Information systems. Computer sciences. Issues of information security

Информационные системы. Информатика. Проблемы информационной безопасности

UDC 004.032.26
https://doi.org/10.32362/2500-316X-2023-11-5-19-33

RESEARCH ARTICLE

Geometric properties of quantum entanglement and machine learning

Sergei V. Zuev @

V.I. Vernadsky Crimean Federal University, Simferopol, 295007 Russia
@ *Corresponding author, e-mail: zuevsv@cfuv.ru*

Abstract
Objectives. Fast data analysis based on hidden patterns is one of the main issues for adaptive artificial intelligence systems development. This paper aims to propose and verify a method of such analysis based on the representation of data in the form of a quantum state, or, alternatively, in the form of a geometric object in a space allowing online machine learning.
Methods. This paper uses Feynman formalism to represent quantum states and operations on them, the representation of quantum computing in the form of quantum circuits, geometric transformations, topological classification, as well as methods of classical and quantum machine learning. The Python programming language is used as a development tool. Optimization tools for machine learning are taken from the SciPy module. The datasets for analysis are taken from open sources. Data preprocessing was performed by the method of mapping features into numerical vectors, then the method of bringing the data to the desired dimension was applied. The data was then displayed in a quantum state. A proprietary quantum computing emulator is used (it is in the public domain).
Results. The results of computational experiments revealed the ability of very simple quantum circuits to classify data without optimization. Comparative indicators of classification quality are obtained without the use of optimization, as well as with its use. Experiments were carried out with different datasets and for different values of the dimension of feature spaces. The efficiency of the models and methods of machine learning proposed in the work, as well as methods of combining them into network structures, is practically confirmed.
Conclusions. The proposed method of machine learning and the model of quantum neural networks can be used to create adaptive artificial intelligence systems as part of an online learning module. Free online optimization learning process allows it to be applied in data streaming, that is, adapting to changes in the environment. The developed software does not require quantum computers and can be used in the development of artificial intelligence systems in Python as imported modules.

Keywords: online learning, adaptive artificial intelligence, quantum machine learning, quantum entanglement

• **Submitted:** 13.02.2023 • **Revised:** 14.06.2023 • **Accepted:** 13.08.2023

For citation: Zuev S.V. Geometric properties of quantum entanglement and machine learning. *Russ. Technol. J.* 2023;11(5):19–33. https://doi.org/10.32362/2500-316X-2023-11-5-19-33

Financial disclosure: The author has no a financial or property interest in any material or method mentioned.

In addition to the noted affiliation, the author is an employee of the V.G. Shukhov Belgorod State Technological University (Belgorod, Russia). However, although this study is carried out independently of this organization, the author has an obligation to indicate his affiliation with it.

НАУЧНАЯ СТАТЬЯ

Геометрические свойства квантовой запутанности и машинное обучение

С.В. Зуев @

Крымский федеральный университет им. В.И. Вернадского, Симферополь, 295007 Россия
@ Автор для переписки, e-mail: zuevsv@cfuv.ru

Резюме
Цели. Быстрая классификация данных на основе имеющихся в них закономерностей является одним из главных вопросов для построения систем адаптивного искусственного интеллекта. Цель работы – предложить и верифицировать метод такой классификации на основе представления данных в виде квантового состояния или (альтернативно) в виде геометрического объекта в пространстве, свойства которого позволяют производить машинное обучение «на лету» (онлайн-обучение).
Методы. В работе используется фейнмановский формализм для представления квантовых состояний и операций над ними, представление квантовых вычислений в виде квантовых схем, геометрические преобразования, топологическая классификация, а также методы классического и квантового машинного обучения. В качестве инструмента разработки использовался язык программирования Python, средства оптимизации для машинного обучения взяты из модуля SciPy. Размеченные данные для анализа взяты из открытых источников. Препроцессинг данных произведен методом отображения признаков в числовые векторы, затем применен метод приведения данных к нужной размерности и далее – отображение данных в квантовое состояние. Используется собственный эмулятор квантовых вычислений (находится в открытом доступе).
Результаты. Результаты вычислительных экспериментов выявили способность очень простых квантовых схем к классификации данных без оптимизации. Получены сравнительные показатели качества классификации без использования оптимизации, а также с ее использованием. Эксперименты проведены с различными датасетами и для различных значений размерности пространств признаков. Работоспособность предложенных в работе моделей и методов машинного обучения, а также методов их объединения в сетевые структуры, подтверждена практически.
Выводы. Предложенный метод машинного обучения и построения квантовых нейронных сетей может быть применен для создания систем адаптивного искусственного интеллекта в составе модуля онлайн-обучения. Отсутствие оптимизации в процессе онлайн-обучения позволяет применять его в потоке данных, т.е., адаптироваться к изменениям среды. Разработанное алгоритмическое обеспечение не требует наличия квантовых компьютеров и может быть применено при разработке программного обеспечения систем искусственного интеллекта на языке Python в качестве импортируемых модулей.

Ключевые слова: онлайн-обучение, адаптивный искусственный интеллект, квантовое машинное обучение, квантовая запутанность

• **Поступила:** 13.02.2023 • **Доработана:** 14.06.2023 • **Принята к опубликованию:** 13.08.2023

Для цитирования: Зуев С.В. Геометрические свойства квантовой запутанности и машинное обучение. *Russ. Technol. J.* 2023;11(5):19–33. https://doi.org/10.32362/2500-316X-2023-11-5-19-33

Прозрачность финансовой деятельности: Автор не имеет финансовой заинтересованности в представленных материалах или методах.

Кроме отмеченной аффилиации, автор является сотрудником БГТУ им. В.Г. Шухова (Белгород, Россия), но настоящая работа выполнена независимо от этой организации, хотя у автора имеется обязанность указать свою принадлежность к ней.

INTRODUCTION

Quantum algorithms are attracting more and more attention, since quantum computers are soon expected to be fully usable. On the other hand, quantum search and factorization algorithms are one of the main reasons for developing quantum computers. Several such computers are currently available around the world. However, their power is relatively low (the largest is the Chinese Jiuzhang with 76 qubits[1]), and they are still used for demonstration and research purposes.

The other pole of progress in information technology is artificial intelligence. Like most human knowledge, artificial intelligence is based on a natural phenomenon known as cognition, which still has no universally recognized quantitative theory. Quantum versions of such theories, inter alia, do not yet have proven clear advantages, although the results of this study can be seen as an indication of certain advantages of quantum methods in machine learning (ML).

The advantages of quantum computing and quantum computers in solving problems in the field of artificial intelligence are covered in review [1], which substantiates the relevance of studies in this area and indicates the main directions in one of which this paper is written. There is also a review of 2023 [2] containing references to all modern advances in this area.

The main possible advantage of quantum version of artificial intelligence is the exponential growth of computational capabilities. While the classical artificial neuron can process input data of N dimensions, quantum neuron can process 2^N-dimensional data. The application of the quantum version can significantly speed up execution of both learning and classification algorithms [3]. At the same time, one of the technical challenges in building a large-scale quantum computer is the need to ensure that there are "qubits that can be initialized with arbitrary values" [4]. This problem is relevant and is a significant obstacle to achieving quantum superiority.

In [5–9], prototypes of quantum neural networks based on constructing a quantum circuit with adjustable parameters were proposed. The present paper shows how this approach can be implemented in connection with the proposed neural network architecture and how such parameter settings can be dispensed with.

Quantum versions of the most popular ML algorithms have already been developed. The above-mentioned quantum neural networks work on a par with traditional ones. In [10], quantum support vector machines (SVMs) using the HHL algorithm [11] for inverting a matrix to generate a hyperplane were proposed. The image classification model presented in 2018 [12] is based on quantum *k*-nearest neighbors. The quantum linear regression using quantum data is proposed in [13]. A quantum analogue of the decision tree developed in [14] uses quantum accuracy and quantum entropy measurement, i.e., it develops the classical ID3 algorithm.

Several quantum ML methods have been developed for clustering in [15]. In particular, a quantum version of the *k*-means algorithm in different variants is presented in [16] and [17]. Another quantum clustering algorithm using Grover's algorithm to determine the cluster median is proposed in [18].

The quantum analogue of the quantum principal component analysis method [19] identifying eigenvectors related to the eigenvalues of an unknown state exponentially faster than any other solution has also been developed.

An area close to the topic of the paper is reinforcement learning, i.e., online learning taking into account the response of the environment. There are several quantum versions of reinforcement learning, such as [20] which uses a superposition of quantum states, and due to this parallelism is achieved and the learning speed increases.

Deep learning occupies a special place in ML. Deep learning methods require significant memory and time resources, thus making their development in the quantum area attractive. Among recent advances in this field is a series of works on quantum generative adversarial networks [21–23] with implementation in [24] using a superconducting quantum processor to generate and learn handwritten digital images by quantum generative Wasserstein adversarial networks [25]. It has also been shown that scalability and stability of quantum generative adversarial model learning improves on quantum Boltzmann machines [26, 27], quantum autocoders [28, 29], and quantum convolutional neural networks [7–9]. Among Russian works in this area, study [6] may be specified.

Improving optimization algorithms is also in the focus of research on quantum algorithms. Quantum enhanced optimization [30] as well as quantum gradient descent [31, 32] is used in quantum neural networks, e.g., in quantum Boltzmann machines [27].

Among recent works is experimental study [33] showing that SVMs outperform their classical counterparts by 3–4% on average, while quantum neural networks made on a quantum computer outperform quantum SVMs by 5% on average, and classical neural networks by 7%.

Quantum entanglement in connection with a model of learning was proposed in 2005. This is a model for the semantics of concept combinations created in a non-decomposition way. It deals with emergent properties/associations/inferences in connection with concept combinations[2]. In the paper, this idea is used

[1] https://en.wikipedia.org/wiki/Jiuzhang_(quantum_computer). Accessed January 01, 2023.

[2] Bruza P.D., Cole R.J. *Quantum Logic of Semantic Space: An Exploratory Investigation of Context Effects in Practical Reasoning*, 2006. https://arxiv.org/abs/quant-ph/0612178. Accessed January 01, 2023.

for a different purpose, i.e., to provide a way to separate labeled data. Although not directly related to learning, these ideas may allow a better way to be found to solve the following current problems of data analysis and artificial intelligence.

The first problem is online learning. It arises when the data environment changes and there is no time or resources for new learning in the system. The comprehensive theory of online learning is presented in the course by Massachusetts Institute of Technology, available online[3]. The main challenge in this problem is finding a compromise between quality and responsiveness. Quality-based learning is often time-consuming, while responsiveness-based learning may produce useless results. The better choice is to create a system that adjusts itself with allowance for the content of the data stream it receives. The paper proposes such a system based on entangled quantum states. Generalizing this idea, it is possible to approximate artificial intelligence systems to living intelligence in the sense of adapting to the environment.

The second problem is fast recognition, especially in the case of moving images. This problem is well described in various blogs and articles. For example, one of the current approaches to this issue is presented by Shao and Vitarsia in [34]. This research focuses on applying the BP neural network, i.e., an artificial neural network of forward propagation. The application of quantum algorithms for solving this problem has not been found in the literature. However, there is a software tool[4] designed to compare streaming video data which already works as a web service[5].

Any progress in solving these problems could lead to technological solutions in industries such as self-driving cars, unmanned aerial, and underwater vehicles, as well as video monitoring and other fields largely related to the detection of anomalies in a changing environment. It is not necessary to use a quantum computer to apply the results of this research, since the proposed algorithms can be implemented on emulators or reformulated in classical form. This would probably destroy quantum superiority, although the efficiency of low-dimensional data may be quite sufficient.

MATERIALS AND METHODS

This paper uses the quantum and classical data described below. The set $\{x^j\}$ of sets of n real numbers $\mathbf{x}^j = \{x_0^j, ..., x_{n-1}^j\}$ with label l^j defined for each set is the *classical* data. The set of quantum states

$$|q^j\rangle \equiv \sum_{k=0}^{2^n-1} a_k^j |k\rangle$$

is considered as *quantum* data. Components a_k^j of the quantum state vector are considered as given in a certain computational basis $|0\rangle, ..., |2^n-1\rangle$. These notations are commonly used in such books as [35]. Before the relationship between these data is established, we shall make a few preliminary remarks.

The type of data used to deliver information from system to system in nature is not obvious. However, human operations require classical information. It can be easily seen that 2^n-dimensional quantum system provides only n-dimensional classical data, although the quantum system operates in 2^n-dimensional state space during calculations. The problem of generating the initial quantum state arises from the fact that the source of quantum data, generally speaking, is unknown. It is certainly impossible to generate this data from classical ones. Thus, the only thing that can definitely be assumed is that the system has already had data in quantum form before the start of computation However, this means that all dependencies are already contained in quantum data, and the quantum intelligent system should use them. This is the basis for further consideration.

We shall first describe the state space structure of the system of n qubits. Proceeding from the way in which quantum data is represented, this space is embedded in $\mathbb{C}^N$, where $N = 2^n$ while $\mathbb{C}$ is the space of complex numbers. In addition, quantum states are described by vectors with an absolute value equal to 1, while vectors differing only by phase coefficient $e^{i\varphi}$ describe the same state. This suggests that the equivalence relation may be considered, as follows:

$$\frac{z^1}{z^0} \equiv w^1, ..., \frac{z^{N-1}}{z^0} \equiv w^{N-1}. \quad (1)$$

The space of such vectors $\mathbf{w}$ is called (*complex*) *projective space* $\mathbb{C}P^{N-1}$, which is a set of vectors with N complex coordinates $(z^1, \ldots, z^{N-1})$ connected by equivalence relation (1). Another condition which can be derived from (1) is the following:

$$|z^0|^2 + ... + |z^{N-1}|^2 = 1. \quad (2)$$

The phases of coordinates w^k are defined to the accuracy of the common multiplier $e^{-i\phi_0}$, where ϕ_0 is

[3] Rakhlin A. *Online Methods in Machine Learning. Theory and Applications*. TA: Arthur Flajolet. https://www.mit.edu/~rakhlin/6.883/. Accessed January 01, 2023.

[4] Biloushenko I.I., Zuev S.V. *Determining the degree of similarity of video fragments*, 2022; Certificate 2022685057 of 20.12.2022 issued by the Federal Service for Intellectual Property (in Russ.).

[5] https://ais.bstu.ru/services/1. Accessed July 05, 2023.

an arbitrary phase of coordinate z^0. Thus, space $\mathbb{C}P^{N-1}$ can be identified with the space of system states of n qubits while coordinates can be represented in the following form:

$$w^k = \frac{|z^k|}{|z^0|} e^{i(\phi_k - \phi_0)}.$$

We can assume without any restriction that $\phi_0 = 0$. According to the above procedure, the space $\mathbb{C}P^{N-1}$, is homeomorphic to surface (2) of dimension $2N-2$ since $z^0_{\text{im}} = 0$ and, therefore,

$$\left|z^0\right|^2 + \ldots + \left|z^{N-1}\right|^2 = \left(z^0_{\text{re}}\right)^2 + \left(z^1_{\text{re}}\right)^2 + \\ + \left(z^1_{\text{im}}\right)^2 + \ldots + \left(z^{N-1}_{\text{re}}\right)^2 + \left(z^{N-1}_{\text{im}}\right)^2 = 1.$$

This is sphere S^{N-2}, and each of its points z_j can be parameterized using the following generalized spherical coordinates:

$$\begin{aligned} z_j^0 &= \cos\delta_j^0 \cos\delta_j^1 \ldots \cos\delta_j^{2^n-2}, \\ z_j^1 &= \sin\delta_j^0 \cos\delta_j^1 \ldots \cos\delta_j^{2^n-2} e^{i\gamma_j^0}, \\ z_j^2 &= \sin\delta_j^1 \cos\delta_j^2 \ldots \cos\delta_j^{2^n-2} e^{i\gamma_j^1}, \\ &\ldots \\ z_j^{2^n-2} &= \sin\delta_j^{2^n-3} \cos\delta_j^{2^n-2} e^{i\gamma_j^{2^n-3}}, \\ z_j^{2^n-1} &= \sin\delta_j^{2^n-2} e^{i\gamma_j^{2^n-2}}, \end{aligned} \tag{3}$$

where

$$\delta_j^0 = \frac{\pi}{2} d_j^0,\ \delta_j^1 = \frac{\pi}{2} d_j^1,\ \ldots,\ \delta_j^{2^n-2} = \frac{\pi}{2} d_j^{2^n-2}, \\ \gamma_j^0 = 2\pi d_j^{2^n-1},\ \ldots,\ \gamma_j^{2^n-2} = 2\pi d_j^{2^{n+1}-3}$$

and d_j^i is the value of the ith feature in the jth data sample in the scaled data ($d_j^i \in [0,1)$).

Thus, it is possible to encode any training dataset into a quantum state using the following formula:

$$\left|q_j\right\rangle = \sum_{k=0}^{2^n-1} z_j^k(d)\left|k\right\rangle. \tag{4}$$

According to the postulates of quantum mechanics, if there are two systems with n_1 and n_2 qubits, respectively, then the states of the combined system have the following form:

$$\left|q\right\rangle = \sum_{k=0}^{2^{n_2}-1} \sum_{m=0}^{2^{n_1}-1} a^m b^k \left|m\right\rangle\left|k\right\rangle,$$

where a and b represent the state amplitudes of the first and the second system, respectively.

The state spaces for each system are $S^{2^{n_1}-2}$ and $S^{2^{n_2}-2}$. The set of states for the combined system is their direct product, i.e., $S^{2^{n_1}-2} \times S^{2^{n_2}-2}$. However, for topological reasons, this is definitely not $S^{2^{n_1+n_2}-2}$.

The part of the system of ($n_1 + n_2$) qubits, which cannot be expressed as a product of subsystem states, forms a set of so-called *entangled states*. The main property of an entangled state is that in order to remove the system from the entangled state, it is necessary to perform a unitary transformation that significantly affects all its subsystems. Entangled states form a basis in the space of states, and further it is called *the entangled basis*.

If the state of a multi-cubit system is entangled, it is impossible to get out of it without affecting each cubit. At the same time, each state of the system can be written in the entangled basis. Thus each state component in this basis affects all qubits significantly. If the amplitudes of these components are measured, it can be seen how subsystems interact in this quantum system. If the state labels are given, then which basis vector corresponds to the label of interest needs to be defined. This can be determined from the statistics of measurement results for a given label. Moreover, if new states of the same system are measured in the same way, it can be predicted with a certain probability that they belong to the labeled class, corresponding to the measurement result that is most relevant to the labeled samples.

To a certain extent, this means that classification can be performed without optimization if the dependencies are already present in the data. The latter is an important addition, since classifying data without dependencies (e.g., when the data is a complete superposition of pure states) would fail. Hence, the dependencies resulting in a given class for classification need to be defined. This is essentially a quantum property related to entanglement. Certainly, this could be interpreted without resorting to quantum representations, but then it would be necessary to consider the topological properties of the set of states of the system, as well as the subsets of its entangled states, in order to build probabilistic models on them. At present, the interpretation in terms of quantum calculations appears simpler.

RESULTS

Quick online classification

We shall consider the marked data set $\{d_j^i, l_j\}$, where d_j^i is the value of the ith feature in the jth sample,

Table 1. The experiment with the heart disease dataset. F1 is the harmonic mean of method precision and recall

Method	Precision, %	Recall, %	F1, %	Learning time, ms	Operation time, ms
Classification by emulated quantum entangled basis	76	65	70	20.5	9
Classification by linear discriminant analysis	100	58	73	7.6	4.9
Classification by logistic regression	68	65	66	5.2	3.1

while l_j is the value of the label (class) for the jth sample. We shall separate all data into training and test samples denoting them by

$$\{d^i_{jt}, l_{jt}\} \text{ and } \{d^i_{jc}, l_{jc}\},$$

respectively. We assume that the values for all features are scaled, while the labels take values from 0 to $L - 1$:

$$d^i_j \in [0,1),\ l_j \in \{0, ..., L-1\},$$

where $L = 2^l$.

The case of two qubits

We shall assume $n = 2$, $l = 1$, i.e., the number of features is 6 and the labels take on values 0 and 1. Then $i = 0, ..., 5$ and

$$\delta^0_j = \frac{\pi}{2} d^0_j,\ \delta^1_j = \frac{\pi}{2} d^1_j,\ \delta^2_j = \frac{\pi}{2} d^2_j,$$
$$\gamma^0_j = 2\pi d^3_j,\ \gamma^1_j = 2\pi d^4_j,\ \gamma^2_j = 2\pi d^5_j.$$

Data encoding into quantum states, according to (3) and (4), may be written as following:

$$z^0_j = \cos\delta^0_j \cos\delta^1_j \cos\delta^2_j,\ z^1_j = \sin\delta^0_j \cos\delta^1_j \cos\delta^2_j e^{i\gamma^0_j},$$
$$z^2_j = \sin\delta^1_j \cos\delta^2_j e^{i\gamma^1_j},\qquad z^3_j = \sin\delta^2_j e^{i\gamma^2_j}, \tag{5}$$

$$\left|q_j\right\rangle = \sum_{k=0}^{3} z^k_j \left|k\right\rangle. \tag{6}$$

We shall consider the following quantum circuit (Fig. 1). This is a well-known circuit for converting Bell states into vectors of computational basis. By using it, the probabilities of how the entangled basis vectors (Bell states) correspond to the vector given at the input is obtained.

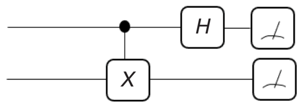

Fig. 1. Quantum circuit for converting Bell states into computational basis vectors. *H*—Hadamard gate, *X*—controlled X gate

The circuit shown in Fig. 1 can classify data containing six features. This is easily verified using the heart disease dataset taken from www.kaggle.com[6].

The original dataset contains 13 features and one label. The features are: age, sex, chest pain, pulse, cholesterol, and others. The label is the presence of heart disease. The dataset contains 303 data instances, of which 165 are labeled 1 while the rest are 0. The examples of analyzing this dataset by linear classifiers given on the website kaggle.com give values of the accuracy metric for predicting disease from 64% to 88%.

The following experiment is performed on this dataset. All feature values are translated into integer ranges from 0 to the feature-dependent limit value. The data is then reduced to six features without loss of information in the data, and parameterization in the form of quantum states of the two-particle system (5) is performed. The separation into training and test samples is done in a ratio of 65/35. The result obtained is compared with the result of the linear discriminant analysis (LDA) classifier[7]. The results are shown in Table 1.

The prepared dataset and program code are presented in the open-access archive[8]. The values shown in Table 1 are not high. However, they are obtained without optimization using the incoming data only, while the algorithm running time is spent mainly on emulation of quantum states and operations. Nevertheless, such classification procedure can work on almost any device in real time, since it does not require optimization and

[6] Akyildiz Ö. Heart disease data. https://www.kaggle.com/datasets/zgeakyldz/heart-desease-data. Accessed August 25, 2023.

[7] https://scikit-learn.ru/1-2-linear-and-quadratic-discriminant-analysis (in Russ.). Accessed January 09, 2023.

[8] Program codes and datasets for the paper are archived at https://disk.yandex.ru/d/JK4dsbdGLP_ZaQ. Accessed January 09, 2023.

can work in parallel streams. The above experiment is carried out on a computer with Intel Core I5 processor and 8GB RAM (Intel, USA) in a single thread.

It may be doubted whether the proposed method is ML (due to the lack of optimization). However, the definition of ML [36] states that ML algorithms build a model based on sample data known as training data, in order to produce predictions or decisions without being explicitly programmed to do so. The method under consideration uses the data and the model contained in it and allows making predictions based on this model, i.e., it fully satisfies the ML definition.

The quality of classification can be improved using known techniques (bagging[9] and boosting[10]). However, it can also be improved by using ML in the traditional sense as a parametric transformation with optimization. The way to use it in the quantum case is shown, for example, in [37]. In the case under consideration, in the circuit shown in Fig. 1, two controlled gates (U_0, U_1) are embedded into the first register (controlled by 0 and by 1), then a simple gate V in the second register, and the second register is measured. The measurement result is related to the label value and the output state set to a clean state. The optimization parameters are the components of the gates. The schematic is shown in Fig. 2.

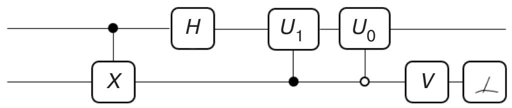

Fig. 2. The simplest quantum classifier with optimization

The two-particle state $|q\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle$ is applied to the input and is converted further as follows:

$$|q\rangle \underset{CX}{\rightarrow} a_0|00\rangle + a_1|01\rangle + a_2|11\rangle + a_3|10\rangle \underset{H}{\rightarrow} \frac{a_0+a_3}{\sqrt{2}}|00\rangle + \frac{a_1+a_2}{\sqrt{2}}|01\rangle + \frac{a_0-a_3}{\sqrt{2}}|10\rangle + \frac{a_1-a_2}{\sqrt{2}}|11\rangle \underset{CU_1,\,CU_0}{\rightarrow}$$

$$\underset{CU_1,\,CU_0}{\rightarrow} \frac{a_0+a_3}{\sqrt{2}}U_0|0\rangle|0\rangle + \frac{a_1+a_2}{\sqrt{2}}U_1|0\rangle|1\rangle + \frac{a_0-a_3}{\sqrt{2}}U_0|1\rangle|0\rangle + \frac{a_1-a_2}{\sqrt{2}}U_1|1\rangle|1\rangle \underset{V}{\rightarrow}$$

$$\underset{V}{\rightarrow} \frac{a_0+a_3}{\sqrt{2}}U_0|0\rangle V|0\rangle + \frac{a_1+a_2}{\sqrt{2}}U_1|0\rangle V|1\rangle + \frac{a_0-a_3}{\sqrt{2}}U_0|1\rangle V|0\rangle + \frac{a_1-a_2}{\sqrt{2}}U_1|1\rangle V|1\rangle.$$

[9] Bootstrap aggregating is a classification technique that uses compositions of algorithms each of which is trained independently. The result of classification is determined by voting.

[10] Boosting is a procedure of successive composition of machine learning algorithms, where each successive algorithm seeks to compensate for the compositional deficiencies of all previous algorithms.

If the parameters of gates U_0, U_1, and V are such that

$$U_0|0\rangle = \cos\alpha_0|0\rangle + \sin\alpha_0 e^{i\psi_0}|1\rangle,$$
$$U_0|1\rangle = -\sin\alpha_0 e^{i\phi_0}|0\rangle + \cos\alpha_0 e^{i(\phi_0+\psi_0)}|1\rangle,$$
$$U_1|0\rangle = \cos\alpha_1|0\rangle + \sin\alpha_1 e^{i\psi_1}|1\rangle,$$
$$U_1|1\rangle = -\sin\alpha_1 e^{i\phi_1}|0\rangle + \cos\alpha_1 e^{i(\phi_1+\psi_1)}|1\rangle,$$
$$V|0\rangle = \cos\beta|0\rangle + \sin\beta e^{iv}|1\rangle,$$
$$V|1\rangle = -\sin\beta e^{i\mu}|0\rangle + \cos\beta e^{i(\mu+v)}|1\rangle,$$

then the resulting state may be written in the following form:

$$A_{00}|00\rangle + A_{01}|01\rangle + A_{10}|10\rangle + A_{11}|11\rangle,$$

where

$$A_{00} = \frac{1}{\sqrt{2}}\Big((a_0+a_3)\cos\alpha_0 - (a_0-a_3)\sin\alpha_0 e^{i\phi_0}\Big)\cos\beta - \frac{1}{\sqrt{2}}\Big((a_1+a_2)\cos\alpha_1 - (a_1-a_2)\sin\alpha_1 e^{i\phi_1}\Big)\sin\beta e^{i\mu},$$

$$A_{01} = \frac{e^{iv}}{\sqrt{2}}\Big((a_0+a_3)\cos\alpha_0 - (a_0-a_3)\sin\alpha_0\Big)\sin\beta + \frac{e^{iv}}{\sqrt{2}}\Big((a_1+a_2)\cos\alpha_1 - (a_1-a_2)\sin\alpha_1 e^{i\phi_1}\Big)\cos\beta e^{i\mu},$$

$$A_{10} = \frac{1}{\sqrt{2}}\Big((a_0+a_3)\sin\alpha_0 e^{i\psi_0} + (a_0-a_3)\sin\alpha_0 e^{i\phi_0}\Big)\cos\beta - \frac{e^{i(\mu+\psi_1)}}{\sqrt{2}}\Big((a_1+a_2)\sin\alpha_1 + (a_1-a_2)\cos\alpha_1 e^{i\phi_1}\Big)\sin\beta,$$

$$A_{11} = \frac{e^{i(v+\psi_0)}}{\sqrt{2}}\Big((a_0+a_3)\sin\alpha_0 + (a_0-a_3)\cos\alpha_0 e^{i\phi_0}\Big)\sin\beta + \frac{e^{i(\mu+v+\psi_1)}}{\sqrt{2}}\Big((a_1+a_2)\sin\alpha_1 + (a_1-a_2)\cos\alpha_1 e^{i\phi_1}\Big)\cos\beta.$$

When labeled 0, this state should produce the result "0" with the highest probability when measured in the second register, which means the following:

$$A_{01} \rightarrow 0,\ A_{11} \rightarrow 0.$$

When labeled 1, on the contrary, it should be the following:

$$A_{00} \rightarrow 0,\ A_{10} \rightarrow 0.$$

The learning procedure is designed to find the best set of gate parameters (α_0, ϕ_0, ψ_0, α_1, ϕ_1, ψ_1, β, μ, and v) which provides the best aspirations. This is an optimization problem.

The classifier shown in Fig. 2 is the smallest possible classifier of this type. It is an analogue of an artificial neuron with two-dimensional input: it receives 2-cubic quantum signal, while outputting 1-cubic quantum signal and 1-bit classical signal (unlike classical neuron, where only one signal is output). The presence of quantum signal at the output allows the further use of quantum information, i.e., creating a network. Training and operation of such a classifier is illustrated in the archive[11]. As the experiment with the dataset on heart disease shows, optimization increases the classification quality indicators but only insignificantly. This means that in the case under consideration, quantum machine learning (QML) based on quantum entanglement can be used, without any optimization. However, this is not a general statement; this may not be the case for higher-dimensional classifiers.

Quantum classifier training

The quantum circuit shown in Fig. 2 contains three gates (U_0, U_1, and V) with parameters that can be adjusted. For example, gate V can be written in the following form:

$$V = \cos\beta|0\rangle\langle 0| - \sin\beta e^{i\tau}|0\rangle \times \\ \times \langle 1| + \sin\beta e^{i\theta}|1\rangle\langle 0| + \cos\beta e^{i(\tau+\theta)}|1\rangle\langle 1|,$$

and the specified parameters are β, θ, and τ. However, changing parameter τ results in the multiplication of the resulting state by the phase multiplier only, i.e., it does not result in a significant effect. This parameter would play a role in systems of higher dimensionality while in the case of two qubits, only two parameters, β and θ, are varied for optimizing vent V.

The same is true for gates U_0 and U_1, which can be written as follows:

$$U_\varepsilon = \cos\alpha_\varepsilon|0\rangle\langle 0| - \sin\alpha_\varepsilon|0\rangle\langle 1| + \sin\alpha_\varepsilon e^{i\rho_\varepsilon}|1\rangle \times \\ \times \langle 0| + \cos\alpha_\varepsilon e^{i\rho_\varepsilon}|1\rangle\langle 1|,\ \varepsilon = 0, 1,$$

where the varying parameters are α_0, α_1, ρ_0, and ρ_1. In total, there are six varying parameters for the two-particle quantum circuit.

We shall construct the likelihood function as the sum of the moduli of the following scalar products:

$$\left(|\hat{a}_j|^2, |\hat{b}_j|^2\right) \text{ and } (1 - l_j, l_j),$$

where $\hat{a}_j$ and $\hat{b}_j$ are the amplitudes of the quantum state leaving the second register on the jth package while l_j is the label of the incoming quantum state. The following are the calculations for the initial state (6). Before the controlled gates:

$$|q_j\rangle \to z_j^0|00\rangle + z_j^1|01\rangle + z_j^2|11\rangle + z_j^3|10\rangle \to \\ \to \frac{1}{\sqrt{2}}\left(z_j^0 + z_j^3\right)|00\rangle + \frac{1}{\sqrt{2}}\left(z_j^1 + z_j^2\right)|01\rangle + \\ + \frac{1}{\sqrt{2}}\left(z_j^0 - z_j^3\right)|10\rangle + \frac{1}{\sqrt{2}}\left(z_j^1 - z_j^2\right)|11\rangle$$

controlled valves U_1 and U_0 operate:

$$\frac{1}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\cos\alpha_0 - \left(z_j^0 - z_j^3\right)\sin\alpha_0\right)|00\rangle + \\ + \frac{1}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\cos\alpha_1 - \left(z_j^1 - z_j^2\right)\sin\alpha_1\right)|01\rangle + \\ + \frac{1}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\sin\alpha_0 e^{i\rho_0} + \left(z_j^0 - z_j^3\right)\cos\alpha_0 e^{i\rho_0}\right)|10\rangle + \\ + \frac{1}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\sin\alpha_1 e^{i\rho_1} + \left(z_j^1 - z_j^2\right)\cos\alpha_1 e^{i\rho_1}\right)|11\rangle.$$

Finally, valve V is active:

$$\frac{1}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\cos\alpha_0 - \left(z_j^0 - z_j^3\right)\sin\alpha_0\right) \times \\ \times \left(\cos\beta|00\rangle + \sin\beta e^{i\theta}|01\rangle\right) + \\ + \frac{1}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\sin\alpha_0 e^{i\rho_0} + \left(z_j^0 - z_j^3\right)\cos\alpha_0 e^{i\rho_0}\right) \times \\ \times \left(\cos\beta|10\rangle + \sin\beta e^{i\theta}|11\rangle\right) + \\ + \frac{1}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\cos\alpha_1 - \left(z_j^1 - z_j^2\right)\sin\alpha_1\right) \times \\ \times \left(-\sin\beta|00\rangle + \cos\beta e^{i\theta}|01\rangle\right) + \\ + \frac{1}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\sin\alpha_1 e^{i\rho_1} + \left(z_j^1 - z_j^2\right)\cos\alpha_1 e^{i\rho_1}\right) \times \\ \times \left(-\sin\beta|10\rangle + \cos\beta e^{i\theta}|11\rangle\right).$$

This is the state before the measurement and can be written in the following form:

$$B_{00}|00\rangle + B_{01}|01\rangle + B_{10}|10\rangle + B_{11}|11\rangle,$$

where

$$B_{00} = \frac{1}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\cos\alpha_0 - \left(z_j^0 - z_j^3\right)\sin\alpha_0\right)\cos\beta - \\ - \frac{1}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\cos\alpha_1 - \left(z_j^1 - z_j^2\right)\sin\alpha_1\right)\sin\beta,$$

[11] https://disk.yandex.ru/d/JK4dsbdGLP_ZaQ. Accessed January 09, 2023.

$$B_{01} = \frac{e^{i\theta}}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\cos\alpha_0 \sin\beta - \left(z_j^0 - z_j^3\right)\sin\alpha_0\right)\sin\beta + \\ + \frac{e^{i\theta}}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\cos\alpha_1 \cos\beta - \left(z_j^1 - z_j^2\right)\sin\alpha_1\right)\cos\beta,$$

$$B_{10} = \frac{e^{i\rho_0}}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\sin\alpha_0 + \left(z_j^0 - z_j^3\right)\cos\alpha_0\right)\cos\beta - \\ - \frac{e^{i\rho_1}}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\sin\alpha_1 + \left(z_j^1 - z_j^2\right)\cos\alpha_1\right)\sin\beta,$$

$$B_{11} = \frac{e^{i(\theta+\rho_0)}}{\sqrt{2}}\left(\left(z_j^0 + z_j^3\right)\sin\alpha_0 + \left(z_j^0 - z_j^3\right)\cos\alpha_0\right)\sin\beta + \\ + \frac{e^{i(\theta+\rho_1)}}{\sqrt{2}}\left(\left(z_j^1 + z_j^2\right)\sin\alpha_1 + \left(z_j^1 - z_j^2\right)\cos\alpha_1\right)\cos\beta.$$

The second register measurement would give 0 with probability $|\hat{a}_j|^2$ and 1 with probability $|\hat{b}_j|^2$, as follows:

$$|\hat{a}_j|^2 = |B_{00}|^2 + |B_{10}|^2,$$
$$|\hat{b}_j|^2 = |B_{01}|^2 + |B_{11}|^2.$$

It can be easily seen that the only remaining phase parameter is $\rho \equiv \rho_1 - \rho_0$, and the likelihood function has the following form:

$$\Phi(\alpha_0, \alpha_0, \beta, \rho) = \sum_j |\hat{a}_j|^2 (1 - l_j) + \sum_j |\hat{b}_j|^2 l_j.$$

It should be maximized so that state $|1\rangle$ is expected in the second register, if $l_j = 1$. This is done using the COBYLA[12] method in the program code available in the archive[13].

The calculation of the likelihood function is given in order for it be used for calculations on a classical computer. In the case of a quantum computer, the likelihood function manually does not need to be calculated manually, since values $|\hat{a}_j|^2$ and $|\hat{b}_j|^2$ would be available as measurement results.

Arbitrary number of qubits

We shall generalize the constructed classifier to the case of an arbitrary number of particles starting with the version without optimization. For this, it is necessary to construct a multi-particle entangled basis. In the case of two particles, the entanglement appears in the form of a combination of the computational basis vector and its inverted vector. Such combinations for the case of three particles may be written as follows:

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle),\ \frac{1}{\sqrt{2}}(|001\rangle + |110\rangle),$$
$$\frac{1}{\sqrt{2}}(|010\rangle + |101\rangle),\ \frac{1}{\sqrt{2}}(|011\rangle + |100\rangle),$$
$$\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle),\ \frac{1}{\sqrt{2}}(|001\rangle - |110\rangle),$$
$$\frac{1}{\sqrt{2}}(|010\rangle - |101\rangle),\ \frac{1}{\sqrt{2}}(|011\rangle - |100\rangle).$$

It can be seen easily that none of these states is the result of the tensor product of three one-particle or any two-particle and one-particle states. Thus, all of them are entangled states and form the basis which is also easy to check. This is the entangled basis for the 3-particle system. Similarly, the entangled basis for an arbitrary n-particle quantum system may be constructed, as follows:

$$\frac{1}{\sqrt{2}}(|0x\rangle + |1\bar{x}\rangle),\ \frac{1}{\sqrt{2}}(|0x\rangle - |1\bar{x}\rangle),$$

where x is a binary notation of a number from 0 to $2^{n-1} - 1$ while the superscript denotes inversion.

The quantum circuit of the multi-particle classifier without optimization is shown in Fig. 3.

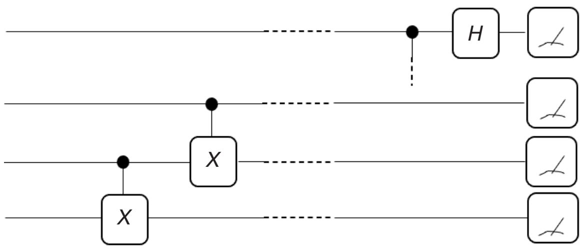

Fig. 3. Classifier without optimization

The generalization of the circuit shown in Fig. 2 is now obvious. It would be enough to set the controlled gates ($2n - 2$ items, two for each control register) to the first register and one gate to the registers starting from the second. This is shown in Fig. 4.

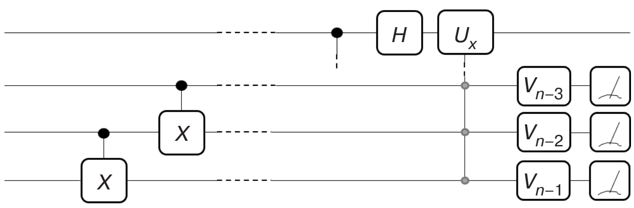

Fig. 4. Optimizable quantum classifier

[12] Constrained optimization by linear approximation (COBYLA).

[13] https://disk.yandex.ru/d/JK4dsbdGLP_ZaQ. Accessed January 09, 2023.

The gray dots in Fig. 4 indicate the 0 or 1 control on one of the registers. The circuit shown in Fig. 4 is a quantum neuron with an arbitrary number of inputs. The circuit shown in Fig. 3 can be viewed in the same way, if the first register is not measured since it would then contain a state which is a superposition of two vectors of the entangled basis. The vectors are determined by the measurement result. Learning without optimization is done, as before, on statistics and on the assumption that there are regularities in the data.

Anomaly detection on streaming data

When data is streaming, the last J incoming packages $d^i_j,\ j=\overline{0,J-1}$ may be taken into account. This data is not marked up a priori. The task is to detect an anomaly in the stream, i.e., a situation where an incoming series of data packages is significantly different from those in the stream before. The difference may not only be in the packages themselves but also in the order they are received.

The classifiers proposed above can be used to detect anomalies. The selected J packages are considered as basic for statistics; a series of measurements from 0 and 1 are obtained from them. For example, in the 2-particle case there are four possible measurement results (if the quantum state output is not considered). The frequencies of these measurement results form a vector of the system current state. If this vector remains approximately constant or takes values only in certain clusters whenever it is computed on J packages, then any new input from J packages on which the mentioned vector does not fall into any of the clusters would be an anomaly.

Two questions naturally arise. The first relates to the situation when there is a large number of features in streaming data, for example, in the case of a video stream. According to the consideration above, if there are F features, then $\log\left(F/2+1\right)$ qubits and $F/2+1$ possible states are required. This can be a quitelarge number. Therefore, a threshold on the frequency of the feature occurrences should be set, and this is a configurable parameter. In the experiment below with the student' dataset, the threshold frequency is set to 10%. If a feature appears in a given state in less than 10% of cases, then it is considered not to appear in that state at all. As this threshold increases, the number of data instances not classified by the network increases. When the threshold is decreased, the metrics deteriorates.

Another question concerns the J value. There are some recommendations for it: J should be less than the number of packages appearing in the maximum decision time and, at the same time, J should be relatively large in order for the statistics to be rich enough, i.e., for training to give effective prediction. If the anomaly is not detected, then the model is undertrained and J needs to be increased. Also if there are a lot of false positives, then the retraining takes place while J needs to be decreased.

An example code of the anomaly detector based on the classifier built on a tangled basis is given in the archive[14].

Artificial neural network from neuron quantum analogs

The quantum neuron (q-neuron) shown in Fig. 4 or 3 can be used to build a quantum neural network (QNN). In general terms, such a network, receiving I quantum states (qubits) at the input, returns $Q < I$ quantum states together with $C = I - Q$ classical bits of information. The QNN can be trained on the basis of the classical information received. Training procedure in the case of q-neurons with optimization coincides with that of the classical case. The parameters of quantum gates act as weights.

Copying of quantum states is not allowed. Therefore, QNNs do not contain branching. However, q-neuron can contain more than one quantum output that can be used for creating networks of different architecture.

Let there be Q_1 q-neurons in the first layer. Clearly, that $Q_1 \le \frac{I}{2}$ since any q-neuron should have at least two qubits at the input. Accordingly, there would be $C_1 \le I - Q_1$ classical bits of information at the layer output. The next layer may have Q_2 q-neurons, with $Q_2 \le \frac{I - C_1}{2}$. This is shown schematically in Fig. 5, where all q-neurons have one quantum output each while $I = 20$, $Q_1 = 8$, and $Q_2 = 3$. If no second layer is added, the network shown in Fig. 5 would produce 8 qubits and 12 bits at the output. If one more q-neuron is added after the second layer (as a third layer), there would be 1 qubit and 19 bits at the output.

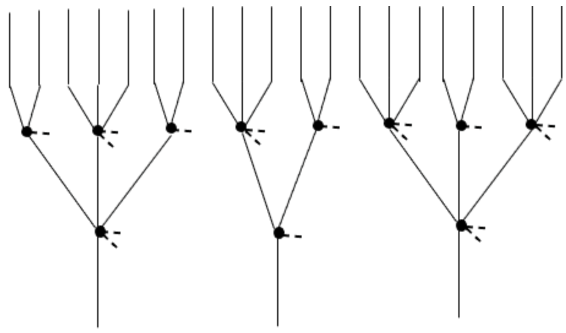

Fig. 5. QNN on 11 q-neurons (black dots): 20 qubits (lines) at the input, 3 qubits at the output, and 17 bits (dashed lines) of classical information

[14] https://disk.yandex.ru/d/JK4dsbdGLP_ZaQ. Accessed January 09, 2023.

The QNN shown in Fig. 5 is similar to the fractal neural network discussed in [38] in architecture. Although classical neural networks with ordinary neurons are considered in [38], the properties noted there such as high learnability and the ability to work with high dimensionality of input vectors remain the same for QNN.

Quantum neural network training

Training a single q-neuron has already been discussed above. We shall generalize this procedure to training the network. For simplicity, only the case of two layers is focused on. We shall take for example the network shown in Fig. 5 in the center containing five qubits at the input, two q-neurons and three bits of classical information on the first layer, and one q-neuron and one bit of classical information on the second layer.

By performing a forward pass on the training sample, the statistics of matching labels and sets of classical bits is obtained. We shall match them with each other. Let the labels take only two possible values in this case. One q-neuron output allows 2 classical bits to be attained when both registers are measured, since there is no task of quantum state propagation further and so it can be measured. Therefore, initially, the most frequent outputs (it could be either 00, 01, 10, or 11) can be allocated to label 0 approximately in proportion to the share of this label in the training sample, while the rest allocated to label 1. Let label ε correspond to e_0, e_1, e_2, and e_3 of all n_0, n_1, n_2, and n_3 outputs of species 00, 01, 10, and 11, respectively. This label has a fraction f_ε in the training sample. We shall select ε with the highest fraction, sort $\frac{e_0}{n_0}, \frac{e_1}{n_1}, \frac{e_2}{n_2}, \frac{e_3}{n_3}$ in descending order and choose the sum of the first elements of the resulting list, with the closest to f_ε fraction of the sum of the corresponding n_i in the total number of packages. The corresponding outputs are allocated to label ε.

Example. A sample of 5,000 batches is separated by labels of 4000 (0) and 1000 (1). For label 0 at the network output, $e_0 = 1500$, $e_1 = 500$, $e_2 = 1400$, and $e_3 = 600$ is obtained while total numbers of outputs are $n_0 = 1600$, $n_1 = 700$, $n_2 = 2000$, and $n_3 = 700$. We have:

$$\frac{e_0}{n_0}, \frac{e_1}{n_1}, \frac{e_2}{n_2}, \frac{e_3}{n_3} = \frac{1500}{1600}, \frac{500}{700}, \frac{1400}{2000}, \frac{600}{700}.$$

The order of n_i is as follows: n_0, n_3, n_1, and n_2. This label has a fraction of 0.8, and the closest matching sum is $n_0 + n_3 + n_1 = 3000$ having a fraction of 0.6. Hence, outputs 00, 01, and 11 should be allocated to label 0, while output 10 to label 1. Although, of course, with so many batches in the sample, it would be better to increase the number of features and, consequently, the number of qubits at the input (the quantum network cannot expand due to the prohibition of copying).

Unlike a conventional neural network and a QNN with optimization, QNN without optimization can be trained in the forward direction instead of backward, i.e., simultaneously with its operation. This can be performed according to the scenario described above, since the same logic may be used to calculate statistics for the first layer: separating outputs by labels and keeping on doing so further in the next layers. The classification quality would improve anyway from layer to layer.

DISCUSSION

Many works are devoted to QMLs and QNNs. In the Russian-language literature, work [1] as already mentioned, in particular, points out the importance of “searching for a QNN model which is optimal in terms of utilizing all the advantages presented by quantum computing and neural networks, as well as ML algorithms”. One of the most recent reviews in these areas [39], contains the following text in the Outlook section: “The first quantum advantages in QML will likely arise from hidden parameter extraction from quantum data. This can be for quantum sensing or quantum state classification/regression. Fundamentally, we know from the theory of optimal measurement that non-local quantum measurements can extract hidden parameters using fewer samples. Using QML, one can form and search over a parameterization of hypotheses for such measurements.” This paper presents one possible path for classifying quantum states.

The QML considered in the paper is of QC or QQ class, i.e., it uses quantum data on classical (emulating) or quantum devices. The proposed q-neuron is ideologically close to known concepts (described in [6] and [39], among others). However, at the same time it has a new essential feature that is exploiting quantum entanglement. In particular, q-neuron can operate without learning in the usual sense; optimization and error back propagation are not needed.

This can be exemplified by the experiment with the analysis of the dataset of student states[15]. For this experiment, the dataset is used without preprocessing. The entire preprocessing is captured in the analysis script available in the archive[16]. The dataset is a labeled one with 34 feature columns and 4424 data instances. The binary label used in the experiment is Dropout = 1, with other values (Enrolled, Graduate) = 0. The task

[15] https://www.kaggle.com/datasets/thedevastator/higher-education-predictors-of-student-retention. Accessed January 09, 2023.

[16] https://disk.yandex.ru/d/JK4dsbdGLP_ZaQ. Accessed January 09, 2023.

Table 2. Experiment with students' dataset

Method	Precision, %	Recall, %	F1, %	Unclassified states, %	Learning time, ms	Operation time, ms
Classification by emulated quantum entangled basis	39	100	57	40	2968	1901
Classification by linear discriminant analysis	100	29	45	0	138	23

is to predict dropout. The features are mapped to integer intervals, the feature space is transformed to dimension 62 without loss of information. This is done for the use of QNN of two q-neurons in the first layer and one q-neuron in the second layer. Training is carried out without optimization. The network configuration implies the first q-neuron with three inputs and the second one with two inputs, in the first layer. The second layer contains a single q-neuron with two inputs. Some of the data instances could not be classified by the neural network, since they are approximately equally close to both 1 and 0 label values. The results can be compared with the classical LDA algorithm taken from the scikit-learn package of the Python language. The results are shown in Table 2.

It can be seen from the experiment that QNN does not ascribe label 0 to the student if he/she drops out: false negatives are equal to 0. At the same time, the share of false positives is quite high: more than half of the dropout labels (1) are false. The QNN considered 40% of the data unsuitable for classification. Certainly, these results are quite different from those of the LDA classifier, and it is unclear in which direction (better or worse). Nevertheless, it can be seen that LDA training time exceeds the running time by 6 times, while QNN has only 1.5 times.

A rigorous examination of the QNN performance of different architectures is planned in the following studies.

CONCLUSIONS

A new direction of QML development applying quantum entanglement significantly is proposed. It allows for the building of intelligent systems working on streaming data and learning online, taking into account changes in the data environment but not reduced to reinforcement learning. The proposed learning method could be called "reinforcement learning in reverse". In reinforcement learning, the agent calculates the classification quality while the environment remains an external factor. However, in the proposed approach, the environment is the carrier of classification patterns and they are recovered directly from it.

Such systems can be used in control systems of unmanned vehicles of any kind, as well as in security systems and intelligent business-assistants. In this case, the use of quantum computers is not mandatory.

Acknowledgments

This study is supported partially by the Russian Foundation for Basic Research (project No. 19-29-09056).

REFERENCES

1. Sigov A.S., Andrianova E.G., Zhukov D.O., Zykov S.V., Tarasov I.E. Quantum informatics: overview of the main achievements. *Russian Technological Journal*. 2019;7(1):5–37 (in Russ.). https://doi.org/10.32362/2500-316X-2019-7-1-5-37
2. Zeguendry A., Jarir Z., Quafafou M. Quantum Machine Learning: A Review and Case Studies. *Entropy*. 2023;25(2):287. https://doi.org/10.3390/e25020287
3. Tacchino F., Macchiavello C., Gerace D., et al. An artificial neuron implemented on an actual quantum processor. *npj Quantum Inf*. 2019;5(1):26. https://doi.org/10.1038/s41534-019-0140-4
4. DiVincenzo D.P. The Physical Implementation of Quantum Computation. *Fortschritte der Physik*. 2000;48(9–11);771–783. https://doi.org/10.1002/1521-3978(200009)48:9/11%3C771::AID-PROP771%3E3.0.CO;2-E
5. Menneer T., Narayanan A. Quantum-inspired neural networks. In: *Proceedings of the Neural Information Processing Systems 95*. Denver, CO, USA, 27–30 November 1995. Available from URL: https://www.researchgate.net/publication/2267350_Quantum-inspired_Neural_Networks
6. Gushanskiy S.M., Buglov V.E. Quantum deep learning of convolutional neural network using variational quantum circuit. *Izvestiya YuFU. Tekhnicheskie nauki = Izvestiya SFedU. Engineering Sciences*. 2021;7(224):167–174 (in Russ.).
7. Cong I., Choi S., Lukin M.D. Quantum convolutional neural networks. *Nat. Phys*. 2019;15(12):1273–1278. https://doi.org/10.1038/s41567-019-0648-8
8. Kerenidis I., Landman J., Prakash A. *Quantum algorithms for deep convolutional neural networks*. arXiv:1911.01117. https://doi.org/10.48550/arXiv.1911.01117
9. Henderson M., Shakya S., Pradhan S., Cook T. Quanvolutional neural networks: Powering image recognition with quantum circuits. *Quantum Mach. Intell*. 2020;2(1):2. https://doi.org/10.1007/s42484-020-00012-y
10. Rebentrost P., Mohseni M., Lloyd S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett*. 2014;113(13):130503. https://doi.org/10.1103/physrevlett.113.130503
11. Harrow A.W., Hassidim A., Lloyd S. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett*. 2009;103(15):150502. https://doi.org/10.1103/physrevlett.103.150502
12. Dang Y., Jiang N., Hu H., Ji Z., Zhang W. Image classification based on quantum K-Nearest-Neighbor algorithm. *Quantum Inf. Process*. 2018;17(9):239. https://doi.org/10.1007/s11128-018-2004-9
13. Schuld M., Sinayskiy I., Petruccione F. Prediction by linear regression on a quantum computer. *Phys. Rev. A*. 2016;94(2):022342. https://doi.org/10.1103/PhysRevA.94.022342
14. Lu S., Braunstein S.L. Quantum decision tree classifier. *Quantum Inf. Process*. 2014;13(3):757–770. https://doi.org/10.1007/s11128-013-0687-5
15. Lloyd S., Mohseni M., Rebentrost P. *Quantum algorithms for supervised and unsupervised machine learning*. arXiv:1307.0411. https://doi.org/10.48550/arXiv.1307.0411
16. Lloyd S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 1982;28(2):129–137. https://doi.org/10.1109/TIT.1982.1056489
17. Kerenidis I., Landman J., Luongo A., Prakash A. *q-means: A quantum algorithm for unsupervised machine learning*. arXiv: 1812.03584. https://doi.org/10.48550/arXiv.1812.03584
18. Aïmeur E., Brassard G., Gambs S. Quantum speed-up for unsupervised learning. *Mach. Learn.* 2013;90(2):261–287. https://doi.org/10.1007/s10994-012-5316-5
19. Lloyd S., Mohseni M., Rebentrost P. Quantum principal component analysis. *Nat. Phys.* 2014;10(9):631–633. https://doi.org/10.1038/nphys3029
20. Dong D., Chen C., Li H., Tarn T.-J. Quantum Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part B* (*Cybern.*) 2008;38(5):1207–1220. https://doi.org/10.1109/TSMCB.2008.925743
21. Lloyd S., Weedbrook C. Quantum Generative Adversarial Learning. *Phys. Rev. Lett*. 2018;121(4):040502. https://doi.org/10.1103/PhysRevLett.121.040502
22. Dallaire-Demers P.-L., Killoran N. Quantum generative adversarial networks. *Phys. Rev. A.* 2018;98(1):012324. https://doi.org/10.1103/PhysRevA.98.012324
23. Situ H., He Z., Wang Y., Li L., Zheng S. Quantum generative adversarial network for generating discrete distribution. *Inf. Sci*. 2020;538:193–208. https://doi.org/10.1016/j.ins.2020.05.127
24. Huang H.-L., Du Y., Gong M., Zhao Y., et al. Experimental Quantum Generative Adversarial Networks for Image Generation. *Phys. Rev. Appl*. 2021;16(2):024051. https://doi.org/10.1103/PhysRevApplied.16.024051
25. Chakrabarti S., Yiming H., Li T., Feizi S., Wu X. *Quantum Wasserstein generative adversarial networks*. arXiv:1911.00111. https://doi.org/10.48550/arXiv.1911.00111
26. Kieferová M., Wiebe N. Tomography and generative training with quantum Boltzmann machines. *Phys. Rev. A.* 2017;96(6):062327. https://doi.org/10.1103/PhysRevA.96.062327
27. Amin M.H., Andriyash E., Rolfe J., Kulchytskyy B., Melko R. Quantum Boltzmann Machine. *Phys. Rev. X.* 2018;892:021050. https://doi.org/10.1103/PhysRevX.8.021050
28. Romero J., Olson J.P., Aspuru-Guzik A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol*. 2017;2(4):045001. https://doi.org/10.1088/2058-9565/aa8072
29. Khoshaman A., Vinci W., Denis B., Andriyash E., Sadeghi H., Amin M.H. Quantum variational autoencoder. *Quantum Sci. Technol*. 2018;4(1):014001. https://doi.org/10.1088/2058-9565/aada1f
30. Jiang Z., Rieffel E.G., Wang Z. Near-optimal quantum circuit for Grover's unstructured search using a transverse field. *Phys. Rev. A.* 2017;95(6):062317. https://doi.org/10.1103/PhysRevA.95.062317
31. Farhi E., Goldstone J., Gutmann S. *A quantum approximate optimization algorithm*. arXiv:1411.4028. https://doi.org/10.48550/arXiv.1411.4028
32. Kerenidis I., Prakash A. Quantum gradient descent for linear systems and least squares. *Phys. Rev. A*. 2020;101(2):022316. https://doi.org/10.1103/PhysRevA.101.022316

33. Simões R.D.M., Huber P., Meier N., Smailov N., Füchslin R.M., Stockinger K. Experimental Evaluation of Quantum Machine Learning Algorithms. *IEEE Access*. 2023;11:6197–6208. https://doi.org/10.1109/ACCESS.2023.3236409
34. Shao Yu., Witarsyah D. Fast recognition method of moving video images based on BP neural networks. *Open Physics*. 2018;16(1):1024–1032. https://doi.org/10.1515/phys-2018-0123
35. Nielsen M.A., Chuang I.L. *Quantum Computation and Quantum Information*. 2nd ed. Cambridge, UK: Cambridge University Press; 2010. 866 p. ISBN 978-1-107-00217-3
36. Samuel A.L. Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Develop*. 1959;44(1–2): 207–226. (Reprinted from *IBM J. Res. Develop*. 1959;3(3):210–229. https://doi.org/10.1147/rd.33.0210)
37. Maheshwari D., Sierra-Sosa D., Garcia-Zapirain B., Variational Quantum Classifier for Binary Classification: Real vs Synthetic Dataset. *IEEE Access*. 2022;10: 3705–3715. https://doi.org/10.1109/ACCESS.2021.3139323
38. Zuev S., Kabalyants P., Polyakov V., Chernikov S. Fractal Neural Networks. In: *2021 International Conference on Electrical, Computer and Energy Technologies* (*ICECET*). 2021. https://doi.org/10.1109/ICECET52533.2021.9698649
39. Cerezo M., Verdon G., Huang H.Y., et al. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci*. 2022;2(9):567–576. https://doi.org/10.1038/s43588-022-00311-3

About the author

Sergei V. Zuev, Cand. Sci. (Phys.-Math.), Associate Professor, Department of Computer Engineering and Modelling, Institute of Physics and Technologies, V.I. Vernadsky Crimean Federal University (4, Vernadskogo pr., Simferopol, 295007 Republic of Crimea, Russia). E-mail: zuevsv@cfuv.ru. Scopus Author ID 57292501000, ResearcherID U-1055-2017, https://orcid.org/0000-0003-4237-0491

Об авторе

Зуев Сергей Валентинович, к.ф.-м.н., доцент, кафедра компьютерной инженерии и моделирования Физико-технического института ФГАОУ ВО «Крымский федеральный университет им. В.И. Вернадского» (295007, Россия, Республика Крым, Симферополь, пр-т Вернадского, д. 4). E-mail: zuevsv@cfuv.ru. Scopus Author ID 57292501000, ResearcherID U-1055-2017, https://orcid.org/0000-0003-4237-0491

Translated from Russian into English by Kirill V. Nazarov
Edited for English language and spelling by Dr. David Mossop