A Hybrid Quantum-Classical Framework For Binary Classification In Online Learning

Corrado Loglisci*, Ivan Diliso and Donato Malerba

Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Abstract

Quantum machine learning recently gained prominence due to the promise of quantum computers in solving machine learning problems that are intractable on a classical computer. Nevertheless, several studies on problems which remain challenging for classical computing algorithms are emerging. One of these is classifying continuously incoming data instances in online fashion, which is studied in this paper through hybrid computational solution that combines classical and quantum techniques. Hybrid approaches represents one of the current ways that opens to the quantum computation in practical applications.

In this paper, we show how typical issues of online learning can be equally addressed with the properties of quantum mechanics, until to offer better results. We aim at keeping the classification capabilities, which have learned on previously processed data instances, preserved as much as possible, and then acquiring new knowledge on new data instances. To this end, we propose a class of quantum neural networks, variational quantum circuits, that being adapted over time by exploiting techniques of model update used in classical neural networks. Experiments are performed on real-world datasets with quantum simulators.

Keywords

Quantum machine learning, Hybrid quantum-classical framework, Online learning, Variational quantum circuits

1. Introduction

Nowadays, Machine Learning models are employed in almost every possible task, including medical diagnoses, fraud detection, and marketing. The widespread adoption of machine learning in various fields can be attributed to the availability of relatively powerful computers in recent times. The computational cost, especially for Deep Learning methods, can be extremely high, necessitating training times of several hours, days, or even months on present-day computers [1]. Furthermore, traditional computers are reaching their physical limits, which will impede their progress in the coming years [2]. As a result, many researchers are investigating alternative computing platforms for training ML models. Among these, quantum computers have emerged as an intriguing option. On the side of Quantum Computing (QC), the current status sees the era of noisy intermediate scale quantum (NISQ) computers [3], which are devices able to deal with low-middle size data problems. An approach which seems bringing practical advantages is the one of the *hybrid* solutions [4], which combine classical and quantum methods

*Corresponding author.

✉ corrado.loglisci@uniba.it (C. Loglisci)

CEUR Workshop Proceedings (CEUR-WS.org)

and allow to exploit quantum physics properties. It is the current standard modality to design quantum software while overcoming existing limitations of quantum devices, such as noisy and decoherence. Clearly, these are not yet the technologies which will guarantee exceptional speed-ups to large data sizes over classical computing, but pave the way to the development of near future algorithms for data-intensive problems. Actually, when the complexity regards the tractability of the problems rather than the scalability to data volumes, the current quantum routines turn out to be already useful, for instance, in cryptography [5].

One of the categories of data-intensive problems in which the research on classical computing has still struggling with is the one of learning from sequences of continuously incoming data. Even the accurate solutions of Deep Learning find challenging working on that data scenario. This is demonstrated by the recent research line addressing the so-called *catastrophic forgetting* [6], which is the tendency of an artificial neural network to abruptly and drastically forget previously learned information upon learning new information. In those cases, it is not important designing algorithms for massive computation, but keeping the quality of the models high over unbounded sequences of data.

Due to the intrinsic dynamic nature of sequential data, it is not immediate designing effective technologies to learn models that may provide accurate predictions [7]. Indeed, the data distribution can change, that is, the statistical properties of the data-generating process can change. Also, possible inter-relationships between descriptive features can evolve and concern other features, as well as, the concept (or class) underlying the data can involve new value ranges. This can threaten the predictive capabilities of the models, in fact, models that were accurate before may not do the same next. Therefore, the algorithms should guarantee robustness to such changes and, to do that, the models be updated by picking new data characteristics. This cannot be done by collecting data and training models only once or in a shot, but, building the models progressively as new worthwhile data are available, in a sort of incremental mode. Problems of such a nature are investigated in *Online learning* [8]. There, models are trained at the beginning and adapted on new incoming data during training sessions, while preserving knowledge learned from previously training sessions, without counting on the availability of the previously processed data and without the need to retrain from scratch at the arrival of new data. This way, models are updated and ready to work on prediction sessions.

QC has the potential and capabilities to reply to these issues and provide alternative computational solutions. Indeed, classical data instances can be represented as quantum states corresponding to points of the Hilbert space, which has more dimensions than the original space. The features of classical data (for instance, a feature vector with three entries) can be blow up into a representational space larger than the classical one (eight dimensions, resulting from 2^3, three is the feature vector size)[9]. However, this kind of representation does not requires addressing sparsity data problem, such as embedding functions, because the operators on quantum states (that is, matrices) inherently handle such a dimensionality. QC offers the possibility to naturally account for inter-relationships between both data instances and features thanks to the property of the *entanglement*. It refers to the phenomenon in which two or more quantum particles become so strongly correlated that their quantum states are no longer independent of one another. Another property of QC is the *interference*, which has not to be confused with entanglement. Interference refers to the phenomenon that occurs when two or more quantum states overlap and interact with each other, which can imply changes of probability, resulting

in the fact that one of the quantum states gains likelihood to determine the final outcome, by excluding the other one. It handles concurrent and equi-probable outcomes and has the ability to bias models toward the desired outcome, when the computation leads to wrong outcomes. By operating within the realm of classical statistical theory, classical machine learning does not offer these properties, unless models and algorithms try to explicitly accommodate them, with additional computational costs, like in the case of the correlation and interactions within data [10].

We leverage these properties to design a quantum-classical framework which builds a classification model in supervised setting and works in online manner by acquiring continuously incoming data instances. the framework adapts continually a classification model and keeps on learning over time. More precisely, it trains and updates a classifier on (sub-)sequences of incoming data instances (data blocks) marked as labelled, while uses it to estimate the class-value of unlabelled data instances. To do that, we resort to a form of quantum models able with learning capabilities, the so called *variational circuits*, which are quantum algorithms characterized by parameters that are *varied* and optimized as new data instances are acquired, in the same style of of the neural models training.

The framework has been tested on the binary classification task by using two real-world datasets. It has been also compared against a classical online data learner. The experimental results are encouraging and show the potential superiority in terms of accurate estimations over different experimental configurations.

2. Basics on Quantum computing

In QC, data correspond to quantum states, described in terms of qubits and represented by vectors of complex numbers. Computations are represented by quantum gates and performed through matrices. The quantum states of a n-qubits register are represented by a 2^n-dimensional complex vectors of the Hilbert space $\mathcal{S} = (\mathbb{C}^2)^{\otimes n}$, while the quantum states of single qubits are vectors of the Hilbert space $\mathcal{S} = \mathbb{C}^2$ and can be formulated as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (in Dirac notation), where $|0\rangle$ and $|1\rangle$ are the computational basis states of the space $\mathbb{C}^2$ that correspond to the vectors [1 0] (α is valued, β is zero) and [0 1] (β is valued, α is zero) respectively. The coefficients α and β are complex numbers called as *amplitudes*.

A quantum state can be modified by gates, which are implemented by unitary matrices operating within $\mathcal{S}$. For instance, the gates which work on single qubits are $2{\times}2$ matrices. In the following, we have the matrix representations of four single qubit gates, *Hadamard (H)*, *Rotation Y-gate (R_y)*, *Rotation Z-gate (R_z)*, *Pauli Z-gate (σ_z)* respectively and the gate *CNOT (CX)* working on two qbits:

$$H : 1/\sqrt{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad R_y(\theta) : \begin{bmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{bmatrix}$$

$$CX : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad R_z(\theta) : \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \quad \sigma_z : \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

It should be noted that the gates R_y and R_z are parameterized on θ. This corresponds to rotate the vector representing the quantum state, by degrees corresponding to θ, around the
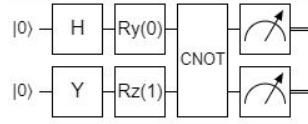
Figure 1: The representation with wires and gates of a quantum circuit, in which the gates are performed from left to right and lead the quantum states of the qubits to evolve.

y-axis and z-axis, within the tri-dimensional space of the Bloch sphere [11].

Algorithms are circuits with n-qubits and consist of quantum gates which operate on the quantum state of the n-qubits. By considering the qubits individually, the circuit consists of a sequence of quantum gates that leads the quantum state of the qubit to evolve. Intuitively, the matrix calculation of the first gate of the circuit is applied to the vector-representation of the qubit, then, the result becomes a (new) quantum state on which the second gate will be applied to. A concrete illustration is reported in Figure 1: the qubit $|0\rangle$ at the top of the circuit is manipulated first by the gate H (we denote as $H|0\rangle$) and then by the gate R_y (we denote as $R_y H|0\rangle$). While, by considering a n-qubit register, the circuit consists of the application in parallel of sequences of quantum gates on individual qubits. This corresponds to the application of tensor-products of gates on n-qubits, which leads the quantum state of the n-qubit register to evolve. For instance, in Figure 1, the input two-qubit register $|00\rangle$ ($|0\rangle \otimes |0\rangle$) is manipulated first by the tensor-product between gate H and gate Y (we denote as $(H \otimes Y)|00\rangle$) and then by the tensor-product between gate $R_y(0)$ and gate $R_z(1)$ (we denote as $R_y(0) \otimes R_z(1))(H \otimes Y)|00\rangle$).

At the end of a circuit, the final operation to perform on the qubits is the *measurement* (the meter symbol in Figure 1), which, in this paper, is finalized to convert quantum states to bitstrings and then to class labels. As result of the measurement, the qubits *collapses* to one of two possible basis states, while the amplitudes indicate the likelihood to collapse in each of the basis states. For instance, in the case of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where the basis states are $|0\rangle$ and $|1\rangle$, the amplitudes α and β indicate the square root of the probability that the qubit measures as either $|0\rangle$ or $|1\rangle$: when the qubit collapses to $|0\rangle$, it returns the quantum state [1 0], while it returns [0 1] if it collapses to $|1\rangle$.

A variational quantum circuit $U(\theta)$ can be represented by a quantum circuit with adjustable real-valued parameters θ (like those of the gate $R_y(\theta)$). This kind of quantum circuits is similar to neural networks and the gate parameters are optimized by classical optimization techniques, such as gradient descent. Typically, a variational quantum circuit $U(\theta)$ can be represented by a has a layered architecture, where the layers are composed of single-qubit gates and two-qubit gates. The same structure of gates is used for all the layers, which are repeated to engage the parameterized gates in a better learning process. The parameters of the gates for all the layers are updated based on the difference between the estimated class label and ground truth. For further details the reader can refer to [11].
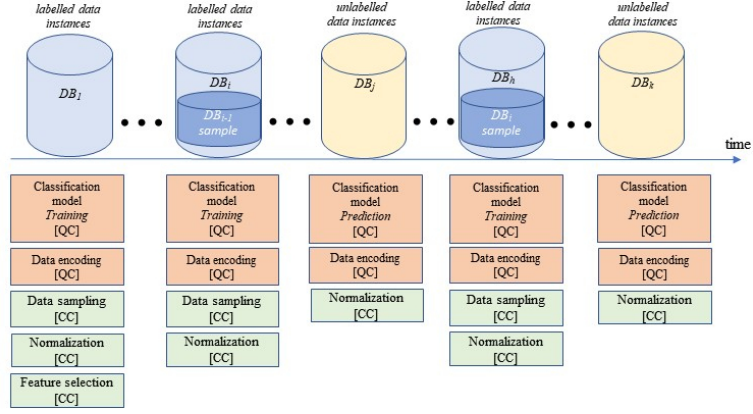
Figure 2: The components of the proposed hybrid framework as they are run in online learning setting for binary classification.

3. Quantum-classical framework for binary classification

The overall framework (illustrated in Figure 2) combines classical computing techniques of feature selection, data sampling, normalization and model optimization with quantum neural models in the form of variational quantum circuits for the problem of binary classification. To distinguish the ones from the others, in Figure 2, they are tagged as either [CC] or [QC]. In the binary classification setting, we have data instances described by $\mathbf{X} \cup y$, $\mathbf{X}$ are values of the set $\mathcal{X}$ of descriptive attributes/features, while $y \in \{-1, 1\}$ denotes the class label.

Feature selection. The component is used only at the beginning and selects the subset of descriptive features which will consider afterwards. It exploits a technique based on the mutual information between the class labels. This operation has been used to perceive the most relevant characteristics and alleviate the problem of choosing the dimension of the quantum circuits in terms of qubits. In fact, the number of the features of the input data determines the number of qubits.

Normalization. The component is used to scale the values of the previously selected features within the range of [0,1] by using the standard min-max function on the original ranges. It is performed for each incoming data block, both those of training and those of prediction.

Data sampling. The component is used to select a subset of the labelled data instances included within the previously processed data blocks. More precisely, when building the data block DB_{i+1}, the component takes a number of data instances equal to the sample size from the data block DB_i (see Figure 2), previously used for a training session. The samples will contain data instances of both the class labels and, for each class label, the component takes data instances with simple random techniques without replacement. The sample size is fixed

Classification model. The component includes two quantum circuits and presents a number of qubits determined by the number features selected by the component Feature selection. In the following, we report the technical details of the two circuits.

- The first quantum circuit takes the classical data and represents them as quantum states to be assigned to the qubits. Initially, all the qubits are set as $|0\rangle$, which is the default

value of the quantum states at the beginning of any circuit. More precisely, this circuit implements a feature mapping operation $\mathcal{F}$ which encodes real-valued data instances $\mathbf{X}$ into quantum states spanning d qubits:

$$|\psi(\mathbf{X})\rangle = \mathcal{F}(\mathbf{X})|0\rangle^{\otimes d} \tag{1}$$

where, $|0\rangle^{\otimes d}$ denotes the register with d-qubits at the state $|0\rangle$ ($|0\rangle \otimes \ldots \otimes |0\rangle$).
In this work, $\mathcal{F}$ has been implemented as follows

$$R_z{}^{\otimes d}(\mathbf{X})H^{\otimes d}|0\rangle^{\otimes d} \tag{2}$$

where, the parameter for each gate R_z is the normalized real-valued of of the feature (corresponding to the qubit on which R_z works). The term $H^{\otimes d}$ denotes the tensor product H $\otimes \ldots \otimes$H over d occurrences (that is, the number of selected features) of the gate H (the same holds for R_z).
This ensures each possible input $\mathbf{X}$ has a unique qubit encoding before being passed to next gates. Clearly, to acquire all the data instances of the data block, the gates of the feature mapping are not replicated along the circuit, instead the input data are enqueued to the circuit as they arrive. We are aware that more sophisticated feature maps could be used [12], but on modern quantum machines, it is also important to use a feature map with limited numbers of gates, since each additional gate could introduce noise into the quantum states.

• The second circuit is variational and manipulates the quantum states returned by the first circuit. It implements a quantum neural network composed of layers of entangled rotation gates. Generally, entangled rotation gates are matrix operations which combine the gates Hadamard, CNOT and Rotation under the quantum physics effect of the entanglement [11]. The second circuit with the first completes the structure of gates which builds the classifier ϕ:

$$|\phi(\mathbf{X}, \theta)\rangle = \mathcal{V}(\theta)|\psi(\mathbf{X})\rangle \tag{3}$$

where, $\mathcal{V}$ is the variational circuit, θ denotes the parameters of the parameterized gates that being optimized.
In this work, $\mathcal{V}$ has been implemented as follows

$$R_y{}^{\otimes d}CX^{\otimes d}R_y{}^{\otimes d} \tag{4}$$

where, each occurrence of the two-qubit gate CX takes one pair of qubits (over the d-qubit register) composed by the consecutive qubits indexed as i and $i + 1$.

Finally, we perform measurements on the qubits and the measured state is recorded. Indeed, we record a collection of measured states because, in quantum machine learning [13], when training a classifier, the variational quantum circuit is run many times with the same input x and parameter θ. So, we can estimate the expectation value of the circuit on x and θ, over multiple runs, with the following:

$$|\mathcal{E}(\mathbf{X}, \theta) = \langle \phi(\mathbf{X}, \theta)|\sigma_z^{\otimes d}|\phi(\mathbf{X}, \theta)\rangle \tag{5}$$

where, $\sigma_z^{\otimes d}$ is the tensor product of the single qubit gate σ_z over d occurrences. As anticipated in Section 2, the notation $|\phi\rangle$ refers to a column vector of $\mathcal{S}$, while the notation $\langle\phi|$ refers to the row column calculated as the transpose conjugate of $|\phi\rangle$.

The gate σ_z has the interesting property that if the measured quantum state has odd parity, it returns -1 (as eigenvalue), while, if the measured quantum state has even parity, it returns 1. This implies that the expectation value of the circuit will always be within the interval $[-1, 1]$. We can use this property to relate the expectation value to the probability that a data instance $\mathbf{X}$ being assigned to a class label y, that is :

$$P(y|\mathbf{X}) = \frac{y\mathcal{E}(\mathbf{X}, \theta) + 1}{2} \tag{6}$$

The probability P(y|$\mathbf{X}$) is exploited in the optimization process concerning the parameters θ. In particular, the optimizer iteratively updates the circuit parameters by minimizing a cost function, which accounts for the negative log-likelihood of the probabilities P(y|$\mathbf{X}$) computed on the current labelled data-blocks, that is:

$$-\frac{1}{size} \sum_{i=1}^{size} log(P(y_i|\mathbf{X}_i)) \tag{7}$$

where, $size$ is the number of data instances of the data block.

The cost function is minimized by a classical computing optimizer based on gradient descent. The derivative concerns the expectation value $\mathcal{E}()$ with respect to the current values of θ and is computed by means of the parameter shift rule [14]:

$$\frac{d\mathcal{E}}{d\theta} = \frac{\mathcal{E}(\theta + \epsilon) - \mathcal{E}(\theta - \epsilon)}{2} \tag{8}$$

The gradient value is the difference between the two output values of the circuit: the first value is the output of the circuit with the parameter θ_k increased by a value ϵ, and the second value is the parameter θ_k decreased by ϵ. Intuitively, the gradient is determined by running the circuit on the same input with two different automatically-computed configurations of the parameters.

Methodology of the framework. Learning classification models on continuously incoming data can be faced with time-windows models [15]. Time-windows models allows us to handle data instances by equally-sized blocks on which we train, update and apply the predictive capabilities of the classifier. Thus, we distinguish training sessions from prediction sessions. During a training session, the component Classification model is activated, which implies the execution the feature mapping $\mathcal{F}$ on the data instances of the current data block and optimization process of the parameters θ of the variational circuit $\mathcal{V}$. During a prediction session, the component Classification model is activated only to estimate the class labels on the current data block by using the classifier up there updated.

To keep the classifier updated, we have to deal with the catastrophic forgetting effect raising when updating neural networks. In the literature, three alternatives are mainly suggested, *replay methods, regularization-based methods, parameter isolation methods* [16]. Shortly, replay methods store a limited set of data instances which are replayed while learning on a new training session. Regularization-based methods introduce regularization term in the loss function, consolidating for new training session the previously learned knowledge. Parameter isolation methods introduce new hyper-parameters for each new training session. Considering that the replay methods represent the solution which asks for less and leave unchanged the number of hyper-parameters of the neural network, we lean for this approach when updating the classifier.

The framework operates in three steps, namely *initialization*, *update*, *prediction*. Training sessions are performed at the initialization and update steps. In the initialization step, the classifier is trained from scratch on the first data block DB_1 (Figure 2). The component *Feature selection* is used only at the initialization step. The other steps work on the feature here selected. Then, the framework prepares the update step and prediction step by collecting labelled data instances and unlabelled data instances for two different data blocks DB_i and DB_j respectively. Actually, the data block DB_i being already populated with the data instances provided by the component of *Data sampling*, while, those necessary to reach the size are taken from the incoming data instances, as they arrive. As explained above, this is done to mitigate the effect of catastrophic forgetting. As soon as one of the two data blocks has been filled (the number of collected data instances is equal to the predefined size), the respective step is performed. By supposing the data block DB_i of labelled data instances is completed for first, the update step will be performed, otherwise it will be the turn of prediction step working on DB_j. Clearly, the data block DB_j contains (unlabelled) data instances, as they arrive.

The succession of training sessions and prediction sessions is not predefined, coherently with the realistic assumption according to which the distribution of labelled and unlabelled data instances is not previously established and therefore not all the data instances are labelled.

4. Experiments on real-world datasets

We implemented the proposed framework in IBM Qiskit [17] and run experiments by using simulators on two real-world datasets, more precisely *Ozone level detection* [1] (having 2536 data instances, 73 features) and *Spambase* (having 4600 data instances, 57 features) [2]. Data blocks have been partitioned so as having a portion of 75% of the dataset as labelled data instances (training sessions) and the remaining 25% as unlabelled data instances (prediction sessions and testing sets of the evaluation). The classical computing components described in Section 3 are those available in the toolkit Scikit-learn [18]. The number of runs of the classification model to estimate the expectation values is 1024, while the number of iterations (epochs) to optimize the parameters is 20. The number of layers for the variational quantum circuit is 3. The size of the sample (component Data sampling) is 20% the data-block size.

The experiments have been performed to emphasize the impact of the technical configuration of the framework on the accuracy, namely number of qubits (corresponding to the features

---

[1]https://archive.ics.uci.edu/ml/datasets/ozone+level+detection
[2]https://archive.ics.uci.edu/ml/datasets/Spambase

selected) and size of the data blocks (number of data instances in each training/prediction session). In Table 1, we report the accuracy of the proposed framework ($HYQOL$) compared to i) a classical computing solution (CC, originally designed for data stream learning) [19] and ii) a baseline of the framework that works on the whole dataset (FQC). The values illustrated have been computed as the average computed over the data blocks of testing. As we can see, except two trials, HYQOL does not never worst than CC, even when the number of qubit is the higher (i.e., 10). Also, we note that the configurations of HYQOL with smallest set of qubits (i.e., 2) are better than those with largest set (i.e., 10), without, however, particular discrepancy between the two endpoints. The size of the data blocks seems not be determinant for the accuracies, but, it is evident that online learning can be beneficial for quantum-based classifiers compared to the version that works on the whole dataset (FQC).

Table 1
Accuracy (in [0,1] of the proposed framework against a classical computing solution and a hybrid solution without online learning. Dataset *Ozone level* at the top. Dataset *Spambase* at the bottom.)

data-block size		#qubits		
		2	5	10
50	HYQOL	0,9	0,9	0,9
	CC	0,9	0,9	0,9
100	HYQOL	0,93	0,89	0,86
	CC	0,87	0,87	0,87
200	HYQOL	0,94	0,94	0,91
	CC	0,92	0,92	0,92
400	HYQOL	0,92	0,91	0,9
	CC	0,9	0,9	0,9
	FQC	0,9	0,87	0,81
data-block size		#qubits		
		2	5	10
50	HYQOL	0,88	0,84	0,84
	CC	0,82	0,82	0,82
100	HYQOL	0,87	0,87	0,85
	CC	0,82	0,82	0,82
200	HYQOL	0,9	0,89	0,8
	CC	0,89	0,89	0,89
400	HYQOL	0,84	0,84	0,84
	CC	0,86	0,9	0,84
	FQC	0,83	0,83	0,8

5. Related work

Hybrid quantum-classical machine learning methods have recently garnered significant attention due to their potential to leverage the strengths of both classical and quantum computing in order to solve computationally intractable problems in a more efficient manner. These methods aim to harness the power of QC to perform certain tasks while still relying on classical computing for other tasks, such as data preprocessing or postprocessing of results.

In [20], the authors proposed a hybrid quantum-classical convolutional Neural Networks model for prediction by on X-Ray images. The quantum part of the model consists of encoding, random quantum circuit, and decoding phases. The hybrid model delivers high accuracy and outperforms previous studies (classical machine learning models approaches) in sensitivity and

F1-measure. A hybrid quantum CNN model was also introduced in [21], where the authors adopted a federated learning approach to protect models and avoid privacy failures attacks. Their experiment results show that the models with additional quantum convolution have slightly improved accuracy than the baseline classical models.

A hybrid quantum-classical model of Long short-term memory (LSTM), a kind of recurrent neural network (RNN), is proposed in [22]. The authors performed a study comparing their proposed hybrid model's capability and performance with its classical counterpart. They found that the hybrid model converges faster and reaches a better accuracy than its classical counterpart. However, their simulations assumed conditions of absence of noise and decoherence. A hybrid quantum-classical approach was designed for generative adversarial learning in [23], in order to develop an anomaly/fraudulent transaction detection. The performances of their model were on par with the classical counterpart in terms of the F1 score.

One of the very few works which investigates the catastrophic forgetting with quantum algorithms has been recently published [24]. It focuses on the incremental learning, which, contrary to online learning, considers training sessions on different classification tasks. Inspired by the replay methods, they propose to constrains the model updated by projecting the gradient direction on the region outlined by previous task gradients. This is done also by storing a fraction of the training data of previous tasks on which the gradient descent is computed. A drawback is the necessity of computing gradients of previous tasks at each training iteration.

Overall, the recent works in the field of hybrid quantum-classical machine learning methods demonstrate the potential for these approaches to outperform classical machine learning algorithms on certain tasks, particularly those that involve large or complex datasets. As quantum computing technology continues to improve, it is likely that we will see further advancements in this field and the development of new, more powerful hybrid quantum-classical algorithms.

6. Conclusions

In this paper, we investigated the viability of quantum machine learning solutions to work on the realistic scenarios of changeability of the statistical properties of the data, which often implies the variability of the performances of the model. We conjecture this can be a machine learning problem in which the quantum solutions can lead innovation. On simulated quantum machines, the hybrid quantum-classical proposal offers encouraging results, in terms of accuracy, often better than a classical computing solution working on data stream and hybrid solution working in batch mode (no online learning). As our opinion, three take-home messages can be identified from this paper. The first one is methodological, in that the online learning opens to practical applications able to combine quantum computing and classical computing techniques, which is likely the only way to concretely use current quantum technologies. The second one is experimental, in that it provides arguments on the fact that stable quantum devices could even do better in terms of performances and quality of the results, when used in predictive tasks. The third one tell us that, although the high-performance computation and tractability of hard problems are the promises of quantum computing which, with the current devices, often are not kept, the research on the lifelong computation can be a field in which quantum computing can already bring interesting results..

Acknowledgment

References

[1] N. C. Thompson, K. Greenewald, K. Lee, G. F. Manso, The computational limits of deep learning, MIT INITIATIVE ON THE DIGITAL ECONOMY RESEARCH BRIEF 4 (2020). doi:10.1140/epjqt/s40507-021-00091-1.

[2] F. Peper, The end of moore's law: Opportunities for natural computing?, New Gen. Comput. 35 (2017) 253–269. doi:10.1007/s00354-017-0020-4.

[3] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum 2 (2018) 79. URL: https://doi.org/10.22331/q-2018-08-06-79. doi:10.22331/q-2018-08-06-79.

[4] A. Callison, N. Chancellor, Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond, Phys. Rev. A 106 (2022) 010101. URL: https://link.aps.org/doi/10.1103/PhysRevA.106.010101. doi:10.1103/PhysRevA.106.010101.

[5] Bova, Francesco, Goldfarb, Avi, Melko, Roger G., Commercial applications of quantum computing, EPJ Quantum Technol. 8 (2021) 2. URL: https://doi.org/10.1140/epjqt/s40507-021-00091-1. doi:10.1140/epjqt/s40507-021-00091-1.

[6] J. Peng, B. Tang, H. Jiang, Z. Li, Y. Lei, T. Lin, H. Li, Overcoming long-term catastrophic forgetting through adversarial neural pruning and synaptic consolidation, IEEE Trans. Neural Networks Learn. Syst. 33 (2022) 4243–4256. URL: https://doi.org/10.1109/TNNLS.2021.3056201. doi:10.1109/TNNLS.2021.3056201.

[7] C. Loglisci, M. Ceci, D. Malerba, Discovering evolution chains in dynamic networks, in: A. Appice, M. Ceci, C. Loglisci, G. Manco, E. Masciari, Z. W. Ras (Eds.), New Frontiers in Mining Complex Patterns - First International Workshop, NFMCP 2012, Held in Conjunction with ECML/PKDD 2012, Bristol, UK, September 24, 2012, Revised Selected Papers, volume 7765 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 185–199. URL: https://doi.org/10.1007/978-3-642-37382-4_13. doi:10.1007/978-3-642-37382-4_13.

[8] S. C. H. Hoi, D. Sahoo, J. Lu, P. Zhao, Online learning: A comprehensive survey, Neurocomputing 459 (2021) 249–289. URL: https://doi.org/10.1016/j.neucom.2021.04.112. doi:10.1016/j.neucom.2021.04.112.

[9] T. Hur, L. Kim, D. K. Park, Quantum convolutional neural network for classical data classification, Quantum Mach. Intell. 4 (2022). URL: https://doi.org/10.1007/s42484-021-00061-x. doi:10.1007/s42484-021-00061-x.

[10] C. Loglisci, Using interactions and dynamics for mining groups of moving objects from trajectory data, Int. J. Geogr. Inf. Sci. 32 (2018) 1436–1468. URL: https://doi.org/10.1080/13658816.2017.1416473. doi:10.1080/13658816.2017.1416473.

[11] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.

[12] H. Yano, Y. Suzuki, R. Raymond, N. Yamamoto, Efficient discrete feature encoding for variational quantum classifier, in: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), 2020, pp. 11–21. doi:10.1109/QCE49297.2020.00012.

[13] D. Arthur, P. Date, Hybrid quantum-classical neural networks, in: IEEE International Conference on Quantum Computing and Engineering, QCE 2022, Broomfield, CO, USA, September 18-23, 2022, 2022, pp. 49–55. doi:10.1109/QCE53715.2022.00023.

[14] D. Wierichs, J. Izaac, C. Wang, C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, Quantum 6 (2022) 677. URL: https://doi.org/10.22331/q-2022-03-30-677. doi:10.22331/q-2022-03-30-677.

[15] M. Ceci, A. Appice, C. Loglisci, C. Caruso, F. Fumarola, D. Malerba, Novelty detection from evolving complex data streams with time windows, in: J. Rauch, Z. W. Ras, P. Berka, T. Elomaa (Eds.), Foundations of Intelligent Systems, 18th International Symposium, ISMIS 2009, Prague, Czech Republic, September 14-17, 2009. Proceedings, volume 5722 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 563–572. doi:10.1007/978-3-642-04125-9_59.

[16] M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 3366–3385. URL: https://doi.org/10.1109/TPAMI.2021.3057446. doi:10.1109/TPAMI.2021.3057446.

[17] M. S. Anis, et al, Qiskit: An open-source framework for quantum computing, 2021. doi:10.5281/zenodo.2573505.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[19] J. Montiel, J. Read, A. Bifet, T. Abdessalem, Scikit-multiflow: A multi-output streaming framework, Journal of Machine Learning Research 19 (2018) 1–5. URL: http://jmlr.org/papers/v19/18-251.html.

[20] E. H. Houssein, Z. Abohashima, M. Elhoseny, W. M. Mohamed, Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images, Journal of Computational Design and Engineering 9 (2022) 343–363. doi:10.1093/jcde/qwac003.

[21] C.-H. H. Yang, J. Qi, S. Y.-C. Chen, P.-Y. Chen, S. M. Siniscalchi, X. Ma, C.-H. Lee, Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6523–6527. doi:10.1109/ICASSP39728.2021.9413453.

[22] S. Y. Chen, S. Yoo, Y. L. Fang, Quantum long short-term memory, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022, IEEE, 2022, pp. 8622–8626. doi:10.1109/ICASSP43922.2022.9747369.

[23] D. Herr, B. Obert, M. Rosenkranz, Anomaly detection with variational quantum generative adversarial networks, Quantum Science and Technology 6 (2021) 045004. URL: https://dx.doi.org/10.1088/2058-9565/ac0d4d. doi:10.1088/2058-9565/ac0d4d.

[24] Quantum continual learning of quantum data realizing knowledge backward transfer, Physica A: Statistical Mechanics and its Applications (2023) 128779. doi:https://doi.org/10.1016/j.physa.2023.128779.