Quantum Generative Modeling via Straightforward State Preparation

Leonardo Lavagna, Francesca De Falco, Simone Piperno,
Andrea Ceschini, Antonello Rosato, and Massimo Panella

Department of Information Engineering, Electronics and Telecommunications,
University of Rome "La Sapienza", 00184 Rome, Italy
Email: {leonardo.lavagna,francesca.defalco,andrea.ceschini,massimo.panella}@uniroma1.it

1 Background and Objectives

Generative modeling is a well-known technique that is gaining more and more attraction in recent times, specifically due to its highly generalizable results in many fields. In brief, it makes use of vast amounts of unlabeled data (observed variables) extracted from an unknown target probability distribution; the goal is to automatically generate new samples from a (parameterized) distribution that closely resembles the target distribution under a suitable norm. Among the possible successful application fields there are computer vision and speech synthesis, with noteworthy architectures such as variational autoencoders and autoregressive models [1].

Due to both theoretical and practical overlapping aspects, there is a strong connection between generative models, statistical and quantum physics. Iconic architectures like Boltzmann machines are linked to the Ising model, or revolve around probability estimation in immense spaces, as is the case with Quantum Born Machines (QBMs) [2], [3]. In particular, these interrelations are evident examining the Boltzmann distribution associated to a Boltzmann machine

$$p_B(x) := \frac{e^{-H(x)}}{\sum_x e^{-H(x)}}\,, \tag{1}$$

where $H = H(x)$ is the Hamiltonian of the system composed of the observed variables. If the observed variables are viewed as a snapshot of a particular quantum state $|\psi\rangle = |\psi(x)\rangle$ collapsed on a fixed basis, we can model via the Born's rule the target probability distribution as:

$$p_Q(x) := \frac{|\psi(x)|^2}{\sum_x |\psi(x)|^2}\,. \tag{2}$$

By using (2), we can translate a generation process into the learning problem of a quantum state, where the quantum circuit preparing p_Q has the capability to reproduce models as in (1).

In this work, we describe a new generative process comparable to traditional hybrid models in the context of variational quantum algorithms. Also, we explore possible modifications of the proposed model in order to increase its expressibility and provide enough flexibility to give an alternative to QBMs, having the ability to work jointly with classical optimization routines. Thanks to an optional constant-time parameter optimization and to a straightforward state preparation, we are able to also avoid some of the known issues of trainability and generalization [3]. Performance assessment of the proposed methodologies is carried out on benchmark datasets, showing promising results in terms of closeness of the recovered post-measurement probabilities to the target density.

2 Quantum Generative Models

Given a target density $p = p(x)$, if we can prepare it as wave function $\psi = \psi(x)$, equation (2) gives the possibility to recover p using its approximation p_Q. Standard state preparation circuits involve matrix product states (MPS) [4], where ψ is parameterized with respect to the set of observed

variables $\mathcal{X} := \{x_1, \ldots, x_n\}$ introducing suitable operators $\mathbf{a}^{(1)x_1}, \ldots, \mathbf{a}^{(n)x_n}$ and taking the trace: $\psi(x_1, \ldots, x_n) = \mathrm{Tr}(\mathbf{a}^{(1)x_1}, \ldots \mathbf{a}^{(n)x_n})$. This representation can be learned from data via the maximum likelihood principle by minimizing $\mathcal{L}(x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} \ln p(x_i)$, which is equivalent to minimizing the KL-divergence between p_Q and the empirical distribution of the elements in the dataset $\mathcal{X}$.

To recover p given $\mathcal{X}$ we may take an initial input state $|\psi_0\rangle$ and a sequence of parameterized unitary operators $U(\theta) := \prod_{j=1}^{k} e^{iH_j\theta_j}$, where k is the number of layers in the corresponding general-purpose Parametrized Quantum Circuit (PQC), $\theta = (\theta_1, \ldots, \theta_k)$ are (real, trainable) parameters, and $H_j \in i\mathfrak{u}(2^n)$ are suitable observables. The evolved state $|\psi_\theta\rangle := U(\theta)|\psi_0\rangle$ yields a parametric model as in (2) with $|\psi_\theta\rangle$ instead of $|\psi\rangle$, and $p_Q = p_Q(x;\theta)$ depends also explicitly on θ. Parameters' optimization happens via a quantum-classical training loop with the typical cost function $\mathcal{M}_\alpha(p_Q, p) := \left|\left|\sum_x p_Q(x;\theta)\alpha(x) - \sum_x p(x)\alpha(x)\right|\right|^2$, where α maps to a reproducing kernel Hilbert space (RKHS) [3].

Taking into consideration the two approaches sketched above, and inspired by [5], we can work with pure states (instead of MPS) and tailored observables (instead of general-purpose PQCs), so as to devise a generative model that is easy to train and generalizes well. In particular, given m qubits, we aim at deriving $|\psi_X\rangle = \sum_{i=0}^{2^m-1} \sqrt{q_X(i)}|i\rangle$ which is a state given as a superposition of standard basis elements, each weighted by a factor $\sqrt{q_X(i)}$ such that, upon measurement, it will hold $||q_X(x) - p(x)|| \to 0$ as $m \to \infty$ thanks to the Born's rule. This way, we will have a quantum circuit on m qubits that we can use to generate new samples taken from the distribution q_X approximating p without the need for costly parameter estimations.

A simple procedure to get the weights $\sqrt{q_X(i)}$ is "binnization": take successive subdivisions $\mathcal{Y}_l$ of the training set $\mathcal{X}$ and aggregate the probabilities in each subdivision to get $q_X(i)$, where $\mathcal{Y}_0 := \mathcal{X}$ and $\mathcal{Y}_l$ is obtained from $\mathcal{Y}_{l-1}$ with a binary "cut". For instance, if n is even, then

$$\mathcal{Y}_1 := (\{x_1, \ldots, x_{n/2}\}, \{x_{n/2+1}, \ldots, x_n\}) \tag{3}$$

and, after a binary "cut",

$$\mathcal{Y}_2 = (\{x_1, \ldots, x_{n/4}\}, \{x_{n/4+1}, \ldots, x_{n/2}\}, \{x_{n/2+1}, \ldots, x_{3n/4}\}, \{x_{3n/4+1}, \ldots, x_n\}), \tag{4}$$

where $q_X(0) = 1$, $q_X(1) = \frac{x_1+\cdots+x_{n/2}}{x_1+\cdots+x_n}$, $q_X(2) = \frac{x_{n/2+1}+\cdots+x_n}{x_1+\cdots+x_n}$, ..., $q_X(7) = \frac{\sum_{x_i \in Y_2[3]} p(x_i)}{\sum_{x_i \in Y_1[1]} p(x_i)}$, with $Y_1[0]$ denoting the first component of $\mathcal{Y}_1$, $Y_2[0]$ the first component of $\mathcal{Y}_2$, and so on. Actually, we can obtain the weights by taking the probability that a given observed variable is in the left side of the bin at subdivision l, divided by the probability that the observed variable was in that bin, and using a binary convention to order the weights.

A quantum circuit can be devised to obtain the desired superposition state $|\psi_X\rangle$ using a combination of CNOT gates and R_y rotation gates [5]: we start with an ancillary register $|\psi_0\rangle = |00\ldots0\rangle$, perform the computation $\sqrt{q_X(i)}|i\rangle|\psi_0\rangle \to \sqrt{q_X(i)}|i\rangle|\theta_i\rangle$ with $\theta_i = \arccos(2\sqrt{q_X(i)/q_X(i-1)})$, then perform a controlled rotation on the $(l+1)$-qubit $\sqrt{q_X(i)}|i\rangle|\psi_0\rangle \to \sqrt{q_X(i)}|i\rangle|\theta_i\rangle(\cos\theta_i|0\rangle + \sin\theta_i|1\rangle)$. By uncomputing the register containing θ_i we are left with the desired superposition state. As we can see, the proposed methodology is straightforward, albeit with expressive power limited by the number of qubits m, whereas typical state preparation circuits based on MPS are extremely powerful, but require elaborate investigations about the tensor network structure used to prepare the wave function and the sampling strategies based on conditional probability calculations [4].

A simple variant of the proposed methodology is obtained by letting some of the θ_i be independent parameters of the observed variables. Some care must be taken since the θ_i are usually mutually dependent due to the chain-like construction of the quantum circuit. A careful study of the entropy of the generated state, as well as an expressibility analysis, are useful to deduce the number of effective parameters. In any case, the training of the corresponding quantum circuit can be done in analogy with the training of a QBM, without some of the related difficulties. Indeed, a QBM will have $O(dm)$ parameters, where d is the number of layers and m the number of qubits, whereas the proposed methodology requires (heuristically) only $O(1)$ parameters since we observed that there are only a constant (problem-dependent) number of meaningful angles θ_i that need to be optimized. In fact, by using the binnization, only θ_1, θ_2 are effective parameters, and considering other angles may hamper the generation process, as shown in the following Figs. 1(f) and 1(h). Also, QBMs will produce only quite restricted function classes, limiting their generalization capabilities, as shown in Figs. 1(a)-1(c).

3 Results and Discussion

We tested the proposed models on a univariate Gaussian dataset with $\sigma = 2\mu = 2$, red line in Figs. 1(a)-1(f), and the 8×8 MNIST dataset. Considering $m = 3$ qubits, the QBM with $d \in \{1, 2, 10\}$ layers, and $3(m+1)d$ parameters optimized for 10 epochs (without gradient information), we cannot recover efficiently the Gaussian model, as shown in Figs. 1(a)-(c) with generated samples in blue, whereas the proposed non-parameterized model performs well in Fig. 1(d). The parametrized model with a parameter for each R_y gate (7 in total) if untrained performs poorly as in Fig. 1(e), but a 10 epoch training yields good samples in Fig. 1(f). Analogue results hold also for the MNIST dataset, where only the proposed model and its variant are shown for reasons of space in in Figs. 1(g)-1(i).

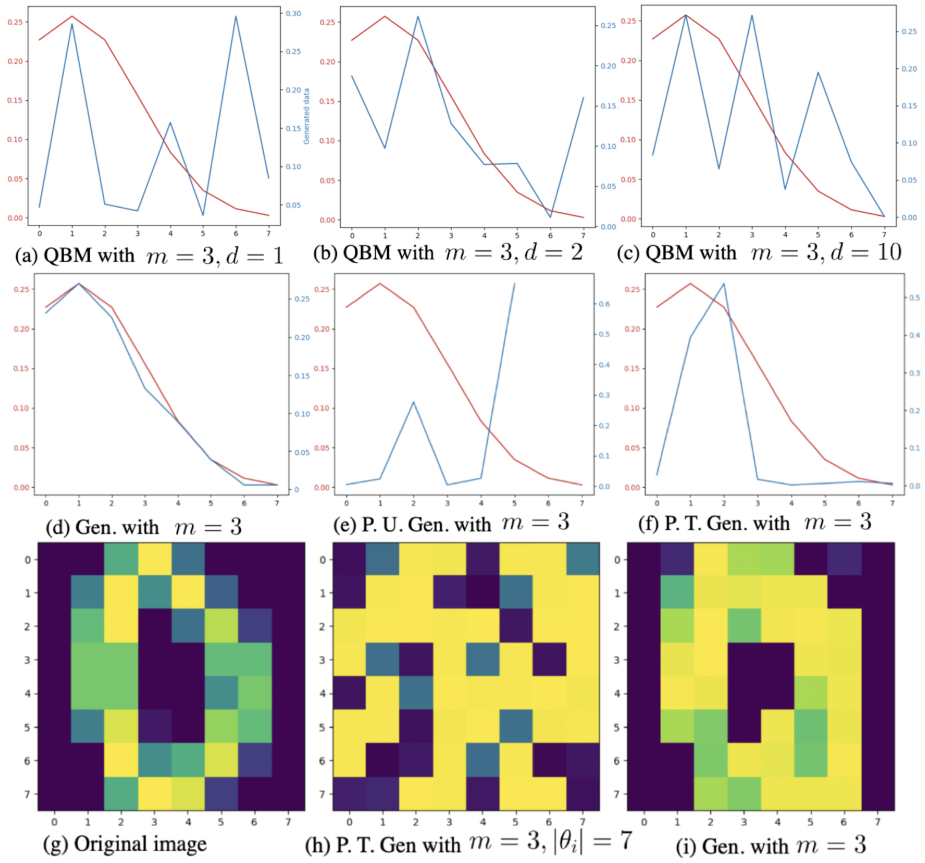

Figure 1: Generative model based on the proposed QBM methodology: without trainable parameters (Gen.); with untrained parameters (P.U. Gen.); with 7 trained parameters on 10 epochs (P.T. Gen.)

References

[1] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7327–7347, Nov. 2022.

[2] S. Cheng, J. Chen, and L. Wang, "Information perspective to probabilistic modeling: Boltzmann machines versus born machines," *Entropy*, vol. 20, no. 8, p. 583, Aug. 2018.

[3] J.-G. Liu and L. Wang, "Differentiable learning of quantum circuit born machines," *Physical Review A*, vol. 98, no. 6, Dec. 2018.

[4] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, "Unsupervised generative modeling using matrix product states," *Physical Review X*, vol. 8, no. 3, Jul. 2018.

[5] L. Grover and T. Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions*, 2002.