

PAPER • OPEN ACCESS

Quantum implementation of an artificial feed-forward neural network

To cite this article: Francesco Tacchino *et al* 2020 *Quantum Sci. Technol.* **5** 044010

View the [article online](#) for updates and enhancements.

A promotional banner for Quantum Machines hardware. On the left, the Quantum Machines logo (a stylized 'QM' with a blue dot) is above the text 'The New Standard for Quantum Control Hardware and Software'. Below this is a blue button with the text 'REQUEST A DEMO >'. On the right, a black, rectangular hardware unit is shown against a dark, space-themed background with stars and nebulae. The unit has a 'QM' logo on its top surface and a front panel with several gold-colored connectors. Labels on the front panel include 'Digital Outputs' (with pins 1-8), 'Analog Outputs' (with pins 1-4), and 'Analog Inputs' (with pins 1-4). Colored wires (red, green, blue) are plugged into the connectors. The unit is resting on a small, dark, rocky base.

QM QUANTUM MACHINES

The New Standard for Quantum Control Hardware and Software

REQUEST A DEMO >

Quantum Science and Technology



PAPER

OPEN ACCESS

RECEIVED
14 May 2020

REVISED
25 August 2020

ACCEPTED FOR PUBLICATION
16 September 2020

PUBLISHED
13 October 2020

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Quantum implementation of an artificial feed-forward neural network

Francesco Tacchino^{1,6} , Panagiotis Barkoutsos² , Chiara Macchiavello^{1,3,4} ,
Ivano Tavernelli² , Dario Gerace¹ and Daniele Bajoni⁵

¹ Dipartimento di Fisica, Università di Pavia, via Bassi 6, I-27100, Pavia, Italy

² IBM Quantum, IBM Research—Zurich, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland

³ INFN Sezione di Pavia, via Bassi 6, I-27100, Pavia, Italy

⁴ CNR-INO, largo E. Fermi 6, I-50125, Firenze, Italy

⁵ Dipartimento di Ingegneria Industriale e dell'Informazione, Università di Pavia, via Ferrata 1, I-27100, Pavia, Italy

⁶ Address for correspondence: IBM Quantum, IBM Research—Zurich, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland.

E-mail: francesco.tacchino01@ateneopv.it

Keywords: quantum neural networks, quantum algorithms, near-term quantum processors

Supplementary material for this article is available [online](#)

Abstract

Artificial intelligence algorithms largely build on multi-layered neural networks. Coping with their increasing complexity and memory requirements calls for a paradigmatic change in the way these powerful algorithms are run. Quantum computing promises to solve certain tasks much more efficiently than any classical computing machine, and actual quantum processors are now becoming available through cloud access to perform experiments and testing also outside of research labs. Here we show in practice an experimental realization of an artificial feed-forward neural network implemented on a state-of-art superconducting quantum processor using up to 7 active qubits. The network is made of quantum artificial neurons, which individually display a potential advantage in storage capacity with respect to their classical counterpart, and it is able to carry out an elementary classification task which would be impossible to achieve with a single node. We demonstrate that this network can be equivalently operated either via classical control or in a completely coherent fashion, thus opening the way to hybrid as well as fully quantum solutions for artificial intelligence to be run on near-term intermediate-scale quantum hardware.

1. Introduction

The field of artificial intelligence was revolutionized by moving from the simple, single layer perceptron design [1] to that of a complete feed-forward neural network (ffNN), constituted by several neurons organized in multiple successive layers [2, 3]. In such artificial neural network designs each constituent neuron receives, as inputs, the outputs (activations) from the neurons in the preceding layer. The advantage of ffNNs with respect to simpler designs such as single layer perceptrons or support vector machines is that they can be used to classify data with relations that cannot be reduced to a separating hyperplane [4]. The present ubiquitous use of artificial intelligence in a wide variety of tasks, ranging from pattern or spoken language recognition to the analysis of large data sets, is mostly due to the discovery that such feed-forward networks can be trained by using well established optimization algorithms [2–4].

Quantum computers hold promise to achieve some form of computing advantage over classical counterparts in the not-so-far future [5]. Indeed, quantum computing has been theoretically shown to offer potentially exponential speedups over traditional computing machines, especially in tasks such as large number factoring, solving linear systems of equations, and data classification [6–10]. More recently, quantum computers have been applied to the fields of artificial intelligence and biomimetics [11–15], and implementations of artificial neurons [16–19] and support vector machines [20, 21] on real quantum

processors, even if limited to simple systems at present, have shown a promising route toward a practical realization of such advantage.

In order to harness the full potentialities that quantum computing may offer to the field of artificial intelligence it is necessary to undergo the passage from single layered to deep ffNNs [22–24], which has so greatly expanded the capabilities of artificially intelligent systems to date. Here we propose the architecture of a quantum ffNN and we test it on a state-of-the-art 20-qubit IBM Quantum processor. We start from a hybrid approach combining quantum nodes with classical information feed-forward, obtained via classical control of unitary transformations on qubits. This design realizes a fully general implementation of a ffNN on a quantum processor assisted by classical registers. A minimal three-node example, specifically designed to carry out a pattern recognition task exceeding the capabilities of a single artificial neuron, is used for a proof-of-principle demonstration on real quantum hardware. We then describe and successfully implement on a seven-qubit register an equivalent fully quantum coherent configuration of the same set-up, which does not involve classical control of the feed-forward links and thus potentially opens the way to the exploration of more complex and classically inaccessible regimes.

The proposed quantum implementation of ffNN can be viewed as a first step toward a viable integration of near-term quantum technologies [25] into machine learning applications: in particular, the hybrid nature of the ffNN itself suggests a seamless combination with existing classical structures and algorithms for neural network computation [26].

2. Design of the hybrid feed-forward neural network

In this section, we outline the general structure of our proposed hybrid ffNN, including a short description of the working principles of single nodes and a more detailed discussion of layer-to-layer connections. While, for the sake of clarity, we will often refer to a specific minimal example with three nodes and two layers, the overall scheme can be generalized to arbitrary feed-forward networks.

2.1. Individual nodes

A ffNN is essentially composed of a set of individual nodes $\{n_i\}$, or artificial neurons, arranged in a sequence of successive layers $\{L_j\}$. Information flows through the network in a well defined direction from the input to the output layer, traveling through neuron–neuron connections (i.e. artificial synapses). Each node performs an elementary non-linear operation on the incoming data, whose result is then passed on to one or more nodes in the successive layer.

In their simplest form, individual nodes can be designed to analyze binary-valued inputs. The artificial neurons that we consider here are based on the well known perceptron model [1]: such computational units analyze information by combining input (\vec{i}) and weight (\vec{w}) vectors, providing an activation response that depends on their scalar product $\vec{i} \cdot \vec{w}$. In our case, input and weight vectors are assumed to be binary-valued m -dimensional arrays [27], i.e.

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

where $i_k, w_k \in \{-1, 1\} \forall k \in [0, m-1]$. The activity of a binary artificial neuron can be implemented on a quantum register of $N = \log_2(m)$ qubits [19] by considering the quantum states

$$|\psi_i\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j |j\rangle, \quad |\psi_w\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j |j\rangle. \quad (2)$$

These encode the corresponding input and weight vectors by effectively exploiting the exponential size of the Hilbert space associated to the quantum register in use. The states of the form presented in equation (2) are real equally-weighted (REW) superpositions of all the computational basis states $|j\rangle \in \{|0 \dots 00\rangle; |0 \dots 01\rangle; \dots, |1 \dots 11\rangle\}$. The quantum procedure carrying out the perceptron-like computation for single artificial neurons can be summarized in three steps [19]. First, assuming that the N -qubits quantum register is initially in the idle configuration, $|0\rangle^{\otimes N}$, we prepare the quantum state encoding the input vector with a unitary operation U_i such that $|\psi_i\rangle = U_i |0\rangle^{\otimes N}$. We then apply the weight factors of vector \vec{w} on the input state by implementing another unitary transformation, U_w , subject to the constraint $|1\rangle^{\otimes N} = U_w |\psi_w\rangle$. An optimized yet exact implementation of U_i and U_w exploits the close relationship between REW quantum states and the class of hypergraph states [19, 28], achieving in the worst case an overall computational complexity which is linear in the size of the classical input, i.e. $O(m)$. A

detailed description of the construction of the U_i and U_w transformations and the corresponding quantum circuits for arbitrary \vec{i} and \vec{w} is provided in the supplementary information (<https://stacks.iop.org/QST/05/044010/mmedia>), for completeness. After the two unitaries have been performed, it is easily seen that the state of the quantum register is

$$|\phi_{i,w}\rangle = U_w U_i |0\rangle^{\otimes N} = \sum_{j=0}^{m-1} c_j |j\rangle \quad (3)$$

where $c_{m-1} = \langle \psi_w | \psi_i \rangle = (1/m) \vec{i} \cdot \vec{w}$. Finally, the non-linear activation of the single artificial neuron can be implemented by performing a multi-controlled NOT gate [6] between the encoding register and an ancilla initialized in the initial state $|0\rangle$

$$|\phi_{i,w}\rangle |0\rangle_a \rightarrow \sum_{j=0}^{m-2} c_j |j\rangle |0\rangle_a + c_{m-1} |m-1\rangle |1\rangle_a \quad (4)$$

followed by a final measure of the ancilla in the computational basis. Hence, the output of the quantum artificial neuron is found in the active state $|1\rangle_a$ with probability $p(1) = |c_{m-1}|^2$.

2.2. Information feed-forward

When several copies of the quantum register implementing the artificial neuron model outlined above work in parallel, the respective ancillae, and the result of the measurements performed on them, can be used to feed-forward the information about the input-weight processing to a successive layer. Indeed, let us suppose that a layer L_j contains ℓ_j independent nodes, $\{n_{kj}\}_{k=1}^{\ell_j}$, each of them characterized by a weight vector \vec{w}_{kj} : in one cycle of operation, every node is provided with a classical input \vec{i}_{kj} (either coming from layer L_{j-1} or directly from the original data set to be analyzed) and, upon measurement, it outputs an activation state $a_{kj} \in \{1, 0\}$, chosen according to a probability $p_{kj}(a_{kj} = 1) \propto |\vec{i}_{kj} \cdot \vec{w}_{kj}|^2$. Assuming for simplicity that the h th neuron $n_{h(j+1)}$ belonging to the L_{j+1} layer collects the outputs of all $\{n_{kj}\}$ nodes, the corresponding binary classical input can be constructed as

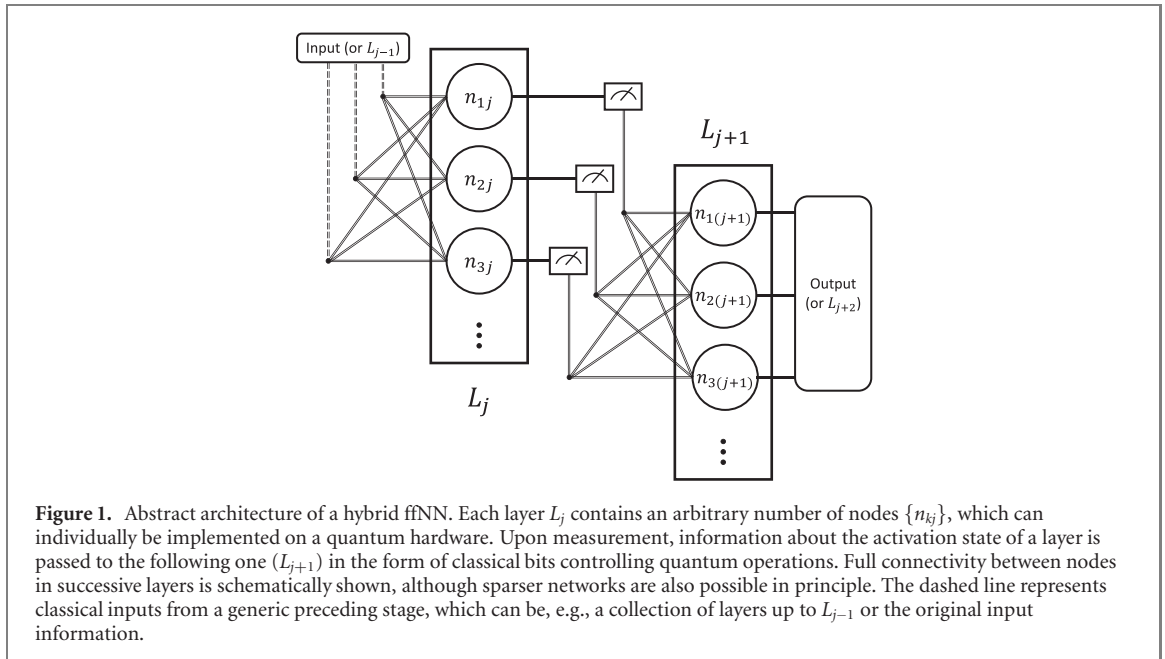
$$\vec{i}_{h(j+1)} = \begin{pmatrix} (-1)^{a_{1j}} \\ (-1)^{a_{2j}} \\ \vdots \\ (-1)^{a_{\ell_j j}} \end{pmatrix}. \quad (5)$$

Such new input vector can then be used to parametrize the appropriate U_i transformation for the $n_{h(j+1)}$ node. The overall computation can then be constructed by iteratively alternating the unitary quantum computation carried out by single layers with non-linear measurement and feed-forward stages. Notice that the design is totally general in terms of the number of nodes in each layer, the number of connections and the size of the various inputs to individual nodes. Moreover, as the information is formally transferred in the form of classical bits, the same input can easily be manipulated, e.g., by making classical copies to be fed to independent nodes sharing similar connections to the previous layer. An abstract representation of the proposed architecture is shown in figure 1.

From the technical point of view, a very natural implementation of the hybrid ffNN architecture onto a quantum processor makes use of classically controlled quantum gates. Independent quantum nodes within the same layer can either be implemented in different quantum registers, and thus computed simultaneously, or run on the same set of qubits, after proper re-initialization and by storing all the observed activation states in different positions of a classical memory register. However, it must also be noted that the classical manipulation of information between layers might limit the role and quantitative advantages brought by the use of quantum mechanical operations, reducing the hybrid architecture to a sequence of shallow quantum convolutional filters. It is therefore interesting to consider, as we will do in section 3, a different scenario in which the network is operated in a fully coherent way.

2.3. Example: pattern recognition

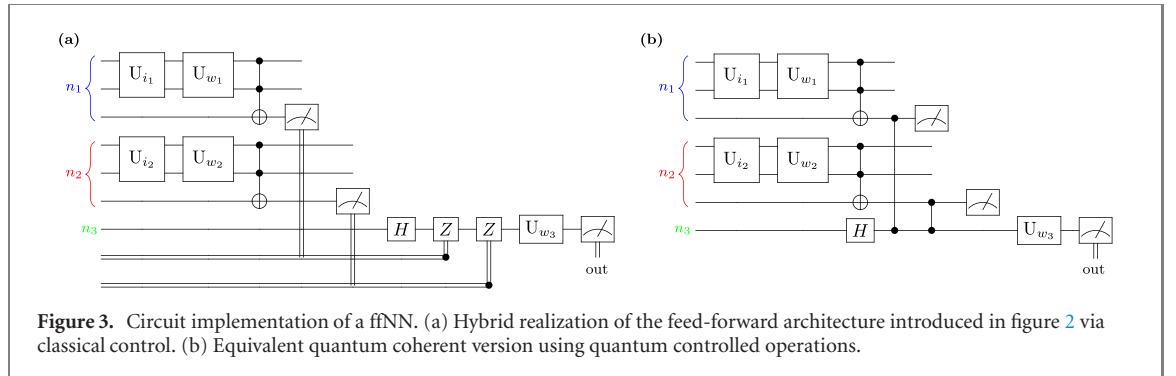
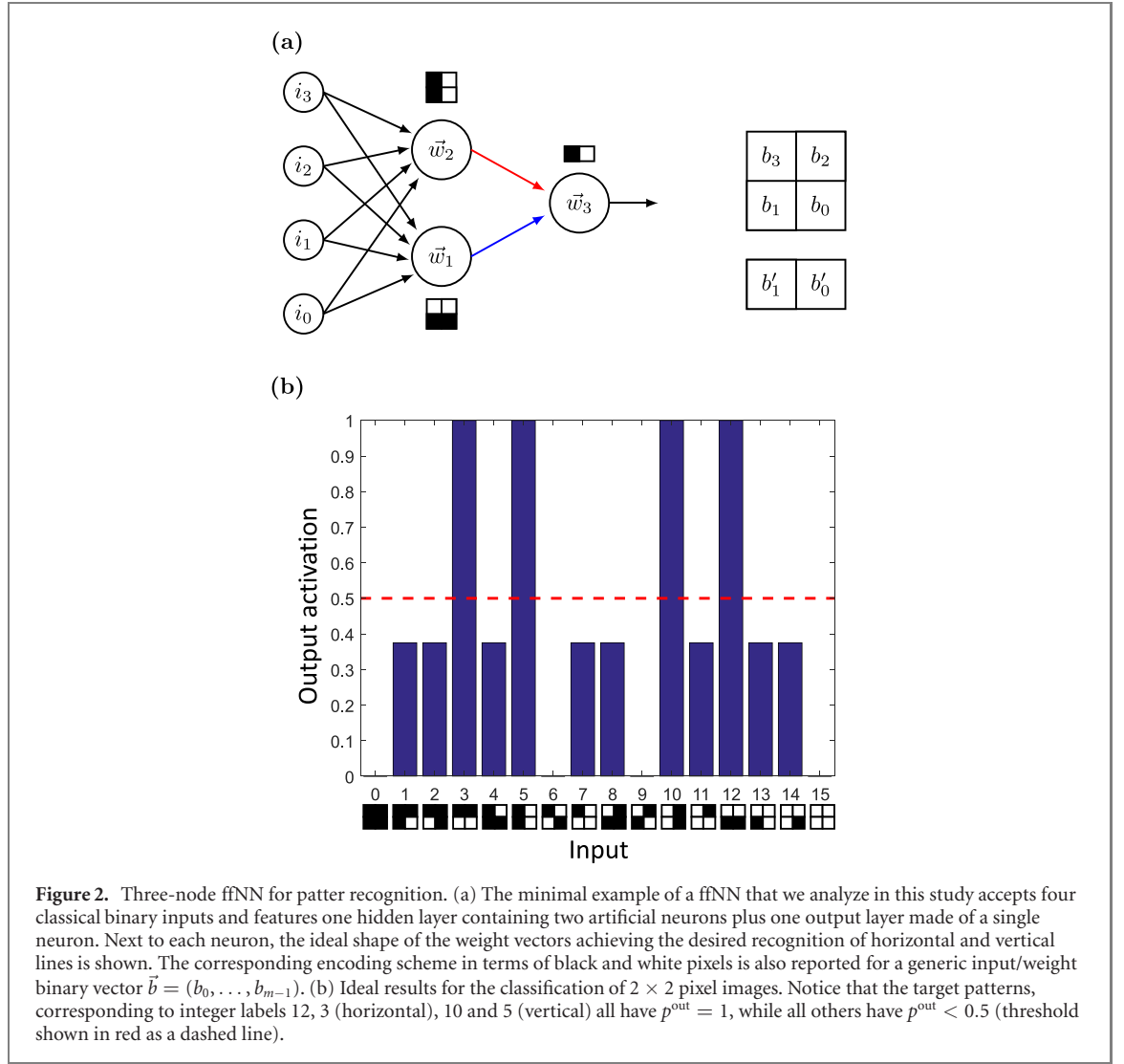
The working principles of our proposed hybrid ffNN, including the above technical details, are actually best clarified by describing an explicit example tailored to solve a well defined elementary classification problem. This will also set the stage for the experimental proof-of-principle demonstration on an actual superconducting quantum hardware to be presented in the next section. First, let us recall that binary input and weight vectors can be visually interpreted as images containing black or white square pixels [19]: a natural encoding scheme associates, e.g., a white spot to a $i_j(w_j) = -1$ entry in the corresponding input



(weight) vector, as shown explicitly in figure 2(a) for the hidden ($m = 4$, i.e. 2×2 pixel images) and output ($m = 2$, i.e. 2×1 pixel images) layers of a minimal ffNN. Moreover, we can identify any such binary pattern with a unique integer label by considering the equivalent decimal representation of the binary number $b_3b_2b_1b_0$ where $b_k = (-1)^{b_k}$, $b_k \in \{0, 1\}$. The task that we set out to solve with our example ffNN is the following: the network should be able to recognize (i.e., give a positive output activation with sufficiently large probability) whether there exist straight lines in 2×2 pixel images, regardless of the fact that the lines are horizontal or vertical. All the other possible input images should be classified as negative. Notice that, as the data vectors encoding horizontal and vertical lines are orthogonal to each other, there is no single hyperplane separating the four positive states from all other possible input images: therefore, the desired classification cannot be carried out by a single node accepting four-bit inputs. This behavior of quantum artificial neurons differs from their usual classical counterparts, which cannot correctly classify sets containing opposite vectors [4]. More explicitly, given an input vector \vec{v}_1 and a weight vector \vec{w} , a single quantum neuron would output a value proportional to $|\vec{v}_1 \cdot \vec{w}|^2$, i.e. $\cos^2 \theta$, where θ is the angle formed by the two vectors. If we take a second input vector $\vec{v}_2 \perp \vec{v}_1$, the output would be upper bounded by $\sin^2 \theta$. As the set of patterns that should yield a positive result includes vectors that are orthogonal (those representing horizontal lines are orthogonal to those representing vertical lines) and vectors that are opposite (for instance, the vector corresponding to a vertical line on the left column of a 2×2 pixel image is opposite to the vector corresponding to a vertical line on the right column), it is therefore impossible to find a weight \vec{w} capable of yielding an output activation larger than 0.5 for all targets in the configuration space. We hereby show that a simple three-node network can accomplish the desired computation. A scheme of such an elementary ffNN is shown in figure 2(a), where the circles indicate individual artificial neurons, and the vectors \vec{w}_i refer to their respective weights. The network features a single hidden layer and a single binary output neuron. On a conceptual level, the working principle of the network can be interpreted as follows: with the *a priori* choice of weights represented in figure 2(a), the top quantum neuron of the hidden layer outputs a high activation if the input vector has vertical lines, while the bottom neuron does the same for the case of horizontal lines. The output neuron in the last layer then recognizes whether one of the neurons in the hidden layer has given a positive outcome.

A possible quantum circuit description of the ffNN introduced above, including the classical feed-forward stage between the hidden and the output layer, is provided in figure 3(a). We assume that each neuron within the hidden layer can accept four-bit inputs, such that each quantum neuron can be represented on a two-qubit encoding register plus an ancilla qubit (i.e., $m = 4$ and $N = 2$ in this case). At the same time, the output neuron takes two-dimensional inputs coming from the previous layer and provides the global activation state of the network, thus requiring a single qubit ($m = 2$, $N = 1$) to be encoded. Classical bits are also included to store the intermediate and final results.

Let us call n_1 and n_2 the two hidden nodes, which actually accept the same classical input but process it in two different ways. As described at the beginning of this section, each artificial neuron will independently provide, upon measurement, an activation pattern $a_k \in \{0, 1\}$ (for $k = 1, 2$), which can be stored in a



classical bit b_k . We denote p_k the probability of actually observing a value $a_k = 1$ from the k th neuron. When such measurement is performed, we set $b_k = a_k$: as a result, the state of the classical two-bit register after the quantum computation in the hidden layer has been completed is one of the following

$$[b_1, b_2] = \begin{cases} [0, 0] \\ [0, 1] \\ [1, 0] \\ [1, 1] \end{cases} \quad (6)$$

with the probability

$$p([b_1, b_2]) = \begin{cases} (1 - p_1)(1 - p_2) \\ (1 - p_1)p_2 \\ p_1(1 - p_2) \\ p_1p_2, \end{cases} \quad (7)$$

respectively. It is easy to see that feed-forwarding the information contained in the classical register to the output neuron n_3 corresponds to providing it with one of the classical binary inputs $\vec{i}_{b_1 b_2}$ reading

$$\begin{aligned} \vec{i}_{00} &= \begin{pmatrix} 1 \\ 1 \end{pmatrix}, & \vec{i}_{01} &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ \vec{i}_{10} &= \begin{pmatrix} -1 \\ 1 \end{pmatrix}, & \vec{i}_{11} &= \begin{pmatrix} -1 \\ -1 \end{pmatrix}. \end{aligned} \quad (8)$$

As shown in figure 3(a), a straightforward strategy for preparing the corresponding $|\psi_i\rangle$ state on the single-qubit register representing n_3 is by first bringing it from the idle state $|0\rangle$ to the superposition $\sqrt{2}|+\rangle = |0\rangle + |1\rangle$ via a Hadamard (H) gate, and then conditioning the application of two Z gates (each of them adds a -1 phase to the $|1\rangle$ component, if applied) on the two classical bits $[b_1, b_2]$. The resulting quantum state will then be

$$|\psi_i\rangle_{n_3} = \frac{1}{\sqrt{2}} (|0\rangle + (-1)^{b_1 \oplus b_2} |1\rangle) \quad (9)$$

where $a \oplus b$ here denotes the usual bit sum modulo 2. If we now choose, as shown in figure 2(a), a weight vector $\vec{w}_3 = (1, -1)$ we obtain (see supplementary information) $U_{w_3} \equiv H$. Therefore, the final state of the third neuron reads

$$|\psi_{\text{out}}\rangle_{n_3} = \begin{cases} |0\rangle & \text{if } b_1 \oplus b_2 = 0 \\ |1\rangle & \text{if } b_1 \oplus b_2 = 1 \end{cases}. \quad (10)$$

The overall probability of observing an active state on the output neuron can thus be written, in general, as

$$p^{\text{out}} = \sum_{[b_1, b_2]} p([b_1, b_2]) p(a_3 = 1 | [b_1, b_2]) \quad (11)$$

where we employed the usual notation for conditional probabilities and

$$p(a_3 = 1 | [b_1, b_2]) = |\langle 1 | \psi_{\text{out}} \rangle_{n_3}|^2. \quad (12)$$

In our specific case, it is easy to see that, given equations (7) and (10), this reduces to

$$p^{\text{out}} = p_1(1 - p_2) + (1 - p_1)p_2. \quad (13)$$

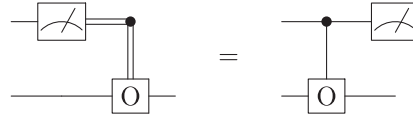
Since in this elementary example n_3 is encoded in a single qubit, the final measurement can be performed directly without the need for an additional ancilla. In figure 2(b) we report the exact result for the convolution of equation (13): as it can be seen, the fNN ideally outputs an active state with $p^{\text{out}} = 1$ for the target horizontal and vertical patterns, while $p^{\text{out}} < 0.5$ in all other cases.

Before moving forward, it is worth mentioning that the construction of a classically conditioned U_i can always be found also in more general cases, e.g. when the hidden layer contains more than two neurons. In particular, any node encoded on N qubits will be able to accept inputs from $m = 2^N$ nodes in the previous layer: indeed, each output configuration from the latter will be one of the 2^m possible bit strings $[b_1, \dots, b_m]$ that can be used to uniquely identify one of the $2^m = 2^{2^N}$ possible input states, and thus to classically program its preparation.

3. Quantum coherent feed-forward

The hybrid feed-forward architecture described so far and realized in a minimal three-node two-layer example can also be reformulated in a fully quantum coherent way. As we will show below, and at difference with the hybrid quantum–classical solution, this version always requires all nodes to be implemented simultaneously on a dedicated quantum register, thus making the quantum computation more demanding. At the same time, however, it reduces the necessity to store and process classical bits during intermediate stages. Moreover, fully coherent quantum neural networks offer more opportunities to be actually used on current quantum processors, as it will be discussed in the final section.

In figure 3(b) we show a fully quantum coherent construction for the ffNN of figure 3(a). The fundamental reason for the actual equivalence lies in the well known principle of deferred measurement [6], stating that in a quantum circuit one can always move a measurement done at an intermediate stage to the end of the computation while replacing classically controlled operations (O) with quantum controlled ones:



Indeed, assuming that the nodes n_1 and n_2 are encoded in parallel and after the operations of the first layer (except for the measurements on the ancillae) have been performed, we can write the global state of the total $(3 + 3 + 1)$ -qubit network as

$$(r_{n_1}|\varphi_{n_1}\rangle|0\rangle_{a_1} + c_{m-1,n_1}|1\dots 1\rangle_{n_1}|1\rangle_{a_1}) \otimes (r_{n_2}|\varphi_{n_2}\rangle|0\rangle_{a_2} + c_{m-1,n_2}|1\dots 1\rangle_{n_2}|1\rangle_{a_2}) \otimes |0\rangle_{n_3} \quad (14)$$

where $r_{n_x} = (1 - c_{m-1,n_x}^2)^{1/2}$ and $|\varphi_{n_x}\rangle$ contains, for each neuron, all the components other than the one leading to activation, see equation (4). Notice that, by construction, $\langle\varphi_{n_x}|1\dots 1\rangle = 0$. In the meantime, the n_3 qubit is brought into the superposition $\sqrt{2}|+\rangle = |0\rangle + |1\rangle$ by applying a single-qubit Hadamard gate, H. Synapses can thereafter be implemented with two CZ gates, as represented in figure 3(b). The overall state of the quantum ffNN then becomes

$$(r_{n_1}r_{n_2}|R_{n_1}\rangle|R_{n_2}\rangle + c_{n_1}c_{n_2}|A_{n_1}\rangle|A_{n_2}\rangle)|+\rangle_{n_3} + (r_{n_1}c_{n_2}|R_{n_1}\rangle|A_{n_2}\rangle + c_{n_1}r_{n_2}|A_{n_1}\rangle|R_{n_2}\rangle)|-\rangle_{n_3} \quad (15)$$

where c_{n_x} is a short-hand notation for c_{m-1,n_x} , and the activated $|A\rangle$ and rest $|R\rangle$ states of n_1 and n_2 are explicitly given as

$$\begin{aligned} |A_{n_x}\rangle &= |1\dots 1\rangle_{n_x}|1\rangle_{a_x} \\ |R_{n_x}\rangle &= |\varphi_{n_x}\rangle|0\rangle_{a_x}. \end{aligned} \quad (16)$$

By applying $U_{w_3} \equiv H$ on n_3 we explicitly obtain an output state

$$|\psi_{\text{out}}\rangle = (r_{n_1}r_{n_2}|R_{n_1}\rangle|R_{n_2}\rangle + c_{n_1}c_{n_2}|A_{n_1}\rangle|A_{n_2}\rangle)|0\rangle_{n_3} + (r_{n_1}c_{n_2}|R_{n_1}\rangle|A_{n_2}\rangle + c_{n_1}r_{n_2}|A_{n_1}\rangle|R_{n_2}\rangle)|1\rangle_{n_3}. \quad (17)$$

It is straightforward to observe at this point that the neurons of the hidden layer can in principle be measured in an activation state $[b_1, b_2] \in \{[0, 0], [0, 1], [1, 0], [1, 1]\}$ with probabilities

$$p([b_1, b_2]) = \begin{cases} |r_{n_1}|^2|r_{n_2}|^2 = (1 - p_1)(1 - p_2) \\ |r_{n_1}|^2|c_{n_2}|^2 = (1 - p_1)p_2 \\ |c_{n_1}|^2|r_{n_2}|^2 = p_1(1 - p_2) \\ |c_{n_1}|^2|c_{n_2}|^2 = p_1p_2, \end{cases} \quad (18)$$

which exactly correspond to the ones reported in equation (7). However, as long as we are interested only in the output state of the network, i.e. the activation state a_3 of n_3 , there is no need to actually perform the final measurements on n_1 and n_2 : similarly to equation (11), we can in fact simply discard the information contained in the variables pertaining to the hidden layer by performing a partial trace operation. This returns a density matrix for the output neuron

$$\rho_{n_3} = \text{Tr}_{\{n_1, n_2\}} [|\psi_{\text{out}}\rangle\langle\psi_{\text{out}}|] = \begin{pmatrix} 1 - p^{\text{out}} & 0 \\ 0 & p^{\text{out}} \end{pmatrix}, \quad (19)$$

which automatically represents the convolution of the hidden nodes, see equation (13). It is worth noticing that the role of the partial trace operation has recently been recognized and extensively discussed in the literature as a possible ingredient for a more general theory of quantum neural networks [29, 30].

To conclude this section, we also point out that the conversion between the two modes of operation (hybrid vs coherent) of our proposed ffNN architecture goes beyond the specific example presented in this work. Indeed, as mentioned at the end of section 2.3, any feed-forward link between successive layers can in general be decomposed in terms of classically controlled operations. Whenever such construction is known, measurement deferral and partial traces can in principle always be employed to obtain the equivalent coherent network, namely by replacing all classical controls with their quantum counterparts and by measuring only the output layer.

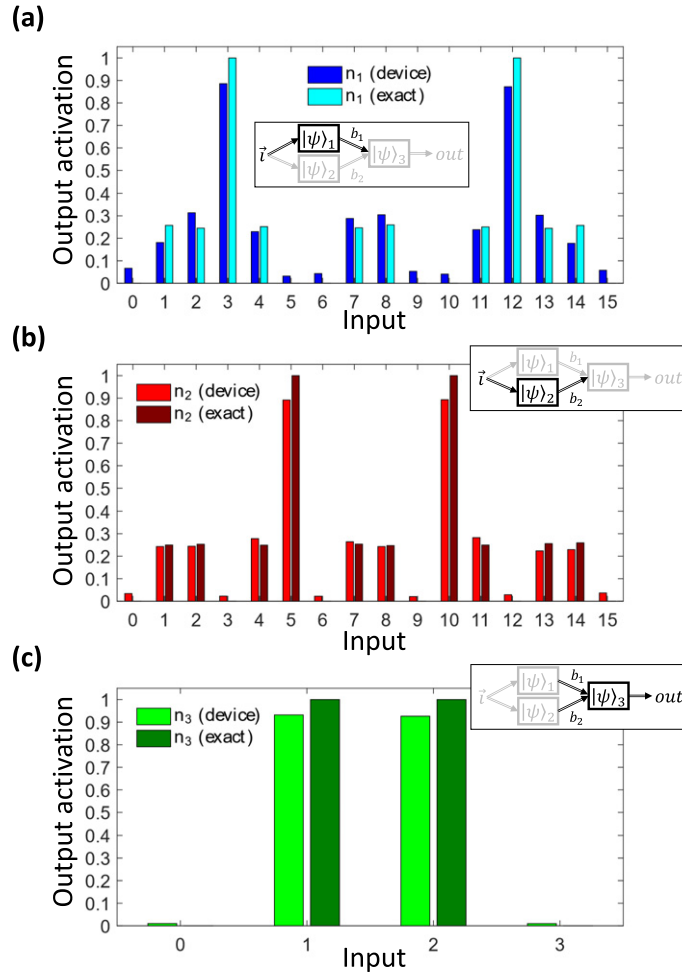


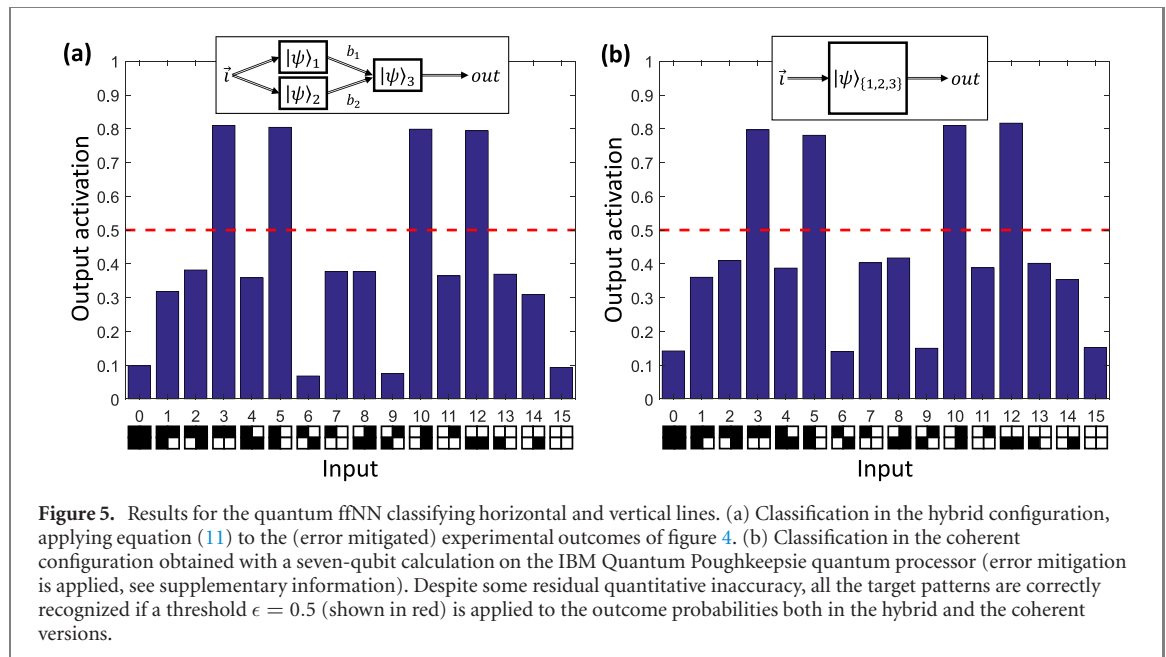
Figure 4. Experimental realization of single nodes on quantum hardware. Single artificial neurons of the ffNN introduced in figure 3(a) implemented on the IBM Quantum Poughkeepsie superconducting processor and compared with ideal noiseless outcomes computed numerically with the Qiskit `qasm_simulator`. (a) Neuron n_1 , recognizing horizontal inputs. (b) Neuron n_2 , recognizing vertical inputs. (c) Neuron n_3 , recognizing two-dimensional inputs with dissimilar entries. Error mitigation is applied to data for n_1 and n_2 (see supplementary information <https://stacks.iop.org/QST/05/044010/mmedia>).

4. Experimental realization on a superconducting quantum processor

We have implemented the ffNN introduced in figure 2 on a real superconducting quantum processor made available on cloud via the IBM Quantum Experience and programmed using the Qiskit python library [31]. Employing the same device, named Poughkeepsie, we realized both the hybrid (figure 3(a)) and the fully coherent (figure 3(b)) configurations, reporting in both cases a remarkable successful completion of all the desired classification tasks.

In figures 4(a) and (b) we show the results for the three-qubit simulation of nodes n_1 and n_2 , respectively corresponding to the first and second set of three qubits in figure 3(a), from which the probabilities p_1 and p_2 can be estimated for all possible input vectors while assuming the weights \vec{w}_1 and \vec{w}_2 shown in figure 2(a). The comparison with ideal results simulated numerically shows an excellent qualitative agreement and a good quantitative match of the outcomes: in particular, notice that each individual node can successfully single out either vertical or horizontal lines, see patterns in figure 2(b) [19]. The agreement is naturally better for the simulation of all possible n_3 circuits, whose results are reported in figure 4(c): indeed, in this case the probability $p(a_3 = 1 | [b_1, b_2])$ can be computed operating on a single qubit. The final outcomes (i.e. p^{out}) for the hybrid configuration of the ffNN, reported in figure 5(a), are then obtained by applying equation (11). The latter is used in place of, e.g., equation (13), in order to avoid introducing unnecessary assumptions or biases in the calculation and to take into account all possible sources of inaccuracy such as, for example, a non exactly zero outcome for $p(a_3 = 1 | [0, 0])$.

Finally, the experimental results for the fully coherent ffNN configuration are reported in figure 5(b). These were obtained by running the seven-qubit quantum circuit introduced in figure 3(b). As it can immediately be appreciated, the outcomes are in good agreement with the corresponding ones in the hybrid



version of the fNN. We stress that such comparison is made non-trivial from the experimental point of view by the fact that, in the fully coherent version, a register of 7 simultaneously active and typically entangled qubits is required. On the contrary, the hybrid solution only requires each individual node to be separately implemented on a three-qubit quantum register and, provided that the classical outcomes are conveniently stored, such quantum computations can be carried out in dedicated runs, thus avoiding, e.g., cross-talk effects. As in the hybrid case, and despite some residual quantitative inaccuracy in the estimation of the activation probabilities, all the possible inputs are classified correctly by the fNN, with the target horizontal and vertical patterns singled out from all other patterns.

We also mention that raw data from the quantum processor, reported in the supplementary information <https://stacks.iop.org/QST/05/044010/mmedia>, already allow for an accurate classification in both hybrid and coherent configurations. However, the overall quality of the outcomes greatly benefits from the application of simple error mitigation techniques [32–35], as described in detail in the supplementary information.

5. Discussion

In this work we have presented an original architecture to build fNNs on universal quantum computing hardware and demonstrated their use in near-term quantum devices. In particular, we have shown how successive layers constituted by artificial neurons and implemented on independent quantum registers can be either connected to each other via classical control operations, thus realizing a hybrid quantum–classical fNN, or by fully coherent quantum synapses. The necessary degree of non-linearity is achieved in one case via explicit quantum measurements, in the other by a partial trace operation that effectively produces a convolution operation. We stress that our proposed procedure is hardware-independent, and it can thus be implemented on any quantum computing machine, in principle, e.g. based on either superconducting qubits [36], trapped-ions [37], or photonic integrated circuits [38, 39]. In particular, it is worth noticing that native multi-qubit operations, such as the typical Mølmer–Sørensen collective entangling gates [40] commonly implemented in trapped ions quantum hardware, could lead to more efficient exact or even approximate [41] realizations of multi-controlled operations that are ubiquitous in the proposed scheme.

We have successfully tested a three-node implementation of our algorithm applied to an elementary pattern classification task, both in the hybrid and fully coherent configurations. Such proof-of-principle demonstration was achieved on the IBM Quantum Poughkeepsie superconducting quantum processor by using up to 7 active qubits, and finding a substantial experimental agreement between the two proposed operating modes of the network. These results represent, to the best of our knowledge, one of the largest quantum neural network computation reported to date in terms of the total size of the actual quantum register. We also notice that the use of quantum artificial neurons as individual nodes gives the prospective advantage of an exponential gain in storage and processing ability: in turn, this confirms that hybrid quantum–classical neural networks could already be able to treat very large input vectors, beyond the

capabilities of current systems. Such ability is becoming increasingly needed to handle, e.g., very large image files, sanitary data for public health, market data for financial applications, and the ‘data deluge’ expected from the Internet of Things. Moreover, the hybrid structure of our proposed ffNN could actually represent a relevant technical feature in the process of integrating quantum and classical processes for machine learning tasks: indeed, it could be envisioned that a few carefully distributed quantum nodes at the input of an otherwise classical network might act as a memory-efficient convolutional layer enabling the treatment of otherwise unmanageable sets of data.

A very natural extension of this work, and particularly of the fully coherent setup, would be an exploration of classically inaccessible regimes with no hybrid (i.e. classically controlled) counterpart. This could be achieved, e.g., by allowing more complex synapses, thus letting activation probabilities for all neurons feeding the same successive layer to interfere in a truly quantum coherent way, or by engineering non-trivial quantum correlations between quantum nodes already within the same layer. In addition to the large advantage in data treatment capacity, this could then also result in new functionalities, such as the ability to deploy complicated convolution filters impossible to be run on classical hardware.

Even further reaching consequences might be expected from the possibility to directly process quantum data instead of quantum-encoded classical information, for instance to search for patterns in the output of a quantum simulator or process quantum states coming from a quantum internet appliance. In these cases, the input would directly be given in the form of a wavefunction or a density matrix [30], without the resource cost associated to a classical input [41–43]. Indeed, it must be recalled that, in general, an exact implementation of the single-node operations U_i and U_w can require a computational cost scaling exponentially in the size of the quantum register. In this respect, it is worth mentioning that the class of quantum hypergraph states considered in this work can be seen as a subset of the more general family of locally maximally entangleable states (LMES) [44]. On one hand, such relationship directly suggests an extension to continuously-valued data [45]. On the other hand, the properties of the LMES could be leveraged in some cases to ease the problem of input preparation: in particular, whenever the required ± 1 phase factors can be described as a degree- k polynomial function of the corresponding computational basis states, the corresponding $|\psi_i\rangle$ can be constructed using only k -qubit operations. Moreover, the LMES can be physically obtained as unique fixed points of engineered dissipative processes or as ground states of frustration-free Hamiltonians [44].

Along parallel lines, the manipulation stage represented by U_w could most effectively be rephrased in terms of parametrized variational circuits [46]. Preliminary results [47] based on recently proposed quantum unsampling protocols [48] have shown that, leveraging the characteristic tensor product structure of the output state $|1\rangle^{\otimes N}$ in an active quantum neuron register under the proposed encoding, an efficient variational implementation of U_w for a single node can be learned using only single-qubit local cost functions. Such adaptive quantum circuit implementation of U_w could also become an essential ingredient when tackling the problem of training the quantum neural network. While in the practical example shown in this work the weights were selected beforehand, and not discovered through an optimization process, it has been already shown that the nonlinearity intrinsically deriving from the measurement on the ancilla of each artificial neuron is sufficient to guarantee the required plasticity for training [19]. This means that the hybrid architecture for quantum artificial neural networks proposed in this work is fully compatible, in principle, with classical and most employed training algorithms such as the Newton–Raphson or the backpropagation method [4], respectively. The latter is certainly best applied in presence of continuously-valued input parameters: as an example, we provide in the supplementary information a numerical description of such procedure carried out on the proposed three-node ffNN, where non-binary weight values are allowed during the learning phase. However, such a procedure is not easily transferred into the fully coherent regime where, in principle, only the final output activation is observed and the hidden layers are discarded. At the same time, it is also true that severe limitations in the training of deep quantum neural networks have been recently discovered, mainly in the form of Barren plateaus [49–51], which may critically affect the scaling of many quantum ffNN architectures even under dissipative frameworks [52]. While in general the ffNN design proposed here, as well as its possible extensions in the variational and continuous variable regimes, may not be immune from such pathological phenomenology, some of the strategies that have been put forward to deal with this issue [50, 52] could be leveraged also in this case. In particular, layerwise approaches [53] could naturally be applied to the ffNN in the hybrid configuration. More in general, we point out that the modular structure of the proposed architecture, featuring subsets of clearly identifiable and potentially shallow nodes together with ancilla-based funnelling of information, may fit a recently discovered scheme [52] suggesting locality of the action of single perceptrons and of the associated cost function as a promising route for lifting the obstacles commonly associated with vanishing gradients.

In conclusion, we provide a clear-cut recipe to map classical ffNNs onto quantum processors, and our results suggest that the whole design may eventually benefit from paradigmatic quantum properties such as superposition and entanglement. This represents a necessary step toward the final goal of approaching quantum advantage in the operation and training of quantum neural network applications.

Acknowledgments

We thank M Fanizza and S Woerner for useful discussions. We acknowledge the University of Pavia Blue Sky Research project number BSR1732907. This research was also supported by the Italian Ministry of Education, University and Research (MIUR): ‘Dipartimenti di Eccellenza Program (2018-2022)’, Department of Physics, University of Pavia and PRIN Project INPhoPOL. IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at <https://www.ibm.com/legal/copytrade>.

ORCID iDs

Francesco Tacchino  <https://orcid.org/0000-0003-2008-5956>
 Panagiotis Barkoutsos  <https://orcid.org/0000-0001-9428-913X>
 Chiara Macchiavello  <https://orcid.org/0000-0002-2955-8759>
 Ivano Tavernelli  <https://orcid.org/0000-0001-5690-1981>
 Dario Gerace  <https://orcid.org/0000-0002-7442-125X>
 Daniele Bajoni  <https://orcid.org/0000-0001-6506-8485>

References

- [1] Rosenblatt F 1957 The Perceptron: a perceiving and recognizing automaton *Technical Report Inc. Report No. 85-460-1* Cornell Aeronautical Laboratory
- [2] Hinton G E, Osindero S and Teh Y-W 2006 A fast learning algorithm for deep belief nets *Neural Comput.* **18** 1527–54
- [3] Hinton G E 2007 Learning multiple layers of representation *Trends Cognit. Sci.* **11** 428–34
- [4] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
- [5] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
- [6] Nielsen M A and Chuang I L 2004 *Quantum Computation and Quantum Information* 1st edn (Cambridge: Cambridge University Press)
- [7] Shor P W 1997 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM J. Comput.* **26** 1484–509
- [8] Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502
- [9] Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3
- [10] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503
- [11] Schuld M, Sinayskiy I and Petruccione F 2014 The quest for a quantum neural network *Quantum Inf. Process.* **13** 2567–86
- [12] Rebentrost P, Bromley T R, Weedbrook C and Lloyd S 2018 Quantum Hopfield neural network *Phys. Rev. A* **98** 042308
- [13] Lloyd S, Mohseni M and Rebentrost P 2013 Quantum algorithms for supervised and unsupervised machine learning (arXiv:1307.0411)
- [14] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195
- [15] Lamata L 2020 Quantum machine learning and quantum biomimetics: a perspective *Machine Learning: Science and Technology* **1** 033002
- [16] Schuld M, Sinayskiy I and Petruccione F 2015 Simulating a perceptron on a quantum computer *Phys. Lett. A* **379** 660–3
- [17] Schuld M, Fingerhuth M and Petruccione F 2017 Implementing a distance-based classifier with a quantum interference circuit *Europhys. Lett.* **119** 60002
- [18] Cao Y, Guerreschi G G and Aspuru-Guzik A 2017 Quantum Neuron: an elementary building block for machine learning on quantum computers arXiv:1711.11240 [quant-ph]
- [19] Tacchino F, Macchiavello C, Gerace D and Bajoni D 2019 An artificial neuron implemented on an actual quantum processor *npj Quantum Information* **5** 26
- [20] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
- [21] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504
- [22] Wan K H, Dahlsten O, Kristjánsson H, Gardner R and Kim M S 2017 Quantum generalisation of feedforward neural networks *npj Quantum Information* **3** 36
- [23] Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green A G and Severini S 2018 Hierarchical quantum classifiers *npj Quantum Information* **4** 65
- [24] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8
- [25] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79
- [26] Mari A, Bromley T R, Izaac J, Schuld M and Killoran N 2019 Transfer learning in hybrid classical-quantum neural networks (arXiv:1912.08278 [quant-ph])

- [27] McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33
- [28] Rossi M, Huber M, Bruß D and Macchiavello C 2013 Quantum hypergraph states *New J. Phys.* **15** 113022
- [29] Torrontegui E and Garcia-Ripoll J J 2019 Unitary quantum perceptron as efficient universal approximator *Europhys. Lett.* **125** 30004
- [30] Beer K, Bondarenko D, Farrelly T, Osborne T J, Salzmann R, Scheiermann D and Wolf R 2020 Training deep quantum neural networks *Nat. Commun.* **11** 808
- [31] Aleksandrowicz G *et al* 2019 Qiskit: an open-source framework for quantum computing (<https://doi.org/10.5281/zenodo.2562110>)
- [32] Temme K, Bravyi S and Gambetta J M 2017 Error mitigation for short-depth quantum circuits *Phys. Rev. Lett.* **119** 180509
- [33] Li Y and Benjamin S C 2017 Efficient variational quantum simulator incorporating active error minimization *Phys. Rev. X* **7** 021050
- [34] Klcio N, Dumitrescu E F, McCaskey A J, Morris T D, Pooser R C, Sanz M, Solano E, Lougovski P and Savage M J 2018 Quantum–classical computation of Schwinger model dynamics using quantum computers *Phys. Rev. A* **98** 032331
- [35] Kandala A, Temme K, Córcoles A D, Mezzacapo A, Chow J M and Gambetta J M 2019 Error mitigation extends the computational reach of a noisy quantum processor *Nature* **567** 491–5
- [36] Wendin G 2017 Quantum information processing with superconducting circuits: a review *Rep. Prog. Phys.* **80** 106001
- [37] Schindler P *et al* 2013 A quantum information processor with trapped ions *New J. Phys.* **15** 123012
- [38] Aspuru-Guzik A and Walther P 2012 Photonic quantum simulators *Nat. Phys.* **8** 285–91
- [39] Takeda S and Furusawa A 2019 Toward large-scale fault-tolerant universal photonic quantum computing *APL Photonics* **4** 060902
- [40] Mølmer K and Sørensen A 1999 Multiparticle entanglement of hot trapped ions *Phys. Rev. Lett.* **82** 1835–38
- [41] Mosca M and Kaye P 2001 Quantum networks for generating arbitrary quantum states *Optical Fiber Communication Conf. and Int. Conf. on Quantum Information* (Optical Society of America) p PB28
- [42] Bergholm V, Vartiainen J J, Möttönen M and Salomaa M M 2005 Quantum circuits with uniformly controlled one-qubit gates *Phys. Rev. A* **71** 052330
- [43] Plesch M and Brukner C 2011 Quantum-state preparation with universal gate decompositions *Phys. Rev. A* **83** 032302
- [44] Kruszynska C and Kraus B 2009 Local entanglability and multipartite entanglement *Phys. Rev. A* **79** 052304
- [45] Mangini S, Tacchino F, Gerace D, Macchiavello C and Bajoni D 2020 Quantum computing model of an artificial neuron with continuously valued input data [arXiv:2007.14288](https://arxiv.org/abs/2007.14288) [quant-ph]
- [46] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid quantum-classical algorithms *New J. Phys.* **18** 023023
- [47] Tacchino F, Barkoutsos P, Macchiavello C, Tavernelli I, Gerace D and Bajoni D 2020 Variational learning for quantum artificial neural networks *The Proc. of 2020 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)*
- [48] Carolan J *et al* 2020 Variational quantum unsampling on a quantum photonic processor *Nat. Phys.* **16** 322–7
- [49] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812
- [50] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 An initialization strategy for addressing barren plateaus in parametrized quantum circuits *Quantum* **3** 214
- [51] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2020 Cost-function-dependent barren plateaus in shallow quantum neural networks ([arXiv:2001.00550](https://arxiv.org/abs/2001.00550) [quant-ph])
- [52] Sharma K, Cerezo M, Cincio L and Coles P J 2020 Trainability of dissipative perceptron-based quantum neural networks ([arXiv:2005.12458](https://arxiv.org/abs/2005.12458) [quant-ph])
- [53] Skolik A, McClean J R, Mohseni M, van der Smagt P and Leib M 2020 Layerwise learning for quantum neural networks ([arXiv:2006.14904](https://arxiv.org/abs/2006.14904) [quant-ph])