Spectral Normalization for Generative Adversarial Networks

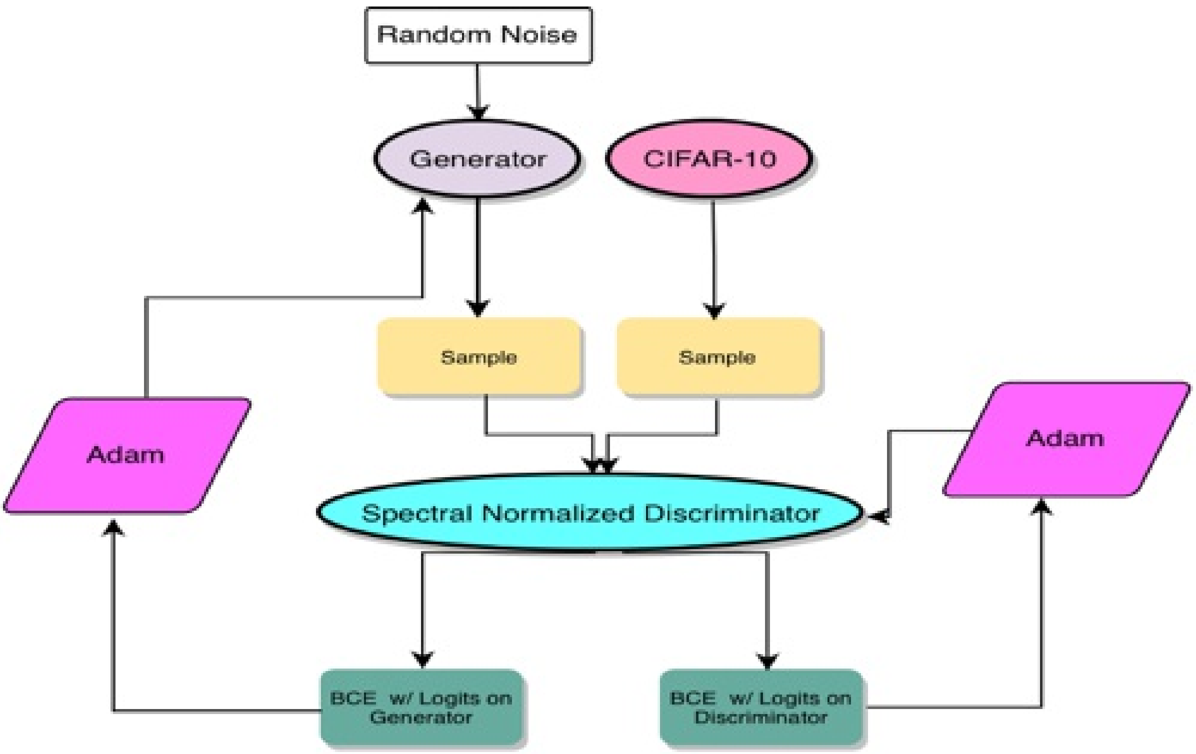
https://arxiv.org/pdf/1802.05957.pdf

EuiYul Song (20214426)

Files

- train.py: trains spectral gans for 320 epochs (50,000 iterations)
- model.py: contains a generator and spectral normalized discriminator
- util/dataset.py: downloads and preprocesses CIFAR-10
- util/generate_fake_image.py: generates fake images for every 1000 iterations of training
- util/generate_inception_score.py: generates inception score for every 1000 iterations of training
- util/inception.py: creates the Inception model
    - If you want to print the print inception score every certain epoch, please copy https://github.com/hvy/chainer-inception-score/blob/master/inception_score.py to this file (I did not add this contents into my file due to plagiarism issue I may face), and uncomment the line 24 in train.py and line 4 in util/generate_inception_score.py
- util/inception_model: pretrained inception model
- util/trainer.py training loop implementation of trainer
- Makefile: a command line automation

Architecture



Overview

A discriminator of the standard GANs is inaccurate and unstable during training and its derivative can be unbounded and incomputable. Thus, this spectral normalization impose regularization on the space outside of the support of a generator and data distribution. The formulas of the weight normalization through spectral normalization are as follows:

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

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

A spectral normalization uses power iteration method below and approximation of the spectral norm to estimate σ(W).

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:
  $$\tilde{\boldsymbol{v}}_l \leftarrow (W^l)^{\mathrm{T}}\tilde{\boldsymbol{u}}_l / \|(W^l)^{\mathrm{T}}\tilde{\boldsymbol{u}}_l\|_2 \tag{20}$$
  $$\tilde{\boldsymbol{u}}_l \leftarrow W^l\tilde{\boldsymbol{v}}_l / \|W^l\tilde{\boldsymbol{v}}_l\|_2 \tag{21}$$
  2. Calculate $\bar{W}_{\mathrm{SN}}$ with the spectral norm:
  $$\bar{W}^l_{\mathrm{SN}}(W^l) = W^l/\sigma(W^l), \text{ where } \sigma(W^l) = \tilde{\boldsymbol{u}}_l^{\mathrm{T}} W^l \tilde{\boldsymbol{v}}_l \tag{22}$$
  3. Update W^l with SGD on mini-batch dataset $\mathcal{D}_M$ with a learning rate α:
  $$W^l \leftarrow W^l - \alpha\nabla_{W^l}\ell(\bar{W}^l_{\mathrm{SN}}(W^l), \mathcal{D}_M) \tag{23}$$

Data

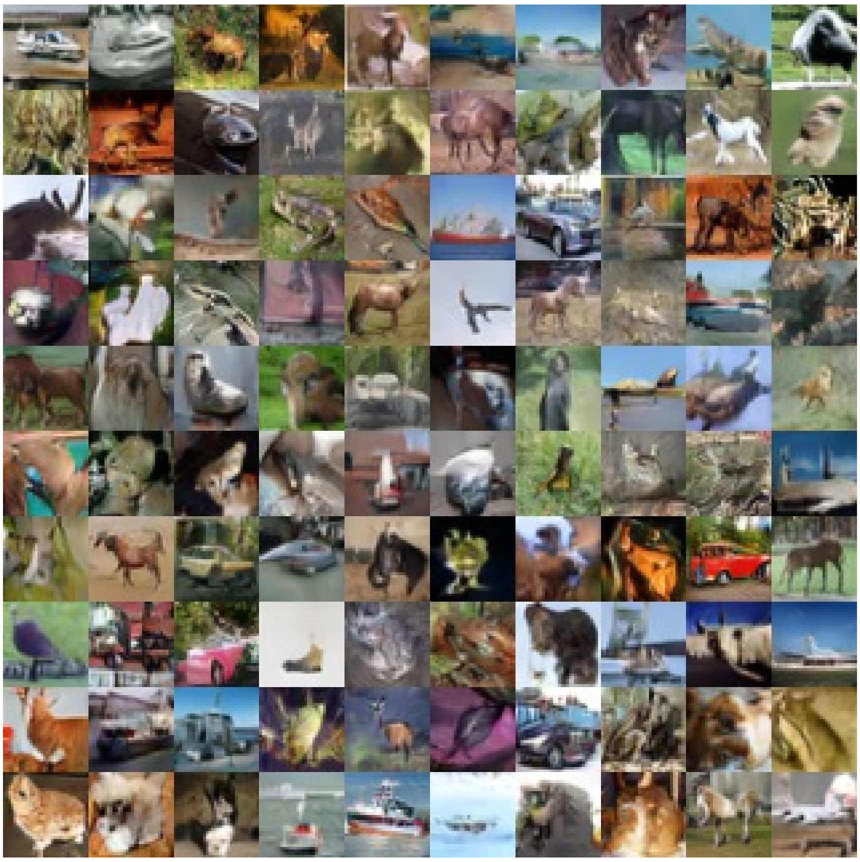
- CIFAR-10: downloads from the chainer library

Command

- make normalize: preprocess, train, and evaluate the model

Result

- My spectral normalized GANs (SNGANs) generate fake images accurately and stably. For instance, the images below are the images generated by the SNGANs after trained for 320 epochs (50,000 iterations). Objects depicted in the image are very intuitive that they are easy to categorize with naked eyes.



- **Inception Scores**

Looking at the chart below, my spectral normalization is relatively robust with aggressive learning rates and momentum parameters

α	β1	β2	ndis	inception_score
0.0002	0.5	0.9	5	8.7
0.0001	0.5	0.9	5	7.6
0.001	0.5	0.9	5	8.2
0.001	0.5	0.999	5	8.6
0.001	0.9	0.999	5	8.2
0.0001	0.5	0.999	1	6.3

Limitation

- Due to time-constraint to complete this project, I was not able to run my model against the STL-10 dataset nor compare my model with other regularization techniques.