机器辅助证明：数学的未来

- 原文标题：Terence Tao - Machine-Assisted Proofs (February 19, 2025)
- 链接：YouTube 视频链接
- 文章类别：学术讲座

摘要

本次演讲探讨了人工智能（AI）如何变革数学研究。尽管数学传统上较为保守，但机器的应用正日益广泛。演讲者回顾了机器在数学中的历史角色，包括计算、实验数学和科学计算。重点介绍了AI的三大新兴应用：机器学习（发现数据模式）、大型语言模型（LLM，辅助次要任务）和形式化证明助手（严格的证明检查，促进大规模协作）。

通过四色定理、开普勒猜想和纽结理论等例子，演讲者展示了形式化证明的演变，以及机器学习与人类智慧结合解决难题的潜力。LLM虽在算术方面有局限，但通过生成代码而非直接计算，能更可靠地应用于数学。人工智能数学奥林匹克竞赛也展示了AI在数学问题求解方面的进展。

未来，AI将在语义搜索、形式化和数据库建设中发挥关键作用。演讲者强调需要改变数学研究的工作流程，接受工具的失败率。AI还有望革新数学教学，使学习更具交互性。总之，AI正以多种方式推动数学研究的边界。

关键词

#机器辅助证明， #数学研究， #人工智能， #形式化验证， #机器学习

讲座框架

- 引言 **(0:09 - 1:26)**
  - 机器正在改变我们进行数学研究的方式。
  - 数学是一个非常传统的学科，但正在经历变革。
  - 相比于其他科学的大规模合作，数学合作规模仍然较小。
  - 机器在数学中的应用越来越广泛，但尚未出现“杀手级应用”。
- 机器在数学中的历史应用 **(1:50 - 6:12)**
  - 计算和制表：算盘、三角函数表、对数表等，现在被数据库取代。
  - 实验数学：例如，18世纪对素数的研究（素数定理的猜想）和现代的BSD猜想，都是先通过实验观察规律，再提出猜想。
  - 在线整数数列百科全书（OEIS）：一个重要的数学数据库，帮助数学家识别和关联不同的数学对象。
- 科学计算/数值计算 **(6:21 - 7:12)**
  - 进行大规模模拟、求解方程等。
  - 早在电子计算机出现之前就已存在（例如，1920年代Laurence使用人工计算机模拟水坝的水流）。

- **SAT**求解器 **(7:25 - 8:11)**
  - 可以解决某些逻辑问题（例如数独），将数学问题转化为一系列真/假陈述和假设。
- 布尔毕达哥拉斯三元数组问题 **(8:30 - 10:51)**
  - 一个通过大规模计算机搜索解决的SAT问题的例子。
  - 证明了将自然数分成两部分，无论如何划分，其中一部分必然包含毕达哥拉斯三元组。
  - 证明过程产生了巨大的数据量（一度是世界上最长的证明）。
- 机器在数学中的新应用 **(11:05 - 13:56)**
  - 机器学习：利用神经网络从大数据集中发现人类难以察觉的模式和关系。
  - 大型语言模型 **(LLM)**：如ChatGPT、Claude、Gemini等，可以辅助数学研究的次要任务，例如代码编写、文献搜索等，但目前还不能直接解决难题。
  - 形式化证明助手：非常严格的证明检查器，可以验证以特定语言编写的证明的每一步，生成证明证书。这使得大规模的数学协作成为可能。
- 形式化证明的早期例子 **(14:01 - 21:11)**
  - 四色定理 (1970s)：早期使用计算机辅助证明的例子，需要检查大量情况。2005年才有了完全形式化的证明。
  - 开普勒猜想 (17世纪提出，1990s证明)：关于球体最密堆积方式的猜想。证明过程非常复杂，涉及到对评分函数的不断调整，最终依赖计算机辅助。证明发表后受到质疑，最终在2017年完成了形式化证明。
- 现代形式化证明 **(21:18 - 25:06)**
  - 技术进步使得形式化证明变得更容易，但仍然比传统方法繁琐。
  - 多项式Freiman-Ruzsa猜想的形式化：一个由20人（多数非专业数学家）在3周内完成的例子，展示了众包形式化证明的潜力。
  - 形式化证明可以实现大规模协作，因为每个贡献都由证明助手验证，无需信任每个参与者。
- **Kevin Buzzard**的形式化费马大定理项目 **(25:52 - 26:21)**
  - 一个为期5年的项目，目标是将费马大定理的整个证明形式化。
- 大型语言模型在数学中的应用：纽结理论的例子 **(26:53 - 31:10)**
  - 纽结理论是拓扑学的一个分支，研究空间中的纽结。
  - 纽结不变量用于区分不同的纽结。
  - 机器学习被用于发现纽结的组合不变量（如signature）和几何不变量之间的关系。
  - 神经网络能够高精度地预测signature，表明存在某种关系，但具体关系需要进一步分析。
  - 通过分析神经网络，发现三个关键的几何输入，并据此提出了一个猜想。
  - 神经网络指出了猜想的不足之处，帮助研究人员修正猜想，并最终证明了该猜想。
  - 这个例子展示了理论、实验、机器学习和人类之间的互动。
- 大型语言模型的局限性和进一步应用 **(31:36 - 34:27)**
  - LLM有时能解决高难度数学竞赛题，有时却无法进行基本算术。
  - LLM擅长和不擅长的任务与人类互补。
  - LLM可用于文献搜索、代码编写、格式化等次要任务，以及提供解题思路。
  - LLM与机器学习、严格验证等方法结合，有望用于构建流体方程的爆破解。
- **LLM**与代码生成 **(34:33 - 35:51)**
  - LLM不擅长算术，直接用于数学计算不可靠。
  - 更可靠的方法是让LLM生成代码（如Python），然后运行代码解决问题。

    - 这种方法已用于改进某些数学问题，例如矩阵乘法算法。
- 人工智能数学奥林匹克竞赛 **(35:57 - 37:11)**
    - 目标是让AI在数学奥林匹克竞赛中达到金牌水平。
    - 目前，开源模型已能在中等难度的奥林匹克竞赛中取得50-60%的成绩。
    - 成功的关键在于让AI生成代码，而不是直接解决问题。
- **AI**与形式化证明语言 **(37:22 - 38:46)**
    - 研究人员开始探索让AI生成形式化证明语言（如Lean）的代码。
    - 这种方法已成功解决了一些奥林匹克级别的问题，但生成的证明非常低效和“陌生”。
- 多问题探索：代数法则 **(39:00 - 43:19)**
    - 利用AI工具同时探索大量数学问题，而不是像人类一样一次只研究一个问题。
    - 生成了4000多条代数法则，并探索它们之间的蕴含关系（共2200万对）。
    - 项目通过众包完成，产生了50多位合著者的论文。
    - 主要使用了自动定理证明器，而不是现代AI，因为预算和时间限制。
    - LLM主要用于构建图形界面。
- 未来展望 **(43:32 - 47:31)**
    - 直接让LLM解决难题目前不可行，但LLM在辅助任务中非常有用。
    - 语义搜索、形式化证明辅助、数据库建设等是未来的发展方向。
    - 需要改变数学研究的工作流程，接受工具的失败率，采用模块化的研究方法。
    - AI可以用于发现新的数学教学方法，使教科书更具交互性。
- 问答环节 **(47:37 - 59:04)**
    - 数据缺乏的领域如何应用机器学习？ 正在通过整理文献，把不成功的经验放入训练数据。
    - 难以形式化的命题有什么共同特征？ 暂时没有发现明显规律。
    - 是否尝试使用范畴论等高层次语言构建证明助手？ 有相关提议，但尚未成为主流，主要原因是大多数数学尚未以这种框架书写。

演讲录

演讲者： 数学家Terence Tao（陶哲轩）

引言

大家好！今天我将谈谈机器如何改变我们进行数学研究的方式。这是一个剧烈变革的时代，我认为，不仅对其他领域，对数学也是如此。数学是一个非常传统的学科，我们仍然在使用黑板——我的办公室里就有四块黑板。其他科学已经拥抱了“大科学”，他们有50、500甚至5000名合作者。而数学家们则有些勉强地从一对一合作，发展到三到五人的合作。

在很多方面，我们仍然沿用着几个世纪以来的方式：研究个体问题，指导个别学生。但是，变化正在发生。我们现在正以许多有趣的方式使用机器。虽然还没有到革命性的阶段，还没有出现“杀手级应用”，但这有点像互联网已经发明，但电子邮件还没有出现的时期。电子邮件是互联网的第一个杀手级应用，真正触发了大规模采用。我们还没到那一步，但已经可以看到它的到来。

今天，我将谈谈机器如何影响数学。我知道你们很多人不是数学家，所以我会以非常高的层次介绍一些数学研究的例子，不会有太多实际的方程式。

机器在数学中的历史应用

从某种意义上说，我们使用机器的时间比任何人都长。几千年来，我们一直在使用机器来辅助数学研究。这是一台机器（展示幻灯片：公元1世纪的算盘）在辅助一位数学家。不同的是，现在机器能帮助我们的规模和性质发生了变化。

那么，我们一直以来都在用机器做什么呢？

- 计算和制表： 算盘就是一个计算的例子。除了计算，还有制表。在中世纪早期，人们开始制作三角函数表和对数表，这使得某些计算变得容易得多。当然，这些表格现在已经完全过时了，因为我们有了计算器、计算机等等。但我们仍然非常依赖大量的表格，只是现在我们称之为“数据库”。

- 实验数学： 长期以来，一直存在实验数学的传统。虽然规模较小，而且老实说，它没有得到应有的尊重（与占数学99%的理论数学相比）。我认为，未来理论和实验之间将会有更平衡的关系，更接近其他科学的情况。但它是有历史的。例如，在18世纪，勒让德和高斯著名地研究了素数。高斯本人在很多方面就像一台“人类计算机”，他制作了前10万个素数的表格。他们猜想了素数定理，这个定理在几个世纪后才被证明。但是，实验是先行的。

  类似地，数论中一个主要的开放问题是BSD猜想（关于椭圆曲线）。它也是首先通过编译大量的椭圆曲线表格，计算各种感兴趣的统计数据，并通过实验注意到两个原本不相关的对象之间存在非常强的关系而被发现的。我们仍然不知道它们为什么相关，但我们坚信它们是相关的。这导致了这个猜想。

- 在线整数数列百科全书（**OEIS**）： 也许最大、最成功的数学数据库是OEIS。许多数学家，在某些领域几乎每天都在使用它。数学文献浩如烟海，如果你发现你正在研究一个数学对象，幸运的话，你知道它的名字，你可以在维基百科上查到它，或者你认识这个领域的专家，你可以问他们关于这个对象的信息。但是，经常会有两个世界各地的数学家在研究同一个对象，但他们没有意识到，因为他们叫它不同的名字。他们没有办法比较或搜索是否有人处理过这个有趣的东西。但是，数学中的许多对象都带有整数序列。例如，如果你在研究柏拉图多面体，有五个柏拉图多面体，四面体有四个顶点，立方体有八个顶点，等等。所以，你可以将特定的数字序列附加到许多数学对象上。Sloane有一个天才的想法：为什么我们不建立一个数据库，包含所有在数学问题中出现过的整数序列，或者至少尽可能完整？这个数据库现在已经发展成OEIS，包含了成千上万的序列。每天，都有数学家试图理解一个对象，你可以尝试从这个对象生成一个整数序列，在数据库中查找它，通常情况下，它或非常相似的东西已经在数据库中出现过了，你可以建立起原本无法发现的联系。

科学计算/数值计算

另一个主要用途是我们所说的科学计算，或者更通俗地说，数值计算。这就是你想到“我们用计算机做数学”时会想到的，进行大规模模拟或求解大量的方程。例如，你想模拟一个动力系统，或者找到一个多项式的根，等等。这就是科学计算。它已经存在了100多年，早于电子计算机。

可以说，第一个主要的科学计算是由亨德里克·洛伦兹在1920年代完成的，他使用一组“人类计算机”来模拟荷兰正在建造的一座水坝的流体流动。他实际上不得不发明浮点运算来运行模拟。水坝成功建成，预测结果也得到了验证。

SAT求解器

这些数值计算可以做大量的算术运算，例如大量的浮点运算等等。但它们也可以执行某些逻辑任务。如果你有一堆陈述，有些是真的，有些是假的，要么真要么假，并且你知道它们之间的一些关系，你可以尝试找出哪些是真的，哪些是假的，就像解数独或逻辑谜题一样。有一类问题叫做SAT（可满足性问题）求解器，我们有自动工具来解决这些问题。

某些数学问题可以简化为SAT问题，即存在一定数量的真/假陈述，一定数量的假设和结论。其中一些可以自动化，只要它不是太大。很多数学问题不能用这种方法解决，因为它只适用于有限数量的假设和结论，但有时它是有效的。

布尔毕达哥拉斯三元数组问题

让我给你们举一个大规模SAT问题的例子。有一个长期存在的猜想，叫做布尔毕达哥拉斯三元数组问题，它只能通过大规模的计算机搜索来解决。

问题是这样的：毕达哥拉斯三元组，如3、4、5，是直角三角形的边。但它们相当稀疏，数量不多。有3、4、5，然后是5、12、13，它们相当罕见。但人们相信，现在也已经被证明，如果你把自然数分成两部分，无论你如何划分，其中一部分都保证包含这些三元组之一。事实上，你甚至不需要划分所有的自然数，你只需要划分7825个。

无论你如何将这个集合分解成两部分，其中一部分都包含一个毕达哥拉斯三元组。另一方面，如果你只用7824个，你可以找到一种划分方法，使得两个类都不包含三元组。这部分相当容易，你只需要展示一个不包含三元组的划分，你可以用一个非常简单的计算机程序来检查。但困难的部分是第二部分，因为将这个集合分成两部分的方法有2的7825次方种，这是一个巨大的数字，你无法手动检查所有这些。这基本上就像一个有7825个格子的超级数独，但你可以证明这个数独没有解。

当时，它创造了世界上最长证明的记录。这个证明生成了一个证书，需要4个CPU年（单位有误，但大致如此）。证明最初是200TB，对我们来说，这是一个很大的数字。他们设法将其压缩到86GB，但在某个时候，它是世界上最长的证明。

机器在数学中的新应用

这是一个传统的使用计算机解决数学问题的例子。但令人兴奋的是，现在有几种新的使用计算机的方法，人们正在找到一些小众的方法，将它们融入到工作流程中。

当然，有一些平凡的方式，我们都使用电子邮件，我们都用计算机写论文，从互联网上搜索和下载东西，等等。但我对这些平凡的事情不感兴趣。

有三种新的方式正在开始变得具有变革性，但还没有达到“杀手级应用”阶段：

1. 机器学习： 使用神经网络等工具，从大型数据集中发现人类无法看到的模式和关系，或者你可能可以从更传统的统计分析中辨别出来。但问题是，大多数数学家没有接受过处理这些数据集的训练，而这些机器学习算法可以大规模地自动化这个过程，这是我们没有接受过训练的。

2. 大型语言模型 **(LLM)**： 如ChatGPT、Claude、Gemini等。首先，它们使其他工具更容易使用。如果你开始让LLM帮你做，运行机器学习或编写其中一个程序会更容易。它们也开始

解决数学中的一些简单问题，或者协助数学活动中的一些次要任务。到目前为止，它们还不擅长直接解决难题，但这可能不是部署它们的最佳方式。

3. 形式化证明助手： 这些是非常互补的工具，它们是非常严格的证明检查器。如果你用一种非常特殊的语言写一个证明，这些证明助手会检查每一行，它会说“正确”，这是一个正确的证明，或者“编译错误”，这是一个正确的证明，直到第567行，现在这是语法错误。这很像编译计算机代码，只是计算机代码生成可执行程序，而形式化证明助手，它们也是计算机语言，它们在技术上也可以这样做，但它们也可以生成证明证书。这本身就很好，但它也促成了其他事情，例如最终允许数学家进行大规模协作，这在以前基本上是不可能的，除非付出巨大的努力。

单独来看，这些工具都变得有用，但真正有希望的是将它们结合起来。这就是“杀手级应用”将要出现的地方，某种所有这些工具的综合。现在，这些工具之间的交互还不太好，但它们应该在未来实现。

形式化证明的早期例子

让我们先谈谈证明助手。计算机辅助证明从70年代左右就开始出现了。最早的著名例子是四色定理。

（此处省略了幻灯片上关于证明的细节）它涉及到生成一个大约5000个特殊图形的列表，并验证这些图形具有各种属性。在当时，其中一些属性可以用计算机验证，我指的是1970年代的穿孔卡片式计算机。其中一些可以用非常机械的方式证明，但不能完全由计算机证明。事实上，我认为Appel和Haken为这5000个图形中的每一个都写下了几行解释（实际上是Haken的女儿写的），对每个图形的某个属性进行了五六行的论证，他们必须这样做5000次。这非常容易出错，他们不得不修改了好几次。花了20年时间，才产生了一个符合现代标准的计算机辅助证明。

新证明更简单一些，它只使用了大约500个图形。但更重要的是，证明的计算部分是一个你可以陈述你想要证明的内容的陈述，然后任何人都可以花一两个小时编写一个计算机程序来验证这个特定的计算任务。你可以在一台计算机上运行几秒钟来验证它，他们提供了代码。但你仍然必须信任代码，也许编写代码的人有一个bug。如果你真的想要一个形式化的证明证书，一个严格的保证，证明这个陈述可以从数学的公理一直推导出来，那是在2005年才完成的，他们使用了证明助手Coq来给出一个完整的证明。所以你看，传统上，要达到能够形式化一个定理的程度，需要几十年的时间。

另一个著名的例子是开普勒猜想。这是17世纪的一个猜想，如果你想在空间中堆积橙子、炮弹或其他东西，有一种显而易见的方法，就是你在超市里看到的橙子的堆积方式，叫做立方中心六边形密堆积。这种堆积方式有一些空隙。球体占据的空间比例约为74%，也就是这里的$\pi/3\sqrt{2}$。开普勒猜想这是最好的方法，没有其他更好的方法来堆积这些球体，除了这种显而易见的算法。但事实证明，这非常难以证明。二维情况并不太难，但三维版本非常非常难。

有一个策略：每当你有一个堆积，它就会产生一种分解空间的方式。每个球体都有一个所谓的Voronoi单元，所有距离这个球体比距离其他任何球体都更近的点，在这个球体周围有一个特定的多面体。所以，每次你有一个堆积，你就会得到所有这些多面体，它们有各种面，各种面积和体积。你可以建立各种关系，例如，如果你能将这些接触的多面体的体积与它们的体积联系起来，你可以做各种不等式。它们也与整个系统的密度有关。所以，理论上，如果你能收集到所有这些不同的多面体之间的足够多的不等式，然后运行某种线性规划，你可能就能推导出密度的界限。

这是一个很有前途的策略。原则上，你只需要有限数量的这些东西，所以计算机有可能做到这一点，而不是整个堆积问题，那里有无限数量的这些板。人们尝试过这个，有很多失败的尝试。最终，Hales和Ferguson在90年代完成了这项工作。

问题是，正如所述的策略并不完全有效。你必须用更复杂的多面体来代替这些多面体，这些多面体没有明显的描述。他不断调整这些Voronoi单元的定义，以及体积之类的东西。他没有采用体积、面积和明显的度量，而是引入了一个分数，每个多面体都有一个数值分数，并且这个分数随着证明的进展而不断变化，因为他们尝试了一件事，它没有成功，他们又增加了一个花哨的东西，它还是没有成功。最后他们成功了，但这是一个非常有争议的证明。

Hales写道："每当我们遇到解决这个问题的困难时，我们都可以调整评分函数来应对困难。这个函数变得越来越复杂，但每一次改变，我们都可以减少几个月或几年的工作。这种不断的调整不受我的同事们的欢迎。每次我在会议上展示我的工作时，我都在最小化一个不同的函数。更糟糕的是，这个函数与早期的论文略有不兼容，这需要回去修补早期的论文。"

所以，他们最终做到了。他们选择了一个非常精心设计的分数函数，有150个变量，他们做了一个线性规划，并验证了它。他们一开始并没有使用计算机辅助证明，他们希望做一个纸笔证明，但最终他们被迫使用计算机辅助证明。在当时，这是一个相当大的证明，250页，以及3GB的各种笔记和数据。

他们把它寄给了顶级的数学期刊《数学年刊》。审稿花了四年时间（实际上，有些论文花了更长的时间，但这已经相当长了）。但审稿人无法重现计算结果，他们说他们只有99%的把握。所以，他们在论文的开头加了一个警告，一个免责声明，说编辑不保证这篇论文的正确性。他们后来从当前的在线版本中删除了这个声明。是的，当时这非常有争议。

因此，特别是Tom Hales有很多动力来真正地用一种计算机语言形式化这个证明，这样就不会有任何疑问，不会有任何小的符号错误导致整个证明无效。所以他发起了一个项目，他估计需要20年才能形式化每一步。他很高兴只花了11年。形式化证明最终在2017年发表，有很多合作者。

现代形式化证明

这就是直到最近的情况：你可以形式化，但它是如此痛苦，以至于只有对于非常非常重要的项目，你才会想这样做。但现在，技术已经有了很大的进步。还不是完美的，但我们开始看到，你不必花几年时间来做这些事情，你可以在几周内完成形式化。

最近，我和四位合著者一起证明了组合学中的一个猜想，叫做多项式Freiman-Ruzsa猜想，尽管名字如此，这个猜想实际上是由计算机科学家Kopparty和Martin提出的。猜想的内容并不重要，它只是一段数学。我们用传统的方法证明了它：我们互相交谈，交换电子邮件，在黑板上工作，等等。我们有一篇33页的论文，证明了这个结果，实际上刚刚被《数学年刊》接受了。但后来，我决定这是一个很好的项目，可以测试最新的基础设施，用更现代的证明助手语言Lean来形式化这些东西。

所以我们启动了一个项目，花了大约3周时间，有20人参与，其中大多数人我以前从未见过。正如我所说，四五个人是目前数学合作的最大规模，因为在数学中，你必须信任论证的每一步。如果有人提供了一步的证明，而你不理解它，也许其中有错误，那么整个证明可能是错误的。所以你必须验证其他人贡献的每一个贡献，这是将合作规模扩大到四五人以上的一个重要限制因素。

但是，你可以众包证明形式化，而你不能用传统的方式众包证明。现代证明形式化的工作方式是，你把一个大的、复杂的陈述分解成许多小块，许多小的引理，每一个引理都相当简单，可能

是其他部分的简单推论。在底部有一个陈述，我们证明的陈述叫做PFR猜想，它代表了底部的这个小气泡。在我截屏的时候，这是一个空的气泡，这意味着它还没有被证明。但还有其他气泡，我们陈述了所有这些其他事实，这个陈述依赖于这四个事实，其中一些我们已经证明了，一些已经准备好被证明了。有一个颜色编码，我就不讲了。但问题是，你可以众包，人们可以自愿地研究一个命题，你只需要证明这个引理可以从上面的三个东西推导出来，你不需要理解整个证明。这是组合学，但大多数贡献者都不是组合学家。事实上，许多人甚至不是专业的数学家，有计算机科学家，有来自工业界的人，有喜欢解决难题的人。我们有非常广泛的一类人来参与这项工作。每一个贡献都必须被形式化，必须被这个语言Lean认证。所以，我们可以信任所有的贡献，即使我不认识这些人，也许他们会犯错误，但如果他们犯了错误，它就不会被项目接受。所以我们能够进行这些大型的合作。

（此处省略了关于Lean代码的幻灯片）在许多方面，它比用传统的方式写东西更乏味。我想说，目前，如果你想写一个形式化的证明，它大约比用传统的方式写一个证明长10倍，使用LaTeX之类的语言。所以，在目前阶段，对于每个人来说，这并不值得。你仍然必须选择你想形式化的项目，所以我们仍然专注于相对重要的项目。

Kevin Buzzard的形式化费马大定理项目

也许目前最重要的项目是Kevin Buzzard正在进行的一个为期5年的项目，他已经进行了一年，目标是形式化费马大定理的整个证明。他估计在5年内，他可能无法形式化所有内容，但Wiles的证明是在90年代中期，他希望至少将这个陈述简化为1980年就已经知道的事实，然后还有一些工作要形式化这些事实，但这似乎是目前的目标。

大型语言模型在数学中的应用：纽结理论的例子

也许我告诉你，这就是数学正在被改变的一种方式，我们现在能够进行大规模的协作。我给你们举了一个例子，我们形式化了一个已经被证明的东西。稍后，我希望我会讲一点关于形式化还没有被证明的数学，形式化结构实际上是我们得出证明的一部分。

另一个方向是使用这些大型语言模型。这是我最喜欢的故事之一。

这是一个来自纽结理论的故事。纽结理论是拓扑学的一个分支，研究空间中的纽结。你可以制作各种各样的纽结，有些纽结彼此不同，你可以把一个纽结变形为另一个，有些是真正不同的。纽结理论中的一个基本问题是，你如何判断两个纽结是否等价。

我们做到这一点的方法之一是，我们给这些东西分配所谓的“不变量”。有一些数字，你可以分配给一个纽结，这样，如果你连续地变形纽结，这些数字不会改变。所以，如果两个纽结有不同的不变量，那么它们就不是同一个纽结。

现在，有各种不同的方法来构造不变量。有一种叫做纽结的signature，这是一个特定的整数。它来自组合学，你计算交叉点，以及你是向上交叉还是向下交叉，还有一个链接矩阵，有一些组合的配方来生成纽结。然后是这些几何不变量，你观察纽结外部的空间。这是三维空间，去除了一个纽结。这通常是一个叫做双曲空间的例子，你可以定义叫做双曲体积和双曲尖点体积的东西，有许多实数和复数与这些纽结相关联。（我不会定义这些东西）但这是一个部分数据库，包含了一堆纽结和各种与它们相关的统计数据。

所以，有两种不相关的方法来生成不变量：组合的配方和几何的配方。这两种方法之间没有已知的联系。

一组数学家（我忘记了作者的名字，这里没有列出）决定把机器学习应用到这个问题上。你能否得到一个神经网络，如果你把这些几何不变量（与这些纽结相关的20个实数和复数）输入神经网络，你能否预测signature？他们发现，有一个神经网络可以做到这一点，准确率达到90%或99%，非常非常准确。这告诉你，至少在不变量之间一定存在某种关系，但这种关系是由这个黑盒子给出的，这个神经网络，这是一个非常复杂的权重和函数的集合。所以你必须以某种方式打开这个黑盒子并理解它。

他们对这个黑盒子做了一个非常基本的分析。一旦你有了这个黑盒子，它基本上就是一个输入20个数字并输出一个预测数字的盒子。你可以转动这些旋钮，你可以改变其中一个数字，看看它对输出的影响有多大。他们发现，在这20个输入中，有17个几乎没有任何影响，但其中3个非常重要。有3个输入对输出产生了很大的影响。

这三个输入是纵向平移，以及meridional平移的实部和复部。这些是他们没有预料到会很重要的输入。例如，他们预计这个体积会是最重要的不变量，但它实际上几乎是微不足道的。

通过一些分析，他们意识到有三个输入是真正关键的。然后他们可以绘制这些输入与signature之间的关系。然后，他们可以在视觉上看到一种关系，并根据视觉上拟合一条曲线到数据上，提出了一个猜想。

一旦他们有了这个预测，他们将其与神经网络进行了比较，神经网络说：“不，这个猜想不可能是正确的，因为这里有一些纽结的例子不符合这个猜想。”但是，猜想失败的方式让他们看到了如何修复猜想，他们添加了一个额外的项，使其更准确。然后，一旦他们有了正确的猜想形式，他们就可以从理论上证明它。

所以，这真的是理论、实验、猜想、机器学习和人类之间的一场对话。他们确实严格地证明了这些事实之间的联系。

大型语言模型的局限性和进一步应用

这是数学正在改变的另一种方式，但它需要数据。使这项工作成功的原因是，已经有一个包含一百万个纽结的数据库。他们又增加了两百万个到现有的数据库中，然后他们才能够做到这一点。现在，我们仍然有很多数学领域没有足够的数据，在很多方面，我们仍然是一门“数据贫乏”的科学。

（此处省略了一些关于LLM的幻灯片）LLM有时可以解决非常困难的问题，例如，现在它们可以解决大多数高中生无法解决的奥林匹克水平的高中数学竞赛题，有时它们可以得到完美的答案。但有时它们却无法进行基本的算术运算。你可以给它一个算术问题，它会高兴地吐出一个错误的答案，然后你指出这是错误的，它会说：“对不起，我打错了。”它是模式匹配，很奇怪，LLM擅长的一组任务几乎与人类擅长的一组任务正交。我们觉得困难的事情，它们觉得容易，我们觉得容易的事情，它们觉得困难。这是一个非常奇怪的工具。

但它们本身已经对各种事情有用了。例如，如果你想快速学习一个不完全在你领域内的科目，它就像一个非常好的维基百科的交互版本。你可以做一些基本的文献搜索，你可以编写代码，你可以格式化，你可以写，如果你有一个棘手的LaTeX图像，你想制作，你可以做到这一点。有很多次要任务，这非常有用。你也可以问它如何解决一个问题，它会给你10个建议，其中7个是垃圾，2个是你已经想到的，但可能有1个是一个有趣的想法。

人们开始将所有这些不同的使用计算机的方式结合起来。

这是一个非常有前途的方法，由不同的团队构建。流体方程中最大的问题之一是，构造流体的初始条件，使流体在有限时间内发展出奇点。有一个特殊的方程叫做Navier-Stokes方程，我们真的很想构造它的解。但实际上，还有很多其他的流体方程更简单，我们很想严格地证明它们会爆炸。

似乎有一个两步过程：我们现在认为，做到这一点的方法是，首先使用机器学习和类似的工具来构造近似解，这些解几乎会爆炸，但几乎爆炸，但精度非常高，它们与真正爆炸的东西之间的距离在10个小数点以内。然后将其与一些真正严格验证的偏微分方程分析相结合，表明任何近似的爆炸解都必须接近一个精确的解。所以你可以使用任何非严格的、容易出错的AI来找到候选解，但然后你在最后将其与一些严格的验证结合起来。这对于简单的流体方程已经奏效了，我们正在慢慢地爬上更复杂的流体方程的阶梯。我认为在10年内，我们实际上可能会得到Navier-Stokes方程的解，这将是一个很大的项目，但我们已经非常接近了。

LLM与代码生成

正如你所看到的，LLM非常不擅长算术，它们会犯很多错误。所以，使用它们直接进行数学计算是非常不可靠的。奇怪的是，如果它是一个非常高级的数学，没有太多的数值计算，它会工作得更好，但数字越多，它就越容易出错。

我们正在慢慢地意识到，前进的道路实际上是让LLM不是直接做数学，而是用一种更可靠的语言（如Python，一种传统的计算语言）生成代码，然后运行该代码来解决你的问题。这似乎是一个更有前途的范式。

例如，有许多问题，任务是构造一些高维的反例，并且存在一些可能候选的高维空间，进行标准的优化或机器学习不起作用。但是，编写一个更小的计算机程序来生成一个高维的候选，运行该程序，并对该代码进行一些迭代，已经开始改进一些数学问题。例如，现在对于大型矩阵，最快的矩阵乘法算法是由于这个AI（AlphaTensor），它是由Google DeepMind创建的。

人工智能数学奥林匹克竞赛

还有这些正在进行的人工智能数学奥林匹克竞赛，我实际上是这个竞赛的科学顾问委员会成员。我们的梦想是让AI在这些奥林匹克竞赛中真正达到金牌水平。希望使用开源的东西，以及可以被其他人复制的东西，而不仅仅是需要1000万美元的计算和大量的微调。

但现在，我们有了开源模型，可以做到这一点。我们还没有在真正的奥林匹克水平上达到这样的性能。其中一个问题实际上是评分，对这些模型的输出进行评分。我们没有足够的人类评分员来评估这里的输出，所以我们不得不满足于处理具有数值解的问题，例如一个答案是三位数的问题，因为我们可以自动检查。但它们做得出奇地好。

在奥林匹克竞赛的中等水平上，它们现在可以达到50-60%的性能。同样，秘诀不是直接解决问题，而是生成代码，然后由代码来解决问题。

（此处省略了一些关于几何问题求解的幻灯片）

AI与形式化证明语言

已经取得了成功。人们开始研究如何让AI输出这些证明验证语言（如Lean）的代码，而不是输出Python。

你可以做到这一点。这是最近在解决奥林匹克级别的问题上取得的成功，通过将问题转换为Lean语句，然后使用AlphaZero程序的一个版本（这是DeepMind用来下围棋的程序），并将这些数学问题视为一个巨大的围棋游戏，你可以做某些动作，你只需要从A到B。然后他们可以证明，他们可以证明相当困难的这些奥林匹克级别的问题，经过大量的计算。

虽然证明真的很奇怪，它们完全没有效率。例如，他们解决了这个IMO问题，第一步是对数字10进行归纳，这没有任何意义。这实际上是一个你可以直接删除的步骤，它仍然是一个有效的证明。然后实际上有一些人类的评论，他们花了169步证明了一个引理，然后它被归纳，没有被使用。这是一种完全陌生的证明类型，这不是人类会写的东西，但它可以解决这些问题，我们真的不明白为什么，部分原因是我们无法访问实际的源代码或任何东西，这只是DeepMind报告的内容。但无论如何，你可以解决这些问题。

多问题探索：代数法则

我参与的一件事是，尝试使用这些工具来实际进行新的数学研究，而不仅仅是解决现有的问题。我创建了这个测试项目，这些工具可以做的一件事是对大量的数学问题进行大规模探索。数学家们现在一次只能研究一个问题，花几个月的时间，然后我们再去研究下一个问题。但现在，你可能可以尝试解决数百万个问题，不是最困难的问题，而是数百万个中等难度的问题，使用计算机。这是一种不同风格的实验数学，我认为它将变得越来越普遍。

这里有一个例子，只是作为概念验证，我们开始做的。你可能熟悉各种代数法则，如交换律（x * y = y * x）和结合律（x * (y * z) = (x * y) * z）。有些运算服从这些法则，有些不服从。

所以我们生成了4000条代数法则。例如，交换律是这个（x * y = y * x），但也许x = x * y，等等。我们生成了所有这些方程，我们给它们起了名字。其中一些蕴含着其他的。例如，如果x = x * y，那么x * y * z总是等于x * w * u，你可以使用一些代数推导，你可以使用其中一些法则来证明其他的。但有些法则是不相关的。例如，这个法则（x = x * y）并不蕴含这个法则（x * y = y * x），因为你可以构造服从这个法则但不服从另一个法则的运算。

所以我们有这4000条左右的法则，它们之间有2000万对，问题是，哪些法则蕴含着其他的？

这个项目基本上就是探索整个图。任何给定的边都相当容易，任何具有研究生水平的代数知识的人都可以手动处理其中一对，花半个小时弄清楚哪个是真的，哪个是假的。但你有2200万个这样的东西，你必须以自动化的方式来做。

我们设法做到了这一点。在大约两三个月内，我们完全确定了这个图，有些很容易，有些很难。我们事先并不知道哪些是容易的，哪些是困难的。

这是众包的，将会有一篇论文，有大约50位合著者，这可能接近数学中的一个记录。我们有这2000万个任务，所以人们可以处理其中一个、十个或一百万个，他们可以部署他们最喜欢的AI工具，并扫描所有这些蕴含关系，看看他们能做什么，不能做什么。然后，所有这些都会被上传到一个中央数据库，所有内容都必须在Lean中得到验证。所以，你不能破坏，我是说，如果其中一个出错了，就会毁掉整个事情，但一切都必须得到认证。我们的大部分合作，我们彼此都不认识。他们中的大多数人甚至不是数学家。

但是，不同的人贡献不同的东西。有些人可能会手工找出其中两个方程之间的关系。然后，另一个人会将其推广，并编写一个计算机工具（比如用Rust）来搜索所有其他的蕴含关系。然后，第三个人会采用那个计算机程序，并使其产生Lean格式的输出，以便上传。没有一个人可以完成所有这些。我们还有这些漂亮的图表，我完全不知道怎么写。

但所有这些都被整合在一起了。我们使用了...实际上，令人惊讶的是，我们没有使用太多现代AI。我们没有使用所有这些花哨的工具，部分原因是我们没有计算机预算。我们使用了更经典的工具，叫做自动定理证明器，我在开头非常简短地提到了。目前，这些大型语言模型的主要应用是围绕项目构建图形界面，但它们在实际任务本身上并不是非常成功。但我认为这只是时间问题。

但这是以前没有所有这些工具无法做到的新型数学研究。我是说，你无法用人类证明2200万个这样的蕴含关系。

未来展望

那么，在不久的将来，我如何看待所有这些工具呢？

理想的情况是，你可以把你最喜欢的问题输入ChatGPT，它会给你答案。这行不通。但是，有很多辅助任务非常非常有用。

- 语义搜索： 它已经开始擅长...你知道，你写一个非常粗略的问题："我需要一个工具，在给定这个假设的情况下控制这个随机变量，但我不知道该搜索什么。"如果它是一个相当标准的东西，它会准确地告诉你它是什么，或者它会给你...它可能不会给你正确的答案，但它会给你一些你可以搜索的关键词。

- 形式化： 正如我所说，形式化证明仍然很痛苦，形式化一个证明比手工编写它要长10倍。但是AI工具正在开始将这个数字从10减少到9，再减少到8。有很多小的AI助手可以加速将一个大证明分解成许多小块，或者填写每个小块的过程。

- 数据库： 我认为一旦我们有了更多的数据库，我们就可以用数学做更多的事情。一个很大的瓶颈是，我们必须让人们创建高质量的数学数据库，而我们没有足够的。

我认为我们需要改变，或者至少拓宽我们进行数学研究的工作流程。传统的项目是，你选择一个难题，或者也许两三个相关的问题，你朝着这个目标努力，每一步都必须正确，你必须有100%的有效性，否则你最后就没有证明。但是，所有这些工具，特别是机器学习和AI工具，都有失败率，有时它们会给你垃圾。所以，要么你需要通过一个验证器（如Lean）过滤所有内容，这是一种可能性。但更一般地说，你需要有一种非常模块化的项目，在那里，有失败率是可以接受的。例如，你不解决一个问题，你解决2200万个问题，如果AI工具只解决了其中的10%，那已经完成了200万个任务，你知道，这已经很多了，你把它加到堆里，然后你运行另一个AI。

我们目前不这样做。但是在其他科学和现实世界中，你会处理失败率。所以，我们需要以某种方式拥抱一些方法，如何使用有效的工具来仍然产生严格的结果。但似乎这是可能的。

AI与数学教学

最后，我认为可能会发生变革的是，我们可以使用这些工具来找到新的数学教学方法。

已经，AI正在被使用。很明显，它们正在被用来做家庭作业。但是，作为辅助，你知道，如果它以一种公开创建的方式完成，AI教学系统不会给你整个答案，而是...你知道，也许它会为你做所有的计算，但你仍然必须提供高级方向，或者相反，它会给你高级提示，你必须做一步一步的细节，它可以真正增强教学法。

我认为教科书可以变得更具交互性。你读一本数学教科书，如果有一个你不理解的步骤，你必须问老师如何解释，或者你需要花几个小时自己弄清楚真正的含义，特别是如果有一个错字。但是，未来的教科书应该配备AI助手，你可以问：“你能解释引理3.5吗？”你可以问非常具体的问题，你可以打开证明，也许你可以得到这个证明的Lean版本，你可以将每一句话扩展成更长的解释，根据需要扩展和收缩。已经有一些原型软件可以半自动地执行此操作。

所以，这也是我认为非常令人兴奋的。我想这些就是我看到的所有不同的事情。非常感谢！

（演讲结束）