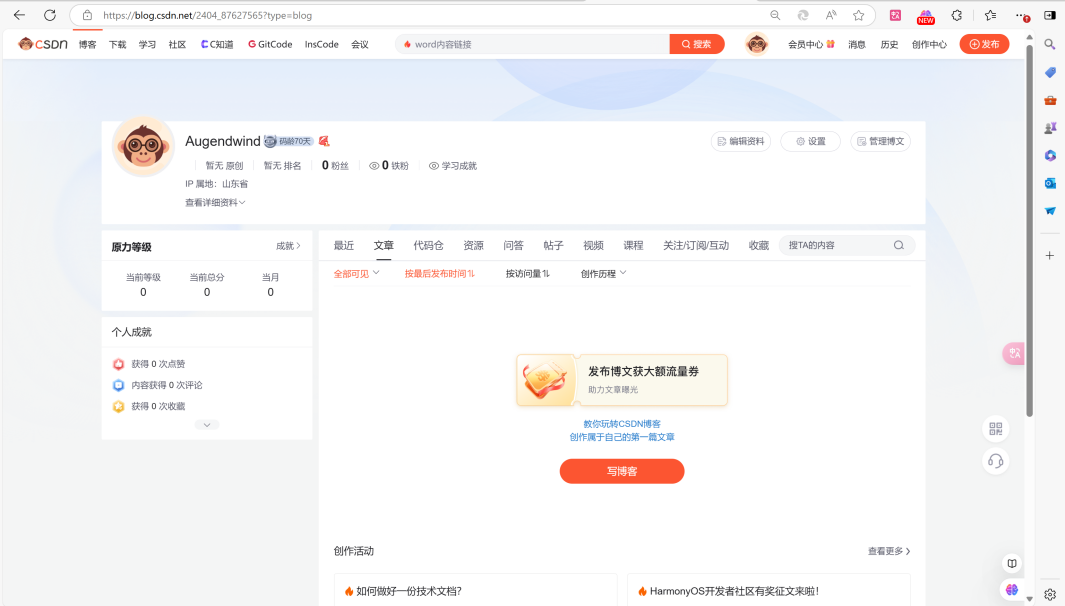


图9 CSDN个人网站

## 稀土掘金

个人网址：<https://juejin.cn/user/1141726586342042>

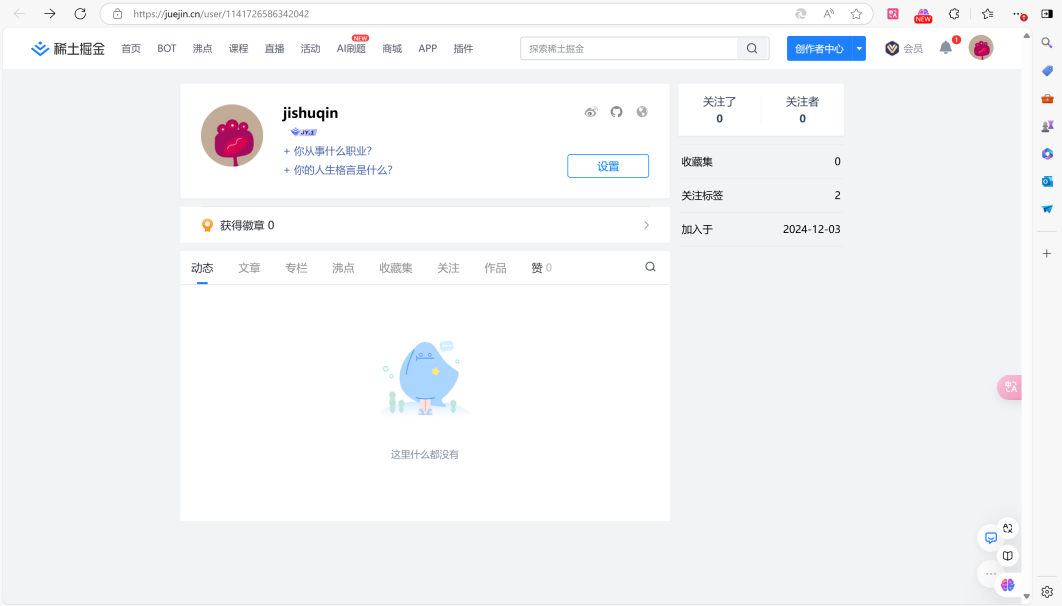


图10 稀土掘金个人网站

## 语雀

个人网址：<https://www.yuque.com/jishuqin-4exlc>

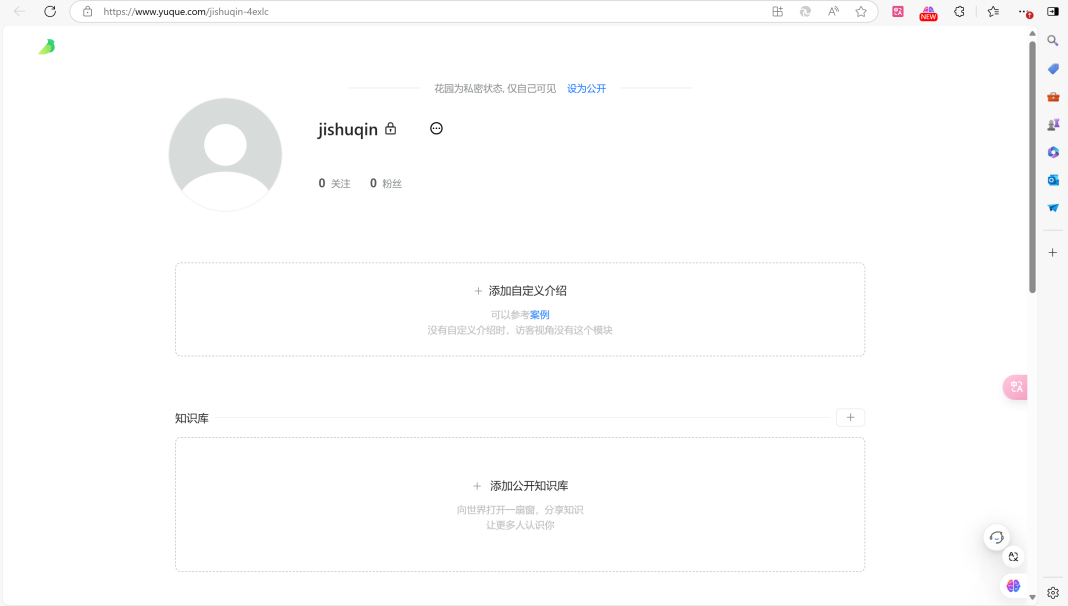


图11 语雀个人网站

## 哔哩哔哩



图12 哔哩哔哩

## 观察者



图13 观察者

# 参考文献

1. 周仁辉.计算机科学与技术的现代化应用途径分析[J].信息记录材料,2024,25(11):73-75.DOI:10.16009/j.cnki.cn13-1295/tq.2024.11.061.
2. 徐胜超,杨波,王宏杰,等.基于云计算的蛋白质折叠空间结构预测[J].电子技术应用,2024,50(08):10-16.DOI:10.16157/j.issn.0258-7998.244973.
3. 王冰彬,唐震洲.一种结合遗传算法和聚类的软件定义网络控制器优化部署机制[J].计算机工程与科学,2024,46(11):1971-1978.
4. Amini R M, Feofanov V, Pauletto L, et al. Self-training: A survey[J]. Neurocomputing ,2025,616128904-128904.
5. Zoph, Barret, et al. "Rethinking pre-training and self-training." *Advances in neural information processing systems* 33 (2020): 3833-3845.
6. ZHOU Zhi-Hua. Disagreement-based Semi-supervised Learning. ACTA AUTO MATICASINICA,2013,39(11):18711878.doi:10.3724/SP.J.1004.2013.01871
7. Blum, Avrim, and Tom Mitchell. "Combining labeled and unlabeled data with co-training." *Proceedings of the eleventh annual conference on Computational learning theory*. 1998.
8. Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada *Proceedings of the 34th International Conference on Machine Learning*, PMLR 70:2988-2997, 2017.
9. Luo Yuhang, Wu Runxiu, Cui Zhihua, Zhang Yiying, He Yeshen, Zhao Jia. Tri-training Algorithm Based on Density Peaks Clustering[J]. Journal of System Simulation, 2024, 36(5): 1189-1198.
10. 孙家广, and 胡事民. 计算机图形学基础教程. 清华大学出版社有限公司, 2005.
11. 陈敏雅, and 金旭东. "浅谈计算机图形学与图形图像处理技术." *长春理工大学学报* 6.1 (2011): 128-139.