Improving Generative Adversarial Networks with Denoising Feature Matching

David Warde-Farley[1] Yoshua Bengio[1]

[1]University of Montreal,

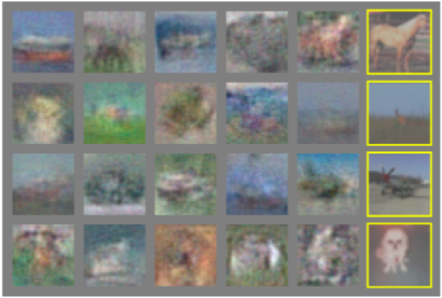
ICLR,2017

Presenter: Bargav Jayaraman

Outline

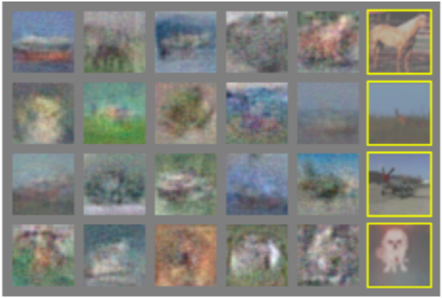
Introduction

GANs do not perform well in the reconstruction of real images with ‘objects’. Following are some examples from CIFAR dataset:



Introduction

GANs do not perform well in the reconstruction of real images with 'objects'. Following are some examples from CIFAR dataset:



Goal: To alter the training criteria to obtain 'objectness' in the synthesis of images.

Outline

Generative Adversarial Networks

- Adversarial game between generator G and discriminator D:

$$\arg\min_G \arg\max_D \mathbb{E}_{x\sim\mathcal{D}} \log D(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z))) \qquad (1)$$

Generative Adversarial Networks

- Adversarial game between generator G and discriminator D:

$$\arg\min_G \arg\max_D \mathbb{E}_{x\sim\mathcal{D}} \log D(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z))) \tag{1}$$

- Minimizing the above with respect to G is difficult and hence the following criterion is used in practice:

$$\arg\max_G \mathbb{E}_{z\sim p(z)} \log D(G(z)) \tag{2}$$

GAN Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

---

for number of training iterations **do**

  for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Outline

Challenges and Limitations of GANs

- Maximizing the original GAN equation with respect to D is infeasible to perform exactly. Thus G minimizes lower bound of correct objective function

$$\arg\min_G \arg\max_D \mathbb{E}_{x\sim\mathcal{D}} \log D(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z)))$$

Challenges and Limitations of GANs

- G collapses to generate near duplicate images in independent draws and with lower diversity of samples than what is observed in the real dataset

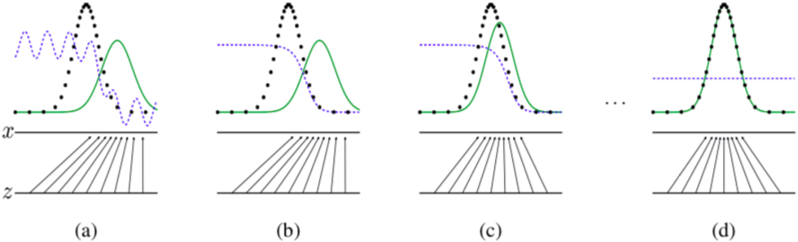


Figure 1: Generative adversarial nets are trained by simultaneously updating the **d**iscriminative distribution (D, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) $p_{\boldsymbol{x}}$ from those of the **g**enerative distribution p_g (G) (green, solid line). The lower horizontal line is the domain from which $\boldsymbol{z}$ is sampled, in this case uniformly. The horizontal line above is part of the domain of $\boldsymbol{x}$. The upward arrows show how the mapping $\boldsymbol{x} = G(\boldsymbol{z})$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g. (a)

Challenges and Limitations of GANs

- GANs lack a closed form of likelihood, and hence it is difficult to quantitatively evaluate the performance

Challenges and Limitations of GANs

- GANs lack a closed form of likelihood, and hence it is difficult to quantitatively evaluate the performance
- Inception score is a metric provided by Salimans et al. which uses Inception CNN to compute:

$$I(\{x\}_1^N) = \exp(\mathbb{E}[D_{KL}(p(y|x)||p(y))])$$

To get high inception score:

- $p(y|x)$ should have low entropy for image with meaningful objects
- $\int p(y|x = G(z))dz$ should have high entropy to identify a wide variety of classes

Related Work

1. Salimans et al. proposed feature matching as an alternative training criterion for GAN generators

$$\arg\min_{\theta_G} \|\mathbb{E}_{x\sim\mathcal{D}}[\phi(x)] - \mathbb{E}_{z\sim p(z)}[\phi(G(z))]\|^2$$

   where ϕ is the high level feature mapping of discriminator. The authors use semi-supervised training.
2. Enegry-based GANs by Zhao et al. replace discriminator with auto-encoder and reconstructs the training data. Assigns low energy to real data and high energy to samples from G
3. Sonderby et al. train a denoising AE to get the difference between synthesized real image and output of denoising AE and pass it as a signal to train super-resolution network.

Improving GAN Training

- *Denoising feature matching* is proposed as an added criterion for training G.
- Denoising AE $r()$ is trained on data from distribution $q(h)$, and estimates via $r(h) - h$ the gradient of true log-density $\frac{\partial \log q(h)}{\partial h}$
- Train denoising AE on $h = \phi(x)$, with $x \sim \mathcal{D}$, then $r(\phi(x') - \phi(x'))$ with $x' = G(z)$ will give the change to make $h = \phi(x')$
- Augmented training criterion for G:

$$\arg\min_{\theta_G} \mathbb{E}_{z\sim p(z)}[\lambda_{\text{denoise}}\|\phi(G(z)) - r(\phi(G(z)))\|^2 - \lambda_{\text{adv}}\log(D(G(z))] \tag{3}$$

$r()$ is trained as (C is the corruption function):

$$\arg_m in_{\theta_r}\mathbb{E}_{x\sim\mathcal{D}}\|\phi(x) - r(C(\phi(x)))\|^2$$

Experimental Setting

- Learning synthesis models from three datasets of increasing divesity and size: CIFAR-10, STL-10 and ImageNet
- Isotropic Gaussian corruption noise with $\sigma = 1$
- Batch normalization of discriminator, generator and all layers of denoising AE except the output layer
- Optimizing with Adam with learning rate of 10^{-4} and $\beta_1 = 0.5$, $\lambda_{\text{denoise}} = 0.03/n_h$ and $\lambda_{\text{adv}} = 1$

CIFAR-10

Real data*	Semi-supervised	Unsupervised	
	Improved GAN (Salimans *et al*)*	ALI (Dumoulin *et al*)†	Ours
$11.24 \pm .12$	$8.09 \pm .07$	5.34 ± 0.05	7.72 ± 0.13

Table 1: Inception scores for models of CIFAR-10. * as reported in Salimans et al. (2016); semi-supervised † computed from samples drawn using author-provided model parameters and implementation.

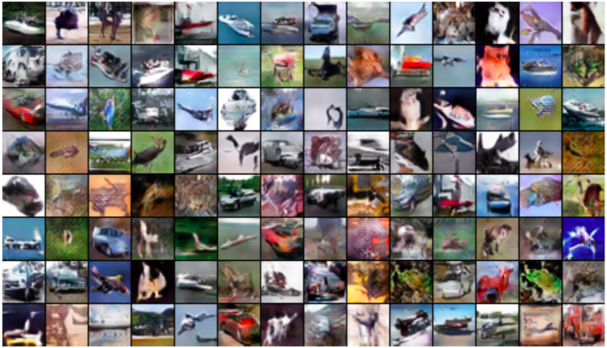


Figure 1: Samples generated from a model trained with denoising feature matching on CIFAR10.

STL-10

Real data	Ours	GAN Baseline
$26.08 \pm .26$	8.51 ± 0.13	$7.84 \pm .07$

Table 2: Inception scores for models of the unlabeled set of STL-10.

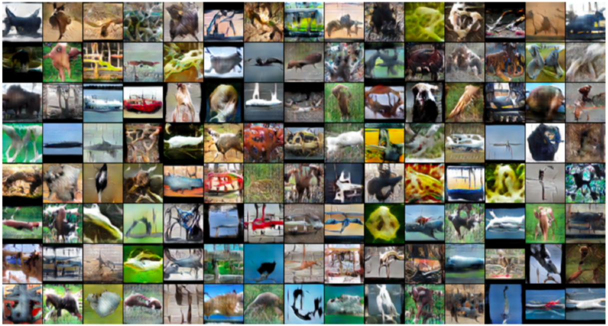


Figure 2: Samples from a model trained with denoising feature matching on the unlabeled portion of the STL-10 dataset.

ImageNet

Real data	Radford *et al*[*]	Ours
$25.78 \pm .47$	8.83 ± 0.14	$9.18 \pm .13$

Table 3: Inception scores for models of ILSVRC 2012 at 32×32 resolution. [*] computed from samples drawn using author-provided model parameters and implementation.

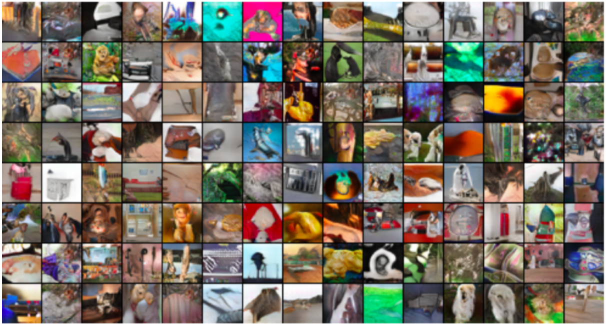


Figure 3: Samples from our model of ILSVRC2012 at 32×32 resolution.

Conclusion

1. Augmented objective criterion for training generator to synthesize distribution similar to real data distribution
2. Unsupervised training with mapping of higher dimension features of discriminator
3. Experimental evaluation on different datasets to show the effectiveness compared to existing approaches on recovering ‘objects’