~~Quantum assisted~~ Quantum-assisted unsupervised data clustering ~~on the basis of neural networks~~

I. D. Lazarev

Institute of Problems of Chemical Physics of Russian Academy of Sciences,
Acad. Semenov av. 1, Chernogolovka, Moscow Region, Russia, 142432 and
Faculty of Fundamental Physical-Chemical Engineering,
Lomonosov Moscow State University, GSP-1, Moscow, Russia 119991

Marek Narozniak

New York University Shanghai, 1555 Century Ave, Pudong, Shanghai 200122, China and
Department of Physics, New York University, New York, NY, 10003, USA.

Tim Byrnes

New York University Shanghai, 1555 Century Ave, Pudong, Shanghai 200122, China
State Key Laboratory of Precision Spectroscopy, School of Physical and Material Sciences,
East China Normal University, Shanghai 200062, China
NYU-ECNU Institute of Physics at NYU Shanghai,
3663 Zhongshan Road North, Shanghai 200062, China
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan and
Department of Physics, New York University, New York, NY 10003, USA

A. N. Pyrkov*

Institute of Problems of Chemical Physics of Russian Academy of Sciences,
Acad. Semenov av. 1, Chernogolovka, Moscow Region, Russia, 142432
(Dated: April 23, 2020)

Unsupervised machine learning is one of the main ~~directions in the area of~~ techniques in artificial intelligence. Quantum computers offer opportunities to speed up such machine learning techniques, where the number of input data elements is extremely large. Here, we propose a realization of quantum assisted unsupervised data clustering ~~on the basis of artificial neural network called a~~ using the self-organizing feature map artificial neural networks. We make a proof-of-concept realization of one of the ~~crucial parts~~ central components of the approach on ~~IBM quantum machine~~ the IBM Q Experience and show that it allows us to reduce number of calculations in a number of clusters. We compare the results with the classical algorithm on a toy example of unsupervised text clustering.

I. INTRODUCTION

The combination of big data and artificial intelligence — dubbed the fourth industrial revolution — has profoundly affected the modern economy in a plethora of different ways from robotics to agriculture [1–5]. Contemporary artificial intelligence methods based on neural networks also have the potential to enhance the role of novel analytical methods in science and engineering [6–8]. Paradoxically, the exact mechanism of why neural networks are so powerful remain unknown (in many cases it is regarded as a black box). It has been speculated that limits to the neural network approach based on the computational power of von Neumann architecture are being approached, and improvements appear only due to heuristic escalation of complexity [9–11]. ~~In particularly~~ In particular, the self-organizing feature map (SOFM) [12–14], is a type of artificial neural network (ANN) that is trained in an unsupervised manner. SOFMs are used in many areas [15–22] and in comparison with many other artificial neural networks they apply competitive learning and preserve the topological properties of the input space [23]. The SOFMs with a small number of nodes are similar to the K-means algorithm but for larger SOFMs they represent data in a fundamentally topological way that allows one to do dimensionality reduction. Once it is trained, the map can classify a vector from the input space by finding the node with the smallest distance metric.

Meanwhile, there has been much interest recently in applying quantum computing techniques to machine learning [24–27]. ~~In the early stage, the main focus in QML~~ The main focus of early works in quantum machine learning (QML) was on the use of quantum computers to perform ~~some~~ basic linear algebra subroutines ~~faster~~ with a quantum speedup (such as Fourier transform and solving systems of linear equations [28–30]) ~~, open up the possibilities for exponential speed-ups~~ by utilizing quantum phenomena such as quantum entanglement and quantum superposition ~~over classical counterparts [25, 26]and, in the result, quantum~~ [25, 26]. Quantum versions were developed for linear regression, principal component analysis, support vector machine, K-means

* Email address:pyrkov@icp.ac.ru

algorithm and ~~so on [31–35]. Recently, a number of shifts occurred and the hottest trend migrated to~~ others [31–35]. More recently, there has been attention on developing quantum neural networks [36–38]. The ~~shift was inspired with the~~ interest in quantum neural networks was inspired by progress in experimental quantum computing when it became possible to ~~play around with~~ use parametrized quantum circuits, ~~whose~~ where the parameters behave much like the weights of a neural network [39]. In particular, quantum algorithms for training and evaluating feedforward neural networks ~~, was developed,~~ which are one of the most usable neural network ~~model, was developed~~ models [40, 41]. Recent results ~~connected~~ connecting the classical Bayesian approach to deep learning allowed ~~to develop~~ for the development of a new algorithm for Bayesian deep learning on quantum computers [42]. The quantum models for convolutional neural networks, which may be suitable for the problems of learning of quantum states were also proposed [43, 44]. Quantum classification, tested on the MNIST dataset, via ~~Slow Feature Analysis~~ slow feature analysis based on the use of ~~Quantum Frobenius Distance~~ quantum Frobenius distance was proposed [45]. Sublinear quantum algorithms for training linear and kernel-based classifiers were designed [46]. Furthermore, continuous-variable quantum neural networks [47], quantum autoencoders [48], quantum approaches for reinforcement learning [49–51] and some other [52–59] were developed. Currently, it is believed that implementation of quantum neural networks can be the main test bed to achieve practical quantum supremacy on ~~NISQ~~ noisy intermediate scale quantum (NISQ) devices.

In this paper, we ~~realize~~ analyze a hybrid quantum assisted SOFM (QASOFM) and apply it to the data clustering problem in an unsupervised manner on a toy example of clustering paper abstracts. We implement the quantum assisted SOFM in such a way that it becomes possible to use the Hamming distance as a distance metric for the training and to reduce the number of distance calculations in the number of clusters. We propose an optimized circuit for realizing the Hamming distance on a quantum machine. We ~~prepare~~ then apply it to a toy example of data for clustering paper abstracts and give a proof-of-concept realization of the quantum assisted SOFM on the IBM Q experience quantum computer [60] and compare it ~~with~~ to the classical case.

II. THE QUANTUM ASSISTED SELF-ORGANIZING FEATURE MAP (SOFM)

A. The classical algorithm

The SOFM is one of the ~~best-known~~ most widely-used unsupervised learning methods ~~that is widely~~ used in various areas of modern science. It was first proposed by Kohonen as a self-organizing unsupervised learning algorithm which produces feature maps similar to those
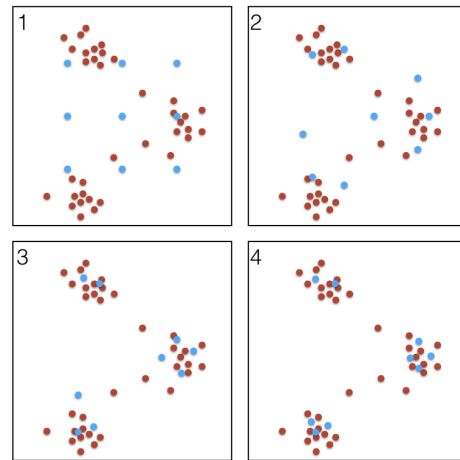

FIG. 1. Schematic illustration of the clustering problem considered in this paper. Blue dots represent clusters and red dots are data points. The training process moves clusters to fit the data points. Note that there are fewer clusters than data points, which is the essence of dimensionality reduction, and is what permits the model to generalize the data.

occurring in the brain [61]. The SOFM algorithm operates with a set of input objects, each represented by a N-dimensional vector, ~~as input~~ and maps them onto nodes of a two-dimensional grid.

The input dimensions are associated with ~~features and the~~ features, and the nodes in the grid ~~,~~ (called cluster vectors~~,~~) are assigned the N-dimensional vectors~~; the~~. The components of these vectors are usually called weights. Initially the weight components are chosen randomly. We then can train our SOFM adjusting the components through the learning process which ~~can be oversimplified into~~ occur in the two basic procedures ~~,~~ of selecting a winning cluster vector and updating its weights (Fig. I). ~~In more detail~~ More specifically, they consist of four step process: ~~1) select~~ 1. selecting an input vector randomly from the set of all input vectors; ~~2) find~~ 2. finding a cluster vector which is closest to the input vector; ~~3) adjust~~ 3. adjusting the weights of the winning node in such a way that it becomes even closer to the input vector; ~~4) repeat this process~~ 4. repeating this process for many iterations until it converges.

After the winning cluster vector is selected, the weights of the vector are adjusted according to

$$\vec{w}_{\text{new}} = \vec{w}_{\text{old}} + \alpha\left(\vec{x} - \vec{w}_{\text{old}}\right). \quad (1)$$

~~In simple words, this~~ The above expression can be ~~understood according to the following~~ interpreted according to: if a component of the input vector is greater than the corresponding weight, increase the weight by a small amount; if the input component is smaller than the weight, decrease the weight by a small amount. The larger the difference between the input component and the weight component,
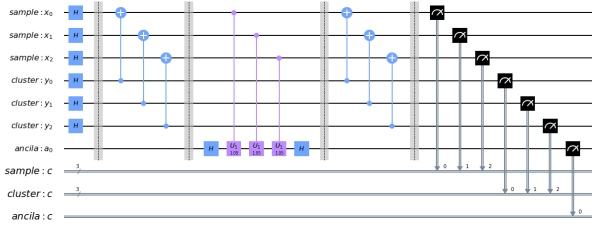
FIG. 2. ~~A quantum~~ Quantum circuit for the quantum parallelized Hamming distance calculation ~~algorithm~~ between all samples and clusters~~at once~~. First, ~~we apply~~ Hadamard gates are applied in order to ~~get~~ obtain a superposition of the sample and cluster registers. Second, we encoded information about pairwise different qubits in a quantum state of the sample register with applying the CNOT gates. Third, Hamming distance values are encoded in the amplitudes of superposition with the control phase rotation and Hadamard gates. Finally, a quantum state of the sample register returned to the initial basis for information retrieval.
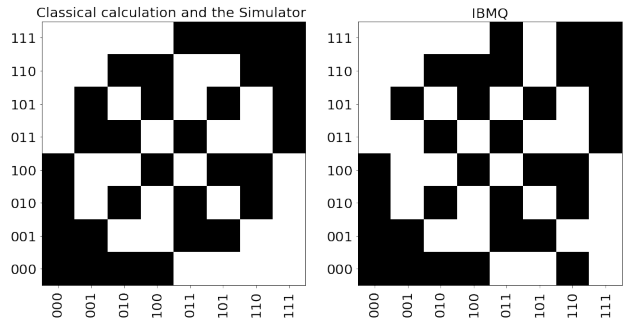

FIG. 3. ~~The~~ Hamming distance matrix between two data sets of binary vectors. Values of distance that are less than the median distance value ~~was~~ are marked black. ~~The classical~~ Classical simulation of the quantum circuit shows perfect agreement to theoretical calculations and presented on the left figure. Result obtained on ~~IBMQ "~~the IBM Q Experience "ibmq_16_melbourne~~"~~" backend is shown on the right figure. We can see good agreement between the distance matrices calculated classically and on the IBM Q Experience.

the larger the increment (decrement). Intuitively, this procedure can be geometrically interpreted as iteratively moving the cluster vectors in space one at a time in a way that ensures each move is following the current trends inferred from their distances to the input objects. A visualisation of this process is ~~provided on figure (Fig. I)~~shown in Fig. I.

Usually the winning cluster vector is selected based on the Euclidean distance between an input vector and the cluster vectors. In our approach, we use the Hamming distance instead of the Euclidean distance to select the winning cluster vector. It allows us to use a simpler encoding of the classical information into the quantum state and use an effective procedure for the calculation of the Hamming distance on the quantum machine, such as to reduce the number of calculations in number of cluster vectors in comparison to the classical case.

B. Optimized quantum scheme for Hamming distance calculation

We now introduce an optimized algorithm for calculating the matrix of Hamming distances [62] between a sample vector and all cluster vectors~~at once~~, making use of quantum parallelism. This allows for a simple encoding of the classical information into a quantum register.

The overall procedure involves two registers of n qubits each, denoted $|X\rangle$ and $|Y\rangle$, along with a single auxiliary qubit $|a\rangle$. During ~~entire processing~~the whole process, the $|Y\rangle$ register is used to store the cluster states~~, then at~~. At the beginning and ~~at the~~end of the ~~processing~~procedure the $|X\rangle$ register stores the input vectors~~, while during the processing~~. Meanwhile, during the procedure it stores the differences between input vectors and cluster states.

~~Now for this demonstration let~~Let us assume we have k input vectors and l cluster states. The ith input vector

and jth cluster vector are respectively denoted as $|x_i\rangle$~~and~~, $|y_j\rangle$.

The registers $|X\rangle$ and $|Y\rangle$ ~~will properly store~~ are initialized to store the input vectors and cluster vectors ~~if initialized as follows.~~ according to

$$|X\rangle = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} |x_i\rangle, \qquad (2)$$

$$|Y\rangle = \frac{1}{\sqrt{l}} \sum_{j=1}^{l} |y_j\rangle. \qquad (3)$$

~~Both~~ The two registers along with the auxiliary qubit ~~compose into~~ comprise the initial state of the ~~entire circuit in a form denoted below.~~ quantum computer according to

$$|\psi_0\rangle = |X\rangle |Y\rangle |a\rangle \qquad (4)$$

where $|a\rangle$ is an auxiliary qubit in the state $|0\rangle$ initially.

Given this initial state we may begin the processing of the problem. We start by applying a CNOT gate between $|X\rangle$ and $|Y\rangle$

$$|\psi_1\rangle = \text{CNOT}(Y, X) |\psi_0\rangle = \frac{1}{\sqrt{kl}} \sum_{i,j}^{k} |d_{ij}^1, \ldots, d_{ij}^n\rangle |y_j^1, \ldots, y_j^n\rangle |0\rangle$$

$$|\psi_1\rangle = \text{CNOT}(Y, X)|\psi_0\rangle$$

$$= \frac{1}{\sqrt{kl}} \sum_{i,j=1}^{k} |d_{ij}^{(1)}, \ldots, d_{ij}^{(n)}\rangle |y_j^{(1)}, \ldots, y_j^{(n)}\rangle |0\rangle \qquad (5)$$

where $d_{ij}^{\alpha} = \text{CNOT}(y_i^{\alpha}, x_j^{\alpha})$, $\alpha = 1 \dots n$, and i, j are the qubit indexes in the registers.

At this stage of the computation the $|X\rangle$ no longer stores the input vectors, instead it stores the information about pairwise different qubits between the input vector $\{X\}$ and cluster vector $\{Y\}$.

Next, for each pair $\{X\}$ and $\{Y\}$, the accumulated information of all the differences is projected onto the amplitude of the superposed state. This is achieved by applying the Hadamard gate on auxiliary qubit, followed by a controlled phase gate on $|Xa\rangle$ defined as

$$R(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{pmatrix}, \quad \phi = \frac{\pi}{n} \tag{6}$$

Finally, another Hadamard gate is applied.

After the first Hadamard on the ancilla qubit the state is

$$|\psi_2\rangle = H_a |\psi_1\rangle = \frac{1}{\sqrt{kl}} \sum_{i,j\,i,j=1}^{k} \left| d_{ij}^{1(1)}, \dots, d_{ij}^{n(n)} \right\rangle \left| y_{j}^{1(1)} \dots \right\rangle \tag{7}$$

Applying the controlled phase gate the state then becomes:

$$|\psi_3\rangle = R_{(X,a)}\left(\frac{\pi}{n}\right) |\psi_2\rangle$$
$$= \frac{1}{\sqrt{2kl}} \sum_{i,j\,i,j=1}^{k} \left| d_{ij}^{1(1)}, \dots, d_{ij}^{n(n)} \right\rangle \left| y_{j}^{1(1)}, \dots, y_{j}^{n(n)} \right\rangle$$
$$+ \frac{1}{\sqrt{2kl}} \sum_{i,j\,i,j=1}^{k} \exp\left(\frac{-i\pi}{n} \sum_{l=1}^{n} d_{ij}^{l(l)}\right) \left| d_{ij}^{1(1)}, \dots, d^{n} \right\rangle$$
$$\times \left| y_{j}^{1(1)}, \dots, y_{j}^{n(n)} \right\rangle |1\rangle \tag{8}$$

Applying another Hadamard on the ancilla qubit we obtain

$$|\psi_4\rangle = \frac{1}{\sqrt{kl}} \sum_{i,j\,i,j=1}^{k} e^{\left(\frac{-i\pi}{2n} \sum_{l}^{n} d_{ij}^{l}\right)} \exp\left(\frac{-i\pi}{2n}\sum_{l=1}^{n} d_{ij}^{(l)}\right)$$
$$\times \left[\cos\left(\frac{\pi}{2n} \sum_{l=1}^{n} d_{ij}^{l(l)}\right) \left| d_{ij}^{1(1)}, \dots, d_{ij}^{n(n)} \right\rangle \left| y_{j}^{1(1)}, \dots, \right\rangle |0\rangle \right.$$
$$\left. + i \sin\left(\frac{\pi}{2n} \sum_{l=1}^{n} d_{ij}^{l(l)}\right) \left| d_{ij}^{1(1)}, \dots, d_{ij}^{n(n)} \right\rangle \left| y_{j}^{1(1)}, \dots, y_{j}^{n} \right\rangle |1\rangle \right] \tag{9}$$

This completes the step for projecting differences between pairs of $\{X\}$ and $\{Y\}$ onto the amplitude of the auxiliary qubit. The process is done in the x-basis, achieved by the surrounding Hadamard gates. There are two possible measurement outcomes of the auxiliary qubit. Each pair of $\{X\}$ and $\{Y\}$ forms a subspace of the Hilbert space, the controlled phase gate ensures to change amplitudes of those outcomes within this subspace depending on how different the spin configurations between $\{X\}$ and $\{Y\}$ are.

At this stage, the information regarding the differences between pairs of $\{X\}$ and $\{Y\}$ is no longer relevant, thus we return to our initial basis for register $|X\rangle$ by applying pairwise CNOT gates:

$$|\psi_f\rangle = \text{CNOT}(Y, X) |\psi_4\rangle$$
$$= \frac{1}{\sqrt{kl}} \sum_{i,j\,i,j=1}^{k} e^{\left(\frac{-i\pi}{2n} \sum_{l}^{n} d_{ij}^{l}\right)} \exp\left(\frac{-i\pi}{2n}\sum_{l=1}^{n} d_{ij}^{(l)}\right) \left[\cos\left(\frac{\pi}{2n} \sum \right. \right.$$
$$\left. + i \sin\left(\frac{\pi}{2n} \sum_{l=1}^{n} d_{ij}^{l(l)}\right) |X_i\rangle |Y_j\rangle |1\rangle \right]. \tag{10}$$

This makes $\{X\}$ store the input vectors again, as in the initial step. It however preserves the amplitudes of the auxiliary qubit which are proportional to how different each pairs of $\{X\}$ and $\{Y\}$ are.

We thus have the Hamming distances encoded into the amplitudes of the final state. From the statistics of the measurement outcomes, the distance matrix between two data sets of binary vectors can be obtained (Fig. II B). In this case, the biggest amplitude of the measurement result coincides with the smallest Hamming distance when the measurement result of the ancilla qubit is 0. If the ancilla qubit is 1, there is an inverse relationship with the measurement result.

Measuring the Hamming distance of a particular pair of input vectors $|X_i\rangle$ and cluster vector $|Y_j\rangle$ consists of extracting the relevant amplitude from the subspace that those states form, this can be done using the following projection operator

$$\Pi_{i,j} = |X_i\rangle\langle X_i| \otimes |X_i\rangle\langle Y_j| \otimes I. \tag{11}$$

Using the above projection operator, the subspace of the Hilbert space formed by a particular pair of input and cluster vectors can be traced out as

$$\rho_{i,j} = \text{Tr}_{1,\dots,2n}(\Pi_{i,j} |\psi_f\rangle\langle\psi_f| \Pi_{i,j}). \tag{12}$$

~~Now the following two relevant~~ From the state the following two amplitudes for the measurement results can be extracted ~~.~~ —

$$a_0(x_i, y_j) = \frac{\langle 0|\rho_{i,j}|0\rangle}{\text{tr}(\rho_{i,j})}, \frac{\langle 0|\rho_{i,j}|0\rangle}{\text{Tr}(\rho_{i,j})} \tag{13}$$

$$a_1(x_i, y_j) = \frac{\langle 1|\rho_{i,j}|1\rangle}{\text{tr}(\rho_{i,j})}, \frac{\langle 1|\rho_{i,j}|1\rangle}{\text{Tr}(\rho_{i,j})}. \tag{14}$$

In order to reduce noise we average the measurement results over different states of the ~~auxiliary~~ ancilla qubit, thus the measured Hamming distance between the input vector $|X_i\rangle$ and cluster vector $|Y_j\rangle$ ~~takes the following form.~~ is

$$d_{i,j}^H \propto 1 - \frac{1}{2}(a_0(x_i, y_j) + (1 - a_1(x_i, y_j))). \tag{15}$$

The Hamming distance measured in this way is ~~contained~~ bounded $0 \leq d_{i,j}^H \leq 1$, where ~~intuitively,~~ 0 ~~would indicate~~ indicates that x_i are y_j identical and 1 ~~would mean~~ means they are completely opposite in terms of their pairwise binary coordinates.

In comparison with the circuits for Hamming distance calculations proposed previously [62], our method allows one to reduce the number of gates in the circuit ~~(at least in~~ and reconstruct all distance matrix in parallel and can be implemented with high fidelity. We note that at least for the $2n$ one-qubit NOT gates, ~~depends on actual realization~~ this depends on the actual realization of the control phase gate in a quantum register ~~on~~ at the hardware level~~), reconstruct all distance matrix at once and can be implemented with high fidelity on current quantum devices~~.

III. ~~DEMONSTRATION~~ EXPEIRMENTAL DEMONSTRATION OF QASOFM~~ON THE IBM Q EXPERIENCE~~

We now show experimental results for a proof-of-concept demonstration of the algorithm introduced in the previous section, on a 16-qubit quantum computer provided by IBM Q Experience. We ~~implement~~ perform unsupervised data clustering for three sets of paper abstracts from three different fields: quantum physics [63–65], medicine [66–68] and biology [69–71]. Each set consists of three random papers that focus on one of following topics: "Quantum Machine Learning" (QML), "Cancer" (MED) and "Gene Expression" (BIO). Abstracts were vectorized by the bag-of-words model in order to choose most defining words in each data set (see Fig. II B). This model represents text as a multiset "bag" of its words taking into account only multiplicity of words. Preparing the bag-of-words we excluded the words that appear only in one abstract and more than in
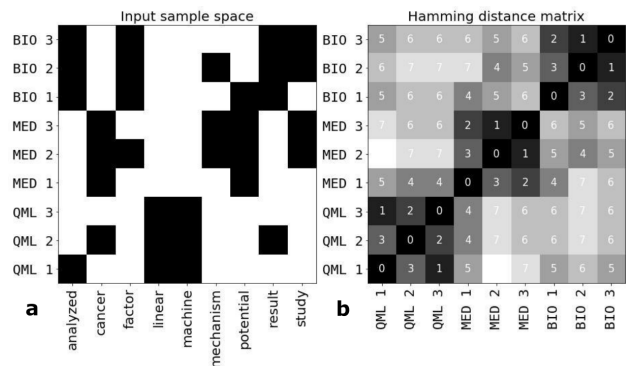

FIG. 4. (a) Representation of the data set of abstracts with the bag-of-words model is shown. Each abstract is represented by a binary vector with 9 elements, corresponding to the 9 words on the horizontal axis. The samples are sorted into groups (QML, MED, BIO) ~~,~~ with 3 papers for each tag, for a total of 9 paper. (b) The Hamming distance between each vectorized abstract is shown ~~with~~ as a number in the matrix. ~~The sets of abstracts are well separated.~~

4 abstracts and we also excluded the word "level" from consideration due to the frequent overlap between the clusters because it gives instabilities for both classical and quantum algorithms. We restricted our bag-of-word size to 9 of the most frequent words from the full ~~"~~bags-of-word ~~"~~ due to limitations of the number of qubits. After vectorizing and pre-processing the data, the clusters are well separated with the Hamming distance. We observe that distances between the abstracts inside clusters are smaller than distances between the abstracts from different clusters, showing ~~a~~ successful self-organization (Fig. II B).

~~Classical~~ In classical SOFM [12], ~~as it has been explained earlier, can be oversimplified into two basic procedures:~~ first a distance calculation between a ~~one~~ a sample vector and all cluster vectors ~~and shifting~~ is made, then the closest cluster vector ~~to~~ is shifted towards the sample vector. ~~In classical SOFM, the~~ The complexity of algorithm, in the sense of the number of distance calculations, scales as $O(LMN)$, where N is number of samples, M is number of randomly sampled cluster vectors, and L is number of the shifts of cluster vectors. In the QASOFM ~~(Sec. II)~~ described in the previous section, distance calculations are realized on a quantum device (i.e. the IBM Q Experience) with the use of circuit presented in Fig. II B. This approach allows one to reduce the number of operations in a number of cluster states with an optimized number of gates that are possible to realize on ~~currently available~~ quantum computing devices ~~with a limited number of qubits~~ that are currently available. The circuit realized in such a manner that calculation of Hamming distance between the sample vector and all cluster vectors is realized in one operation. The complexity of the quantum assisted SOFM then scales as $O(LN)$.

In order to check that our algorithm gives the ~~correct~~ expected results we compare it to classical calculations of
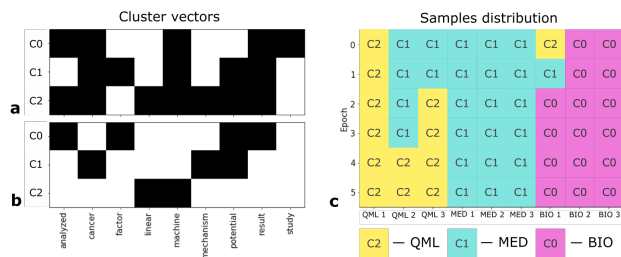
FIG. 5. (a) Initial random binary vectors of ~~clusters are presented. We randomly sampled 3 binary vectors of size 9 as initial~~ cluster vectors (labeled ~~by 0~~as C0, ~~1~~C1, ~~2~~C2)~~and it is shown how~~, for the selected 9 words from the bag of words model~~presented in them~~. (b) The result of applying our QASOFM implemented on the ~~IBMQ "~~IBM Q Experience "ibmq_16_melbourne"~~"~~ backend~~is shown on the bottom left figure~~. Vectors mean cluster elements for BIO, MED, QML groups, for the C0, C1, C2 cluster vectors, respectively~~(up to down)~~. (c) The evolution of label distribution on each learning epoch~~is presented on the right figure. Good convergence already is observed on the fourth epoch.~~

the distance matrix on two data sets of binary vectors, as shown in Fig. II B. We see good agreement between the distance matrices calculated classically and on the IBM Q Experience. The theoretical calculations and classical simulations show perfect agreement with each other.

An example of the QASOFM learning process is ~~given on~~ shown in Fig. ~~??~~5. Initially, the cluster vectors were randomly chosen (see Fig. ~~??) and~~5) and the label of sample distribution is shown for the zeroth epoch in Fig. ~~??~~5(a). In order to prepare a superposition of cluster vectors needed for the calculation of the distance matrix ~~at once~~ we use the standard initialization of QISKIT library. Each epoch of the algorithm requires 9 distance calculations in the quantum implementation (number of samples in general case) or 27 distance calculations for classical realization (product of number of samples and number of cluster vectors in general case). ~~In order to realize the circuit for the distance calculation for the quantum case we used the IBM Q machine.~~ After the distance calculation from each sample to all cluster vectors at each epoch we label each sample with the index of the closest cluster vector and shift the closest cluster vectors to the sample one. The shift is made by the change of the first binary element in the cluster vectors different from the sample one. The evolution of the labels presented on Fig. ~~??~~5(c). Good convergence is already observed in the fourth epoch.

IV. DISCUSSIONS

We have developed a quantum assisted SOFM and showed a proof-of-concept experimental demonstration

that it can be used to solve clustering problems in an unsupervised manner~~with potential to outperform classical counterpart in real life tasks~~. The procedure of solving such clustering problem requires calculating the distance many times in iterative way, which is calculated using a hybrid quantum-classical procedure. We introduced an optimized circuit for Hamming distance calculations that can be implemented on currently available quantum computing devices with high fidelity. Our ~~circuit could act as a~~ quantum circuit performs the distance-computing component of a classical SOFMs algorithm and in this way improve its performance. Due to wide use of classical SOFMs in different areas of modern research and technology, this can give opportunities for the use of QASOFM in practical applications in near term, outperforming classical algorithms.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

KEYWORDS

quantum self-organizing feature map; quantum machine learning; neural network; quantum unsupervised data clustering; Hamming distance.

[1] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). URL https://doi.org/10.1038/nature14539.

[2] Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **521**, 452–459 (2015). URL https://doi.org/10.1038/nature14541.

[3] Schwab, K. *The Fourth Industrial Revolution* (Currency Press, Redfern, New South Wales, 2017).

[4] Esteva, A. *et al.* A guide to deep learning in healthcare. *Nature Medicine* **25**, 24–29 (2019). URL https://doi.org/10.1038/s41591-018-0316-z.

[5] Tyrsa, V., Carrio, A., Sampedro, C., Rodriguez-Ramos, A. & Campoy, P. A review of deep learning methods and applications for unmanned aerial vehicles. *Journal of Sensors* **2017** (2017). URL https://doi.org/10.1155/2017/3296874.

[6] Kaggle. higgs boson machine learning challenge. (2014).

[7] Radovic, A. *et al.* Machine learning at the energy and intensity frontiers of particle physics. *Nature* **560**, 41–48 (2018). URL https://doi.org/10.1038/s41586-018-0361-2.

[8] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018). URL https://doi.org/10.1038/s41586-018-0337-2.

[9] Marcus, G. Deep learning: A critical appraisal. *arxiv: 1801.00631* (2018).

[10] Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. S. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE* **105**, 2295–2329 (2017).

[11] Kourtis, K. *et al.* Compiling neural networks for a computational memory accelerator. *arxiv: 2003.04293* (2020).

[12] Kohonen, T. The self-organizing map. *Proceedings of the IEEE* **78**, 1464–1480 (1990).

[13] Kohonen, T., Oja, E., Simula, O., Visa, A. & Kangas, J. Engineering applications of the self-organizing map. *Proceedings of the IEEE* **84**, 1358–1384 (1996).

[14] Kohonen, T. Exploration of very large databases by self-organizing maps. In *Proceedings of International Conference on Neural Networks* (IEEE, 1997).

[15] Vilibic, I. *et al.* Self-organizing maps-based ocean currents forecasting system. *Scientific Reports* **6**, 22924 (2016). URL https://doi.org/10.1038/srep22924.

[16] Deboeck, G. & Kohonen, T. *Visual Explorations in Finance* (Springer-Verlag London, Reading, Massachusetts, 1998).

[17] Doszkocs, T. E., Reggia, J. & Lin, X. Connectionist models and information retrieval. annual review of information science and technology. *ARIST* **25**, 209–260. (1990).

[18] Jones, F. C. *et al.* The genomic basis of adaptive evolution in threespine sticklebacks. *Nature* **484**, 55–61 (2012).

[19] Mori, T., Takaoka, H., Yamane, J., Alev, C. & Fujibuchi, W. Novel computational model of gastrula morphogenesis to identify spatial discriminator genes by self-organizing map (som) clustering. *Scientific Reports* **9** (2019).

[20] Corsello, S. M. *et al.* The drug repurposing hub: a next-generation drug library and information resource. *Nature Medicine* **23**, 405–408 (2017).

[21] Zhu, D., Cao, X., Sun, B. & Luo, C. Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system. *IEEE Transactions on Cognitive and Developmental Systems* **10**, 304–313 (2018).

[22] Chea, R., Grenouillet, G. & Lek, S. Evidence of water quality degradation in lower mekong basin revealed by self-organizing map. *PLOS ONE* **11** (2016).

[23] Kiviluoto, K. Topology preservation in self-organizing maps. In *Proceedings of International Conference on Neural Networks* (IEEE, 1996).

[24] Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics* **81**, 074001 (2018).

[25] Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017).

[26] Schuld, M., Sinayskiy, I. & Petruccione, F. An introduction to quantum machine learning. *Contemporary Physics* **56**, 172–185 (2014).

[27] Carleo, G. *et al.* Machine learning and the physical sciences. *Reviews of Modern Physics* **91** (2019).

[28] Wiebe, N., Braun, D. & Lloyd, S. Quantum algorithm for data fitting. *Physical Review Letters* **109** (2012).

[29] Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Physical Review Letters* **103** (2009).

[30] Childs, A. M., Kothari, R. & Somma, R. D. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing* **46**, 1920–1950 (2017).

[31] Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv: 1307.0411* (2013).

[32] Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nature Physics* **10**, 631–633 (2014).

[33] Dunjko, V., Taylor, J. M. & Briegel, H. J. Quantum-enhanced machine learning. *Phys. Rev. Lett.* **117**, 130501 (2016). URL https://link.aps.org/doi/10.1103/PhysRevLett.117.130501.

[34] Paparo, G. D., Dunjko, V., Makmal, A., Martin-Delgado, M. A. & Briegel, H. J. Quantum speedup for active learning agents. *Phys. Rev. X* **4**, 031002 (2014). URL https://link.aps.org/doi/10.1103/PhysRevX.4.031002.

[35] Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Physical Review Letters* **113** (2014).

[36] Kamruzzaman, A., Alhwaiti, Y., Leider, A. & Tappert, C. C. Quantum deep learning neural networks. In *Lecture Notes in Networks and Systems*, 299–311 (Springer International Publishing, 2019).

[37] Schuld, M., Sinayskiy, I. & Petruccione, F. The quest for a quantum neural network. *Quantum Information Processing* **13**, 2567–2586 (2014).

[38] Jeswal, S. K. & Chakraverty, S. Recent developments and applications in quantum neural network: A review. *Archives of Computational Methods in Engineering* **26**, 793–807 (2019). URL https://doi.org/10.1007/s11831-018-9269-0.

[39] Lewenstein, M. Quantum perceptrons. *Journal of Modern Optics* **41**, 2491–2501 (1994).

[40] Allcock, J., Hsieh, C.-Y., Kerenidis, I. & Zhang, S. Quantum algorithms for feedforward neural networks. *arXiv:1812.03089* (2018).

[41] Tacchino, F. *et al.* Quantum implementation of an artificial feed-forward neural network. *arXiv: 1912.12486* (2019).

[42] Zhao, Z., Pozas-Kerstjens, A., Rebentrost, P. & Wittek, P. Bayesian deep learning on a quantum computer. *Quantum Machine Intelligence* **1**, 41–51 (2019).

[43] Cong, I., Choi, S. & Lukin, M. D. Quantum convolutional neural networks. *Nature Physics* **15**, 1273–1278 (2019).

[44] Liu, J. *et al.* Hybrid quantum-classical convolutional neural networks. *arXiv: 1911.02998* (2019).

[45] Kerenidis, I. & Luongo, A. Quantum classification of the mnist dataset via slow feature analysis. *arXiv:1805.08837* (2018).

[46] Li, T., Chakrabarti, S. & Wu, X. Sublinear quantum algorithms for training linear and kernel-based classifiers. *arXiv:1904.02276* (2019).

[47] Killoran, N. *et al.* Continuous-variable quantum neural networks. *Physical Review Research* **1** (2019).

[48] Bondarenko, D. & Feldmann, P. Quantum autoencoders to denoise quantum data. *arXiv: 1910.09169* (2019).

[49] Dunjko, V., Taylor, J. M. & Briegel, H. J. Advances in quantum reinforcement learning. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE, 2017).

[50] Nautrup, H. P., Delfosse, N., Dunjko, V., Briegel, H. J. & Friis, N. Optimizing quantum error correction codes with reinforcement learning. *Quantum* **3**, 215 (2019).

[51] Fösel, T., Tighineanu, P., Weiss, T. & Marquardt, F. Reinforcement learning with neural networks for quantum feedback. *Physical Review X* **8** (2018).

[52] Rebentrost, P., Bromley, T. R., Weedbrook, C. & Lloyd, S. Quantum hopfield neural network. *Phys. Rev. A* **98**, 042308 (2018). URL `https://link.aps.org/doi/10.1103/PhysRevA.98.042308`.

[53] Purushothaman, G. & Karayiannis, N. Quantum neural networks (QNNs): inherently fuzzy feedforward neural networks. *IEEE Transactions on Neural Networks* **8**, 679–693 (1997).

[54] Verdon, G. *et al.* Quantum graph neural networks. *arXiv: 1909.12264* (2019).

[55] Cherny, V. V., Byrnes, T. & Pyrkov, A. N. Nontrivial attractors of the perturbed nonlinear schrödinger equation: Applications to associative memory and pattern recognition. *Advanced Quantum Technologies* **2**, 1800087 (2019).

[56] Byrnes, T., Koyama, S., Yan, K. & Yamamoto, Y. Neural networks using two-component bose-einstein condensates. *Scientific Reports* **3** (2013).

[57] Mishra, N. *et al.* Cancer detection using quantum neural networks: A demonstration on a quantum computer. *arXiv: 1911.00504* (2019).

[58] Vinci, W. *et al.* A path towards quantum advantage in training deep generative models with quantum annealers. *arXiv: 1912.02119* (2019).

[59] Lu, S., Duan, L.-M. & Deng, D.-L. Quantum adversarial machine learning. *arXiv: 2001.00030* (2019).

[60] https://www.ibm.com/quantum-computing/. URL `https://www.ibm.com/quantum-computing/`.

[61] Solan, Z. & Ruppin, E. Similarity in perception: A window to brain organization. *Journal of Cognitive Neuroscience* **13**, 18–30 (2001).

[62] Trugenberger, C. A. Probabilistic quantum memories. *Physical Review Letters* **87** (2001).

[63] Asselmeyer-Maluga, Torsten. Quantum computing and the brain: quantum nets, dessins d´enfants and neural networks. *EPJ Web Conf.* **198**, 14 (2019). URL `https://doi.org/10.1051/epjconf/201919800014`.

[64] Daskin, A. A simple quantum neural net with a periodic activation function. In *2018 IEEE International Conference on Systems, Man, and Cybernetics*, 2887–2891 (2018).

[65] Nguyen, N. T. & Kenyon, G. T. Image classification using quantum inference on the d-wave 2x. *2018 IEEE International Conference on Rebooting Computing* (2018). URL `http://dx.doi.org/10.1109/ICRC.2018.8638596`.

[66] Cheng, Y. *et al.* Targeting epigenetic regulators for cancer therapy: mechanisms and advances in clinical trials. *Signal Transduction and Targeted Therapy* **4**, 62 (2019). URL `https://doi.org/10.1038/s41392-019-0095-0`.

[67] Steck, S. E. & Murphy, E. A. Dietary patterns and cancer risk. *Nature Reviews Cancer* (2019). URL `https://doi.org/10.1038/s41568-019-0227-4`.

[68] Ghasemi, S. Cancer's epigenetic drugs: where are they in the cancer medicines? *The Pharmacogenomics Journal* 1473–1150 (2019). URL `https://doi.org/10.1038/s41397-019-0138-5`.

[69] Wang, X. *et al.* Circular rna ttn acts as a mir-432 sponge to facilitate proliferation and differentiation of myoblasts via the igf2/pi3k/akt signaling pathway. *Molecular Therapy - Nucleic Acids* **18**, 966 – 980 (2019). URL `http://www.sciencedirect.com/science/article/pii/S2162253119303294`.

[70] hua Zhou, B. *et al.* Eukaryotic elongation factor 2 is involved in the anticoccidial action of diclazuril in the second-generation merozoites of eimeria tenella. *Veterinary Parasitology* **276**, 108991 (2019). URL `http://www.sciencedirect.com/science/article/pii/S0304401719302729`.

[71] Jiao, L. *et al.* Correlation of cpg methylation of the pdcd1 gene with pd-1 expression on cd8+ t cells and medical laboratory indicators in chronic hepatitis b infection. *The Journal of Gene Medicine* **22**, e3148 (2019). URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jgm.3148`.