Synthetic Data Generation using Hybrid Quantum-Classical Generative Adversarial Networks

Fuat Gunduz

June 14, 2025

Abstract

This study proposes a hybrid model that combines quantum and classical computing paradigms for synthetic data generation. Our model is based on a Generative Adversarial Network (GAN) architecture, consisting of two main components: a Generator built using a parametrized quantum circuit (PQC) and a Discriminator implemented as a standard deep neural network. To improve training stability and mitigate common GAN issues such as mode collapse, we adopt the Wasserstein GAN with Gradient Penalty (WGAN-GP) technique. The model is implemented using the PennyLane quantum machine learning library and PyTorch deep learning framework. Experimental results show that our hybrid model successfully learns and reproduces a simple one-dimensional Gaussian distribution. This work demonstrates the potential of quantum circuits in generative modeling and the feasibility of integrating them with existing classical infrastructures.

1 Introduction

Generative Adversarial Networks (GANs), introduced by Goodfellow et al.[1], have revolutionized the field of machine learning. Based on a minimax game between a generator and a discriminator, GANs have achieved breakthrough results in image synthesis, data augmentation, and style transfer. However, training standard GANs is challenging due to issues like vanishing gradients and mode collapse.

In recent years, quantum computing has emerged as a promising field for generative models, particularly due to its natural ability to model complex probability distributions[2]. Quantum circuits, thanks to quantum mechanical properties such as superposition and entanglement, can capture correlations that are hard to model classically.

This work aims to leverage this potential by designing and implementing a hybrid Quantum GAN (QGAN) model. In our model, the data generation task is handled by a quantum circuit, while the quality evaluation of the generated data is performed by a classical neural network. To ensure training stability, we employ the WGAN-GP approach[3], evaluating how effectively our model can learn a simple target distribution.

2 Methodology

The proposed hybrid QGAN architecture consists of two main competing components: a Quantum Generator and a Classical Discriminator. The overall flow of the architecture is illustrated in Figure 1.

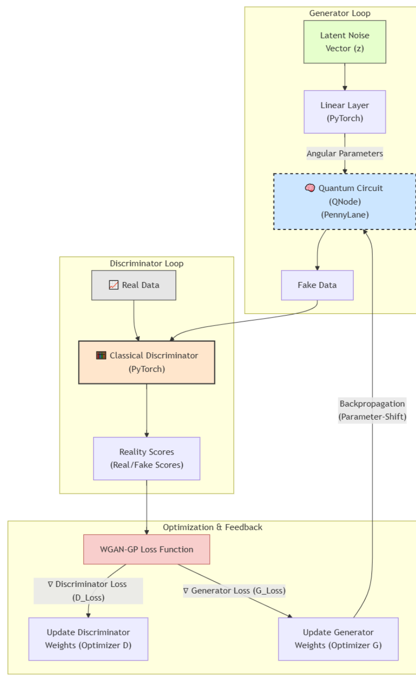

Figure 1: Hybrid Quantum-Classical GAN architecture showing generator and discriminator loops along with optimization feedback.

2.1 Quantum Generator

The generator transforms a random noise vector z into a data point. It consists of a classical preprocessing layer and a quantum circuit:

Linear Layer: A standard PyTorch `nn.Linear` layer maps the low-dimensional z vector to the number of parameters expected by the quantum circuit (in this work, equal to `n_qubits`).

Parametrized Quantum Circuit (PQC): The core of the generator, implemented as a PennyLane QNode, includes:

- **Embedding:** Classical parameters are encoded into qubit rotation angles using `AngleEmbedding`.
- **Variational Layers:** Consist of trainable quantum gates like RY, RX, and entangling CNOT gates. These layers enhance expressivity and introduce entanglement among qubits.
- **Measurement:** The expected value of the Pauli-Z operator on a chosen qubit is measured via `qml.expval(qml.PauliZ(0))`, reducing the quantum state to a classical value between -1 and +1.

2.2 Classical Discriminator

The discriminator is a feedforward neural network that assigns a "reality score" to a given data point, indicating whether it is real or fake. Following the WGAN setup, the final layer does not use any activation function (e.g., Sigmoid).

2.3 Training Paradigm: WGAN-GP

To ensure stable training, we use the Wasserstein GAN with Gradient Penalty (WGAN-GP) approach. The gradient penalty term keeps the discriminator's gradient norm close to 1, mitigating the vanishing gradient problem. The loss functions are:

$$\mathcal{L}_D = D(x_{\text{fake}}) - D(x_{\text{real}}) + \lambda \cdot GP \tag{1}$$

$$\mathcal{L}_G = -D(x_{\text{fake}}) \tag{2}$$

Gradients of the quantum circuit in the generator are computed analytically using the Parameter-Shift Rule[4] and are integrated into PyTorch's autograd mechanism.

3 Implementation and Results

Our model was configured with `n_qubits=2` and `latent_dim=2`. The target distribution was chosen as a standard Gaussian $\mathcal{N}(0, 0.2)$. Training was performed for 8000 epochs using the Adam optimizer with a learning rate of 0.0001.

As training progressed, the distribution of data generated by the Quantum Generator began to closely resemble the target distribution. Figure 2 shows the final result.

The QGAN-generated distribution successfully approximates both the mean and variance of the real distribution, indicating the model's capacity to learn basic probability distributions.

4 Conclusion and Future Work

This study presented the design, implementation, and evaluation of a hybrid QGAN model composed of a quantum generator and a classical discriminator. We showed that the WGAN-GP training strategy can ensure stable training in hybrid settings.

While this work demonstrates the potential of quantum generative models, it also highlights the limitations of simulator-based approaches. Future research directions include:

- Scaling to higher-dimensional and more complex datasets (e.g., low-resolution MNIST).
- Exploring the impact of different quantum circuit architectures (ansatze) on model expressivity.
- Implementing and evaluating the model on real NISQ quantum hardware.

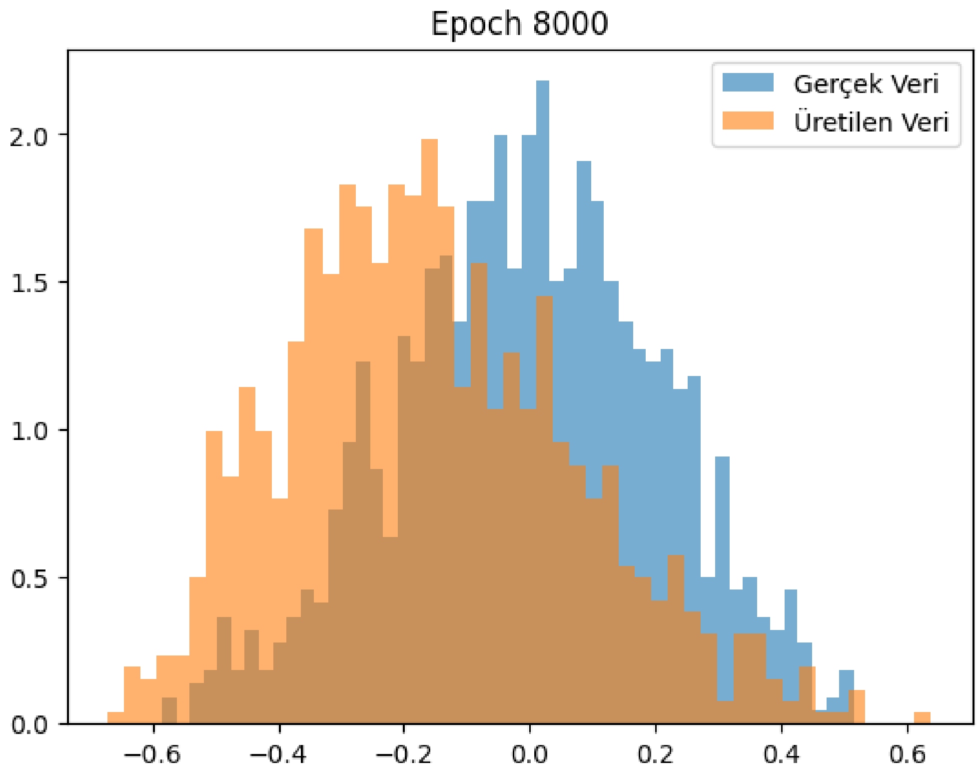

Figure 2: Histogram comparison of real data (blue) and synthetic data generated by the QGAN (orange) after training.

References

[1] Goodfellow, I., et al. (2014). Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27.

[2] Perdomo-Ortiz, A., et al. (2018). Quantum-assisted Helmholtz machines. *Quantum Science and Technology*, 3(3).

[3] Gulrajani, I., et al. (2017). Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, 30.

[4] Schuld, M., et al. (2019). Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3).