Spectral Normalization on GANs

Bingbing Hu

July 17, 2019

SIST, ShanghaiTech

Outline

1. Motivation
2. Method
3. Experiments
4. Conclusion

Motivation

Motivation

GAN training is unstable!

Motivation

GAN training is unstable!

- Discriminator can be curcial to the GAN training
- There exists a discriminator that can perfectly distinguish the model distribution from target
- Such discriminator can cause zero gradient, hence training collapse

Motivation

GAN training is unstable!

- Discriminator can be curcial to the GAN training
- There exists a discriminator that can perfectly distinguish the model distribution from target
- Such discriminator can cause zero gradient, hence training collapse

All of these motivate us to bring some constraints to discriminator.

Method

GAN Structure

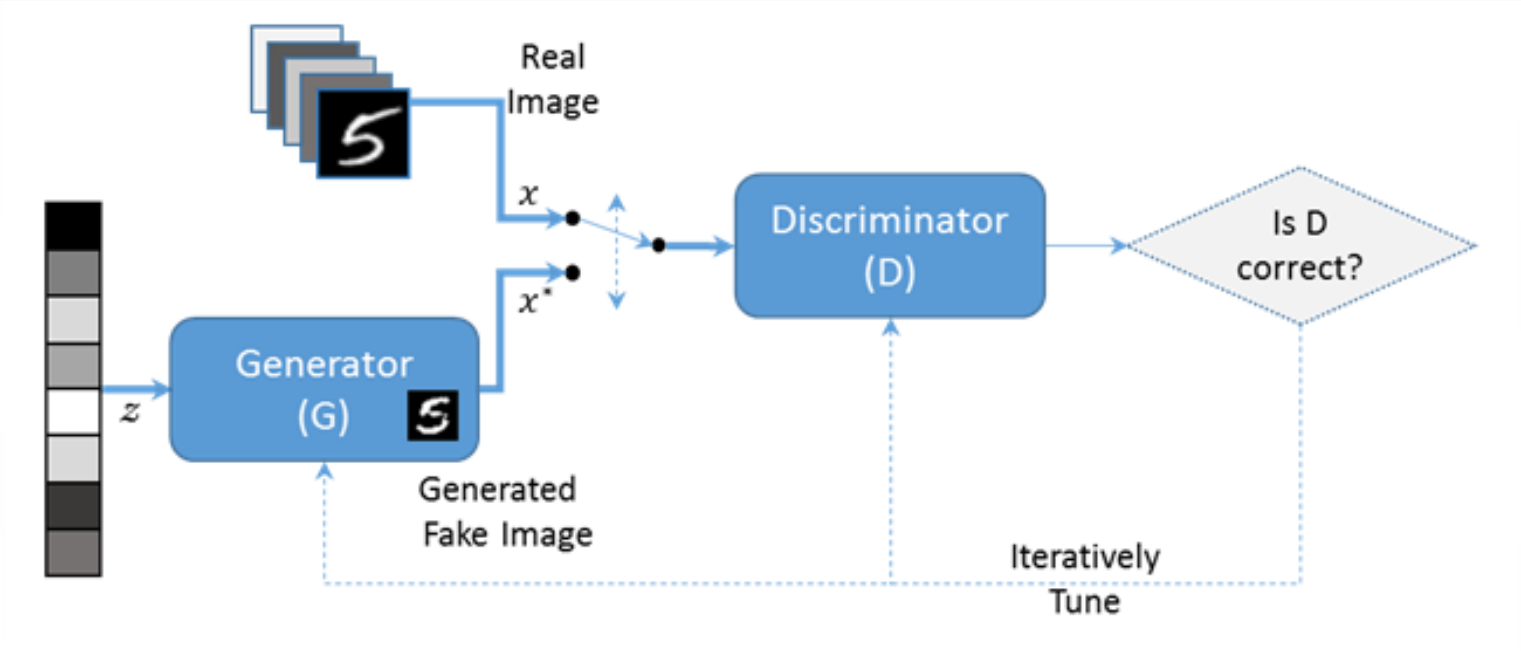


Figure 1: GAN Structure

GAN Structure

The standard formulation of GAN is given by

$$\min_G \max_D V(G, D)$$

The conventional form of $V(G, D)$ is given by

$$\mathbb{E}_{\boldsymbol{x} \sim q_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\hat{\boldsymbol{x}} \sim p_{\text{G}}}[\log(1 - D(G(\hat{\boldsymbol{x}})))] \tag{1}$$

Why Unstable?

For the conventional form of $V(G, D)$ (1) and a fixed generator G, it is known that [1], the optimal discriminator is given by

$$D_G^\star(\boldsymbol{x}) = \frac{q_{\text{data}}(\boldsymbol{x})}{q_{\text{data}}(\boldsymbol{x}) + p_G(\boldsymbol{x})}.$$

This gives

$$\begin{aligned} D_G^\star(\boldsymbol{x}) &= \sigma(f^*(\boldsymbol{x})), \\ \text{where } f^*(\boldsymbol{x}) &= \log q_{\text{data}}(\boldsymbol{x}) - \log p_G(\boldsymbol{x}) \\ \implies \nabla_{\boldsymbol{x}} f^*(\boldsymbol{x}) &= \frac{1}{q_{\text{data}}(\boldsymbol{x})} \nabla_{\boldsymbol{x}} q_{\text{data}}(\boldsymbol{x}) - \frac{1}{p_G(\boldsymbol{x})} \nabla_{\boldsymbol{x}} p_G(\boldsymbol{x}). \end{aligned}$$

The derivate of $f^*(\boldsymbol{x})$ can be unbounded or even incomputable.

Method

Formulate the discriminator as follows:

$$f(\boldsymbol{x}, \theta) = W^{L+1} a_L(W^L(a_{L-1}(W^{L-1}(\dots a_1(W^1 \boldsymbol{x}) \dots))))$$

- $\boldsymbol{x}$: input of the network
- $\theta := \{W^1, \dots, W^L, W^{L+1}\}$: learning parameters set
- a_l: element-wise non-linear function

Method

Formulate the discriminator as follows:

$$f(\boldsymbol{x}, \theta) = W^{L+1} a_L(W^L(a_{L-1}(W^{L-1}(\dots a_1(W^1 \boldsymbol{x}) \dots))))$$

- $\boldsymbol{x}$: input of the network
- $\theta := \{W^1, \dots, W^L, W^{L+1}\}$: learning parameters set
- a_l: element-wise non-linear function

The final output of the discriminator is given by

$$D(\boldsymbol{x}, \theta) = \mathcal{A}(f(\boldsymbol{x}, \theta))$$

Spectral Normalization

The spectral normalization controls the Lipschitz constant of the discriminator function f by iterativelly constraining the spectral norm of each layer $g : \boldsymbol{h}_{in} \mapsto \boldsymbol{h}_{out}$.

Definition 1

The Lipschitz norm $\|f\|_{\text{Lip}}$ of funtion f is smallest value M such that

$$\frac{\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\|_2}{\|\boldsymbol{x} - \boldsymbol{x}'\|_2} \leq M$$

for any $\boldsymbol{x} \neq \boldsymbol{x}'$.

Spectral Normalization

Definition 2

The spectral norm of a matrix $\mathbf{A}$ is defined by

$$\sigma(\mathbf{A}) := \sup_{\boldsymbol{h}:\boldsymbol{h}\neq\mathbf{0}} \frac{\|\mathbf{A}\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \sup_{\|\boldsymbol{h}\|_2\leq 1} \|\mathbf{A}\boldsymbol{h}\|_2$$

- spectral norm is actually matrix 2-norm
- spectral norm is the max singular value of $\mathbf{A}$

Spectral Normalization

Therefore, for a linear layer $g(\boldsymbol{h}) = W\boldsymbol{h}$,

$$\|g\|_{\mathrm{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h})) = \sup_{\boldsymbol{h}} \sigma(W) = \sigma(W)$$

If $\|a_l\|_{\mathrm{Lip}} = 1$ [1], we can use the inequality

$$\|g_1 \circ g_2\|_{\mathrm{Lip}} \leq \|g_1\|_{\mathrm{Lip}} \cdot \|g_2\|_{\mathrm{Lip}}$$

to get

$$\begin{aligned}
\|f\|_{\mathrm{Lip}} \leq & \|(\boldsymbol{h}_L \mapsto W^{L+1}\boldsymbol{h}_L)\|_{\mathrm{Lip}} \cdot \|a_L\|_{\mathrm{Lip}} \cdot \\
& \|(\boldsymbol{h}_{L-1} \mapsto W^{L}\boldsymbol{h}_{L-1})\|_{\mathrm{Lip}} \cdots \|a_1\|_{\mathrm{Lip}} \cdot \|(\boldsymbol{h}_0 \mapsto W^{1}\boldsymbol{h}_0)\|_{\mathrm{Lip}} \\
= & \prod_{l=1}^{L+1} \|(\boldsymbol{h}_{l-1} \mapsto W^{l}\boldsymbol{h}_{l-1})\|_{\mathrm{Lip}} = \prod_{l=1}^{L+1} \sigma(W^l).
\end{aligned}$$

[1] *e.g.*, ReLU and leaky ReLU

Spectral Normalization

Normalize the spectral norm of the weight matrix W such that $\sigma(W) = 1$:

$$\bar{W}_{\mathrm{SN}} := W/\sigma(W). \tag{2}$$

If we normalize each W^l using (2), together with the fact $\sigma(\bar{W}_{\mathrm{SN}}) = 1$, we see that $\|f\|_{\mathrm{Lip}}$ is bounded above by 1.

Power Iteration

How to compute the max singular value $\sigma(W)$?

- SVD? – computation inefficient
- Power iteration – ok

Power Iteration

Brief view of power method. Suppose $A \in \mathbf{R}^{n\times n}$ is diagonalizable, with eigenvalues ordered that

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|.$$

Then we have $A = V\Lambda V^{-1}$ where $\Lambda = \mathbf{diag}(\lambda_1, \cdots, \lambda_n)$, now

$$A^k = V\Lambda^k V^{-1}$$

$$\Longrightarrow \; A^k V = V\Lambda^k.$$

Randomly chose $\boldsymbol{x} = V\tilde{\boldsymbol{x}} \in \mathbf{R}^n$ and consider

$$A^k\boldsymbol{x} = A^k V\tilde{\boldsymbol{x}} = V\Lambda^k\tilde{\boldsymbol{x}} = \sum_{j=1}^{n} \boldsymbol{v}_j \lambda_j^k \tilde{x}_j$$

$$= \lambda_1^k \sum_{j=1}^{n} \left(\frac{\lambda_j}{\lambda_1}\right)^k \tilde{x}_j \boldsymbol{v}_j \to \lambda_1^k \tilde{x}_1 \boldsymbol{v}_1.$$

Power Iteration

If $|\lambda_1| > |\lambda_2|$, $A^k \boldsymbol{x} \to c\boldsymbol{v}_1$ for sufficient large k. This is the idea behind power iteration:

$$\boldsymbol{x}^{(k+1)} = \frac{A\boldsymbol{x}^{(k)}}{\|A\boldsymbol{x}^{(k)}\|} = \frac{A^k\boldsymbol{x}^{(0)}}{\|A^k\boldsymbol{x}^{(0)}\|}.$$

As $\boldsymbol{v}_1$ is the eigenvector corresponding to the max eigenvalue, noting that V is orthonormal, we have $\boldsymbol{x}^{(k)} \to \boldsymbol{v}_1$. So the max eigen value of A is given by

$$\lambda_1 = \boldsymbol{v}_1^T A \boldsymbol{v}_1 = \left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{x}^{(k+1)}.$$

Gradient Analysis

Recall that

$$\bar{W}_{\mathrm{SN}} := W/\sigma(W).$$

The gradient of the normalized weight matrix w.r.t. W_{ij} is

$$\frac{\partial \bar{W}_{\mathrm{SN}}}{\partial W_{ij}} = \frac{E_{ij}}{\sigma(W)} - \frac{W}{\sigma^2(W)} \frac{\partial \sigma(W)}{\partial W_{ij}} \tag{3}$$

$$= \frac{E_{ij}}{\sigma(W)} - \frac{[\boldsymbol{u}_1 \boldsymbol{v}_1^T]_{ij}}{\sigma^2(W)} W \tag{4}$$

$$= \frac{1}{\sigma(W)} \left(E_{ij} - [\boldsymbol{u}_1 \boldsymbol{v}_1^T]_{ij} \bar{W}_{\mathrm{SN}} \right), \tag{5}$$

where E_{ij} is the matrix whose (i, j)-th entry is 1 and zero otherwise, and $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are respectively the first left and right singular vectors of W.

Gradient Analysis

If $\boldsymbol{h}$ is the hidden layer to be transformed by $\bar{W}_{\mathrm{SN}}$, the derivative of the $V(G, D)$ calculated over the mini-batch w.r.t. W of the discriminator D is given by

$$\frac{\partial V(G,D)}{\partial W} = \frac{1}{\sigma(W)}(\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T] - (\hat{E}[\boldsymbol{\delta}^T \bar{W}_{\mathrm{SN}}\boldsymbol{h}])\boldsymbol{u}_1\boldsymbol{v}_1^T) \tag{6}$$

$$= \frac{1}{\sigma(W)}(\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T] - \lambda\boldsymbol{u}_1\boldsymbol{v}_1^T) \tag{7}$$

- $\boldsymbol{\delta} := (\partial V(G,D)/\partial(\bar{W}_{\mathrm{SN}}\boldsymbol{h}))^T$
- $\lambda := \hat{E}[\boldsymbol{\delta}^T(\bar{W}_{\mathrm{SN}}\boldsymbol{h})]$
- $\hat{E}[\cdot]$: empirical expectation over the mini-batch

Gradient Analysis

$$\frac{\partial V(G,D)}{\partial W} = \frac{1}{\sigma(W)}(\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T] - (\hat{E}[\boldsymbol{\delta}^T \bar{W}_{\mathrm{SN}}\boldsymbol{h}])\boldsymbol{u}_1\boldsymbol{v}_1^T) \tag{6}$$

$$= \frac{1}{\sigma(W)}(\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T] - \lambda\boldsymbol{u}_1\boldsymbol{v}_1^T) \tag{7}$$

- The first term $\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T]$ of (7) is s ame as the derivative w/o spectral normalization
- The second term can be seen as a regularizer with adaptive coefficient λ
- $\lambda > 0$ if $\boldsymbol{\delta}$ and $\bar{W}_{\mathrm{SN}}\boldsymbol{h}$ point to same direction, thus gives a penalty

Gradient Analysis

$$\frac{\partial V(G,D)}{\partial W} = \frac{1}{\sigma(W)}(\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T] - (\hat{E}[\boldsymbol{\delta}^T \bar{W}_{\mathrm{SN}}\boldsymbol{h}])\boldsymbol{u}_1\boldsymbol{v}_1^T) \tag{6}$$

$$= \frac{1}{\sigma(W)}(\hat{E}[\boldsymbol{\delta}\boldsymbol{h}^T] - \lambda\boldsymbol{u}_1\boldsymbol{v}_1^T) \tag{7}$$

Spectral normalization prevents the transformation of each layer from being too sensitive in one direction.

Experiments

Metrics

Assessment metrics:

- Inception Score (IS) [2]: KL divergence between conditional and marginal class distribution, higher is better
- Fréchet Inception Distance (FID) [3]: Wassertein-2 distance between generated and real images, lower is better

Experiments

The spectral norm of each layer in the discriminator dosen't change much in the course of training, they tends to be stablized.

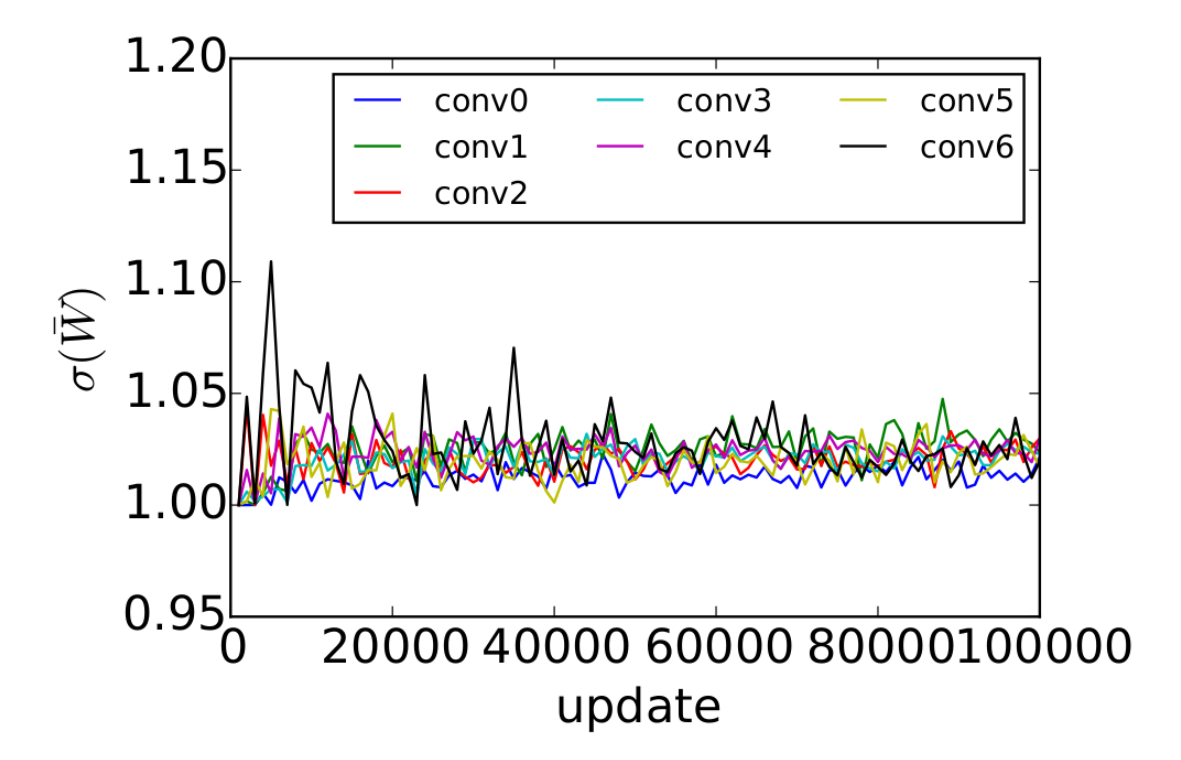
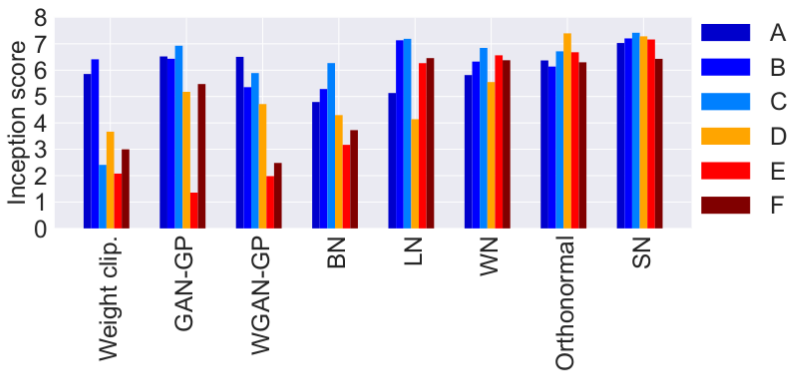


Figure 2: Spectral norm changes during training

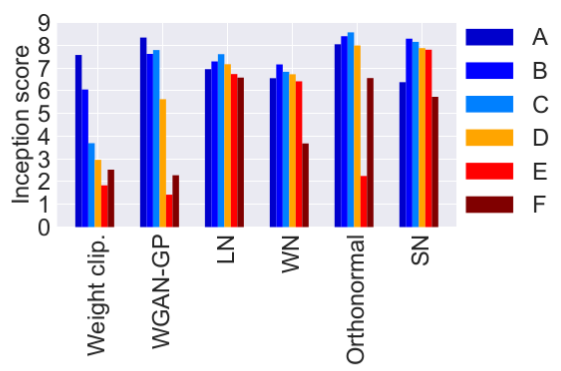
Experiments

Spectral normalization is more robust than others w.r.t. different settings of hyperparameters.

Setting	α	β_1	β_2	n_{dis}
A†	0.0001	0.5	0.9	5
B‡	0.0001	0.5	0.999	1
C*	0.0002	0.5	0.999	1
D	0.001	0.5	0.9	5
E	0.001	0.5	0.999	5
F	0.001	0.9	0.999	5



(a) CIFAR-10



(b) STL-10

Figure 3: Inception scores on CIFAR-10 and STL-10

Spectral normalized GAN (SN-GAN) generates better images. (Hope you can see it.)

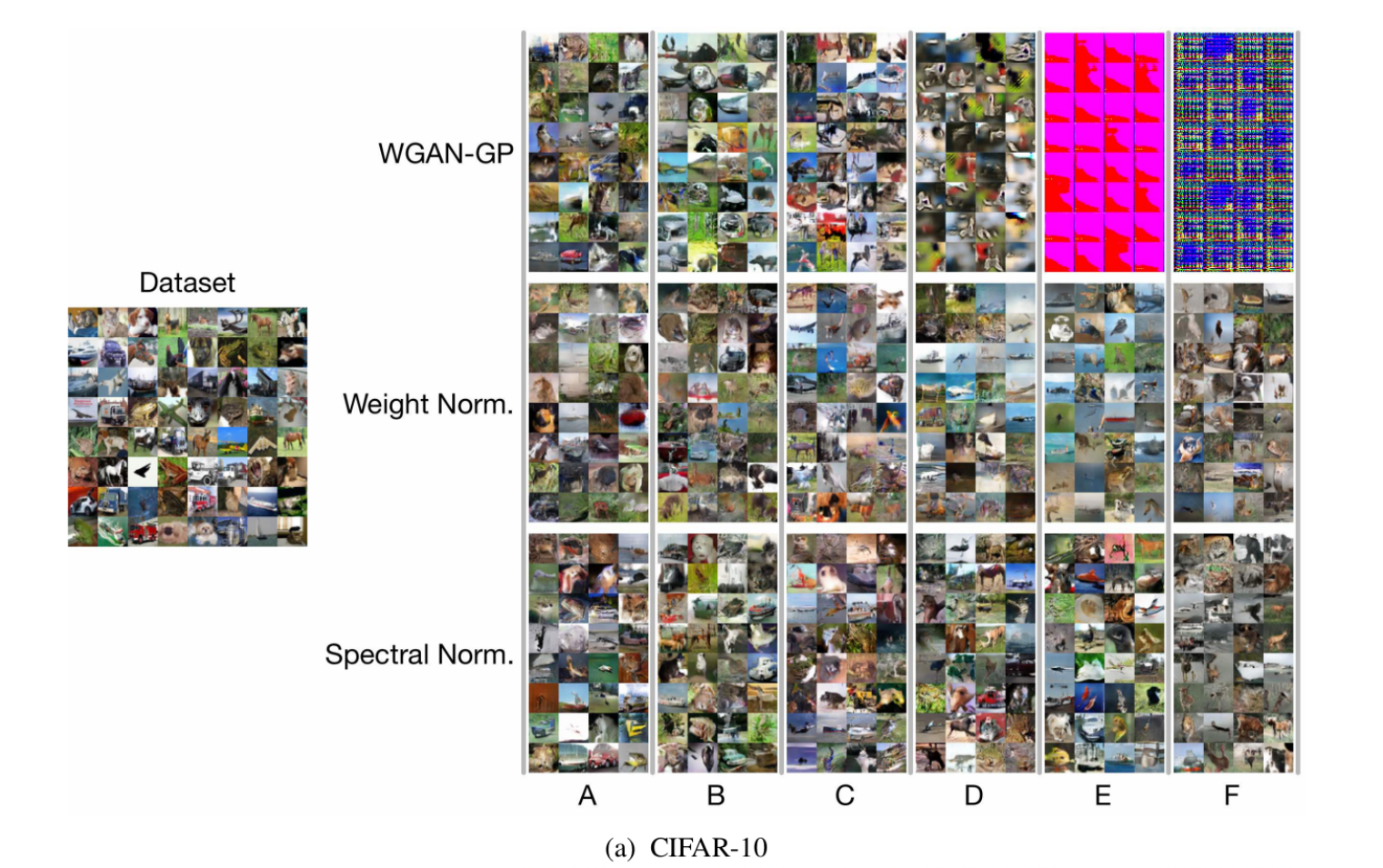


Figure 4: Generated images on different methods on CIFAR-10

ILSVRC2012 dataset: large high dimensional.

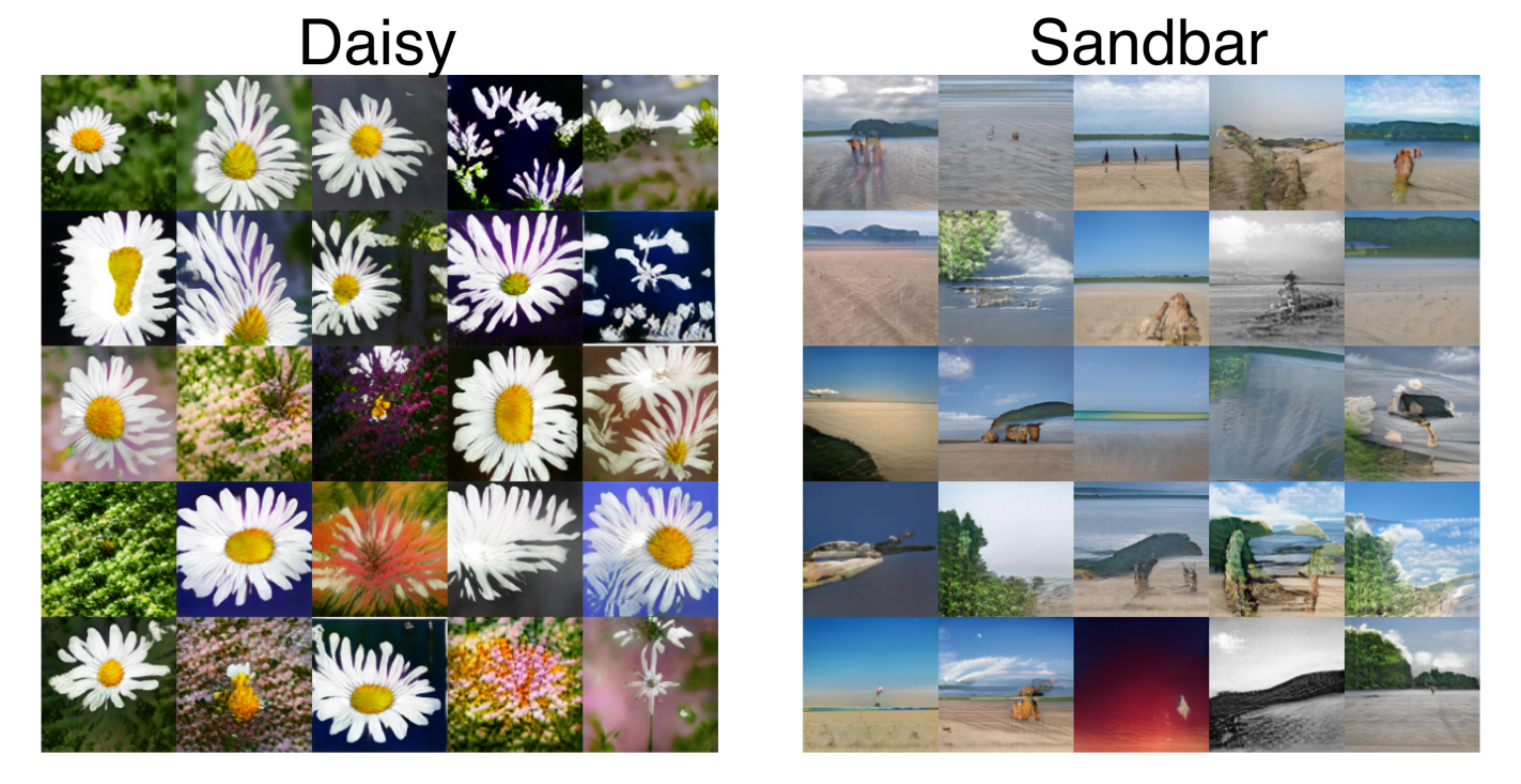
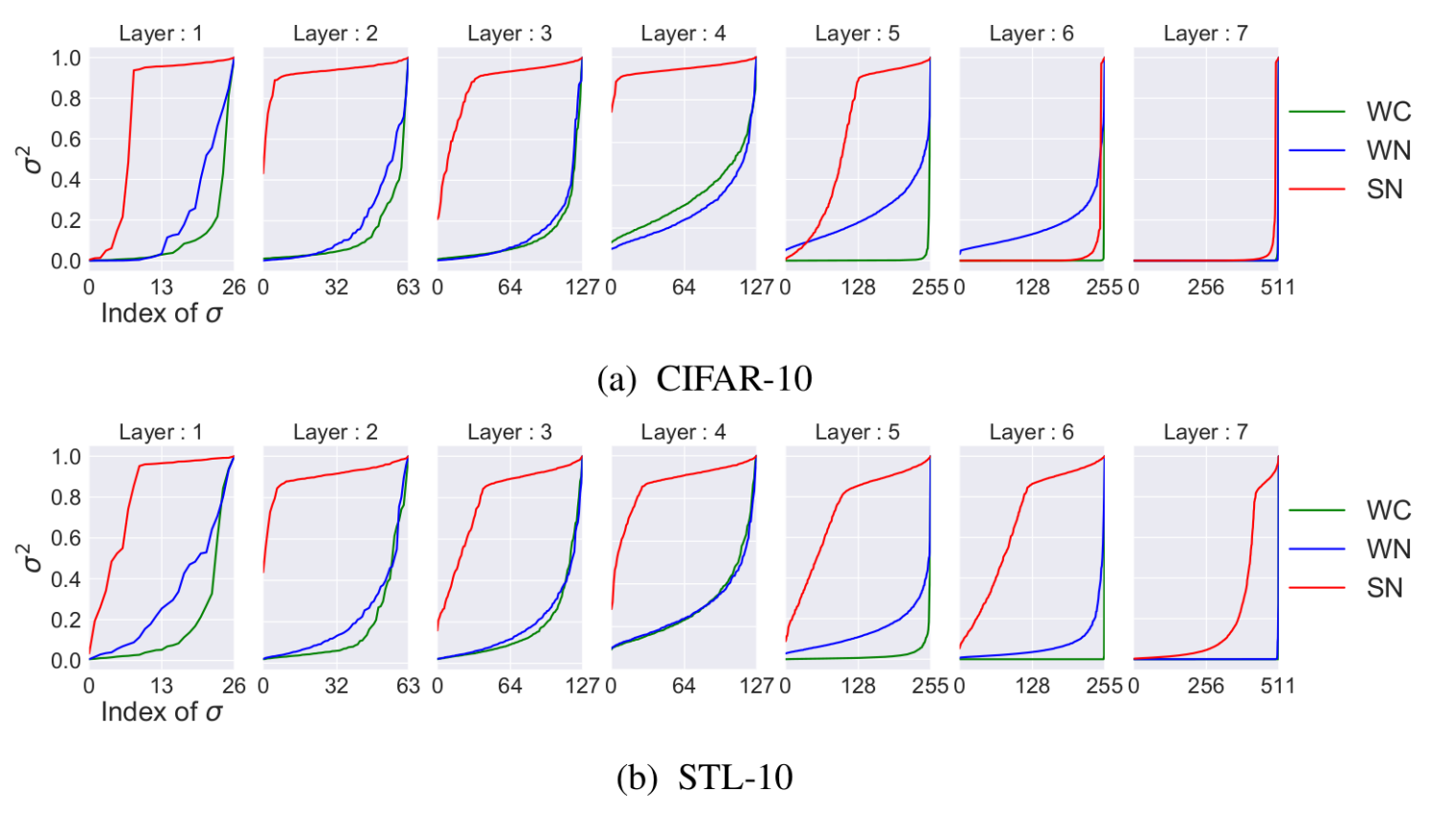


Figure 5: 128x128 pixel images generated by SN-GANs trained on ILSVRC2012 dataset

Singular Value Visualization



(a) CIFAR-10

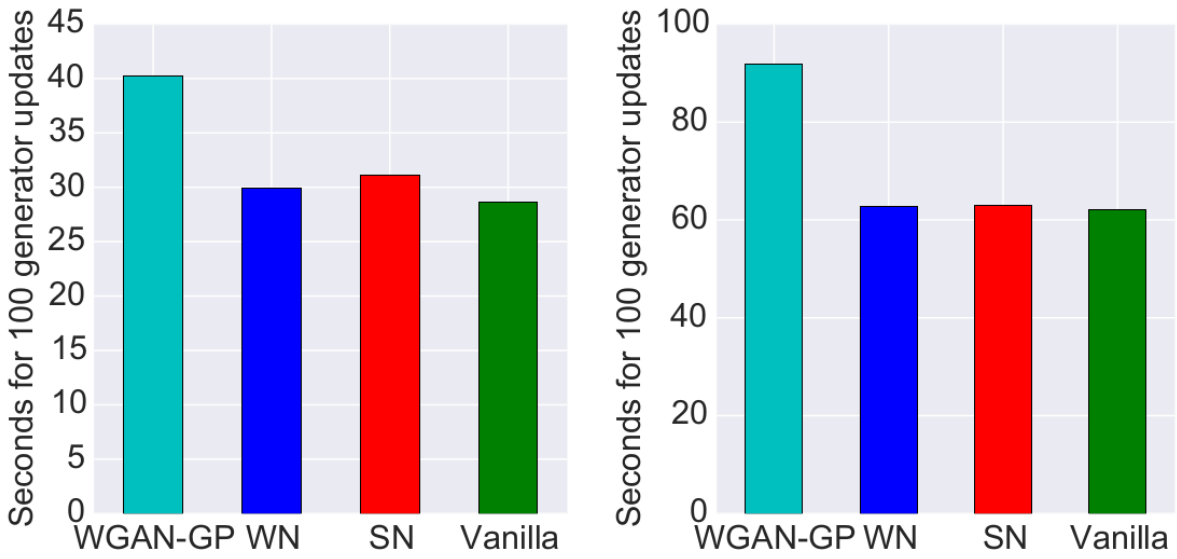
(b) STL-10

Figure 6: Squared singular value of weigth matrix trained with different methods: Weight Clipping (WC), Weight Normalization (WN), Spectral Normalization (SN)

Singular Value Visualization

- recall that SN can prevent the column space of weight matrix from being concentrating on one direction
- thus the singular values is more broadly distributed
- thus SN-GAN can generate more diversity, relatively

Training Time



(a) CIFAR-10 (image size:32 $\times$ 32 $\times$ 3)

(b) STL-10 (images size:48 $\times$ 48 $\times$ 3)

Figure 6: Computational time for 100 updates

Conclusion

Conclusion

- Proposed spectral normalization as a stabilizer
- SN-GANs generate more diversity and have better performance
- Spectral normalization is computational light

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*, pp. 2672–2680, 2014.

[2] 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*, pp. 2234–2242, 2016.

[3] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *arXiv preprint arXiv:1706.08500*, vol. 12, no. 1, 2017.

Thank you!