



PERSPECTIVE

Quantum computing models for artificial neural networks

To cite this article: S. Mangini *et al* 2021 *EPL* **134** 10002

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

Perspective

Quantum computing models for artificial neural networks

S. MANGINI^{1,2}, F. TACCHINO³, D. GERACE¹, D. BAJONI⁴ and C. MACCHIAVELLO^{1,2,5(a)}

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

² *INFN Sezione di Pavia - Via Bassi 6, I-27100, Pavia, Italy*

³ *IBM Quantum, IBM Research-Zurich - Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland*

⁴ *Dipartimento di Ingegneria Industriale e dell'Informazione, Università di Pavia - Via Ferrata 1, I-27100, Pavia, Italy*

⁵ *CNR-INO - Largo E. Fermi 6, I-50125, Firenze, Italy*

received 5 February 2021; accepted in final form 18 March 2021

published online 14 May 2021

Abstract – Neural networks are computing models that have been leading progress in Machine Learning (ML) and Artificial Intelligence (AI) applications. In parallel, the first small-scale quantum computing devices have become available in recent years, paving the way for the development of a new paradigm in information processing. Here we give an overview of the most recent proposals aimed at bringing together these ongoing revolutions, and particularly at implementing the key functionalities of artificial neural networks on quantum architectures. We highlight the exciting perspectives in this context, and discuss the potential role of near-term quantum hardware in the quest for quantum machine learning advantage.

perspective

Copyright © 2021 EPLA

Introduction. – Artificial Intelligence (AI) broadly refers to a wide set of algorithms that have shown in the last decade impressive and sometimes surprising successes in the analysis of large datasets [1]. These algorithms are often based on a computational model called Neural Network (NN), which belongs to the family of differential programming techniques.

The simplest realization of a NN is in a “feedforward” configuration, mathematically described as a concatenated application of affine transformations and elementwise nonlinearities, $f_j(\cdot) = \sigma_j(\mathbf{w}_j \cdot + b_j)$, where σ_j is a nonlinear *activation function*, \mathbf{w}_j is a real matrix containing the so-called *weights* to be optimized, and b_j is a *bias* vector, which is also to be optimized. The action of the NN is defined through repeated applications of this function to an input vector, x , leading to an output $\hat{y} = f_L \circ f_{L-1} \circ \dots \circ f_1(x)$, where L denotes the number of layers in the NN architecture, as schematically shown in fig. 1(a). The success of NNs as a computational model is primarily due to their trainability [2], *i.e.*, the fact that the entries of the weight matrices can be adjusted to learn a given target task. The most common use case is *supervised learning*, where the algorithm is asked to reproduce the associations between some inputs x_i and the desired correct outputs (or label) y_i . If properly trained, the NN

should also be able to predict well on new data, *i.e.*, data that were not used during training. This crucial property is called *generalization*, and it represents the key aspect of learning algorithms, which ultimately distinguish them from standard fitting techniques. Generalization encapsulates the idea that the learning models discover hidden patterns and relations in the data, instead of applying pre-programmed procedures [1,3,4].

The size of the data and features elaborated by classical NNs, and the number of parameters defining their structure, have been steadily increasing over the last years. These advancements have come at the expense of a dramatic rise in the demand for energy and computational power [5] from the field of AI. In these regards, quantum computers may offer a viable route forward to continue such growth in the dimension of the treatable problems and, in parallel, enable completely new functionalities [6,7]. In fact, quantum information is elaborated by exploiting superpositions in vector spaces (the so-called Hilbert spaces) [8], and quantum processors are planned to soon surpass the computational capabilities of classical supercomputers. Moreover, genuinely quantum correlations represent by themselves a resource to be considered in building new forms of ML applications.

With respect to classical computational paradigms, quantum computing comes with its own peculiarities. One of these is that quantum computation is intrinsically

^(a) E-mail: chiara@unipv.it (corresponding author)

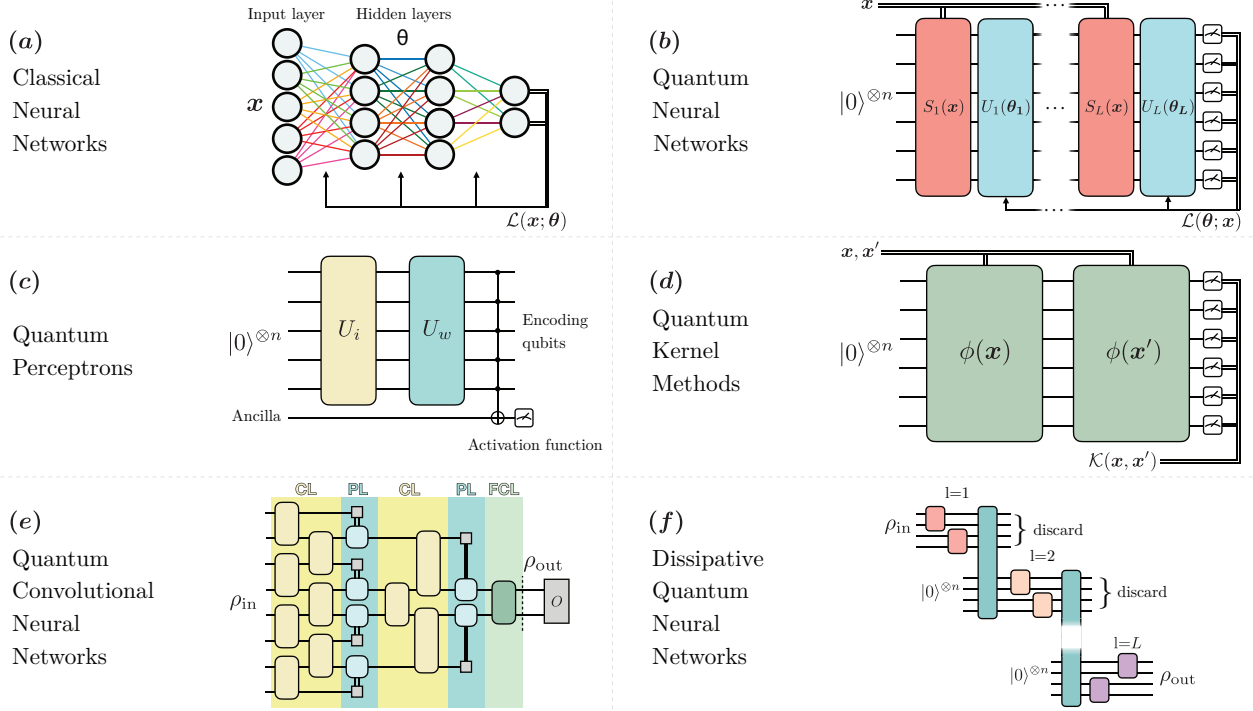


Fig. 1: Schematic representation of classical and quantum learning models. (a) Classical feedforward NNs process information through consecutive layers of nodes, or *neurons*. (b) A quantum neural network in the form of a variational quantum circuit: the data *re-uploading* procedure ($S_i(x)$) is explicitly illustrated, and non-parametrized gates (W_i) of eq. (1) are incorporated inside the variational unitaries ($U_i(\theta_i)$). (c) A possible model for a quantum perceptron [29], which uses operations U_i and U_w to load input data and perform multiplication by weights, and a final measurement is used to emulate the neuron activation function. (d) Quantum kernel methods map input data to quantum states [33], and then evaluate their inner product $\langle \phi(x) | \phi(x') \rangle$ to build the kernel matrix $\mathcal{K}(x, x')$. (e) A Quantum convolutional NN [44] consists of a repeated application of Convolutional Layers (CL) and Pooling Layers (PL), ultimately followed by a Fully Connected Layer (FCL), *i.e.*, a general unitary on all remaining qubits applied before measuring an output observable, O . (f) In a dissipative QNN [46], qubits in one layer are coupled to ancillary qubits from the following layer in order to load the result of the computation, while previous layers are discarded (*i.e.*, *dissipated*).

linear. While this property could ensure, *e.g.*, that an increase in computational power does not come with a parallel increase in energy costs, it also means that implementing the nonlinear functions forming the backbone of NNs is a non-trivial task. Also, information cannot be copied with arbitrary precision in quantum algorithms, meaning that tasks such as repeatedly addressing a variable are impossible when the latter is represented by a quantum state. As a consequence, classical NN algorithms cannot be simply ported to quantum platforms.

In this article, we briefly review the main directions that have been taken through the years to develop artificial NNs for quantum computing environments, the strategies used to overcome the possible difficulties, and the future prospects for the field.

Quantum neural networks. – In gate-based quantum computers, such as those realized with superconducting qubits, ion traps, spins in semiconductors, or photonic circuits [9], a quantum computation is fully described by its quantum circuit representation, *i.e.*, the arrangement of subsequent operations (gates) applied to the qubits

in the quantum processing unit. These gates can be parametrized by continuous variables, like a Pauli-X rotation, $R_x(\theta) = e^{i\theta X/2}$. A quantum circuit that makes use of parametrized gates is known as a *Parametrized Quantum Circuit* (PQC) [10]. *Variational Quantum Algorithms* (VQAs) [11–13] are a large class of PQC-based protocols whose goal is to properly tune the parameters in order to solve a target task. For example, the task might be the minimization of the expectation value of an observable $\langle O \rangle = \langle 0 | U(\theta)^\dagger O U(\theta) | 0 \rangle$, where $U(\theta)$ is the unitary operator representing the quantum circuit parametrized by angles $\theta = (\theta_1, \theta_2, \dots)$ and acting on a quantum register initialized in a reference state $|0\rangle = |0\rangle^{\otimes n}$. VQAs are made by three main constituents: data encoding, variational *ansatz*, and final measurements with a classical update of the parameters. The first stage accounts for loading input data (classical or quantum) into the quantum circuit using quantum operations that depend on the given input; the second stage consists of a variational circuit with a specific structure, called *ansatz*, which often consists of repeated applications of similar operations (*layers*). Finally, a measurement on the qubits is performed to infer some relevant

information about the system. Given the outcome, the parameters in the circuits are updated through a classical optimizer to minimize the cost function defining the problem. Such algorithms generally require limited resources to be executed on real quantum devices, thus they are expected to take full advantage of current noisy quantum hardware [14,15].

The definitions above highlight many similarities between PQCs and classical NNs. In both models, information is processed through a sequence of parametrized *layers*, which are iteratively updated using a classical optimizer. The key difference lies in the way information is processed. Quantum computers promise to possibly achieve some form of computational advantage, *i.e.*, speedups or better performances [7], due to inherently quantum effects not available in classical learning scenarios. While many different definitions of Quantum Neural Networks (QNN) are possible, the most general and currently accepted one is that of a parametrized quantum circuit whose structure is directly inspired by classical neural networks. Thus, we formally define a QNN as a PQC whose variational ansatz contains multiple repetitions of self-similar layers of operations, that is

$$U_{\text{QNN}}(\boldsymbol{\theta}) = \prod_{i=L}^1 U_i(\boldsymbol{\theta}_i) W_i \\ = U_L(\boldsymbol{\theta}_L) W_L \cdots U_1(\boldsymbol{\theta}_1) W_1, \quad (1)$$

where $U_i(\boldsymbol{\theta}_i)$ are variational gates, W_i are (typically entangling) fixed operations that do not depend on any parameter, and L is the number of layers (see fig. 1(b) for a schematic picture). Depending on the problem under investigation and the connectivity of the quantum device (*i.e.*, related to the native gate set available [9]), various realizations exist for both the entangling structure and the parametrized gates [13]. Note that this definition almost coincides with that of VQAs, and in fact the terms VQAs and QNNs are often used interchangeably.

While being rather general, a growing body of research has recently been highlighting that such a structure, in its naive realization, may be poorly *expressive*, in the sense that it can only represent a few functions—actually, mostly sine functions—of the input data [16–20]. A milestone of classical learning theory, the *Universal Approximation Theorem* [21], guarantees that classical NNs are powerful enough to approximate any function $f(\mathbf{x})$ of the input data \mathbf{x} . Interestingly, it was shown that a similar result also holds for QNNs, provided that input data are loaded into the quantum circuit multiple times throughout the computation. Intuitively, such data *re-uploading* can be seen as a necessary feature to counterbalance the effect of the *no-cloning theorem* of quantum mechanics [8]. In fact, while in classical NNs the output of a neuron is copied and transferred to every neuron in the following layer, this is impossible in the quantum regime, thus it is necessary to explicitly introduce some degree of classical redundancy. As a rule of thumb, the more often data are

re-uploaded during the computation, the more *expressive* the quantum learning model becomes, as it is able to represent higher-order features of the data. Thus, for a more effective construction, each layer in eq. (1) should be replaced with $U_i W_i \rightarrow S(\mathbf{x}) U_i W_i$, where $S(\mathbf{x})$ denotes the data encoding procedure mapping input \mathbf{x} to its corresponding quantum state.

This formulation still leaves a lot of freedom in choosing how the encoding and the variational circuit are implemented [22–24]. However, as it happens in the classical domain, also different architectures for QNNs exist, and in the following, we give a brief overview of some of the ones that received more attention.

Quantum perceptron models. Early works on quantum models for single artificial neurons (known as *perceptrons*) were focused on how to reproduce the non-linearities of classical perceptrons with quantum systems, which, on the contrary, obey linear unitary evolutions [25–28]. In more recent proposals (*e.g.*, fig. 1(c)), quantum perceptrons were shown to be capable of carrying out simple pattern recognition tasks for binary [29] and gray scale [30] images, possibly employing variational unsampling techniques for training and to find adaptive quantum circuit implementations [31]. By connecting multiple quantum neurons, models for quantum artificial neural networks were also realized on proof-of-principle experiments on superconducting devices [32].

Kernel methods. Directly inspired by their classical counterparts, the main idea of *kernel methods* is to map input data into a higher-dimensional space, and leverage this transformation to perform classification tasks otherwise unfeasible in the original, low-dimensional representation. Typically, any pair of data vectors, $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, is mapped into quantum states, $|\phi(\mathbf{x})\rangle, |\phi(\mathbf{x}')\rangle$, by means of an encoding mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$, where \mathcal{H} denotes the Hilbert space of the quantum register. The inner product $\langle\phi(\mathbf{x})|\phi(\mathbf{x}')\rangle$, evaluated in an exponentially large Hilbert space, defines a similarity measure that can be used for classification purposes. The matrix $\mathcal{K}_{ij} = \langle\phi(\mathbf{x}_i)|\phi(\mathbf{x}_j)\rangle$, constructed over the training dataset, is called *kernel*, and can be used along with classical methods such as Support Vector Machines (SVMs) to carry out classification [33,34] or regression tasks, as schematically displayed in fig. 1(d). A particularly promising aspect of such quantum SVMs is the possibility of constructing kernel functions that are hard to compute classically, thus potentially leading to quantum advantage in classification.

Generative models. A generative model is a probabilistic algorithm whose goal is to reproduce an unknown probability distribution $p_{\mathcal{X}}$ over a space \mathcal{X} , given a training set, $\mathcal{T} = \{\mathbf{x}_i | \mathbf{x}_i \sim p_{\mathcal{X}}, i = 1, \dots, M\}$. A common classical approach to this task leverages probabilistic NNs called Boltzmann Machines (BM) [2,35], which are physically inspired methods resembling Ising models. Upon changing classical nodes with qubits, and substituting the energy function with a quantum Hamiltonian, it is possible

to define Quantum Boltzmann Machines (QBM)s [36–38], dedicated to the generative learning of quantum and classical distributions. On a parallel side, a novel and surprisingly effective classical tool for generative modeling is represented by *Generative Adversarial Networks* (GANs). Here, the burden of the generative process is offloaded onto two separate neural networks called *generator* and *discriminator*, respectively, which are programmed to compete against each other. Through a careful and iterative learning routine, the generator learns to produce high-quality new examples, eventually fooling the discriminator. A straightforward translation to the quantum domain led to the formulation of Quantum GANs (QGANs) models [39–42], where the generator and discriminator consist of QNNs. Even if not inspired by any classical neural network model, it is also worth mentioning Born machines [43], where the probability of generating new data follows the inherently probabilistic nature of quantum mechanics represented by Born’s rule.

Quantum convolutional neural networks. As the name suggests, these models are inspired by the corresponding classical counterpart, nowadays ubiquitous in the field of image processing. In these networks, schematically shown in fig. 1(e), inputs are processed through a repeated series of so-called *convolutional* and *pooling* layers. The former apply a convolution of the input —*i.e.*, a quasi-local transformation in a translationally invariant manner— to extract some relevant information, the latter apply a compression of such information in a lower-dimensional representation. After many iterations, the network outputs an informative and low-dimensional enough representation that can be analyzed with standard techniques, such as fully connected feedforward NNs. Similarly, in quantum convolutional NNs [44,45] a quantum state first undergoes a convolution layer, consisting of a parametrized unitary operation acting on individual subsets of the qubits, and then a pooling procedure obtained by measuring some of the qubits. This procedure is repeated until only a few qubits are left, where all relevant information was stored. This method proved particularly useful for genuinely quantum tasks such as quantum phase recognition and quantum error correction. Moreover, the use of only logarithmically many parameters with respect to the number of qubits is particularly important to achieve efficient training and effective implementations on near-term devices.

Quantum dissipative neural networks. Dissipative QNNs, as proposed in [46], consist of models where each qubit represents a node, and edges connecting qubits in different layers represent general unitary operations acting on them. The use of the term *dissipative* is related to the progressive discarding of layers of qubits after a given layer has interacted with the following one (see, *e.g.*, fig. 1(f) for a schematic illustration). This model can implement and learn general quantum transformations, carrying out a universal quantum computation. Endowed

with a backpropagation-like training algorithm, it represents a possible route towards the realization of quantum analogues of deep neural networks for analyzing quantum data.

Trainability of QNN. — For decades, AI methods remained merely academic tools, due to the lack of computational power and proper training routines which made the use of learning algorithms practically unfeasible on available devices. This underlines a very pivotal problem of all learning models, *i.e.*, the ability to efficiently train them.

Barren plateaus. Unfortunately, also quantum machine learning models were shown to suffer from serious trainability issues, which are commonly referred to as *barren plateaus* [47]. In particular, using a parametrization as in eq. (1), and assuming that such architecture forms a so-called unitary 2-design¹, it can be shown that the derivative of any cost function of the form $C(\boldsymbol{\theta}) = \langle \mathbf{0} | U_{\text{QNN}}(\boldsymbol{\theta})^\dagger O U_{\text{QNN}}(\boldsymbol{\theta}) | \mathbf{0} \rangle$ with respect to any of its parameters θ_k , yields $\langle \partial_k C \rangle = 0$, and $\text{Var}[\partial_k C] \approx 2^{-n}$, where n is the number of qubits. This means that, on average, there will be no clear direction of optimization in the cost function landscape upon random initialization of the parameters in a large enough circuit, being it essentially flat everywhere, except in close proximity to the minima. Moreover, barren plateaus were shown to be strongly dependent on the nature of the cost function driving the optimization routine [48]. In particular, while the use of a global cost function always hinders the trainability of the model (independently of the depth of the quantum circuit), a local cost function allows for an efficient trainability for circuit depths of order $O(\log n)$. Besides, also entanglement plays a major role in hindering the trainability of QNNs. In fact, it was shown that whenever the objective cost function is evaluated only on a small subset of available qubits (visible units), with the remaining ones being traced out (hidden units), a barren plateau in the optimization landscape may occur [49]. This arises from the presence of entanglement, which tends to spread information across the whole network instead of its single parts. Moreover, also the presence of hardware noise during the execution of the quantum circuit can itself be the reason for the emergence of noise-induced barren plateaus [50].

It is worth noticing that most of these results stem from strong assumptions on the shape of the variational ansatz, for example, its ability to approximate an exact 2-design. While these assumptions are often well-matched by proposed quantum learning models, QNNs should be analyzed case by case to have precise statements about their trainability. As an example, dissipative QNNs were shown to suffer from barren plateaus [51] when global unitaries are used, while convolutional QNNs seem to be immune to the barren plateaus phenomena [52]. In general terms,

¹Roughly speaking, it is sufficiently random so that sampling over its distribution matches the Haar distribution —a generalization of the uniform distribution in the space of unitary matrices— up to the second moment.

however, recent results indicate that the trainability of quantum models is closely related to their expressivity, as measured by their distance from being an exact 2-design. Highly expressive ansatzes generally exhibit flatter landscapes and will be harder to train [53]. At last, it was recently argued that the training of any VQA is by itself an NP-hard problem [54], thus being intrinsically difficult to tackle. Despite their intrinsic theoretical value, these results should not represent a severe threat, as similar conclusions also hold in general for classical NNs, particularly in worst-case analysis, and did not prevent their successful application in many specific use cases.

Avoiding plateaus. Several strategies have been proposed to avoid, or at least mitigate, the emergence of barren plateaus. A possible idea is to initialize and train parameters in batches, in a procedure known as layerwise learning [55]. Similarly, another approach is to reduce the effective depth of the circuit by randomly initializing only a subset of the total parameters, with the others chosen such that the overall circuit implements the identity operation [56]. Introducing correlations between the parameters in multiple layers of the QNN, thus reducing the overall dimensionality of the parameter space [57], has also been proposed. Along a different direction, leveraging classical recurrent NNs to find good parameter initialization heuristics, such that the network starts close to a minimum, has also proven effective [58]. Finally, as previously mentioned, it is worth reminding that the choice of a local cost function and of specific entanglement scaling between hidden and visible units is of key importance to ultimately avoid barren plateaus [48,59].

Optimization routines. The most common training routine in classical NNs is the backpropagation algorithm [1], which combines the chain rule for derivatives and stored values from intermediate layers to efficiently compute gradients. However, such a strategy cannot be straightforwardly generalized to the quantum case, because intermediate values from a quantum computation can only be accessed via measurements, which disturb the relevant quantum states [60]. For this reason, the typical strategy is to use classical optimizers leveraging gradient-based or gradient-free numerical methods. However, there exist scenarios where the quantum nature of the optimization can be taken into account, leading to strategies tailored to the quantum domain like the *parameter shift rule* [19,60,61], the quantum natural gradient [62], and closed update formulas [20]. Finally, while a fully coherent quantum update of the parameters is certainly appealing [63], this remains out of reach for near-term hardware, which is limited in the number of available qubits and circuit depth [9].

Quantum advantage. – Quantum computers are expected to become more powerful than their classical counterparts on many specific applications [64–66], and particularly in sampling from complex probability distributions —a feature particularly relevant for

generative modeling applications [67,68]. It is therefore natural to ask whether this hierarchy also applies to learning models. While a positive answer is widely believed to be true, clear indications on how to achieve such a quantum advantage are still under study, with only a few results in this direction.

Power of quantum neural networks. In the previous sections, we used the term *expressivity* quite loosely. In particular, we used it with two different meanings: the first to indicate the ability of QNNs to approximate any function of the inputs [17], the second to indicate their distance from being an exact 2-design, *i.e.*, the ability to span uniformly the space of quantum states [53]. Each definition conveys a particular facet of an intuitive sense of *power*, highlighting the need for a unifying measure of such power yet to be discovered. Nevertheless, some initial results towards a thorough analysis of the capacity (or power) of quantum neural networks have recently been put forward [69–71].

Starting from the Vapnik-Chervonenkis (VC) dimension, which is a classical measure of *richness* of a learning model related to the maximum number of independent classifications that the model can implement, a universal metric of capacity related to the maximum information that can be stored in the parameters of a learning model, be it classical or quantum, was defined [69]. Based on this definition, it was claimed that quantum learning models have the same overall capacity as any classical model having the same number of weights. However, as argued in ref. [70], measures of capacity like the VC dimension can become vacuous and difficult —if not completely impractical— to be measured, and therefore cannot be used as a general faithful measure of the actual power of learning models. For this reason, the use of *effective dimension* as a measure of power is proposed, motivated by information-theoretical standpoints. This measure can be linked to generalization bounds of the learning models, leading to the result that quantum neural networks can be more expressive and efficient during training than their classical counterparts, thanks to a more evenly spread spectrum of the Fisher Information. Some evidence was provided through the analysis of two particular QNNs instances: one having a trivial input data encoding, and the other using the encoding scheme proposed in [33], which is thought to be classically hard to simulate and indeed shows remarkable performances. Despite such specificity, this is one of the first results drawing a clear separation between quantum and classical neural networks, highlighting the great potentials of the former.

The role of data. In the same spirit, but with different tools, the role of data in analyzing the performances of quantum machine learning algorithms was taken into account [71], by devising geometric tests aimed at clearly identifying those scenarios where quantum models can achieve an advantage. In particular, focusing on kernel methods, a geometric measure based exclusively on the

training dataset and the kernel function induced by the learning model, which is actually independent of the actual classification/function to be learned, was defined. If such quantity is similar for the classical and quantum learning case, then classical models are proved to perform similarly or better than quantum models. However, if this geometric distance is large, then there exists some dataset where the quantum model can outperform the classical one. Notably, one key result of the study is that poor performances of quantum models could be attributed precisely to the careless encoding of data into exponentially large quantum Hilbert spaces, spreading them too far apart from each other and resulting in the kernel matrix approximating the identity. To avoid this setback, authors propose the idea of *projected quantum kernels*, which aim at reducing the effective dimension of the encoded data set while keeping the advantages of quantum correlations. On engineered data sets, this procedure was shown to outperform all tested classical models in prediction error. From a higher perspective, the most important handout of these results is that data themselves play a major role in learning models, both quantum and classical.

Quantum data for quantum neural networks. Some data set could be particularly suited for QNNs. An example is provided by those proposed in [72] and based on the discrete logarithm problem. Also, it is not hard to suppose that quantum machine learning will prove particularly useful when analyzing data that are inherently quantum, for example coming from quantum sensors, or obtained from quantum chemistry simulations of condensed matter physics. However, even if this sounds theoretically appealing, no clear separation has yet been found in this respect. A particularly relevant open problem concerns, for example, the identification and generation of interesting quantum dataset. Moreover, based on information-theoretic derivations in the quantum communication formalism, it was recently shown [73] that a classical learner with access to quantum data coming from quantum experiments performs similarly to quantum models with respect to *sample complexity* and *average* prediction error. Even if an exponential advantage is still attainable if the goal is to minimize the *worst-case* prediction error, classical methods are again found to be surprisingly powerful when given access even to quantum data.

Quantum-inspired methods. When discussing the effectiveness of quantum neural networks, and in order to achieve a true quantum advantage, one should compare not only to previous quantum models but also to the best available classical alternatives. For this reason, and until fault-tolerant quantum computation is achieved, quantum-inspired procedures on classical computers could pave the way towards the discovery of effective solutions [74–76]. One of the most prominent examples is represented by tensor networks [77], initially developed for the study of quantum many-body systems, which show remarkably good performances for the simulation

of quantum computers in specific regimes [78]. In fact, many proposals for quantum neural networks are inspired by or can be directly rephrased in the language of tensor networks [44,45], so that libraries developed for those applications could be used to run some instances of these quantum algorithms. Fierce competition is therefore to be expected in the coming years between quantum processors and advanced classical methods, including the so-called neural network quantum states [79,80] for the study of many-body physics.

Outlooks. – Quantum machine learning techniques, and particularly quantum neural network models, hold promise of significantly increasing our computational capabilities, unlocking whole new areas of investigation. The main theoretical and practical challenges currently encompass trainability, the identification and suitable treatment of appropriate data sets, and a systematic comparison with classical counterparts. Moreover, with steady progress on the hardware side, significant attention will be devoted in the near future towards experimental demonstrations of practical relevance. In fact, small to medium scale implementations of quantum machine learning techniques have already been tested on actual quantum hardware, such as currently available digital quantum circuits based on superconducting qubits [29,30,32,33,81], nanophotonic architectures [82,83], and devices based on trapped ion qubits [84]. Research evolves rapidly, with seminal results highlighting prominent directions to explore in the quest for quantum computational advantage in AI applications, and new exciting achievements are expected in the years to come.

* * *

This research was partly supported by the Italian Ministry of Education, University and Research (MIUR) through the “Dipartimenti di Eccellenza Program (2018–2022)”, Department of Physics, University of Pavia, the PRIN-2017 project 2017P9FJBS “INPhoPOL”, and by the EU H2020 QuantERA ERA-NET Cofund in Quantum Technologies project QuICHE.

REFERENCES

- [1] GOODFELLOW I., BENGIO Y. and COURVILLE A., *Deep Learning* (MIT Press) 2016.
- [2] HINTON G. E., OSINDERO S. and TEH Y.-W., *Neural Comput.*, **18** (2006) 1527.
- [3] LECUN Y., BENGIO Y. and HINTON G., *Nature*, **521** (2015) 436.
- [4] HASTIE T., TIBSHIRANI R. and FRIEDMAN J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, *Springer Series in Statistics* (Springer) 2009.
- [5] ANDRAE A. S. G. and EDLER T., *Challenges*, **6** (2015) 117.

- [6] LAMATA L., *Mach. Learn.: Sci. Technol.*, **1** (2020) 033002.
- [7] BIAMONTE J. *et al.*, *Nature*, **549** (2017) 195.
- [8] NIELSEN M. A. and CHUANG I. L., *Quantum Computation and Quantum Information*, 10th edition (Cambridge University Press, Cambridge, UK) 2010.
- [9] TACCHINO F. *et al.*, *Adv. Quantum Technol.*, **3** (2020) 1900052.
- [10] BENEDETTI M. *et al.*, *Quantum Sci. Technol.*, **4** (2019) 043001.
- [11] MCCLEAN J. R. *et al.*, *New J. Phys.*, **18** (2016) 023023.
- [12] PERUZZO A. *et al.*, *Nat. Commun.*, **5** (2014) 4213.
- [13] CEREZO M. *et al.*, arXiv:2012.09265 (2020).
- [14] PRESKILL J., *Quantum*, **2** (2018) 79.
- [15] BHARTI K. *et al.*, arXiv:2101.08448 (2021).
- [16] GIL VIDAL F. J. and THEIS D. O., *Front. Phys.*, **8** (2020) 297.
- [17] SCHULD M., SWEKE R. and MEYER J. J., *Phys. Rev. A*, **103** (2021) 032430 (arXiv:2008.08605).
- [18] PÉREZ-SALINAS A. *et al.*, *Quantum*, **4** (2020) 226.
- [19] MITARAI K. *et al.*, *Phys. Rev. A*, **98** (2018) 032309.
- [20] OSTASZEWSKI M., GRANT E. and BENEDETTI M., *Quantum*, **5** (2021) 391.
- [21] HORNIK K., *Neural Netw.*, **4** (1991) 251.
- [22] LAROSE R. and COYLE B., *Phys. Rev. A*, **102** (2020) 032420.
- [23] LLOYD S. *et al.*, arXiv:2001.03622 (2020).
- [24] KILLORAN N. *et al.*, *Phys. Rev. Res.*, **1** (2019) 033063.
- [25] SCHULD M., SINAYSKIY I. and PETRUCCIONE F., *Quantum Inf. Process.*, **13** (2014) 2567.
- [26] SCHULD M., SINAYSKIY I. and PETRUCCIONE F., *Phys. Lett. A*, **379** (2015) 660.
- [27] CAO Y., GUERRESCHI G. G. and ASPURU-GUZIĆ A., arXiv:1711.11240 (2017).
- [28] TORRONTGUI E. and GARCIA-RIPOLL J. J., *EPL*, **125** (2019) 30004.
- [29] TACCHINO F. *et al.*, *npj Quantum Inf.*, **5** (2019) 26.
- [30] MANGINI S. *et al.*, *Mach. Learn.: Sci. Technol.*, **1** (2020) 045008.
- [31] TACCHINO F. *et al.*, *IEEE Trans. Quantum Eng.*, **2** (2021) 3101110 (arXiv:2103.02498).
- [32] TACCHINO F. *et al.*, *Quantum Sci. Technol.*, **5** (2020) 044010.
- [33] HAVLÍČEK V. *et al.*, *Nature*, **567** (2019) 209.
- [34] SCHULD M. and KILLORAN N., *Phys. Rev. Lett.*, **122** (2019) 040504.
- [35] ACKLEY D. H., HINTON G. E. and SEJNOWSKI T. J., *Cogn. Sci.*, **9** (1985) 147.
- [36] AMIN M. H. *et al.*, *Phys. Rev. X*, **8** (2018) 021050.
- [37] ZOUFAL C., LUCCHI A. and WOERNER S., *Quantum Mach. Intell.*, **3** (2021) 7.
- [38] KIEFEROVÁ M. and WIEBE N., *Phys. Rev. A*, **96** (2017) 062327.
- [39] ZOUFAL C., LUCCHI A. and WOERNER S., *npj Quantum Inf.*, **5** (2019) 103.
- [40] LLOYD S. and WEEDBROOK C., *Phys. Rev. Lett.*, **121** (2018) 040502.
- [41] DALLAIRE-DEMERS P.-L. and KILLORAN N., *Phys. Rev. A*, **98** (2018) 012324.
- [42] HUANG H.-L. *et al.*, arXiv:2010.06201 (2020).
- [43] COYLE B. *et al.*, *npj Quantum Inf.*, **6** (2020) 60.
- [44] CONG I., CHOI S. and LUKIN M. D., *Nat. Phys.*, **15** (2019) 1273.
- [45] GRANT E. *et al.*, *npj Quantum Inf.*, **4** (2018) 65.
- [46] BEER K. *et al.*, *Nat. Commun.*, **11** (2020) 808.
- [47] MCCLEAN J. R. *et al.*, *Nat. Commun.*, **9** (2018) 4812.
- [48] CEREZO M. *et al.*, arXiv:2001.00550 (2020).
- [49] MARRERO C. O., KIEFEROVÁ M. and WIEBE N., arXiv:2010.15968 (2020).
- [50] WANG S. *et al.*, arXiv:2007.14384 (2020).
- [51] SHARMA K. *et al.*, arXiv:2005.12458 (2020).
- [52] PESAH A. *et al.*, arXiv:2011.02966 (2020).
- [53] HOLMES Z. *et al.*, arXiv:2101.02138 (2021).
- [54] BITTEL L. and KLIESCH M., arXiv:2101.07267 (2021).
- [55] SKOLIK A. *et al.*, *Quantum Mach. Intell.*, **3** (2021) 5.
- [56] GRANT E. *et al.*, *Quantum*, **3** (2019) 214.
- [57] VOLKOFF T. and COLES P. J., *Quantum Sci. Technol.*, **6** (2021) 025008.
- [58] VERDON G. *et al.*, arXiv:1907.05415 (2019).
- [59] PATTI T. L. *et al.*, arXiv:2012.12658 (2020).
- [60] SCHULD M. *et al.*, *Phys. Rev. A*, **99** (2019) 032331.
- [61] BANCHI L. and CROOKS G. E., *Quantum*, **5** (2021) 386.
- [62] STOKES J. *et al.*, *Quantum*, **4** (2020) 269.
- [63] VERDON G., PYE J. and BROUGHTON M., arXiv:1806.09729 (2018).
- [64] HARROW A. W. and MONTANARO A., *Nature*, **549** (2017) 203.
- [65] ARUTE F. *et al.*, *Nature*, **574** (2019) 505.
- [66] ZHONG H.-S. *et al.*, *Science*, **370** (2020) 1460.
- [67] SWEKE R. *et al.*, arXiv:2007.14451 (2020).
- [68] DU Y. *et al.*, *Phys. Rev. Research*, **2** (2020) 033125.
- [69] WRIGHT L. G. and MCMAHON P. L., arXiv:1908.01364 (2019).
- [70] ABBAS A. *et al.*, arXiv:2011.00027 (2020).
- [71] HUANG H.-Y. *et al.*, arXiv:2011.01938 (2020).
- [72] LIU Y., ARUNACHALAM S. and TEMME K., arXiv:2010.02174 (2020).
- [73] HUANG H.-Y., KUENG R. and PRESKILL J., arXiv:2101.02464 (2021).
- [74] AARONSON S., *Nat. Phys.*, **11** (2015) 291.
- [75] TANG E., *A quantum-inspired classical algorithm for recommendation systems*, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing STOC 2019* (Association for Computing Machinery, New York, NY, USA) 2019, pp. 217–228.
- [76] ARRAZOLA J. M. *et al.*, *Quantum*, **4** (2020) 307.
- [77] MONTANGERO S., *Introduction to Tensor Network Methods* (Springer Nature Switzerland AG, Cham, Switzerland) 2018.
- [78] ZHOU Y., STOUDENMIRE E. M. and WAIN TAL X., *Phys. Rev. X*, **10** (2020) 041038.
- [79] CARLEO G. and TROYER M., *Science*, **355** (2017) 602.
- [80] TORLAI G. *et al.*, *Nat. Phys.*, **14** (2018) 447.
- [81] MARI A. *et al.*, *Quantum*, **4** (2020) 340.
- [82] ARRAZOLA J. M. *et al.*, *Nature*, **591** (2021) 54.
- [83] LAMATA L., *Photonics*, **8** (2021) 33.
- [84] RUDOLPH M. S. *et al.*, arXiv:2012.03924 (2020).