# Spectral normalization in GANs

Kirill Mazur, Alexei Pankov, Maria Taktasheva

# Background

- There are many recent breakthroughs in GANs
- It is **hard to make the training process stable**
- Popular idea: enforce **Lipschitz continuity** for stability

As a result: many tricks to enforce Lipschitz continuity

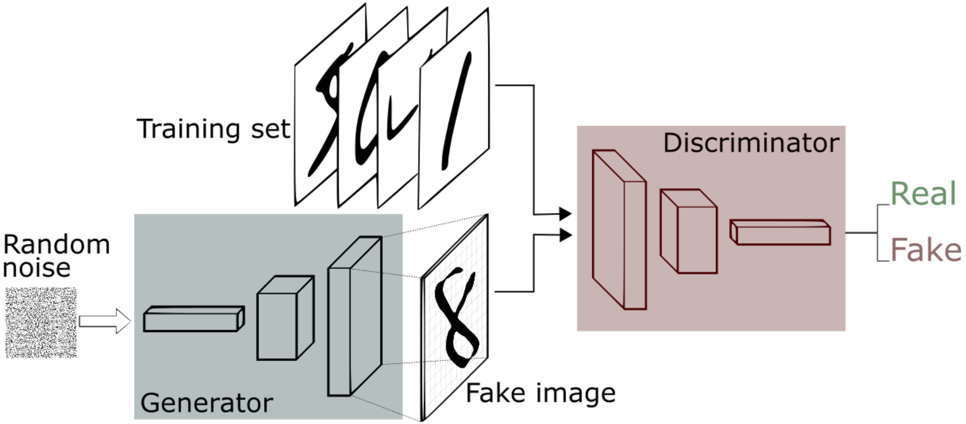



Image: https://skymind.ai/wiki/generative-adversarial-network-gan

Samples from BigGAN (https://arxiv.org/abs/1809.11096)

# GANs are hard to train

The formulation of GANs is given as a **min-max problem:**

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} \log D(x) + \mathbb{E}_{z \sim p_G(z)} \log(1 - D(G(z)))$$

- The main problem is **vanishing gradient** — the gradient of discriminator is unbounded, so the generator learns nothing
- A practical way to bound the gradient is to choose the Discriminator as a K-Lipschitz continuous function:

$$D = \arg \max_{\|f\|_{Lip} \leq K} V(G, D)$$

where $\|f\|_{Lip} : \dfrac{\|f(x) - f(x')\|}{\|x - x'\|} \leq K$

# Spectral normalization

- Spectral normalization enforces Lipschitz continuity by **controlling the spectral norm of each layer**
- The spectral norm is found via power iteration
- However, in the original paper, spectral normalization for convolutional layers is **implemented by simply reshaping the weights matrix**
- This is **not the true estimation of the operator norm**

$$W_{new} := \frac{W}{\sigma(W)}$$

$$\sigma(A) = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$$

Miyato, Takeru, et al. “Spectral normalization for generative adversarial networks.” arXiv preprint arXiv: 1802.05957 (2018).

# Spectral normalization

- A way to obtain a **true spectrum through FFT** was proposed in an ICLR 2019 Submission paper
- However, a computation of the whole spectrum is excess and computationally inefficient
- We tried to use **power iteration without kernel reshape** for true spectral norm estimation

**Calculate spectral norm:**

Initialize $u$ and $v$ as 3D tensors

Iterate:

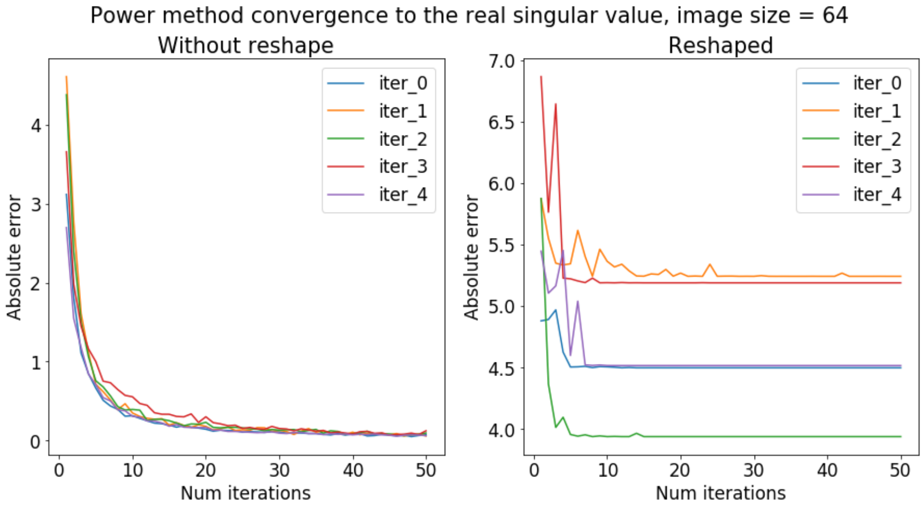
$$v := ConvTransposed(u) = W^T u$$
$$v := normed(v)$$
$$u := Conv(v) = Wv$$
$$u := normed(u)$$
$$\sigma(W) := v^T W^T u = \frac{v^T W^T W v}{\|Wv\|} = \|Wv\|$$

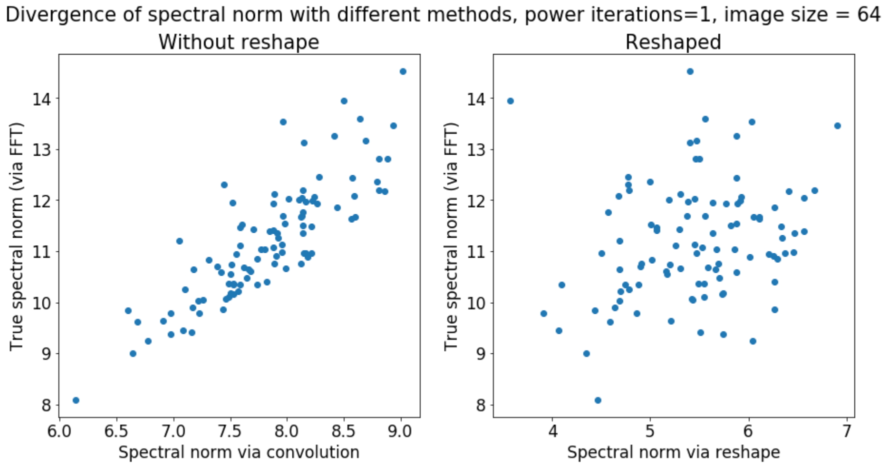
Where the norm is taken for flattened vectors

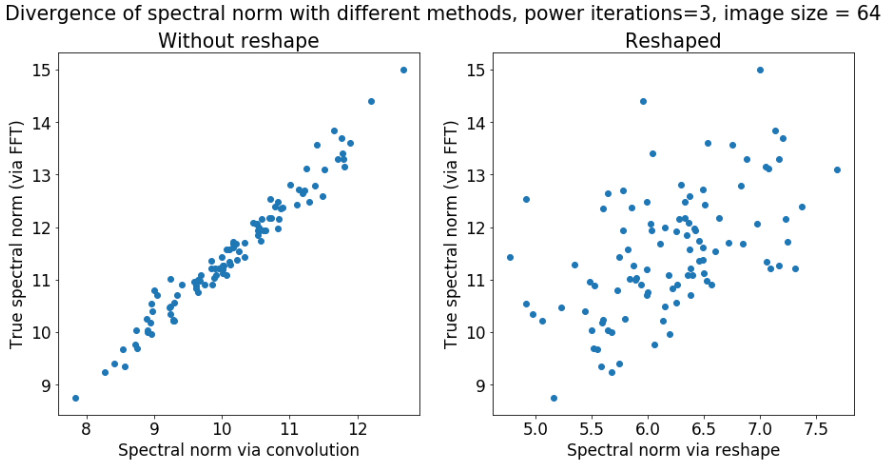
"The Singular Values of Convolutional Layers." ICLR 2019 Conference Blind Submission, https://openreview.net/forum?id=rJevYoA9Fm

# Spectral norm estimation

- The three approaches were first compared on random matrices
- **Spectral norm obtained with weight matrix reshape did not converge to a true norm**



# Training GAN

**Architecture:** Convolutional ResNet

**Dataset:** CIFAR-10

- We **tried both differential and non-differential version of power iteration** for spectral norm without reshape

- But the **GAN collapses almost immediately**

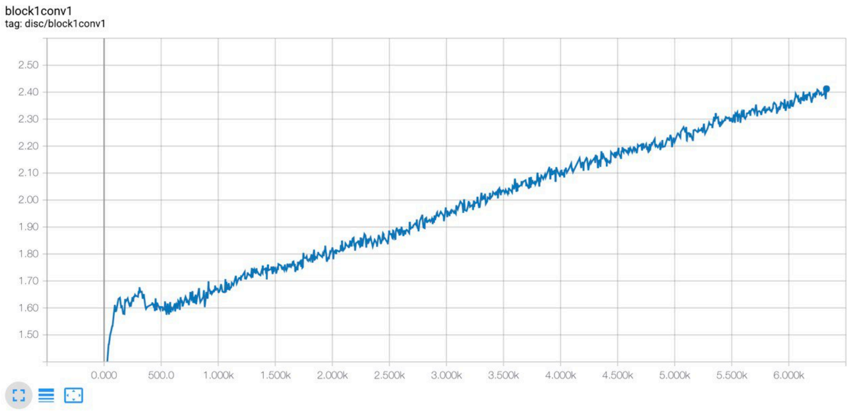
- Different hyperparameters did not help much

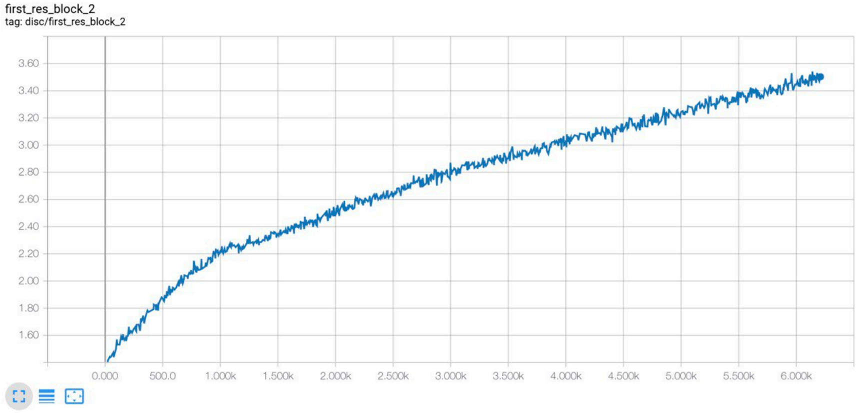
# Training GAN

- Singular values of **reshaped** convolutional layers increase from 1.5-1.6 to 2.5-3.6 during the training process

- Singular values of **non-reshaped** convolutional layers converge to 1 after some time, but it may happen due to GAN collapse

**Singular values of reshaped convolutional layers**



**The main question:**
why the reshaped
version works so well?

# References


- Miyato, Takeru, et al. “Spectral normalization for generative adversarial networks.” arXiv preprint arXiv:1802.05957 (2018).
- “The Singular Values of Convolutional Layers.” ICLR 2019 Conference Blind Submission, https://openreview.net/forum?id=rJevYoA9Fm
- Gouk, Henry, et al. “Regularisation of Neural Networks by Enforcing Lipschitz Continuity.” arXiv preprint arXiv:1804.04368 (2018).

# Team members contribution

**Kirill Mazur** — coding, theoretical justifications, report, presentation

**Alexei Pankov** — coding, report, presentation

**Maria Taktasheva** — spectral norm estimation on toy examples, report, presentation