ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Logarithmic Quantum Forking

Alessandro Berti *

Department of Computer Science, University of Pisa
Largo B. Pontecorvo, 3 56127 Pisa - Italy

Abstract. Quantum algorithms evolve an initial quantum state into another during computation to obtain meaningful results. However, this evolution introduces the cost of re-preparing the same initial quantum state for different tasks. Unfortunately, since quantum memory is not yet available, this cost cannot be ignored in Quantum Artificial Intelligence (QAI), where the initial quantum state typically coincides with a quantum dataset. Redundant state preparations for different tasks on the same dataset can reduce the advantages of quantum computation. To address this issue, this work proposes a new technique: the Logarithmic Quantum Forking (LQF). LQF performs state preparation for an initial quantum state once and employs additional qubits to compute an exponential number of tasks over the initial quantum state. LQF enables more efficient use of quantum computation in QAI by amortizing the cost of preparing the initial quantum state.

1 Introduction

Quantum computers promise to reduce the cost of solving problems across various domains. However, preparing an initial quantum state $|\psi\rangle$ [1, 2, 3] for a given quantum algorithm $\mathcal{A}$ can be computationally expensive. Nevertheless, running $\mathcal{A}$ on several inputs requires the initial quantum state $|\psi\rangle$ to be re-prepared for each input. Typically, a quantum algorithm evolves the prepared initial quantum state $|\psi\rangle$ during computation into another one, making it no longer usable unless it is re-prepared at some cost by uncomputation [4]. A workaround would be to *copy* the known initial quantum state multiple times after preparation. However, the No-cloning Theorem of quantum mechanics [5] prohibits the exact copy of a quantum state. Therefore, copying a quantum state as a means to overcome the expense of multiple state preparations becomes impractical. As a result, this theorem has practical implications that limit the advantages of quantum algorithms. A typical use case where having multiple copies of an initial quantum state would be beneficial is the Quantum Artificial Intelligence (QAI) realm. Indeed, a classical dataset can be large; thus, the cost of encoding classical data into the corresponding quantum state $|\psi\rangle$, representing a quantum database, can be significant. Hence, if we want to run more instances of a quantum algorithm $\mathcal{A}$ on the same quantum dataset $|\psi\rangle$ over different inputs in the same circuit, we pay the cost of preparing $|\psi\rangle$ for each input. Finding

*I thank my supervisors Anna Bernasconi, Gianna M. Del Corso, and Riccardo Guidotti (University of Pisa) for supporting me in the review of this article. This work is partially supported by INdAM-GNCS Project CUP_E53C22001930001, and by Institutional Research Grants - Pr. no. PRA_2020-2021_92, "Quantum Computing, Technologies and Applications"

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

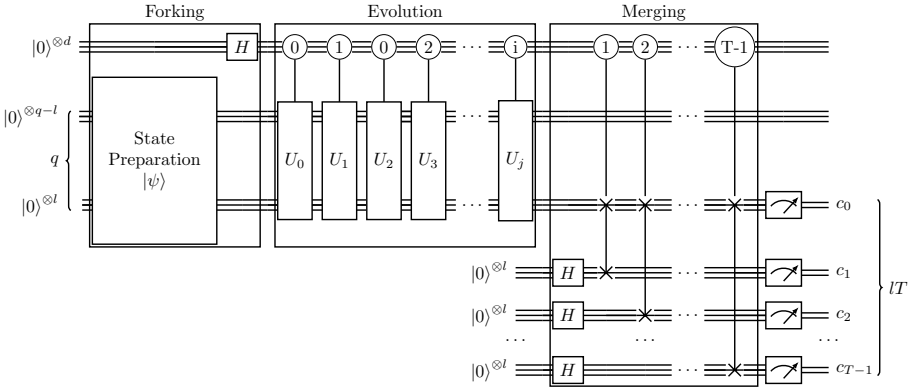

Fig. 1: Logarithmic Quantum Forking circuit framework.

innovative ways to reduce the state preparation cost remains an active area of research in quantum computing. Although a quantum state cannot be copied, it is possible to fork a quantum state. In [3], the authors introduce the idea of quantum forking (QF). They observe that forking a quantum state is similar to forking processes in operative systems, where each fork represents a process that evolves independently. In [6], the authors provide a quantum forking technique: Quantum Forking-based Sampling (QFS). Given an initial quantum state $|\psi\rangle$ of q qubits, and T forks, QFS requires $d+q(T-1)$ additional qubits. Hence, when q or T are large, the forking cost in terms of qubits is not negligible.

This work presents a new technique based on QF: the Logarithmic Quantum Forking (LQF). LQF creates a number T of forks of the initial state $|\psi\rangle$ of q qubits employing only $d = \log_2 T$ control qubits. Then, each fork evolves independently through controlled gates on the d control qubits. Eventually, we measure the qubits representing the outputs of each fork at once employing only T additional quantum registers of l qubits, where l is the number of qubits encoding the result of a given fork, and $l \ll q$. With respect to QFS, LQF reduces the number of additional qubits by a factor $T(q-l)$. Moreover, LQF keeps the success probability of each fork constant when an algorithm $\mathcal{A}$ running on different forks requires post-selections. To summarise, the LQF technique: 1) scales the number of T forks in the number d of control qubits exponentially, and 2) reduces the number of state preparation of $|\psi\rangle$ routines by $\Theta(T)$ by adding $l(T-1)$ qubits. LQF proves particularly useful in the QAI field, where an efficient use of the additional qubits required for implementing QF techniques is crucial due to the potentially large number of tasks involving the same dataset. This work also showcases an application of LQF for a quantum version of the well-known K-nearest neighbors algorithm (KNN) [7], along with a Qiskit implementation[1] of this use case for both the LQF and QFS forking techniques.

[1]Code available at: `https://github.com/Brotherhood94/logarithmic_quantum_forking.git`

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

2 Logarithmic Quantum Forking

Forking. Let $|\psi\rangle$ be an initial quantum state of q-qubits obtained from a state preparation technique, and let T be the number of tasks involving $|\psi\rangle$. For simplicity, let us assume T is a power of 2. Eventually, let $d = \log_2 T$ be the control qubits. Hence, we fork the quantum state $|\psi\rangle$ in T forks by applying a Hadamard gate on each of the d control qubits: $H|0\rangle^{\otimes d}|\psi\rangle = \frac{1}{\sqrt{T}}(|0\rangle + \cdots + |T-2\rangle + |T-1\rangle) \otimes |\psi\rangle = \frac{1}{\sqrt{T}}\sum_{i=0}^{T-1}|i\rangle \otimes |\psi\rangle$, where i denotes the integer representation of a d-bit binary number (Fig. 1).

Evolution. Each binary configuration $i \in [0, 1, \ldots, T-1]$ of the d control qubits flags a specific fork of the quantum state $|\psi\rangle$. By leveraging the control qubits, it is now possible to independently evolve each fork of $|\psi\rangle$ by means of any gate U controlled by the d control qubits. That is, if we want to evolve the i-th fork of the quantum state $|\psi\rangle$ according to a generic gate U_j, then we apply U_j controlled by $|i\rangle$. It is worth noting that we can evolve each fork in any order, and it is possible to evolve a given fork i with subsequent gates controlled by the same configuration $|i\rangle$ of the control qubits. For example, according to Fig. 1 we have the following state: $\frac{1}{\sqrt{T}}(|0\rangle\otimes U_2U_0|\psi\rangle + |1\rangle\otimes U_1|\psi\rangle + |2\rangle\otimes U_3|\psi\rangle + \cdots + |i\rangle\otimes U_j|\psi\rangle)$, which we can write more concisely defining $|\psi_i\rangle$ as the quantum state $|\psi\rangle$ evolved in fork i, then: $\frac{1}{\sqrt{T}}\sum_{i=0}^{T-1}|i\rangle|\psi_i\rangle$.

Merging. The merging step is fundamental to retrieve the results of each fork simultaneously. After the Evolution step, each fork encodes the result of the corresponding fork in superposition in a subset of l-qubits of $|\psi\rangle$. Then, we prepare $T-1$ quantum registers of l-qubits, each in equal superposition: $\frac{1}{\sqrt{T}}\sum_{i=0}^{T-1}|i\rangle|\psi_i\rangle\bigotimes_{i=0}^{T-1}|+_i\rangle^{\otimes l}$. Lastly, for each fork $i \in [0, 1, \ldots, T-1]$, we swap the l-qubits of $|\psi\rangle$ encoding the results of each fork, with the quantum registers $|+_i\rangle^{\otimes l}$ controlled on the configuration $|i\rangle$ of the control qubits. From a high-level perspective, the controlled swaps store the results of each fork on a different register of $l-$qubits. It is possible to show that this yields the following state:

$$\frac{1}{2^T}\sum_{c=0}^{2^T-1}\left(\frac{1}{\sqrt{T}}\sum_{i=0}^{T-1}\alpha_{\lfloor\frac{c}{2^{il}}\rfloor\%2^l}|i\rangle|\psi_i\rangle\right)|c\rangle,$$

where $\left\|\alpha_{\lfloor\frac{c}{2^{il}}\rfloor\%2^l}\right\|^2$ corresponds to the probability of measuring $\left|\lfloor\frac{c}{2^{il}}\rfloor\%2\right\rangle = |c_i\rangle^{\otimes l}$ in fork i. We observe that the $|c\rangle^{\otimes lT} = |c_0\rangle^{\otimes l}|c_1\rangle^{\otimes l}\ldots|c_{T-1}\rangle^{\otimes l}$ describes the results of each T-forks in a single binary string. Eventually, we apply a measurement on $|c\rangle$ in the computational basis (Fig. 1). The output probability distribution peaks on the configuration $|c_0c_1\ldots c_{T-1}\rangle$ with the most probable outcome assignment for each fork by preparing the state $|\psi\rangle$ once.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

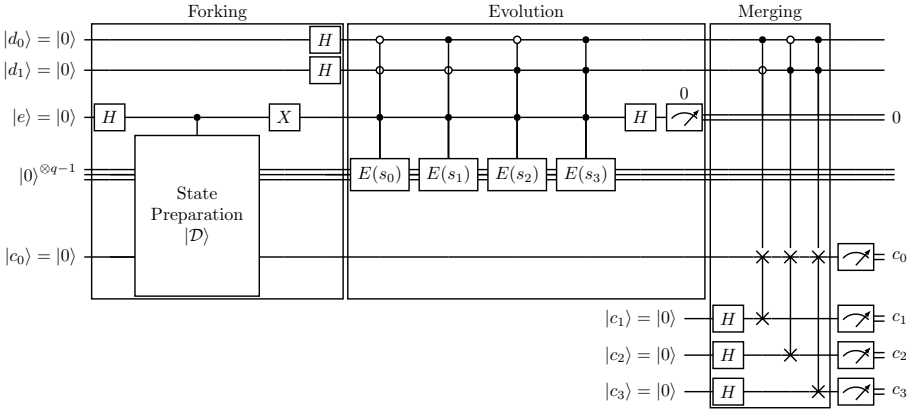

Fig. 2: Classification of samples s_0, s_1, s_2, s_3 with LQF.

3 Quantum Forking for AI

The forking framework provided by the LQF technique proves useful when (1) we want to use a given quantum state $|\psi\rangle$ multiple times in the same circuit, (2) the state preparation procedure for $|\psi\rangle$ is non-trivial, and (3) the number q of qubits of the quantum state $|\psi\rangle$ is not negligible. Follows an use case.

Setting the stage. We employ LQF over a quantum distance-based classifier to retrieve the classification results of more samples at once. Let $|\mathcal{D}\rangle$ be the initial quantum state encoding a dataset $\mathcal{D} = \langle X, Y \rangle$, where $X = \{x_1, x_2, \ldots, x_n\}$ is the set of training instances, and $Y = \{0, 1\}$ is the set of class labels. Eventually, let $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$ be the set of samples to classify. Therefore, the quantum distance-based classifier [8] assigns a class label c_i to sample s_i according to a weighting of the Euclidean distances between s_i and the instances $x_j \in X$.

Forking. In this setting, the LQF technique employs $\log_2 |\mathcal{S}| = \log_2 4 = 2$ control qubits $|d\rangle = |d_0 d_1\rangle$ (i.e., a fork for each sample). By putting the two control qubits into equal superposition, we create four forks of $|\mathcal{D}\rangle$ in an elegant and effective way. Each control qubits configuration (i.e., $|00\rangle, |01\rangle, |10\rangle, |11\rangle$) addresses a fork of $|\mathcal{D}\rangle$: $\frac{1}{\sqrt{4}} \sum_{i=0}^{3} |i\rangle \otimes |\mathcal{D}\rangle$. Since the set of labels $|Y|$ consists of two classes, the classifier requires a single qubit to encode the class labels. Eventually, let $|e\rangle$ be an ancillary qubit necessary to compute the Euclidean distances between all the instances $x_i \in X$ and each sample $s_i \in \mathcal{S}$ (Fig. 2).

Evolution. Let us amplitude encode [9] the samples in $\mathcal{S} = \{s_0, s_1, s_2, s_3\}$. Thus, let $E(s_i)$ be a gate that encodes a sample $s_i \in \mathcal{S}$. By controlling $E(s_i)$ along with $|e\rangle$ and each configuration $|d\rangle$ of the control qubits (i.e., $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$), each fork of $\mathcal{D}$ independently evolves according to a specific input $E(s_i)$. Then, the last Hadamard gate on $|e\rangle$ and the post-selection on $|e\rangle = 0$ compute the Euclidean distance on each fork of $\mathcal{D}$. (Fig. 2).

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

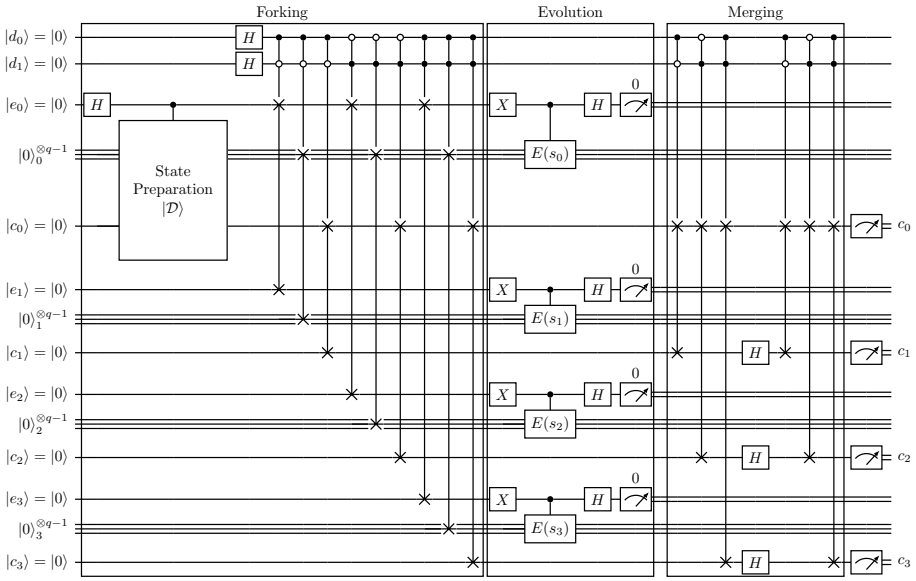

Fig. 3: Classification of samples s_0, s_1, s_2, s_3 with QFS.

Merging. In our use case, the measurements target only the qubits encoding the assigned class labels and an ancillary qubit related to the Euclidean distance. Right after the Evolution part, the qubit $|c_0\rangle$ encodes the classification results of the four samples (i.e., $\{s_0, s_1, s_2, s_3\}$) in superposition. In particular, $|c_0\rangle$ is a 1-qubit register since $l = \log_2(|Y|) = 1$. Then, we add $l(T-1) = 3$ ancillary qubits (respectively, c_1, c_2, c_3), we apply a Hadamard gate to each ancillary qubits, and we perform three controlled swap gates to store the results on different qubits (Fig 2). Eventually, the measurement of c_0, c_1, c_2, and c_3 occurs. The configuration of $c_3c_2c_1c_0$ with the highest probability denotes the output classification assignment of each sample s_0, s_1, s_2, and s_3.

4 Discussion

The Logarithmic Quantum Forking (LQF) technique turns out to be useful when the number of forks and qubits encoding an initial quantum state are not negligible. In general, given: a quantum register of q qubits encoding an initial quantum state $|\psi\rangle$, T different inputs to compute from $|\psi\rangle$, $d = \log_2 T$ control qubits, T quantum registers of l-qubits each, the total number of additional qubits for LQF is $d + l(T-1)$. On the other hand, the QFS technique [6], under the same assumptions, requires $d + q(T-1)$ additional qubits. When $l \ll q$, the LQF technique provides an advantage by a factor of $T(q-l)$ with respect to the QFS technique since it forks an initial quantum state $|\psi\rangle$ by decoupling the size q of $|\psi\rangle$ and the l-qubits needed to measure. Fig. 3 shows the circuit based on QFS

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

to address the same use case presented in Section 3. The QFS technique requires replicating the circuit of the Euclidean distance on each fork, thus, introducing $q(T-1)$ additional qubits. Furthermore, utilizing QFS involves a postselection for each $|e_i\rangle$, where $0 \leq i \leq T-1$. This aspect significantly impacts the success probability of a given run. In fact, according to [8], the success probability of the postselection on a single $|e_i\rangle$ is $\frac{1}{2}$ (i.e., $P(|e\rangle = 0) = \frac{1}{2}$), therefore the success probability of a given run decrease exponentially according to the formula $P(success)_{QFS} = \prod_{i=0}^{T-1} P(|e_i\rangle = 0) = \frac{1}{2^T}$. Whereas, LQF performs the postselection on the single $|e\rangle$, hence the success probability remains constant: $P(success)_{LQF} = \frac{1}{2}$.

5 Conclusion

In Quantum Artificial Intelligence (QAI), preparing an initial quantum state $|\psi\rangle$ is crucial. AI algorithms typically require massive amounts of data, and encoding it in a quantum state $|\psi\rangle$ can be costly. Moreover, each time a task evolves $|\psi\rangle$, $|\psi\rangle$ has to be re-prepared. Hence, given that quantum memory is not yet available, a new state preparation is required for each task. The LQF technique resolves this issue by forking $|\psi\rangle$ to avoid redundant state preparations for different tasks. LQF scales the number of forks of $|\psi\rangle$ exponentially in the number of control qubits in a simple and effective way while serializing the evolution of each fork. The LQF technique keeps the post-selection probability of a quantum algorithm constant if any post-selection is present and reduces the number of additional qubits compared to the state-of-the-art. Eventually, this work provided the Qiskit implementations of LQF and QFS. Future works will explore the applicability of LQF to other use cases.

References

[1] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.

[2] V.V. Shende, S.S. Bullock, and I.L. Markov. Synthesis of quantum-logic circuits. *IEEE TCAD*, 25(6):1000–1010, 2006.

[3] D.K. Park, F. Petruccione, and J.K.K. Rhee. Circuit-based quantum random access memory for classical data. *Scientific reports*, 9(1):1–8, 2019.

[4] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2016.

[5] W.K. Wootters and W.H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[6] D.K. Park, I. Sinayskiy, M. Fingerhuth, F. Petruccione, and J.K.K. Rhee. Parallel quantum trajectories via forking for sampling without redundancy. *New Journal of Physics*, 21(8):083024, 2019.

[7] P.N. Tan, M. S. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[8] M. Schuld, M. Fingerhuth, and F. Petruccione. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhysics Letters)*, 119(6):60002, 2017.

[9] M. Schuld and F. Petruccione. *Supervised learning with quantum computers*, volume 17. Springer, 2018.