Multi-objective training of Generative Adversarial Networks with multiple discriminators

Isabela Albuquerque[1,*], João Monteiro[1,*], Thang Doan[2], Breandan Considine[3], Tiago H. Falk[1], Ioannis Mitliagkas[3]
[1]INRS-EMT, Université du Québec
[2]Desautels Faculty of Management, McGill University
[3]Mila, Université de Montréal

Abstract

Recent literature has demonstrated promising results on the training of Generative Adversarial Networks by employing a set of discriminators, as opposed to the traditional game involving one generator against a single adversary. Those methods perform single-objective optimization on some simple consolidation of the losses, e.g. an average. In this work, we revisit the multiple-discriminator approach by framing the simultaneous minimization of losses provided by different models as a multi-objective optimization problem. Specifically, we evaluate the performance of multiple gradient descent and the hypervolume maximization algorithm on a number of different datasets. Our results indicate that hypervolume maximization presents a better compromise between sample quality and diversity, and computational cost than previous methods.

1 Introduction

Generative Adversarial Networks (GANs) [1] offer a new approach to generative modeling, using game-theoretic training schemes to implicitly approximate a probability density represented by training data. Prior to the emergence of GAN architectures, realistic generative modeling remained elusive. When offering unparalleled realism, GAN training remains fraught with stability issues. Commonly reported shortcomings involved in the GAN game are the lack of useful gradients provided by the discriminator, and mode collapse, i.e. lack of diversity in the generator's samples. Considerable research effort has been devoted in recent literature in order to overcome training instability, i.e. divergence and mode-collapse of the generator when the discriminator is able to easily distinguish real and fake samples during training [2, 3]. Neyshabur et al. [4], for instance, proposed a GAN variation such that one generator is trained against a set of discriminators, where each discriminator sees a fixed random projection of the inputs. Prior work, including GMAN [5] has also explored the use of multiple discriminators in GANs training.

In this paper, we build upon Neyshabur et al.'s introduced framework [4] and propose treating the *1 generator vs. many discriminators* setting as a multi-objective game. Specifically, we propose treating the loss signal provided by each discriminator as an independent objective function, and simultaneously minimize such losses. We exploit previously introduced methods in literature such as the multiple gradient descent algorithm (MGD) [6]. However, due to MGD's prohibitively high cost in the case of large neural networks, we propose the use of more efficient alternatives such as hypervolume maximization. In contrast to Neyshabur et al.'s approach, where the average loss is minimized when training the generator, hypervolume maximization (HV) is shown to optimize a weighted loss, and the generator's training will adaptively assign greater importance to feedback from discriminators against which it performs poorly.

[*]Equal contribution. Correspondence to {isabelamcalbuquerque,joaomonteirof}@gmail.com

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NIPS 2018), Montréal, Canada.

2 Preliminaries

Multi-objective optimization. We define a multi-objective optimization problem [7] as finding $\mathbf{x}$ such that: $\min \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_K(\mathbf{x})]^T, \quad \mathbf{x} \in \Omega$, where K is the number of objectives, Ω is the variables space and $\mathbf{x} = [x_1, x_2, ..., x_n]^T \in \Omega$ is a decision vector or possible solution to the problem. $\mathbf{F} : \Omega \to \mathbb{R}^K$ is a set of K-objective functions that maps the n-dimensional variables space to the K-dimensional objective space.

Pareto-dominance. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two decision vectors. $\mathbf{x}_1$ is said to dominate $\mathbf{x}_2$ (denoted by $\mathbf{x}_1 \prec \mathbf{x}_2$) if and only if $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ for all $i \in \{1, 2, \ldots, K\}$ and $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ for some $j \in \{1, 2, \ldots, K\}$. If a decision vector $\mathbf{x}$ is dominated by no other vector in Ω, $\mathbf{x}$ is said to be non-dominated.

Pareto-optimality. A decision vector $\mathbf{x}^* \in \Omega$ is said to be Pareto-optimal if and only if there is no $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^*$, i.e. $\mathbf{x}^*$ is a non-dominated solution. The Pareto-optimal Set (PS) is defined as the set of all Pareto-optimal solutions $\mathbf{x} \in \Omega$, i.e., $PS = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal}\}$. The set of all objective vectors $\mathbf{F}(\mathbf{x})$ such that $\mathbf{x}$ is Pareto-optimal is called Pareto front (PF), that is $PF = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^K | \mathbf{x} \in PS\}$.

Pareto-stationarity. Pareto-stationarity is a necessary condition for Pareto-optimality. For f_k differentiable everywhere for all k, $\mathbf{F}$ is said to be Pareto-stationary at $\mathbf{x}$ if there exists a set of scalars $\alpha_k, k \in \{1, \ldots, K\}$, such that $\sum_{k=1}^K \alpha_k \nabla f_k = \mathbf{0}, \quad \sum_{k=1}^K \alpha_k = 1, \quad \alpha_k \geq 0 \quad \forall k$.

Multiple Gradient Descent. Multiple gradient descent [6, 8, 9] was proposed for the unconstrained case of multi-objective optimization of $\mathbf{F}(\mathbf{x})$ assuming a convex, continuously differentiable and smooth $f_k(\mathbf{x})$ for all k. MGD finds a common descent direction for all f_k by defining the convex hull of all $\nabla f_k(\mathbf{x})$ and finding the minimum norm element within it. Consider $\mathbf{w}^*$ given by: $\mathbf{w}^* = \text{argmin}||\mathbf{w}||, \quad \mathbf{w} = \sum_{k=1}^K \alpha_k \nabla f_k(\mathbf{x}), \quad \text{s.t.} \quad \sum_{k=1}^K \alpha_k = 1, \quad \alpha_k \geq 0 \quad \forall k$. $\mathbf{w}^*$ will be either $\mathbf{0}$ in which case $\mathbf{x}$ is a Pareto-stationary point, or $\mathbf{w}^* \neq \mathbf{0}$ and then $\mathbf{w}^*$ is a descent direction for all $f_i(\mathbf{x})$. Similar to gradient descent, MGD consists in finding the *common* steepest descent direction $\mathbf{w}_t^*$ at each iteration t, and then updating parameters with a learning rate λ according to $\mathbf{x}_{t+1} = \mathbf{x}_t - \lambda \frac{\mathbf{w}_t^*}{||\mathbf{w}_t^*||}$.

3 Related work

3.1 Training GANs with multiple discriminators

One of the known difficulties in the vanilla GAN training is due to the discriminator quickly learning to distinguish real and generated samples [10], thus providing no meaningful error signal to improve the generator thereafter. Durugkar et al. [5] proposed the Generative Multi-Adversarial Networks (GMAN) which consist in training the generator against a *softmax* weighted arithmetic average of K different discriminators, according to:

$$\mathcal{L}_G = \sum_{k=1}^{K} \alpha_k \mathcal{L}_{D_k}, \tag{1}$$

where $\alpha_k = \frac{e^{\beta \mathcal{L}_{D_k}}}{\sum_{j=1}^K e^{\beta \mathcal{L}_{D_j}}}$, $\beta \geq 0$, and $\mathcal{L}_{D_k}$ is the loss of discriminator k and defined as:

$$\mathcal{L}_{D_k} = -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log D_k(\mathbf{x}) - \mathbb{E}_{\mathbf{z} \sim p_z} \log(1 - D_k(G(\mathbf{z}))), \tag{2}$$

where $D_k(\mathbf{x})$ and $G(\mathbf{z})$ are the outputs of the k-th discriminator and the generator, respectively. The goal of using the proposed averaging scheme is to privilege discriminators with respect to which generator's performance is worse. Experiments were performed with $\beta = 0$ (equal weights), $\beta \to \infty$ (only worst discriminator is taken into account), $\beta = 1$, and β learned by the generator. Models with $K = \{2, 5\}$ were tested and evaluated using a proposed metric and the Inception score [11]. However, results showed that the simple average of discriminator's losses provided the best scores in most of the considered cases. Neyshabur et al. [4] proposed training a GAN using K discriminators having the same architecture. Each discriminator D_k sees a different randomly projected lower-dimensional version of inputs. Random projections are defined by a random matrix W_k, which remains fixed during training. An upper bound provided shows the distribution induced by the generator will approximate the real data distribution as long as there is a sufficient number

of discriminators. Intuitively, real and fake samples are more alike after projection, thus avoiding early convergence of discriminators, which leads to common stability issues in GAN training such as mode-collapse [10]. Losses of each discriminator $\mathcal{L}_{D_k}$ are the same as shown in Eq. 2. The generator loss $\mathcal{L}_G$ is defined as simply the average of the losses provided by each discriminator, i.e. $\mathcal{L}_G = -\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{\mathbf{z}\sim p_z}\log D_k(G(\mathbf{z}))$.

4 Multi-objective training of GANs with multiple discriminators

We introduce a variation of the GAN game such that the generator solves the following multi-objective problem: $\min \boldsymbol{\mathcal{L}}_G(\mathbf{x}) = [l_1(\mathbf{z}), l_2(\mathbf{z}), ..., l_K(\mathbf{z})]^T$, where $l_k = -\mathbb{E}_{z\sim p_z}\log D_k(G(z))$, $k \in \{1, ..., K\}$, is the loss provided by the k-th discriminator. Training proceeds with alternate updates of discriminators and the generator. Updates of each discriminator are performed to minimize the loss described in Eq. 2. A natural choice for generator's updates is MGD. However, computing the direction of steepest descent $\mathbf{w}^*$ before every parameter update step, as required in MGD, can be prohibitively expensive for large neural networks. Therefore, we propose an alternative scheme for multi-objective optimization and argue that both our proposal and previously published methods can all be viewed as performing computationally more efficient versions of MGD update rule without the burden of having to solve a quadratic program, i.e. computing $\mathbf{w}^*$ every iteration.

4.1 Hypervolume maximization for training GANs

Fleischer [12] has shown that maximizing the hypervolume defined by the region within a set of cost functions and a shared upper bound (referred to as *nadir point*) yields Pareto-optimal solutions. Since MGD converges to a set of Pareto-stationary points, hypervolume maximization results in a sub-set of the solutions obtained by MGD. We exploit such property and define the generator loss as the negative log-hypervolume $\mathcal{L}_G = -\mathcal{V} = -\sum_{k=1}^{K}\log(\eta - l_k)$, where η is an upper bound for all l_k. In Fig. 1 we provide an illustrative example for the case where $K = 2$. The highlighted region corresponds to $e^{\mathcal{V}}$. Since the nadir point $\boldsymbol{\eta}^*$ is fixed, $\mathcal{V}$ will only be maximized, and consequently $\mathcal{L}_G$ minimized, if each l_k is minimized. Moreover, by adapting the results shown in [13], the gradient of $\mathcal{L}_G$ with respect to any generator's parameter θ is given by:

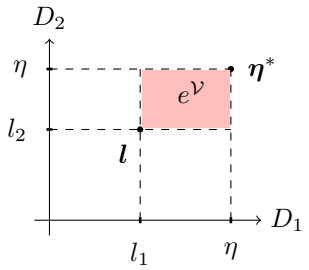

Figure 1: 2D example of the objective space where the generator loss is being optimized.

$$\frac{\partial \mathcal{L}_G}{\partial \theta} = \sum_{k=1}^{K}\frac{1}{\eta - l_k}\frac{\partial l_k}{\partial \theta}. \tag{3}$$

In other words, the gradient can be obtained by computing a weighted sum of the gradients of the losses provided by each discriminator, whose weights are defined as the inverse distance to the nadir point components. This formulation will naturally assign more importance to higher losses in the final gradient, which is another useful property of hypervolume maximization. Similarly to [13], we propose an adaptive scheme for η such that at iteration t: $\eta_t = \delta \max_k\{l_{k,t}\}$, where $\delta > 1$ is a user-defined hyperparameter. This enforces $\min_k\{\eta - l_k\}$ to be higher when $\max_k\{l_{k,t}\}$ is high and low otherwise, which induces a similar behavior as an average loss when training begins and places more importance on high loss discriminators as training progresses.

5 Experiments and Discussion

We first exploited the relatively low dimensionality of MNIST and used it as testbed for a comparison of MGD with the other approaches, i.e. average loss minimization (AVG) [4], GMAN's weighted average loss [5] and hypervolume maximization (HV). Experiments over 100 epochs with 8 discriminators are reported in Fig. 2 and Fig. 3. In Fig. 2, box-plots refer to 30 independent computations of the Fréchet Inception Distance (FID) [14] over 10000 images sampled from the generator which achieved the minimum FID at train time. FID results are measured at train time over 1000 images

and the best values are reported in Fig. 3 along with the necessary wall-clock time to achieve it.

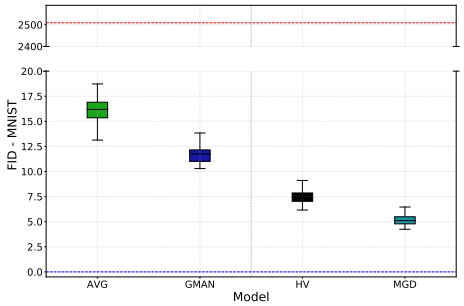

Figure 2: Box-plots corresponding to 30 independent FID computations with 10000 images. MGD performs consistently better than other methods, followed by hypervolume maximization. Models that achieved minimum FID at train time were used. Red and blue dashed lines are the FIDs of a random generator and real data, respectively.

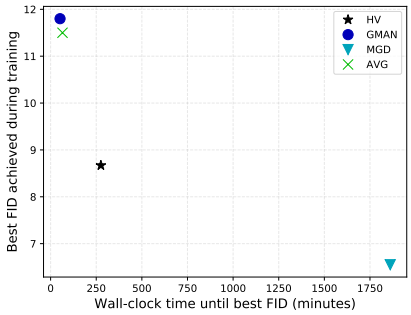

Figure 3: Time vs. best FID achieved during training for each approach. FID values are computed over 1000 generated images after every epoch. MGD performs relevantly better than others in terms of FID, followed by HV. However, MGD is approximately 7 times slower than HV. HV is well-placed in the time-quality trade-off.

We further evaluate the performance of HV compared to baseline methods using the CIFAR-10 dataset. DCGAN [15] and WGAN-GP [16] were included in the experiments for FID reference. Same architectures as in [4] were employed for all multi-discriminators settings. An increasing number of discriminators was used. A ResNet-18 [17] trained on CIFAR-10 until reaching approximately 95% test accuracy was used to compute FID values.

In Fig. 4, we report the box-plots of 15 independent evaluations of FID on 10000 images for the best model obtained with each method across 3 independent runs. Results once more indicate that HV outperforms other methods in terms of quality of the generated samples. Moreover, performance clearly improves as the number of discriminators grows. Furthermore, we repeat the experiments in [18] aiming to analyze how the number of discriminators impacts the sample diversity of the corresponding generator when trained using hypervolume maximization. The stacked MNIST dataset is employed and results are reported in Table 1.

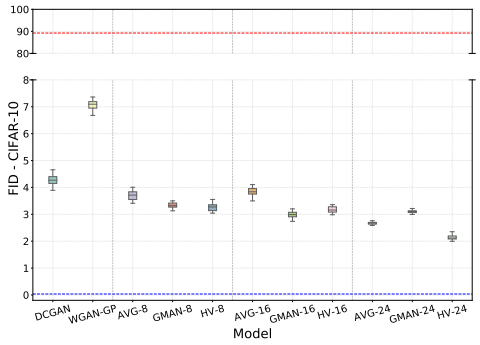

Figure 4: Box-plots of 15 independent FID computations with 10000 images. Dashed lines are real data (blue) and random generator (red) FIDs.

	Modes (Max 1000)	KL
DCGAN [15]	99.0	3.400
ALI [19]	16.0	5.400
Unrolled GAN [20]	48.7	4.320
VEEGAN [18]	150.0	2.950
PacDCGAN2 [21]	1000.0 ± 0.0	0.060 ± 0.003
HV - 8 disc.	776.8 ± 6.4	1.115 ± 0.007
HV - 16 disc.	1000.0 ± 0.0	0.088 ± 0.002
HV - 24 disc.	1000.0 ± 0.0	0.084 ± 0.002

Table 1: Number of covered modes and reverse KL divergence for stacked MNIST.

All evaluated models using HV outperformed DCGAN, ALI, Unrolled GAN and VEEGAN. Moreover, HV with 16 and 24 discriminators achieved state-of-the-art coverage values. The increase in "capacity" via using more discriminators directly resulted in an improvement in generator's coverage.

6 Conclusion

We introduced a multi-objective optimization framework for studying multiple discriminator GANs. The proposed approach was observed to consistently yield higher quality samples in terms of FID as compared to baseline methods. Furthermore, increasing the number of discriminators was shown to increase sample diversity. Such approach can be easily combined with other GAN training schemes.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, 2017.

[3] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *CoRR*, vol. abs/1703.10717, 2017. [Online]. Available: http://arxiv.org/abs/1703.10717

[4] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing GAN training with multiple random projections," *arXiv preprint arXiv:1705.07831*, 2017.

[5] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," *arXiv preprint arXiv:1611.01673*, 2016.

[6] J.-A. Désidéri, "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization," *Comptes Rendus Mathematique*, vol. 350, no. 5-6, pp. 313–318, 2012.

[7] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[8] S. Schäffler, R. Schultz, and K. Weinzierl, "Stochastic method for the solution of unconstrained vector optimization problems," *Journal of Optimization Theory and Applications*, vol. 114, no. 1, pp. 209–222, 2002.

[9] S. Peitz and M. Dellnitz, "Gradient-based multiobjective optimization with uncertainties," in *NEO 2016*. Springer, 2018, pp. 159–182.

[10] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[12] M. Fleischer, "The measure of pareto optima applications to multi-objective metaheuristics," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 519–533.

[13] C. S. Miranda and F. J. V. Zuben, "Single-solution hypervolume maximization and its use for improving generalization of neural networks," *CoRR*, vol. abs/1602.01164, 2016. [Online]. Available: http://arxiv.org/abs/1602.01164

[14] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6629–6640.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] A. Srivastava, L. Valkoz, C. Russell, M. U. Gutmann, and C. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 3310–3320.

[19] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[20] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[21] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "PacGAN: The power of two samples in generative adversarial networks," *arXiv preprint arXiv:1712.04086*, 2017.