Quantum enhanced machine learning: an overview

Pavlo V. Zahorodko[a], Yevhenii O. Modlo[d], Olga O. Kalinichenko[a], Tetiana V. Selivanova[a] and Serhiy O. Semerikov[a,b,c]

[a]*Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine*

[b]*Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine*

[c]*Institute of Information Technologies and Learning Tools of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*

[d]*State University of Economics and Technology, 5 Stephana Tilhy Str., Kryvyi Rih, 50006, Ukraine*

Abstract

Machine learning is now widely used almost everywhere, primarily for forecasting. The main idea of the work is to identify the possibility of achieving a quantum advantage when solving machine learning problems on a quantum computer.

Keywords

machine learning, quantum computing, quantum software engineering

1. Introduction

Traditionally, quantum computing is defined as a type of nonclassical computing that operates on the quantum state of subatomic particles, which represent information as elements denoted as quantum bits (qubits). A qubit can represent all possible values simultaneously (superposition) until read. Qubits can be linked with other qubits, a property known as entanglement. Quantum algorithms manipulate linked qubits in their undetermined (entangled) state, a process that can address problems with vast combinatorial complexity [1], reaching "quantum supremacy".

Identifying potential applications for quantum computing, Kasey Panetta points out that they "will be narrow and focused, as general-purpose quantum computing will most likely never be economical" [2]. In his opinion, quantum computing could enable breakthroughs by machine learning, finance, healthcare, creation of new materials, artificial intelligence (which requires 100s − 1000s qubits), chemistry and biochemistry (100–200 qubits). In particular, for finance, quantum computing could enable faster, more complex Monte Carlo simulations (for

CS&SE@SW 2020: 3rd Workshop for Young Scientists in Computer Science & Software Engineering, November 27, 2020, Kryvyi Rih, Ukraine

pavelzagorodko@outlook.com (P.V. Zahorodko); eugenemodlo@gmail.com (Y.O. Modlo); olgakalinichenko6@gmail.com (O.O. Kalinichenko); vitro090@gmail.com (T.V. Selivanova); semerikov@gmail.com (S.O. Semerikov)

https://kdpu.edu.ua/personal/ookalinichenko.html (O.O. Kalinichenko); https://kdpu.edu.ua/personal/vstania.html (T.V. Selivanova); https://kdpu.edu.ua/semerikov (S.O. Semerikov)

0000-0003-2037-1557 (Y.O. Modlo); 0000-0002-7057-2675 (O.O. Kalinichenko); 0000-0003-2635-1055 (T.V. Selivanova); 0000-0003-0789-0272 (S.O. Semerikov)

example, trading, trajectory optimization, market instability, price optimization and hedging strategies) and machine learning methods, which in the general case are reduced to problems of finding the extremum of a multidimensional function along the nonlinear response surface.

Currently, computing devices capable of performing quantum computing (quantum computers) are available for consumers of computing services using the QCaaS (quantum computing as a service) model. As of June 2020, the maximum number of qubits available for simultaneous use does not exceed 60, which is significantly less than the number required to achieve "quantum supremacy". This raises the problem of investigating the possibilities of quantum programming for machine learning tasks implementation, namely, the use of machine learning algorithms, implemented by the quantum programming language, to analyze traditional data and compare the performance of quantum and von-neumanns implementations at the present stage of their development.

2. Fundamentals of Quantum Software Engineering

2.1. Basic research concepts

Quantum computer is a computing device using quantum-mechanical phenomena (superposition, entanglement, etc.) for data transmission and processing.

Quantum programming is a software development process for quantum computer.

"Classical" applications of quantum computers (by Richard Feynman) – modeling complex [many-particle physical] systems: Zalka and Wiesner's algorithm.

"New" applications of quantum computers are tasks that require enumerating a large number of options: Grover's algorithm (general task), Shor's algorithm (factorization), Abrams and Lloyd's algorithm (identification of periodic properties), etc.

Quantum machine learning is an application of machine learning algorithms for quantum data analysis.

Quantum-enhanced machine learning is the use of machine learning algorithms implemented in the quantum programming language for the analysis of traditional data.

Software Engineering is a systematic application of engineering approaches to the design, implementation, testing and documenting of software.

2.2. Concept of Quantum Software Engineering

The first systems presentation of the Quantum Software Engineering concept was made by John Clark and Susan Stepney in 2002 [3]. Researchers believe that quantum computing cannot be effectively implemented in the traditional computer Von Neumann architecture, the mathematical model of which is the Turing machine. The authors [4] refer to the main challenges that Quantum Software Engineering will face in 2020:

- the question of what a quantum programming language should be – an extension of traditional languages, a logical programming language in a low-level programming language or a language that implements a new paradigm,
- the need to develop compilers for quantum programming languages,

- the need to develop new quantum algorithms and define the classes of traditional algorithms that can be quantised,
- feasibility of developing quantum computer simulators for use on traditional computer systems,
- despite the fact that quantum execution is in principle unobservable, debugging and testing techniques are necessary for quantum programming languages,
- quantum algorithms require visualization for their understanding, design, and implementation.

The criteria and success indicators of Quantum Software Engineering proposed by John Clark and Susan Stepney are summarized in table 1.

In 2020, Quantum Software Engineering includes such components [5]:

- Paradigms for developing quantum software
- Quantum software design
- Quantum software testing
- Quantum software verification
- Quantum software coding practices
- Quantum software reuse
- Quantum software experimentations
- Quantum software execution
- Industrial applications
- Empirical evaluations

In February 2020, at QANSWER 2020: 1st International Workshop on the QuANtum Soft-Ware Engineering & pRogramming, the Talavera Manifesto for Quantum Software Engineering and Programming [6] was adopted, containing a set of principles and commitments:

Quantum Software Engineering

- *is agnostic regarding quantum programming languages and technologies*;
- *embraces the coexistence of classical and quantum computing*, and advocates the use of reengineering techniques to integrate new quantum algorithms with the existing classical information systems. Reverse engineering techniques are also needed to parse and abstract quantum program information that is to be integrated into classical programs;

- *supports the management of quantum software development projects*, delivering quantum software that fulfils the initial business goal and requirements, while at the same time ensuring that quality, time, and cost constraints are being properly observed; methodologies for developing quantum programs must be created or adapted from the existing ones; effort estimation methods for quantum software development need to be provided as well;
- *considers the evolution of quantum software*: quantum software should be maintained and evolved from inception to removal, and quantum software evolution must be handled throughout the whole quantum software lifecycle;
- *aims at delivering quantum programs with desirable zero defects*: it is in charge of defining and applying testing and debugging techniques to quantum programs in such a way that most defects can be detected and solved before the program is released;
- *assures the quality of quantum software*: quality management for both process and product are essential if quantum software with expected quality levels is to be produced; since we cannot improve what we cannot measure, new metrics for quantum programs and quantum processes have to be developed;
- *promotes quantum software reuse*, helping development teams to share, index, and find quantum software that can be reused: this requires study of design and architectural patterns for quantum programs, facilitate technical communication, and work on creating libraries of reference examples and application demonstrations;
- *addresses security and privacy by design*: quantum information systems must be secure and guarantee the privacy of data and of users from the initial phases of quantum software development, i.e., by design;
- *covers the governance and management of software*: managers should be aware of the particular processes, organizational structures, principles, policies and frameworks, information, culture, ethics and behaviour, people, skills and competences, as well as the services, infrastructure and applications that are associated with quantum software and that are (or should be) provided by organizations.

The authors of the manifesto separately appeal to educators with a request to integrate quantum software engineering in curricula within the existing software engineering degrees and/or courses in this or other disciplines, and clearly specify which competences and skills are required for future quantum software engineers [6].

2.3. Quantum Software Engineering tools

The execution of quantum programs on personal computer equipment is difficult to access due to its lack of prevalence, so for more than a quarter-century, quantum simulators – software tools that simulate quantum circuits – have been the main means of their execution. The first mention of QCaaS (Quantum Computing as a Service) occurs only in 2015 in the article [7] by Mijanur Rahaman and Md. Masudul Islam.

The world's largest QCaaS providers:

- D-Wave Systems Inc. (Canada) – SDK Ocean [8] (Python, C++),
- International Business Machines Corporation (USA) – SDK ProjectQ [9] (Python), Qiskit [10] (Python),
- Cambridge Quantum Computing Limited (Great Britain) – SDK t|ket> [11] (Python),
- QC Ware, Corp. (USA) – SDK Forge (Python),
- StationQ – Microsoft (USA) – SDK LIQUi|> [12] (F#), Microsoft Quantum Development Kit [13] (F#),
- Rigetti Computing (USA) – SDK Forest [14] (Python).

Thus, the main programming language for cloud access to quantum computing is Python. Another criterion for choosing a QCaaS vendor is computing power, measured in qubits. This indicator is the largest in D-Wave Advantage – 5000 (in clusters of 8) qubits based on quantum annealing, which narrows the scope of its application to solving optimization problems, which boil down to finding the ground state for a set of spins. For universal quantum computers on quantum circuits, the number of qubits is significantly lower and today (June 2020) is the highest in IBM Q 53 (53 qubits) and Google Bristlecone (72 qubits). Unfortunately, Google's Quantum Computing Playground [15] is a browser-based quantum simulator, and there is no open cloud access to Google's Bristlecone. For cloud access to IBM Q, you can use both their library – Qiskit, and a third-party – ProjectQ. Considering that the highest level of specialization is provided by its own SDK, Qiskit was chosen for further work.

3. Quantum-enhanced machine learning

3.1. Quantum models of machine learning

Srinivasan Arunachalam and Ronald de Wolf in [16] offer three main quantum learning models:

1. *Quantum exact learning* based on membership queries to find the most accurate unknown function (quantum approximation problem). The efficiency of quantum algorithms in relation to classical ones in this case depends on how the learning efficiency is measured. If the measure of efficiency is the training time, then there are such classes of functions for which quantum algorithms are much faster than classical ones, assuming that the queries implementation in a quantum superposition is possible.
2. *Quantum Probably Approximately Correct* (PAC) learning to find an unknown function over a set of samples (quantum supervised learning). The difference between quantum PAC learning and classical learning is that the dataset can be in a state of quantum superposition.
3. *Quantum agnostic learning* to search for the $(n + 1)$-th bit, which is a continuation of a sequence with n bits (quantum prediction task).

The authors point to three types of complexity that arise when applying quantum learning models [16]:

1. query complexity of quantum exact learning: the number of quantum membership queries needed to exactly learn a target concept can be polynomially smaller than the number of classical membership queries, but not much smaller than that,
2. sample complexity: for the distribution-independent models of PAC and agnostic learning, quantum examples give no significant advantage over classical random examples: for every concept class, the classical and quantum sample complexities are the same up to constant factors. In contrast, for some fixed distributions (e.g., uniform) quantum examples can be much better than classical examples,
3. time complexity: there exist concept classes that can be learned superpolynomially faster by quantum computers than by classical computers, for instance based on Shor's or Simon's algorithm.

In the case of applying quantum machine learning models to the analysis of traditional data, we are talking about quantum-enhanced machine learning. Frank Phillipson [17] defines three main benefits of quantum machine learning:

- improving runtime (for example with a quantum hybrid Helmholtz machine)
- learning capacity improvements (for example with a quantum Hopfield neural network)
- learning efficiency improvements: less training information or simpler models needed to produce the same results or more complex relations can be learned from the same data

Various methods can be applied to increase the efficiency of training, one of which is variational quantum circuits – VQC [17].

Evidence of the intensity of quantum-enhanced machine learning development is the fact that the systematic review of the problem in 2016, carried out by Peter Wittek in [18], today (November 2020) is already considered as a classic, and that is indicated by the co-author in a new review [19].

Vedran Dunjko and Peter Wittek also highlight such perspective directions in the development of quantum machine learning in general:

- supervised and unsupervised learning: continuous-variable quantum neural networks, quantum convolutional neural networks, quantum algorithms for feedforward neural networks, Bayesian deep learning, sublinear quantum algorithms for training linear and kernel-based classifiers,
- reinforcement learning: quantum algorithms for solving dynamic programming problems (including hidden quantum Markov models), quantum gradient estimation.

The authors conclude that "the entire field of "genuinely quantum" machine learning (where the data itself is quantum) is still finding its right place and full recognition. Perhaps as quantum technologies mature, and problems of quantum learning become genuinely practical, the field

will crystallize and grow. ... In summary, QML [quantum machine learning] is diverse, growing, inclusive, and it is rich in open questions. ... Capturing all the QML trends, which will in the end be central is, for the time being, an impossible task – and, in a way, this is the key message of this note" [19].

3.2. An overview of quantum-enhanced machine learning tools in Qiskit

Qiskit provides the ability to develop quantum software both at the quantum circuits level using OpenQASM [20] and at a high level of abstraction using Python in a Jupyter notebook. The main components of the library are:

- quantum circuits modeling tools (Terra),
- implementation of standard quantum algorithms (Aqua – Algorithms for QUantum Applications), in particular, for solving optimization tasks
- cloud quantum computing tools (Aer),
- tools for simulating quantum noise (Ignis).

Aqua includes modules for research in finance (qiskit.finance), machine learning (qiskit.ml), optimization (qiskit.optimization) and chemistry (qiskit.chemistry) [20].

The machine learning module contains standard datasets and ways to access custom. Various optimization algorithms can be used to process them:

- ADMMOptimizer – an implementation of the ADMM-based heuristic (ADMM – alternating direction method of multipliers)
- CobylaOptimizer – the SciPy COBYLA optimizer (COBYLA – Constrained Optimization BY Linear Approximation)
- CplexOptimizer – the CPLEX optimizer for linear, integer and quadratic programming tasks
- GroverOptimizer – uses Grover Adaptive Search (GAS) to find the minimum of a QUBO function (QUBO – quadratic unconstrained binary optimization)
- MinimumEigenOptimizer – minimum eigen solvers
- RecursiveMinimumEigenOptimizer – a meta-algorithm that applies a recursive optimization

The qiskit.aqua.components.optimizers module offers a set of algorithms for local (Analytic quantum gradient descent optimizer, constrained optimization by linear approximation optimizer, Nelder-Mead optimizer, Nakanishi-Fujii-Todo algorithm, Powell optimizer, truncated Newton optimizer, etc.) and global optimizations (controlled random search with local mutation optimizer, evolutionary optimizer, etc.). It is advisable to use quantum support vector machine (QSVM) and variational quantum classifier (VQC) algorithms to solve classification tasks.

4. Conclusions

1. The core of Quantum Software Engineering is quantum programming – the process of developing programs for a quantum computer: a computing device that uses the phenomena of quantum mechanics to process data. Due to the low level of availability of such devices, it is advisable to access them under QCaaS model (quantum computing as a service). The conducted review of Quantum Software Engineering tools provided an opportunity to single out their main classes (quantum simulators, libraries, visualizers and cloud quantum services) and recommend using IBM Q as a hardware platform for quantum computing, Qiskit as a library of quantum algorithms, Python as a programming language and IBM Quantum Experience as QCaaS Provider.
2. The use of machine learning algorithms for the analysis of quantum data can be described by three quantum machine learning models (quantum exact learning, quantum Probably Approximately Correct learning and quantum agnostic learning), in the application of which there are three types of difficulties associated with the query complexity of quantum exact learning, quantum the intricacy of datasets and the sensitivity of quantum algorithms to them. A prospective direction in the machine learning development is the use of quantum learning models for analyzing traditional data, the implementation of which in Qiskit Aqua 0.7.3 is still a limited solution to classification tasks.

References

[1] Gartner, Quantum Computing Gartner Glossary, 2021. URL: https://www.gartner.com/en/information-technology/glossary/quantum-computing.

[2] K. Panetta, The CIO's Guide to Quantum Computing, 2019. URL: https://www.gartner.com/smarterwithgartner/the-cios-guide-to-quantum-computing/.

[3] J. Clark, S. Stepney, Quantum software engineering, in: Workshop on Grand Challenges for Computing Research, e-Science Institute, Edinburgh, 2002. URL: http://web.archive.org/web/20200721161705/http://www.ukcrc.org.uk/press/news/call/a5.cfm.

[4] C.-H. Chenf, L.-Y. Wei, New entropy clustering analysis method based on adaptive learning, in: P. P. Wang (Ed.), Information Sciences 2007: Proceedings of the 10th Joint Conference, Salt Lake City, Utah, USA, 18–24 July 2007, 2007, pp. 1196–1202. doi:10.1142/9789812709677_0169.

[5] Q-SE2020, First International Workshop on Quantum Software Engineering (Q-SE 2020) co-located with ICSE 2020, 2021. URL: https://q-se.github.io/qse2020/.

[6] M. Piattini, G. Peterssen, R. Perez-Castillo, J. L. Hevia, M. A. Serrano, G. Hernández, I. G. R. de Guzmán, C. A. Paradela, M. Polo, E. Murina, L. Jiménez, J. C. Marqueño, R. Gallego, J. Tura, F. Phillipson, J. M. Murillo, A. Niño, M. Rodríguez, The Talavera Manifesto for Quantum Software Engineering and Programming, CEUR Workshop Proceedings 2561 (2020) 1–5.

[7] M. Rahaman, M. M. Islam, A Review on Progress and Problems of Quantum Computing as aService (QCaas) in the Perspective of Cloud Computing, Global Journal of Computer Science and Technology: B Cloud and Distributed 15 (2015) 15 – 18. URL: https://globaljournals.org/GJCST_Volume15/3-Cloud-Data-Storage.pdf.

[8] D-Wave Systems Inc, D-Wave Ocean Software Documentation, 2021. URL: https://ocean.dwavesys.com/.
[9] D. Steiger, T. Häner, ProjectQ – Open Source Software for Quantum Computing, 2017. URL: https://projectq.ch/.
[10] Qiskit, Qiskit, 2021. URL: https://qiskit.org/.
[11] Cambridge Quantum Computing, Technology, 2020. URL: https://cambridgequantum.com/technology/.
[12] Microsoft, Language-Integrated Quantum Operations: LIQUi|>, 2016. URL: https://www.microsoft.com/en-us/research/project/language-integrated-quantum-operations-liqui/.
[13] Microsoft, Microsoft Quantum Documentation and Q# API Reference - Microsoft Quantum, 2021. URL: https://docs.microsoft.com/en-us/quantum/.
[14] Rigetti Computing, Rigetti QCS, 2020. URL: https://qcs.rigetti.com/sdk-downloads.
[15] Google, Quantum Computing Playground, 2016. URL: http://www.quantumplayground.net.
[16] S. Arunachalam, R. de Wolf, A Survey of Quantum Learning Theory, 2017. arXiv:1701.06806.
[17] F. Phillipson, Quantum Machine Learning: Benefits and Practical Examples, CEUR Workshop Proceedings 2561 (2020) 51–56.
[18] P. Wittek, Quantum Machine Learning: What Quantum Computing Means to Data Mining, Elsevier Insights, Academic Press, San Diego, 2016.
[19] V. Dunjko, P. Wittek, A non-review of Quantum Machine Learning: trends and explorations, Quantum Views 4 (2020) 17. doi:10.22331/qv-2020-03-17-32.
[20] M. Pistoia, J. Gambetta, Qiskit Aqua – A Library of Quantum Algorithms and Applications, 2018. URL: https://medium.com/qiskit/qiskit-aqua-a-library-of-quantum-algorithms-and-applications-33ecf3b36008.

Table 1
The criteria and success indicators of Quantum Software Engineering (according to [3])

Criteria	Indicators
It arises from scientific curiosity about the foundation, the nature or the limits of a scientific discipline	Quantum computation has broadened the fundamental limits of computer science and software engineering
The ability to create new engineering solutions	The physical infrastructure is constantly evolving, each solution is new
Technological continuity	The existence of high level languages and development techniques that can be used by computer scientists and software engineers with only the same style of training they receive today (so, no need to teach the fundamentals of quantum mechanics to all)
Research community support	Support for all interested in new computing paradigms and new levels of computing power
International character of research	This is a new fundamental area of software engineering
It is generally comprehensible, and captures the imagination of the general public, as well as the esteem of scientists in other disciplines	It is not generally understood, but is known for its worldwide interpretation
The problem has a long-standing statement, but has not yet been resolved	Formulated by Richard Feynman in the late 1970s
It promises to go beyond what is initially possible, and requires development of understanding, techniques and tools unknown at the start of the project	Problems exist on every level, from developing a whole new conceptual paradigm, to building intellectual and simulation tools
It calls for planned co-operation among identified research teams and communities	Research is needed in a number of areas (languages, algorithms, tools, simulation, visualisation, etc.)
It encourages and benefits from competition among individuals and teams, with clear criteria on who is winning, or who has won	There need not be a single “winner”, diversity of solutions should be encouraged, as in classical software engineering, to be applicable to a range of application domains
It decomposes into identified intermediate research goals, whose achievement brings scientific or economic benefit, even if the project as a whole fails	There are several components of the problem that can be explored in parallel
It will lead to radical paradigm shift	Quantum computing is a radical paradigm shift