Int. J. Appl. Math. Comput. Sci., 2020, Vol. 30, No. 4, 733–744
DOI: 10.34768/amcs-2020-0054

BASIC QUANTUM CIRCUITS FOR CLASSIFICATION AND APPROXIMATION TASKS

JOANNA WIŚNIEWSKA [a,*], MAREK SAWERWAIN [b], ANDRZEJ OBUCHOWICZ [b]

[a]Institute of Information Systems
Military University of Technology
ul. Gen. S. Kaliskiego 2, 00-908 Warsaw, Poland
e-mail: jwisniewska@wat.edu.pl

[b]Institute of Control and Computation Engineering
University of Zielona Góra
ul. Szafrana 2, 65-516 Zielona Góra, Poland
e-mail: {M.Sawerwain,A.Obuchowicz}@issi.uz.zgora.pl

We discuss a quantum circuit construction designed for classification. The circuit is built of regularly placed elementary quantum gates, which implies the simplicity of the presented solution. The realization of the classification task is possible after the procedure of supervised learning which constitutes parameter optimization of Pauli gates. The process of learning can be performed by a physical quantum machine but also by simulation of quantum computation on a classical computer. The parameters of Pauli gates are selected by calculating changes in the gradient for different sets of these parameters. The proposed solution was successfully tested in binary classification and estimation of basic non-linear function values, e.g., the sine, the cosine, and the tangent. In both the cases, the circuit construction uses one or more identical unitary operations, and contains only two qubits and three quantum gates. This simplicity is a great advantage because it enables the practical implementation on quantum machines easily accessible in the nearest future.

Keywords: quantum circuits, data classification, supervised learning, qubits, qudits.

1. Introduction

Machine learning as a field of the computer science is very popular. The idea of artificial intelligence is exhilarating since the 1940s when the first model of an artificial neuron (the so-called McCulloch–Pitts neuron) was presented. Artificial neural networks and other tools of machine learning still evolve, and play a more and more important role in data processing. Today, the whole world is on the verge of a quantum revolution where data are going to be encoded as quantum states and processed with the use of laws of quantum mechanics (Nielsen and Chuang, 2010; Kołaczek *et al*., 2019) and machine learning methods are also developed for quantum computational systems. Researchers deal with the different kinds of quantum machine learning (Biamonte *et al.*, 2017; Schuld *et al.*, 2014; 2015), e.g., quantum neural networks (Narayanan and Menneer, 2000; Zoufal *et al*., 2019), quantum kNN methods (Wiebe *et al*., 2015), quantum self-organized maps (Weigang, 1998), or the quantum k-means method (Veenman and Reinders, 2005).

In this work, we would like to propose a new solution from the group of quantum machine learning methods: a classifying quantum circuit (the article is an extended version of an earlier conference paper by Wiśniewska and Sawerwain (2020)). The circuit differs from the existing approaches because input data are in a form of quantum states (classical observations as input data were suggested by Pérez-Salinas *et al*. (2020)), and only two kinds of quantum gates are used (much more complicated circuits were proposed by Mitarai *et al*. (2018)). These gates realize operations of rotations and introduce entanglement. The angles of rotations are calculated with optimization methods (Li *et al*., 2017).

The article is organized as follows. In Section 2 basic definitions referring to quantum computing are presented. They cover a fundamental part of quantum mechanics

*Corresponding author

needed to present the content connected with the quantum approach to machine learning. A broad introduction to quantum computing can be found in the monograph by MacMahon (2007).

Section 3 contains information concerning the conversion of classical data to quantum states, construction of the circuit, and the learning algorithm. In Section 4, we present results of numerical experiments for classification tasks realized as simulation of quantum computation. Section 5 includes a summary and conclusions.

2. Brief introduction to quantum information processing

We utilize standard denotation, e.g., $\mathbb{N}$ for natural numbers. For convenience and clarity, all the symbols used are gathered in Table 1.

An equivalent of the classical bit in quantum computing is the so-called qubit. According to the laws of quantum mechanics, a description of a qubit is a description of a quantum state. While quantum states may be expressed as elements of a Hilbert space, i.e., vectors whose entries are complex numbers $\mathbb{C}$, a single qubit's state may be presented as a vector

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}, \tag{1}$$

under the normalization condition $|\alpha_0|^2 + |\alpha_1|^2 = 1$, where $\alpha_0, \alpha_1 \in \mathbb{C}$, and $|\psi\rangle$ denotes a column vector in the so-called Dirac notation.

A qubit may be broaden to a qudit, i.e., to a general quantum information unit. Let d stand for the freedom level of a qudit. The state of a single qudit is a normalized d-element vector of complex numbers.

In classical computing, the bits are usually denoted as one or zero. Of course, two arbitrary separate states are sufficient to realize a calculation with the use of Boolean algebra. Naturally, the logic invented by George Boole can be extended to multivalued logic (Augusto, 2017). A similar approach is utilized in quantum computing, where we employ the so-called computational basis. A basis for a d-level qudit contains d orthonormal d-dimensional vectors. A basis which is the most used in the area of quantum computational methods is termed the standard basis. The vectors of this basis are constructed in the following way: $d-1$ elements are zeros and one element equals one (this element occupies a different position in each basis vector). For example, if we deal with qutrits (qudits with $d = 3$), then

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad |2\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{2}$$

Table 1. Symbols and notation used in the paper.

Notation	Description
$\mathbb{N}$, $\mathbb{R}$, $\mathbb{C}$	integer, real, and complex numbers
$\mathbb{U}$	the set of unitary operators
$\langle M \rangle$	the expectation of operator M on quantum state $\|\psi\rangle$: $\langle M \rangle = \langle\psi\|M\|\psi\rangle$
i, j, m, k, l, r	indexes, i.e., integer values used to enumerate operators, probes, attributes; always used locally in the context of a given notion or equation
G, Γ_k	the generalized set of Pauli operators of dimensionality d (including traditional X, Y, Z operators for $d = 2$), together with an identity operator, is denoted by G, and Γ_k refers the k-th operator constructed from the $SU(d)$ group which is more general than the generalized set of Pauli operators
R_{Γ_k}	rotation operator, where Γ_k represents one of the generalized Pauli operators
$\Theta^{k,j}$, $\beta^{k,j}$, $\eta^{r,r}$	examples of $SU(d)$ operators
ξ_d^l	the l-th root of unity of degree d
CNOT	the controlled negation gate
U_i	one of unitary operators used to build the classification circuit
ζ	the total number of U operators
θ, θ_i	all θ parameters and one or set of parameters for a given operator U_i
H_c, D_c	the height and depth (number of layers) of the classifying circuit
$\|\psi\rangle$	quantum state of one qubit or qudit
$\|\psi_d\rangle$, $\|\psi_c\rangle$	a general description of qubits or qudits containing: data sample (also called an observation), class label
$\|\psi_d^{x_i}\rangle$, $\|\psi_c^{x_i}\rangle$	quantum states which describe a given data sample x_i ($\|\psi_d^{x_i}\rangle$), and a class label related to sample x_i ($\|\psi_c^{x_i}\rangle$)
X	a data set
x_i	the i-th value of from the data set
$\mathbb{L}$	loss function
d	the number of attributes for a given observation, and also the freedom level of a qudit

All basis vectors satisfy the normalization condition, and are orthogonal one to another.

Aparat from the normalized vectors, the principle of superposition is often utilized to express the quantum

state. This notion describes the state in relation to the chosen computational basis. For example, the state of a d-level qudit in the standard basis is

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{d-1}|d-1\rangle, \quad (3)$$

where $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$, and $\alpha_i \in \mathbb{C}$. Coefficients α_i are called the probability amplitudes or just the amplitudes.

A single qubit or a qudit are not very useful for any calculation. Let n be the number of units in the so-called quantum register. The state of a register may be expressed as a tensor product of vectors describing consecutive qudits. Of course, the same rule concerns the states written as a superposition. The state of two qubits is

$$\begin{aligned} |\Psi\rangle &= |\psi\rangle \otimes |\phi\rangle \\ &= (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle \\ &\quad + \alpha_1\beta_1|11\rangle, \end{aligned} \quad (4)$$

where products $\alpha_i\beta_j$ are amplitudes of the state and satisfy the normalization condition.

Remark 1. (*Quantum entangled states*) It should be emphasized that some quantum states cannot be presented directly as a tensor products of other states. Such states are called entangled states.

Apart from vectors and superposition, quantum states may be denoted as density matrices. The discussion in this work is based on pure states where the density matrix ρ is calculated as the outer product of state vectors,

$$\rho = |\Psi\rangle\langle\Psi|, \quad (5)$$

where $\langle\Psi|$ is the Hermitian adjoint of $|\Psi\rangle$ which represents an arbitrary pure quantum state (also termed the vector state). The quantum states described by density matrices are used in the description of the learning algorithm in Section 3.3.

If we want to perform computation on an input quantum state, we have to transform it. This operation may have a unitary or a non-unitary character. The unitary transformations are reversible. We can denote the action performed on a quantum state $|\psi_{\text{in}}\rangle$ (in order to obtain $|\psi_{\text{out}}\rangle$) with the use of a unitary operator U as

$$U|\psi_{\text{in}}\rangle = |\psi_{\text{out}}\rangle, \quad (6)$$

and the operator is unitary if $U^\dagger = U^{-1}$ (where $\dagger$ stands for the Hermitian adjoint operation, and -1 signifies the inversion operation). Operator U may be expressed as a matrix. If we deal with an n-qudit state, the size of the matrix is $d^n \times d^n$ (so the vector $|\psi_{\text{out}}\rangle$ may be computed by the multiplication of matrix U by the vector $|\psi_{\text{in}}\rangle$).

Quantum gates are unitary operators. In this work, we utilize the negation gate, the gate realizing qubit's rotation through π radians around the z-axis, and the Hadamard gate for qubits, and their generalizations for qudits.

The state of a single qubit can be visualized on the three-dimensional (x, y, z) Bloch sphere where the angle between orthonormal vectors is π. The Pauli operators realize basic π-radian rotations around the x-axis, y-axis and z-axis, respectively,

$$\begin{aligned} X &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ Y &= \begin{bmatrix} 0 & -\mathbb{i} \\ \mathbb{i} & 0 \end{bmatrix}, \\ Z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned} \quad (7)$$

Let us denote by $\Gamma = \{I, X, Y, Z\}$ the set of Pauli operators together with the identity operator I (identity matrix).

The gate X is termed the negation gate. In a two-qubit system, we can introduce the controlled negation gate:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (8)$$

The qubit gates can be also directly generalized to their qudit versions. Denote by $|l\rangle$ one of d vectors from the standard basis. The single qudit negation gate X performs the operation

$$X|l\rangle = |(l+1) \mod d\rangle. \quad (9)$$

The rotation gate Z changes the qudit's phase:

$$Z|l\rangle = \exp((2\pi\mathbb{i}l)/d)|l\rangle = \xi_d^l|k\rangle, \quad (10)$$

where $\xi_d^l = \exp((2\pi\mathbb{i}l)/d)$ are the roots of unity. The operators X and Z are elements of the so-called generalized Pauli group where they are denoted as $G^{j,l}$ (an additional index $j = 0, \ldots, d-1$):

$$G^{j,l} = Z^j X^l = \xi_d^{jl} X^l Z^j, \quad (11)$$

where $X^d = Z^d = I$, and $I_{d\times d}$ represents the identity operator. However, it should be mentioned that the generalized operators from the Pauli group can be fully reconstructed with the $SU(d)$ operators which are given in the following paragraph.

The construction of the classifying circuit, presented in this work, needs gates realizing rotations by any angle $\vartheta \in \mathbb{R}$. Therefore, we can introduce a set of gates $R_{\Gamma_k}(\vartheta)$ which realizes rotations by angle ϑ on the planes pointed out by an appropriate set of operators from the $SU(d)$

group, and this allows us to replace the generalized Pauli group G with the $SU(d)$ operators.

The procedure of constructing $SU(d)$ operators is universally defined for qudits, so it generates gates for pointed freedom level d. We begin with defining the set of projectors $P^{k,j}$ for each k and j such that $1 \leq k < j \leq d$, and υ, μ fulfill the relation $1 \leq \upsilon, \mu \leq d$:

$$P^{k,j} = |k\rangle\langle j| = [p_{\upsilon,\mu}]_{d\times d}, \quad p_{\upsilon,\mu} = \delta_{\upsilon,j}\delta_{\mu,k}, \tag{12}$$

where $P^{k,j}$ means that a value of one is in the k-th row and the j-th column, $p_{\upsilon,\mu}$ are entries of $P^{k,j}$. The symbols $\delta_{\upsilon,j}$ $\delta_{\mu,k}$ are the Kronecker deltas, and the result of their product is placed in the k-th row and j-th column of matrix P. The set of $d(d-1)$ generators, derived from the group $SU(d)$, is

$$\Theta^{k,j} = P^{k,j} + P^{j,k}, \quad \beta^{k,j} = -i(P^{k,j} - P^{j,k}) \tag{13}$$

for each pair of indexes k and j, where $\Theta^{k,j}$ are counterparts of the Pauli X operator, and $\beta^{k,j}$ are equivalents of the Pauli Y operator. Finally, the last set of $d-1$ generators is

$$\eta^{r,r} = \sqrt{\frac{2}{r(r+1)}}\left[\Big(\sum_{j=1}^{r} P^{j,j}\Big) - rP^{r+1,r+1}\right], \tag{14}$$

where $1 \leq r \leq (d-1)$, and $\eta^{r,r}$ correspond to the Pauli Z operator.

Other details about the construction of $SU(d)$ operators are given by Bertlmann and Krammer (2008).

The qudit-rotating gates $R_{\Gamma_k}(\vartheta)$ can be defined with the use of operators $\Theta^{k,j}$, $\beta^{k,j}$, and $\eta^{r,r}$ as consecutive Γ_k (where $k = 1, \ldots, d^2-1$). Particular rotations can be realized as the operator

$$R_{\Gamma_k}(\vartheta) = \exp\left(\frac{\mathbb{i}\Gamma_k\vartheta}{d}\right), \tag{15}$$

where $\vartheta \in \mathbb{R}$. These rotation gates are also unitary, which is easy to verify through $R_{\Gamma_k}(\vartheta)R^{\dagger}_{\Gamma_k}(\vartheta) = I$.

To induce an entanglement in a two-qubit system, we need only one Hadamard gate and one CNOT gate. The CNOT for qudits may be defined as

$$\mathrm{CX}|x\rangle|y\rangle = |x\rangle|(-x-y) \mod d\rangle. \tag{16}$$

Let us now briefly discuss non-unitary operations. In this work, we utilize a non-reversible operation which is quantum measurement. In detail, this transformation is the von Neumann projective measurement. Projective measurements always refer to a given computational basis because these operations tend to project a whole quantum state or just some specified qudits to one of basis vectors.

Let us mark the initial one-qudit state as $|\psi_{in}\rangle$, and a basis as a orthonormal set of vectors $\{|u_0\rangle, |u_1\rangle, \ldots, |u_{d-1}\rangle\}$. Performing the von Neumann measurement, which is used in the discussed classification circuit, requires the definition of the observable

$$M = \sum_{i=0}^{d-1} \lambda_i P_i, \tag{17}$$

where λ_i are the eigenvalues associated with projectors P_i, and consecutive $P_i = |u_i\rangle\langle u_i|$. The values λ_i are the measurement results. The probability of receiving particular λ_i is $p(\lambda_i) = \langle\psi_{\text{in}}|P_i|\psi_{\text{in}}\rangle$. The state of the qudit after the measurement is described by

$$|\psi_{\text{out}}\rangle = \frac{P_i|\psi_{\text{in}}\rangle}{\sqrt{\langle\psi_{\text{in}}|P_i|\psi_{\text{in}}\rangle}} = \frac{P_i|\psi_{\text{in}}\rangle}{\sqrt{p(\lambda_i)}}. \tag{18}$$

Because the character of quantum computations is probabilistic, each experiment must be performed many times, and finished with a measurement. The final result of computation is obtained as a probability distribution from all received outcomes. We use the expected value $\langle M\rangle$ which describes an average value of measurement results performed by observable M on state $|\psi\rangle$:

$$\langle M\rangle = \langle\psi|M|\psi\rangle. \tag{19}$$

Additionally, to evaluate if the classification was carried out correctly, we need a measure which estimates the similarity of two quantum states. In this work, we refer to the fidelity measure (MacMahon, 2007). The fidelity measure for pure quantum states is calculates as

$$F(\rho,\delta) = \mathrm{Tr}\Big(\sqrt{\sqrt{\rho}\delta\sqrt{\rho}}\Big) = |\langle\phi|\psi\rangle|, \tag{20}$$

where ρ and δ are density matrices: $\rho = |\psi\rangle\langle\psi|$, $\delta = |\phi\rangle\langle\phi|$, and $0 \leq F(\rho,\delta) \leq 1$. The higher fidelity, the more similar states $|\psi\rangle$ and $|\phi\rangle$.

3. Quantum computations for data classification

In this section, we propose a classifying quantum circuit based on a supervised learning approach. The aim of this section is to present introductory information concerning a unitary operation responsible for classification. Next, some details about the probability distribution and the loss function are given. The preparation of classical data to be suitable for a quantum circuit is described in Section 3.1.

We assume that there is a function f which classifies N elements of a data set $X = \{x_i\}$. A set $R = \{r_i\}$, where $r_i = f(x_i)$, contains information specifying which classes particular samples/observations x_i belong to.

The above-mentioned quantum circuit may be understood as a unitary operator which transforms a vector quantum state during the classification. This

operation and its equivalent notation (with the use of a density matrix) may be presented as

$$|\psi_c\rangle = U(\theta)|\psi_d\rangle, \quad \rho_c = U(\theta)\rho_d U^\dagger(\theta), \tag{21}$$

where $\rho_c = |\psi_c\rangle\langle\psi_c|$ and $\rho_d = |\psi_d\rangle\langle\psi_d|$.

The state $|\psi_d\rangle$ (or ρ_d) serves to encode the sample of classical information x_i. Marking $|\psi_d^{x_i}\rangle$ describes the correct quantum state which refers to the observation x_i (the procedure of converting classical data to quantum states is presented in Section 3.1). Here $|\psi_c\rangle$ includes the information regarding the class, and especially the state $|\psi_c^{x_i}\rangle$ contains a label for the observation x_i. The unitary operation U, depending on the parameter set θ, acts as a classification function f.

Remark 2. (*Probability distribution of the final state measurement*) It should be emphasized that after the operation U a measurement of the resulting quantum state must be performed. Estimation of the probability distribution for $|\psi_c\rangle$ requires many repetitions of the whole process. Precisely, to reach accuracy ε, an experiment has to be performed $O(N/\varepsilon^2)$ times.

If the classifying circuit generates a state $|\psi_c\rangle$, which assigns an observation to a class correctly, then it is possible to introduce the distance between the value $f(x_i)$ and the expectation for the sample x_i:

$$\text{dist} = \left|\left| f(x_i) - \Lambda\left(\langle M\rangle_{\psi_c}^{x_i}\right)\right|\right|, \tag{22}$$

where $\langle M\rangle_{\psi_c}^{x_i}$ is the expected value connected to the measurement of the final state $|\psi_c\rangle$, x_i being an observation from the data set X. Here Λ may be used as an element of additional classical processing (see Remark 3).

Small values of dist specify correct classification. For a given set of observations, we seek a form of $U(\theta)$ minimizing the value of the loss function

$$\mathbb{L} = \sum_i \left|\left| f(x_i) - \Lambda\left(\langle M\rangle_{\psi_c}^{x_i}\right)\right|\right|^2, \tag{23}$$

where parameters θ are decision variables in the optimization problem, and Λ again is an additional function which can be used to connect the obtained expected value with the class label of the sample x_i.

Remark 3. (*Role of the Λ function*) It should be emphasized that the values of $f(x_i)$ and $\langle M\rangle_{\psi_c}^{x_i}$ are real numbers. The role of the function Λ is supplementary conversion of expectation to a real number representing a class label of a sample x_i, i.e., it can rescale the expectation to another range compatible with the values of the $f(\cdot)$ function. However, in our case, we do not use the function Λ in our experiments, because the probability values after the measurement of state $\langle M\rangle_{\psi_c}^{x_i}$ are sufficient to make a correct classification (otherwise, the use of a function Λ should be considered).

3.1. Converting classical data to quantum states.

Quantum computation needs specific data preparation. The observations have to be correct quantum states. In Section 4, we present the results of classification but, beforehand, it is meaningful to show the methods of classical data conversion to quantum states for examples described in Section 4.

The first step in the conversion is a procedure of data normalization (Li and Li, 2015), used in classical machine learning and data mining. Assume that a learning set X contains N observations with d attributes/variables. The normalization needs calculating the range of accepted values for each i-th attribute ($i = 0, \ldots, (d-1)$), so extreme values have to be determined, $\max_i X$ and $\min_i X$. If $x_{i,m}$ stands for the i-th attribute in the m-th observation ($m = 1, \ldots, N$), the normalized value is

$$\bar{x}_{i,m} = \frac{x_{i,m} - \min_i X}{\max_i X - \min_i X}. \tag{24}$$

Remark 4. After normalization, we have $\bar{x}_{i,m} \in [0, 1]$. In classical machine learning and data mining, this solution avoids exaggerated influence of some variables (e.g., with different magnitude values than other variables) on the model. The normalization is also helpful during the process of converting classical data to quantum states.

At this point, each attribute value within one observation is less than one, but the sum of their squared moduli, probably, does not equal one. Now, we have to carry out the second part of the conversion to meet the quantum normalization condition.

We would like to encode one d-attribute observation as a state of one d-level qudit. This qudit's i-th amplitude value is calculated as

$$\alpha_i^m = \sqrt{\frac{\bar{x}_{i,m}}{\sum_{l=0}^{d-1} \bar{x}_{l,m}}}, \tag{25}$$

where m is the number of the observation. We process all attributes this way, in each observation, and obtain an N-element set of quantum states needed for the experiments with the classifying quantum circuit.

After normalization, we obtain quantum states which can be transformed by a quantum circuit. However, these data may be utilized, e.g., as a basis for the construction of a two-label quantum state:

$$|\psi\rangle = \sin(a_0 + b_0\alpha_0^m)|0\rangle + \cos(a_1 + b_1\alpha_1^m)|1\rangle, \tag{26}$$

where a_0 and b_0 represent the additional constants encoding data for the class label zero, and respectively, a_1, b_1 for the class label one. These parameters help us to improve the quality of classification. In the case of qudits

(where dimensions are denoted by d), the quantum state can be expressed with the use of $R_{\Gamma_k}(\vartheta)$ rotation gates:

$$|\psi\rangle = \prod_{i=0}^{d-1} R_{\Gamma_k}(a_i + b_i \alpha_i^m)|i\rangle, \tag{27}$$

where the Γ_k's describe the gates based on the operator $\Theta^{k,j}$, and constants a_i, b_i play the same role as in the previous case for qubits.

Remark 5. (*Value selection for a_i and b_i*) It should be emphasized that constants a_i and b_i are directly set during learning. In the presented examples, we assume that $a_i = 0$ and $b_i = 1$. Selection of these values in the process of data preparation introduces preliminary demarcation of samples.

3.2. Construction of the classifying circuit. Figure 1 depicts a scheme of the classifying circuit. The circuit contains rotation gates $R_Y(\theta_i)$ and gates E introducing an entanglement (we use CNOT gates). The angles θ_i are calculated for each rotation gate with the learning methods described in Section 3.3. It is crucial that the rotations are performed before the entanglement. Let us mark a set of rotation gates and one succeeding entanglement gate as a unitary operation U_i which acts on two qubits (in general qudits) i_0 and i_1:

$$U_i = R_Y^{(i_0)}(\theta_i) R_Y^{(i_1)}(\theta_{i+1}) \mathrm{CNOT}^{(i_0,i_1)}, \tag{28}$$

where i represents the number of unitary operations U in the quantum circuit. Notation $R_Y^{(i_0)}$ and $R_Y^{(i_1)}$ means that application of the R_Y operation to the i_0-th and i_1-th qubits (it is not necessary for the corresponding qubits to be adjacent). The whole classifying circuit is built of the sets of U_i operations, and all U_i operations will be denoted as U_Ω.

Additionally, if we want to focus on some subset of gates, e.g., numbers from 1 to m, it will be expressed as

$$U_{1:m} = \prod_{k=1}^{m} U_k(\theta_k). \tag{29}$$

Let us denote by $\mathcal{L} \in \mathbb{N}$ the number of qubits in the circuit. The value of $\mathcal{L}$ will be also termed as the height of a circuit, and $\mathcal{L} = L + 1$ where one qubit state (this with letter d or c in superscript) is crucial and other L states play the role of auxiliary states. The initial crucial state is $|\psi^d\rangle$. It corresponds to $|\psi_d^{x_i}\rangle$, i.e., it contains a single observation which is subject to classification. The final crucial state is denoted as $|\psi^c\rangle$, and relates to $|\psi_c^{x_i}\rangle$ because it represents the class which were pointed out by the circuit for a given observation. On the other hand, the number of operations in U_Ω, denoted by ζ, also determines the depth of a circuit. According to Fig. 1, the depth of the circuit is marked as $\mathcal{D}$.

Remark 6. The classifying circuit, presented in this article, utilizes rotation gates R_Y for data encoded in qubit states. In one of the experiments (three dimensional blobs), we use qutrits, and then gates R_{Γ_k} are used where Γ_k points the $\beta^{k,j}$ operators, defined by (13). The CNOT gates, introducing entanglement, are replaced by CX gates described by (16).

It should be emphasized that we do not restrict the number of qubits to one in $|\psi_d\rangle$ and $|\psi_c\rangle$. The presented approach remains correct if, e.g., $|\psi_d\rangle$ is a three-qubit state and $|\psi_c\rangle$ is a two-qubit state. However, the laws of quantum mechanics, like non-cloning (Park, 1970; Wootters and Zurek, 1982; Ortigoso, 2018) and no-deleting (Pati and Braunstein, 2000) theorems impose the same number of qubits on the circuit's input and output, unlike, for example, in classical neural networks.

The circuit is able to perform the classification task after the process of learning, which is given in Section 3.3. During training, an observation is written in $|\psi_d\rangle$. Next, the values of parameters θ are estimated. Finally, the state $|\psi_c\rangle$ is obtained as a transformation of $|\psi_d\rangle$ by the circuit. The state $|\psi_c\rangle$ should contain a class label for the analyzed observation. Naturally, the process of learning has to be repeated many times to estimate correctly the values θ, and to obtain the probability distribution for $|\psi_c\rangle$.

Remark 7. (*Qudits advantage*) Utilizing qudits simplifies the construction of the classifying circuit for multidimensional data. Qudits may be directly applied to represent data, according to the conversion presented in Section 3.1. Of course, also two qubits may be adopted to present four-dimensional data, but using ququads (i.e., qudits with $d = 4$) allows us to build the circuit with the same logical structure for qubits as for qudits (we simply replace CNOT and R_Y gates with their qudit counterparts).

After learning, the usage of the circuit is very similar. An observation is encoded in $|\psi_d\rangle$ and the initial quantum state is transformed by the circuit, but in this case parameters θ are not changed. At the end, the probability distribution for the final state is prepared to receive information concerning the class label for the analyzed observation.

Remark 8. (*Structure property of the quantum circuit for classification*) Utilizing rotation and entanglement gates, and their regular arrangement, as adjacent operations, simplifies the implementation of the circuit for the physical solutions like IBM (IBM, 2019), Rigetti (Rigetti, 2019), or Google (Gibney, 2019), where the circuit's topology requires direct connections between some qubits, and some operations have to be adjacent.

3.3. Learning algorithm. The computational steps required in a process of quantum classifier learning are described by Algorithm 1.

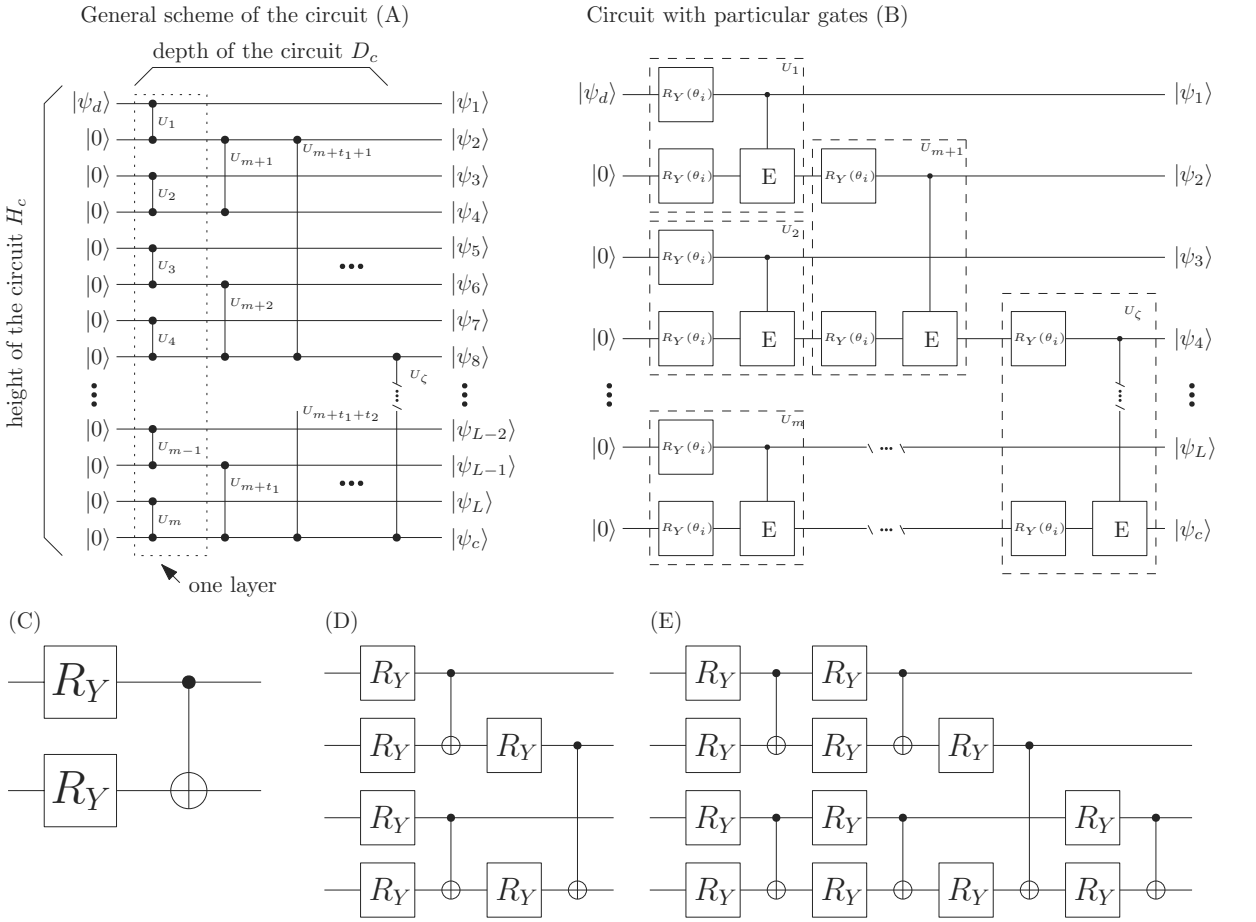

Fig. 1. General structure of the quantum circuit for data classification. The quantum circuit (a) depicts how the operation U_i is performed on particular qubits/qudits. The circuit (b) presents types of gates utilized in the classifying circuit. The controlled gates E introduce the entanglement, and here the CNOT gates are used. Gates $R_Y(\theta_i)$, with optimized parameters θ, are responsible for rotations around the y-axis (for qubits, but for qudits rotation planes are pointed out by the operators $R_{\Gamma_k}(\vartheta_i)$). The qubit/qudit $|\psi_d\rangle$ represents the input data, and the qubit/qudit $|\psi_c\rangle$ is the final state pointing out a class of the analyzed sample. The additional qubits/qudits $|\psi_l\rangle$ (where $1 \leq j \leq L$, and $L \in \mathbb{N}$) play the ancillary role and are not significant for data classification. The circuits (c), (d) and (e) are examples of circuits which may be constructed as special cases of schemes (a) and (b). The circuit (e) was used in the numerical experiments in this work (the parameters θ_i for R_Y gates are omitted to increase readability).

Remark 9. After learning, validation is recommended. The new form of the operator $U(\theta)$ should be verified on a test data set to assess if the selection of parameters θ is effective, i.e., the value of $\mathbb{L}$ was diminished.

The most important step in the procedure of learning is calculating the angles θ_i for each rotation gate, in the context of input data. The classifying circuit realizes a sequence of unitary operations U_i, and the expectation for the observable M, utilized in the last measurement, depends on the parameters θ_i:

$$\langle M(\theta)\rangle = \mathrm{Tr}\left(MU_i\rho_d{U_i}^\dagger\right) \quad \text{for all } i. \tag{30}$$

The operations U_i consist of gates R and CNOT, so the gradient may be calculated as

$$\frac{\partial\langle M\rangle}{\partial\theta_i} = -\frac{\mathrm{i}}{2}\mathrm{Tr}\left(MU_{i:\zeta}\left[\Gamma_k^{(i)}, U_{1:i-1}\rho_d U_{1:i-1}^\dagger\right]{U_{i:\zeta}}^\dagger\right), \tag{31}$$

where $\Gamma_k^{(i)}$ denotes that one of the generalized Pauli operators is applied to the i-th qubit/qudit. The above relation contains a commutator which cannot be calculated directly. The operations U_i appear on both the sides of the equation, and values θ_i are also unknown. However, the rotation angles θ may be computed with the use of arbitrary ρ (Li *et al.*, 2017), the density matrix of the quantum state,

Algorithm 1. Learning algorithm.

(I) Converting classical data $\{x_i\}$ to the form of quantum states (see Sec. 3.1).

(II) Performing transformation U with parameters θ on $|\psi_d\rangle$ to obtain a state $|\psi_c\rangle$.

(III) Realizing the measurement procedure and checking the class label of the sample (during the learning process expectations M are calculated for each desirable measurement of significant qubit/qubits in the state $|\psi_c\rangle$, i.e., $f(x_i) \equiv \langle MU(\theta)|\psi_d^{x_i}\rangle\rangle_{\psi_c}^{x_i}$.

(IV) Minimizing the loss function $\mathbb{L}$ by adjusting parameters θ.

$$[\Gamma_k^{(l)}, \rho] = \mathbb{i}\left[R_{\Gamma_k}^{(l)}\left(\frac{\pi}{2}\right)\rho R_{\Gamma_k}^{(l)}\left(\frac{\pi}{2}\right)^{\dagger} - R_{\Gamma_k}^{(l)}\left(-\frac{\pi}{2}\right)\rho R_{\Gamma_k}^{(l)}\left(-\frac{\pi}{2}\right)^{\dagger}\right], \tag{32}$$

and operators $R_{\Gamma_k}^{(l)}$ realize the $\pm\pi/2$ angle rotation around the X, Y or Z axis with the use of one Pauli operator Γ_k on a qubit l (and similarly, in a general case, for qudits where suitable $SU(d)$ operators should be used, however, the correct phase changing needs not only introducing minus or $\mathbb{i}$; for qudits this process requires d-degree roots of unity). Utilizing this commutator allows calculating the gradient for angles θ, taking into consideration expectation M:

$$\frac{\partial\langle M\rangle}{\partial\theta_i} = \frac{1}{2}\mathrm{Tr}\left(MU_{i+1:\zeta}U_i\left(\frac{\pi}{2}\right)\rho_i U_i\left(\frac{\pi}{2}\right)^{\dagger}U_{i+1:\zeta}^{\dagger}\right) - \frac{1}{2}\mathrm{Tr}\left(MU_{i+1:\zeta}U_i\left(-\frac{\pi}{2}\right)\rho_i U_i\left(-\frac{\pi}{2}\right)^{\dagger}U_{i+1:\zeta}^{\dagger}\right), \tag{33}$$

where $\rho_i = U_{1:i}\rho_d {U_{1:i}}^{\dagger}$, and U_i with parameters θ_i take previous or initial θ_i values. Therefore, choosing appropriate versions of rotation gates from the set γ allows calculating precise values of θ_i for the expectation $\langle M\rangle_i^{\pm}$, which is also the observable of the $|\psi^c\rangle$ state:

$$\frac{\partial\langle M\rangle}{\partial\theta_i} = \frac{\langle M\rangle_i^{+} - \langle M\rangle_i^{-}}{2}. \tag{34}$$

The utilized method is directly derived from the works of Li *et al*. (2017) and Mitarai *et al*. (2018).

4. Numerical experiments

4.1. Data classification. The correctness of classification, described in this article, is presented with the use of Moons, Circles and Blobs data sets (Pedregosa *et al*., 2011), which contain pairs *observation–class label*. In this work, data from the first two sets were encoded in qubit states. The Blobs set gives as a prospect to generate two and three dimensional observations. We perform two separable experiments for this set. In the first one data samples have two attributes and are encoded as qubits. In the second experiment we utilize samples with three attributes which are encoded as qutrits, i.e., qudits with $d = 3$.

In each experiment, 768 samples were used, where 512 played the role of a learning set, and the other 256 were included in the test (validation) set. The classifying circuit utilized two qubits or qutrits, and contained four layers of gates. The general scheme of the circuit is presented in Fig. 1(e).

Figure 2 shows the data sets for binary classification. These sets were subjected to experiments with circuits of Figs. 1(c) and (d) (see Fig. 1). Figures 2(a)–(g) present data after the normalization process, while Figs. 2(d)–(f) depict the same data but as the probability amplitudes of quantum states before the process of learning. Finally, Figs. 2(g)–(i) show results of the classification (after the learning process).

It can be seen that after learning the circuit was able to separate quantum states into two different classes. In the presented cases, the gaps between classes differs. The sizes of the gap are described by parameters a_r and b_r, given in (26) and (27), and are obtained during the learning process.

It must be emphasized that Figs. 2(d)–(f) display the representation of quantum states from the learning set before the learning process. These states are the input of the circuit and represent data after the normalization, and encoding as the state given in (26), with $a_i = 0$ and $b_i = 1$. After learning, the quantum state may be calculated using the partial trace, to inspect the changes before the final measurement. In each case, the states are separated into two classes. Additionally, we can see that the states for the Blobs set are properly separated even before the classification process—this is why only one U_i operation is sufficient to realize the classification task.

Table 2 shows the values of fidelity for the test sets. As we can see, the obtained intervals clearly define if the sample belongs to the set labeled as zero or to the set labeled as one. The samples with the fidelity measure greater than 0.75 are labeled as zero. The samples included in the first class reach maximally the value of fidelity 0.64. This difference in fidelity can be directly depicted with the previously presented examples from Fig. 2.

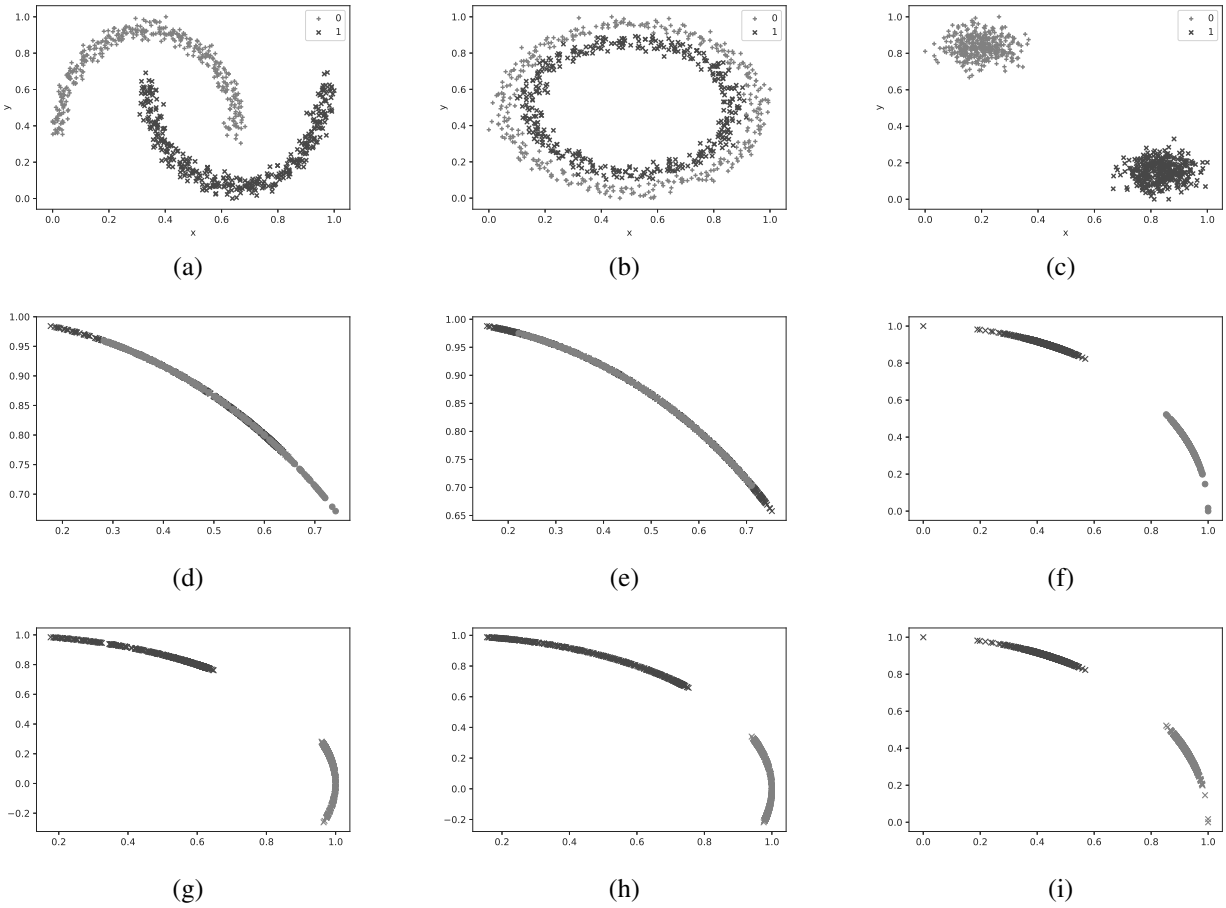

Fig. 2. Classification of two-dimensional data from the sets Moons, Circle and Blobs. Panels (a)–(c) represent original classical data, (d)–(f) depict data after the normalization, and (g)–(i) show the probability amplitudes of $|\psi^c\rangle$ after learning, but before a measurement operation on this state.

Table 2. Results of the classification for the sets Moons, Circles, Blobs (for qubits and qutrits) as the intervals of obtained Fidelity measure values. The symbol F_0 refers to the class labeled as zero, and F_1 to that labeled as one.

Set	F_0	F_1
Moons	(0.80, 0.95)	(0.11, 0.60)
Circles	(0.75, 0.91)	(0.03, 0.46)
Blobs	(0.83, 0.92)	(0.16, 0.64)
Blobs qutrits	(0.80, 0.93)	(0.15, 0.59)

Figure 3 shows the data classification for the Blobs set with three attributes. In this case, we use qutrits instead of qubits, but the general structure of the circuit remains the same. The gates R_Y and CNOT are replaced by their qutrit equivalents. There is no need to perform any other additional operations connected with data encoding, except the steps presented in Section 3.1. If we would like to utilize qubits for this set, then the data should be extra pre-processed or the initial state $|\psi^d\rangle$ should be a two-qubit state.

4.2. Function approximation. The presented classification circuits can be also used as universal approximators. It means that they are able to estimate values of non-linear functions like the sine, the cosine, the hyperbolic tangent or, e.g., $\exp(-x^2)$.

The approximation of non-linear functions can be realized by the same circuit which was used for the classification for sets Moons, Circles and Blobs. In the learning set, we can place input-output pairs (x_0, x_1) where x_0 contains the values of function domain ($x_0 = \langle -1, 1\rangle$), and x_1 is the value of the analyzed function. Now, the initial and final states are

$$|\psi_d\rangle = \sin(x_0)|0\rangle + \cos(x_0)|1\rangle,$$
$$|\psi_c\rangle = \sin(x_1)|0\rangle + \cos(x_1)|1\rangle.$$

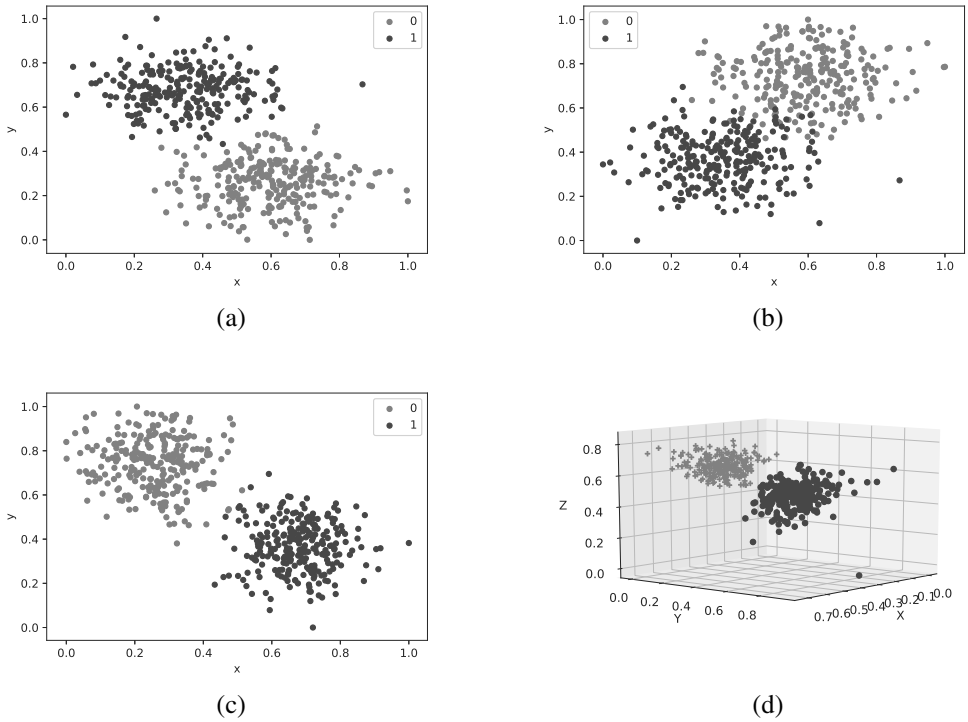

Fig. 3. Binary classification for the three-attribute data from the Blobs set. Panels (a)–(c) show the data after the normalization process as projections on planes XY, XZ and YZ. Panels (d) presents the final states $|\psi_c\rangle$ of the qutrit, where particular amplitudes are used as coordinates in a 3D space: two classes of objects are noticeable.

In the learning process, we do not change the loss function but a criterion for the evaluation of the approximation quality must be added. The quality of approximation for the function f can be expressed as $\hat{f}$ for the sample x_i:

$$\hat{f}(x_i) = \langle \psi_d^{x_i} | V | \psi_d^{x_i} \rangle, \tag{35}$$

where V is a diagonal operator, subjected to optimization for the data samples from the set X. The quality rate, in this case, is

$$Q = \sum_{i=1}^{n} \left(f(x^{(i)}) - \hat{f}(x^{(i)}) \right)^2, \tag{36}$$

which is the mean-square error calculated as the aggregated squared difference between the value of non-linear function f and the value $\hat{f}$ approximated by the quantum circuit.

5. Conclusions

In this article, we have demonstrated that the results of the previous work of Mitarai *et al*. (2018), can be applied in a form of much simpler quantum circuits which are able to solve a classification task. Additionally, the construction of the corresponding circuits is regular, simple, and easy to scale for tasks with more variables. Another advantage is that the circuits presented in this paper are also universal approximators of non-linear functions.

The approach presented in this work is characterized by the methods of parameters selection, and data read-in, i.e., properly converted data are an input signal for the circuit, and the parameters selection is based on adjusting angles θ of rotating gates. Therefore, the data is transformed by the circuit as the input states, and the circuit parameters θ after the learning process remain constant for the given data set, which makes the circuit reusable (without any changes in parameters θ).

The proposed method of selecting parameter values θ is based on a gradient change in rotation angles. It is able to adjust the circuit parameters so efficiently for binary classification that the task is solved without any errors (after the learning process, the tests were performed with another set of samples, so the model's overfitting is excluded). Naturally, some data sets contain samples with coordinates pointing out their location close to samples from another class, and in these cases the classification process is harder to realize. It should be emphasized that the process of learning may be simulated on a classical computer but also can be performed on a quantum machine. The simplicity of the proposed circuit makes this solution suitable for quantum hardware in the

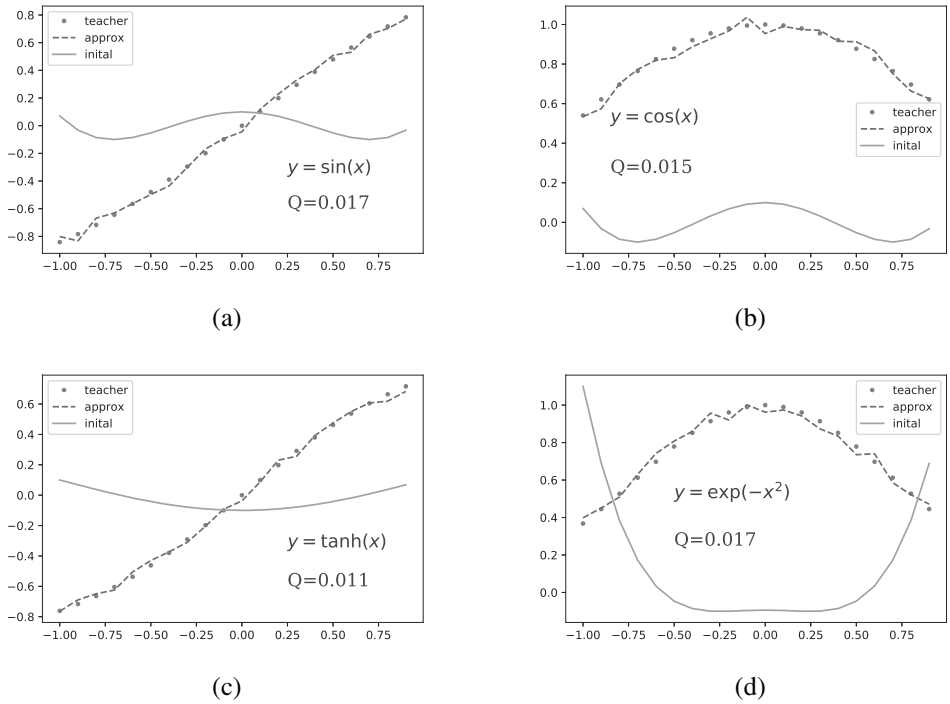

Fig. 4. Classifying circuit as an approximator of non-linear functions. In all cases, a good quality (the value of parameter Q) of approximation was reached. The initial line represents the initial values, and approximated values are depicted by the dashed line. The number of analyzed samples is 20.

nearest future.

Acknowledgment

We would like to thank the Q-INFO group at the Institute of Control and Computation Engineering (ISSI) of the University of Zielona Góra, Poland, for useful discussions. We also wish to thank the anonymous referees for valuable comments on the preliminary version of this paper. The numerical results were obtained using the hardware and software available at the GPU μ-Lab located at the Institute of Control and Computation Engineering of the University of Zielona Góra, Poland.

References

Augusto, L.M. (2017). *Many-Valued Logics: A Mathematical and Computational Introduction*, College Publications, London.

Bertlmann, R. and Krammer, P. (2008). Bloch vectors for qudits, *Journal of Physics A: Mathematical and Theoretical* **41**(23): 235303, DOI: 10.1088/1751-8113/41/23/235303.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S. (2017). Quantum machine learning, *Nature* **549**(7671): 195–202, DOI: 10.1038/nature23474.

Gibney, E. (2019). Hello quantum world! Google publishes landmark quantum supremacy claim, *Nature* **574**(7779): 461–462, DOI: 10.1038/d41586-019-03213-z.

IBM (2019). *Q Experience*, `https://quantum-computing.ibm.com/`.

Kołaczek, D., Spisak, B.J. and Wołoszyn, M. (2019). The phase-space approach to time evolution of quantum states in confined systems: The spectral split-operator method, *International Journal of Applied Mathematics and Computer Science* **29**(3): 439–451, DOI: 10.2478/amcs-2019-0032.

Li, J., Yang, X., Peng, X. and Sun, C. (2017). Hybrid quantum-classical approach to quantum optimal control, *Physical Review Letters* **118**(15): 150503, DOI: 10.1103/PhysRevLett.118.150503.

Li, Z. and Li, P. (2015). Clustering algorithm of quantum self-organization network, *Open Journal of Applied Sciences* **05**(6): 270–278, DOI: 10.4236/ojapps.2015.56028.

MacMahon, D. (2007). *Quantum Computing Explained*, John Wiley, Hoboken, NJ.

Mitarai, K., Negoro, M., Kitagawa, M. and Fujii, K. (2018). Quantum circuit learning, *Physical Review Letters* **98**(3): 032309, DOI: 10.1103/PhysRevA.98.032309.

Narayanan, A. and Menneer, T. (2000). Quantum artificial neural network architectures and components, *Information Sciences* **128**(3–4): 231–255, DOI: 10.1016/S0020-0255(00)00055-4.

Nielsen, M. and Chuang, I. (2010). *Quantum Computation and Quantum Information, 10th Anniversary Edition*, Cambridge University Press, Cambridge.

Ortigoso, J. (2018). Twelve years before the quantum no-cloning theorem, *American Journal of Physics* **86**(3): 201–205, DOI: 10.1119/1.5021356.

Park, J. (1970). The concept of transition in quantum mechanics, *Foundations of Physics* **1**(1): 23–33, DOI: 10.1007/BF00708652.

Pati, A.K. and Braunstein, S.L. (2000). Impossibility of deleting an unknown quantum state, *Nature* **404**(6774): 164–165, DOI: 10.1038/35004532.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830, DOI: 10.5555/1953048.2078195.

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E. and Latorre, J. (2020). Data re-uploading for a universal quantum classifier, *Quantum* **4**: 226, DOI: 10.22331/q-2020-02-06-226.

Rigetti (2019). *Quantum Computing Systems*, `https://www.rigetti.com/systems`.

Schuld, M., Sinayskiy, I. and Petruccione, F. (2014). Quantum computing for pattern classification, *in* D.-N. Pham and S.-B. Park (Eds), *PRICAI 2014: Trends in Artificial Intelligence*, Springer, Cham, pp. 208–220, DOI: 10.1007/978-3-319-13560-1_17.

Schuld, M., Sinayskiy, I. and Petruccione, F. (2015). An introduction to quantum machine learning, *Contemporary Physics* **56**(2): 172–185, DOI: 10.1080/00107514.2014.964942.

Veenman, C. and Reinders, M. (2005). The nearest sub-class classifier: a compromise between the nearest mean and nearest neighbor classifier, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(9): 1417–1429, DOI: 10.1109/TPAMI.2005.187.

Weigang, L. (1998). A study of parallel self-organizing map, *arXiv*: quant-ph/9808025v3.

Wiebe, N., Kapoor, A. and Svore, M. (2015). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning, *Quantum Information and Computation* **15**(3–4): 316–356.

Wiśniewska, J. and Sawerwain, M. (2020). Simple quantum circuits for data classification, *in* N.T. Nguyen *et al.* (Eds), *Intelligent Information and Database Systems*, Springer, Cham, pp. 392–403, DOI: 0.1007/978-3-030-41964-6_34.

Wootters, W. and Zurek, W. (1982). A single quantum cannot be cloned, *Nature* **299**(5886): 802–803, DOI: 10.1038/299802a0.

Zoufal, C., Lucchi, A. and Woerner, S. (2019). Quantum generative adversarial networks for learning and loading random distributions, *Quantum Information* **5**(1): 103, DOI: 10.1038/s41534-019-0223-2.

Joanna Wiśniewska received her MSc degree in computer science in 2003 at the Military University of Technology in Warsaw, Poland. Then she started working as a research assistant in the same university at the Faculty of Cybernetics. She obtained her PhD degree in 2012 in the field of quantum algorithms. As an assistant professor, she continues her research concerning the problems of pattern recognition, quantum algorithms, quantum communication and computer simulation. One of her most important achievements is being a co-author of a monograph on quantum computing published by Polish Scientific Publishers PWN in 2015.

Marek Sawerwain received his ME and PhD degrees in 2004 and 2010, respectively, at the Faculty of Electrical Engineering, Computer Science and Telecommunications of the University of Zielona Góra, Poland. His current scientific interests include quantum communication methods, quantum computation methods, and the theory of quantum programming languages. He also conducts research in the field of quantum computation modeling and simulation, and focuses on creating effective algorithms for solutions based on modern multicore CPU and GPU technology. He is currently associated with the Faculty of Computer, Electrical and Control Engineering, University of Zielona Góra, as an assistant professor.

Andrzej Obuchowicz obtained his MSc (1987) and PhD (1992) degrees in physics, and his DSc (2004) in automatic control and robotics, all from the Wrocław University of Technology. Since 2015 he has held a full professorial title. He is currently the dean of the Faculty of Computer, Electrical and Control Engineering of the University of Zielona Góra, Poland. His research interests include soft computing methods, especially neural networks, evolutionary algorithms and other metaheuristic techniques, and their application to global optimization problems, adaptation in non-stationary environments, computer-aided medical diagnosis, fault tolerant and fault diagnosis systems, the max-plus algebra approach to discrete event systems as well as quantum computing. He is the author or a co-author of 4 books and over 120 other publications.

Received: 23 January 2020
Revised: 9 April 2020
Re-revised: 1 July 2020
Accepted: 29 October 2020