

2022-12-06

PCCC22(第22回PCクラスタシンポジウム)「HPCシステム技術の最前線」



# 量子古典融合コンピューティングに向けた 量子アルゴリズム開発

藤堂眞治 <[wistaria@phys.s.u-tokyo.ac.jp](mailto:wistaria@phys.s.u-tokyo.ac.jp)>

東京大学大学院理学系研究科



# How Fast Do Algorithms Improve?

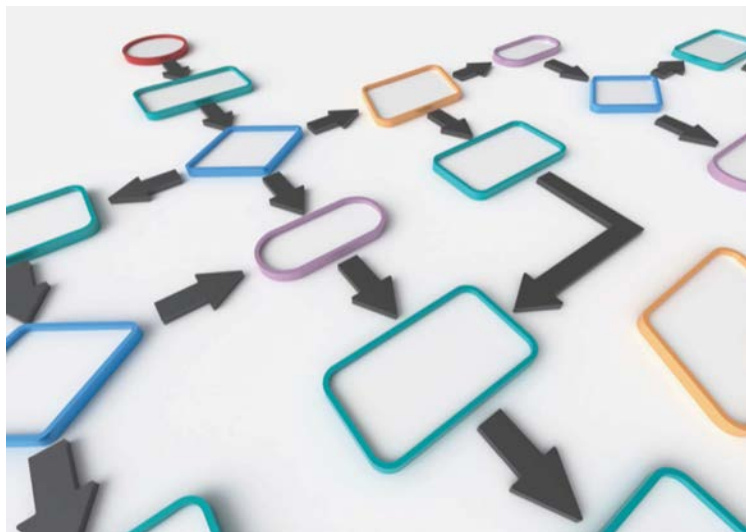
By **YASH SHERRY**

MIT Computer Science & Artificial Intelligence Laboratory, Cambridge, MA 02139 USA

**NEIL C. THOMPSON** 

MIT Computer Science & Artificial Intelligence Laboratory, Cambridge, MA 02139 USA

MIT Initiative on the Digital Economy, Cambridge, MA 02142 USA

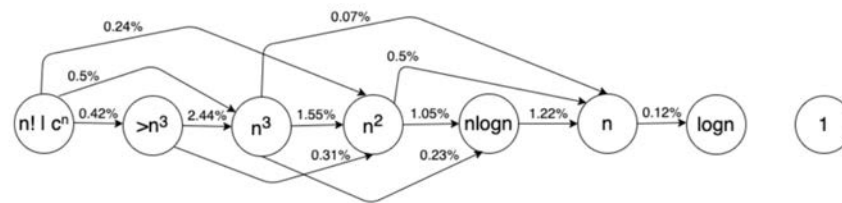


**A**lgorithms determine which calculations computers use to solve problems and are one of the central pillars of computer science. As algorithms improve, they enable scientists to tackle larger problems and explore new domains and new scientific techniques [1], [2]. Bold claims have been made about the pace of algorithmic progress. For example, the President's Council of Advisors on Science and Technology (PCAST), a body of senior scientists that advise the U.S. President,

Is progress faster in most algorithms? Just some? How much on average?

A variety of research has quantified progress for particular algorithms, including for maximum flow [5], Boolean satisfiability and factoring [6], and (many times) for linear solvers [4], [6], [7]. Others in academia [6], [8]–[10] and the private sector [11], [12] have looked at progress on benchmarks, such as computer chess ratings or weather prediction, that is not strictly comparable to algorithms since they lack either mathematically defined problem statements or verifiably optimal answers. Thus, despite substantial interest in the question, existing research a limited, fragmented algorithm progress.

In this article, we first comprehensively analyze algorithm progress



- analyzing data from 57 textbooks and 1137 research papers
- almost half of all algorithm families little or no improvement
- 14% of algorithmic families transformative improvements
- 30%-43% of algorithmic families improvements comparable or greater than Moore's Law
- “Algorithms are one of the most important sources of improvement in computing...”

# Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning

Pablo Villalobos\*, Jaime Sevilla\*<sup>†</sup>, Lennart Heim\*<sup>§</sup>, Tamay Besiroglu\*<sup>‡</sup>, Marius Hobbhahn \*<sup>¶</sup>, Anson Ho\*

**Abstract**—We analyze the growth of dataset sizes used in machine learning for natural language processing and computer vision, and extrapolate these using two methods; using the historical growth rate and estimating the compute-optimal dataset size for future predicted compute budgets. We investigate the growth in data usage by estimating the total stock of unlabeled data available on the internet over the coming decades. Our analysis indicates that the stock of high-quality language data will be exhausted soon; likely before 2026. By contrast, the stock of low-quality language data and image data will be exhausted only much later; between 2030 and 2050 (for low-quality language) and between 2030 and 2060 (for images). Our work suggests that the current trend of ever-growing ML models that rely on enormous datasets might slow down if data efficiency is not drastically improved or new sources of data become available.

## KEY TAKEAWAYS

- We project the growth of training datasets for vision and language models using both the historical growth rate and the compute-optimal dataset size given current scaling laws and existing compute availability estimates (Section III-A).
- We also project the growth in the total stock of unlabeled data, including high-quality language data (Section III-B).
- Language datasets have grown exponentially by more

seems likely to be around 18% to 31% per year. The current largest dataset is 3e9 images (Section IV-A).

- The stock of vision data currently grows by 8% yearly, but will eventually slow down to 1% by 2100. It is currently between 8.11e12 and 2.3e13 images – three to four orders of magnitude larger than the largest datasets used today (Section IV-C).
- Projecting these trends highlights that we will likely run out of vision data between 2030 to 2070 (Section IV-D).

## I. INTRODUCTION

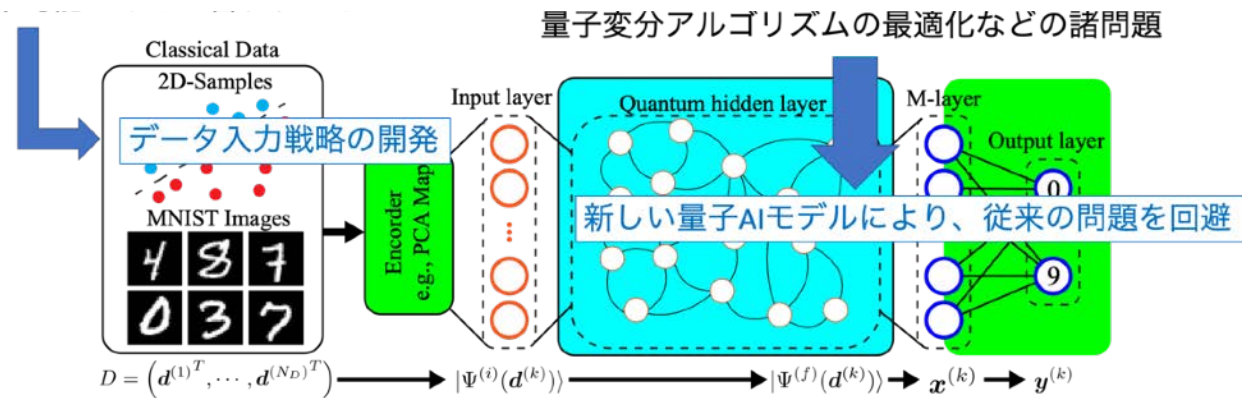
Training data is one of the three main factors that determine the performance of Machine Learning (ML) models, together with algorithms and compute. Current understanding of scaling laws suggests that future ML capabilities will strongly depend on the availability of large amounts of data to train large models [2, 3].

Previous work compiled a database of more than 200 training datasets used in ML models [1] and estimated historical rates of growth in dataset size for vision and language models.

# 量子機械学習の可能性

## 量子・古典ハイブリッド画像認識

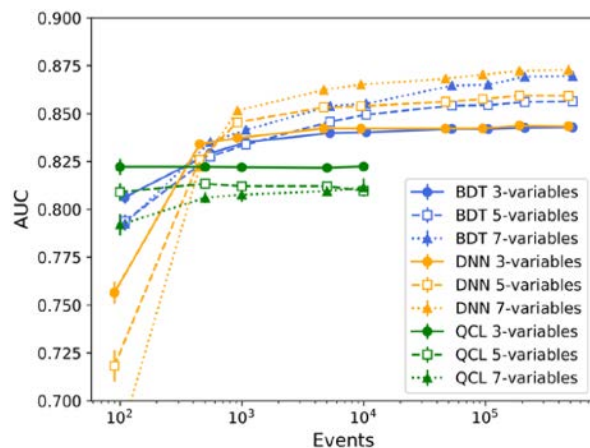
- 従来、画像認識はデータが大きすぎるため、量子機械学習は不可能であると考えられていた



## 古典検出器データの量子機械学習

A. Sakurai, et al. arXiv:2108.12131

- 少数データ・少数パラメータ量子機械学習



K. Terashi, et al. Comput. Softw. Big Sci. 5, 2 (2021)

拠点名称：量子ソフトウェアとHPC・シミュレーション技術の共創によるサステナブルAI研究拠点

代表機関	東京大学	プロジェクトリーダー	藤堂眞治 東京大学大学院理学系研究科 教授
参画機関	(大学等) 慶應義塾大学、理化学研究所、沖縄科学技術大学院大学、シカゴ大学 (企業等) 川崎市、Amoeba Energy株式会社、SCSK株式会社、株式会社Quemix、京セラ株式会社、JSR株式会社、株式会社TIER IV、凸版印刷株式会社、トヨタ自動車株式会社、日鉄ソリューションズ株式会社、International Business Machines Corporation、株式会社三井住友フィナンシャルグループ、株式会社日本総合研究所、株式会社バイトルヒクマ、BIPROGY株式会社、blueqat株式会社、みずほサーチ&テクノロジーズ株式会社、三菱ケミカル株式会社、株式会社三菱UFJフィナンシャル・グループ、村田機械株式会社		

プロジェクトの概要

量子ソフトウェアとHPC・シミュレーション技術の融合により、サステナブルなAI技術を開拓する。具体的には、少数データ・少数パラメータでも有効な量子機械学習手法と、固体・原子核・実時間ダイナミクスを扱える多体問題量子シミュレーション手法を開発し、テンソルネットワークなどの量子埋め込み技術とさまざまな量子最適化技術を駆使することで、量子機械学習・量子シミュレーション・量子計測デバイスを結合した量子AI技術を創出する。さらに、量子オフローディングや量子AIエッジコンピューティングのための量子HPC基盤を構築・展開する。それによりクラウドからエッジまで誰もがシームレスに量子AI技術を利用できる融合環境を実現し、健康・福祉、エネルギー問題といった重要な社会課題の解決に貢献する。量子HPC基盤の活用が進むことで、新しいアイデアに基づく新規ビジネス・企業の参入が促され、新たに直面する課題が次の世代の量子AI技術の研究開発を加速する。このような量子AI技術に支えられた「創発を生み出すサステナブルな好循環」を実装することで、我が国の生産性革命の促進や新産業創出・国際競争力が持続する社会の実現を目指す。



# 古典AIの課題を克服し「サステナブルAI」の創出へ

## ・AIの「スケーラビリティ」

- ・従来の機械学習:精度を上げるにはより**大量の学習データ**と**複雑な学習モデル**が必要
- ・ムーアの法則の終焉:HPC技術における**微細化・高速化の限界**

## ・量子AIの可能性・将来性

- ・近年の**量子デバイス技術**の進展
- ・**少数データ、少数パラメータ**による高度な量子機械学習
- ・古典計算が難しい**量子多体問題**の量子シミュレーション
- ・量子シミュレーション・量子計測と量子機械学習の**直接結合**
- ・**古典HPC基盤**との融合

## ・拠点ビジョン ⇒ 3つのターゲット ⇒ 5つの研究開発課題

### ビジョン



量子ソフトウェアとHPC・シミュレーション技術の共創によるサステナブルAIが拓く未来

- ・エネルギーの不安なしに情報技術を存分に活用できる社会
- ・携帯端末にいたるまで量子技術が普及し数千万人の人々がその恩恵を受けられる社会
- ・量子AIに支えられた生産性革命や新産業創出が持続する社会

### ターゲット

サステナブル量子機械学習手法の創出

量子AIにむけた多体問題量子シミュレーション手法の革新

量子HPC基盤の構築と展開

### 研究開発課題

1. 量子機械学習による高汎化技術と最適化の統合 (量子機械学習)

2. 物質・材料科学のための多体問題量子シミュレーション手法開発 (量子シミュレーション)

3. 量子埋め込みに基づく量子古典融合アルゴリズム開発 (量子埋め込み)

4. 量子機械学習/量子シミュレーションの高度化のための最適化技術開発 (量子最適化)

5. CPU~GPU~QPUの統合による量子HPC基盤構築 (量子HPC)

創薬

金融

自動運転

ライフサイエンス

プラント最適化

素粒子・原子核物理

プロセス最適化

材料開発

### 量子機械学習による高汎化技術と最適化の統合

- ・少数データ・少数パラメータに対する量子機械学習手法の開発
- ・データの符号化と学習・推論結果の統合技術の開発

### 物質・材料科学のための多体問題量子シミュレーション手法開発

- ・周期系、格子系に対する多体問題量子シミュレーション手法
- ・量子機械学習とのダイレクト結合によるマテリアルズ・インフォマティクス

### 量子埋め込みに基づく量子古典融合アルゴリズム開発

- ・テンソルネットワーク・サンプリングによる新しい量子/古典アルゴリズム
- ・テンソルネットワークの枠組みによる量子埋め込み

### 量子機械学習/量子シミュレーションの高度化のための最適化技術開発

- ・ゲート式量子コンピュータ/量子アニーリングマシン/イジングマシンと機械学習の融合アルゴリズム
- ・量子回路のゲート最適化・量子コンパイラ

### CPU~GPU~QPUの統合による量子HPC基盤構築

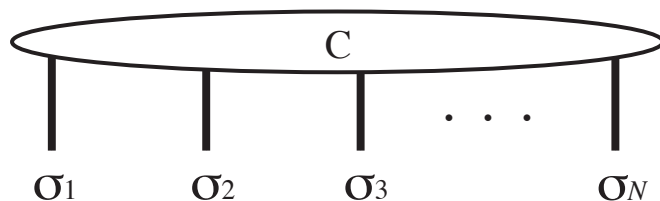
- ・新しいデータやアルゴリズム表現形式によるシームレスな統合
- ・量子コンピュータと次世代HPCIの統合環境構築とユーザへの展開

# 多体波動関数とテンソル




- $N$  量子ビット (spin-1/2 スピン) 系の多体波動関数

$$|\Psi\rangle = \sum_{\sigma_1, \sigma_2, \dots, \sigma_N} C_{\sigma_1, \sigma_2, \dots, \sigma_N} |\sigma_1 \sigma_2 \dots \sigma_N\rangle$$

- $2^N$  状態の重ね合わせ  $\rightarrow 2^N$  個の係数 ( $C_{\sigma_1, \sigma_2, \dots, \sigma_N}$ ) を決める必要がある  $\rightarrow$  データ量  $2^N$
- $C$  は  $N$  本脚 (階数  $N$ ,  $N$  階) の **テンソル** とみなすことができる



- **テンソル = 多次元配列 = ベクトルや行列の一般化**

- 0本脚のテンソル  $\rightarrow$  スカラー 
- 1本脚のテンソル  $\rightarrow$  ベクトル 
- 2本脚のテンソル  $\rightarrow$  行列 
- ...
- $N$  本脚のテンソル  $\rightarrow$  データ量、計算量とも  $N$  に関して **指数関数的** に増加



# さまざまなテンソル

## • テンソルによる表現

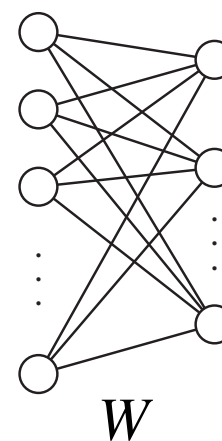
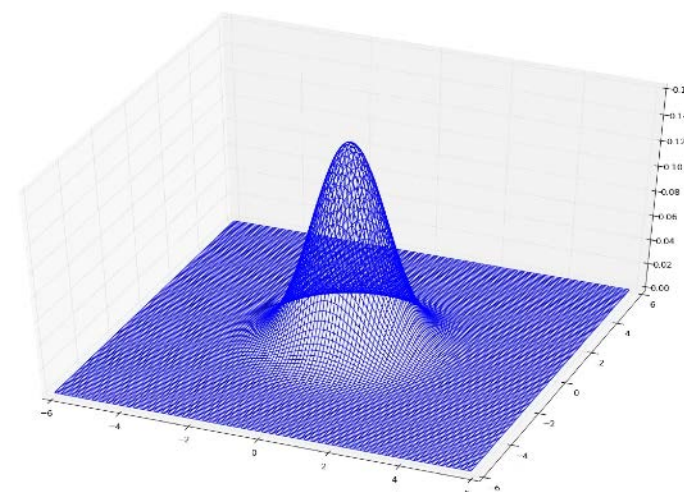
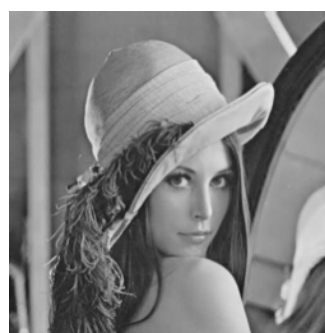
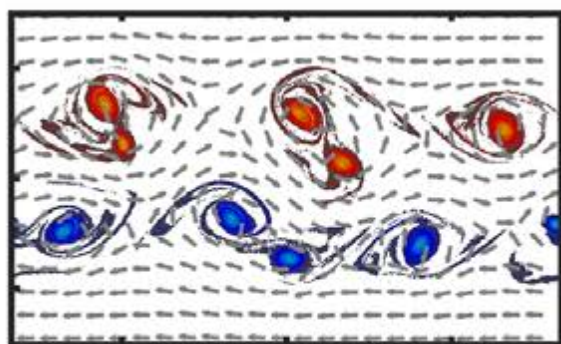
- 確率分布関数

$$P(s_1, s_2, \dots, s_N)$$

- 多次元データ
- グリッドデータ、画像

$$g(x, y) = g(x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N)$$

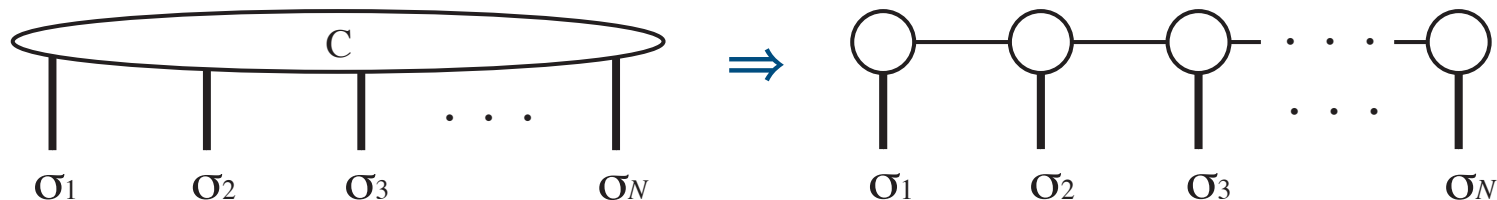
- $(x_1, x_2, \dots, x_N)$  と  $(y_1, y_2, \dots, y_N)$  はそれぞれ  $x$  と  $y$  の2進数表現
- 例) 256x256の画像は、256x256の行列あるいは  $2^{16}$  (16本脚)のテンソルとみなせる



- ニューラルネットワーク
  - 重み行列 → 多数の脚からなるテンソル

# テンソル分解

- 大きな(=脚の多い)テンソルを小さな(=脚の少ない)テンソルの積に分解

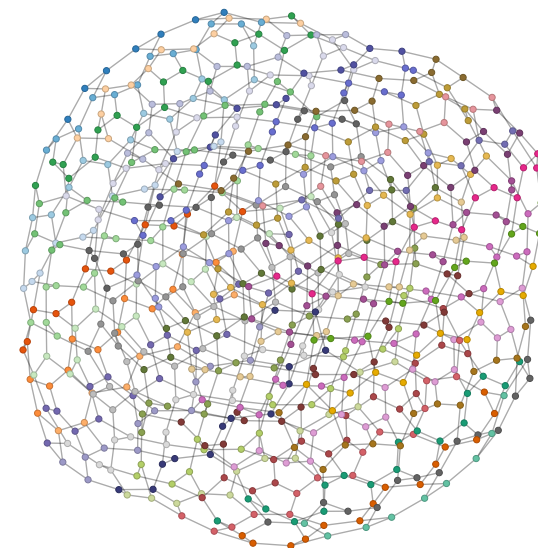


- テンソルをつなぐ線は脚の添字を揃えて和をとることを意味する(=縮約)
- テンソル  $\Rightarrow$  テンソル分解  $\Rightarrow$  テンソルネットワーク
- 他にも、ボルツマンマシン、経路積分、DNNなどさまざまな分解/表現手法
- テンソルネットワークの利点

- データ量の圧縮によるコンパクトな表現  $O(\exp(N)) \rightarrow O(N)$
- 縮約順序に関する自由度
- 特異値分解に基づく高精度近似

## テンソルネットワーク表現

- もともとの大きなテンソルの高精度近似(例: 変分波動関数)
- 厳密に分解できる場合もある(例: GHZ状態、AKLT状態)
- 問題そのものがテンソルネットワークで表現されている場合
  - 古典統計力学・量子統計力学、量子回路



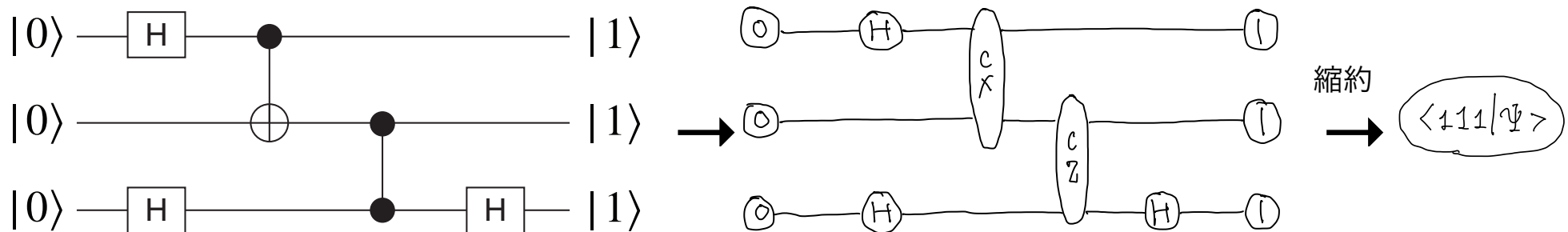
# 量子コンピュータの古典シミュレーション

---

- 量子ソフトウェア・アルゴリズムの精度・性能評価
  - 古典/量子ハイブリッドアルゴリズム、量子機械学習、量子誤り訂正
- 古典計算の限界を知る
  - 量子超越の本当の境界はどこか？
- 量子回路は典型的な量子多体系
  - 量子回路のシミュレーション技術 → 物理やその他の分野への展開
  - 超伝導・量子液体・素粒子・原子核
  - 複雑系・時系列データ・機械学習
- 量子コンピューティングの本質を知る
  - 別の形に書き直してはじめて理解できる
- 量子技術の古典コンピュータ上での利用
  - 量子コンピュータが普及する前に誰もがどこでも量子計算を利用できるように
- 旧来のシミュレーション手法 ( $N$  qbit系)
  - シュレディンガーシミュレーション(状態ベクトルシミュレーション): メモリ・計算量  $\sim \exp N$
  - ファインマンシミュレーション(経路数え上げ): 計算量  $\sim \exp N$
  - ファインマンサンプリング(量子モンテカルロ): 負符号問題

# 量子回路のテンソルネットワークシミュレーション

## 量子回路 ⇒ テンソルネットワーク



- 1ビットゲート → 2脚テンソル
- 2ビットゲート → 4脚テンソル
- 初期状態・出力状態 → 1脚テンソル(ベクトル)の組
- 出力状態の振幅 → テンソルネットワークを縮約することで得られる
- 量子回路では必ず左から右に時間が進む
  - テンソルネットワークの縮約はどのような順番で計算しても結果は同じ
- テンソルネットワークの古典コンピュータでの計算は非常にコストが高い
  - 縮約順序を工夫することで計算量を劇的に減らせる可能性
  - 近似的な縮約も可能(低ランク近似、テンソルくりこみ群)

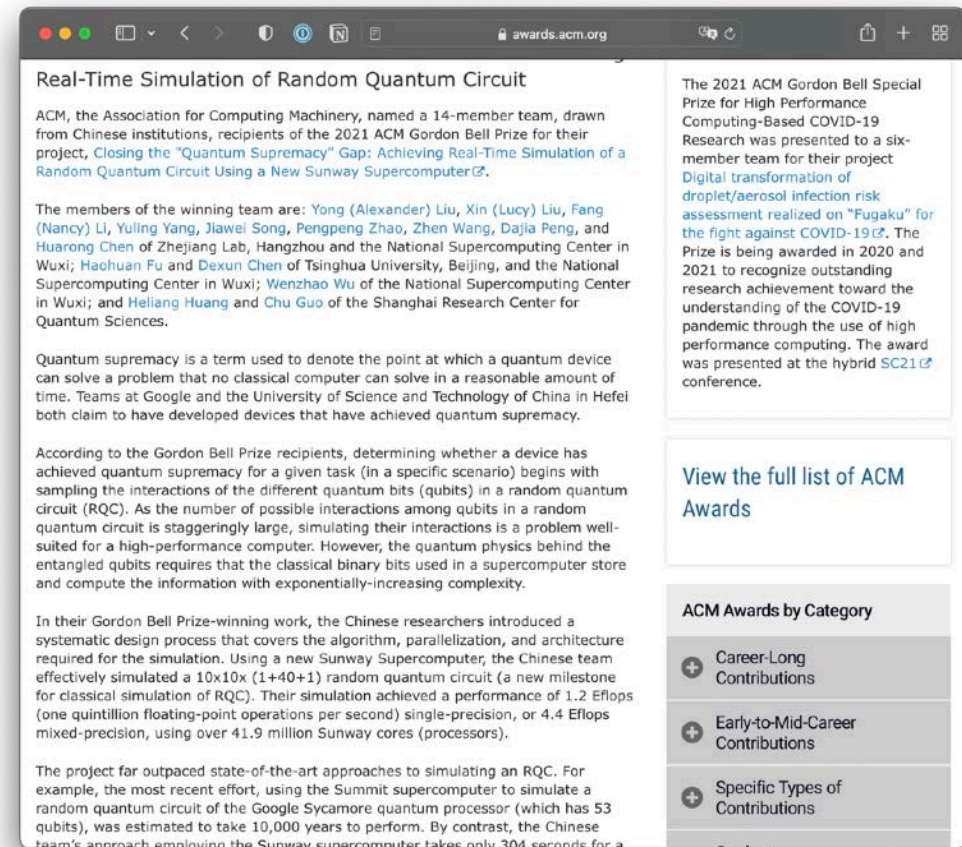
# 量子回路のテンソルネットワークシミュレーション

- Y. A. Liu et al., Closing the “quantum supremacy” gap: Achieving real-time simulation of a random quantum circuit using a new sunway supercomputer. International Conference for High Performance Computing, Networking, Storage and Analysis, SC (2021)

- Gordon Bell Prize Winner in 2021

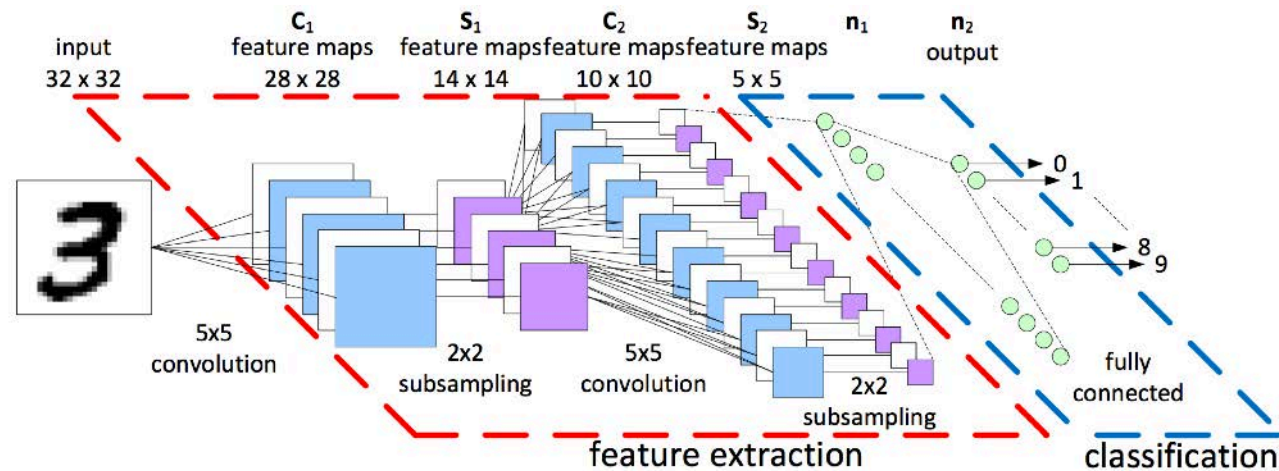
- しかしながらテンソルネットワーク規模が大きくなると厳密な縮約は結局不可能に
- 情報圧縮(SVD)を使った高精度近似
- テンソルネットワーク繰り込み群

<https://awards.acm.org/bell>



# 手書き文字認識

## ・畳み込みニューラルネットワーク(CNN)による機械学習



<https://www.kaggle.com/cdeotte/how-to-choose-cnn-architecture-mnist>

- 入力: グレースケール画像
- 出力: 10次元ベクトル
- 認識精度 > 99.5% (MNIST)

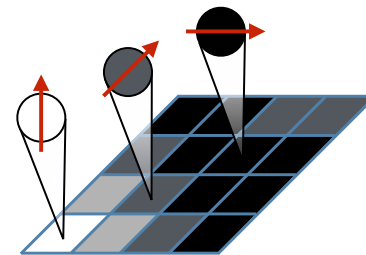
# テンソルネットワークによる教師あり学習

- テンソルネットワークは多重線形性をもつ
  - いかにして非線形性を入れて表現能力を高めるか？
- 入力データのエンコード
  - $N$ 次元ベクトル( $N$ ピクセル画像)を  $2^N$ 次元の特徴ベクトルに変換

$$\Phi^{s_1 s_2 \dots s_N}(\mathbf{x}) = \phi^{s_1}(x_1) \otimes \phi^{s_2}(x_2) \otimes \dots \otimes \phi^{s_N}(x_N) = \begin{array}{cccccc} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ \circ & \circ & \circ & \circ & \circ & \circ \\ \phi^{s_1} & \phi^{s_2} & \phi^{s_3} & \phi^{s_4} & \phi^{s_5} & \phi^{s_6} \end{array}$$

- $\Phi(x)$  は2次元局所特徴ベクトルのテンソル積

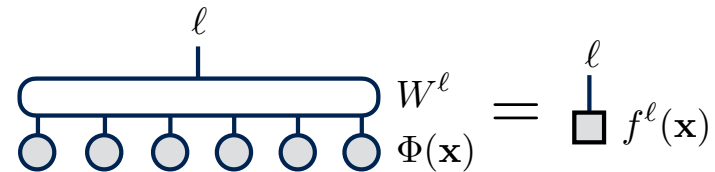
$$\phi^{s_j}(x_j) = \left[ \cos\left(\frac{\pi}{2}x_j\right), \sin\left(\frac{\pi}{2}x_j\right) \right]$$



# 分類モデル

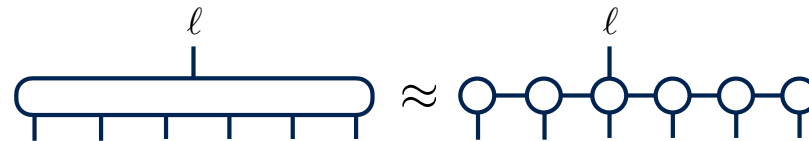
- $f^\ell(\mathbf{x}) = W^\ell \cdot \Phi(\mathbf{x})$

- $W$  は巨大な ( $10 \times 2^N$ ) 行列



- $W$  をテンソルネットワーク(行列積状態)に分解

$$W_{s_1 s_2 \dots s_N}^\ell = \sum_{\{\alpha\}} A_{s_1}^{\alpha_1} A_{s_2}^{\alpha_1 \alpha_2} \dots A_{s_j}^{\ell; \alpha_j \alpha_{j+1}} \dots A_{s_N}^{\alpha_{N-1}}$$

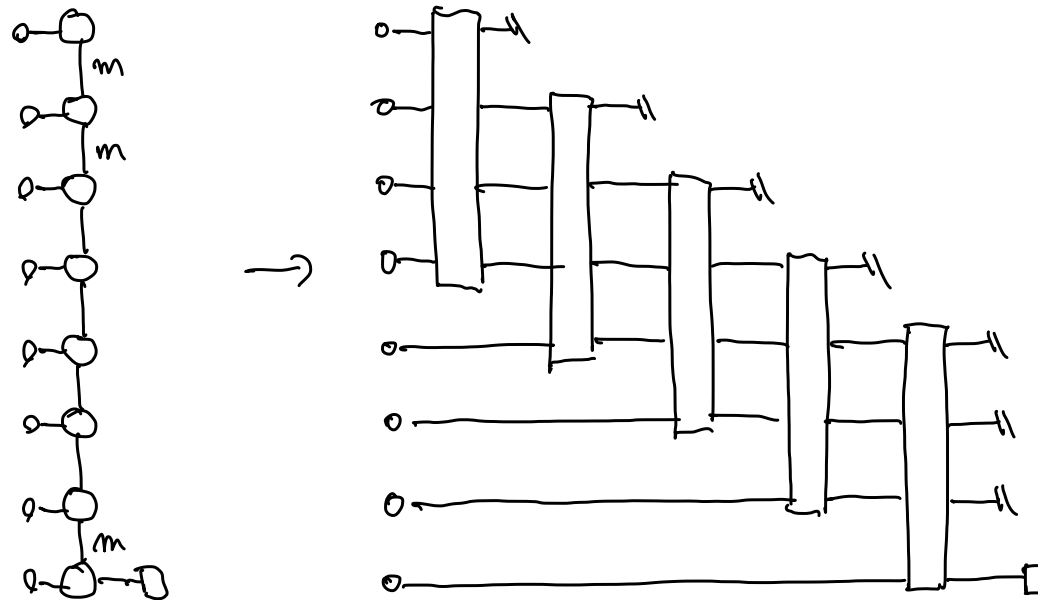


- それぞれのテンソルは  $2m^2$  個の要素をもつ ( $m$  はテンソル間のボンド次元)



# テンソルネットワークの量子回路への変換

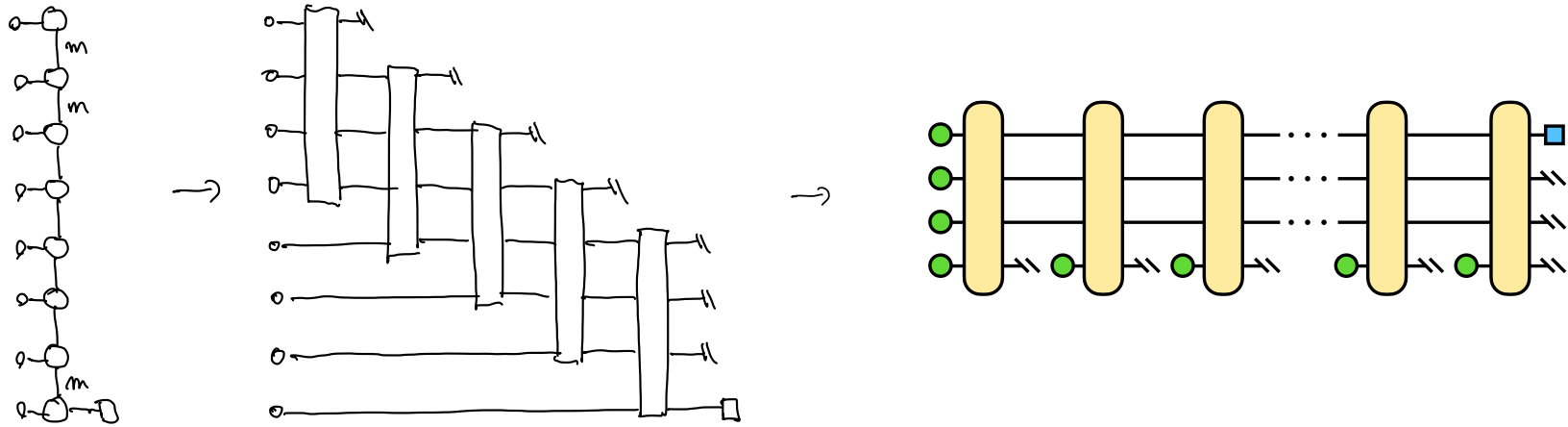
- $N = 8, m = 8$  の場合



- テンソルの最適化 → 量子回路のパラメータの最適化 → 量子・古典ハイブリッドアルゴリズム

# 量子ビット効率のよい実装

- 量子ビットを初期化して再利用



- 必要となる物理量子ビット数は、入力サイズ  $N$  に依存しない！

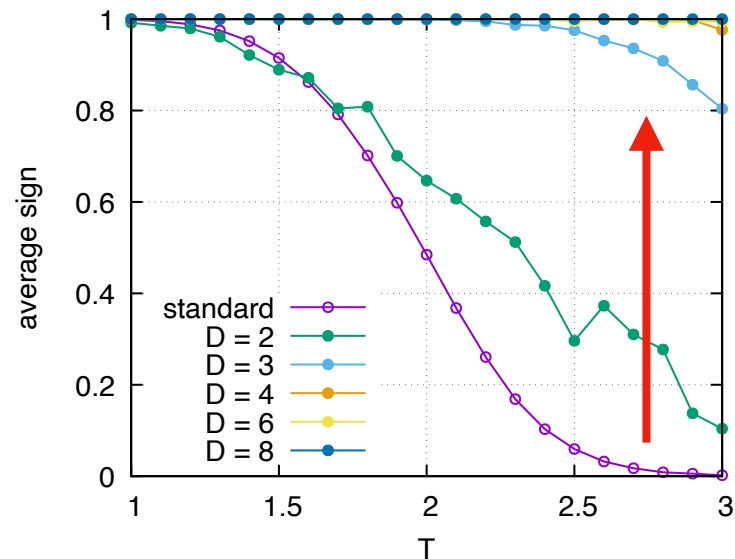
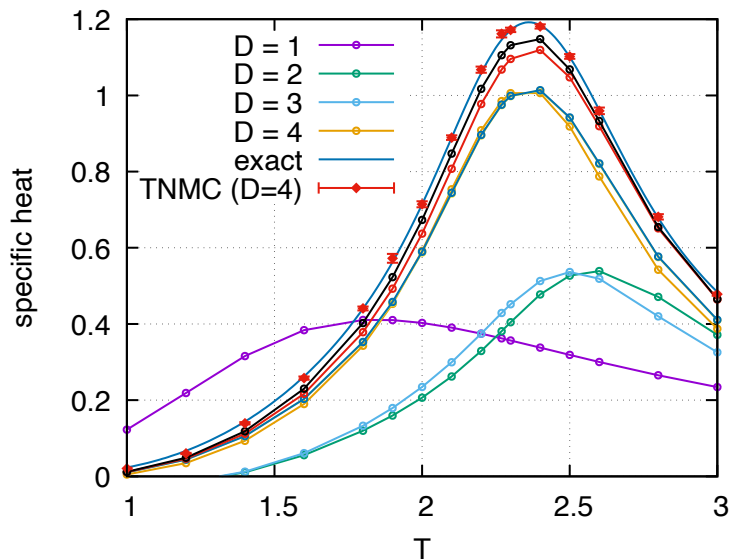
# 量子アルゴリズムの本質

---

- アダマールゲートなどを使い、多くの状態の重ね合わせを準備
  - この状態で非選択的射影測定を行うとエントロピーは  $O(N)$  (量子ビット数に比例)
- 欲しい答えが大きな振幅を持つように状態をユニタリー操作し干渉させる
  - 量子状態のエントロピーはゼロのまま
    - 非選択的射影測定をした場合のエントロピーは減少
    - 一種のデータ圧縮?
- 選択的射影測定する(測定して結果を見る)と正解が高い確率で得られる
  - エントロピーのできるだけ小さな状態からのサンプリング
    - 一種の情報抽出
- ⇒ 量子インスパイアードアルゴリズム

# テンソルネットワーク形式でのモンテカルロ法

- 経路積分に基づく従来の量子モンテカルロ法
  - 実時間発展の場合には深刻な負符号問題に直面
- 量子回路をテンソルネットワークとして表現
  - テンソルネットワーク表現のままサンプリングを実行
- 精度を落とさず高速化
  - 並列性
  - 特異値分解に起因する系統誤差を消去
  - 実時間発展・フラストレーション・フェルミ統計における負符号問題を解決



# まとめ「量子埋め込み」

---

- 「大きなテンソル」に「テンソルネットワーク」を埋め込む
  - 「大きなテンソル」 =
    - 量子状態、量子回路、古典コンピュータ、機械学習...
  - 「テンソルネットワーク」「ボルツマンマシン」 =
    - 近似波動関数・試行波動関数、古典模型・量子模型、量子回路...
- 古典コンピュータに量子回路を埋め込む
  - 量子回路シミュレータ ⇒ 量子コンピュータを使わずに量子AIを
  - 量子状態を量子状態として取り出す ⇒ 量子・古典インターフェース
- 量子回路にテンソルネットワークを埋め込む
  - 多項式時間で縮約可 ⇒ 古典・量子模型の量子シミュレーション
  - 異なる量子回路間の接続
- 前処理・変換の重要性
  - 量子回路やテンソルネットワークの書き換え、確率的な表現、観測との組み合わせ...
- 量子コンピュータのロードマップ