Variational learning for quantum artificial neural networks

Francesco Tacchino*§, Panagiotis Kl. Barkoutsos*, Chiara Macchiavello†,
Dario Gerace†, Ivano Tavernelli* and Daniele Bajoni‡

*IBM Quantum, IBM Research – Zurich, 8803 Rüschlikon, Switzerland
†University of Pavia, Department of Physics, via Bassi 6, 27100 Pavia, Italy
‡University of Pavia, Department of Industrial and Information Engineering, via Ferrata 1, 27100 Pavia, Italy
§Email: fta@zurich.ibm.com

Abstract—In the last few years, quantum computing and machine learning fostered rapid developments in their respective areas of application, introducing new perspectives on how information processing systems can be realized and programmed. The rapidly growing field of Quantum Machine Learning aims at bringing together these two ongoing revolutions. Here we first review a series of recent works describing the implementation of artificial neurons and feed-forward neural networks on quantum processors. We then present an original realization of efficient individual quantum nodes based on variational unsampling protocols. While keeping full compatibility with the overall memory-efficient feed-forward architecture, such a construction effectively reduces the quantum circuit depth required to determine the activation probability of single neurons upon input of the relevant data-encoding quantum states. This suggests a viable approach towards the use of quantum neural networks for pattern classification on near-term quantum hardware.

I. INTRODUCTION

In classical machine learning, artificial neurons and neural networks were originally proposed, more than a half century ago, as trainable algorithms for classification and pattern recognition [1], [2]. A few milestone results obtained in subsequent years, such as the backpropagation algorithm [3] and the Universal Approximation Theorem [4], [5], certified the potential of deep feed-forward neural networks as a computational model which nowadays constitutes the cornerstone of many artificial intelligence protocols [6], [7].

In recent years, several attempts were made to link these powerful but computationally intensive applications to the rapidly growing field of quantum computing, see also Ref. [8] for a useful review. The latter holds the promise to achieve relevant advantages with respect to classical machines already in the near term, at least on selected tasks including e.g. chemistry calculations [9], [10], classification and optimization problems [11]. Among the most relevant results obtained in Quantum Machine Learning it is worth mentioning the use of trainable parametrized digital and continuous-variable quantum circuits as a model for quantum neural networks [12]–[21], the realization of quantum Support Vector Machines (qSVMs) [22] working in quantum-enhanced feature spaces [23], [24] and the introduction of quantum versions of artificial neuron models [25]–[31]. However, it is true that very few clear statements have been made concerning the concrete and quantitative achievement of quantum advantage

in machine learning applications, and many challenges still need to be addressed [8], [32], [33].

In this work, we review a recently proposed quantum algorithm implementing the activity of binary-valued artificial neurons for classification purposes. Although formally exact, this algorithm in general requires quite large circuit depth for the analysis of the input classical data. To mitigate for this effect we introduce a variational learning procedure, based on quantum unsampling techniques, aimed at critically reducing the quantum resources required for its realization. By combining memory-efficient encoding schemes and low-depth quantum circuits for the manipulation and analysis of quantum states, the proposed methods, currently at an early stage of investigation, suggest a practical route towards problem-specific instances of quantum computational advantage in machine learning applications.

II. A MODEL OF QUANTUM ARTIFICIAL NEURONS

The simplest formalization of an artificial neuron can be given following the classical model proposed by McCulloch and Pitts [1]. In this scheme, a single node receives a set of binary inputs $\{i_0, \ldots, x_{m-1}\} \in \{-1, 1\}^m$, which can either be signals from other neurons in the network or external data. The computational operation carried out by the artificial neuron consists in first weighting each input by a synapse coefficient $w_j \in \{-1, 1\}$ and then providing a binary output $X \in \{-1, 1\}$ denoting either an active or rest state of the node determined by an integrate-and-fire response

$$X = \begin{cases} 1 & \text{if } \sum_j w_j x_j \geq \theta \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

where θ represents some predefined threshold.

A quantum procedure closely mimicking the functionality of a binary valued McCulloch-Pitts artificial neuron can be designed by exploiting, on one hand, the superposition of computational basis states in quantum registers, and on the other hand the natural non-linear activation behavior provided by quantum measurements. In this section, we will briefly outline a device-independent algorithmic procedure [28] designed to implement such a computational model on a gate-based quantum processor. More explicitly, we show how classical input and weight vectors of size m can be encoded on a quantum

hardware by using only $N = \log_2 m$ qubits [28], [34], [35]. For loading and manipulation of data, we describe a protocol based on the generation of quantum hypergraph states [36]. This exact approach to artificial neuron operations will be used in the main body of this work as a benchmark to assess the performances of approximate variational techniques designed to achieve more favorable scaling properties in the number of logical operations with respect to classical counterparts.

Let $\vec{i}$ and $\vec{w}$ be binary input and weight vectors of the form

$$\vec{i} = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{m-1} \end{pmatrix} \quad \vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix} \tag{2}$$

with $i_j, w_j \in \{-1, 1\}$ and $m = 2^N$. A simple and qubit-effective way of encoding such collections of classical data can be given by making use of the relative quantum phases (i.e. factors ± 1 in our binary case) in balanced superpositions of computational basis states. We then define the states

$$|\psi_i\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j |j\rangle$$
$$|\psi_w\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j |j\rangle \tag{3}$$

where, as usual, we label computational basis states with integers $j \in \{0, \ldots, m-1\}$ corresponding to the decimal representation of the respective binary string. The set of all possible states which can be expressed in the form above is known as the class of hypergraph states [36].

According to Eq. (1), the quantum algorithm must first perform the inner product $\vec{i} \cdot \vec{w}$. It is not difficult to see that, under the encoding scheme of Eq. (3), the inner product between inputs and weights is contained in the overlap [28]

$$\langle \psi_w | \psi_i \rangle = \frac{\vec{w} \cdot \vec{i}}{m} \tag{4}$$

We can explicitly compute such overlap on a quantum register through a sequence of $\vec{i}$- and $\vec{w}$-controlled unitary operations. First, assuming that we operate on a N-qubit quantum register starting in the blank state $|0\rangle^{\otimes N}$, we can load the input-encoding quantum state $|\psi_i\rangle$ by performing a unitary transformation U_i such that

$$U_i |0\rangle^{\otimes N} = |\psi_i\rangle \tag{5}$$

It is important to mention that this preparation step would most effectively be replaced by, e.g., a direct call to a quantum memory [37], or with the supply of data encoding states readily generated in quantum form by quantum sensing devices to be analyzed or classified. It is indeed well know that the interface between classical data and their representation on quantum registers currently constitutes one of the major bottlenecks for Quantum Machine Learning applications [8]. Let now U_w be a unitary operator such that

$$U_w |\psi_w\rangle = |1\rangle^{\otimes N} = |m-1\rangle \tag{6}$$

In principle, any $m \times m$ unitary matrix having the elements of $\vec{w}$ appearing in the in the last row satisfies this condition. If we apply U_w after U_i, the overall N-qubits quantum state becomes

$$U_w |\psi_i\rangle = \sum_{j=0}^{m-1} c_j |j\rangle \equiv |\phi_{i,w}\rangle \tag{7}$$

Using Eq. (6), we then have

$$\langle \psi_w | \psi_i \rangle = \langle \psi_w | U_w^\dagger U_w | \psi_i \rangle = $$
$$= \langle m-1 | \phi_{i,w} \rangle = c_{m-1} \tag{8}$$

We thus see that, as a consequence of the constraints imposed to U_i and U_w, the desired result $\vec{i} \cdot \vec{w} \propto \langle \psi_w | \psi_i \rangle$ is contained up to a normalization factor in the coefficient c_{m-1} of the final state $|\phi_{i,w}\rangle$.

The final step of the algorithm must access the computed input-weight scalar product and determine the activation state of the artificial neuron. In view of constructing a general architecture for feed-forward neural networks [30], it is useful to introduce an ancilla qubit a, initially set in the state $|0\rangle$, on which the $c_{m-1} \propto \langle \psi_w | \psi_i \rangle$ coefficient can be written through a multi-controlled NOT gate, where the role of controls is assigned to the N encoding qubits [28]:

$$|\phi_{i,w}\rangle |0\rangle_a \to \sum_{j=0}^{m-2} c_j |j\rangle |0\rangle_a + c_{m-1} |m-1\rangle |1\rangle_a \tag{9}$$

At this stage, a measurement of qubit a in the computational basis provides a probabilistic non-linear threshold activation behavior, producing the output $|1\rangle_a$ state, interpreted as an active state of the neuron, with probability $|c_{m-1}|^2$. Although this form of the activation function is already sufficient to carry out elementary classification tasks and to realize a logical XOR operation [28], more complex threshold behaviors can in principle be engineered once the information about the inner product is stored on the ancilla [27], [29]. Equivalently, the ancilla can be used, via quantum controlled operations, to pass on the information to other quantum registers encoding successive layers in a feed-forward network architecture [30]. It is worth noticing that directing all the relevant information into the state of a single qubit, besides enabling effective quantum synapses, can be advantageous when implementing the procedure on real hardware on which readout errors constitute a major source of inaccuracy. Nevertheless, multi-controlled NOT operations, which are inherently non-local, can lead to complex decompositions into hardware-native gates especially in the presence of constraints in qubit-qubit connectivity. When operating a single node to carry out simple classification tasks or, as we will do in the following sections, to assess the performances of individual portions of the proposed algorithm, the activation probability of the artificial neuron can then equivalently be extracted directly from the register of N encoding qubits by performing a direct measurement of $|\phi_{i,w}\rangle$ targeting the $|m-1\rangle \equiv |1\rangle^{\otimes N}$ computational basis state.

A. Exact implementation with quantum hypergraph states

A general and exact realization of the unitary transformations U_i and U_w can be designed by using the generation algorithm for quantum hypergraph states [28]. The latter have been extensively studied as useful quantum resources [36], [38], and are formally defined as follows. Given a collection of N vertices V, we call a k-hyper-edge any subset of exactly k vertices. A hypergraph $g_{\leq N} = \{V, E\}$ is then composed of a set V of vertices together with a set E of hyper-edges of any order k, not necessarily uniform. Notice that this definition includes the usual notion of a mathematical graph if $k = 2$ for all (hyper)-edges. To any hypergraph $g_{\leq N}$ we associate a N-qubit quantum hypergraph state via the definition

$$|g_{\leq N}\rangle = \prod_{k=1}^{N} \prod_{\{q_{v_1},\ldots,q_{v_k}\} \in E} C^k Z_{q_{v_1},\ldots,q_{v_k}} |+\rangle^{\otimes N} \qquad (10)$$

where $q_{v_1}, \ldots, q_{v_k}$ are the qubits connected by a k-hyper-edge in E and, with a little abuse of notation, we assume $C^2Z \equiv CZ$ and $C^1Z \equiv Z = R_z(\pi)$. For N qubits there are exactly $\mathcal{N} = 2^{2^N-1}$ different hypergraph states. We can make use of well known preparation strategies for hypergraph states to realize the unitaries U_i and U_w with at most a single N-controlled $C^N Z$ and a collection of p-controlled $C^p Z$ gates with $p < N$. It is worth pointing out already here that such an approach, while optimizing the number of multi-qubit logic gates to be employed, implies a circuit depth which scales linearly in the size of the classical input, i.e. $\mathcal{O}(m) \equiv \mathcal{O}(2^N)$, in the worst case corresponding to a fully connected hypergraph [28].

To describe a possible implementation of U_i, assume once again that the quantum register of N encoding qubits is initially in the blank state $|0\rangle^{\otimes N}$. By applying parallel Hadamard gates ($H^{\otimes N}$) we obtain the state $|+\rangle^{\otimes N}$, corresponding to a hypergraph with no edges. We can then use the target collection of classical inputs $\vec{i}$ as a control for the following iterative procedure:

for $P = 1$ to N **do**
    for $j = 0$ to $m - 1$ **do**
        if ($|j\rangle$ has exactly P qubits in $|1\rangle$ **and** $i_j = -1$) **then**
            Apply $C^P Z$ to those qubits
            Flip the sign of i_k in $\vec{i}$ $\forall k$ such that $|k\rangle$ has the same P qubits in $|1\rangle$
        end if
    end for
end for

Similarly, U_w can be obtained by first performing the routine outlined obove (without the initial parallel Hadamard gates) tailored according to the classical control $\vec{w}$: since all the gates involved in the construction are the inverse of themselves and commute with each other, this step produces a unitary transformation bringing $|\psi_w\rangle$ back to $|+\rangle^{\otimes N}$. The desired transformation U_w is completed by adding parallel $H^{\otimes N}$ and $NOT^{\otimes N}$ gates [28].

III. VARIATIONAL REALIZATION OF A QUANTUM ARTIFICIAL NEURON

Although the implementation of the unitary transformations U_i and U_w outlined above is formally exact and optimizes the number of multi-qubit operations to be performed by leveraging on the correlations between the ± 1 phase factors, the overall requirements in terms of circuit depth pose in general severe limitations to their applicability in near-term quantum devices. Moreover, although with such an approach the encoding and manipulation of classical data is performed in an efficient way with respect to memory resources, the computational cost needed to control the execution of the unitary transformations and to actually perform the sequences of quantum logic gates remains bounded by the corresponding classical limits. The aim of this section is therefore to explore conditions under which some of the operations introduced in our quantum model of artificial neurons can be obtained in more efficient ways by exploiting the natural capabilities of quantum processors.

In the following, we will mostly concentrate on the task of realizing approximate versions of the weight unitary U_w with significantly lower implementation requirements in terms of circuit depth. Although most of the techniques that we will introduce below could in principle work equally well for the preparation of encoding states $|\psi_i\rangle$, it is important to stress already at this stage that such approaches cannot be interpreted as a way of solving the long standing issue represented by the loading of classical data into a quantum register. Instead, they are pursued here as an efficient way of *analyzing* classical or quantum data presented in the form of a quantum state. Indeed, the variational approach proposed here requires ad-hoc training for every choice of the target vector $\vec{w}$ whose U_w needs to be realized. To this purpose, we require access to many copies of the desired $|\psi_w\rangle$ state, essentially representing a quantum training set for our artificial neuron. As in our formulation a single node characterized by weight connections $\vec{w}$ can be used as an elementary classifier recognizing input data sufficiently close to $\vec{w}$ itself [28], the variational procedure presented here essentially serves the double purpose of training the classifier upon input of positive examples $|\psi_w\rangle$ and of finding an efficient quantum realization of such state analyzer.

A. Global variational training

According to Eq. (6), the purpose of the transformation U_w within the quantum artificial neuron implementation is essentially to reverse the preparation of a non-trivial quantum state $|\psi_w\rangle$ back to the relatively simple product state $|1\rangle^{\otimes N}$. Notice that in general the qubits in the state $|\psi_w\rangle$ share multipartite entanglement [38]. Here we discuss a promising strategy for the efficient approximation of the desired transformation satisfying the necessary constraints based on variational techniques. Inspired by the well known variational quantum eigensolver (VQE) algorithm [39], and in line with a recently introduced unsampling protocol [40], we define the following optimization problem: given access to independent copies of $|\psi_w\rangle$ and to a variational quantum circuit,
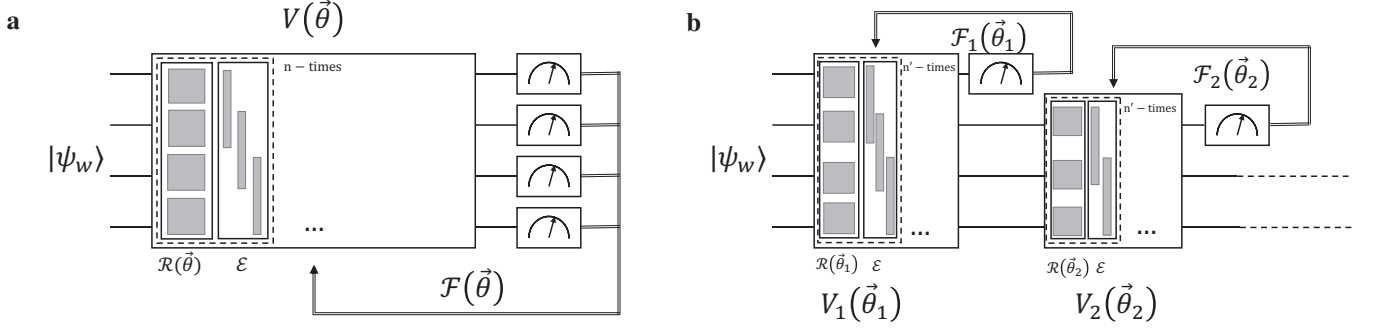
Fig. 1. Variational learning via unsampling. (a) Global strategy, with optimization targeting all qubits simultaneously. (b) Local qubit-by-qubit approach, in which each layer is used to optimize the operation for one qubit at a time.

characterized by a unitary operation $V(\vec{\theta})$ and parametrized by a set of angles $\vec{\theta}$, we wish to find a set of values $\vec{\theta}_{opt}$ that guarantees a good approximation of U_w. The heuristic circuit implementation typically consists of sequential blocks of single-qubit rotations followed by entangling gates, repeated up to a certain number that guarantees enough freedom for the convergence to the desired unitary [9].

Once the solution $V(\vec{\theta}_{opt})$ is found, which in our setup corresponds to a fully trained artificial neuron, it would then provide a form of quantum advantage in the analysis of arbitrary input states $|\psi_i\rangle$ as long as the circuit depth for the implementation of the variational ansatz is sub-linear in the dimension of the classical data, i.e. sub-exponential in the size of the qubit register. As it is customarily done in near-term VQE applications, the optimization landscape is explored by combining executions of quantum circuits with classical feedback mechanisms for the update of the $\vec{\theta}$ angles. In the most general scenario, and according to Eq. (6), a cost function can be defined as

$$\mathcal{F}(\vec{\theta}) = 1 - |\langle 11\ldots 1|V(\vec{\theta})|\psi_w\rangle|^2 \qquad (11)$$

The solution $\vec{\theta}_{opt}$ is then represented by

$$\vec{\theta}_{opt} = \arg\min_{\vec{\theta}} \mathcal{F}(\vec{\theta}) \qquad (12)$$

and leads to $V(\vec{\theta}_{opt}) \simeq U_w$. We call this approach a *global* variational unsampling as the cost function in Eq. (11) requires all qubits to be simultaneously found as close as possible to their respective target state $|1\rangle$, without making explicit use of the product structure of the desired output state $|1\rangle^{\otimes N}$. It is indeed well known that VQE can lead in general to exponentially difficult optimization problems [14], however the characteristic feature of the problem under evaluation may actually allow for a less complex implementation of the VQE for unsampling purposes [40], as outlined in the following section. A schematic representation of the global variational training is provided in Fig. 1a.

B. Local variational training

An alternative approach to the global unsampling task, particularly suited for the case we are considering in which

the desired final state of the quantum register is a fully unentangled product state, makes use of a local, qubit-by-qubit procedure. This technique, which was recently proposed and tested on a photonic platform as a route towards efficient certification of quantum processors [40], is highlighted here as an additional useful tool within a general quantum machine learning setting.

In the local variational unsampling scheme, the global transformation $V(\vec{\theta})$ is divided into successive layers $V_j(\vec{\theta}_j)$ of decreasing complexity and size. Each layer is trained separately, in a serial fashion, according to a cost function which only involves fidelity of a single qubit to its desired final state. More explicitly, every $V_j(\vec{\theta}_j)$ operates on qubits $j,\ldots,N$ and has an associated cost function

$$\mathcal{F}_j(\vec{\theta}_j) = 1 - \langle 1|\operatorname{Tr}_{j+1,\ldots,N}[\rho_j]|1\rangle \qquad (13)$$

where the partial trace leaves only the degrees of freedom associated to the j-th qubit and, recursively, we define

$$\rho_j = \begin{cases} V_j(\vec{\theta}_j)\rho_{j-1}V_j^\dagger(\vec{\theta}_j) & j > 1 \\ |\psi_w\rangle\langle\psi_w| & j = 0 \end{cases} \qquad (14)$$

At step j, it is implicitly assumed that all the parameters $\vec{\theta}_k$ for $k = 1,\ldots,j-1$ are fixed to the optimal values obtained by the minimization of the cost functions in the previous steps. Notice that, operationally, the evaluation of the cost function $\mathcal{F}_j$ can be automatically carried out by measuring the j-th qubit in the computational basis while ignoring the rest of the quantum register, as shown in Fig. 1b.

The benefits of local variational unsampling with respect to the global strategy are mainly associated to the reduced complexity of the optimization landscape per step. Indeed, the local version always operates on the overlap between single-qubit states, at the relatively modest cost of adding $N-1$ smaller and smaller variational ansatzes. In the specific problem at study, we thus envision the local approach to become particularly effective, and more advantageous than the global one, in the limit of large enough number of qubits, i.e. for the most interesting regime where the size of the quantum register, and therefore of the quantum computation, exceeds the current classical simulation capabilities.
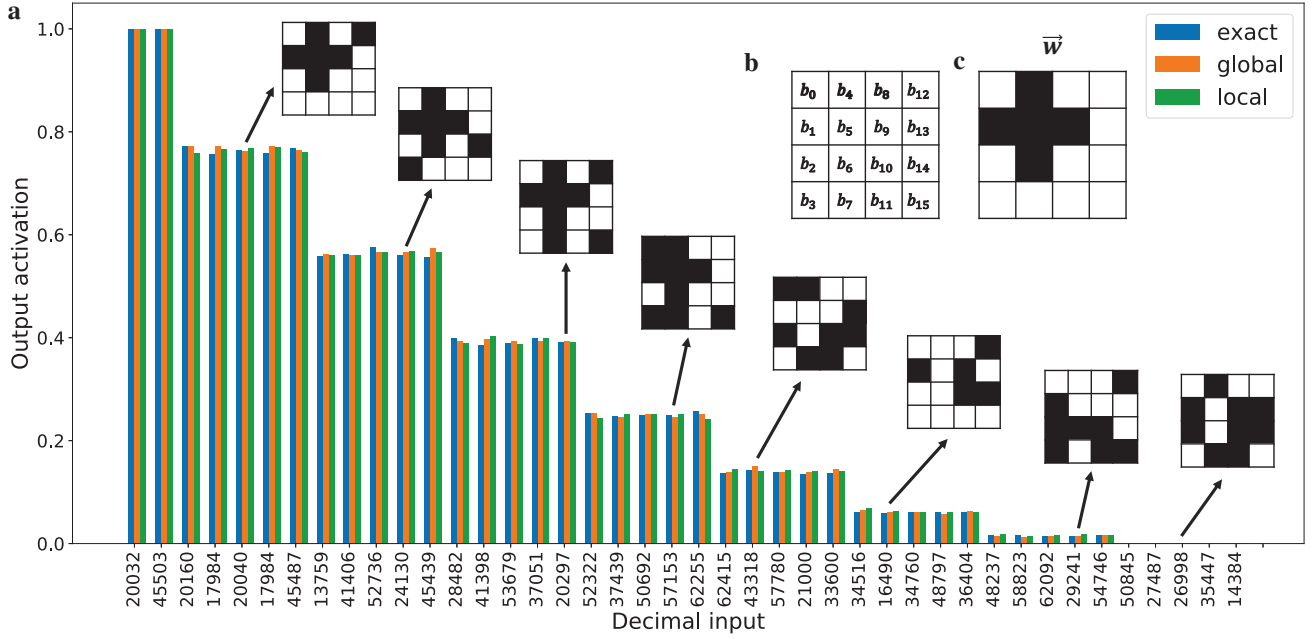
Fig. 2. Comparison of output activation $p_{out} = |\langle \psi_w | \psi_i \rangle|^2$ between the exact (hypergraph states routine), global ($n = 3$) and local ($n' = 2$) approximate implementations of U_w. The inset shows the general mapping of any 16-dimensional binary vector $\vec{b}$ onto the 4×4 binary image (b) and the cross-shaped $\vec{w}$ used in this example (c). The selected inputs on which the approximations are tested were chosen to cover all the possible cases for p_{out}, and are labeled with their corresponding integer k_i (see main text).

C. Case study: pattern recognition

To show an explicit example of the proposed construction, let us fix $m = 16$, $N = 4$. Following Ref. [28], we can visualize a 16-bit binary vector $\vec{b}$, see Eq. (2), as a 4×4 binary pattern of black ($b_j = -1$) and white ($b_j = 1$) pixels. Moreover, we can assign to every possible pattern an integer label k_b corresponding to the conversion of the binary string $\mathbf{k}_b = \mathbf{b}_0 \ldots \mathbf{b}_{15}$, where $b_j = (-1)^{\mathbf{b}_j}$. We choose as our target $\vec{w}$ the vector corresponding to $k_w = 20032$, which represents a black cross on white background at the north-west corner of the 16-bit image, see Fig 2.

Starting from the global variational strategy, we construct a parametrized ansatz for $V(\vec{\theta})$ as a series of entangling ($\mathcal{E}$) and rotation ($\mathcal{R}(\vec{\theta})$) cycles:

$$V(\vec{\theta}) = \left(\prod_{c=1}^{n} \mathcal{R}(\theta_{c,1} \ldots \theta_{c,4}) \mathcal{E} \right) \mathcal{R}(\theta_{0,1} \ldots \theta_{0,4}) \quad (15)$$

where n is the total number of cycles which in principle can be varied to increase the expressivity of the ansatz by increasing its total depth. Rotations are assumed to be acting independently on the $N = 4$ qubits according to

$$\mathcal{R}(\theta_{c,1} \ldots \theta_{c,4}) = \bigotimes_{q=1}^{4} \exp\left(-i \frac{\theta_{c,q}}{2} \sigma_y^{(q)} \right) \quad (16)$$

where $\sigma_y^{(q)}$ is the Pauli y-matrix acting on qubit q. At the same time, the entangling parts promote all-to-all interactions

between the qubits according to

$$\mathcal{E} = \prod_{q} \prod_{q'=q+1}^{4} \mathrm{CNOT}_{qq'} \quad (17)$$

where $\mathrm{CNOT}_{qq'}$ is the usual controlled NOT operation between control qubit q and target q' acting on the space of all 4-qubits. For n cycles, the total number n_θ of θ-parameters, including the initial rotation layer $\mathcal{R}(\theta_{0,1} \ldots \theta_{0,4})$, is therefore $n_\theta = 4 + 4n$.

A qubit-by-qubit version of the ansatz can be constructed in a similar way by using the same structure of entangling and rotation cycles, decreasing the total number of qubits by one after each layer of the optimization. We choose a uniform number n' of cycles per qubit, thus setting $\forall j \neq 4$

$$V_j(\vec{\theta}_j) = \left(\prod_{c=1}^{n'} \mathcal{R}(\theta_{c,j} \ldots \theta_{c,4}) \mathcal{E} \right) \mathcal{R}(\theta_{0,j} \ldots \theta_{0,4}) \quad (18)$$

For $j = 4$, we add a single general single-qubit rotation with three parameters

$$V_4(\alpha, \beta, \gamma) = \exp\left(-i \frac{(\alpha, \beta, \gamma) \cdot \vec{\sigma}^{(4)}}{2} \right) \quad (19)$$

where $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ are again the usual Pauli matrices.

We implemented both versions of the variational training in Qiskit [41], combining exact simulation of the quantum circuits required to evaluate the cost function with classical Nelder-Mead [42] and Cobyla [43] optimizers from the scipy Python library. We find that the values $n = 3$ and $n' = 2$ allow
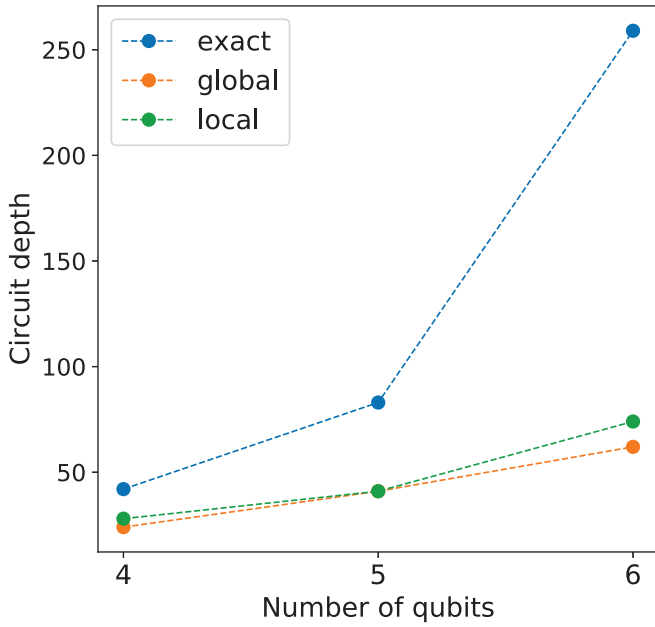
Fig. 3. Scaling of circuit depth for the implementation of U_w computed with Qiskit. The number of ansatz cycles used for both the global (n) and local/qubit-by-qubit (n') variational constructions are increased with the number of qubits up to the minimum value guaranteeing a fidelity of the approximations above 99%. For $N = 5, 6$ the convergence criterion for the classical optimizer was set to 10^{-4}.

the routine to reach total fidelities to the target state $|1\rangle^{\otimes N}$ well above 99.99%. As shown in Fig 2, this in turn guarantees a correct reproduction of the exact activation probabilities of the quantum artificial neuron with a quantum circuit depth of 19 (29) for the global (qubit-by-qubit) strategy, as compared to the total depth equal to 49 for the exact implementation of U_w using hypergraph states. This counting does not include the gate operations required to prepare the input state, i.e. it only evidences the different realizations of the U_w implementation assuming that each $|\psi_i\rangle$ is provided already in the form of a wavefunction. Moreover, the multi-controlled $C^P Z$ operations appearing in the exact version were decomposed into single-qubit rotations and CNOTs without the use of additional working qubits. Notice that these conditions are the ones usually met in real near-term superconducting hardware endowed with a fixed set of universal operations.

We tested the scaling properties of our proposed approach by increasing the size of the qubit register up to $N = 6$. The target $\vec{w}$ was extended to 32 and 64 bit ($N = 5$ and $N = 6$, respectively) by appending the required extra 0s in front of the binary representation k_w. The promising results, reported in Fig. 3, show the significant advantage brought by the variational approach in terms of total circuit depth compared to the exponential increase of complexity required by the exact formulation of the algorithm.

IV. CONCLUSIONS

In this work, we reviewed an exact model for the implementation of artificial neurons on a quantum processor and we

introduced variational training methods for efficiently handling the manipulation of classical and quantum input data. Our work suggests that quantum unsampling techniques represent a useful resource, upon input of quantum training sets, to be integrated in quantum machine learning applications. From a theoretical perspective, our proposed procedure allows for an explicit and direct quantification of possible quantum computational advantages for classification tasks. It is also worth pointing out that such a scheme remains fully compatible with recently introduced architectures for quantum feed-forward neural networks [30], which are needed in general to deploy e.g. complex convolutional filters. Moreover, although the interpretation of quantum hypergraph states as memory-efficient carriers of classical information guarantees an optimal use of the available dimension of a N-qubit Hilbert space, the variational techniques introduced here can in principle be used to learn different encoding schemes designed, e.g., to include continuous-valued features or to improve the separability of the data to be classified [23], [24], [44]. In all envisioned applications, our proposed protocols are intended as an effective method for the analysis of quantum states as provided, e.g., by external devices or sensors, while it is worth stressing that the general problem of efficiently loading classical data into quantum registers still stands open. Finally, on a more practical level, a successful implementation on near-term quantum hardware of the variational learning algorithm introduced in this work will necessarily rely on a deeper analysis of the impact of realistic noise effects both on the training procedure and on the final optimized circuit. In particular, we anticipate that the reduced circuit depth produced via the proposed method could critically lessen the quality requirements for quantum hardware, eventually leading to meaningful implementation of quantum neural networks within the near-term regime.

REFERENCES

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec 1943.
[2] F. Rosenblatt, "The Perceptron: A perceiving and recognizing automaton," Cornell Areonautical Laboratory, Inc., Tech. Rep. 85-460-1, 1957.
[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
[4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989.
[5] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, Jan. 1991.
[6] J. M. Zurada, *Introduction to Artificial Neural Systems*. West Group, 1992.
[7] R. Rojas, *Neural Networks: A Systematic Introduction*. Springer, 1996.

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.

[9] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, Sep. 2017.

[10] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, "Quantum Chemistry in the Age of Quantum Computing," *Chemical Reviews*, vol. 119, no. 19, pp. 10 856–10 915, Oct. 2019, publisher: American Chemical Society.

[11] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, no. 3, p. 030503, Jul. 2018.

[12] E. Farhi and H. Neven, "Classification with Quantum Neural Networks on Near Term Processors," *arXiv:1802.06002 [quant-ph]*, Feb. 2018.

[13] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, "Hierarchical quantum classifiers," *npj Quantum Information*, vol. 4, no. 1, p. 65, Dec. 2018.

[14] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, no. 1, p. 4812, Dec. 2018.

[15] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Phys. Rev. Research*, vol. 1, p. 033063, Oct 2019.

[16] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, Nov. 2019, publisher: IOP Publishing.

[17] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, "Transfer learning in hybrid classical-quantum neural networks," *Quantum*, vol. 4, p. 340, 2020.

[18] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, pp. 1273–1278, Aug. 2019.

[19] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Phys. Rev. A*, vol. 101, p. 032308, Mar 2020.

[20] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: powering image recognition with quantum circuits," *Quantum Machine Intelligence*, vol. 2, no. 1, pp. 1–9, Jun. 2020.

[21] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, M. Leib, A. Skolik, M. Streif, D. Von Dollen, J. R. McClean, S. Boixo, D. Bacon, A. K. Ho, H. Neven, and M. Mohseni, "TensorFlow Quantum: A Software Framework for Quantum Machine Learning," *arXiv:2003.02989 [cond-mat, physics:quant-ph]*, Mar. 2020.

[22] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum Support Vector Machine for Big Data Classification," *Physical Review Letters*, vol. 113, no. 13, p. 130503, Sep. 2014.

[23] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019.

[24] M. Schuld and N. Killoran, "Quantum Machine Learning in Feature Hilbert Spaces," *Physical Review Letters*, vol. 122, no. 4, p. 040504, Feb. 2019.

[25] M. Schuld, I. Sinayskiy, and F. Petruccione, "Simulating a perceptron on a quantum computer," *Physics Letters A*, vol. 379, no. 7, pp. 660–663, Mar. 2015.

[26] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum Perceptron Models," *arXiv:1602.04799 [quant-ph, stat]*, Feb. 2016.

[27] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, "Quantum Neuron: an elementary building block for machine learning on quantum computers," *arXiv:1711.11240 [quant-ph]*, Nov. 2017.

[28] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, "An artificial neuron implemented on an actual quantum processor," *npj Quantum Information*, vol. 5, p. 26, 2019.

[29] E. Torrontegui and J. J. Garcia-Ripoll, "Unitary quantum perceptron as efficient universal approximator," *EPL (Europhysics Letters)*, vol. 125, no. 3, p. 30004, Mar. 2019.

[30] F. Tacchino, P. Barkoutsos, C. Macchiavello, I. Tavernelli, D. Gerace, and D. Bajoni, "Quantum implementation of an artificial feed-forward neural network," *Quantum Science and Technology*, vol. 5, no. 4, p. 044010, Oct 2020.

[31] L. B. Kristensen, M. Degroote, P. Wittek, A. Aspuru-Guzik, and N. T. Zinner, "An Artificial Spiking Quantum Neuron," *arXiv:1907.06269 [cond-mat, physics:quant-ph]*, Jul. 2019.

[32] L. G. Wright and P. L. McMahon, "The Capacity of Quantum Neural Networks," *arXiv:1908.01364 [physics, physics:quant-ph]*, Aug. 2019.

[33] N.-H. Chia, A. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang, "Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning," *arXiv:1910.06151 [quant-ph]*, Oct. 2019.

[34] M. Schuld, M. Fingerhuth, and F. Petruccione, "Implementing a distance-based classifier with a quantum interference circuit," *EPL (Europhysics Letters)*, vol. 119, no. 6, Sep. 2017.

[35] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd, "Quantum Hopfield neural network," *Physical Review A*, vol. 98, no. 4, p. 042308, Oct. 2018.

[36] M. Rossi, M. Huber, D. Bruß, and C. Macchiavello, "Quantum hypergraph states," *New Journal of Physics*, vol. 15, no. 11, p. 113022, Nov. 2013.

[37] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum Random Access Memory," *Physical Review Letters*, vol. 100, no. 16, p. 160501, Apr. 2008.

[38] M. Ghio, D. Malpetti, M. Rossi, D. Bruß, and C. Macchiavello, "Multipartite entanglement detection for hypergraph states," *Journal of Physics A: Mathematical and Theoretical*, vol. 51, no. 4, p. 045302, Jan. 2018.

[39] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, p. 4213, Sep. 2014.

[40] J. Carolan, M. Mohseni, J. P. Olson, M. Prabhu, C. Chen, D. Bunandar, M. Y. Niu, N. C. Harris, F. N. C. Wong, M. Hochberg, S. Lloyd, and D. Englund, "Variational quantum unsampling on a quantum photonic processor," *Nature Physics*, vol. 16, p. 322, Jan. 2020.

[41] A. Héctor and al., "Qiskit: An open-source framework for quantum computing," 2019.

[42] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 01 1965.

[43] M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, S. Gomez and J.-P. Hennart, Eds. Dordrecht: Kluwer Academic, 1994, pp. 51–67.

[44] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, "Quantum Fingerprinting," *Physical Review Letters*, vol. 87, no. 16, p. 167902, Sep. 2001.