量子搜索黑科技：Grover算法

返朴 2025年10月18日 08:44 北京

在我们的日常生活中，搜索是一项非常常见而且重要的任务。无论是查找最短的驾车路线、寻找特定书籍的内容，还是在互联网上查找信息，我们都需要通过搜索来找到我们想要的东西。

用经典计算机的语言来说，搜索问题可以被总结成一个数学问题：如何在一个集合中的众多数据中，找到满足条件的那一个？

撰文｜张文昊

自然，我们使用的搜索算法也是基于经典逻辑的，例如线性搜索或二分搜索——但是这些方法要么效率较低，要么对数据的结构有一定的要求。举个具体的例子来说，就好比我们现在有许多外形相同、标有编号的小球，需要在这堆小球中找到上面写有“ 42 ”的那一个。如果这些小球已经整整齐齐地按照数字大小排列好了，那我们自然可以使用二分法来快速定位我们要找的小球；但是如果所有的小球都乱七八糟地堆在一起，那我们似乎别无他法，只能挨个抓起来检查，期待好运气尽快把我们想要的那一个小球带到眼前。

上述情境便是无结构化搜索（Unstructured Search）的一个例子。在这一问题中，待搜索的数据集是混乱的、没有额外限制的。抽象地说，我们可以把待搜索的数据集合看作一堆二进制字符串，需要在其中找到一个满足要求的字符串——就好比我们有一大堆形状各异的钥匙，需要在其中找到能打开门锁的那一把。这一问题的数学表述是：

> **无结构化搜索问题**
>
> 已知有布尔函数 $f:\{0,1\}^n\to\{0,1\}$ ，它可以读取一个长度为 n 的二进制字符串输入，并且输出 1 （代表接受）或者 0 （代表拒绝）。我们希望找到一个使 f 能够接受的输入——也就是令其输出为 1 。

在这种表述下，全部长为 n 的二进制字符串即为所有可能的“钥匙”，共计 $N=2^n$ 把；而函数 f 是那把用于测试钥匙的“锁”。可以想见，对于经典计算机而言，只能一把钥匙接一把钥匙地尝试，没有什么办法可以做得更好了。平均看来，需要试过大约一半的钥匙才能找到想要的那一把，也就是说，这一算法的平均尝试次数是 $O(2^n/2)$ ，也即 $O(N)$ 。

但是，在量子世界中，有一种搜索“黑科技”——Grover 算法，它可以神奇地加速这一搜索过程。这一量子算法最早由美国计算机科学家 Lov Grover 在1996年提出[1]，旨在对无结构数据集的搜索任务进行加速。相比线性时间复杂度 $O(N)$ 的经典搜索算法，Grover 算法的时间复杂度仅为 $O(\sqrt{N})$ [1-3]，具备“平方加速”的特点。这意味着 Grover 算法可以比传统算法更快地找到所需的目标。

这一量子加速的效果究竟如何呢？让我们用一个例子来说明 Grover 算法的强大之处：假设我们想要暴力破解一个长度 4 字节的密码，也就是需要搜寻一个 32 位的二进制字符串。对于经典算法来说，这意味着要在 $2^{32}\approx4.295\times10^9$ 个可能的目标中大海捞针。假如我们的经典计算机每秒可以尝试 100 个密码，想要穷尽全部可能的密码组合也需要一年零四个月—— 平均看来，即使我们运气较好，长达数月的计算时间也是免不了的。但是，如果我们能够有一种平方加速的算法，相当于我们需要的操作步骤的量级可以骤降至 $\sqrt{2^{32}}\approx6.55\times10^4$ 。也就是说，即使我们每秒只能进行一步操作，也可以在一天之内破译这个密码！

那么，Grover 算法是如何做到这一点的呢？让我们简单解释一下它的工作原理。在量子计算中，我们使用量子比特（Qubits）而不是经典比特（Bits）来存储信息，而量子比特远比经典比特强大。这是因为：一个经典比特只能处于两个可能状态之一（例如，电位的高低、或者磁场的两个不同方向等等），因此可以存储单独的一个 0 或 1 的信息，但是量子比特具有一种称作“叠加性”的奇特特性：它可以“同时”处在两个可能状态！严格地说，量子比特会处在 $|0\rangle$ 和 $|1\rangle$ 两个量子状态的“叠加态”，称之为 $|\psi\rangle$[2]：

$$|\psi\rangle=a|0\rangle+b|1\rangle$$

这里的 a 和 b 都是复数，而且满足 $|a|^2+|b|^2=1$ 。这个要求其实正是对于测量概率的要求：当我们想去观察这个量子比特是处在 $|0\rangle$ 还是 $|1\rangle$ 时，我们恰好会以 $|a|^2$ 的概率测得 $|0\rangle$ ，以 $|b|^2$ 的概率测得 $|1\rangle$ 。而且，这里的概率和我们直观理解的概率不同：这些测量概率并不是说量子态“有概率”处在两个状态之一，只是我们不确定是哪种情况——量子态本身就确定性地处于 $|\psi\rangle=a|0\rangle+b|1\rangle$ 这个量子态上！仅仅当我们去“测量”这个态的时候，它才会依概率坍缩至 $|0\rangle$ 态或者 $|1\rangle$ 态上。

这种量子叠加性虽然和我们日常生活中的直觉相悖，但它却是量子世界里真实存在的一种性质。也正是这种神奇的特性，使量子计算机能够比经典计算机更为高效地进行并行计算。试想：如果我们想要用一个经典计算机上的函数 $f(x)$ 来计算 $f(0)$ 和 $f(1)$ 的函数值，那么需要分别把 0 和 1 作为输入，调用函数 $f(x)$ 来作两次运算。但是，如果我们想要用量子计算机上的函数 $f(x)$ 来作这一计算，只需要将 $|0\rangle$ 和 $|1\rangle$ 的叠加态输入进去，那么只需调用一次函数 $f(x)$ ，便可以同时获取 $f(0)$ 和 $f(1)$ 的信息了！

Grover 算法正是利用了量子并行的思想进行搜索：假设我们要开的“锁”是一个量子计算机上的函数 f ，而我们想要找到对应的“钥匙”——一个特定的二进制字符串 x_0 ，使得 $f(x_0)=1$ ，那么我们无需依次将不同的 x 作为输入，而可以将全部字符串 $|x\rangle$ 的叠加态作为输入：那么输出结果中实际就包括了全部二进制字符串 x 的 $f(x)$ 信息了！接下来，我们只需要想办法在结果中寻找到那个为 1 的 $f(x)$ 就行了！

不过，说来简单，但是要从量子计算机的输出结果中得到我们所需的答案，还是件麻烦事：因为量子计算机的输出结果实际上也是由 $N=2^n$ 个量子态叠加而成的！即使这里面

包含了最终结果 x 的信息，但是这部分信息也只占据全部输出结果的 $1/2^n$ 。想在这里面找到特定的一条信息，仍然无异于大海捞针。因此，Grover 算法实际上还采用了处理量子态的另一项"黑科技"：振幅放大技术[5,6]。

还记得我们之前提到的量子态 $a|0\rangle + b|1\rangle$ 吗？我们对其测量的时候，测得 $|0\rangle$ 和 $|1\rangle$ 的概率分别是 $|a|^2$ 和 $|b|^2$ 。也就是说，这里的系数 a 和 b 并不直接表示概率！用量子物理的语言来说， a 和 b 被称作"振幅"。那么，对于我们刚刚得到的输出结果来说，我们希望找到的部分是其中有关于 $|x_0\rangle$ 的，而"找到它"的概率是 $1/N$ ——也就是说，当我们对输出的量子态进行测量时，测得 x_0 的概率仅有 $1/N$ 。但是别忘了：它的"振幅"其实是 $1/\sqrt{N}$ 而不是 $1/N$ ！因此，这个量子态实际上长这个样子：

$$\frac{1}{\sqrt{N}}|x_0\rangle + \sum_{x\neq x_0}\frac{1}{\sqrt{N}}|x\rangle$$

概率与振幅之间的平方关系正是 Grover 算法能够实现平方加速的理论基础[5]。试想如果某个实验结果的发生概率为 $1/N$ ，如果我们想要用经典的方式来高概率地获取这一结果，可以去多次重复实验，那么预期需要大约 N 次实验才能够得到想要的结果。或者说，大约需要的操作次数为 $1/N$ 的倒数，才能把概率从 $1/N$ 放大到接近 1 。但是如果我们在一个量子态中能够以 $1/N$ 的概率测量得到想要的结果，那正说明这一测量结果的振幅为 $1/\sqrt{N}$ ——我们便可以通过数学操作来想办法直接放大振幅：容易想见，我们只需要大约 $1/\sqrt{N}$ 的倒数，也就是约 $\sqrt{N}$ 次操作，便可以把振幅从 $1/\sqrt{N}$ 放大到接近 1 。这一数学操作便是前述的"振幅放大技术"，有兴趣的读者可以进一步阅读更严格的数学推导[2,3]。最后，我们只要对最终的量子态进行测量，便可以很高概率地得到 $|x_0\rangle$ ——因为这一量子态的振幅已经被我们放大到很接近 1 了。

振幅放大技术除了用于高效解决搜索问题以外，还广泛用于许多量子算法中，或者作为一些量子算法的子步骤来提供平方加速[5,6]。然而，现实中的量子算法并没有理想中那么美好：这是因为，虽然 Grover 算法以及其他使用振幅放大技术的算法可以在理论上实现高效加速，但是在实际应用中仍然面临挑战。首先，我们在应用 Grover 算法时，假设了我们可以很容易地将函数 $f(x)$ 在量子计算机上实现——或者说，编码（Encode）到量子线路上，而具体如何编码、编码是否容易实现等等问题都暂时被我们忽略了；另一方面，即使假设我们已经成功编码了实现 $f(x)$ 的量子结构，调用它的时间消耗也可能会很大。这些困难都没有在前述分析中考虑到。实际上，这种能够实现特定函数 $f(x)$ 的量子装置被称为量子黑箱（Oracle），在进行理论分析时，我们往往不去考虑如何实现这一个量子黑箱，而只需假定我们可以任意调用这样的黑箱，喂给它输入，它便可以吐出相应的输出，而无需关心它内部是怎样的结构。用量子黑箱的调用次数来分析得到的算法复杂度也称为量子黑箱复杂度（Oracle Complexity）。因此，我们通常提到的 Grover 算法的 $O(\sqrt{N})$ 的复杂度实际上并非真实操作的时间消耗，而是黑箱复杂度；而具体实现黑箱操作的时间和资源消耗其实被隐藏在这一看似简洁的表达式后面了。

除了这一理论上的困难之外，Grover 算法的实际应用还面临着更严重的阻碍。在真正的量子计算机上，Grover 算法或者其他量子算法的实现仍然受到当前量子硬件发展水平的掣肘。即使对于目前最先进的量子计算机来说，其可靠性、稳定性和能够处理的量子比特数目水平也远远未达到能大幅超越经典计算机的表现[4]。近年来，量子科学家们主要的关注点还是在如何将规模和稳定性有限的量子计算机与经典优化算法相结合，从而高效地解决一些实际问题。可以说，我们目前在量子计算机领域的发展阶段还处在 NISQ 时代——即中等规模量子比特数目、且由于有噪声，无法实现可靠量子计算的时代（Noisy Intermediate-Scale Quantum Era），距离实现容错量子计算（Fault-Tolerant Quantum Computing）的量子设备问世还有很长的路要走。

不过，尽管在近期稳定实现的机会渺茫，Grover 算法仍然是量子计算领域的一个重要里程碑。它展示了量子计算的巨大潜力，并为未来的量子算法研究提供了宝贵的经验和启示。随着量子软硬件技术的不断发展和成熟，我们可以期待看到 Grover 算法及相关的思路发展出更多的应用，并且在今后的容错量子计算时代，真正得以在量子计算机上稳定实现，为未来的高速计算带来无限可能。

参考文献

[1] Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. In Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC).

[2] Nielsen, Michael A., and Isaac L. Chuang. Quantum Computing and Quantum Information. Cambridge University Press, 2000.

[3] Watrous, J. Lecture Notes on Quantum Computing and Quantum Information. https://johnwatrous.com/wp-content/uploads/2023/08/QC-notes.pdf

[4] Mandviwalla A, Ohshiro K, Ji B. Implementing Grover's algorithm on the IBM quantum computers[C]//2018 IEEE international conference on big data (big data). IEEE, 2018: 2531-2537.

[5] Yuan, Xiao. Lecture Notes on Quantum Information.

[6] Lin L. Lecture notes on quantum algorithms for scientific computation[J]. arXiv preprint arXiv:2201.08309, 2022.

本文经授权转载自微信公众号"北京大学前沿计算研究中心"。

特 别 提 示

1. 进入『返朴』微信公众号底部菜单“精品专栏”，可查阅不同主题系列科普文章。
2. 『返朴』提供按月检索文章功能。关注公众号，回复四位数组成的年份+月份，如“1903”，可获取2019年3月的文章索引，以此类推。

找不到《返朴》了？快加星标！！

长按下方图片关注「返朴」，查看更多历史文章

微信实行乱序推送，常点“在看”，可防失联