Google の「量子超越性」について

田浦健次朗

2019/11/26

Contents

Contents

なぜこれについて話す?

日本經濟新聞

朝刊・夕刊　ストーリー　Myニュ

トップ　速報　経済・金融　政治　ビジネス　マーケット　テクノロジー　国際　オピニオン　スポーツ　社会

量子計算、世界が競う　性能はスパコンの15億倍

イノベーションロードマップ　ネット・IT　北米　科学&新技術

2019/10/24 23:17 | 1574文字 ［有料会員限定］

保存　共有　印刷　その他

現在のスーパーコンピューターの15億倍もの性能をもつ次世代コンピューターの登場が現実味を帯びてきた。米グーグルが現在のスパコンでは困難な問題を簡単に解ける量子コンピューターの開発にメドをつけたためで、産業や金融から軍事までそのかたちを一変させる可能性を秘める。実用化にはまだ20〜30年かかりそうだが、人工知能（AI）と組み合わせて影響は世界に及ぶ。開発に向けた攻防は国家の覇権を左右する。

【関連記事】 創薬・素材開発、劇的に速く　量子コンピューターで

グーグルは24日に会見を開き、成果の内容を発表した。今回、最先端のスパコンで約1万年かかる乱数をつくる問題を、量子コンピューターはわずか3分20秒で解いた。計算

なぜこれについて話す?

創薬・素材開発、劇的に速く　量子コンピューターで

2019/10/24 23:00 | 2519文字 ［有料会員限定］

保存　共有　印刷　その他

画像の拡大

グーグルが開発した量子コンピューター=米グーグル提供

米グーグルが開発中の量子コンピューターが実用化されれば、半導体の処理能力に縛られていたコンピューターが新たな発展段階に入る。膨大な選択肢の中から瞬時に最適解を導き出せるため、経済や産業への影響は大きい。一方、現行の暗号化方式が簡単に破られるリスクがあり、政府や企業の間には不安もある。

日本経済

トップ　速報　経済・金融　政

量子計算、世界

イノベーションロードマップ

2019/10/24 23:17 | 1574文

現在のスーパーコンピ
現実味を帯びてきた。米
コンピューターの開発に
変させる可能性を秘める
組み合わせて影響は世界

【関連記事】 創薬・

グーグルは24日に会見を開き、成果の内容を発表した。今回、最先端のスパコンで約1万年かかる乱数をつくる問題を、量子コンピューターはわずか3分20秒で解いた。計算

なぜこれについて話す?

2019/11/12 07:00

グーグル「量子超越性」実証の衝撃

グーグル「量子超越性」実証の衝撃

性能16億倍、夢の計算機を徹底検証

中田 敦＝日経コンピュータ

日経コンピュータ

世界のエネルギー問題を解決

グーグルによれば100量子ビットがあれば、バクテリアが有機物を分解してアンモニアを合成するプロセスを分子レベルでモデル化し、シミュレーションできるという。もしバクテリアが常温でアンモニアを合成するプロセスを解明して再現できれば、人類に多大な恩恵をもたらす。なぜなら現在、空気を高温高圧状態にすることで窒素と水素からアンモニアを化学合成する「ハーバー・ボッシュ法」を実行するために、世界の燃料消費の1〜2%が投じられているからだ。この分を節約できる可能性が出てくる。

ーターで

その他

画像の拡大

グーグルが開発した量子コンピューター=米グーグル提供

米グーグルが開発中の量子コンピューターが実用化されれば、半導体の処理能力に縛られていたコンピューターが新たな発展段階に入る。膨大な選択肢の中から瞬時に最適解を導き出せるため、経済や産業への影響は大きい。一方、現行の暗号化方式が簡単に破られるリスクがあり、政府や企業の間には不安もある。

組み合わせて影響は世界

【関連記事】 創薬・

グーグルは24日に会見を開き、成果の内容を発表した。今回、最先端のスパコンで約1万年かかる乱数をつくる問題を、量子コンピューターはわずか3分20秒で解いた。計算

なぜこれについて話す?

2019/11/12 07:00

グーグル「量子超越性」実証の衝撃

グーグル「量子超越性」実証の衝撃

性能16億倍、夢の計算機を徹底検証

2019/10/25 11:00

ニュース解説

IBMがGoogleの“量子超越性実証”を全否定

野澤 哲生＝日経 xTECH／日経エレクトロニクス

日経 XTECH

PR 最新の五感センサー×AI、xR、BMIの詳細 12月3日開催【人間拡張セミナー】

米IBMは2019年10月21日、「“量子超越性”について」というタイトルのブログを公開し、米Googleによる「53量子ビットの量子コンピューターで量子超越性を実証した」とする論文発表を真っ向から批判した。「（Googleの論文は）量子超越性に必要な条件をいまだ満たしていない」（同ブログ）という。

不自然な設計のスパコンを想定

き出せるため、経済や産業への影響は大きい。一方、現行の暗号化方式が簡単に破られるリスクがあり、政府や企業の間には不安もある。

グーグルは24日に会見を開き、成果の内容を発表した。今回、最先端のスパコンで約1万年かかる乱数をつくる問題を、量子コンピューターはわずか3分20秒で解いた。計算

なぜこれについて話す?

1. 世の中では「誇大広告」「報道の誇張・誤解・単純化・庶民狙いの表現」があふれる (今回のがそうだということではなく)

なぜこれについて話す?

1. 世の中では「誇大広告」「報道の誇張・誤解・単純化・庶民狙いの表現」があふれる (今回のがそうだということではなく)
2. インパクトの大きそうな記事は常にそのことを意識して受け止める必要がある
    - 技術的な記事に限らない (スキャンダルの類を含め)

なぜこれについて話す?

1. 世の中では「誇大広告」「報道の誇張・誤解・単純化・庶民狙いの表現」があふれる (今回のがそうだということではなく)
2. インパクトの大きそうな記事は常にそのことを意識して受け止める必要がある
   - 技術的な記事に限らない (スキャンダルの類を含め)
3. 特に, 自分の専門に近い分野 (電子情報技術) については, 自分なりの真贋を持とうという姿勢が重要
   3.1 中身を詳しく理解する, 裏を取る
   3.2 庶民向け記事の先 (より技術的な記事や原典) を見る

なぜこれについて話す?

1. 世の中では「誇大広告」「報道の誇張・誤解・単純化・庶民狙いの表現」があふれる (今回のがそうだということではなく)
2. インパクトの大きそうな記事は常にそのことを意識して受け止める必要がある
   - 技術的な記事に限らない (スキャンダルの類を含め)
3. 特に, 自分の専門に近い分野 (電子情報技術) については, 自分なりの真贋を持とうという姿勢が重要
   3.1 中身を詳しく理解する, 裏を取る
   3.2 庶民向け記事の先 (より技術的な記事や原典) を見る
4. そのもとで自分なりの心象形成 (誇張か? 将来技術に対する展望 (悲観・楽観) の差か?) をする
   - もちろんこれを喋っている人の言うことも鵜呑みにせず

なぜこれについて話す?

1. 世の中では「誇大広告」「報道の誇張・誤解・単純化・庶民狙いの表現」があふれる (今回のがそうだということではなく)
2. インパクトの大きそうな記事は常にそのことを意識して受け止める必要がある
   - 技術的な記事に限らない (スキャンダルの類を含め)
3. 特に, 自分の専門に近い分野 (電子情報技術) については, 自分なりの真贋を持とうという姿勢が重要
   3.1 中身を詳しく理解する, 裏を取る
   3.2 庶民向け記事の先 (より技術的な記事や原典) を見る
4. そのもとで自分なりの心象形成 (誇張か? 将来技術に対する展望 (悲観・楽観) の差か?) をする
   - もちろんこれを喋っている人の言うことも鵜呑みにせず
5. この先自分が何をしたいかを判断する糧にもなる

Contents

要するに何を達成したか?

- 53 の「量子ビット (qubit, qbit, Q ビット; 以下 Q ビット)」を持つ量子計算機 (正確には量子ゲート型の計算機. 後述) を作った
- 発表は Google

  "Quantum supremacy using a programmable superconducting processor" *https://doi.org/10.1038/s41586-019-1666-5*

量子計算機の開発は前から行われている．今回は何が違う?

- 「量子超越性 (Quantum Supremacy)」を達成したと発表した

  . . . This dramatic increase in speed compared to all known classical algorithms is an experimental realization of quantum supremacy *for this specific computational task, heralding a much-anticipated computing paradigm.*

  (強調は田浦)
- 論文発表前に一時的に原稿がオンラインにあがったりしたことも話題を呼んだ一因かも知れない
- 53 Q ビットがこれまで達成された Q ビット数と桁違いというわけではない

量子超越性の達成って?

- 「量子計算機で普通の (デジタル) 計算機よりも高速に計算できた」ことを意味するのだが,「何が」計算できたのかを知っておくことが重要

量子超越性の達成って?

- 「量子計算機で普通の (デジタル) 計算機よりも高速に計算できた」ことを意味するのだが, 「何が」計算できたのかを知っておくことが重要
- デジタル計算機でできる計算の { 多く/ほとんど/ましてやすべて } が量子計算機で, より高速に出来るようになった, ということではない

量子超越性の達成って?

- 「量子計算機で普通の (デジタル) 計算機よりも高速に計算できた」ことを意味するのだが, 「何が」計算できたのかを知っておくことが重要
- デジタル計算機でできる計算の { 多く/ほとんど/ましてやすべて } が量子計算機で, より高速に出来るようになった, ということではない
- デジタル計算機では手に終えないが量子計算機なら解ける問題 (ベンチマーク問題) を 1 つ定義し, 実際にそれを実証した

量子超越性の達成って?

- 「量子計算機で普通の (デジタル) 計算機よりも高速に計算できた」ことを意味するのだが, 「何が」計算できたのかを知っておくことが重要
- デジタル計算機でできる計算の { 多く/ほとんど/ましてやすべて } が量子計算機で, より高速に出来るようになった, ということではない
- デジタル計算機では手に終えないが量子計算機なら解ける問題 (ベンチマーク問題) を 1 つ定義し, 実際にそれを実証した
- 前述の目的達成のため, それをより正確に理解しようというのが今日の話題

ベンチマーク問題: 量子回路からのサンプリング

- *"Characterizing quantum supremacy in near-term devices"*
  https://doi.org/10.1038/s41567-018-0124-x

  Given a random quantum circuit U of depth d, *we are interested in* sampling from the distribution $p_U(x) \equiv |\langle x|\phi_d\rangle|^2$ *of bit-strings in the computational basis* $\{|x\rangle|\}$.

(強調は田浦)

- 以下はこれが何のことか, なぜ古典的な計算機だと難しいのか, を平易に説明する

Contents

Contents

ビットとデジタル回路

- ビット: 0 または 1

  0 または 1

- (n 個の) ビット列 ≈ ビットが n 個並んだもの

  0 1 1 0 1 0 • • • 0 1 など

- デジタル回路 ≈ ビット列の値を変化させるもの

Contents

ひとつの Q ビット

- Q ビット ≈ 0 と 1 の重ね合わせ

$$q = \alpha_0 \boxed{0} + \alpha_1 \boxed{1}$$

- q : (Q ビットの状態) 0 および 1 の「混合割合, 確率」

ひとつの Q ビット

- Q ビット ≈ 0 と 1 の重ね合わせ

$$q \quad = \quad \alpha_0 \boxed{0} + \alpha_1 \boxed{1}$$

- q : (Q ビットの状態) 0 および 1 の「混合割合, 確率」みたいなもの

ひとつのQビット

- Qビット ≈ 0と1の重ね合わせ

$$q = \alpha_0 \boxed{0} + \alpha_1 \boxed{1}$$

- q : (Qビットの状態) 0および1の「混合割合, 確率」みたいなもの
- 正確には, 複素数2つ組であって, 大きさが1:

$$q = (\alpha_0, \alpha_1), \text{ただし } |\alpha_0|^2 + |\alpha_1|^2 = 1$$

  - 注: $|\alpha|^2 \equiv \alpha\bar{\alpha}$

ひとつのQビット

- Qビット ≈ 0と1の重ね合わせ

$$q = \alpha_0 \boxed{0} + \alpha_1 \boxed{1}$$

- q : (Qビットの状態) 0および1の「混合割合, 確率」みたいなもの
- 正確には, 複素数2つ組であって, 大きさが1:

$$q = (\alpha_0, \alpha_1), \text{ただし } |\alpha_0|^2 + |\alpha_1|^2 = 1$$

  - 注: $|\alpha|^2 \equiv \alpha\bar{\alpha}$
- 観測すると, 0である確率が $|\alpha_0|^2$, 1である確率が $|\alpha_1|^2$

ひとつのQビット

- Qビット ≈ 0と1の重ね合わせ

$$q = \alpha_0 \boxed{0} + \alpha_1 \boxed{1}$$

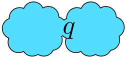
- q : (Qビットの状態) 0および1の「混合割合, 確率」みたいなもの
- 正確には, 複素数2つ組であって, 大きさが1:

$$q = (\alpha_0, \alpha_1),\text{ ただし } |\alpha_0|^2 + |\alpha_1|^2 = 1$$

  - 注: $|\alpha|^2 \equiv \alpha\bar{\alpha}$
- 観測すると, 0である確率が $|\alpha_0|^2$, 1である確率が $|\alpha_1|^2$
- ここで悩まない. これは「公理」. そういうものと思って先へ

複数の Q ビット

- 2 個の Q ビット: $00, 01, 10, 11$ の $4\ (= 2^2)$ つの状態の重ね合わせ



$$= \quad \alpha_{00}\,\boxed{0}\boxed{0} + \alpha_{01}\,\boxed{0}\boxed{1} + \alpha_{10}\,\boxed{1}\boxed{0} + \alpha_{11}\,\boxed{1}\boxed{1}$$

- つまり複素数 4 つ組であって, 大きさが 1:

$$q = (\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11})$$

ただし

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$$

- 00 が観測される確率が $|\alpha_{00}|^2$, etc.

n 個の Q ビット

- 一般に n 個の Q ビット: あらゆる n ビットの列 — $00\cdots00, \cdots, 11\cdots11$ の 2^n 個 —の状態の重ね合わせ

$$
\begin{aligned}
&= \quad \alpha_{00\cdots00}\ \boxed{0}\boxed{0}\cdots\boxed{0}\boxed{0} \\
&+ \quad \alpha_{00\cdots01}\ \boxed{0}\boxed{0}\cdots\boxed{0}\boxed{1} \\
&+ \quad \alpha_{00\cdots10}\ \boxed{0}\boxed{0}\cdots\boxed{1}\boxed{0} \\
&+ \quad \alpha_{00\cdots11}\ \boxed{0}\boxed{0}\cdots\boxed{1}\boxed{1} \\
& \quad\ \cdots \\
&+ \quad \alpha_{11\cdots11}\ \boxed{1}\boxed{1}\cdots\boxed{1}\boxed{1}
\end{aligned}
$$

- q は複素数 2^n 個の組であって, 大きさが 1:

$$|\alpha_{00\cdots00}|^2 + \cdots + |\alpha_{11\cdots11}|^2 = 1$$

- 観測すると $i \in \{0,1\}^n$ が観測される確率が $|\alpha_i|^2$

プログラマな人向けの説明

- n ビット (bool が n 個):

```
template<int n>
struct bits { bool a[n] };
```

- n 量子ビット (複素数が 2^n 個)

```
template<int n>
struct qbits { double complex a[1 << n] }; # 2^n
```

Q ビットの驚くべきところ

- 「n 個の Q ビット」は Q ビットがバラバラ (独立) に n 個存在しているのとは違う

Q ビットの驚くべきところ

- 「n 個の Q ビット」は Q ビットがバラバラ (独立) に n 個存在しているのとは違う
- 「独立に n 個」であれば, 1 つの Q ビットの状態 (複素数 2 個) $\times n$, つまり複素数 $2n$ 個分の情報しか持っていない. つまり,

```
template<int n>
struct qbits* { double complex a[2 * n] }; # 2n
```

となっていたところである

Q ビットの驚くべきところ

- 「n 個の Q ビット」は Q ビットがバラバラ (独立) に n 個存在しているのとは違う
- 「独立に n 個」であれば, 1 つの Q ビットの状態 (複素数 2 個) $\times n$, つまり複素数 $2n$ 個分の情報しか持っていない. つまり,

```
template<int n>
struct qbits* { double complex a[2 * n] }; # 2n
```

  となっていたところである
- n ビットのあらゆる組み合わせに対する値を (長さ 1 という制限以外は) 自由に持てるところが驚くべきところ

物理的な注 (1): 量子力学で習ったこととの関係

- 量子力学で習った波動関数 $\phi(x)$ は 1 つの量子 (例えば電子) の状態を記述するものだった
- $|\phi(x)|^2 =$ 量子が場所 x に見出される確率 (密度)
- $|\phi(x)|^2$ の全空間に渡る和 (積分) $= 1$

$$\int_{\text{全空間}} |\phi(x)|^2 dx = 1$$

- $\phi(x)$ が時間と共にどう発展していくかを記述するのが, Schrödinger 方程式

物理的な注 (2): n 量子の波動関数

- 量子が n 個,「もつれ合い状態」にある場合の波動関数は, $\phi(x_0, x_1, \ldots, x_{n-1})$
- $|\phi(x_0, x_1, \ldots, x_{n-1})|^2$ は, 観測時に,
  - 0 個目の量子を場所 x_0 に,
  - 1 個目の量子を場所 x_1 に,
  - ...
  - $(n-1)$ 個目の量子を場所 x_{n-1} に

  見出す確率密度

物理的な注 (3): Q ビットと波動関数の関係

- Q ビットは量子 (電子, 光子) で実現され, 従って波動関数を持つ
- ただしここでの波動関数は, 位置ではなく, 二値 (0/1) の関数
- つまり二状態系, 例えば電子のスピンのようなもの (これ以上の物理は田浦には無理)
- 要するに 1 Q ビットの波動関数は $\phi(s)$ $(s \in \{0, 1\})$.
- $|\phi(s)|^2$ = 量子が状態 s (0 または 1) に見出される確率
  - これをこれまで α_s と書いていた
- 条件

$$\int_{\text{全空間}} |\phi(x)|^2 dx = 1$$

に相当するのが,

$$|\phi(0)|^2 + |\phi(1)|^2 = 1$$

つまり

$$|\alpha_0|^2 + |\alpha_1|^2 = 1$$

物理的な注 (4): n 個の Q ビットの場合

- 量子が n 個ある場合の波動関数は, $\phi(s_0, s_1, \ldots, s_{n-1})$ で, これをこれまで, $\alpha_{s_0 s_1 \cdots s_{n-1}}$ と書いていた
- n 個の 2 値 (0/1) の関数であることから, 2^n 個の情報を持っていることになる
- $|\phi(s_0, s_1, \ldots, s_{n-1})|^2$ は, 観測時に
  - 0 個目の量子を状態 s_0 に,
  - 1 個目の量子を状態 s_1 に,
  - $\ldots$
  - $(n-1)$ 個目の量子を状態 s_{n-1}

  に見出す確率

量子回路

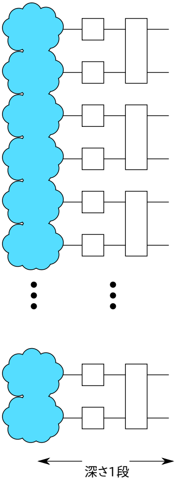
- 量子回路 $\approx$ Q ビット列の値を, ($|q| = 1$ を保ちつつ) 変化させるもの

量子回路の深さ (depth)

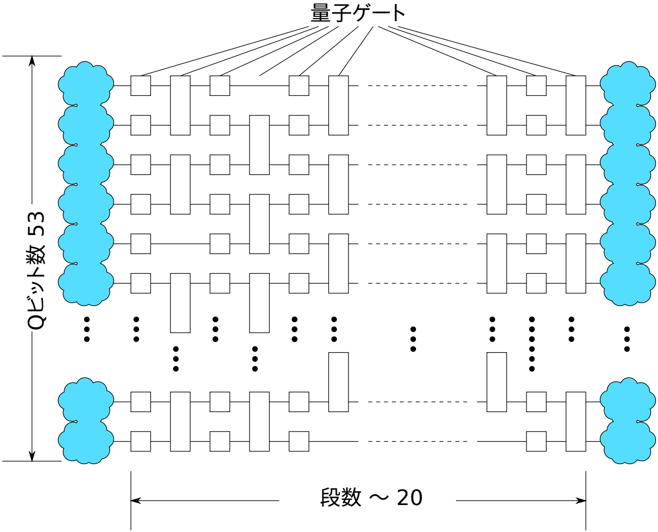
Given a random quantum circuit U of depth d, ...

- デジタル回路が原子的な素子 (ゲート) — NOT, AND, OR など — を組み合わせてできていたのと同様, 量子回路も原子的な「量子ゲート」を組み合わせてできる
- ひとつの量子ゲートは少数 (1 または 2) の Q ビットに「作用」する (正確な意味は後述)
- Google の実験では,
  - 全 Q ビットに 1 Q ビットゲートを適用
  - 全 Q ビットに 2 Q ビットゲートを適用

  を「1 cycle」と定義
- 深さ d の量子回路 $\equiv$ d サイクルの量子回路



量子回路の深さ (depth)

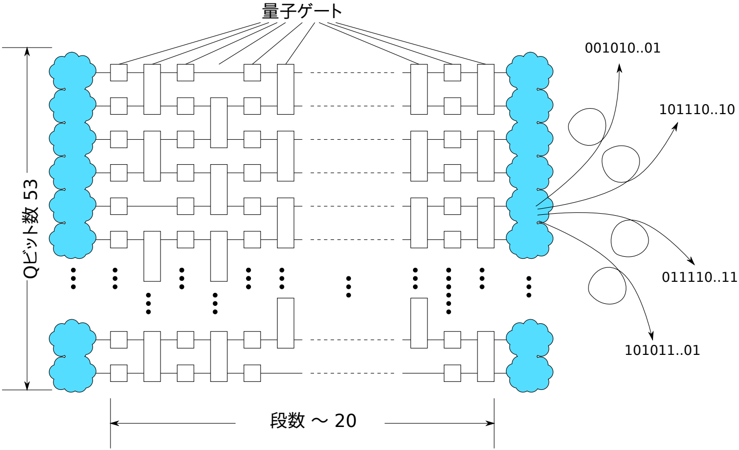


Contents

量子回路を経た状態からのサンプリング

Given a random quantum circuit U of depth d, we are interested in sampling from the distribution *. . .*

- 深さ d の量子回路を通した後の状態 (波動関数) から, ビット列をサンプリングする



量子計算機にできること

- ところで, 量子計算機にできることはまさに,
  - Q ビット群に量子ゲートを作用させる, すなわち, 波動関数 ($\approx$ 状態の確率分布) を変化させる
  - Q ビットを観測して, n ビット列を取り出す; このとき $|\alpha_i|^2$ の確率でビット列 i が取り出される

  ことである
- 従ってこの問題設定は, 「量子計算機を作ったのでできるものなら, シミュレートしてください」というのとほとんど同じ

問題設定に関する注

- この問題設定「深さ d の量子回路を通した後の状態 (波動関数) から, ビット列をサンプリングする」は, 「各ビットパターンが現れる確率を求めよ」とは違うのか?

問題設定に関する注

- この問題設定「深さ d の量子回路を通した後の状態 (波動関数) から, ビット列をサンプリングする」は, 「各ビットパターンが現れる確率を求めよ」とは違うのか?
- 量子計算機には, 前者はたやすいが後者は無理
  - 量子計算機は, α_* の値そのものを取り出せるわけではない
  - 観測 = サンプリング (観測した時点で n ビット列が観測される)

問題設定に関する注

- この問題設定「深さ d の量子回路を通した後の状態 (波動関数) から, ビット列をサンプリングする」は, 「各ビットパターンが現れる確率を求めよ」とは違うのか?
- 量子計算機には, 前者はたやすいが後者は無理
  - 量子計算機は, α_* の値そのものを取り出せるわけではない
  - 観測 = サンプリング (観測した時点で n ビット列が観測される)
- 古典的計算機には前者を出来る可能性があるが, 2^{53} 通りのビット列すべてに対し, 確率を求めると, それだけで 2^{53} 個の数が必要になる

問題設定に関する注

- この問題設定「深さ d の量子回路を通した後の状態 (波動関数) から, ビット列をサンプリングする」は, 「各ビットパターンが現れる確率を求めよ」とは違うのか?
- 量子計算機には, 前者はたやすいが後者は無理
  - 量子計算機は, α_* の値そのものを取り出せるわけではない
  - 観測 = サンプリング (観測した時点で n ビット列が観測される)
- 古典的計算機には前者を出来る可能性があるが, 2^{53} 通りのビット列すべてに対し, 確率を求めると, それだけで 2^{53} 個の数が必要になる
- 1 つの確率を表すのに 8 バイト必要とすると, 確率分布そのものを記述するには以下が必要

$$8\,\text{バイト} \times 2^{53} = 2^{56} \approx 6.4 \times 10^{16} \approx 64\text{PB}$$

問題設定に関する注

- この問題設定「深さ d の量子回路を通した後の状態 (波動関数) から, ビット列をサンプリングする」は, 「各ビットパターンが現れる確率を求めよ」とは違うのか?
- 量子計算機には, 前者はたやすいが後者は無理
  - 量子計算機は, α_* の値そのものを取り出せるわけではない
  - 観測 = サンプリング (観測した時点で n ビット列が観測される)
- 古典的計算機には前者を出来る可能性があるが, 2^{53} 通りのビット列すべてに対し, 確率を求めると, それだけで 2^{53} 個の数が必要になる
- 1 つの確率を表すのに 8 バイト必要とすると, 確率分布そのものを記述するには以下が必要

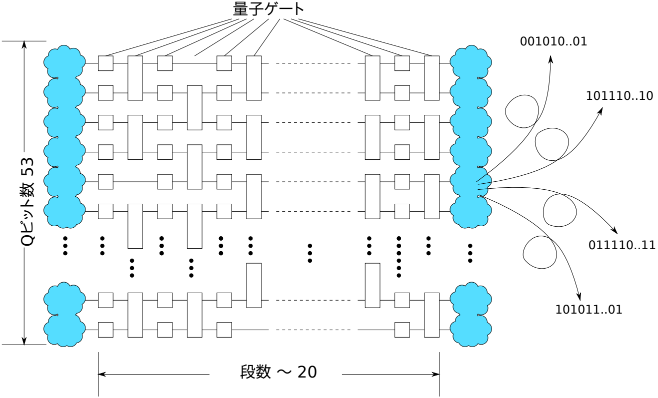
$$8\text{ バイト} \times 2^{53} = 2^{56} \approx 6.4 \times 10^{16} \approx 64\text{PB}$$

- したがって古典計算機でも, 「確率分布そのものを求める」のは無理で「確率分布から (多数回, 例えば 100 万回) サンプリングせよ (そのサンプルの分布が本物の分布に十分近ければ正解とする)」のが妥当な問題設定

Contents

量子超越性検証に使われた回路

- Q ビット数 ≤ 53
- 段数 ≤ 20



デジタル計算機で計算する方法

- n Q ビット $\equiv 2^n$ 個の複素数の組であったことを思い出そう
  - 以下，「状態ベクトル」
- もっとも単純な方法 (Schrödinger アルゴリズム)：状態ベクトルが量子ゲート (従って量子回路) によってどう変化していくかを，明示的に計算する

```
template<int n>
struct qbits { double complex a[1 << n] }; # 2^n
```

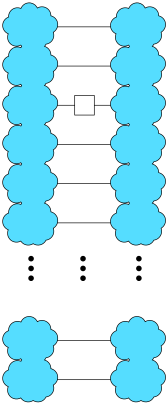
量子ゲート

- デジタル回路の NOT, OR, AND, NAND, NOR, etc. に相当するもの
- あらゆる量子回路は $|$ 状態ベクトル $| = 1$ を保つ変換 (ユニタリ変換) でなくてはならない
- 以下でいくつかの量子ゲートの例示をする (どうすれば古典的計算機で計算できるかの雰囲気を知るため; 細部は重要ではない)

1 Q ビットゲートの例: NOT ゲート

- 例えば第 0 Q ビットに NOT ゲートを作用させる (α が作用前, β が作用後の状態ベクトル)

$$\left\{ \begin{array}{lcl} \beta_{0**} & = & \alpha_{1**}, \\ \beta_{1**} & = & \alpha_{0**} \end{array} \right. \text{(}** \text{ は任意の } (n-1) \text{ ビット列)}$$



一般の１Ｑビットゲート

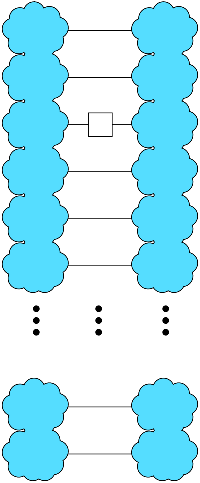
- 他にもいくつかの名前のついた１Ｑビットゲートがあるが, 一般には, k ビット目に作用するゲートによる効果は

$$\begin{cases} \beta_{**0\star\star} = u_{00}\ \alpha_{**0\star\star} + u_{01}\ \alpha_{**1\star\star}, \\ \beta_{**1\star\star} = u_{10}\ \alpha_{**0\star\star} + u_{11}\ \alpha_{**1\star\star} \end{cases}$$

の形. ここで,

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$$

が 2×2 のユニタリ行列 $(U^*U = I)$



一般の１Ｑビットゲートの計算法

- 上記

$$\begin{cases} \beta_{**0\star\star} & = & u_{00}\ \alpha_{**0\star\star} + u_{01}\ \alpha_{**1\star\star}, \\ \beta_{**1\star\star} & = & u_{10}\ \alpha_{**0\star\star} + u_{11}\ \alpha_{**1\star\star} \end{cases}$$

  のプログラムへの翻訳

```
void apply_single_qbit_gate(u[2][2], k, alpha[2^n], beta[2^n]) {
  /* は整数 k; 残りは複素数 */
  for (i = 0; i < 2^n; i++) {
    int b = kth bit of i;
    beta[i] = u[b][0] * alpha[clr_kth_bit(i,k)]
            + u[b][1] * alpha[set_kth_bit(i,k)];
  }
}
```

- 注:
  - set_kth_bit(i,k): i の第 k bit を 1 にしたもの
  - clr_kth_bit(i,k): i の第 k bit を 0 にしたもの

表記法上の慣習

- 上記
$$\left\{\begin{array}{lcl}\beta_{**0**} & = & u_{00}\,\alpha_{**0**} + u_{01}\,\alpha_{**1**},\\ \beta_{**1**} & = & u_{10}\,\alpha_{**0**} + u_{11}\,\alpha_{**1**}\end{array}\right.$$
を通常単に,
$$\left\{\begin{array}{lcl}\beta_0 & = & u_{00}\,\alpha_0 + u_{01}\,\alpha_1,\\ \beta_1 & = & u_{10}\,\alpha_0 + u_{11}\,\alpha_1\end{array}\right.$$
や
$$\begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix} = \begin{pmatrix}u_{00} & u_{01}\\ u_{10} & u_{11}\end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}$$
と書く.

2 Q ビットゲートの例: 制御 NOT ゲート

- 2 つの Q ビットに作用. 「制御ビット」と「標的ビット」
- 制御ビットが 0 ならば標的ビットに NOT ゲートを適用
- たとえば第 0 ビット (制御), 第 1 ビットに作用する場合

$$\begin{cases} \beta_{00**} & = & \alpha_{00**} \\ \beta_{01**} & = & \alpha_{01**} \\ \beta_{10**} & = & \alpha_{11**} \\ \beta_{11**} & = & \alpha_{10**} \end{cases}$$

より一般的な 2 Q ビットゲート: 制御 U ゲート

- 制御ビットが 1 ならば標的ビットに 1 Q ビットゲート U を適用
- 第 i Q ビット (制御), 第 j Q ビット (標的) (図示の都合上 $i < j$ とする) の場合

$$\begin{cases} \beta_{**0**0\star\star} & = & \alpha_{**0**0\star\star} \\ \beta_{**0**1\star\star} & = & \alpha_{**0**1\star\star} \\ \beta_{**1**0\star\star} & = & u_{00}\ \alpha_{**1**0\star\star} + u_{01}\ \alpha_{**1**1\star\star} \\ \beta_{**1**1\star\star} & = & u_{10}\ \alpha_{**1**0\star\star} + u_{11}\ \alpha_{**1**1\star\star} \end{cases}$$

- 表記上の慣習

$$\begin{pmatrix} \beta_{00} \\ \beta_{01} \\ \beta_{10} \\ \beta_{11} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & u_{00} & u_{01} \\ & & u_{10} & u_{11} \end{pmatrix} \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

要するに...

- 細かい数式は飛ばして, 要するに Schrödinger アルゴリズムでは, 量子ゲート一つ分の計算は, 状態ベクトルの一要素につき, 数回の演算でできるということ

```
for (i = 0; i < 2^n; i++) {
  beta[i] = 数回の加減乗算;
}
```

普通の計算機での必要メモリと時間

- 53 Q ビットを仮定
- メモリ: 1 要素につき 1 複素数 = 2 浮動小数点数

$$\begin{aligned} 16 \times 2^{53} &= 2^{57} \\ &= 1.28 \times 10^{17} \\ &= 128 \text{ PB} \end{aligned}$$

- メモリアクセス量: 1 量子ゲートの適用ですべての要素を一度は触ると仮定

$$\begin{aligned} 16 \times 2^{53} &= 2^{57} \\ &= 128 \text{ PB} \end{aligned}$$

- 演算量 (浮動小数点演算数): 1 要素計算につき 2 の複素数乗算 = 8 の浮動小数点演算が必要と仮定

$$\begin{aligned} 8 \times 2^{53} &= 2^{56} \\ &= 6.4 \times 10^{16} \\ &= 64 \text{ Pflops} \end{aligned}$$

スーパーコンピュータでのコスト見積もり

- 注: 以下は独自の大雑把な概算
- Oakridge Summit `https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/`

CPU	IBM Power9 $\times$ 2 / node
GPU	NVIDIA Volta V100 $\times$ 6 /node
nodes	4608
CPU メモリ	256GB / CPU
GPU メモリ	16GB / GPU
GPU メモリ性能	900GB/sec / GPU
GPU 演算性能	7.8 TFLOPS / GPU

- 主な障害: 搭載メモリ $\ll$ 必要メモリ
  - 搭載 $= (256\text{GB} \times 2 + 16\text{GB} \times 6) \times 4608 = 2.8\text{PB}$
  - 必要 $= 130$ PB

時間の見積もり

メモリ不足を度外視して時間を計算

- メモリアクセス/メモリ性能

$$\begin{aligned} &= 128\text{PB}/(900\text{GB} \times 6) \\ &= 23700 \text{ 秒} \end{aligned}$$

- 演算量/演算性能

$$\begin{aligned} &= 64\text{Pflops}/(7.8\text{TFLOPS} \times 6) \\ &= 1367 \text{ 秒} \end{aligned}$$

- 注: 上記は 1 量子ゲート分の時間だが, 詳細は省略して, 楽観的には深さ 1 の回路に対する時間だと思って良い
- ∴ 深さ 20 の回路に対する計算時間

$$23700 \times 20 = 474000 \text{ 秒} \approx 5 \text{ 日}$$

メモリが足りない事に対するアプローチ

- Google : Schrödinger アルゴリズムではない方法 (Schrödinger-Feynman 法) を使った
  - メモリは少なくて済むが計算時間は深さの指数関数に比例して増える
  - → 約 10000 年かかると主張
- IBM : 2 次記憶 (ディスク) を使えばよいと主張
  Pednault et al. "Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits"

Contents

まとめ

- この度示された「量子超越性」 $\approx$ 合成された量子回路のシミュレーション
- ともかく,「古典的なデジタル計算機では膨大な (2^{53}) 計算やメモリを必要とする計算がひとつ, 量子計算機で達成できた」ということ
- これ自身は,「人工的な問題」であって, 解けたから役に立つ問題ということではない

興味がわいた人へ

- 「創薬・素材開発」なんのこと? — 現在出来るということでは決してないが, どういうつながり・可能性があるのかを勉強する
- 「量子機械学習」なんのこと? — 上記同様
- 53 Q ビットで, 「少しは役に立つ」計算の例 — 53 の電子のスピンを計算したりするには役に立つのだろう
- 様々な古典的計算方法, またはその限界— 「量子回路のサンプリング」を古典的計算機で上手くやる方法の詳細 (参考文献参照)
- Q ビット数では遥かに上を行く, 「量子アニーリング型 ($\neq$ 量子ゲート型)」計算機

参考文献 I

- Boixo et al. Characterizing quantum supremacy in near-term devices. Nature Phys 14, 595600 (2018) doi:10.1038/s41567-018-0124-x
- Arute et al. “Quantum supremacy using a programmable superconducting processor.” Nature 574, 505510 (2019) doi:10.1038/s41586-019-1666-5
- Pednault et al. “Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits” (2019) https://arxiv.org/abs/1910.09534
- Markov et al. “Quantum Supremacy Is Both Closer and Farther than It Appears” (Schrödinger-Feynman 法) (2018) https://arxiv.org/abs/1807.10749
- Chen et al. “64-qubit quantum circuit simulation.” (Schrödinger-Feynman 法) (2018) https://arxiv.org/abs/1802.06952

関連記事 I

- Financial Times 2019/9/21 “Google claims to have reached quantum supremacy” `https://www.ft.com/content/b9bb4e54-dbc1-11e9-8f9b-77216ebe1f17`
- つくばサイエンスニュース. 2019/10/15. Google の「量子超越性」実証とは何なのか?
- IBM 2019/10/21 “On Quantum Supremacy” `https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/`
- Reuters 2019/10/23 Google claims 'quantum supremacy'; others say hold on a qubit (米グーグル、「量子超越性」達成と発表　スパコン超える) `https://www.reuters.com/article/alphabet-quantum/google-claims-quantum-supremacy-others-say-hold-on-a-qubit-`
- Reuters 2019/10/23 Google unveils quantum computer breakthrough; critics say wait a qubit `https://www.reuters.com/article/us-alphabet-quantum/google-unveils-quantum-computer-breakthrough-critics-say-wai`

関連記事 II

- 日経新聞 2019/10/24 創薬・素材開発、劇的に速く　量子コンピューターで https://www.nikkei.com/article/DGXMZO51374320U9A021C1EA2000/?n_cid=DSREA001
- 日経新聞 2019/10/24 量子計算、世界が競う　性能はスパコンの 15 億倍 https://www.nikkei.com/article/DGXMZO51375810U9A021C1MM8000/?n_cid=DSREA001
- 日経 xTECH 2019/10/25 IBM が Google の "量子超越性実証" を全否定 https://tech.nikkeibp.co.jp/atcl/nxt/column/18/00001/03066/
- 日経新聞 2019/10/28 Google 量子超越が示した意義　「飛行機は飛べる」https://www.nikkei.com/article/DGXMZO51384140V21C19A0000000/
- 日経コンピュータ 2019/11/14 グーグル「量子超越性」実証の衝撃 https://tech.nikkeibp.co.jp/atcl/nxt/mag/nc/18/110600140/