Diversity Regularized Adversarial Deep Learning

Babajide O. Ayinde[1], Keishin Nishihama, and Jacek M. Zurada[1,2]

[1]Electrical and Computer Engineering Department,
University of Louisville, Louisville, KY, USA
[2]Information Technology Institute, University of Social Science, 90-113 Lodz, Poland
{babajide.ayinde,jacek.zurada}@louisville.edu
keishinkickback@gmail.com

Abstract. The two key players in Generative Adversarial Networks (GANs), the discriminator and generator, are usually parameterized as deep neural networks (DNNs). On many generative tasks, GANs achieve state-of-the-art performance but are often unstable to train and sometimes miss modes. A typical failure mode is the collapse of the generator to a single parameter configuration where its outputs are identical. When this collapse occurs, the gradient of the discriminator may point in similar directions for many similar points. We hypothesize that some of these shortcomings are in part due to primitive and redundant features extracted by discriminator and this can easily make the training stuck. We present a novel approach for regularizing adversarial models by enforcing diverse feature learning. In order to do this, both generator and discriminator are regularized by penalizing both negatively and positively correlated features according to their differentiation and based on their relative cosine distances. In addition to the gradient information from the adversarial loss made available by the discriminator, diversity regularization also ensures that a more stable gradient is provided to update both the generator and discriminator. Results indicate our regularizer enforces diverse features, stabilizes training, and improves image synthesis.

Keywords: Deep learning, feature correlation, generative model, adversarial learning, feature redundancy, generative adversarial networks, regularization.

1 Introduction

Convolutional neural networks (CNNs) have become the powerhouse for tackling many image processing and computer vision tasks. By design, CNNs learn to automatically optimize a well-defined objective function that quantifies the quality of results and their performance on the task at hand. As shown in previous studies [1], designing effective loss functions for many image prediction problems is daunting and often requires manual effort and in-depth experts' knowledge and insights. For instance, naively minimizing the Euclidean distance between predicted and ground truth pixels have shown to result in blurry outputs since the Euclidean distance is minimized by averaging all conceivable outputs [1–4]. One plausible way of training models with high-level objective specifications is by allowing CNNs to automatically learn the appropriate loss functions that satisfy these desired objectives. One of such objectives could be as simple as asking the model to make the output not distinguishable from the groundtruth.

As established in [1, 5–7], GANs are trained to automatically learn an objective function using a discriminator network to classify if its input is real or synthesized while simultaneously training a generative model to minimize the loss. In GAN framework, both the discriminator and generator aim to minimize their own loss and the

solution to the game is the Nash equilibrium where neither player can independently improve their individual loss [5, 8]. This framework can also be interpreted from the viewpoint of a statistical divergence minimization between the learned model distribution and the true data distribution [9–11].

Even though GANs have resulted in new and interesting applications and achieved promising performance, they are still hard to train and very sensitive to hyperparameter tuning. A peculiar and common training challenge is the performance control of the discriminator. The discriminator is usually inaccurate and unstable in estimating density ratio in high dimensional spaces, thus leading to situations where the generator finds it difficult to model the multi-modal landscape in true data distribution. In the event of total disjoint between the supports of model and true distributions, a discriminator can trivially distinguish between model distribution and that of true data [12], thus leading to situations where generator stops training because the derivative of the resulting discriminator with respect to the input has vanished. This problem has seen many recent works to come up with workable heuristics to address many training problems such as mode collapse and missing modes.

We argue in line with the hypothesis that some of the problems associated with the training of GANs are in part due to lack of control of the discriminator. In light of this, we propose a simple yet powerful diversity regularizer for training GANs that encourages the discriminator to extract near-orthogonal filters. The problem abstraction is that in addition to the gradient information from the adversarial loss made available by the discriminator, we also want the GAN system to benefit from extracting diverse features in the discriminator. Experimental results consistently show that, when correctly applied, the proposed regularization enforces diverse features in the discriminator and better stabilize the GAN training with mostly positive effects on the generated samples.

The contribution of this work is two-fold: (i) we propose a new method to regularize adversarial learning by inhibiting the learning of redundant features and availing a stable gradient for weights updates during training and (ii) we show that the proposed method stabilizes the adversarial training and enhances the performance of many state-of-the-art methods across many benchmark datasets. The rest of the paper is structured as follows: Section II highlights the state-of-the-art and Section III discusses in detail the formulation of diversity-regularized adversarial learning. Section IV discusses the detailed experimental designs and presents the results. Finally, conclusions are drawn in Section V.

2   Related Work

As originally introduced in [5], GANs consist of generator and the discriminator that are parameterized by deep neural networks and are capable of synthesizing interesting local structure on select datasets. The representation capacity of original GAN was extended in conditional GANs [13] by incorporating an additional vector that enables the generator to synthesize samples conditioned on some useful information. This extension has motivated several conditional variants of GAN in diverse applications such as edge map [14, 15], image synthesis from text [16], super-resolution [17], style transfer [18], just to mention a few. Learning useful representation with GANs has shown to heavily rely on hyperparameter-tuning due to various instability issues during training [8, 12, 19]. GANs are remarkably hard to train in spite of their success on variety of task. Robustly and systematically stabilizing the training of GANs has come in many forms such as selective architectural design [6], matching of intermediate features [7], and unrolling the optimization of discriminator [20].
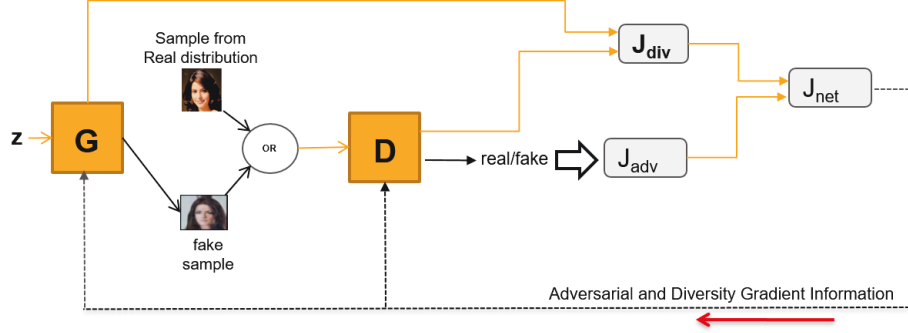
Fig. 1. Schema of Diversity Regularized Adversarial Learning (DiReAL)

Many recent advances inspired by either theoretical insights or practical considerations have been attempted in form of regularization and normalization to address some of the issues associated with training of GANs. Imposing Lipschitz constraint on the discriminator has shown to stabilize the adversarial training and avoid an over-optimization scenario where the discriminator still distinguishes and allots different scores to nearly indistinguishable samples [12]. By satisfying the Lipschitz constraint, the discriminator's joint/compressed representation of the true and synthesized data distributions is guaranteed to be smooth; thus ensuring a non-zero learning signal for the generator [12,19]. Enforcing the discriminator to satisfy the Lipschitz constraints has been approximated and implemented via ancillary means such as gradient penalties [21] and weight clipping [12]. Using a Gaussian classifier over the real/fake indicator variables has also been shown to have a smoothing effect on the discriminator function [19]. Injecting label noise [7] and gradient penalty have equally been shown to have a tremendous regularizing effect on GANs. Schemes such as weighted gradient [22] and missing modes penalty [23] have been utilized to alleviate some training and missing modes issues in GAN learning.

Weight vectors of discriminator have been l_2-normalized with Frobenius norm, which constraints the sum of the squared singular values of the weight matrix to be 1 [7]. However, normalizing using Frobenius norm translates to utilizing a single feature to discriminate the model probability distribution from the target thus, reducing the rank and hence the number of discriminator features [24]. In addition to weight clipping [10,12], weight normalization approaches yield primitive discriminator model that maps the target distribution only with select few features. The most closely related work to ours is orthonormal regularization of weights [25] that sets all the singular values of weight matrix in the discriminator to one, which translates to using as many features as possible to distinguish the generator distribution from the target distribution. Our approach, however, imposes much softer orthogonality constraint on the weight vectors by allowing a degree of feature sharing in upper layers of the discriminators. Other related work is spectral normalization of weights that guarantees 1-Lipschitzness for linear layers and ReLu activation units resulting in discriminators of higher rank [24]. The advantage of spectral normalization is that weight matrices are constrained and Lipschitz. However, bounding the spectral norm of the convolutional kernel to 1 does not bound the spectral norm of the convolutional mapping to unity.

3   Method

The training of GAN can be abstracted as a non-cooperative game between two players, namely the generator G and the discriminator D. The discriminator tries to distinguish if the generated sample is from the real (p_{data}) or fake data distribution (p_z), while G tries to trick D into believing that generated sample is from p_{data} by moving the generation manifold towards the data manifold. The discriminator aims to maximize $\mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[log D(\mathbf{x})]$ when the input is sampled from real distribution and given a fake image sample $G(\mathbf{z})$, $\mathbf{z} \sim p_z(\mathbf{z})$, it is trained to output probability, $D(G(\mathbf{z}))$, close to zero by maximizing $\mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[log(1 - D(G(\mathbf{z})))]$. The generator network, however, is trained to maximize the chances of D producing a high probability for a fake image sample $G(\mathbf{z})$ thus by minimizing $\mathbb{E}_{\mathbf{z} \sim p_z}[log(1 - D(G(\mathbf{z})))]$.

The adversarial cost is obtained by combining the objectives of both D and G in a min-max game as given in 1 below:

$$J_{adv} = \min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[log D(\mathbf{x})]$$
$$+ \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[log(1 - D(G(\mathbf{z})))] \tag{1}$$

Training D can be conceived as training an evaluation metric on sample space [23] that enables G to use the local gradient $\nabla \log D(G(\mathbf{z}))$ information made available by D to improve itself and move closer to the data manifold.

3.1   Feature diversification in GAN

Both D and G are commonly parameterized as DNNs and over the past few years, the general trend has been that DNNs have grown deeper, amounting to huge increase in number of parameters. The number of parameters in DNNs is usually very large offering possibility to learn very flexible high-performing models [26]. Observations from many previous studies [27–30] suggest that layers of DNNs typically rely on many redundant filters that can be either shifted version of each other or be very similar with little or no variations. For instance, this redundancy is evidently pronounced in filters of AlexNet [31] as emphasized in [28, 32, 33]. To address this redundancy problem, we train layers of the discriminator under specific and well-defined diversity constraints.

Since G and D rely on many redundant filters, we regularize them during training to provide more stable gradient to update both G and D. Our regularizer enforces constraints on the learning process by simply encouraging diverse filtering and discourages D from extracting redundant filters. We remark that convolutional filtering has found to greatly benefit from diversity or orthogonality of filters because it can alleviate problems of gradient vanishing or exploding [25, 34–36].

Typically, both D and G consist of input, output, and many intermediate processing layers. By letting the number of channels, height, and width of input feature map for l^{th} layer be denoted as n_l, h_l, and w_l, respectively. A convolutional layer in both D transforms input $\mathbf{x}_l \in \mathbb{R}^p$ into output $\mathbf{x}_{l+1} \in \mathbb{R}^q$, where $\mathbf{x}_{l+1}$ is the input to layer $l + 1$; p and q are given as $n_l \times h_l \times w_l$ and $n_{l+1} \times h_{l+1} \times w_{l+1}$, respectively. $\mathbf{x}_l$ is convolved with n_{l+1} $3D$ filters $\chi \in \mathbb{R}^{n_l \times k \times k}$, resulting in n_{l+1} output feature maps. Unrolling and combining all layer l^{th} filters into a single matrix results in kernel matrix $\overset{(l)}{\Theta^D} \in \mathbb{R}^{m \times n_{l+1}}$ where $m = k^2 n_l$. Then, $\overset{(l)}{\theta^D}_i$, i=1,...$n_l$, denotes filters in layer l, each $\overset{(l)}{\theta^D}_i \in \mathbb{R}^m$ corresponds to the i-th column of the kernel matrix

$\overset{(l)}{\Theta^D} = [\overset{(l)}{\theta^D}_1, \ ... \overset{(l)}{\theta^D}_{n_l}] \in \mathbb{R}^{m \times n_{l+1}}$; the bias term of each layer is omitted for simplicity. Given that $\overset{(l)}{\Theta^D} \in \mathbb{R}^{m \times n_l}$ contain n_l normalized filter vectors as columns, each with m elements corresponding to connections from layer $l-1$ to i^{th} neuron of layer l, then, the diversity loss J_D for all layers of D is given as:

$$J_D(\theta^D) = \sum_{l=1}^{L} \left(\frac{1}{2} \sum_{i=1}^{n_l} \sum_{j=1}^{n_l} \left(\overset{(l)}{\Omega_{ij}^D} \right)^2 \overset{(l)}{\mathbf{M}_{ij}^D} \right) \tag{2}$$

where $\overset{(l)}{\Omega^D} \in \mathbb{R}^{n_l \times n_l}$ denotes $(\overset{(l)}{\Theta^D})^T \overset{(l)}{\Theta^D}$ which contains the inner products of each pair of columns i and j of $\overset{(l)}{\Theta^D}$ in each position i,j of $\overset{(l)}{\Omega^D}$ in layer l; $\overset{(l)}{\mathbf{M}^D} \in \mathbb{R}^{n_l \times n_l}$ is a binary mask for layer l defined in (5); L is the number of layers to be regularized.

$$\overset{(l)}{\mathbf{M}_{ij}^D} = \begin{cases} 1 & \tau \leq |\overset{(l)}{\Omega_{ij}^D}| \leq 1 \\ 0 & i = j \\ 0 & otherwise \end{cases} \tag{3}$$

Similarly, the diversity loss J_G for generator G is given as:

$$J_G(\theta^G) = \sum_{l=1}^{L} \left(\frac{1}{2} \sum_{i=1}^{n_l} \sum_{j=1}^{n_l} \left(\overset{(l)}{\Omega_{ij}^G} \right)^2 \overset{(l)}{\mathbf{M}_{ij}^G} \right) \tag{4}$$

and

$$\overset{(l)}{\mathbf{M}_{ij}^G} = \begin{cases} 1 & \tau \leq |\overset{(l)}{\Omega_{ij}^G}| \leq 1 \\ 0 & i = j \\ 0 & otherwise \end{cases} \tag{5}$$

In order to enforce feature diversity in both G and D while training GANs, the diversity regularization terms in (4) is added to the conventional adversarial cost J_{adv} in (1) as given in (6).

$$J_{net} = J_{adv} + J_{div} \tag{6}$$

where $J_{div} = \lambda_G J_G(\theta^G) - \lambda_D J_D(\theta^D)$, λ_G and λ_D is the diversity penalty factors for generator and discriminator, respectively. The derivative of diversity loss J_D with respect to weights of D is given as

$$\nabla_{\Theta_{i,j}^{(l)}} J_D(\theta^D) = \sum_{k=1}^{n} \overset{(l)}{\Theta_{i,k}^D} \overset{(l)}{\Omega_{k,j}^D} \overset{(l)}{\mathbf{M}_{k,j}^D} \tag{7}$$

and the derivative of diversity loss J_G with respect to weights of G is

$$\nabla_{\Theta_{i,j}^{(l)}} J_G(\theta^G) = \sum_{k=1}^{n} \overset{(l)}{\Theta_{i,k}^G} \overset{(l)}{\Omega_{k,j}^G} \overset{(l)}{\mathbf{M}_{k,j}^G} \tag{8}$$

The idea behind diversifying features is that in addition to adversarial gradient information provided by D, we provide additional diversity loss with more stable gradient to refine both G and D. The diversity loss encourages weights of both generator and

(a)

(b)

(c)

(d)
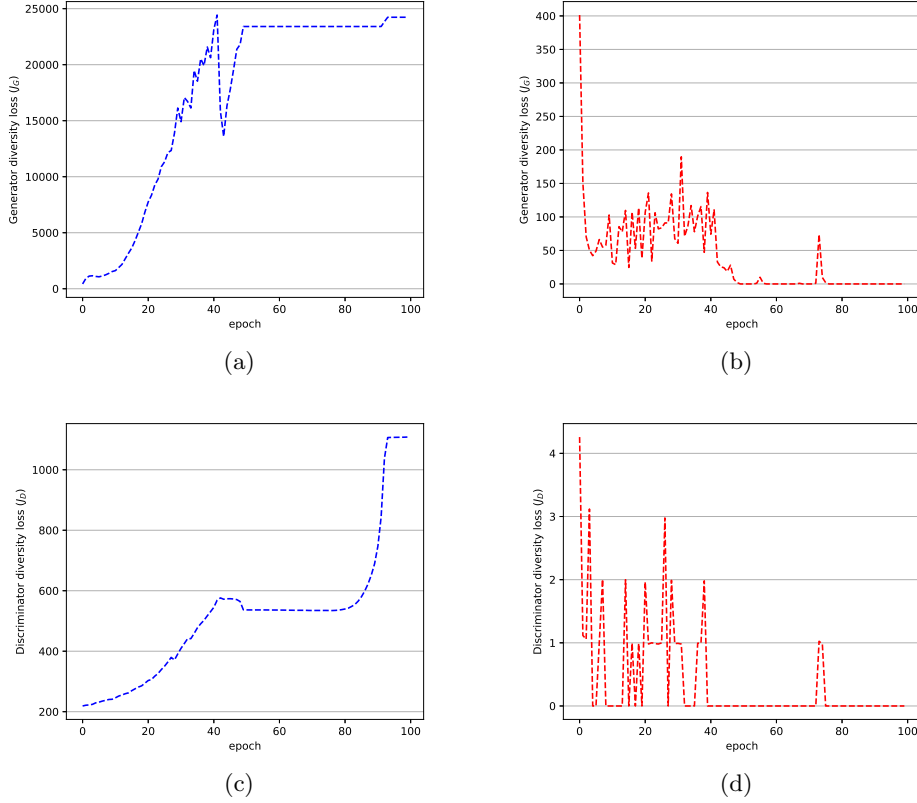
Fig. 2. Diversity loss of (a) generator J_G with no regularization (b) generator J_G with diReAL (c) discriminator J_D with no regularization, and (d) discriminator J_D with DiReAL trained on MNIST dataset.

discriminator to be diverse by pushing them towards the nearest orthogonal manifold. Our proposed regularization provides more efficient gradient flow, a more stable optimization, richness of layer-wise features of resulting model, and improved sample quality compared to benchmarks and baseline. The diversity regularization ensures the column space of $\overset{(l)}{\Theta^D}$ and $\overset{(l)}{\Theta^G}$ for l^{th} layer does not concentrate in few direction during training thus preventing them to be sensitive in few and limited directions. The proposed diversity regularized adversarial learning alleviates some of the main failure mode of GAN by ensuring features are diverse.

4   Experiments

All experiments were performed on Intel(r) Core(TM) i7-6700 CPU @ 3.40Ghz and a 64GB of RAM running a 64-bit Ubuntu 16.04 edition. The software implementation has been in PyTorch library [1] on two Titan X 12GB GPUs. Implementation of DiReAL is available at `https://github.com/keishinkickback/DiReAL`. Diversity regularized adversarial learning (DiReAL) was evaluated on MNIST dataset of handwritten digits [37], CIFAR-10 [38], STL-10 [39], and Celeb-A [40] databases. In the

---
[1] https://pytorch.org/

first set of experiments, an ubiquitous deep convolutional GAN (DCGAN) in [6] was trained using MNIST digits. The standard MNIST dataset has 60000 training and 10000 testing examples. Each example is a grayscale image of an handwritten digit scaled and centered in a 28×28 pixel box. Both the discriminator and generator networks contain 5 layers of convolutional block. Adam optimizer [41] with batch size of 64 was used to train the model for 100 epochs and τ and learning rate in DiReAL were set to 0.5 and 0.0001, respectively. In similar vein, λ_D and λ_G were to 1.0 and 0.01, respectively. Adam optimizer ($\beta_1 = 0.0$, $\beta_2 = 0.9$) [41] with batch size of 64 was used to train the model for 100 epochs

Fig. 2 shows the diversity loss of both generator and discriminator for DiReAL and unregularized counterpart. It can be observed that DiReAL was able to minimize the pairwise feature correlations compared to the highly correlated features extracted by the unregularized counterpart. Specifically, DiReAL was able to steadily minimize the diversity loss as training progresses compared to the unregularized DCGAN, where extraction of similar features grows with epoch of training, thus increasing the diversity loss. The divergence between discriminator output for real handwritten digits and generated samples over 30 batches for regularized and the unregularized networks is shown in Fig. 3a. The divergence was measured using the Wasserstein distance measure [42] and it can be observed that the regularizing effect of DiReAL stabilizes the adversarial training and prevents mode collapse. For unregularized network, however, the mode started to collapse around 45th epoch. Closer look into the diversity of the generator in Fig. 2a, it is evident that just around the epoch of collapse the generator starts extracting more and more redundant filters. We suspect that DiReAL was able to stabilize the training by pushing features to lie close to the orthogonal manifold, thus preventing learned features from collapsing to an undesirable manifold. Fig. 3b shows the handwritten digit samples synthesized with and without DiReAL and it can be observed that diversification of features is beneficial for stabilizing adversarial learning and ultimately improving the samples' quality. Another observation is that DiReAL also prevents learned weights from collapsing to an undesirable manifold thus highlighting some of the benefits of pushing weights near the orthogonal manifold.

In the second large-scale experiments, CIFAR-10 dataset was used to train GAN

(a)   (b)
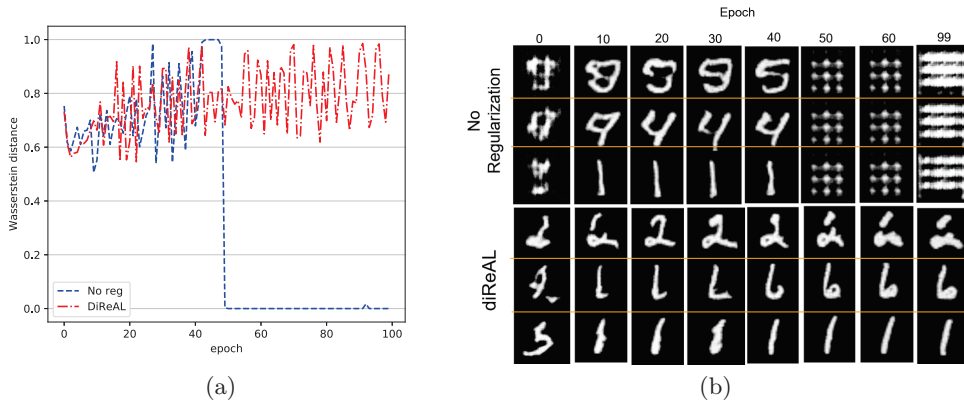
Fig. 3. (a) Divergence, as measured by Wasserstein distance, between the discriminator output for synthesized and real MNIST samples (b) Synthesized hand-written digits with and without diversity regularization.

(a)   (b)
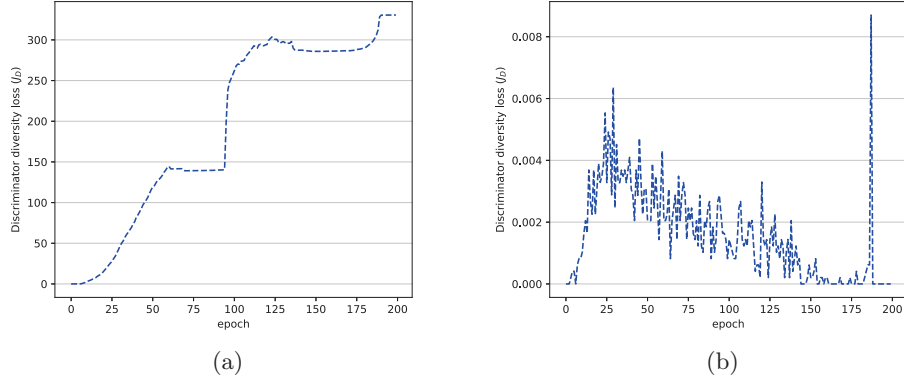
Fig. 4. Diversity loss of (a) discriminator J_D with no regularization, and (b) discriminator J_G with diReAL trained on CIFAR-10 dataset.
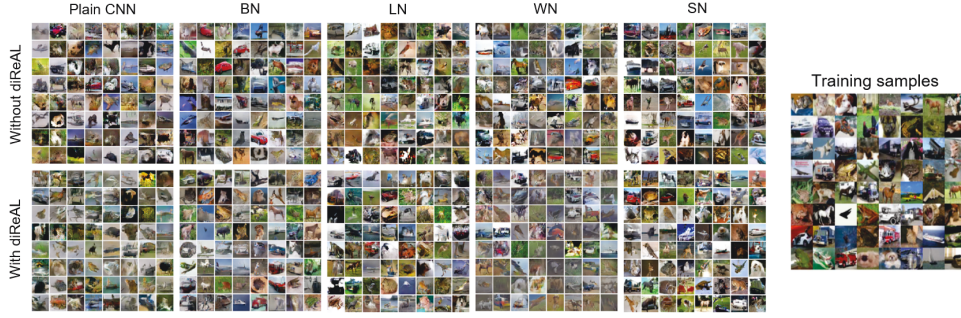
Fig. 5. Generated images with and without DiReAL trained on CIFAR-10 dataset.
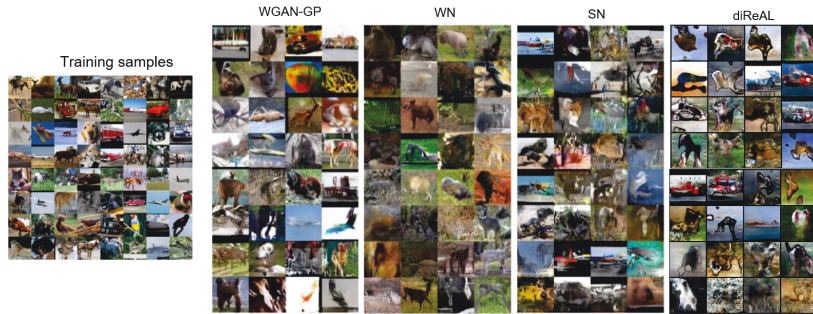
Fig. 6. Qualitative comparison of generated images with four regularization techniques for models trained on STL-10 dataset.

using DiReAL and the results compared to the unregularized training. The dataset is split into 50000 and 10000 training and testing sets, respectively. Similar to experiments with MNIST, Fig. 4b shows the diversity loss of the discriminator with and without DiReAL trained on CIFAR-10 database. It can be observed that DiReAL was able to minimize the diversity loss and encourages diverse features that benefit the adversarial training. On the other hand, Fig. 4b shows that the diversity loss of the unregularized is higher and unconstrained compared to that of DiReAL. The images

Method	Inception Score
Real data	9.04
-Standard CNN-	
Unregularized [6]	4.00 ± 0.15
DiReAL (ours)	4.17 ± 0.03
Batch Normalization [43]	5.48 ± 0.19
Layer Normalization [44]	5.05 ± 0.12
Weight Normalization [45]	4.66 ± 0.14
Spectral Normalization [24]	$\mathbf{6.50} \pm 0.30$
Weight Normalization + DiReAL	4.68 ± 0.06
Batch Normalization + DiReAL	5.48 ± 0.15
Layer Normalization + DiReAL	5.64 ± 0.15
Spectral Normalization + DiReAL	$\mathbf{6.87} \pm 0.12$

Table 1. Inception Scores with unsupervised image generation on CIFAR-10

Fig. 7. Generated images with and without diversity Regularization trained on CELEB-A dataset.

synthesized with DiReAL was compared and contrasted with state-of-the-art methods such as batch normalization [43], layer normalization [44], weight normalization [45], and spectral normalization [24]. It is remarked that DiReAL can be used in tandem with the other regularization techniques and could also be deployed as stand-alone regularization tool for stabilizing adversarial learning. In this light, DiReAL was also combined with these techniques. It must be noted that spectral normalization uses a variant of DCGAN architecture with an eight-layer discriminator network. See [24] for more implementation details.

It can be observed in Fig. 5 that diversity regularization was able to synthesize more diverse and complex images compared to unregularized counterpart. Other benchmark regularizers were able to generate better image samples compared to using only DiReAL. However, when DiReAL was combined with other regularizers the quality of the generated samples was significantly improved. For quantitative evaluation of generated examples, inception score metric [45] was used. Inception score has been found

to highly correlate with with subjective human judgment of image quality [24, 45]. Similar to [24, 45], inception score was computed for 5000 synthesized images using generators trained with each regularization technique. Every run of the experiment is repeated five times and averaged to combat the effect of random initialization. The average and the standard deviation of the inception scores are reported.

The proposed regularization is also compared and contrasted in terms inception score with many benchmark methods as summarized in Table 1. It can be again observed that DiReAL was able to improve the image generation quality compared to unregularized counterpart and when combined with spectral normalization, we observed a 6% improvement in the inception score. By combining DiReAL with layer normalization, an improvement of 11.68% on inception was observed. However, no significant improvement was observed when DiReAL was combined with batch normalization and weight normalization. It must be remarked that the calculation of Inception Scores is library dependent and that is why the scores reported in Table 1 is different for those reported by Miyato et al. [24]. While our implementation was in PyTorch, [24] was in Chainer [2].

In the next set of large-scale experiments, STL-10 dataset was used to train generator under diversity regularization and compared with other state-of-the-art regularization techniques. As can be observed in Fig. 6, images synthesized by generator trained with DiReAL was able to generate images with competitive quality in comparison with other regularization methods considered. Performance of DiReAL was also observed to be competitive to regularization methods such as WGAN-GP and spectral normalization. In Fig. 7 we show the images produced by the generators trained with DiReAL using Celeb-A dataset. It can be again be observed that DiReAL was able to stabilize the training and avoid mode collapse in comparison to the unregularized counterpart.

5 Conclusion

This paper proposes an interesting and effective method of stabilizing the training of GANs using diversity regularization to penalize both negatively and positively correlated features according to features differentiation and based on features relative cosine distances. It has been shown that diversity regularization can help alleviate a common failure mode where the generator collapses to a single parameter configuration and outputs identical points. This has been achieved by providing additional stable diversity gradient information in addition to adversarial gradient information to update both the generator and discriminator's features. The performance of the proposed regularization in terms of extracting diverse features and improving adversarial learning was compared on the basis of image synthesis with recent regularization techniques namely batch normalization, layer normalization, weight normalization, weight clipping, WGAN-GP, and spectral normalization. It has also been shown on select examples that extraction of diverse features improves the quality of image generation, especially when used in combination with spectral normalization. This concept is illustrated using MNIST handwritten digits, CIFAR-10, STL-10, and Celeb-A Dataset.

Acknowledgement

---

[2] https://chainer.org/

References

1. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
2. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.
3. R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision.* Springer, 2016, pp. 649–666.
4. B. O. Ayinde and J. M. Zurada, "Deep learning of constrained autoencoders for enhanced understanding of data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 3969–3979, Sep. 2018.
5. 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.
6. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
7. 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.
8. K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, "The gan landscape: Losses, architectures, regularization, and normalization," 2018.
9. S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.
10. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.
11. X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on.* IEEE, 2017, pp. 2813–2821.
12. M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.
13. M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
14. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on.* IEEE, 2017, pp. 2242–2251.
15. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2017, pp. 5967–5976.
16. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *33rd International Conference on Machine Learning*, 2016, pp. 1060–1069.
17. C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.* IEEE, 2017, pp. 105–114.
18. S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer."
19. K. Grewal, R. D. Hjelm, and Y. Bengio, "Variance regularizing adversarial learning," *arXiv preprint arXiv:1707.00309*, 2017.
20. L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *ICLR*, 2017.
21. 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. 5767–5777.

22. K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Advances in Neural Information Processing Systems*, 2017, pp. 2018–2028.
23. T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *ICLR*, 2017.
24. T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *ICLR*, 2018.
25. A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks," *arXiv preprint arXiv:1609.07093*, 2016.
26. C. Liu, Z. Zhang, and D. Wang, "Pruning deep neural networks by optimal brain damage," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
27. P. Xie, Y. Deng, and E. Xing, "On the generalization error bounds of neural networks under diversity-inducing mutual angular regularization," *arXiv preprint arXiv:1511.07110*, 2015.
28. P. Rodríguez, J. Gonzàlez, G. Cucurull, J. M. Gonfaus, and X. Roca, "Regularizing cnns with locally constrained decorrelations," *arXiv preprint arXiv:1611.01967*, 2017.
29. A. Dundar, J. Jin, and E. Culurciello, "Convolutional clustering for unsupervised learning," *arXiv preprint arXiv:1511.06241*, 2015.
30. B. O. Ayinde and J. M. Zurada, "Building efficient convnets using redundant feature pruning," *arXiv preprint arXiv:1802.07653*, 2018.
31. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
32. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.
33. B. O. Ayinde and J. M. Zurada, "Clustering of receptive fields in autoencoders," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 1310–1317.
34. B. O. Ayinde, T. Inanc, and J. M. Zurada, "Regularizing deep neural networks by enhancing diversity in feature extraction," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2019.
35. A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *ICLR*, 2014.
36. B. O. Ayinde and J. M. Zurada, "Nonredundant sparse feature extraction using autoencoders with receptive fields clustering," *Neural Networks*, vol. 93, pp. 99–109, 2017.
37. Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.
38. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical report, University of Toronto*, 2009.
39. A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.
40. Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3730–3738.
41. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
42. S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.
43. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
44. J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
45. T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.