Federated Learning Strategies of Generative Adversarial Networks for High Energy Physics Calorimeter Simulation

Mohamed A. Hemdan [1] José Cabrero-Holgueras [2,3] Sofia Vallecorsa [2]

[1]The American University in Cairo [2]CERN [3]Universidad Carlos III de Madrid

Abstract

Calorimeter simulation using Monte Carlo is a very computationally intensive process. Generative Adversarial Networks provide an alternative by capturing the complex underlying data distribution[1]. However, GANs usually require a large amount of data for training which is limited by the centralized node data capacity. We propose using federated learning for GAN training to leverage the vast data generated across large distributed computing infrastructures.

Introduction

Generative Adversarial Networks (GANs) have shown great potential in High Energy Physics (HEP) due to their ability to capture very complex data distributions. They have been used extensively in the simulations of energy showers in particle detectors [2, 1]. They were also used in other HEP applications, including cosmic ray interaction simulation and adversarial optimization [3]. However, GANs are characterized by being data-hungry, which means they rely heavily on large amounts of diverse and high-quality data to produce high-fidelity samples. Losing such a requirement creates uncertainty in the quality of the generated samples, especially if the underlying distribution is very complex, which is a typical example in HEP simulations [4]. On the other hand, increasing the training set size may not be possible due to the limited storage of the training node since data is usually generated on a cluster distributed across many different nodes.

Federated learning (FL) is a technique for training machine learning (ML) models on distributed data across many clients [5]. A centralized node sends copies of the global model to each client, which trains the model copy on its local data. These local models are then sent back to the centralized node for aggregation (e.g., averaging model parameters). Through many rounds, the global model is gradually updated to learn and perform well on the whole distributed data. FL enables models to train on larger data sets and reduces biases associated with locally trained models. Different approaches have been proposed for the federated training of GANs [6, 5, 7]. In our experiments, we consider the procedure of sharing both the generator and the discriminator on the client and the server since it proves to work well with i.i.d data [7].

To prove the effectiveness of our framework, we adapted the 2DGAN [1], a GAN-based model for calorimeter simulation, to federated learning and analyzed the changes with respect to centralized training. We experimented with different hyper-parameters (e.g., number of clients, local epochs, rounds) and studied their effect on model performance. Our results show that the FL training reaches equivalent performance to the centralized scenario.

Methodology

Federated learning is based on training an ML model on data from different clients. We used the FedAvg algorithm for model aggregation[5]. The FedAvg algorithm supposes there exist K clients, n is the total number of data points on all clients, P_j the data existing on client j, and n_j the number of samples at client j. We want to minimize the overall loss f, which decomposes into the following.

$$\min_{w \in R^d} f(w) \qquad s.t. \qquad f(w) = \sum_{j=1}^{K} \frac{n_j}{n} F_j(w), \qquad F_j(w) = \frac{1}{n_j} \sum_{i \in P_j} f_i(w)$$

Where $f(w)$ is the overall loss function for model parameters w and $f_i(w)$ is the loss of sample i and $F_j(w)$ is the total loss at client j. The 2DGAN model uses the mean squared error (MSE) between the generated and the actual energy depositions as the loss function $f(w)$. GANs where the generator and the discriminator are trained concurrently in a min-max game, trying to optimize this loss.

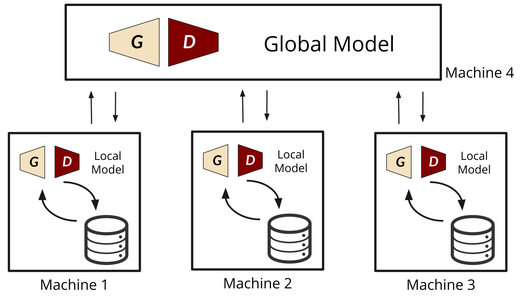

Figure 1. Federated Learning Model Training

Evaluation

To carry out the experiments, we used data from 20000 energy showers generated using Monte Carlo and then distributed across a number of different clients. We adapted the 2DGAN model to federated learning and experimented with different hyperparameters, namely the number of local epochs and the number of clients, due to their significant effect on convergence rate and training time. The experiments run on a server node with an Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz with 64 cores, 196 GB of RAM, and 4 Nvidia Tesla V100S GPUs.

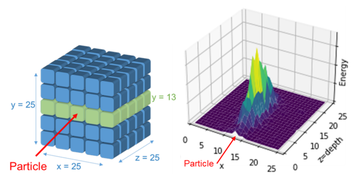

Figure 2. 3D Representation and Particle Shower development at $y = 13$

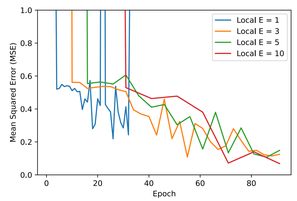

Figure 3. MSE Validation Loss per epoch for different number of local epochs

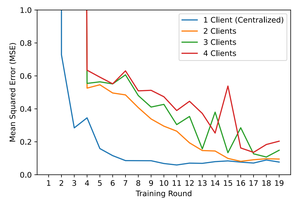

Figure 4. MSE validation loss per epoch evaluated for different number of clients

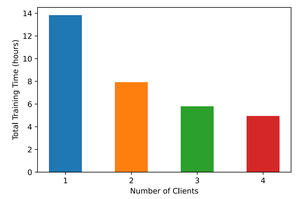

Figure 5. Total Training Time for different number of clients

Takeaways

1. Federated and Centralized learning are different. In centralized learning, only one client/machine is used for training. In contrast, federated learning uses many clients for training a global model, which is constructed by aggregating the local models from each client. These local models are updated every E number of epochs, known as a round. During each round, each local model converges to the local data, producing biased models. With many clients, each having a different model, aggregation can hardly produce a generalizable model, slowing down convergence. For example, as the number of clients increases (e.g., 2-4 clients, shown in Fig. 4), the convergence rate decreases. However, such delay is catered for by the parallelism of local epochs, which reduces the overall training time, as shown in Fig. 5.
2. Local epochs influence model convergence. Large local epochs allow for more local training and more divergence from the initial model. When aggregated, the clients' local models are not able to produce a general model, making the training inefficient. In contrast, a low number of epochs does not allow the model to grasp the local data features. For example, local E=1 causes the model to diverge as shown in Fig. 3. Thus, a tradeoff is needed to choose the best number of local epochs. Our results showed that local epochs of 3-10 can produce stable results.
3. In addition, our model guarantees privacy since no information is shared across the clients or even with the server. Local data is kept securely at each client.

Conclusion and Future Work

Federated learning can provide an effective strategy to increase the performance of current HEP machine learning applications. It enables GANs to train on much larger datasets, reducing uncertainties in the generated data. In this work, we explored the adaptation of federated learning to a GAN-based calorimeter simulation technique. We experimented with different hyperparameters and different numbers of clients. Our experiments showed more stable learning curves for high number of local epochs, and smaller training times for large number of clients. As an overall summary, our federated training reached the same performance as the centralized approach.

Future works aim to explore other federated GAN configurations (e.g., centralized generator) and different FL aggregation strategies in the context of calorimeter simulation and their effect on the learning curve.

References

[1] Florian Rehm, Sofia Vallecorsa, Kerstin Borras, and Dirk Krücker. Validation of deep convolutional generative adversarial networks for high energy physics calorimeter simulations, 2021.

[2] F Carminati, A Gheata, G Khattak, P Mendez Lorenzo, S Sharan, and S Vallecorsa. Three dimensional generative adversarial networks for fast simulation. *Journal of Physics: Conference Series*, 1085:032016, sep 2018.

[3] Gilles Louppe, Joeri Hermans, and Kyle Cranmer. Adversarial variational optimization of non-differentiable simulators. 2017.

[4] Konstantin Matchev, Alexander Roman, and Prasanth Shyamsundar. Uncertainties associated with GAN-generated datasets in high energy physics. *SciPost Physics*, 12(3), mar 2022.

[5] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. 2016.

[6] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. MD-GAN: Multi-discriminator generative adversarial networks for distributed datasets. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, may 2019.

[7] Rachid Guerraoui, Arsany Guirguis, Anne-Marie Kermarrec, and Erwan Merrer. Fegan: Scaling distributed gans. pages 193–206, 12 2020.

[8] Eckhard Elsen. A roadmap for HEP software and computing r&ampd for the 2020s. *Computing and Software for Big Science*, 3(1), November 2019.

[9] Luke de Oliveira, Michela Paganini, and Benjamin Nachman. Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science*, 1(1), sep 2017.