

برتریِ کوانتومی، کوانتوم برتری

سید سجاد کاهانی
sskahani@ce.sharif.edu

۲۷ مهر ۱۳۹۸

ویراستاران: علی بهجتی، امیررضا نگاری، مهرگان درودیانی
خبری که احتمالاً شنیده‌اید، این است که گوگل به برتریِ کوانتومی رسید، چیزی که به نظر می‌رسد
اتفاقِ شگرفی باشد. حداقل از حجمِ خبرهایی که در این باره نوشته شده این طور به نظر می‌رسد.
به طور خلاصه خبر این بود که گوگل، با کامپیوتری که قبلاً ساخته بود، توانسته کاری تقریباً غیرممکن
را انجام دهد و از درستی نتیجه آن مطمئن شود. کارِ تقریباً غیرممکن یعنی کاری که هیچ کامپیوترِ
معمولی ای نمی‌تواند به این زودی‌ها انجامش دهد.
جزئیاتِ خبر، حتی اگر از نظرِ ریاضی برای ما قابلِ فهم باشد، برای یک مهندسِ کامپیوتر چندان
هیجان‌انگیز به نظر نمی‌آیند. اما همچنان برای همه ما جذاب است که سر از کارِ این کامپیوترهای
کوانتومی دریاوریم. برای همین در چند قصه بی‌ربط به هم، از بنیان‌های مکانیکِ کوانتومی، به مسئله
می‌رسیم.

۱ گربه آقای شرودینگر



شکل ۱: از [۹]

فیزیک کوانتوم، تاریخی پیچیده و طولانی‌ای دارد که گفتش حداقل به ملموس‌تر شدن مطلب، کمک زیادی می‌کند. اما به خاطر بی‌سوادی نویسنده در تاریخ کوانتوم، خود کوانتوم و البته برای ایجاز، از گفتن آن می‌پرهیزیم.

داستان را با گربه مظلوم آقای شرودینگر شروع می‌کنیم^۱ که در جعبه‌ای است که هیچ ارتباطی با بیرون ندارد. در آن جعبه، یک ظرف سم وجود دارد که با یک تابش رادیواکتیو (یا هر پدیده کوانتومی‌ای که شما دوست دارید) فعال می‌شود. از مکانیک کوانتومی برمی‌آید که این گربه حالا هم مرده و هم زنده است. (یا به عبارت بهتر برهم‌نهی دو حالت مرده و زنده)

برای این قسمت خوب است با فضای برداری آشنا باشید. فضای برداری را می‌توانید از یک جزوه جبر خطی یا یک ویدیوی یوتوب یاد بگیرید. یا حتی اگر می‌خواهید خیلی خیلی بیشتر بدانید فضای هیلبرت را یاد بگیرید.

برای نمایش ریاضی این وضعیت، یک فضای برداری دوبعدی مختلط بگیریم که پایه‌های آن e_1 (به معنای سلامت گربه) و e_2 (به معنای پرپر شدن گربه) هستند که برهم عمودند،^۲ حالا گربه در حالت

^۱ درباره مظلومیت گربه آقای شرودینگر همین بس که هدفش بیان تناقض در تفسیر کپنهاگی مکانیک کوانتومی بوده [۱۲] که اکنون ما دقیقاً برای آموزش تفسیر کپنهاگی از آن بهره می‌جویم.

^۲ اگر بخواهیم مته به خشخاش^۳ بگذاریم، در اصل باید حالت‌های کل سیستم داخل جعبه، یعنی سم و گربه را به شکل توهمان بیان کنیم. یعنی e_1 می‌شود «گربه سالم است و سم منتشر نشده» و e_2 هم حالت «گربه پرپر شده و سم منتشر شده»

^۳ محمدامین خشخاشی‌مقدم، ورودی ۹۳ مهندسی کامپیوتر

زیر قرار دارد.

$$\text{state} = \frac{1}{\sqrt{2}}(\vec{e}_1 + \vec{e}_2)$$

که به احترام آقای دیراک برای نشان دادن بردارهایمان به جای \vec{v} از $|v\rangle$ استفاده می‌کنیم. در حالت کلی، این‌طور می‌گوییم که اگر سیستمی قبلاً، یکی از حالت‌های $Q = \{q_1 \dots q_n\}$ را به خود اختصاص می‌داد، در مکانیک کوانتومی، حالت سیستم، برداریست با اندازه یک در فضای مختلطی که اعضای پایه متعامدش $|q_1\rangle \dots |q_n\rangle$ هستند، یعنی عضوی از مجموعه \mathbb{C}^Q . همین‌طور، گذار خودبه‌خودی یک سیستم، که قبلاً می‌شد با تابعی از Q به Q بیان شود، در مکانیک کوانتومی با یک تبدیل خطی در فضای برداری حالت‌ها بیان می‌شود. این تبدیل اندازه بردار را حفظ می‌کند (چون گفتیم هر بردار حالتی اندازه‌ای برابر یک دارد) پس یکانی‌ست. اکنون اگر در جعبه را باز کنیم با یکی از این دو واقعیت روبه‌رو می‌شویم که گربه زنده است یا گربه پرپر شده. یعنی هر بردار دلخواهی که وضعیت گربه پیش از باز شدن بوده، باید به یکی از پایه‌ها تبدیل شود.^۴ با بی‌خیال شدن جزئیات و کلیات، احتمال این‌که بعد از باز کردن در جعبه، گربه از حالت $|v\rangle$ به عضو پایه $|q_i\rangle$ تبدیل شود و حالت q_i را مشاهده کنیم، می‌شود

$$\Pr(q_i) = |\langle v|q_i\rangle|^2$$

که این علامت $\langle a|b\rangle$ همان ضرب داخلی‌ست.

۲ از دکتر آدام تا دکتر وزیرانی

برای این قسمت خوب است با ماشین تورینگ و نمایش ریاضی آن آشنا باشید. برای این کار هم ویدیویی از یوتوب نگاه کنید و هم از روی جزوه یا ویکی‌پدیا، فرم ریاضی آن را ببینید. همچنین خوب است کلامی مسئله‌های \mathcal{P} و \mathcal{NP} را هم از روی یک ویدیوی یوتوب یاد بگیرید.

از ماشین تورینگ، همین را به خاطر بیاورید که تابعی داشت که از $Q \times \Gamma$ که حالت‌ها و الفبای ورودی روی نوار هستند، به $Q \times \Gamma \times \{L, R\}$ می‌رفت. این تابع گذار، مشخص می‌کرد که وقتی در حالتی هست و علامتی را روی نوار می‌بیند، به چه حالتی برود، چه علامتی در جایگاه فعلی نوار بنویسد و به کدام سمت روی نوار حرکت کند. حالا برای کوتاه‌نویسی این دوتا را تعریف می‌کنیم.

$$A := Q \times \Gamma$$

^۴ رمبیدن که ترجمه collapse است فعل صحیح‌تری از تبدیل شدن است، اما برای حفظ خوانایی از آن استفاده نمی‌کنیم.

$$B := Q \times \Gamma \times \{L, R\}$$

یکی دیگر از این ماشین‌های خیالی، ماشین تورینگ احتمالاتی است. تابع گذار این ماشین، احتمالاتی است. یعنی برای هر مرحله، تاس می‌اندازد. به شکل ریاضی‌تر، اگر برای تابع گذار ماشین تورینگ داشتیم $\delta : A \rightarrow B$ حالا داریم که

$$\delta : (A \times B) \rightarrow [0, 1]$$

و این تابع، به ازای هر A یک توزیع احتمال است که یعنی برای هر عضو A مانند a داریم

$$\begin{cases} \forall b \in B : \delta(a, b) = \Pr(b) \geq 0 \\ \sum_{b \in B} \delta(a, b) = \sum_{b \in B} \Pr(b) = 1 \end{cases}$$

اگر چیزی را این ماشین تورینگ احتمالاتی بتواند با احتمال بیش‌تر از $\frac{2}{3}$ تشخیص دهد، می‌گوییم آن چیز عضو کلاس BPP است و همین اول می‌شود حدس زد که $\mathcal{P} \subseteq BPP$.^۵ حالا ماشین تورینگ کوانتومی، احتمالاً یک تابع گذار به شکل یک تبدیل خطی خواهد داشت که اندازه را حفظ می‌کند. آن را به این صورت تعریف می‌کنیم. [۷]

$$\delta : A \rightarrow \mathbb{C}^B$$

این تابع به هر حالت A یک بردار مختلط در فضایی با پایه‌های B نسبت می‌دهد. اما می‌دانیم که می‌توانیم در برهم‌نهی‌ای از حالت‌های A باشیم، یعنی حالتان برداری به نام $|\psi\rangle$ عضو \mathbb{C}^A باشد. که به شکل زیر قابل نوشتن است

$$|\psi\rangle = \sum_{a_i \in A} \beta_i |a_i\rangle$$

حالا در تبدیل حالتی که اتفاق می‌افتد، به شکل خطی، هریک از مؤلفه‌های $|a_i\rangle$ به بردار مربوط، یعنی $\delta(a_i)$ می‌رود. پس گذار به این شکل انجام می‌شود.

$$|\psi\rangle \rightarrow \sum_{a_i \in A} \beta_i \delta(a_i)$$

درباره حفظ شدن اندازه بردار حالت نیز باید بگوییم اگر حالت‌های یک ماشین (شامل محل نشان و وضعیت نوار و وضعیت ماشین و چیزهای دیگر) را مجموعه S بنامیم، در هر مرحله، تحولی که صورت می‌گیرد، باید اندازه بردارهای فضای \mathbb{C}^S را حفظ کند.

می‌دانیم که در کوانتوم، در انتها، با اندازه‌گیری (باز کردن جعبه)، بردار حالت به یکی از پایه‌ها تبدیل می‌شود، که این یک فرایند احتمالاتی است. پس برای همین، مشابه کلاس BPP ، تعریف می‌کنیم کلاس

^۵ یکی از مسئله‌های جالب، با ته‌مایه‌هایی از فلسفه این است که آیا این دو کلاس مساوی هستند یا نه و اگر باشند و نباشند هریک چه معنایی دارد. توضیحات بیشتر را در این مقاله [۸] ببینید.

BQP ، شامل چیزهایی است که یک ماشین کوانتومی با احتمال بیشتر از $\frac{2}{3}$ تشخیص می‌دهد. آقایان برنشتاین و وزیرانی اثبات کرده‌اند که $BPP \subset BQP$. [۵] و همچنین اثبات شده که $NP \not\subseteq BQP$. [۴]

۳ برتری کوانتومی، یک برخورد صادقانه و مهندسی

مقالات مختلف تئوری [۲] و عملی [۶]، به خاطر تفاوت دیدگاه‌ها، معمولاً تعاریف و شروط مختلفی را برای برتری کوانتومی می‌گویند، اما به طور کلی، می‌توان این دو شرط را بیان کرد که البته هرکدام قابل بحث هستند

محاسباتی طراحی و انجام شود که:

- با یک کامپیوتر کوانتومی قابل انجام باشد ولی با کامپیوتر معمولی غیر قابل انجام باشد. (مثلاً زمان زیادی طول بکشد یا حافظه زیادی لازم داشته باشد)
- جواب مسئله قابل ارزیابی باشد.

اگر مسئله‌ای عضو BQP باشد و در عین حال عضو NP باشد (و طبیعتاً عضو P نباشد)، به این معنی است که مطمئنیم وقتی اندازه مسئله از حدی بزرگ‌تر شود کامپیوترهای کوانتومی آن را سریع‌تر حل می‌کنند و همچنین در زمان تقریباً کوتاهی می‌توانیم پاسخ این مسئله را با یک کامپیوتر معمولی ارزیابی کنیم.

پس احتمال می‌دهیم مسئله‌هایی از این دست نظیر «تجزیه عدد به عوامل اول» یکی از کاندیداهای خوب برای نمایش برتری کوانتومی باشند. اما در حقیقت، بیشتر طرح‌هایی که برای نمایش برتری کوانتومی ارائه می‌شوند، روی مسئله‌های متفاوتی نظیر نمونه‌برداری از یک توزیع تمرکز دارند. چرا؟ [۲]

- کامپیوتر کوانتومی همه‌کاره بدونِ نويز درست‌حسابی نداریم.
- فکر می‌کنیم که مسئله تولید اعداد تصادفی خیلی خیلی سخت‌تر از یک تجزیه عدد برای کامپیوترهای معمولی است.
- البته که علایق فیزیک‌دانان در طرح این مسائل دخیل بوده است.

۴ مسئله مدار تصادفی

یک مدار منطقی معمولی، یک تابعی است که از $\{0, 1\}^N$ به $\{0, 1\}^M$ می‌رود. حالا یک مدار منطقی کوانتومی (که لزوماً برگشت‌پذیر هم هست) یک تبدیل خطی یکانی است، در فضایی که پایه‌هایش $\{0, 1\}^N$ ،

یعنی رشته‌های N -بیتی اند. به این فضا، فضای N -کیوبیتی و به این مدار نیز مدار N -کیوبیتی یا گیت N -کیوبیتی می‌گوییم.

فرض کنید که حالت $|\psi_0\rangle$ را به عنوان ورودی به مدار کوانتومی U داده‌ایم، سپس اندازه‌گیری‌ای کرده‌ایم که نتیجه‌ش یکی از اعداد $1 \dots 2^N$ مانند q است. احتمال هر خروجی را $\text{Pr}_{\text{ideal}}(q|U)$ می‌نامیم.

همچنین در نظر بگیرید، برای ساخت مدار U ، مدارهای یک‌ورودی-یک‌خروجی و دوورودی-دو‌خروجی را به شکل تصادفی ترکیب کرده‌ایم. یعنی با ترکیب کردن تعدادی از مدارهای ساده و کوچک -که مشخص هستند-، با یک الگوی تصادفی، یک مدار تصادفی بزرگ ساخته‌ایم.

اگر به این مدار تصادفی U ، ورودی‌ای بدهیم و خروجی آن را اندازه بگیریم، q عددی از توزیعی خاص خواهد بود. نمونه‌برداری از این توزیع، مسئله ماست.

$$\text{Pr}(q) = \sum_U \text{Pr}_{\text{ideal}}(q|U) \text{Pr}(U)$$

شبیه‌سازی این فرایند و نمونه‌برداری از توزیع آن، توسط کامپیوترهای معمولی کار بسیار مشکلی خواهد بود. [۶] این به نظر می‌رسد که فوق‌العاده است، اما باید به سؤالات پاسخ داد:

- از کجا مطمئنیم این کار برای یک کامپیوتر معمولی به اندازه کافی مشکل است. شاید فقط ما بلد نیستیم.

- از کجا مطمئن شویم واقعاً کامپیوتر کوانتومی مان درست کار می‌کند و نمونه‌ها واقعاً از توزیع مذکور هستند؟

در پاسخ به سؤال اول، باید صادقانه بگوییم که نمی‌دانیم. [۲] این بیشتر شبیه یک تحدی است که ادعا می‌کنیم این محاسبات را هیچ‌کسی نمی‌تواند انجام دهد و باید منتظر بمانیم و شاید مردی از خویش برون آید و الگوریتمی نو دراندازد و کاری بکند. پاسخ دادن به سؤال دوم اما چندان آسان نیست (بوی پیچاندن می‌آید)، پاسخ کوتاه این است که نمی‌توانیم مطمئن شویم [۱] و پاسخ بلند خودش قصه طولانی‌ای است که در ادامه می‌گوییم.

۵ و ارزیابی‌ش

برای این قسمت، لازم است که آمار و احتمال بلد باشید. تقریباً به اندازه درس آمار و احتمال که به شکل مرسوم ارائه می‌شود.

اگر مدار تصادفی U که در قسمت قبل گفتیم، با احتمال F ، با خطا عمل کند، آنگاه می‌توان این فرایند را به این صورت بیان کرد

$$\text{Pr}_{\text{experiment}}(q|U) = F \text{Pr}_{\text{ideal}}(q|U) + (1 - F) \text{Pr}_{\text{error}}(q|U)$$

که در Pr_{error} توزیع نتیجه آزمایش در صورت بروز خطاست.
با دو فرض زیر می توان سنجه ای میزان خطا پیدا کرد.

- با توجه به این که این احتمالات ناشی از رخداد های کوانتومی هستند خطاهایی که مدار های مختلف رخ می دهند، در مجموع همه مدار ها، اریب نباشند و تقارنی نسبت به q ها داشته باشند. یعنی به طور میانگین (برای همه U ها) این توزیع، یکنواخت باشد. [۳]

$$\mathbb{E}_U[\text{Pr}_{\text{error}}(q|U)] = \frac{1}{2^N}$$

- با توجه به شیوه ای که این مدار های تصادفی را می سازیم، نتایج اندازه گیری از توزیع پورتر-توماس^۶ تبعیت کند.

این توزیع بیان می کند که برای یک q دلخواه، U یک متغیر تصادفی ست پس $p = \text{Pr}_{\text{ideal}}(q|U)$ نیز یک متغیر تصادفی ست که تابع چگالی احتمال آن به شکل زیر است^۷

$$f(p) = 2^N e^{-2^N p}$$

با این دو فرض، می توانیم به این معادله برسیم

$$\mathbb{E}_U \left[\mathbb{E}_{\text{experiment}} [2^N \text{Pr}_{\text{ideal}}(q|U) - 1] \right] = F$$

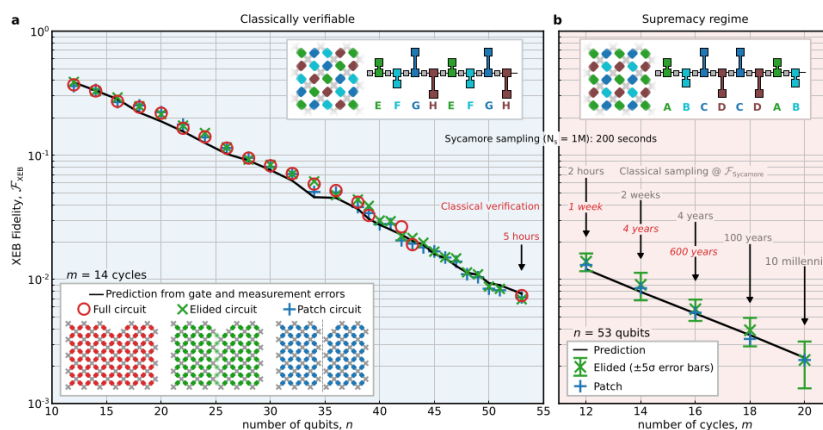
و این یعنی با دانستن مقدار $\text{Pr}_{\text{ideal}}(q|U)$ و با نمونه برداری از آن، به ازای U ها و q های مختلف (برای S_U تا مقدار از U و هر کدام با S_q مقدار از q)، طبق قانون اعداد بزرگ، تخمینی از F را به دست می آوریم.

$$\frac{1}{S_U S_q} \sum_U \sum_q (2^N \text{Pr}_{\text{ideal}}(q|U) - 1) = F + \mathcal{O}\left(\frac{1}{\sqrt{S_U S_q}}\right)$$

و همان طور که پیش تر پارامتر F را تعریف کرده ایم، احتمال سلامت آزمایش است، که معیاری از کیفیت انجام آزمایش است.
اما حساب کردن $\text{Pr}_{\text{ideal}}(q|U)$ مستلزم شبیه سازی کامل مدار در کامپیوتر معمولی ست. این یعنی تا جایی که امکان شبیه سازی مسئله در کامپیوتر وجود دارد، می توانیم آن را ارزیابی کنیم.

^۶ این توزیع که در ابتدا از داده های یک آزمایش مربوط به نوسان های واکنش های هسته ای به دست آمده است [۱۱] و بعدها در مسئله های مختلف آشوب کوانتومی مورد توجه قرار گرفته، با توجه به آشوبناک بودن این مسئله و نتایج شبیه سازی ها [۶] برای حالت ایده آل این آزمایش (با تعداد مراحل کافی) معتبر می باشد.
^۷ این که $f(p)$ برای $p > 1$ مقداری بزرگ تر از صفر دارد و این شاید نادرست به نظر برسد، اما در شرایطی که $2^N \gg 1$ [۳]، این مقدار بسیار ناچیز و تقریباً برابر با صفر است. البته که این صحبت دقیق نیست و تنها برای تقریب ذهن است.

ولی داستان به همین جا ختم نمی‌شود. گوگل، ابتدا از طریق بررسی خطاها، پیش‌بینی اولیه‌ای از کیفیت نمونه‌برداری در شرایط مختلف (تعداد کیوبیت‌ها و تعداد مراحل شرایط مسئله هستند) به دست می‌آورد. سپس، چند نوع مدار تصادفی ساده‌شده نیز طراحی می‌کند و در نهایت نشان می‌دهد که کیفیت مدارهای ساده‌شده با مدار اصلی با پیش‌بینی برابر است. سپس، با افزایش تعداد مراحل (سخت‌تر کردن مسئله) مدارهای ساده‌شده همچنان منطبق بر شبیه‌سازی‌ها هستند اما ارزیابی کیفیت مدار اصلی دیگر غیرممکن است و این یعنی به برتری کوانتومی دست پیدا کرده‌ایم. [۳]



شکل ۲: شکل برتری کوانتومی در گزارش گوگل [۳]

۶ کازینوی مونت کارلو

برای این قسمت لازم است با مفهوم تانسور، ضرب تانسوری، ضرب ماتریسی و روش‌های نمونه‌برداری آشنا باشید. دانستن مفهوم ضرب تانسوری تقریباً حیاتی است، پس یوتوب را باز کنید.



شکل ۳: کازینوی بزرگِ مونته‌کارلو، شاهزاده‌نشینِ موناکو

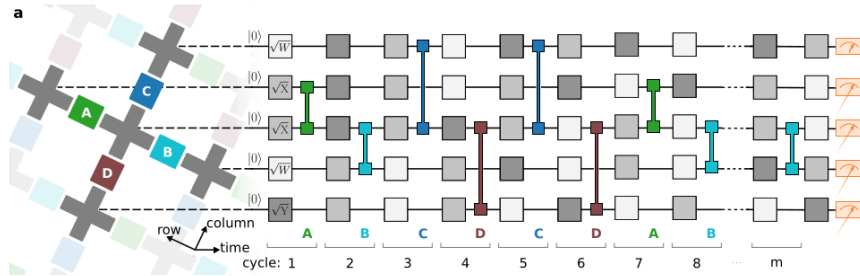
یکی از قسمت‌های جذاب و قابل فهم مسئله این است که چه کدی توی کامپیوترهای معمولی مان بزنیم که همین مسئله را (در ابعاد کوچک) شبیه‌سازی کند؟
 برای یک شبیه‌سازی کاملاً واقعی، پنج کیوبیت را در نظر بگیرید، که حالت آن‌ها تشکیل یک بردار مختلط در فضای 2^5 -بعدی (یا یک تانسور $2 \times 2 \times 2 \times 2 \times 2$) می‌دهد. فرض کنید حالت اولیه این پنج کیوبیت $|00000\rangle$ باشد، یعنی مقدار آن بردار برابر $(1, 0, \dots, 0)$ خواهد بود.
 در ساختار تانسوری، اگر بخواهیم یک گیت تک کیوبیتی را (که به ماتریس 2×2 است) روی کیوبیت سوم اعمال کنیم، باید یک ضرب ماتریسی بر روی بعد سوم تانسور حالتمان انجام دهیم. این کار واقعی پایتون همین کار را می‌کند.

```

1 import numpy
2
3 state_shape = (2, 2, 2, 2, 2)
4 state = np.reshape(np.array([1] + 31 * [0]), state_shape)
5
6 # a valid quantum state must have norm = 1
7 assert(np.linalg.norm(state) == 1.0)
8
9 gate = np.matrix([[0, 1j], [-1j, 0]])
10
11 # a valid quantum gate must be unitary (maintains norm)
12 assert(np.all(np.matmul(gate.H, gate) == np.identity(2)))
13
14 # apply gate on 3th qubit (2nd if we start from 0)
15 np.tensordot(gate, state, axes=(1, 2))

```

حالا به سراغ دستورالعمل گوگل برای مدار تصادفی می‌رویم.



شکل ۴: الگوی گوگل در ایجاد مدار تصادفی [۳]

این دستورالعمل، از m مرحله تشکیل شده که در هر مرحله این دو فرایند انجام می‌شود

- به ازای هر کدام از کیوبیت‌ها، به شکل تصادفی یکی از سه گیت \sqrt{X} , \sqrt{Y} , \sqrt{W} ^۸ را انتخاب می‌کنیم و اعمال می‌کنیم. اگر در مرحله پیش یکی از این سه گیت را روی این گیت اثر داده‌ایم، از انتخاب‌هایمان حذف می‌کنیم و از بین دو گیت باقی‌مانده یکی را برمی‌گزینیم.

- مطابق الگوی هر مرحله، بر روی زوج کیوبیت‌های مشخص شده توسط الگو، گیت دوکیوبیتی‌ای را که $fSim(\pi/2, \pi/6)$ نام دارد، اعمال می‌کنیم. این گیت نسبت به دوکیوبیت متقارن است.^۹

الگوی هر مرحله، یکی از حروف A تا H می‌تواند باشد که مشخص می‌کند که بر روی کدام زوج کیوبیت‌ها این گیت دوکیوبیتی اثر کند (درج شده در جدول ۱). دنباله الگوی مرحله‌ها در این مورد شبیه‌سازی ما عبارت تکرارشونده ABCDCDAB است. [۳]

الگو	کیوبیت‌ها
A	۱، ۲
B	۲، ۳
C	۰، ۲
D	۲، ۴

جدول ۱: داده‌های الگوهای اعمال گیت دوکیوبیتی استخراج شده از روی شکل ۶

مقدار ماتریس این گیت‌ها به این ترتیب است:

$$\sqrt{X} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$$

^۸ مهم نیست این رادیکال‌ها و نامگذاری‌ها چه معنی‌ای می‌دهند و چه دلیلی دارند، ما می‌دانیم هر گیت تک کیوبیتی، یک ماتریس 2×2 است و تنها مقدار این ماتریس‌ها را بدانیم.

^۹ یک گیت دوکیوبیتی یک تبدیل خطی بر روی فضای 2×2 بعدی است. به عبارت دیگر، یک ماتریس 4×4 است.

$$\sqrt{Y} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\sqrt{W} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -\sqrt{i} \\ \sqrt{-i} & 1 \end{bmatrix}$$

$$\text{fSim}(\pi/2, \pi/6) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & e^{-i\frac{\pi}{6}} \end{bmatrix}$$

در مرحله آخر، تنها گیت یک کیوبیتی اعمال می‌کنیم و سپس به سراغ اندازه‌گیری می‌رویم. برای اندازه‌گیری این بردار حالت که به شکل $2 \times 2 \times 2 \times 2 \times 2$ است، برای هر مؤلفه آن که v باشد داریم $|v|^2$ احتمال رخداد رشته بیت مربوط به آن مؤلفه است. حالا کافیست از این توزیع نمونه‌هایی را برداریم و پارامتر F را که در قسمت قبل معرفی کردیم برای آن‌ها محاسبه کنیم. و در نهایت، با کد زیر می‌خواهیم صد بار، با شش مرحله مدارهای تصادفی‌ای بسازیم و هر بار با ده بار نمونه‌برداری از خروجی، پارامتر F را تخمین بزنیم.

```

1 import numpy as np
2 from random import choice
3
4 n = 5
5 cycles = 6
6 state_shape = (2, 2, 2, 2, 2)
7
8 # define single-qubit gates
9 xsqrt = np.array([[1, -1j], [-1j, 1]]) / np.sqrt(2)
10 ysqrt = np.array([[1, -1], [1, 1]]) / np.sqrt(2)
11 wsqrt = np.array([[1, -np.sqrt(1j)], [np.sqrt(-1j), 1]]) /
    np.sqrt(2)
12
13 single_gates = [xsqrt, ysqrt, wsqrt]
14
15 # define double-qubit gates
16 double_gate = np.array([[1, 0, 0, 0], \
17                          [0, 0, -1j, 0], \
18                          [0, -1j, 0, 0], \
19                          [0, 0, 0, np.exp(1j * np.pi/6)]])
20 double_gate = np.reshape(double_gate, (2, 2, 2, 2))
21
22 # double-qubit gate pattern (from shape) ABCDCDAB
23 pattern = [(1,2), (2,3), (0,2), (2,4), (0,2), (2,4), (1,2),
    (2,3)]

```

```

24
25 # sample from different 100 random circuits
26 samples_U = 100
27
28 # list of F values
29 fs = []
30
31 for _ in range(samples_U):
32     last_applied_gate = [None] * n
33
34     # define input state
35     state = np.reshape(np.array([1] + 31 * [0]), state_shape)
36
37     # iterate over cycles
38     for c in range(cycles):
39         # iterate over qubits to apply single gates
40         for i in range(n):
41             # apply a random gate on ith qubit
42             gate = choice([g for g in single_gates if np.all(g !=
43                             last_applied_gate[i])])
44             state = np.tensordot(gate, state, axes=(1, i))
45
46         # apply double-qubit gate
47         state = np.tensordot(double_gate, state, axes=((2,3),
48                 pattern[c % len(pattern)]))
49
50     # last half cycle
51     for i in range(n):
52         gate = choice([g for g in single_gates if np.all(g !=
53                     last_applied_gate[i])])
54         state = np.tensordot(gate, state, axes=(1, i))
55
56     # let's dice!
57     ps = (abs(state)**2).flatten()
58     # number of samples from output of this circuit
59     samples_q = 10
60     for _ in range(samples_q):
61         fs.append(2**n * np.random.choice(ps, p=ps) - 1)
62
63 print('F = ', np.mean(fs), '±', np.std(fs) / np.sqrt(len(fs)
64     ) - 1))

```

نتیجه اجرای این کد، برای من این شد، انتظار داشتیم چند بشود؟

$$F = 0.98 \pm 0.05$$

بعد از این اتفاق‌ها آی‌بی‌ام ادعا کرده که برتری کوانتومی رخ نداده و این محاسبات را می‌توان در طی دو روز با یک سوپر کامپیوتر انجام داد. [۱۰]
 (امروز که این نوشته تمام شد سالگرد مرتضی کیوان بود. کاشکی به جای این همه حرفِ مفت، فقط می‌نشستیم و یادی از او می‌کردیم)

مراجع

- [1] Scott Aaronson. *Scott's Supreme Quantum Supremacy FAQ*. URL: <https://www.scottaaronson.com/blog/?p=4317> (visited on 10/19/2019).
- [2] Scott Aaronson and Lijie Chen. "Complexity-theoretic foundations of quantum supremacy experiments". In: *arXiv preprint arXiv:1612.05903* (2016).
- [3] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510.
- [4] Charles H Bennett et al. "Strengths and weaknesses of quantum computing". In: *SIAM journal on Computing* 26.5 (1997), pp. 1510–1523.
- [5] Ethan Bernstein and Umesh Vazirani. "Quantum complexity theory". In: *SIAM Journal on computing* 26.5 (1997), pp. 1411–1473.
- [6] Sergio Boixo et al. "Characterizing quantum supremacy in near-term devices". In: *Nature Physics* 14.6 (2018), p. 595.
- [7] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.
- [8] Oded Goldreich. "In a world of $P = BPP$ ". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Springer, 2011, pp. 191–232.
- [9] Abstruse Goose. *Schrödinger's Infinitesimal Miscalculation*. URL: <https://abstrusegoose.com/6> (visited on 10/06/2019).
- [10] Edwin Pednault et al. *Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits*. 2019. arXiv: 1910.09534 [quant-ph].

- [11] C. E. Porter and R. G. Thomas. “Fluctuations of Nuclear Reaction Widths”. In: *Phys. Rev.* 104 (2 1956), pp. 483–491. DOI: 10.1103/PhysRev.104.483. URL: <https://link.aps.org/doi/10.1103/PhysRev.104.483>.
- [12] Erwin Schrödinger. “Die gegenwärtige Situation in der Quantenmechanik”. In: *Naturwissenschaften* 23.49 (1935), pp. 823–828.