Variational Quantum Classifier

Author: Elies M. Gil Fuster
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.[*]

Advisor: José Ignacio Latorre

Abstract: In this work we propose a quantum alternative to Artificial Neural Networks in classification tasks. We design a set of different neural networks and quantum circuits and test their performances. We found that a Variational Quantum Classifier can outperform a classical model using far less free parameters and, thus, being more efficient. Further, a complex classification task requires deeper quantum circuits, which nevertheless grow at a slower pace than the number of neurons needed in a Neural Network for the same task.

I. INTRODUCTION

In the vast realm of *Machine Learning*, classification tasks carry a highly relevant role with them [1]. In a supervised learning scenario, a structure is trained to recognize labeled subsets within given data. This structure learns the features that together define each label, and once trained, it can be used to sort unlabeled data [2]. There is a plethora of computing paradigms that can be used as learning structures. *Artificial Neural Networks* (ANN) constitute a remarkable approach to such problems, rendering promising results in several cases [3, 4].

A continual effort is also being made in the search for scenarios where quantum solutions pose an advantage over usual classic algorithms. This effort is a search for quantum algorithms which are exponentially faster than their digital counterparts, or exponentially more efficient. If such a breakthrough were achieved, we would talk about *quantum advantage* [5].

It is a natural thought then to propose quantum alternatives to current machine learning algorithms in the field of classification tasks [6]. By reviewing the relevant literature, it appears clear that several incursions in this direction have already been conducted. Many focus on an almost literal translation from some component of classical neural networks to quantum computing ones [7, 8]. This path has indeed proven difficult as one inevitably faces quantum mechanical properties that cannot be avoided [9].

An ideal framework, on the other hand, for a quantum approach would be that of a fault-tolerant quantum computer with an arbitrarily large amount of *quantum bits* (qubits). However, that is still far from realization. The present period has become known as the *Noisy Intermediate-Scale Quantum* era, broadly referred to as the NISQ era [10]. This term illustrates the kind of quantum technology we can rely on having on the coming time. Specifically, this refers to systems with about less than 100 qubits whose quantum logic gates still carry some error with them. Thus, it becomes vital that our algorithms be somewhat resistant to errors.

In this light, the class of algorithms known as *Variational Quantum Algorithms* occupies a privileged position. They behave according to a dual nature [11]: a quantum circuit determined by some scalar parameters and a classic algorithm, which updates those parameters following some rules. These hybrid circuits work in a loop. The quantum circuit initially runs and produces an output. Next, the classic algorithm tries to achieve something on that output by tuning the parameters of the circuit, normally minimizing a function. Then, the process begins again, this time with the new parameters, and it is repeated until some stopping flag is reached, for example, convergence. As an example, there is the so-called *Variational Quantum Eigensolver* [12], an algorithm used to compute the ground state energy of a state.

We propose a model named *Variational Quantum Classifier* (VQC), which behaves under similar heuristics. It involves a quantum circuit and a function defined with its outcome, which will indeed depend on some free parameters. A classic algorithm will reiteratively try to minimize this function. It is submitted that this definition clearly satisfies the fault tolerance requirements since possible errors are absorbed by fine-tuning the circuit parameters.

This paper is structured as follows: Initially, in Section II, we introduce the theoretical models of ANN and VQC, immediately followed by a benchmark of both classification systems in Section III. Finally, we close by outlining our conclusions in Section IV.

II. CLASSICAL AND QUANTUM CLASSIFIERS

This study represents a proof of concept for a quantum learning algorithm that can successfully perform classification. We arrive at this goal by first tackling rather simple exercises, which in turn give useful ideas on how to scale to real problems. With this in mind, two different tasks are composed.

In both tasks, we are given a set of labeled points (x_1, x_2) with $-1 \leq x_i \leq 1$, $i \in \{1, 2\}$. Note this domain is a square centered at $(0, 0)$. Problem one has a

[*] Electronic address: egilfust7.alumnes@ub.edu

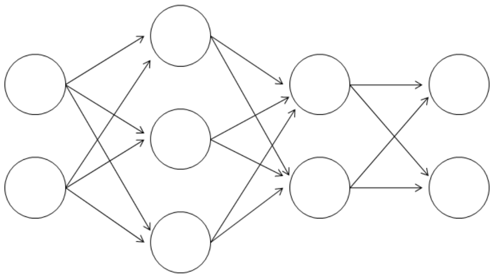

FIG. 1: An ANN with 2 input and output neurons, and two hidden layers with 3 and 2 neurons. Every neuron has a bias value, and every connection a weight, which is tuned to produce an adequate output.

different set of labels to problem two. The set of labels is defining for each task and we will proceed analogously in facing both. We begin by equally splitting the data into two different subsets: a *training set* and a *validation set*. As one may infer, the learning algorithm runs using the first one only and, afterward, the circuit classifies the second one. Eventually, what we will use to compare performances are the points in the validation set, not in the training set. Since we know which label corresponds to each validation point, we can ultimately compute the accuracy of the model's ability to sort the points.

Firstly, a binary classification task is stated: sort the points as to whether they belong to a centered circle or not, this is $x_1^2 + x_2^2 \leq r^2$ for a given r. We define r such that the circle's area covers half the surface of the square. This way, a "dumb" classifier, one which gives the same label to every point, always scores about 50%.

Secondly, we face a classification task with more than two possible outcomes. There will be three different circles in the square and the model must be able to tell in which shape every point is or if the point simply sits outside all the circles.

We will present both problems to a rich set of ANN and VQC. As a result, we are able to compare models and also to benchmark classic against quantum. The following epigraphs introduce what models we use, how they work, and how we acknowledge them.

A. Artificial Neural Networks

Feedforward networks of *sigmoid neurons* offer a range of learning possibilities. An appropriate choice of weights and biases classifies the data set by mimicking the labels of points. We refer to the networks we use with a tuple of integers $[N_1, \ldots, N_L]$. The length of the tuple L expresses how many layers the network has, whereas the entries N_i reveal the number of neurons in each layer. So, for instance, the ANN depicted in Fig. 1 is written $[2, 3, 2, 2]$. The first and last layers are called the *input* and *output layers* respectively; everything in between them is named *hidden layers*.

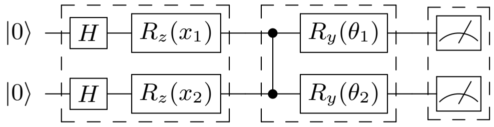

FIG. 2: A Quantum Circuit Diagram that depicts one of the simplest circuits we used. It consists of three parts:*Encoding*, *Classification*, and *Measurement*. Alternative circuits may contain more CZ gates and rotations about other axes.

Due to the level of difficulty in our tasks, we have enough with architectures of sensible size as, for instance, the one in Fig. 1. Since we do not need more than two hidden layers, it is suitable for us to make use of a combination of *Stochastic Gradient Descent* [13] plus *Backpropagation algorithm* [14].

Ultimately, we attempt to determine the best configuration for each given task. In particular, our interest will reside in the most efficient network, i.e., the one with the lowest amount of free parameters that can successfully classify the data.

The input layer, made of two neurons, encodes the data. Every point is characterized by the input values x_1 and x_2. Since the first task consists of binary classification, the output layer will contain two neurons. Conversely, for task two there are four regions into which a point can belong, therefore, we shall use a four neuron output. Every input point has a label $a(x_1, x_2)$, and an output $y(x_1, x_2)$ produced by the ANN. What we want is for the output to be as similar as possible to the label for every point in the set.

Very often in these scenarios, a classification problem becomes one of studying the local minima of a given *cost function*, which gives an idea of how far the model is from perfect classification. It is usually understood as a metric. Different cost functions suit different problems better just as different architectures do. We use the *quadratic cost function*: $C = \frac{1}{2n}\sum_x ||y(x) - a(x)||^2$, which can be thought of as a mean squared Euclidean metric.

The so-called set of *hyperparameters* includes the network design, the choice of cost function, and the choice of variables related to the algorithm itself, which remain fixed during the training. The cost function and the architecture are commonly set according to some general rules of thumb. After those are chosen a pre-training session can take place where the remaining variables are tuned. Oppositely, they may be fixed from the start.

B. Variational Quantum Classifier

Again, we want to use the least possible amount of computational resources. We achieve that by using the simplest quantum gates available. This is why our VQC consists of a number of one-qubit gates: the Hadamard gate H, and rotational gates; plus some entangling gates:

the controlled-NOT and the controlled-Z. All of them are introduced in the Appendix, where one will find their explicit form together with the names used to call them. Our circuits operate on quantum systems initialized in the $|0\rangle^{\otimes n}$ state. The reason behind this is that $|00\ldots0\rangle$ states can be conveniently prepared. Also *almost every* unitary gate acting on n qubits can be decomposed as a set of these basic gates, with a certain precision ε, as stated in the *Solovay-Kitaev theorem* [17].

The angles applied by the rotations, $\theta_1, \ldots, \theta_l$, determine how accurate the classification is, where l is the total number of rotations. The second part of the VQC is fulfilled by a classic algorithm: first, we compute a function using the final state of the quantum circuit, which will, of course, be a function of θ_i. Then, the algorithm tries to minimize this function, much in a similar fashion to that of the ANN.

The quantum circuit we are going to use consists of three main parts, as they appear in Fig. 2.

1. *Encoding:* Prepare a superposed state $H^{\otimes n}|0\rangle^{\otimes n}$, apply rotations of angle equal to the input coordinates.
2. *Classification:* Apply different rotations of angles $\theta_1, \ldots, \theta_l$ and, optionally, include some entanglement gates among them.
3. *Measurement:* Collapse each qubit's wave function to obtain one of the quantum states of the computational basis.

The classical counterpart, completing the hybrid nature of a variational quantum algorithm, also comprises three steps. First, we define a *target function*, which takes the physical final state as an argument and returns an appropriately shaped output. Second, we again use the *quadratic cost function*. Third and last, we use a *gradient descent* algorithm to minimize the cost function.

There are many settings that can be customized when comparing different architectures. Mainly, the differences from one circuit to another lie in what specific rotations we apply, that is, R_x, R_y, or R_z, seen in the Appendix. For an input (x_1, x_2), right after the H gates, we choose to encode using $R_z(x_i\pi)$.

The presence of much higher complexity in the second task requires that we use systems with more qubits. Some reasoning may associate the high performance of neural networks with their redundancy when processing data. The output of every neuron is typically fed to several others in the next layers, so every piece of information is treated by several parallel entities. It is for this reason that it might be logical to introduce redundancy in our quantum classifier, i.e., we will want to introduce the input (x_1, x_2) several times. Hence, we shall use quantum circuits with an even number of qubits, starting with four. Therefore, there is a wider range of options we should be testing. Among the myriad of possible directions, however, our focus remains on classifying with only R_x and R_y rotations, entanglement gates, and an increasing number of qubits. Now, a more sophisticated encoding includes entanglement gates and free parameter rotations mixed with the R_z rotations. Every circuit is repeated and tried with different choices of target function.

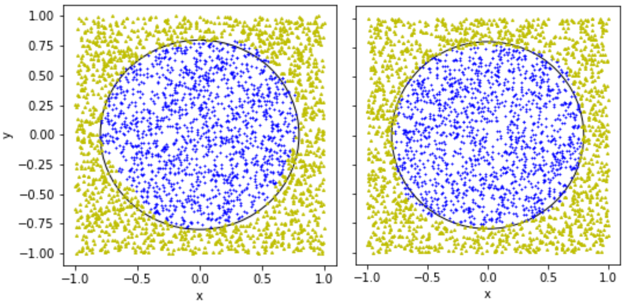

FIG. 3: Binary classification of the points in a circle using ANN vs. VQC, corresponding to the left and right plots, respectively. The classic model requires 23 free parameters, while only 2 already render the quantum result.

Rotation angles are initialized at random and updated during training sessions to minimize the cost function. The optimal amount of entanglement is not a trivial thing, and it is part of the optimization problem [15]. For instance, if we had 4 qubits, first we would entangle the first with the second qubits and the third with the fourth. Subsequently, in order to bond both pairs we would link the first to the fourth and the second to the third qubits. Higher dimensional scenarios require some creativity, but they raise no questions too deep in nature.

III. BENCHMARKING

Illustrative data are presented in Table I. The most promising cases, highlighted in bold, are plotted in Fig. 3. It is clear that quantum alternatives require significantly lower amounts of parameters, but are accompanied by less consistent performances. Moreover, the very nature of quantum mechanics allows parallel operations to be computed at once. This way, the demand for computing resources sinks and the overall efficiency increases as the system grows larger, in contrast to classic alternatives.

By comparing highly scoring models we are able to establish links between both classical and quantum kingdoms. In turn, some isolated examples with higher complexity and poorer performance help us build accurate conceptual limits of the models we use. Completely opposite cases are discovered too, where a particularly simple system fulfills the classifying task with sharp accuracy. Nevertheless, prudence provides simple explanations that tag these experiments as something similar to "false positives".

There is no denying an apparent correlation between the number of parameters and performance. Typically, high complexity should result in successful outcomes. It is indeed among the exceptions to this rule that some

Task	Classifier		Parameters	Phys.Ops.[a]	Best Result
			(#)	(#)	(%)
Binary Circle	ANN	[2,2,2]	12	24	75.97
		[2,3,2]	17	32	94.30
		[2,3,2,2]	**23**	**44**	**96.53**
		[2,3,3,2]	29	53	76.73
	VQC	(2,2,0,0)	2	3	55.03
		(2,0,2,1)	**2**	**4**	**96.63**
		(2,1,1,1)	2	4	75.60
Three Circles	ANN	[2,2,4]	18	36	78.97
		[2,2,5,4]	45	78	82.67
		[2,3,4,4]	45	78	87.40
		[2,5,5,4]	69	111	95.93
	VQC	$(4,2,4,3)_1$	6	7	21.50
		$(4,2,4,3)_2$	6	7	32.00
		(4,4,2,3)	6	7	39.00

[a] Number of physical operations required to classify one point.

TABLE I: Results of the most illustrative scenarios. ANN are encoded with a string of numbers between square brackets $[N_1, \ldots, N_L]$, where L is the number of layers and N_i the number of neurons in layer i. VQC are codified with a 4 figure number inside parentheses (n, X, Y, E), where n is the number of qubits, X and Y are the number of rotational gates about the x and y axes, and E is the count of entangling gates. $(4243)_1$ differs from $(4243)_2$ in the choice of target function, for the same circuit. Plots in Fig. 3 relate to rows highlighted in bold.

of the deepest insights arise. For example note that $[2, 3, 3, 2]$ scores remarkably worse than $[2, 3, 2, 2]$. This and similar instances bring us to the conclusion that too intricate a network will struggle when facing very simple tasks. It is therefore important to properly size the designs we use since the number of parameters grows rather fast with the dimension of the network. It is enlightening to question ourselves about the two ANN with 45 parameters as well as the figures relating to the quantum alternative.

Considering the case with ANN first, sheer reason would say that different architectures should render different results. Moreover, there is a set of heuristics that help to find one's way during the design process. As an example, in tasks similar to ours it makes little sense to have a layer with fewer neurons than both of their neighbor layers. One can guess that the reason for this is that, by doing so, we are losing either information or capacity of processing.

Let us focus for a moment in how VQC deal with the first task. One can see that, for the same number of parameters, accuracy levels are leaps apart from one another. Some might be convinced that a classifier using R_y gates necessarily works better than one based on R_x rotations. Yet, as a matter of fact, it does not. Rather, what is happening is that the particular specifics of task one closely relate to some inner feature of the y-rotation gate. If the task were to characterize a horizontal stripe, our plots say R_x would perform comparably better.

Quantum Classifiers struggle at first while tackling task two. This way, there is a stronger correlation between performance and the number of parameters towards the lower end of the spectrum. This insinuates that the proposed circuits in those experiments might not be complex enough to perceive the behavior of such non-local classification.

It needs to be said, though, that the scope of advantages in the use of quantum circuits reaches further than the number of parameters only. Classical computers are sequential, meaning their capabilities of parallelism are limited. Quite oppositely, quantum computers can perform a number of operations at once. There is a resemblance, thus, between quantum circuits and neural networks, where one can define *layers* of gates. Further steps following this study could well prove fruitful in this direction.

IV. CONCLUSIONS

Our goal was to test whether a variational quantum algorithm was a valid classification model. We achieved it by defining a pair of problems and testing them with both the ANN and VQC models. We arranged a rich set of networks using different architectures and sizes, which were then trained to solve the problems. In parallel, we designed different quantum circuit designs depending on some parameters, which we also trained with the same purpose. We finally compared the performances of both models.

Overall, VQC present themselves as a potentially valid classification model. As opposed to ANN, quantum alternatives use a significantly lower number of parameters, which implies a lower energy consumption and are, therefore, more efficient. Moreover, classical computers work sequentially, contrary to the ones introduced in this paper. Therefore, there is room for optimism in the search for *quantum advantage* in the direction of variational algorithms.

Given this, there are some clear areas in which future study should be considered. Namely, a remarkable discovery would be listing heuristics and general rules linking circuit design with the key features of problems since this would bring us closer to better results in the long run. Light must also be shed on the selection of target functions, as one could expect that a smart choice would make better scoring possible. Also, it might be interesting to pursue the building of thought bridges between the defining elements of neural networks and some features of variational circuits, so a clearer intuition can be drawn from the parallelism.

Acknowledgments

I would like to thank my advisor Dr. José Ignacio Latorre and the BSC-QUANTIC group for their extremely professional guidance. I am especially thankful to my supervisor Alba Cervera for taking me under her wing.

APPENDIX: QUANTUM GATES

In Quantum Computing, the system undergoes changes in a similar fashion as in classical computing. One can understand operations on a string of qubits as a collection of logic gates, such as the NAND or the XOR ones [16]. Quantum Algorithms have a possible similar interpretation, where gates are unitary operators, elements of the $U(2^n, \mathbb{C})$ unitary group. Under this algebraic formalism, the base states are vectorized as:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{and} \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{1}$$

which means we are considering the projections on the z-axis.

The *Solovay-Kitaev Theorem* states that a finite set of basic gates is enough to efficiently approximate any n-qubits gate with ε accuracy [17]. Among the finite sets that fulfill said property, there is one whose gates can be implemented in a fault-tolerant way [18]. With them, we can produce any gate, operating over an arbitrary number of qubits, with fixed precision ε. Gates which have appeared throughout the present work are commonly explicitly expressed in the following way:

- Hadamard.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2}$$

- x-Rotation.

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{3}$$

- y-Rotation.

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{4}$$

- z-Rotation.

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{5}$$

- Controlled-NOT.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{6}$$

- Controlled-Z.

$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{7}$$

As said, all of them can be efficiently approximated using only a comfortably small set of gates. For instance, we use CNOT and CZ almost indistinctly because they are somewhat equivalent:

$$(\mathbb{I} \otimes H)\text{CNOT}(\mathbb{I} \otimes H) = \text{CZ}, \tag{8}$$

where $\mathbb{I}$ is the identity operator.

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, (Springer, Cambridge, 2006).

[2] J. Allcock, C. Hsieh, I. Kerenidis, and S. Zhang, arXiv:1812.03089 (2018).

[3] M. A. Nielsen, *Neural Networks and Deep Learning*, Determination Press (2015).

[4] F. Lluís, J. Pons, and X. Serra, arXiv: 1810.12187 (2018).

[5] S. Boixo *et al.*, Nat. Phys. **14**, 595 (2018).

[6] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv: 1307.0411 (2013).

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, Contemp. Phys. **56**, 172 (2015).

[8] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Lett. A **379**, 660 (2015).

[9] M. V. Altaisky, arXiv: quant-ph/0107012 (2001).

[10] J. Preskill, Quantum **2**, 79 (2018).

[11] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New J. Phys. **18**, 023023 (2016).

[12] A. Peruzzo, *et al.*, Nat. Commun. **5**, 4213 (2014).

[13] L. Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent*, (Springer-Verlag, Berlin, 2010).

[14] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Nature **323**, 533 (1968).

[15] J. Preskill, arXiv: 1203.5813 (2012).

[16] R. C. Jaeger and T. N. Blalock, *Microelectronic Circuit Design*, (McGraw-Hill, New York, 1997).

[17] C. M. Dawson, and M. A. Nielsen, Quantum Inf. and Comput. **6**, 81 (2006).

[18] M. A. Nielsen, and I. Chuang, *Quantum Computation and Quantum Information*, (Cambridge University Press, Cambridge, 2002).