Quantum Generative Adversarial Networks

CERN openlab Technical Workshop 2021

Su Yeon Chang (CERN, EPFL)
Steven Herbert (CQC), Sofia Vallecorsa (CERN), Elias Combarro (U. Oviedo, CERN)
11/03/2021
MOTIVATION

Why Quantum Generative Adversarial Networks (GAN)?

Detectors simulation:
- Tremendous amount of time required by Monte Carlo based simulation
  → Generative Adversarial Networks

Quantum Machine Learning:
- Compressed data representation in quantum states
- Expect faster training with less number of parameters
  → Potential advantage of Quantum GAN
- Initial work using qGAN model constructed by IBM
  → limited in reproducing a probability distribution over discrete variables

→ Explore different prototypes of quantum GAN to improve the model
Quantum GAN

Practical qGAN model constructed by IBM

- Hybrid model: **Quantum** Generator + **Classical** Discriminator
- Efficient in loading and learning a probability over discrete values → $p_g(\phi)$ to approach $p_{\text{real}}$

2D image summed **over longitudinal direction**

- Normalized & Binned into $3^2 = 8$ pixels
- Averaged over 20,000 samples
Limitation

IBM qGAN model

- Limited in reproducing an average probability distribution over pixels
- Aim to reproduce a distribution over continuous variables

Need to find alternative ways to reproduce a “set” of images

- Dual-PQC GAN model (in collaboration with Cambridge Quantum Computing)
- Continuous Variable Quantum GAN
Dual-PQC GAN model

Role of single generator shared by two parameterized quantum circuits (pqc)

- **PQC1** – Reproduce the distribution over $2^{n_1}$ images of size $2^n$
- **PQC2** – Reproduce amplitudes over $2^n$ pixels on one image

$2^{n_1}$ images of size $2^n$
Application of Dual-PQC GAN in HEP

$n = 2, n_1 = 4, n_2 = 4, \text{depth}_{g_1} = 2, \text{depth}_{g_2} = 16$

- 2D image summed over longitudinal direction
- Binned into 4 pixels & normalized

Real

Generated

Quantum Generative Adversarial Networks
CV qGAN

Quantum GAN with a generator constructed by Continuous-variable NN

- Continuous-variable QC : Fundamental information-carrying units = Qumodes
- CV Neural network with CV gates (N. Killoran et al. 2019) → Construct CV qGAN

Hybrid model : Quantum Generator & Classical Discriminator

Fully Quantum model : Quantum Generator & Quantum Discriminator
Increasing latent space dimension

Latent space dimension = 3

- Classical GAN, $n_{\text{params}} \approx 45000$
- Hybrid CV qGAN, $n_{\text{params}} \approx 260$

- Faster convergence for CV qGAN
- Can achieve similar performance with 170x less parameters

Real Images

https://arxiv.org/abs/2101.11132

- 2D image summed over longitudinal direction
- Binned into 3 pixels
- No normalization required
Conclusion

Dual-PQC GAN & CV qGAN

- Two different prototypes of quantum GAN to reproduce a set of images
  1) Dual-PQC GAN, 2) CV qGAN
- Able to reproduce images with reduced size (3~4 pixels)

Future works
- Test fully quantum CV qGAN model
- Increase problem size
- Extend to other use-cases (e.g. Image generation for Earth Observation)
QUESTIONS?

su.yeon.chang@cern.ch
Appendix A : qGAN in HEP (details)

Preparation of Initial State

1. **Uniform** : Equiprobable Superposition of $|0\rangle, \ldots, |N - 1\rangle$
2. **Normal** : Normally distributed with empirical mean and std of training set
3. **Random** : Randomly distributed over $|0\rangle, \ldots, |N - 1\rangle$

Classical Discriminator

- PyTorch Discriminator
- 512 nodes + Leaky ReLU $\rightarrow$ 256 nodes + Leaky ReLU $\rightarrow$ single-node + sigmoid
- AMSGRAD optimizer for both generator and discriminator
Appendix B: qGAN in HEP (Results)

- **Uniform**
  - Simulation
  - Target

- **Random**
  - Simulation
  - Target
Appendix C : Why \( n_2 > n \)?

\[
M(j) = \begin{pmatrix}
|I_{0,j}\rangle & \frac{1}{2} e^{i\phi_{0,j}} \\
\vdots & \vdots \\
|I_{2^n-1,j}\rangle & \frac{1}{2} e^{i\phi_{2^n-1,j}}
\end{pmatrix}, \quad \phi_{ij} \in [0, 2\pi[ \quad \text{where } I_{ij} = \text{Amplitude at pixel } i \text{ for image } j \rightarrow \text{Normalized}
\]

**Case \( n_2 = n \)**
- Quantum Circuit consists of reversible gates \(\rightarrow\) **Unitary matrix**
- Inputs = computational basis \(\rightarrow\) \(M(j) = j^{th}\) column at \(M_{PQC_2}\)
  \(\rightarrow\) Cannot train PQC2 with \(n\) qubits if \(M(j)\) do not form an orthonormal basis

**Case \( n_2 = 2n \)**
- First \(2^n\) columns of PQC2 is constructed as : \(M_{PQC_2}(i) = |i\rangle \otimes |M(i)\rangle\) where \(|i\rangle \in \{|0\}, ..., |2^n - 1\rangle\},
  \(\rightarrow\) \(\langle M_{PQC_2}(i)|M_{PQC_2}(j)\rangle = \langle i|j\rangle\langle M(i)|M(j)\rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}\)
  \(\rightarrow\) \(2^{2n} - 2^n\) columns can be chosen freely to construct a unitary matrix
## Appendix D: Qubit vs. CV

<table>
<thead>
<tr>
<th></th>
<th>CV</th>
<th>Qubit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fundamental Unit</strong></td>
<td>Qumodes: ${</td>
<td>x\rangle}_{x\in \mathbb{R}}$, $</td>
</tr>
<tr>
<td><strong>Relevant Operators</strong></td>
<td>Position $\hat{x}$, Momentum $\hat{p}$</td>
<td>Pauli Operators $\sigma_x, \sigma_y, \sigma_z$</td>
</tr>
<tr>
<td></td>
<td>Mode operators $\hat{a}, \hat{a}^t$</td>
<td></td>
</tr>
<tr>
<td><strong>Common Gates</strong></td>
<td>Displacement: $D_i(\alpha) = \exp(\alpha \hat{a}_i^t - \alpha^* \hat{a}_i)$</td>
<td>Phase Shift, Rotation, Hadamard, Controlled-U gate</td>
</tr>
<tr>
<td></td>
<td>Rotation: $R_i(\phi) = \exp(i\phi \hat{n}_i)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Squeezing: $S_i(z) = \exp \left( \frac{1}{2} (z^* \hat{a}_i^2 - z \hat{a}_i^{t2}) \right)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beam Splitters: $BS_{ij}(\theta, \phi) = \exp \left( \theta (e^{i\phi} \hat{a}_i^{t} \hat{a}_j^t - e^{-i\phi} \hat{a}_i \hat{a}_j) \right)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kerr: $K_i(\kappa) = \exp(i\kappa n_i^2)$</td>
<td></td>
</tr>
<tr>
<td><strong>Measurements</strong></td>
<td>Homodyne: $</td>
<td>x_\phi\rangle \langle x_\phi</td>
</tr>
<tr>
<td></td>
<td>Heterodyne: $\frac{1}{\pi}</td>
<td>\alpha\rangle \langle \alpha</td>
</tr>
<tr>
<td></td>
<td>Photon Counting: $</td>
<td>n\rangle \langle n</td>
</tr>
</tbody>
</table>

*https://doi.org/10.22331/q-2019-03-11-129*
### Appendix E : CVNN

- Fully connected layer: \( x \rightarrow \phi(Wx + b) \quad W = \text{Weight matrix}, \; b = \text{bias}, \; \phi(x) = \text{Activation function} \\
- Weight matrix \( W \) decomposed using **singular value decomposition**:

\[
W = O_2 \Sigma O_1
\]

1. Multiplication by an orthogonal matrix \( O_1 \rightarrow \text{Apply an interferometer } U_1 \)
2. Multiplication by a diagonal matrix \( \Sigma \rightarrow \text{Apply a squeezing gate } S(r)|x\rangle = e^{-1/2 \Sigma i \xi_i}|\Sigma x\rangle \)
3. Multiplication by another orthogonal matrix \( O_2 \rightarrow \text{Apply an interferometer } U_2 \)
4. Addition of bias \( b \rightarrow \text{Apply a displacement gate } D(\alpha)|x\rangle = |x + \alpha\rangle \)
5. Non-linear function \( \phi(x) \rightarrow \text{Apply a Kerr gate } \Phi|x\rangle = |\phi(x)\rangle \)

---

https://doi.org/10.1038/ncomms13795

---

Quantum Generative Adversarial Networks

CERN OpenLab EPFL ICFO