A Privacy Preserved Image-to-Image Translation Model in MRI: Distributed Learning of WGANs

Tolga Ergen
ergen@stanford.edu
SUNet ID: ergen

Batu Ozturkler
ozt@stanford.edu
SUNet ID: ozt

Berivan Isik
berivan.isik@stanford.edu
SUNet ID: berivan0

Abstract

In this project, we introduce a distributed training approach for Generative Adversarial Networks (GANs) on Magnetic Resonance Imaging (MRI) tasks. In our distributed framework, we have n discrimnator and a single generator. We first generate fake images via the generator, which are fed to the discriminator. In addition to the fake images, we uniformly distribute the real images into n discriminators. Each discriminator first computes a gradient using the local data revealed to itself. Then, we update a global parameter via the average of the computed gradients. With this approach, since we distribute both the data and processing to several discriminators, we reduce the computational complexity and storage demands for each discriminator. Moreover, we preserve the privacy thanks to our novel training aproach that only needs local data.

1 Motivation & Related Work

In recent years, Generative Adversarial Networks (GAN) have been widely used in Magnetic Resonance Imaging (MRI) tasks such as image-to-image translation and image reconstruction. GANs are a class of methods to learn generative models based on a game theoretical point of view [1]. Particularly, in these networks, we have a generator that generates sample from the data distribution and a discriminator that distinguishes the generated samples from the real ones. Even though GANs have shown considerable performance improvement in several computer vision and medical imaging tasks [2, 3], their training is difficult due to the mode collapse issue where the generator overfits to the data during training. To circumvent this issue, Wasserstein GANs (WGANs) are introduced [4]. WGANs mitigate the mode collapse issue of GANs, and offer improved training stability, and therefore are widely used in many machine learning tasks [4].

However, even WGANs might suffer from training problems since all GAN architectures, including WGANs, demand a large amount of high dimensional samples to be adequately trained, which causes complexity and storage issues [5, 6]. This problem is even more exacerbated in medical imaging applications, where the data dimensionality and the number of samples is extremely high. Furthermore, medical data privacy regulations inhibit utilizing patient data in a centralised manner. To mitigate such problems, we employ a distributed approach, where we distribute the training samples to different processing units with low storage and processing capabilities. After training a relatively small number of samples in each unit separately, we collect and merge the training results to obtain a final result. Therefore, we are able to process MRI images without suffering from complexity and storage issues. Furthermore, since data is held locally in different units, we will also preserve data privacy.

Our goal in this work is to achieve a distributed approach that has the following advantages:

- Privacy preservation
- Low computational complexity
- Low storage demand

2 Methods

In the project, we use a WGAN architecture for image reconstruction along with a distributed training approach. For the current phase, we separately implemented a WGAN architecture for image reconstruction and a distributed training approach for image processing, particularly, image classification.

2.1 Generative Adversarial Networks

A GAN architecture consists of two parts, i.e., generator and discriminator. Given a noisy input, the generator aims to generate samples that look like real samples. We then feed the output of the generator and real samples to the discriminator in order to distinguish fake samples from the real ones. Let us denote the distribution over the noisy inputs, the generator's distribution over the real samples, and the distribution of the real samples as p_z, p_g, and p_r, respectively. We also denote the discriminator and generator as D and G, respectively. On the one hand, D aims to improve the accuracy by maximizing $\mathbf{E}_{x\sim p_r(x)}[\log(D(x))]$, where x represents the real input samples. Meanwhile, given a noisy input $z \sim p_z(z)$, D aims to output a probability $D(G(z))$ that is close to zero by maximizing $\mathbf{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$. On the other hand, G is trained to increase the chances of D producing a high probability for a fake sample. Therefore, it aims to minimize $\mathbf{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$. Overall, both G and D try to optimize the following min-max problem

$$\min_G \max_D L(G, D)$$

where

$$L(G,D) = \mathbf{E}_{x\sim p_r(x)}[\log(D(x))] + \mathbf{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$
$$= \mathbf{E}_{x\sim p_r(x)}[\log(D(x))] + \mathbf{E}_{x\sim p_g(x)}[\log(1-D(x))].$$

2.1.1 Wasserstein Distance

Wasserstein distance is a measure of the distance between two probability distributions. In our project, we utilize Wasserstein-1 distance, also known as the earth-movers (EM) distance. The main upside of utilizing this distance metric is that it is continuous everywhere, unlike the Jensen-Shannon (JS) divergence deployed in EGANs and the distance deployed by the LSGANs. Wasserstein-1 distance, between probability mass p_r and p_g is defined as

$$W(p_r,p_g) = \inf_{J\in J(p_r,p_g)} \int \left\|x-y\right\|_1 dJ(x,y)$$

where $J(p_r,p_g)$ is the set of all joint distributions for x and y whose marginals are p_r and p_g, respectively [7].

2.1.2 Wasserstein GAN

WGAN aims to minimize the Wasserstein-1 distance. Minimizing the Wasserstein-1 distance, one can write the modified objective function as

$$\min_G \max_D \mathbf{E}_{y\sim p_r(x)}[D(y)] + \mathbf{E}_{x\sim p_g(x)}[D(x)] + \eta\mathbf{E}_{t\sim p_t}[\left\|\nabla_t D(t)\right\|_2 - 1]^2.$$

where $t = \alpha x + (1-\alpha)y$ and $0 \leq \alpha \leq 1$.

2.2 Distributed Learning

In our framework, we perform the training of WGANs utilizing a distributed learning setting. In particular, a centralized generator outputs the fake images and sends these images to the discriminators. Each discriminator compares the fake image with the local real image and sends the updates to the generator. The framework of distributed training for WGANs is given in Figure 1.

Let us assume that we have n discriminators. Each of them compute an average gradient,$g_k = \nabla F_k(w_t)$, on their local data at the current model w_t and send these model gradients to the central node where we train the generator. The average of the gradients collected from the discriminators, $\frac{1}{n}\sum_{k=1}^{n} g_k$, is used for the global model update as follows,

$$w_{t+1} \leftarrow w_t - \eta\frac{1}{n}\sum_{k=1}^{n} g_k$$

where η is the learning rate [8]. Then, the updated global model parameters are sent back to each discriminator.

3 Datasets & Experimental Setup

3.1 Experimental setup

In order to generate fake samples and discriminate them, we use neural network architectures. Particularly, for both G and D, we use a 3-layer Convolutional Neural Network (CNN) architecture with kernel size 5 and stride 2. We then train each using a first order gradient based optimization algorithm, i.e., gradient descent, where we use the tensorflow implementation of Adam optimizer.

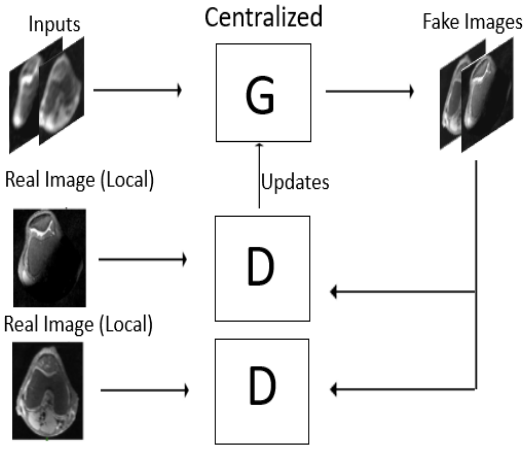


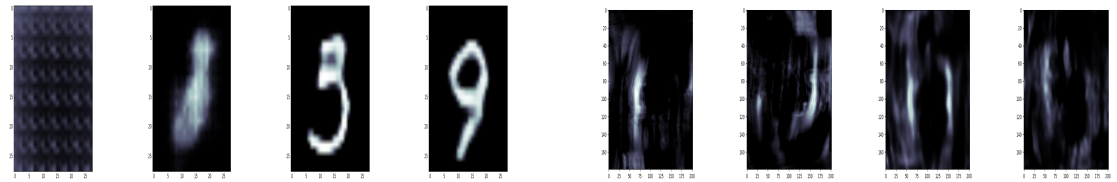
Figure 1: Distributed training of WGANs for MRI image reconstruction. A centralized generator outputs the fake images and communicates the corresponding images to each discriminator. Multiple discriminators compare the fake images with the locally stored real image and send their updates to the generator

3.2 Dataset

We used the MNIST dataset [9], and a fully-sampled multi-slice 2D cardiac cine MRI dataset.[10]. The MNIST dataset has 60000 training and 10000 test images, which are numbers from 0 to 9. Each of these images are 28x28, which are place into the dataset as 784x1 row vectors. The MRI dataset consisted of labeled examples where the labels are fully-sampled, normalized greyscale images, and inputs are undersampled, normalized, greyscale images generated by using variable density undersampling masks. The examples were obtained using a 1.5T MRI scanner where the examples had a matrix size of 180x202.

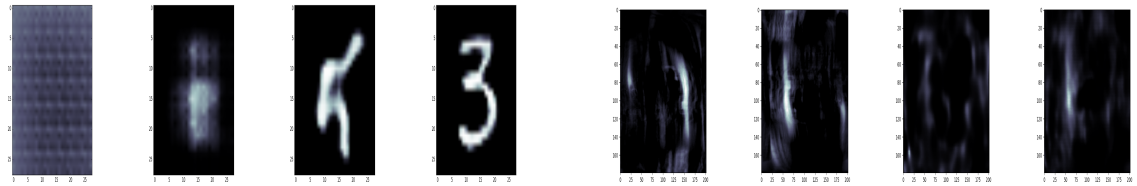
4 Results & Discussion

We evaluated the performance of our model for MNIST and an MRI dataset. Our setting worked succesfully on MNIST dataset as can be seen from the evolution of fake images in Figures 2, 3, 4. Similarly from the same figures, it is seen that we were able to generate fake images (two left-most images) very similar to the original input images (two right-most images). It can be concluded from Figure 5, 6 and Table 1 that distributed learning setting did not cause any increase in the loss or Wasserstein distance which are the basic performance measures for wGANs. Additionally, we also observed that our approach is able to provide a comparable performance with respect to the centralized approach, which has access to all the data samples unlike our approach that has access only to the local samples. Furthermore, our distributed



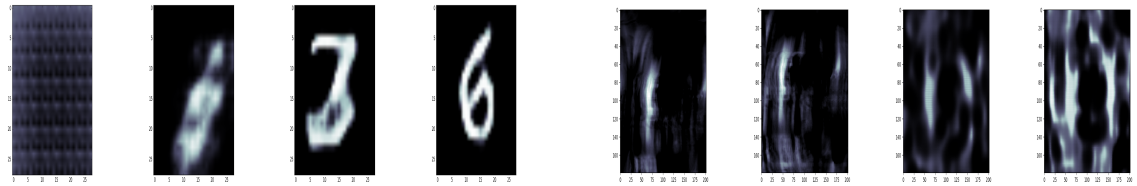
(a) Centralized-MNIST (b) Centralized-cardiac cine.

Figure 2: Centralized approach.



(a) Distributed 2 users-MNIST (b) Distributed 2 users-cardiac cine.

Figure 3: Distributed approach with 2 users.



(a) Distributed 5 users-MNIST (b) Distributed 5 users-cardiac cine.

Figure 4: Distributed approach with 5 users.

Table 1: Comparison on MNIST and cardiac cine Dataset.

	Loss-MNIST	**Wasserstein Distance-MNIST**	**Loss-MRI**	**Wasserstein Distance-MRI**
Centralized	-8.335952	-11.32122	-42.23892	-45.473563
Distributed-2	-8.401217	-13.544862	-45.66187	-35.75627
Distributed-5	-8.8937645	-13.643525	-50.28717	-28.68978

approach can even surpass the performance of the centralized approach as illustrated in Figure 5. Even though this seems to be an unexpected outcome, a distributed approach can achieve a better performance than a centralized approach. The reason for this situation is that our tasks, e.g., MNIST, have so many linearly dependent samples, which might significantly slow down the learning procedure for a centralized framework. However, when we distribute the data to several nodes, then, in a sense, we uniformly undersample the dataset and get rid off linearly dependent samples. Thus, each node trains its discriminator on a local dataset that does not have so many linearly dependent samples. As a consequence of this fact, our distributed approach surpasses the centralized approach in Figure 5.

5 Conclusion & Future Work

In this project, we succesfully implemented and trained our distributed WGAN model. We also verified the performance of our model for MNIST and an MRI dataset. Our main contributions in this work can be summarized as follows:

- We preserved each user's privacy since each node stores its local data, which is especially crucial for datasets involving medical information.
- We reduced the complexity and storage demands by distributing both the processing and the data to several nodes, which allows training large scale networks with high dimensional data.

The following points can be considered as the future work for our project:

- Showing the performance of our model for large-scale tasks, i.e. using larger networks such as ResNet, and evaluating the performance for larger datasets such as ImageNet.
- Compression methods for data transfer between the centralized node and the users can be considered, since communicating updates at each iteration is cumbersome for large neural networks.

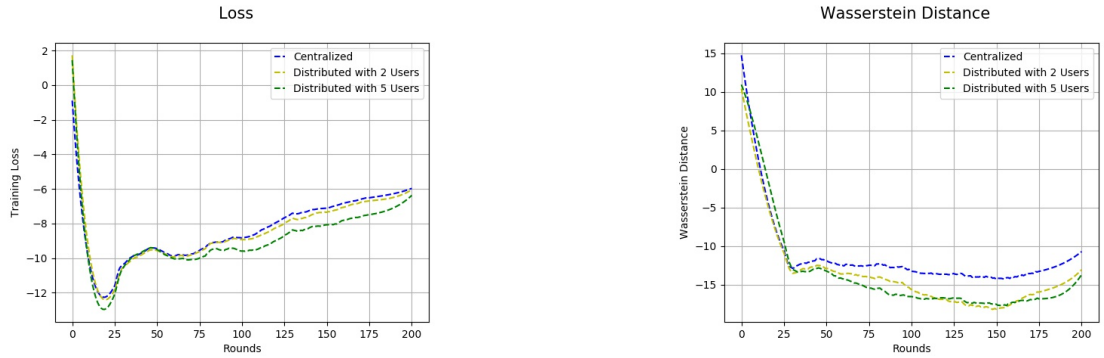


Figure 5: Generator loss and Wasserstein distance curves for training phase on MNIST.

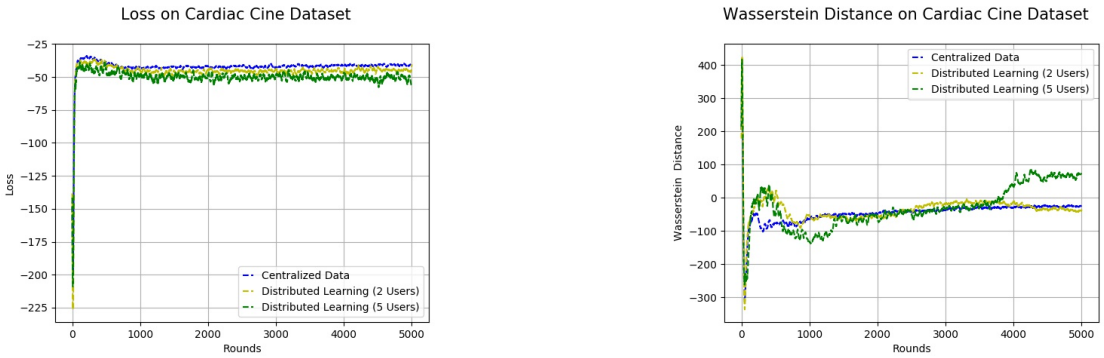


Figure 6: Generator loss and Wasserstein distance curves for training phase on cardiac cine dataset.

- Adoption of different merging techniques to further improve the performance of distributed approach, e.g., using a weighted and adaptive average of the gradients.

6 Contributions of Team Members

Each team member completed the following tasks:

- **Tolga** worked on implementing the WGAN architecture and produced results for imagea reconstruction with WGANs using the MNIST dataset.
- **Batu** worked on implementing the WGAN architecture and produced results for image reconstruction with WGANs using the cardiac dataset.
- **Berivan** worked on implementing a generic distributed architecture for WGAN training.

7 Code

Our code is available at `https://github.com/BerivanIsik/Distributed-Learning-of-wGANs.git`. In our code, we use the tensorflow based WGAN implementation in `https://github.com/adler-j/minimal_wgan`.

References

[1] Ian Goodfellow et al. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical image analysis*, page 101552, 2019.

[3] Yuhua Chen et al. Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. 2018.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. 2017.

[5] Rajagopal. A and Nirmala. V. Federated ai lets a team imagine together: Federated learning of gans. 2019.

[6] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets, 2018.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[8] H. Brendan McMahan et al. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[9] Yann LeCun. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[10] Christopher M. Sandino, Peng Lai, Shreyas S. Vasanawala, and Joseph Y. Cheng. Accelerating cardiac cine mri beyond compressed sensing using dl-espirit, 2019.