Quantum Machine Learning for Real-World, Large Scale Datasets with Applications in Earth Observation

Soronzonbold Otgonbaatar\textsuperscript{1,2}, Mihai Datcu\textsuperscript{1}, Xiao Xiang Zhu\textsuperscript{3}, and Dieter Kranzlmüller\textsuperscript{2}

\textsuperscript{1}German Aerospace Center (DLR) Oberpfaffenhofen, \textsuperscript{2}Ludwig-Maximilians-Universität München (LMU München), \textsuperscript{3}Technical University of Munich (TUM)

Contact: Soronzonbold.Otgonbaatar@dlr.de

AI4EO Symposium, Technical University of Munich (TUM)
13.10 – 14.10.2022
Contents

1. Quantum vs classical physics
2. Quantum vs classical computer
3. Machine learning on quantum computer or Quantum Machine Learning with 3 applications in Earth Observation
CHSH game

Referee: Charlie on the Earth

Player 1: Alice on the Mars

Player 2: Bob on the Moon

Players win if

\[ x \cdot y = a \text{xor} b; \ x, y, a, b \in \{0, 1\} \]
CHSH game: classical world

Player 1: Alice on the Mars

Referee: Charlie on the Earth

Player 2: Bob on the Moon

Players winning probability in classical world: 75 percent
CHSH game: quantum world

Referee: Charlie on the Earth

Player 1: Alice on the Mars

Player 2: Bob on the Moon

They share a so-called entangled quantum bits (particles).
CHSH game: quantum world

Referee: Charlie on the Earth

Player 1: Alice on the Mars

Entangled: non-local interaction (no classical channel)

Player 2: Bob

Players winning probability in quantum world: 85 percent
States **entangled** in a **quantum computer** yield higher correlation values (saw in CHSH game) than states in a **classical computer**. Classical Machine Learning involves the concepts of **probability and correlation**. Thus, this validates to study Machine Learning and deploy it on a **quantum computer**: Quantum Machine Learning (QML)
Classical & Quantum computer

CC: bits

QC: quantum bits (or qubits) which can exist in superposition and are entangled.

\[|0001110000001111111\rangle = c_1|110000110001111100\rangle + c_2|010100100101100100\rangle + \ldots + c_n|010100110001110111\rangle\]
Classical & Quantum computer

Transistor error:

\[ p \approx 10^{-27} \]

CC: bits

Qubit error:

\[ p \approx 10^{-3} \]

QC: quantum bits (or qubits)

Qubit error:

\[ p \sim 10^{-3} \]

An error-corrected quantum computer, say \( p \sim 10^{-27} \), is called a \textbf{fault-tolerant quantum computer}, and a \textbf{noisy-intermediate scale quantum computer (NISQ)}, say \( p \sim 10^{-13} \), otherwise.

QC: quantum bits (or qubits)
Computational Complexity

Classical Computer

Fault-Tolerant Quantum Computer

Bounded-Error Quantum Polynomial Time

NP-complete

NP

Post-quantum crypto?

Factoring

BQP

P

Quantum Algorithm Evolution

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer
Peter W. Shor

Abstract
A digital computer is generally believed to be an efficient universal computing device, that is, it is believed able to simulate any physical computing device with a finite number of states, and in a reasonable amount of time when quantum mechanics is taken into consideration. This paper contains several algorithms and proofs which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed implementations. Efficient quantum algorithms are given for three major problems on a hypothetical-quantum computer. These algorithms take a number of steps polynomial in the input size, i.e., the number of digits of the integer to be factored.

Keywords: quantum computer; prime factorization; discrete logarithm; Church-Turing's thesis; superdense coding; foundations of quantum mechanics; open systems; Peter Shor's algorithms

AMS subject classifications: 81P05, 11Y11, 81Q40, 81Q99

Quantum Factoring

- 1996
- 2009
- 2018
- 2020
Fast quantum (HHL) algorithm for a system of equations

1996

2009

2018

2020
A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone
Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139

Sam Gutmann

Abstract

We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as $p$ is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with $p$ times (at worst) the number of constraints. If $p$ is fixed, that is, independent of the input size, the algorithm makes use of efficient classical pre-processing. If $p$ grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed $p$. For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.0024 times the size of the optimal cut.
The power of quantum neural networks

Amiko Abbadi, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli and Stefan Woerner

It is unknown whether near-term quantum computers are advantageous for machine learning tasks. In this work we address this question by trying to understand how powerful and tractable quantum machine learning models are in relation to popular classical neural networks. We propose the effective dimension—a measure that captures these qualities—and prove that it can be used to assess any statistical model’s ability to generalise on new data. Crucially, the effective dimension is a data-dependent measure that depends on the Fisher information, a quantity that describes the geometry of a model’s parameter space and is central in both statistical and quantum machine learning.

We argue that the effective dimension is a robust measure through proof of a generalisation error bound and supporting numerical analyses, and thus use this measure to end the power of a popular class of neural networks in both classical and quantum learning settings.

Despite a lack of quantitative statements on the power of quantum neural networks another issue is related to the feasibility of these models. A precise connection between expressibility and stability for certain classes of quantum neural networks is outlined in ref. 1. Quantum neural networks often suffer from the barrier phase phenomenon, wherein the loss landscape is potentially flat and parameter optimisation is therefore extremely difficult.

As shown in ref. 2, barren plateaux may be noted when certain noise models are assumed on the hardware. In other words, the effect of hardware noise can make it very difficult to effectively train neural networks.

The quantum part, however, is of great interest. The quantum neural network has been known to have an advantage over classical neural networks, which is connected to the design of a neural network and reduces the complexity of the problem. Methods for such networks have been explored in refs. 3, 4. But more induced barren plateaux remain problematic.

A particular attempt to understand the best outcome of quantum models uses the Hamiltonian, which quantifies the properties of a model’s loss function as a point in parameter space. Properties of the Hamiltonian have been shown to be related to the power of the quantum algorithm for solving the problem.

Consequently, we can examine the difficulty of quantum and classical neural networks by studying the influence of parameter space, which is incorporated by the effective dimension. In this way, we can explicitly relate the effective dimension to model complexity.

We find that a class of quantum neural networks is able to achieve a considerably higher capacity and faster training ability compared to classical feedforward neural networks. A higher capacity is captured by a higher effective dimension, whereas

![Quantum Factoring](1996)

Fast HHL algorithm (2009)

Quantum Variational Algorithm (2015)

Power of quantum variational algorithms (2020)
Fault-tolerant quantum computers

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor

Abstract

A quantum computer is necessarily hard to be considered as an efficient universal computing device. The reason is that it is designed to simulate any physical computing device well as to simulate any algorithm that can be used to perform a polynomial-time calculation. This is why the quantum computer has to be designed to perform polynomial-time calculations. This is why the quantum computer has to be designed to perform polynomial-time calculations. This is why the quantum computer has to be designed to perform polynomial-time calculations. This is why the quantum computer has to be designed to perform polynomial-time calculations.

Keywords: algorithmic number theory, prime factorization, discrete logarithms, Church-Turing hypothesis, quantum computers, foundations of quantum mechanics, quantum cryptography, Shor's algorithm, Grover's algorithm

ASH subject classifications: 059, 1101, 081, 082

Power of QVA

Quantum Factoring

1996

Fast HHL algorithm

2009

Quantum Variational Algorithm (QVA)

2015

Noisy-intermediate scale quantum computers

The power of quantum neural networks

A QVA that is applied to a quantum neural network is shown to be able to outperform classical neural networks.

Quantum neural networks (QNNs) are a type of artificial neural network that uses quantum bits (qubits) instead of classical bits. They are able to perform certain tasks much faster than classical neural networks. However, QNNs are also more difficult to implement and require more resources.

The QNNs are used to solve a specific problem, such as image classification, by mapping the input data to a quantum state, which is then processed by the quantum circuit. The output of the quantum circuit is then measured and used to train the parameters of the classical neural network.

The QVA is a way to improve the performance of QNNs by using variational methods. The variational methods allow the user to optimize the parameters of the quantum circuit, making it more efficient.

The QVA is applied to a quantum neural network to classify images. The results show that the QVA is able to outperform classical neural networks.

Quantum computers

Quantum computers are a type of computer that uses quantum bits (qubits) to perform operations. They have the potential to solve certain problems much faster than classical computers. However, they are also more difficult to implement and require more resources.

Quantum computers are used to perform certain tasks, such as factorization, by mapping the input data to a quantum state, which is then processed by the quantum circuit. The output of the quantum circuit is then measured and used to train the parameters of the classical computer.

Quantum computers are also used to perform certain tasks, such as factorization, by mapping the input data to a quantum state, which is then processed by the quantum circuit. The output of the quantum circuit is then measured and used to train the parameters of the classical computer.

Quantum computers are also used to perform certain tasks, such as factorization, by mapping the input data to a quantum state, which is then processed by the quantum circuit. The output of the quantum circuit is then measured and used to train the parameters of the classical computer.
Noisy-intermediate scale quantum (NISQ) computers

Heterogeneous Computing: HPC+nQC

The power of quantum neural networks

CPU: Optimization

GPU: Multiplication and Addition

QVA

Power of QVA

2015

2020
Heterogeneous Computing: HPC+nQC

- Imperfect Quantum Computer
- Classical Computer
- NISQ nQPs: QVA
  - CPU: Optimization
  - GPU: Multiplication and Addition

nQC (or nQPs): superconducting, neural atomic, photonic quantum computers, or quantum annealer
Lets forget about quantum advantage. BUT What is exactly a quantum computer and How to make it work for machine learning tasks or for processing big datasets in Earth Observation?
Quantum Variational Algorithm for Earth Observation: Case I

Quantum Variational Algorithm for Earth Observation: Case I

Test QVA on a real-world RGB image of Berlin, Germany (trained QVA on Eurosat)
Quantum Variational Algorithm for Earth Observation: Case II

Data sets: different distribution

out-of-distribution? $\rightarrow$ physics to rescue
Quantum Variational Algorithm for Earth Observation: Case II

Data sets: different distribution

out-of-distribution? → quantum physics to rescue
Quantum Variational Algorithm for Earth Observation: Case II

PolSAR: San Francisco

PolSAR: DLR Oberpfaffenhofen
Quantum Variational Algorithm for Earth Observation: Case II

Say one 0-10 cm, other 20-30 cm wavelength
Quantum Variational Algorithm for Earth Observation: Case II

Quantum Variational Algorithm for Earth Observation: Case II
Quantum Annealer for Earth Observation: Case III

The Last Slide of This Talk

- **Input big data**
- **How to**
- **Small quantum computer**

**Hybrid classic-quantum computing**

- **Identify** Earth observation use cases (EO UCs) of practical relevance
- **Encoding**
- **Invent and analyse** quantum machine learning networks
- **Analyse** and test the integration and support between a quantum machine and a classical supercomputer

**Analyse** a quantum machine and a classical supercomputer for real-world, large scale datasets in order to obtain quantum advantage as early and efficiently as possible
Next Question: Can we really demonstrate **quantum advantage** by leveraging a HPC+nQC system over a conventional heterogeneous system, since we now know **what is** a quantum computer and **how to** make it work for large data sets?
Next Question: quantum advantage on HPC+nQC system
Next Question: quantum advantage on HPC+nQC system

<table>
<thead>
<tr>
<th>N</th>
<th>Number of causal DAGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>543</td>
</tr>
<tr>
<td>5</td>
<td>29281</td>
</tr>
<tr>
<td>6</td>
<td>3781503</td>
</tr>
<tr>
<td>7</td>
<td>1138779265</td>
</tr>
<tr>
<td>8</td>
<td>783702329343</td>
</tr>
<tr>
<td>9</td>
<td>1213442454842881</td>
</tr>
<tr>
<td>10</td>
<td>4175098976430598143</td>
</tr>
<tr>
<td>11</td>
<td>31603459396418917607425</td>
</tr>
<tr>
<td>12</td>
<td>521939651343829405020504063</td>
</tr>
<tr>
<td>13</td>
<td>19676600744432035186664816926721</td>
</tr>
<tr>
<td>14</td>
<td>1439428141044398334941790719839535103</td>
</tr>
</tbody>
</table>