ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

An quantum algorithm for feedforward neural networks tested on existing quantum hardware

Francesco Tacchino[1], Dario Gerace[1], Chiara Macchiavello[1],
Panagiotis Barkoutsos[2], Ivano Tavernelli[2] and Daniele Bajoni[1] *

1- Università di Pavia, Italy

2- IBM Research Zurich, Rüschlikon, Switzerland

Abstract. We present a memory-efficient quantum algorithm implementing the action of an artificial neuron according to a binary-valued model of the classical perceptron. The algorithm, tested on noisy IBM-Q superconducting real quantum processors, succeeds in elementary classification and image-recognition tasks through a hybrid quantum-classical training procedure. Here we also show that this model is amenable to be extended to a multilayered artificial neural network, which is able to solve a task that would be impossible to a single one of its constituent artificial neurons, thus laying the basis for a fully quantum artificial intelligence algorithm run on noisy intermediate-scale quantum hardware.

1 Quantum algorithm for a single artificial neuron

Artificial neural networks are nowadays recognized as a fundamental class of computational models with huge potential in applications such as pattern recognition, classification, and decision making. Prospective quantum computers seem particularly well suited for implementing artificial neural networks[1], as they can in principle represent and store large complex valued vectors, as well as perform linear operations on such vectors more efficiently than their classical counterparts [2]. However, current applications are limited to Noisy Intermediate Scale Quantum (NISQ) devices, such as the IBM quantum processors available for cloud quantum computing with up to 20 non-error corrected superconducting qubits, or trapped ions quantum computers. Here we introduce an efficient quantum information based algorithm allowing to implement on a NISQ processor the simplest model of an artificial neuron, the so called "perceptron", and then we extend it to a multilayered artificial neural network. The elementary unit of our algorithm, i.e. the quantum neuron, takes a real valued vector ($\vec{i}$) as the input and combines it with a real valued weight vector ($\vec{w}$): the output is evaluated as a binary response function applied to the inner product of the two vectors. In its simplest realization, $\vec{i}$ and $\vec{w}$ are binary valued vectors themselves [4]. We show that our quantum algorithm potentially allows for exponential advantage in storage resources over alternative realizations [5].

*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.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

The general quantum information-based procedure relies on the generation of multipartite entangled states known as hypergraph states [6], and it allows to optimize the quantum computational resources to be employed. In fact, any m-dimensional binary input $\vec{i}$ and weight $\vec{w}$ vector ($\vec{i}, \vec{w} \in \{-1, 1\}^m$) is encoded using the $m = 2^N$ coefficients in a real equally weighted wavefunction $|\psi_i\rangle$ of N qubits:

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

Here, the states $|j\rangle$ represent elements of the computational basis. First, the algorithm prepares the state $|\psi_i\rangle$: this step is efficiently implemented by applying an optimal generation procedure [6]. Assuming the qubits to be initialized in the state $|00\ldots00\rangle \equiv |0\rangle^{\otimes N}$, we perform a unitary transformation U_i such that $U_i|0\rangle^{\otimes N} = |\psi_i\rangle$. The second step computes the inner product between $\vec{w}$ and $\vec{i}$ using the quantum register. This task can be performed by defining a unitary transformation U_w such that $U_w|\psi_w\rangle = |1\rangle^{\otimes N} = |m-1\rangle$. If we apply U_w after U_i, it is easily seen that $\vec{w} \cdot \vec{i} = m\langle\psi_w|\psi_i\rangle$ is contained, up to a normalization factor, in the coefficient c_{m-1} of the final state $|\phi_{i,w}\rangle = U_w U_i |0\rangle^{\otimes N}$. In order to extract such an information and obtain a non-linear activation function, we use an ancilla qubit (a) initially set in the state $|0\rangle$. A multi-controlled NOT gate between the N encoding qubits and the target a leads to:

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

By measuring the state of the ancilla qubit in the computational basis the output $|1\rangle_a$ (i.e., an activated perceptron) is obtained with probability $|c_{m-1}|^2$. Our quantum perceptron model is particularly suited as an image classifier, since it allows to interpret a given pattern and its negative on equivalent footing due to the global phase symmetry in the encoding quantum wavefunction. Overall, a N-qubit implementation of our model allows to process 2^{2^N} different binary inputs or weights.

2 Experiments on real quantum processors

First, we implemented the algorithm for a single quantum perceptron both on classical simulators working out the matrix algebra of the circuit and on cloud-based IBM-Q real backends [7]. Due to the constraints imposed by the actual hardware in terms of connectivity between the different qubits, we limited the experimental proof-of-principle to the $N = 2$ case. Nevertheless, even this small-scale example is already sufficient to show all the distinctive features of our proposed set up. Here, $2^2 = 4$ binary images can be managed, and thus $2^{2^2} = 16$ different patterns can be analyzed. All binary inputs and weights can easily be converted into black and white patterns, thus providing a visual interpretation of the artificial neuron activity, see Fig. 1(b). Remarkably, all the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

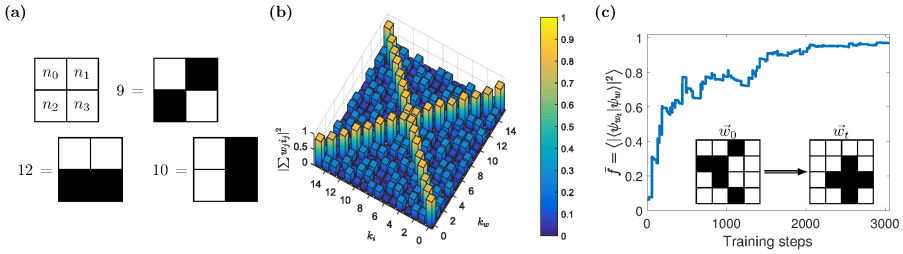

Fig. 1: (a) Encoding scheme for the $N = 2$ case. Input and weight binary vectors can be seen as 2-by-2 pixels black and white images, labeled with an integer $k_{i(w)}$ corresponding to the decimal representation of the binary string $\mathtt{n}_0\mathtt{n}_1\mathtt{n}_2\mathtt{n}_3$, where $i(w)_j = (-1)^{n_j}$ and $n_j = 0(1)$ for a black (white) pixel. (b) Output of the perceptron computed experimentally on the IBM Q-5 "Tenerife" quantum processor for the $N = 2$ case. The perceptron successfully singles out any weight pattern and its negative from all possible inputs. (c) Training of the perceptron for the $N = 4$ case. The target cross-shaped pattern is effectively learned starting from a random initial state.

classification tasks are correctly carried out despite the intrinsic noise on the non error-corrected NISQ processor, as shown explicitly in Fig. 1(b). Notice that the global phase symmetry is naturally embedded in the algorithm itself, and the results show symmetric performances all over the range of possible inputs and weights. All combinations of $\vec{i}$ and $\vec{w}$ yield results either larger than 0.75 (ideal value 1) or smaller than 0.3 (ideally 0 or 0.25, depending on the pair of $\vec{i}$ and $\vec{w}$ vectors), in good quantitative agreement with the expected results. We notice that the algorithm presented here is fully general and could be implemented and run on any platform capable of performing universal quantum computation. While we have employed a quantum hardware that is based on superconducting technology and qubits, a very promising alternative is the trapped-ion based quantum computer, in which multi-qubit entangling gates might also be readily available.

3 Hybrid training of the artificial neuron

To fully assess the potential of this model for classification purposes, we have implemented an elementary hybrid quantum-classical training scheme for the $N = 4$ case, adapting the perceptron update rule to our algorithm. For $N = 4$, 2^{32} possible combinations of $\vec{i}$ and $\vec{w}$ vectors are possible, far too many to explore the whole combinatorial space as previously done for the 2 qubits. After preparing a random training set containing pre-labelled positive and negative inputs, the binary valued artificial neuron is trained to recognize a pre-defined target vector, $\vec{w}_t$. The latter is chosen to correspond to a simple cross-shaped pattern when represented as a 4×4 pixels image. This training procedure is obtained by computing the artificial neuron output through our proposed quantum algorithm

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

on the noiseless Qiskit simulator [7], while the optimization of the weight vector $\vec{w}$ is performed by a classical processor. We selected a random vector to start with, $\vec{w}_0$, and then we let the artificial neuron process the training set according to well defined rules, without ever conveying explicit information about the target $\vec{w}_t$. In Fig. 1c, we report the average value of the fidelity of the quantum state $|\psi_w\rangle$ encoding the trained $\vec{w}$ with respect to the target state $|\psi_{w_t}\rangle$ over 500 realization of the training scheme, all with the same initial pattern $\vec{w}_0$ and the same training set. The quantum artificial neuron effectively learns the target cross-shaped pattern.

4 A multilayered artificial network

As in the historical development of classical artificial intelligence, we move from the simple single layer perceptron design to that of a complete feedforward neural network [8], constituted by several neurons organized in multiple successive layers. In such artificial neural network design, each constituent neuron receives, as inputs, the outputs (activations) from the neurons in the preceding layer. Here we present an original architecture in which two successive layers of the quantum artificial neurons introduced above [10] are connected through synapses to build a simple feedforward artificial neural network. The network is used to solve a classification task that would be impossible for a single artificial neuron. A scheme of an elementary extension of our previous work is given in Fig. 2(a), in which the circles indicate quantum artificial neurons, and the vectors $\vec{w}_i$ refer to their respective weights as before. We chose a simple design, with a single hidden layer and a single binary (i.e., yes/no) output neuron to solve an elementary but quite meaningful problem: the network should be able to recognize wether the 2×2 input image contains lines, regardless of the fact the lines are horizontal or vertical. All the other possible input images should be classified as negative. The working principle of the network is straightforward: the top quantum neuron of the hidden layer is activated if the input vector has vertical lines, while the bottom neuron does the same for the case of horizontal lines, then the output neuron in the last layer "decides" whether one of the previous neurons has given a positive outcome.

As we mentioned above this problem is significant because it is impossible to solve by using a single layer perceptron model. Indeed the set of patterns that should yield a positive result include 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 is opposite to the vector corresponding to a vertical line on the right column). Given an input vector $\vec{v}_1$ and a weight vector $\vec{w}$ a single quantum neuron would output a value proportional to $\vec{w} \cdot \vec{v}_1$ i.e. $\cos^2(\vartheta)$ where ϑ is the angle formed by the two vectors. Now if we take a second input vector $\vec{v}_2 \perp \vec{v}_1$ the output would clearly be proportional to $\sin^2(\vartheta)$. It is therefore impossible to find a weight $\vec{w}$ capable of yielding an output activation larger than 0.5 for both $\vec{v}_1$ and $\vec{v}_2$, i.e. it is impossible to find an hyperplane correctly

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

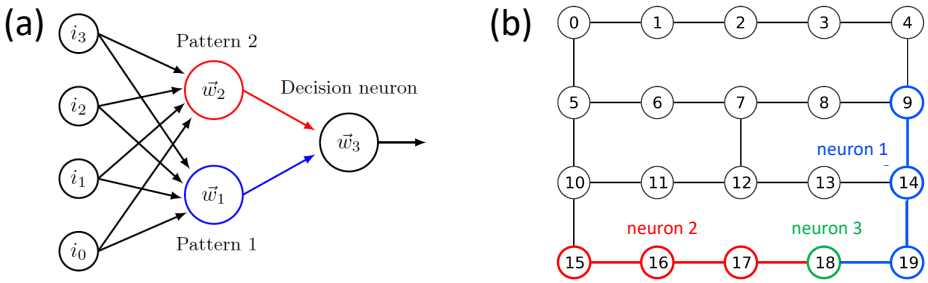

Fig. 2: (a) scheme of the quantum deep neural network studied in this work, where each node is represented by a quantum artificial neuron, a hidden layer made of two neurons is interposed between an input layer and a single output neuron; (b) scheme of the experimental implementation of the quantum deep neural network represented in (a) on the IBM Q "Poughkeepsie" quantum processor, in which the two neurons of the hidden layer are encoded into a suitable combination of three qubits on the real hardware, and the output layer is made of a single qubit. Quantum synapses are implemented through multi-controlled operations [10].

separating the configuration space. Conversely, when using a single classical perceptron, it is well known that it is impossible to find a hyperplane capable of classifying a set containing opposite vectors [9].

We show the potentialities of NISQ hardware by generalizing the quantum algorithm in Fig. 1, i.e. by implementing the artificial neural network of Fig. 2(a). First, we have modeled and implemented the hidden and output layers in a circuit based model of quantum computation, and connected them through "quantum synapses", which are remarkably able to set the input of the last neuron based on the outputs of the hidden layer in a quantum coherent way. Hence, we tested this algorithm on a state-of-the art 20-qubit IBMQ quantum processor. In particular, we obtained preliminary experimental results with a 7-qubit calculation on the IBMQ Poughkeepsie quantum processor, whose schematic layout is shown in Fig. 2(b), in which a possible combination of the qubits to be employed for such calculation is highlighted. The experimental results were analyzed with the inclusion of error mitigation techniques. The overall outcome shows that, despite some residual quantitative inaccuracy, all the target patterns of the working network are correctly recognized and all other patterns are rejected, which represents a crucial step in view of building feedforward deep neural networks on universal quantum computing hardware.

5 Conclusions

The results presented here form a first proof that neural networks with architectures equivalent to classical neural networks can be implemented on quantum processors. It is important to notice that the present implementation can be

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

equivalently realized via a semi-classical approach in which the ancillas of the neurons in a layer are measured and the weights of the neurons in the next layer are set using classically controlled unitary operations. Another semi-classical approach could consist in using only classical neuron layers after a few quantum neuron layers. Such a design could be beneficial in tapering convolutional neural networks. The exponential memory advantage of quantum neurons could be used convolutional filters for pattern too complicated to be treated classically, while the simpler parts of the network would be run on tradtional hardware.

The current approach of running the whole network in fully quantum coherent way is however interesting as it points to the possibility of putting the network in a superposition state for different parameters of its constituent neurons. This result, if achieved, would be the first step toward the possibility of usign quantum algorithms for the training of neural networks.

References

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd, *Quantum machine learning*, Nature **549**, 195–202 (2017).

[2] M. Schuld, I. Sinayskiy and F. Petruccione, *The quest for a Quantum Neural Network*, Quantum Information Processing **13**, 2567–2586 (2014).

[3] F. Rosenblatt, *The Perceptron: A perceiving and recognizing automaton*, Tech. Rep. Inc. Report No. 85-460-1 (Cornell Aeronautical Laboratory, 1957).

[4] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics **5**, 115–133 (1943).

[5] F. Tacchino, C. Macchiavello, D. Gerace and D. Bajoni, *An Artificial Neuron Implemented on an Actual Quantum Processor*, npj Quantum Information **5**, 26 (2019).

[6] M. Rossi, M. Huber, D. Bruß and C. Macchiavello, *Quantum hypergraph states*, New Journal of Physics **15**, 113022 (2013).

[7] See https://quantumexperience.ng.bluemix.net, https://qiskit.org/

[8] G. E. Hinton, S. Osindero and Y. Teh, *A fast learning algorithm for deep belief nets.* Neural Computation **18**, 1527 (2006).

[9] Ian Goodfellow and Yoshua Bengio and Aaron Courville *Deep Learning*, MIT Press. See http://www.deeplearningbook.org (2016).

[10] F. Tacchino, P. Barkoutsos, C. Macchiavello, I. Tavernelli, D. Gerace, and D. Bajoni, *Quantum implementation of an artificial feed-forward neural network*, in preparation (2019).