Project Report
SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS

Yao Zhang
Department of Computer Science
Binghamton University
Bnumber: B00961692
Email: yzhan480@binghamton.edu

***Abstract*—There is a persisting challenge of generative adversarial networks which is the instability of its training. And the reported paper proposes a novel weight normalization technique called spectral normalization to stabilize the training of GAN's discriminator. The paper claims that this normalization technique is computationally light and easy to incorporate into existing implementations. And the authors tested the efficacy of spectral normalization on CIFAR10, STL-10, and ILSVRC2012 datasets. The experiment results show that spectrally normalized GANs (SN-GANs) is capable of generating images of better or equal quality relative to the previous training stabilization techniques. And I applied this technique on variant of standard DCGAN for CIFAR-10 dataset comparing the effect to standard DCGAN using batch normalization. This is the github source code repository link of my project: https://github.com/yzhan480/CS536FinalProject .**

I. Introduction

Generative adversarial networks (GANs) (Goodfellow et al., 2014) [1] have achieved great success as a framework of generative models in recent years and it has been applied to many types of tasks and datasets (Radford et al., 2016; Salimans et al., 2016; Li et al., 2017) [2] [3] [4]. GANs are a generative framework which study a collection of training examples and learn the probability distribution that generated them. GANs are then able to generate more examples from the estimated probability distribution. And the framework consists of two models: a generator G that captures the data distribution, and a discriminator D that distinguishes that a sample is from training data or generator G. And it estimates data distribution through an adversarial process in which we simultaneously train two models D and G. And the goal is to reduce the difference between the model G distribution and the target distribution measured by the best discriminator possible at each step of the training. GANs have been drawing attention in the machine learning community not only for its ability to learn real world data probability distribution but also for its theoretical aspects. (Nowozin et al., 2016; Uehara et al., 2016; Mohamed & Laksh minarayanan, 2017) [5] [6] [7] revealed that the training of the discriminator amounts to the training of a good estimator for the density ratio between the model distribution and the target.

However, the instability of training GANs is a challenge which is affected by the performance control of the discriminator. The density ratio estimation by discriminator is often inaccurate and unstable during training so generator fails to learn the target distribution. For example, when discriminator can perfectly distinguish the model distribution from the target, the training of the generator comes to complete stop because the derivative of the so-produced discriminator with respect to the input turns out to be 0. So the reported paper proposed a technique to restrict the discriminator.

The authors propose a novel weight normalization method called *spectral normalization* that can stabilize the training of discriminator networks. And it has following favorable properties.

- Lipschitz constant is the only hyper-parameter to be tuned, and the algorithm does not require intensive tuning of the only hyper-parameter for satisfactory performance.
- Implementation is simple and the additional computational cost is small.
- This method also functioned well even without tuning Lipschitz constant.

From paper, this normalization method also functioned well even without tuning Lipschitz constant, which is the only hyper parameter.

II. Method

In this section, I will paraphrase the core theoretical groundwork for this method.

The machine learning community has been pointing out recently that the function space from which the discriminators are selected crucially affects the performance of GANs. A number of works (Uehara et al., 2016; Qi, 2017; Gulrajani et al., 2017) [6] [8] [9] advocate the importance of Lipschitz continuity in assuring the boundedness of statistics. For example, the optimal discriminator of GANs on the above standard formulation takes the form.

$$D_G^*(\mathbf{x}) = \frac{q_{data}(\mathbf{x})}{q_{data}(\mathbf{x}) + p_G(\mathbf{x})} = sigmoid\left(f^*(\mathbf{x})\right)$$

where

$$f^*(\mathbf{x}) = \log q_{data}(\mathbf{x}) - \log p_G(\mathbf{x}) \tag{1}$$

and its derivative

$$\nabla_{\mathbf{x}} f^*(x) = \frac{1}{q_{\text{data}}(\mathbf{x})}\nabla_{\mathbf{x}} q_{\text{data}}(\mathbf{x}) - \frac{1}{p_G(\mathbf{x})}\nabla_{\mathbf{x}} p_G(\mathbf{x}) \tag{2}$$

can be unbounded or even incomputable. This prompts authors to introduce some regularity condition to the derivative of $f(x)$.

A particularly successful works in this array are (Qi, 2017; Arjovsky et al., 2017; Gulrajani et al., 2017) [8] [10] [9], which proposed methods to control the Lipschitz constant of the discriminator by adding regularization terms defined on input examples x. The authors follow their footsteps and search for the discriminator D from the set of K-Lipschitz continuous functions.

$$\underset{\|f\|_{\text{Lip}} \le K}{\arg\max} V(G, D), \tag{3}$$

A. SPECTRAL NORMALIZATION

Spectral normalization controls the Lipschitz constant of the discriminator function f by literally constraining the spectral norm of each layer $g : h_{in} \mapsto h_{out}$. By definition, Lipschitz norm $\|g\|_{Lip}$ is equal to $sup_h \sigma(\nabla g(h))$, where $\sigma(A)$ is the spectral norm of the matrix A (L_2 matrix norm of A)

$$\sigma(A) := \max_{\mathbf{h}:\mathbf{h}\neq\mathbf{0}} \frac{\|A\mathbf{h}\|_2}{\|\mathbf{h}\|_2} = \max_{\|\mathbf{h}\|_2 \le 1} \|A\mathbf{h}\|_2 \tag{4}$$

which is equivalent to the largest singular value of A. Therefore, for a linear layer $g(\mathbf{h}) = W\mathbf{h}$, the norm is given by $\|g\|_{Lip} = \sup_{\mathbf{h}} \sigma(\nabla g(\mathbf{h})) = \sup_{\mathbf{h}} \sigma(W) = \sigma(W)$. If the Lipschitz norm of the activation function $\|a_l\|_{Lip}$ is equal to 1, we can use the inequality $\|g_1 \circ g_2\|_{Lip} \le \|g_1\|_{Lip} \cdot \|g_2\|_{Lip}$ to observe the following bound on $\|f\|_{Lip}$:

$$\begin{gathered} \|f\|_{Lip} \le \\ \left\|\left(\mathbf{h}_L \mapsto W^{L+1}\mathbf{h}_L\right)\right\|_{Lip} \cdot \|a_L\|_{Lip} \cdot \left\|\left(\mathbf{h}_{L-1} \mapsto W^L \mathbf{h}_{L-1}\right)\right\|_{Lip} \\ \cdots \|a_1\|_{Lip} \cdot \left\|\left(\mathbf{h}_0 \mapsto W^1\mathbf{h}_0\right)\right\|_{Lip} \\ = \prod_{l=1}^{L+1} \left\|\left(\mathbf{h}_{l-1} \mapsto W^l \mathbf{h}_{l-1}\right)\right\|_{Lip} = \prod_{l=1}^{L+1} \sigma\left(W^l\right) \end{gathered} \tag{5}$$

spectral normalization normalizes the spectral norm of the weight matrix W so that it satisfies the Lipschitz constraint $\sigma(W) = 1$.

$$\bar{W}_{\text{SN}}(W) := W/\sigma(W) \tag{6}$$

If we normalize each W_l using (6), we can appeal to the inequality (5) and the fact that $\sigma\left(\bar{W}_{\text{SN}}(W)\right) = 1$ to see that $\|f\|_{Lip}$ is bounded from above by 1.

B. FAST APPROXIMATION OF THE SPECTRAL NORM $\sigma(W)$

The spectral norm $\sigma(W)$ that we use to regularize each layer's weight matrix W of the discriminator is the largest singular value of W. However, applying singular value decomposition to compute the $\sigma(W)$ at each round of the algorithm costs huge computational resources. Instead, the authors use the power iteration method to estimate $\sigma(W)$ (Golub & Van der Vorst, 2000; Yoshida & Miyato, 2017) [11] [12]. With power iteration method, the nets can estimate the spectral norm with very small additional computational resources relative to the full computational costs of the vanilla GANs. Algorithm 1 is the summary of the actual spectral normalization algorithm.

Algorithm 1 Algorithm 1 SGD with spectral normalization

```
Initialize $\tilde{\boldsymbol{u}}_l \in \mathcal{R}^{d_l}$ for $l = 1, \ldots, L$ with a random vector (sampled from isotropic distribution).
For each update and each layer $l$ :
1. Apply power iteration method to a unnormalized weight $W^l$ :
    $\tilde{\boldsymbol{v}}_l \leftarrow \left(W^l\right)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l / \left\|\left(W^l\right)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l\right\|_2$
    $\tilde{\boldsymbol{u}}_l \leftarrow W^l \tilde{\boldsymbol{v}}_l / \left\|W^l \tilde{\boldsymbol{v}}_l\right\|_2$
2. Calculate $\bar{W}_{\text{SN}}$ with the spectral norm:
    $\bar{W}_{\text{SN}}^l\left(W^l\right) = W^l / \sigma\left(W^l\right)$ , where $\sigma\left(W^l\right) = \tilde{\boldsymbol{u}}_l^{\mathrm{T}} W^l \tilde{\boldsymbol{v}}_l$
3. Update $W^l$ with $SGD$ on mini-batch dataset $\mathcal{D}_M$ with a learning rate $\alpha$ :
    $W^l \leftarrow W^l - \alpha \nabla_{W^l} \ell\left(\bar{W}_{\text{SN}}^l\left(W^l\right), \mathcal{D}_M\right)$
```

III. My Contribution

The original project code is written in tensorflow framework. And I refered to other github repositories to implement my project using pytorch. Here is the github source code repository link of my project: https://github.com/yzhan480/CS536FinalProject . I use pytorch to implement SN-GAN (abbreviation of the spectrally normalized GANs) for CIFAR-10 dataset (Torralba et al., 2008) [13]. Generator and discriminator are from DCGAN (Radford et al., 2016) [14] which are different from the architecture used in the original paper. And I compare the performance of applying spectral normalization on variant of DCGAN to the performance of well-designed DCGAN which use batch normalization. Performance is measured by FID score (Heusel et al., 2017) [16]. After that, I delete all batch norm layers in standard DCGAN to see the effect of spectral normalization and batch normalization. My code runs on Colab powered by Google. I think my work can prove the generalization and effectiveness of spectral normalization. In next part, I will discuss the details of my experiment.

IV. Experiment

In order to evaluate the efficacy and generalization of this approach, I designed 3 models to conduct a set of experiments of unsupervised image generation on CIFAR-10 dataset and compared this method against other normalization technique batch normalization. First, I will discuss the objective functions I used to train the architecture, and then I will describe the optimization settings I used in the experiments. I will then present performance measure on the images to evaluate the images produced by the trained generators.

As for the architecture of the discriminator and generator, I used standard DCGAN which is different from the original paper. I used the following standard objective function for the adversarial loss:

$$V(G,D) := \mathop{\mathrm{E}}_{x\sim q_{\mathrm{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathop{\mathrm{E}}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \quad (7)$$

For optimization, I used the Adam optimizer (Kingma & Ba., 2015) [15] in all of my experiments. Learning rate for generator is 0.0001 and for discriminator is 0.0004. Momentum parameters $(beta_1, beta_2)$ of Adam are 0.5 and 0.999. For quantitative assessment of generated examples, I used Frechet inception distance (FID score).

A. Results

In this section, I put my experiment results here. I report the FID score of my three models. I report generator loss and discriminator loss of my first two models, and the loss of third model make the loss function overflow. And I present the images generated by my three models.

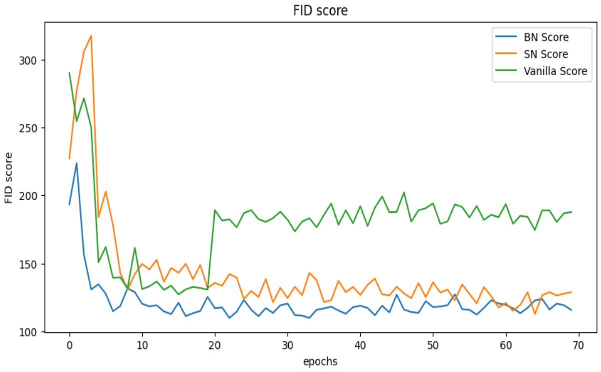

Figure1: FID scores of three models on CIFAR-10(lower is better).

In Figure 1 I show the FID score of three models varies from training epochs. I see that SN-GAN performs nearly as good as well-designed DCGAN. However, if DCGAN deletes all its batch norm layers, the performance is quite poor which can prove DCGAN does not have a very robust architecture of discriminator.

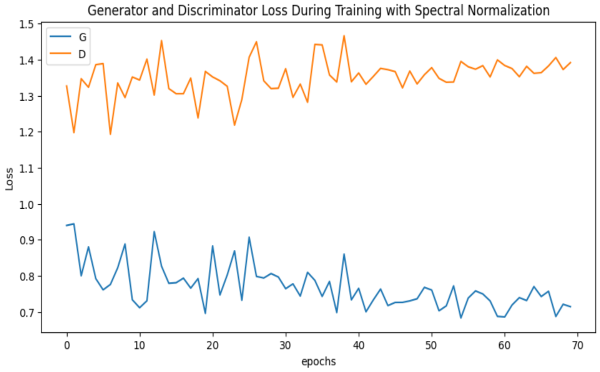

Figure2: Loss of generator and discriminator of SN-GAN

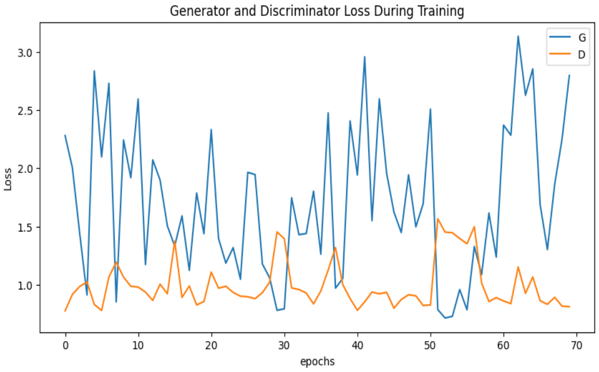

Figure3: Loss of generator and discriminator of DCGAN

In Figure 2 and Figure 3 I show the loss of generator and discriminator of SN-GAN and DCGAN respectively. The loss of discriminator of SN-GAN is a bit larger than loss of DCGAN which represents that the spectral normalization regularizes the performance of discriminator during training. Thus, during training process SN-GAN makes its generator to learn enough knowledge and then generates high-quality images.

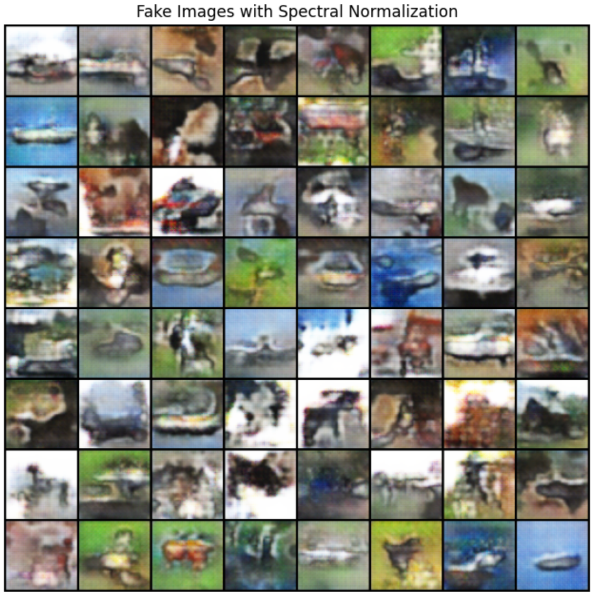

Figure 4: Images generated by SN-GAN

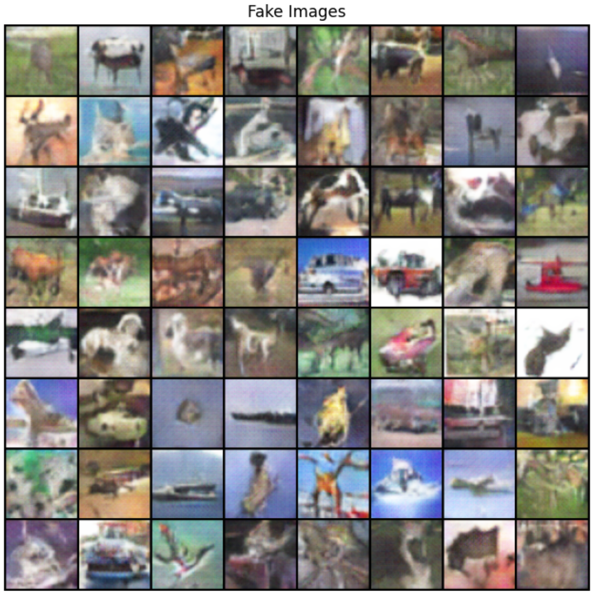

Figure 5: Images generated by standard DCGAN

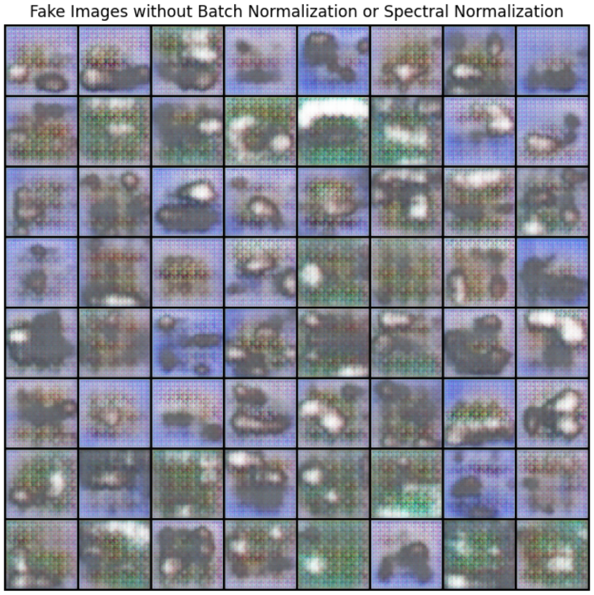
Fake Images without Batch Normalization or Spectral Normalization

Figure 6: Images generated by DCGAN without batch normalization

In Figure 4, Figure 5, Figure 6 I show the generated images of 3 models. And the FID score is lower, the model generates better images. After 70 epochs, FID score of standard DCGAN is 115.7578, FID score of SN-GAN is 128.8853, and FID score of DCGAN without batch normalization is 187.7866. We can tell the differences of FID score from the generated images. The third model did not generate recognizable images which represents that the model did not work on CIFAR-10 dataset. The robustness of the standard DCGAN is not good enough.

The computational time of SN-GAN is almost the same as vanilla DCGAN which costs 1.5 hours running on Colab.

V. Discussion

This paper proposes spectral normalization as a stabilizer of training of GANs. When applying spectral normalization to DCGAN on image generation tasks, the generated samples achieve FID score as good as that of images generated by standard DCGAN. In the future work, the author would like to further investigate where their methods stand amongst other methods on more theoretical basis, and experiment their algorithm on larger and more complex datasets.

Acknowledgment

Thanks to professor Adnan Siraj Rakin. Because this project gives me very valuable experiences on researching and makes me feel comfortable to do more research in future days.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

[2] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[3] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NIPS*, pp. 2226–2234, 2016.

[4] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *EMNLP*, pp. 2147–2159, 2017.

[5] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *NIPS*, pp. 271–279, 2016.

[6] Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Generative adversarial nets from a density ratio estimation perspective. *NIPS Workshop on Adversarial Training*, 2016.

[7] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *NIPS Workshop on Adversarial Training*, 2017.

[8] Guo-Jun Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

[10] Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223, 2017.

[11] Gene H Golub and Henk A Van der Vorst. Eigenvalue computation in the 20th century. *ournal of Computational and Applied Mathematics*, 123(1):35–65, 2000.

[12] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv*:1705.10941, 2017.

[13] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint*:1511.06434, 2016.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Gunter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv*:1706.08500, 2017.