Self Supervised GANs via Auxiliary loss and Learning rate Decay

Rohan Raj
New York University
rr2685@nyu.edu

Sushanth Samala
New York University
sushanthsamala@nyu.edu

Abstract

Conditional GANs are at the forefront of natural image synthesis. The main drawback of such models is the need for labelled data. In this work, we use two popular unsupervised learning techniques self-supervision and Learning rate Decay. Self-supervision ensures discriminator learns meaningful feature representations from the data and retains them. We propose to use Learning Rate Decay strategy to adjust the learning rate for each iteration and subsequently for each epoch, it can make the model converge considerably faster. Finally, we show that this approach to self-supervised learning attains an FID of 215.51 on MNIST synthesis and an FID of 226.76 on CIFAR images synthesis.

1 Introduction

Image synthesis is a central problem in computer vision. There has been remarkable progress in this direction with the emergence of Generative Adversarial Networks (GANs)[1]. A generative adversarial network (GAN) is a class of machine learning systems invented by Ian Goodfellow and his colleagues in 2014.[1] Two neural networks contest with each other in a game. Given a training set, this technique learns to generate new data with the same statistics as the training set. In a 2016 seminar, Yann LeCun described GANs as "the coolest idea in machine learning in the last twenty years"[2].

The generative network generates candidates while the discriminative network evaluates them.[1] The contest operates in terms of data distributions. Typically, the generative network learns to map from a latent space to a data distribution of interest, while the discriminative network distinguishes candidates produced by the generator from the true data distribution. The generative network's training objective is to increase the error rate of the discriminative network. A known dataset serves as the initial training data for the discriminator. Training it involves presenting it with samples from the training dataset, until it achieves acceptable accuracy. The generator trains based on whether it succeeds in fooling the discriminator. Typically the generator is seeded with randomized input that is sampled from a predefined latent space (e.g. a multivariate normal distribution). Thereafter, candidates synthesized by the generator are evaluated by the discriminator. Backpropagation is applied in both networks so that the generator produces better images, while the discriminator becomes more skilled at flagging synthetic images.[3] The generator is typically a de-convolutional neural network, and the discriminator is a convolutional neural network. Training GANs is challenging because one searches for a Nash equilibrium of a non-convex game in a high-dimensional parameter space. GANs are typically trained with alternating stochastic gradient descent. However, this training procedure is unstable and lacks guarantees.

Self-supervised learning is autonomous supervised learning. It is a representation learning approach that eliminates the pre-requisite requiring humans to label data. Self-supervised learning systems extract and use the naturally available relevant context and embedded metadata as supervisory signals. Coupled with GANs generate realistic looking images. However, by carefully examining the generated samples from these models, we can observe that convolutional GANs [4] have much more

Preprint. Under review.

difficulty in modeling some image classes than others when trained on datasets. Unconditional GANs have shown remarkable success in generating realistic, high quality samples when trained on class specific datasets. Recent advances in machine learning offer an opportunity to substantially improve the quality of image models.

In this work, we use GANs with self-supervised learning and a fixed learning with a method of decaying learning rate to control the magnitude of the update. Learning Rate Decay strategy can speed up convergence and improve model accuracy, we demonstrate its superiority through GANs.

2 Challenge

Discriminator Forgetting: GANs involve training two networks in an adversarial game, where each network's task depends on its adversary. Recently, several works have framed GAN training as an online or continual learning problem[5]. We focus on the discriminator, which must perform classification under an (adversarially) shifting data distribution. When trained on sequential tasks, neural networks exhibit forgetting. For GANs, discriminator forgetting leads to training instability. To counter forgetting, we encourage the discriminator to maintain useful representations by adding a self-supervision.[6] The original value function for GAN training is:

$$V(G, D) = E_{x \sim P_{data}(x)}[\log P_D(S = 1|x)] + E_{x \sim P_G(x)}[\log(1 - P_D(S = 1|x))] \text{ [1]}$$

Where P_{data} is the true data distribution, and P_G is the data distribution induced by feeding noise drawn from a simple distribution z ~ P(z) through the generator, x = G(z). P_D(S|x) is the discriminator's Bernoulli distribution over the sources (real or fake).

The generator maximizes this Equation, while the discriminator minimizes it. In the training phase, neural networks tend to forget previous tasks [7]. Goodfellow et al. [1] show that the optimal discriminator estimates the likelihood ratio between the generated and real data distributions. Therefore, given a perfect generator, where PG = PD, the optimal discriminator simply outputs 0.5, and has no requirement to retain any meaningful representation. Discriminator forgetting may cause training difficulties[6] because it does not learn meaningful representations to guide the generator, and the generator can revert to generating old images to fool it. Therefore, we add self-supervision to encourage the discriminator to retain useful representations.

3 Method

Self-supervised learning is a family of methods for building representations from unsupervised data. Self-supervision works by creating artificial supervised tasks from unsupervised data, training on these tasks, and then extracting representations from the resulting networks[6]. Here, we apply the successful image rotation self-supervision method [8]. In this method, the self-supervised network predicts the angle of rotation of an image. In Figure 1(b) we motivate this loss using our toy problem. When we add the self-supervised loss, the network learns features that transfer across tasks; performance continually improves, and does not drop to 0.5 when the distribution shifts.

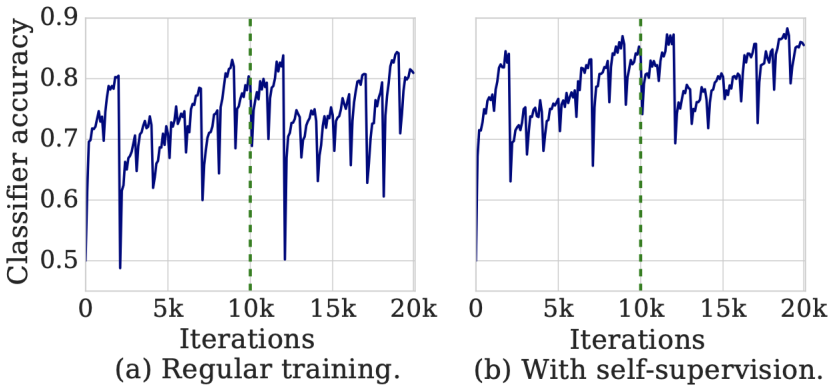

Figure 1: Image classification accuracy when the underlying class distribution shifts every 1k iterations[9]. The vertical dashed line indicates the end of an entire cycle, and return to the original classification task at t = 0. Left: vanilla classifier. Right: classifier with additional self-supervised loss. This example demonstrates that a classifier/discriminator may fail to learn generalizable representations in a non-stationary environment.

For the self-supervised GAN, the specific losses we use for the generator and discriminator are:

$$L_G = -V(G,D) - \alpha E_{x\sim P_G}[\sum_{r\in R} \log P'_D(R=r|x^r)], (2) r \in R$$
$$L_D = V(G,D) - \beta E_{x\sim P_{data}}[\sum_{r\in R} \log P'_D(R=r|x^r)], (3) r \in R$$

where V (G, D) is the original GAN loss in Equation 1. r $\in$ R is a rotation selected from a set of possible rotations. We use $\{0°, 90°, 180°, 270°\}$ as in Gidaris et al. [8]. PD' (R|xr) is the discriminator's distribution over rotations, and xr is the image x transformed by rotation r.

A Note on Convergence: With $\alpha > 0$ convergence, even under optimal conditions, to the true data distribution $P_G = P_{data}$ is not guaranteed. This may not be a concern because current GANs are far from attaining the optimal solution. If it is, one could anneal α to zero during training. Our intuition is that the proposed loss encourages the discriminator to learn and retain meaningful representations that allow it to distinguish rotations as well as true/fake images. The generator is then trained to match distributions in this feature space which encourages the generation of realistic objects.[6]

4 Related Work

Current attempts at unsupervised representation learning by rotating images propose a self-supervised task that is very simple and at the same time, offers a powerful supervisory signal for semantic feature learning[8]. This method exhaustively evaluates self-supervised method under various settings (e.g. semi-supervised or transfer learning settings) and in various vision tasks (i.e., CIFAR-10, ImageNet, Places, and PASCAL classification, detection, or segmentation tasks). In all of them, the self-supervised formulation demonstrates state-of-the-art results with dramatic improvements w.r.t. prior unsupervised approaches. As a consequence we see that for several important vision tasks, self-supervised learning approach significantly narrows the gap between unsupervised and supervised feature learning.[8]

One key method is using auxiliary rotation loss as in [9] which propose a deep generative model that combines adversarial and self-supervised learning. The resulting self-supervised model match equivalent conditional GANs on the task of image synthesis, without having access to labeled data. The self-supervised GAN could be used in a semi-supervised setting where a small number of labels could be used to fine-tune the model.

Our work is strongly influenced by the adversarial training and self-supervision techniques used in [9] and semantic feature learning in [8]. We implement an unsupervised generative model that combines adversarial training with self-supervised learning, thus recovering the benefits of conditional GANs, but without the requirement of labeled data. Finally we augment the model with learning rate decay which reduces the learning rate for each iteration thus shrinking the learning rate.

5 Implementation

We implement a mechanism with which the discriminator is allowed to learn useful representations, independently of the quality of the current generator. The main idea behind self-supervision is to train a model on a pretext task like predicting rotation angle or relative location of an image patch, and then extracting representations from the resulting networks[10].

We start off by implementing the self-supervised method based on image rotation[8]. In this method, the images are rotated, and the angle of rotation becomes the artificial label. The self-supervised

task is then to predict the angle of rotation of an image. When coupled with the self-supervised loss, the network learns representations that transfer across tasks and the performance continually improves. On the second cycle through the tasks, from 10k iterations onward, performance is improved. Intuitively, this loss encourages the classifier to learn useful image representations to detect the rotation angles, which transfers to the image classification task. We augment the model with Wasserstein Loss, weight normalization which is a reparameterization that decouples the magnitude of a weight tensor from its direction. Weight normalization is implemented via a hook that recomputes the weight tensor from the magnitude and direction. Wasserstein Loss is an extension of the GAN that seeks an alternate way of training the generator model to better approximate the distribution of data observed in a given training dataset. Instead of using a discriminator to classify or predict the probability of generated images as being real or fake, the WGAN changes or replaces the discriminator model with a critic that scores the realness or fakeness of a given image.This approach did not give us the results we expected.

In another attempt we implemented Hinge Loss along with weight normalization, hinge loss is defined by

$$V(f(\vec{x}), y) = \max(0, 1 - yf(\vec{x})) = |1 - yf(\vec{x})|_+$$

The hinge loss provides a relatively tight, convex upper bound on the 0–1 indicator function. Specifically, the hinge loss equals the 0–1 indicator function the empirical risk minimization of this loss is equivalent to the classical formulation for support vector machines (SVMs). Correctly classified points lying outside the margin boundaries of the support vectors are not penalized, whereas points within the margin boundaries or on the wrong side of the hyperplane are penalized in a linear fashion compared to their distance from the correct boundary.[11]

In our final attempt we augmented the base model with auxiliary loss and learning rate decay method which reduces the learning rate for each iteration thus shrinking the learning rate to compensate for the stronger curvature. We perform another experiment changing up the Learning Rate Decay strategy, by now applying Learning Rate Decay at the end of every epoch instead of every iteration, thus allowing the model to adapt slowly compared to the previous approach and the results reflect this change in implementation. We use ResNet architectures for the discriminators and generator.

6 Results

We demonstrate empirically that (1) self-supervision improves the representation quality with respect to baseline GAN models, and that (2) it leads to improved unconditional generation for complex datasets.

We focus primarily on MNIST, database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. We resize the images to $32 * 32 * 1$. We provide additional comparison on to CIFAR dataset, for which unconditional GANs can be successfully trained. CIFAR10 contains 60k images $32 * 32 * 1$, partitioned into 50k training instances and 10k test instances.

Dataset	Method	FID
CIFAR	SS-LD-EI	301.415
	SS-LD-EE	226.76
MNIST	SS-LD-EI	284.6
	SS-LD-EE	215.51

Table 1: Best FID attained across three random seeds. In this setting the proposed approach recovers most of the benefits of conditioning. Where SS-LD-EI = Self-Supervised-Learning rate Decay- for Every Iteration, SS-LD-EE = Self-Supervised-Learning rate Decay- for Every Epoch.

Comparision We observe the FID scores are using our Learning Rate Decay in conjunction with Self Supervised learning. The models using Learning Rate Decay for every epoch perform better than the models that used Learning Rate Decay for every iteration.

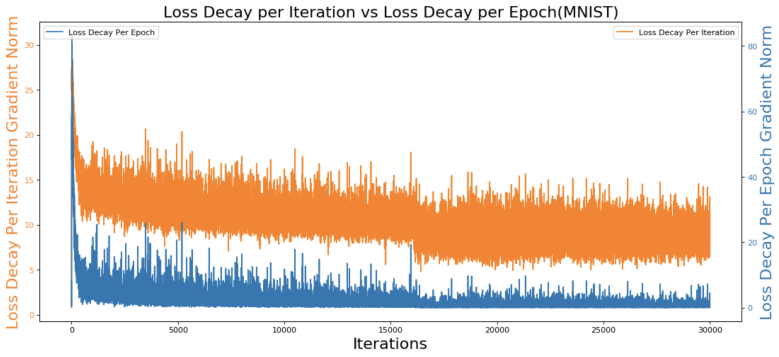

Figure 2: Convergence of the model every iteration on MNIST dataset. The orange plot indicates the model implementing Learning Rate Decay for every iteration, and the blue plot indicates the model implementing Learning Rate Decay for every epoch. This example demonstrates that a GAN that implements that Learning Rate Decay for every epoch performs better than a GAN that implements Learning Rate Decay for every iteration.

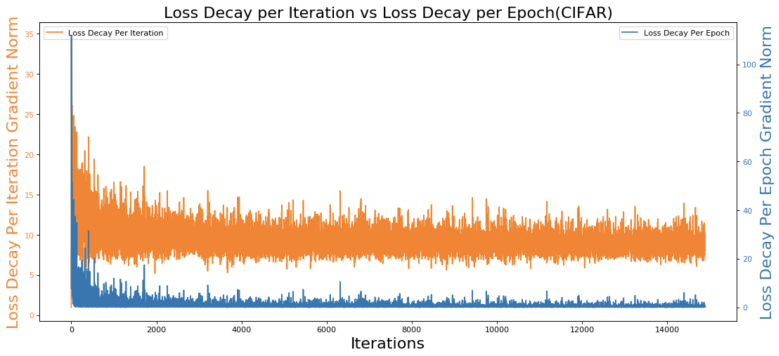

Figure 3: Convergence of the model every iteration on CIFAR dataset. The orange plot indicates the model implementing Learning Rate Decay for every iteration, and the blue plot indicates the model implementing Learning Rate Decay for every epoch. This example demonstrates that a GAN that implements that Learning Rate Decay for every epoch performs better than a GAN that implements Learning Rate Decay for every iteration.

We have plotted the the convergence of the model for both MNIST and CIFAR. In Figure 2 to observe the distinction between the plots for Learning Rate Decay for every epoch and Learning Rate Decay for every iteration we have scaled the figure. We observe that the model which implements Learning Rate Decay for every epoch performs much better than Learning Rate Decay for every iteration in the case of MNIST and the case of CIFAR as well from Figure 3, but the convergence is not that strong when compared to MNIST.

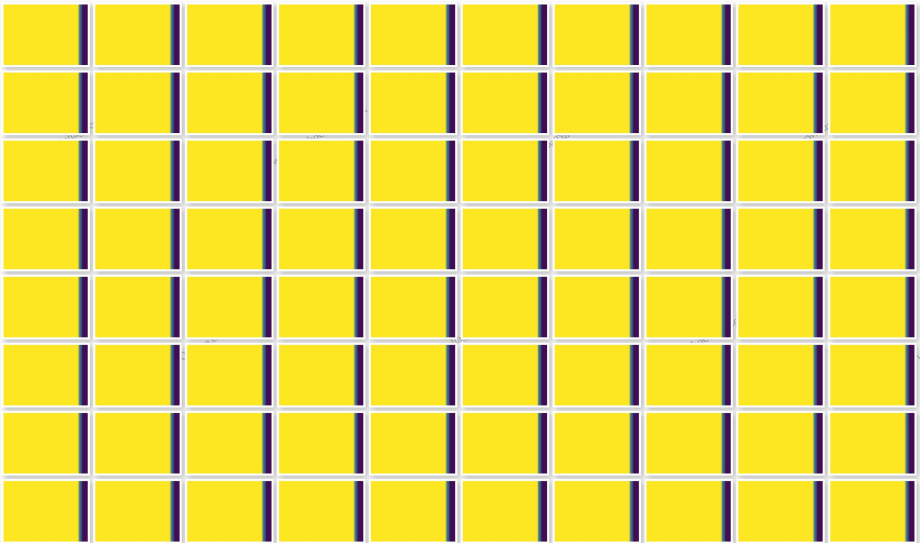

Figure 4: A sample of images from the self-supervised model with Wasserstein Loss, Weight Normalization, without Learning Rate Decay on our GAN trained on MNIST dataset images.

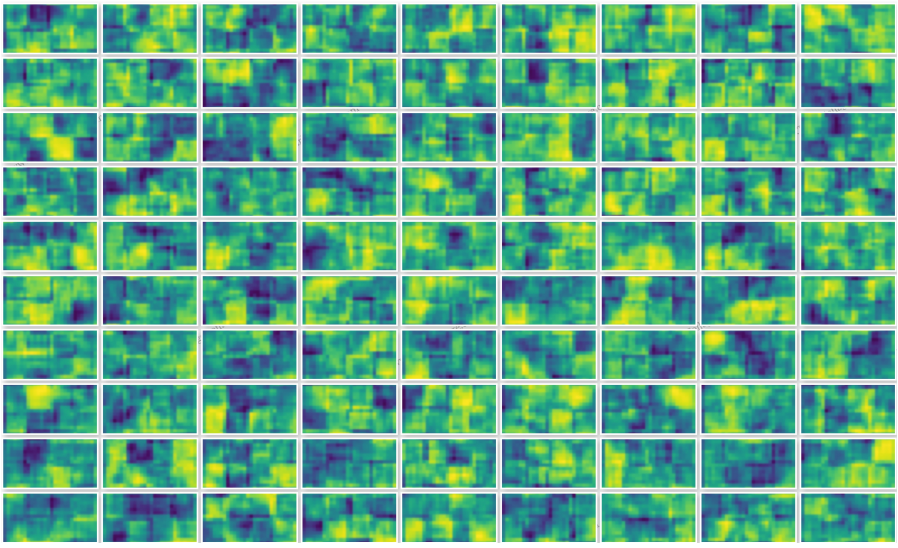

Figure 5: A sample of images from the self-supervised model with Hinge Loss, without Learning Rate Decay on our GAN trained on MNIST dataset images.

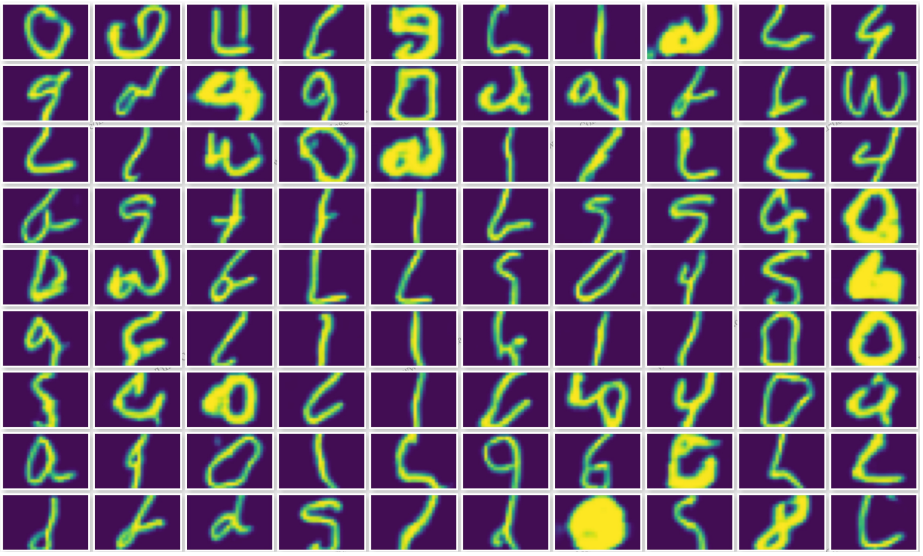

Figure 6: A sample of images from the self-supervised model with Spectral Normalization, Learning Rate Decay on GAN trained on MNIST dataset images.

Conclusion

We observe that the images generated by the GAN implementing Spectral Normalization, Learning Rate Decay are significantly better quality than the images generated by GAN with Weight Norm or GAN with Hinge Loss. We use ResNet architectures for the generator and discriminator as in Miyato et al. [12]. We emphasize that our discriminator architecture is optimized for image generation, not representation quality.

References

[1] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014) *Generative Adversarial Networks* , pp. 2672–2680. Montreal: NIPS 2014.

[2] LeCun, Yann. *RL Seminar: The Next Frontier in AI: Unsupervised Learning*.

[3] Andrej Karpathy; Pieter Abbeel; Greg Brockman; Peter Chen; Vicki Cheung; Rocky Duan; Ian Goodfellow; Durk Kingma; Jonathan Ho; Rein Houthooft; Tim Salimans; John Schulman; Ilya Sutskever; Wojciech Zaremba(2016), *Generative Models*, OpenAI.

[4] Augustus Odena; Christopher Olah; Jonathon Shlens , (2017) *Conditional Image Synthesis with Auxiliary Classifier GANs*, Sydney: ICML'17.

[5] Kevin J Liang; Chunyuan Li; Guoyin Wang; Lawrence Carin;(2019) *Generative Adversarial Network Training is a Continual Learning Problem* ICLR.

[6] Ting Chen; Xiaohua Zhai; Neil Houlsby(2018) *Self-Supervised GAN to Counter Forgetting* , NeurIPS.

[7] James Kirkpatricka; Razvan Pascanua; Neil Rabinowitza; Joel Venessa; Guillaume Desjardinsa; Andrei A. Rusua; Kieran Milana; John Quana; Tiago Ramalhoa; Agnieszka Grabska-Barwinska a; Demis Hassabisa; Claudia Clopathb; Dharshan Kumarana; and Raia Hadsella; (2016)*Overcoming catastrophic forgetting in neural networks* arXiv.

[8] Spyros Gidaris; Praveer Singh; Nikos Komodakis;(2018). *Unsupervised Representation Learning by Predicting Image Rotations* ICLR.

[9] Ting Chen; Xiaohua Zhai; Marvin Ritter; Mario Lucic; Neil Houlsby,(2019) *Self-Supervised GANs via Auxiliary Rotation Loss* CVPR.

[10] Carl Doersch; Abhinav Gupta; Alexei A. Efros;(2015) *Unsupervised Visual Representation Learning by Context Prediction* ICCV.

[11] Piyush Rai;(2011) *Support Vector Machines (Contd.), Classification Loss Functions and Regularizers* CS5350.

[12] Takeru Miyato; Toshiki Kataoka; Masanori Koyama; Yuichi Yoshida;(2018) *Spectral Normalization for Generative Adversarial Networks* ICLR.

[13] Ting Chen; Xiaohua Zhai; Marvin Ritter; Mario Lucic; Neil Houlsby,(2019) *Self-Supervised GANs via Auxiliary Rotation Loss* https://paperswithcode.com/paper/self-supervised-generative-adversarial