

DeepCoder

DeepCoder是微软和剑桥大学联合开发的一种计算机算法，可以用于自行编写代码并解决简单的数学问题。该系统分成两个组成部分：代码编写算法，以及搜索潜在代码的机制。该算法发表于2016年11月的论文中。

一、Introduction

DeepCoder所用的技术叫程序合成（program synthesis），**通过截取已有软件的代码行来组成新的程序**。通过学习一系列代码片（code fragment）的输入和输出数据，DeepCoder能自动摘取出对目标任务有用的代码片。

二、Re-Implement

1.Keras: <https://github.com/dkamm/deepcoder>

2.C++: <https://github.com/dkamm/deepcoder>

智能编译相关论文：

ICML上的DeepCoder: Learning to Write Programs

arXiv上最近的AI Programmer: Autonomously Creating Software Programs Using Genetic Algorithms

三、网上评论

1. 虽然DeepCoder的技术水准和技术意义明显高于AI Programmer,但从本质上来说现阶段的人工智能自动编程还是一个“在有限时间内搜索最优解”的优化问题,只是不同的项目采取了不同搜索方法。

但我们很难完美定义所有的输入输出,这会造成学习中的歧义。举例,上面的这3个输出输出也可以用乘法计算得到,大家可以试试。所以现阶段的人工智能自动编程算法面临的最大的困难就是需要大量的人工时间来完成简单的任务,且随着任务难度上升,需要的运算时间会以指数甚至更高的速度增长。以AI Programmer为例,正确输出"hello world"所需要的时间是"hello"的5倍,而字符串长度仅上升1倍。打个不恰当的比方,现阶段的自动编程更像"猴子打字",学术叫法是"无限猴子定理"。

2. Kite相当于一个人工配对程序员。在编写代码时，它会显示用户使用的库和终端命令的示例文档。Kite甚至可以自动检测并解决用户的简单的错误和需求，使用户专注编程的整体项目，无需担心细节问题。

“在Kite，我们的目标是帮助开发人员在更短的时间内创建更好的程序，”Kite的首席执行官兼创始人亚当·史密斯(Adam Smith)说。

在耗资1700万美元的首轮融资之后，Kite推动了AI辅助编程的前沿，它使开发人员可以使用自动完成建议(类似于Gmail中的智能组合)更快地完成Python代码的完整行。

Python程序员可以使用Kite以更少的麻烦构建可转换的应用程序，而不是复制和粘贴StackOverflow，编写样板代码，并反复修改简单的错误。

自从Kite在1月份首次推出新的代码行完成引擎以来，用户在编码时的代码使用完成量增加了一倍。

通过改进其类型推断引擎，允许在名称中间输入补全，Kite将显示给用户的代码补全数量增加了40%。

这使得选择Kite的用户的代码行完成率增至以前的两倍。

Kite在PyCon上发布了支持Linux的新闻，PyCon是最大的Python用户会议，在会上他们展示了开发人员如何使用Kite消除重复工作，并在Windows、Mac和Linux环境中达到编码效率的峰值。

Kite使用来自高阶开发人员的数千个公开可用的代码源来训练它的机器学习模型。

全世界有超过40,000名Python开发人员使用Kite，目前它可用于所有流行的Python编码环境，包括Atom、Pycharm、Sublime Text、Vim和VS代码。

Kite的创始团队由多位毕业于斯坦福，牛津，MIT，伯克利等名校的工程师组成，定位于San Francisco，致力于改变人们开发产品的方式。

不过，“程序猿”们目前并不担心这种人工智能的出现会让他们失业。目前，能自动完成编程的人工智能通常还只能完成较为简单的任务，所需运行时间也很长，距离能代替人类程序员的程度还有很长的路要走。

3. 从现有软件中提取出代码片段并不难，很多人类程序员也会这样做，只需要明确每条代码的意义，并将其用于完全不同用途的另一程序。然而不同于人类程序员的是，AI能够在很大范围内全面搜索现有程序，并用独特方式整合在一起，这些是人类程序员不太容易想到的方式。此外可以肯定的是，整个编码过程也会大大加快，DeepCoder在几分之一秒内就能编写一个程序。可以想见的是，程序员的工作效率会有一个质的飞跃，从前靠人力的编程手段，“进化”到自动化的编码行为。未来的工作模式，将会是一场新的工业革命，未来的行业中，智力产出品最终也可以像工业品一样流水生产，这样一来人类就可以免除重复性的脑力劳动，投入到更有价值的事情中去，哪怕只是有空闲下来喝杯咖啡、享受一下生活呢？

四、可行性评价

这种自动编程的实现方式，难点不仅在于算法，更重要的是搜索潜在代码的机制。数据集应该有现成的相关数据集，但领域应该不小，一般都是以编程语言的类别作为划分，如Java、Python。个人评价：

目的：与我们的功能其实一个方向的两个领域，他们是代码生成，我们是代码预测或说是代码推荐。

算法：存在一定难度，目前我只试运行Keras版提供的demo。

数据集：现有的数据集领域更大，具体情况还未调研。制作数据集，难度偏高。

搜索机制：这方面我们可能一无所知。