On WGANs Optimization: Competitive Gradient Descent

D'Angeli Gabriele, Di Gennaro Federico, Tian Yuzhu
Departement of Computer Science, EPFL, Switzerland

***Abstract*—In this work we approached the optimization problem of a zero-sum game, more specifically a two-player game, with a competitive optimization algorithm, referenced to as *Competitive Gradient Descent*. Our primary objective is to show that by applying this algorithm to the optimization problem of the Wasserstein Generative Adversarial Network (WGAN) on MNIST dataset, which embodies a zero-sum game involving two players, namely the Generator and the Discriminator, we can outperform the quality of generated images.**

I. Introduction

Let us examine a zero-sum game involving two players, where

$$f, g : R^m \times R^n \to R$$

are the cost functions of each player. Notably, this type of game entails an objective for each player to minimize their individual cost function, which generally encompasses dependencies on the actions taken by the opposing player:

$$\min_{x \in R^m} f(x, y), \ \min_{y \in R^n} g(x, y). \tag{1}$$

Typically, in the context of competitive optimization, the focus lies in identifying the *Nash equilibrium* of such a problem, which proves to be difficult to compute when the functions f, g are not smooth enough. The problem represented by (1) can be reformulated when considering a zero-sum game scenario, where the objective functions exhibit a symmetrical relationship as $f = -g$.

$$\min_{x \in R^m} \max_{y \in R^n} f(x, y) \tag{2}$$

In the case of differentiable objective function, the most naive approach to solve this problem is by employing the Gradient Descent Ascent Algorithm (**GDA**). This means that each player independently computes their respective updates without considering the updates made by the other player. Unfortunately, this procedure often gives rise to oscillations or divergence even in simple examples, such as the bilinear model $f(x, y) = xy$.

The main drawback associated to conventional approaches (such as the one proposed by Goodfellow et al (2014) in [3]) for resolving the problem stated in (2), is the fact that, in order to avoid divergence, the step size has to be chosen inversely proportional to the magnitude of the interaction of the two players. Consequently, this necessitates the adoption of small step sizes, resulting in slow convergence. Moreover, determining these step sizes in advance proves to be arduous, as they often necessitate a tedious trial-and-error methodology.

II. Competitive Gradient Descent

In [1] the *Competitive Gradient Descent* algorithm (**Algorithm 1**) was introduced as a means to address the limitations encountered in the usual algorithms when facing the problem outlined in equation (2). The paper provides several notable results, including the (local) convergence of CDG in context of (locally) convex-concave zero-sum games. Aditionally, it demonstrates that stronger interactions between the two players lead to improved convergence properties, eliminating the need for step size adaptation that is typically required in other approaches. Regarding the gradient descent update, by employing the linear approximatio of each agent's loss function, the game can be reformulated as follows:

$$\min_{x \in R^m} f + D_x f(x - x_k) + D_y f(y - y_k) + \frac{1}{2\eta}||x - x_k||^2 \tag{3}$$

$$\min_{y \in R^n} g + D_x g(x - x_k) + D_y g(y - y_k) + \frac{1}{2\eta}||y - y_k||^2 \tag{4}$$

Therefore, the optimal strategy for each player is independent of the other player, which is exactly the update rule of SimGD. However, authors in [1] argue the poor convergence properties of SimGD are due to the fact they ignore the competitive nature of the game. In light of this, the authors extend the linear approximation to a multi-linear approximation, thereby formulating a game that exhibits the following structure:

$$\min_{x \in R^m} f + D_x f(x - x_k) + (x - x_k)^T D^2_{xy} f(y - y_k) + D_y f(y - y_k) + \frac{1}{2\eta}||x - x_k||^2 \tag{5}$$

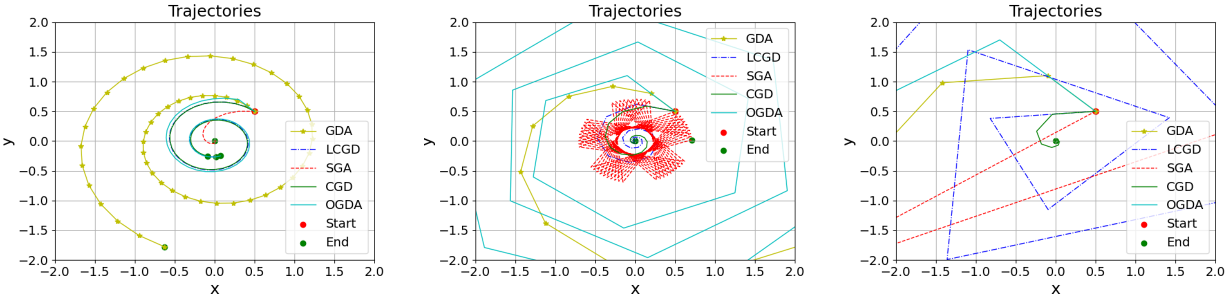
$$\min_{y \in R^n} g + D_x g(x - x_k) + (x - x_k)^T D^2_{xy} g(y - y_k) + D_y g(y - y_k) + \frac{1}{2\eta}||y - y_k||^2 \tag{6}$$

And the resulting algorithm:

Algorithm 1 Competitive Gradient Descent

```
for $0 \le k \le N - 1$ do
    $x_{k+1} = x_k - \eta(I - \eta^2 D^2_{xy} f D^2_{yx} g)^{-1}(\nabla_x f - \eta D^2_{xy} f \nabla_y g)$
    $y_{k+1} = y_k - \eta(I - \eta^2 D^2_{yx} g D^2_{xy} f)^{-1}(\nabla_y g - \eta D^2_{yx} g \nabla_x f)$
end for
return $(x_N, y_N)$
```

As an initial step towards assessing the efficacy of **Algorithm 1**, we applied it to address the optimization problem defined in Equation (2) using three distinct choices of $f(x, y)$. Subsequently, we conducted a comparative analysis with other traditinonal algorithms. The algorithms selected for comparison with CGD comprised Gradient



(a) α = 1

(b) α = 3

(c) α = 6

Fig. 1: First 50 iterations of GDA, LCGD, ConOpt, OGDA, CGD with parameters $\eta = 0.2$, $\gamma = 1$ and $(x_0, y_0) = (0.5, 0.5)$ on bilinear problem $f(x, y) = \alpha xy$.

Descent Ascent (GDA), Linearized Competitive Gradient Descent (LCGD), Symplectic Gradient Adjustmen (SGA), Optimistic Gradient Descent Ascent (OGDA), Competitive Gradient Descent (CGD).

The main difference between Competitive Gradient Descent and the algorithms mentioned before lies in the inclusion of the "**equilibrium term**" [1], which arises when seeking Nash equilibria. **This term allows each player to prioritize strategies that are less susceptible to the actions of the opposing player.**

PROBLEM 1: We first considered the bilinear problem with parameter $\alpha \in R$, described by the optimization task:

$$\min_{x\in R} \max_{y\in R} f(x, y) = \alpha xy \tag{7}$$

It is well-known that GDA algorithm fails to converge when applied to this bilinear problem, irrespective of the chosen learning rate η. To investigate this further, we selected three values for α, namely $\alpha \in 1, 3, 6$, and solved Equation (7) using the aforementioned algorithms. We observed the trajectory of each optimization method for this problem.

From Fig 1, we can notice that for $\alpha = 1$ all methods but GDA converged towards the Nash Equilibrium in $(0, 0)$, with ConOpt and SGA converging faster due to stronger gradient correction ($\gamma > \eta$). However, when we set $\alpha = 3$, OGDA, ConOpt, and SGA failed to converge. Lastly, for $\alpha = 6$, all methods, with the exception of CGD, experienced divergence. This specific example demonstrates a scenario in which CGD outperforms traditional algorithms.

PROBLEM 2: We then investigated the convex-concave problem by considering the following optimization task:

$$\min_{x\in R} \max_{y\in R} f(x, y) = \alpha(x^2 - y^2) \tag{8}$$

with $\alpha \in R$. Figure 4a in the Appendix provides an illustration of the results. It is evident that for $\alpha = 1$, all algorithms converge at an exponential rate (as depicted in the linear plot with a logarithmic scale on the y-axis), with ConOpt demonstrating the fastest convergence. However, as we increase the value of α, the algorithms begin to exhibit divergence. For $\alpha = 3$ (Figure 4b in the Appendix), both OGDA and ConOpt diverge, while the remaining algorithms continue to converge at an exponential rate. When $\alpha = 6$ (Figure 4c in the Appendix), all algorithms exhibit divergence. It is important to note that in this problem, convergence is desirable.

PROBLEM 3: Lastly, we examined the concave-convex problem defined as follows:

$$\min_{x\in R} \max_{y\in R} f(x, y) = \alpha(-x^2 + y^2) \tag{9}$$

with $\alpha \in R$. It is essential to note that in this problem, divergence is the desired outcome. The critical point $(0, 0)$ does not correspond to a Nash equilibrium, since both players are playing their worst possible strategy. However, for $\alpha = 1$, ConOpt converges to this point, providing an example in which the *consensus regularization* through the terms $\pm D^2_{xx}\nabla_x f$, $\mp D^2_{yy}\nabla_y f$ leads to a spurious solution.

As α increases, Figure 5 illustrates that every algorithm diverges as desired. For $\alpha \in$ 3, 6, the radius of attraction of $(0, 0)$ under ConOpt decreases, causing this algorithm to diverge accordingly.

III. CGD on GANs optimization

The problem that Competitive Gradient Descent (CGD) optimizes is fundamentally the same problem encountered in training Generative Adversarial Networks (GANs). In this context, the Generator (G) aims to produce realistic data, while the Discriminator (D) endeavors to differentiate between generated and real data.

For the purpose of our project, we selected the Wasserstein GAN (WGAN) framework introduced by Martin Arjovsky et al. in [4]. This choice was motivated by empirical evidence demonstrating that enforcing a 1-Lipschitz discriminator helps the convergence of GANs trained with Jensen-Shannon-based losses. The Wasserstein GAN game can be characterized as a two-player zero-sum game and is described as follows:

[1] Equilibrium term: $(I - \eta^2 D^2_{xy} f D^2_{yx} g)^{-1}$, $(I - \eta^2 D^2_{yx} g D^2_{xy} f)^{-1}$

$$\min_{G} \max_{D \in \mathcal{D}} E_{x \sim p_d}[D(x)] - E_{x \sim p_g}[D(x)] \tag{10}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions.
The original Wasserstein GAN (WGAN) framework incorporates weight clipping as a means to enforce Lipschitz continuity. However, Gulrajani et al. highlighted in [5] that this approach canlead to suboptimal performance. Consequently, in our comparative analysis, we expanded our investigation to include the Wasserstein GAN with gradient penalty (WGAN-GP) proposed by Kodaly et al. in [6]. It is worth noting that both alternatives typically employ the Adam optimizer in their optimization process.

Previous research, including [7] and [8], has explored the notion that neural networks emerge from the dynamics of gradient-based training, which tends to converge towards classifiers with strong generalization capabilities (implicit regularization phenomena). However, in the context of Generative Adversarial Networks (GANs), it has been observed by Guo et al. in [9] that even considering implicit regularization, a fully trained discriminator fails to provide informative gradients for training the generator effectively. In light of these considerations, our objective is to demonstrate, using the MNIST dataset, that the introduction of an implicit competitive regularization can indeed enhance the quality of generated images. We evaluated the image quality based on metrics suchs as *FID* and *Inception* scores [10][11]. Our experimentation maintained a fixed number of epochs and did not involve any tuning of the underlying hyperparameters.

We opted to undertake a comparative analysis of various training procedures for the Wasserstein Generative Adversarial Network (WGAN) on the MNIST dataset, namely **Baseline WGAN**, **WGAN** with **Gradient Penalty** and **WGAN** using **Adaptive Competitive Gradient Descent**. It is important to note that ACGD corresponds to **Algorithm 1** augmented with a straightforward *RMSprop*-type heuristic for learning rate adjustment [12] . For the whole experiment, we kept fixed the architectures of Generator (G) and of Discriminator (D). The only things that change are the loss definition for WGAN-GP (extra term $\lambda E_{\hat{x} \sim p_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]$, where $p_{\hat{x}}$ is the distribution obtained as interpolation of real and fake distribution) and the optimizers as the combinations described above already explain.

Following 100 epochs of training on the MNIST dataset, the WGAN employing ACGD demonstrated superior performance, a finding that had already been experimentally established on the CIFAR10 dataset in the work by [2] (see in Appendix our replication of the experiment on CIFAR10).

Figure 2 illustrates the FID scores of the generated images under the different WGAN settings on MNIST. The graph indicates that among the three training procedures, the WGAN with ACGD optimizer achieved the lowest FID score by the end of the training period. Additionally, it is worth noting that the FID score rapidly decreases after a few iterations for this approach, whereas the WGAN (with Adam) and WGAN (with Adam and GP) exhibit a much slower rate of decline. On the other hand, Figure 3 presents the Inception Scores for the various training procedures. Unlike the findings in [2] for the CIFAR10 dataset, the Inception Score did not indicate the optimal performance among the procedures considered. However, it is evident that the Inception Scores at the end of training are very similar, and the WGAN with ACGD optimizer consistently demonstrates the earliest convergence to plateau.

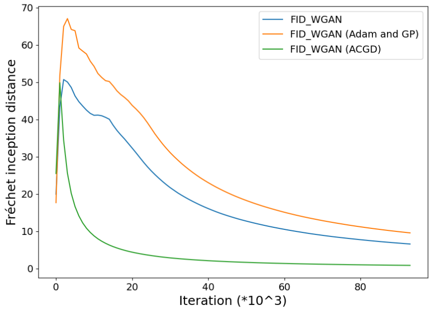


Fig. 2: *FID score of generated images in different WGAN settings on MNIST.*

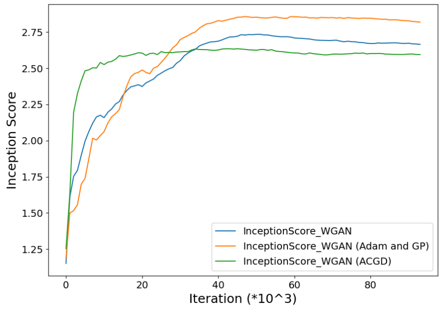


Fig. 3: *IS score of generated images in different WGAN settings on MNIST.*

IV. Conclusions

This paper aimed to investigate the effectiveness of the Competitive Gradient Descent algorithm as an optimization algorithm within a zero-sum game setting. In Section II, we extensively examined the properties of CGD and its performance in two-player games. Our findings revealed that CGD outperforms other standard algorithms, mitigating oscillatory and divergent behaviors that are commonly observed in other algorithms. Moreover, in Section III, we explored the application of CGD as an optimizer within the context of WGAN training on MNIST. We found that CGD introduces an implicit competitive regularization (ICR) effect, which significantly enhances the stability of training and yields superior results compared to conventional WGANs within a relatively short training time, eliminating the need for hyperparameter tuning.

References

[1] Competitive Gradient Descent, *Florian Schäfer, Anima Anandkumar*. NeurIPS, 2019.

[2] Implicit competitive regularization in GANs. *Florian Schaefer, Hongkai Zheng, Animashree Anandkumar Proceedings of the 37th International Conference on Machine Learning, PMLR 119:8533-8544, 2020*.

[3] Generative Adversarial Networks. *Ian J. Goodfellow et al.*, NIPS, 2014.

[4] Wasserstein GAN. *Martin Arjovsky, Soumith Chintala, Léon Bottou*, 2017.

[5] Improved Training of Wasserstein GANs. *Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville*, 2017.

[6] On Convergence and Stability of GANs. *Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron CourvilleNaveen Kodali, Jacob Abernethy, James Hays, Zsolt Kira*, 2017.

[7] Exploring Generalization in Deep Learning. *Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, Nathan Srebro*, 2017.

[8] Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. *Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, Ruosong Wang*, 2019.

[9] On Calibration of Modern Neural Networks. *Chuan Guo, Geoff Pleiss, Yu Sun, Kilian Q. Weinberger*, 2017.

[10] How to Evaluate GANs using Frechet Inception Distance (FID) *Ayush Thakur*, 2022.

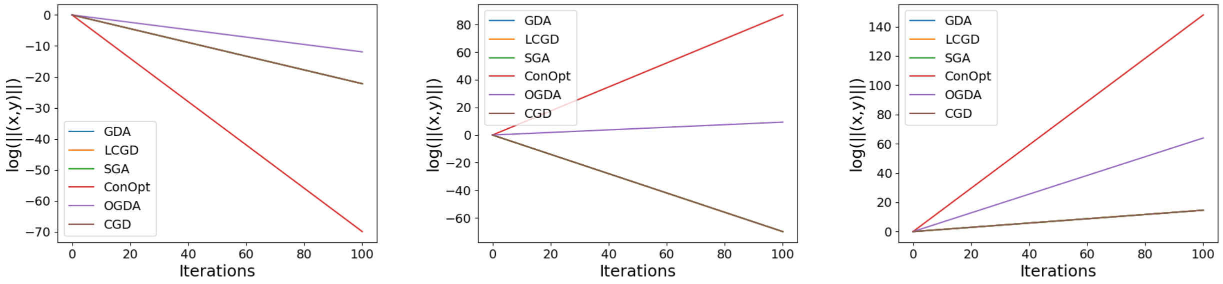
[11] A simple explanation of the Inception Score. *David Mack*, 2019.

[12] Learning Parameters, Part 5: AdaGrad, RMSProp, and Adam *Akshay L Chandra*, 2019

APPENDIX

We presents here a collection of plots utilized to evaluate the validity of our analysis. Included among them are the plots for **PROBLEM 2** and **PROBLEM 3**, as well as plots illustrating the progressions of losses for all the procedures examined on both the CIFAR10 and MNIST datasets.

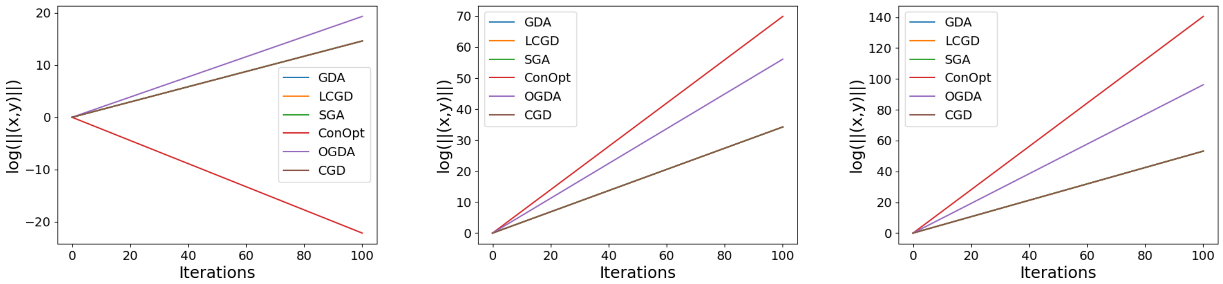
PROBLEM 2:



(a) α = 1 (b) α = 3 (c) α = 6

Fig. 4: First 100 iterations of GDA, LCGD, ConOpt, OGDA, CGD with parameters $\eta = 0.2$ and $\gamma = 1$ on PROBLEM 2 proposed in Section II with starting point $(0.5, 0.5)$. On x-axis we have the iteration and in the y-axis we have $\log(||(x,y)||_1)$

PROBLEM 3:



(a) α = 1 (b) α = 3 (c) α = 6

Fig. 5: First 100 iterations of GDA, LCGD, ConOpt, OGDA, CGD with parameters $\eta = 0.2$ and $\gamma = 1$ on PROBLEM 3 proposed in Section II with starting point $(0.5, 0.5)$. On x-axis we have the iteration and in the y-axis we have $\log(||(x,y)||_1)$

LOSS FUNCTIONS:

1) Loss function of Generator and Discriminator for the training of Baseline WGAN on CIFAR10.

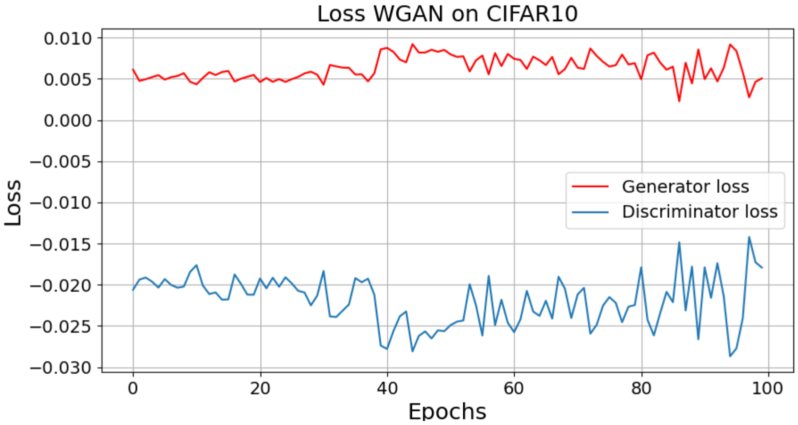


Fig. 6: *Generator and Discriminator losses in the training of Baseline WGAN on CIFAR10 dataset.*

2) Loss function of Generator and Discriminator for the training of WGAN with Gradient Penalty on CIFAR10.

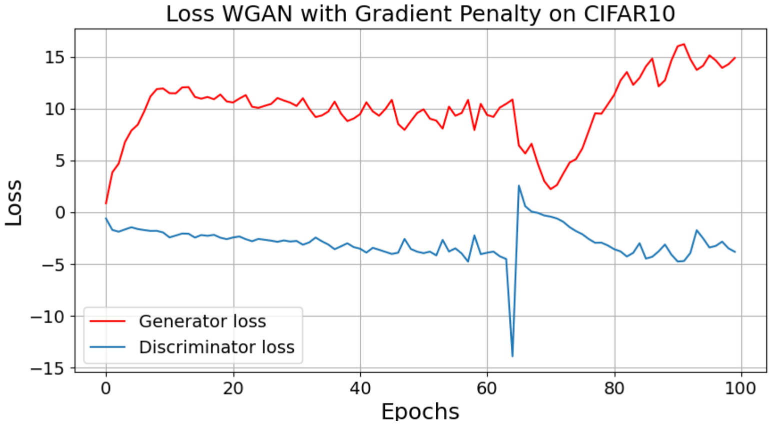


Fig. 7: *Generator and Discriminator losses in the training of WGAN with Gradient Penalty on CIFAR10 dataset.*

3) Loss function of Discriminator for the training of WGAN with Adaptive Competitive Gradient Descent on CIFAR10.

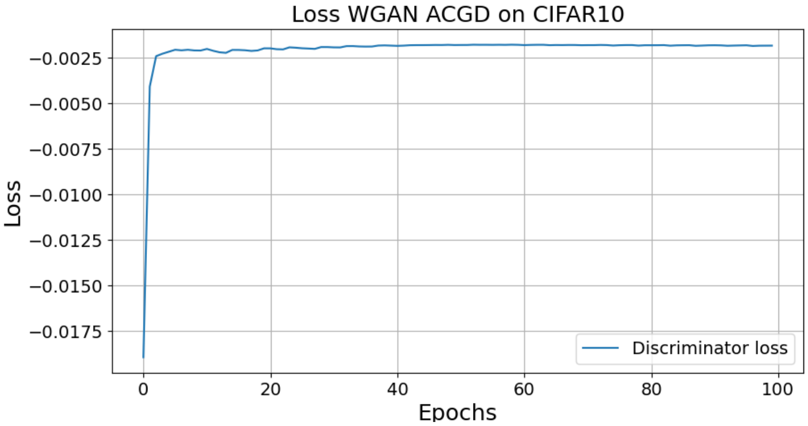


Fig. 8: *Discriminator loss in the training of WGAN with Adaptive Competitive Gradient Descent on CIFAR10 dataset.*

4) Loss function of Generator and Discriminator for the training of Baseline WGAN on MNIST.

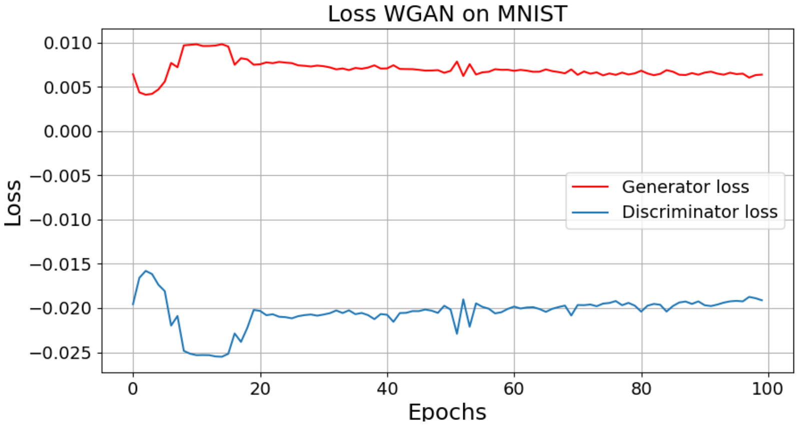


Fig. 9: *Generator and Discriminator losses in the training of Baseline WGAN on MNIST dataset.*

5) Loss function of Discriminator for the training of WGAN with Gradient Penalty on MNIST.

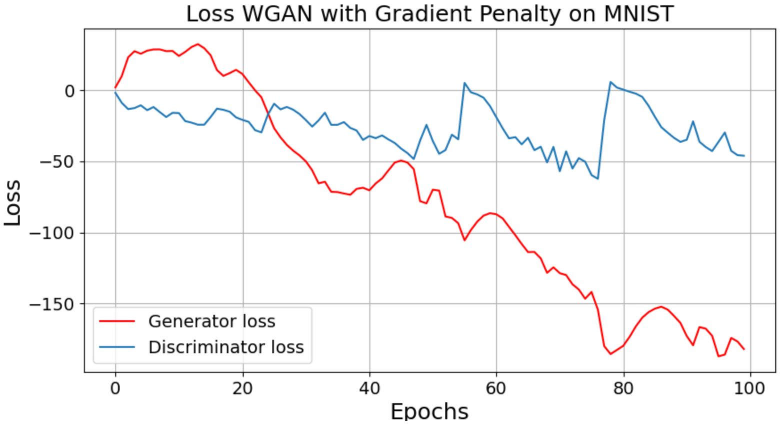


Fig. 10: *Generator and Discriminator losses in the training of WGAN with Gradient Penalty on MNIST dataset.*

6) Loss function of Discriminator for the training of WGAN with Adaptive Competitive Gradient Descent on MNIST.

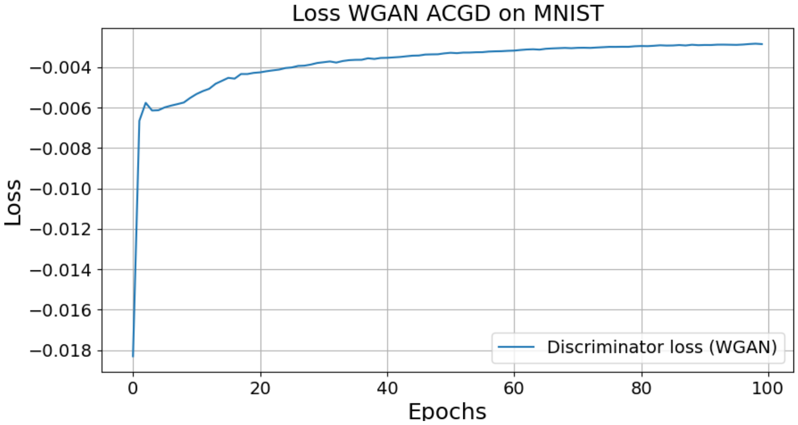


Fig. 11: *Discriminator loss in the training of WGAN with Adaptive Competitive Gradient Descent on MNIST dataset.*