Generative Adversarial Networks (GAN)

Hanxiao Liu

April 17, 2017

Outline

- Implicit Generative Models
- GAN
- Variants
  - f-GAN
  - Wasserstein GAN
- Use cases
  - Conditional generation
  - SSL

Implicit Generative Models

Working with likelihood could be expensive

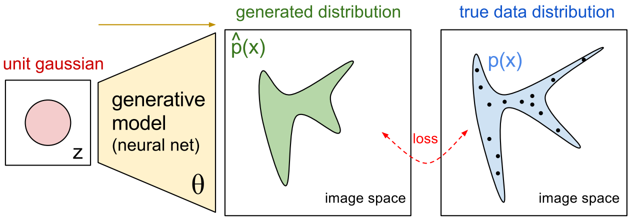
- Variational Autoencoder (variational inference)
- Boltzmann Machines (MCMC)
- PixelRNN (conditional prob.)

Representation learning may not require likelihood.

- Sometimes we are more interested in taking samples from $p(x)$ instead of p itself.

More discussions: [Mohamed and Lakshminarayanan, 2016]

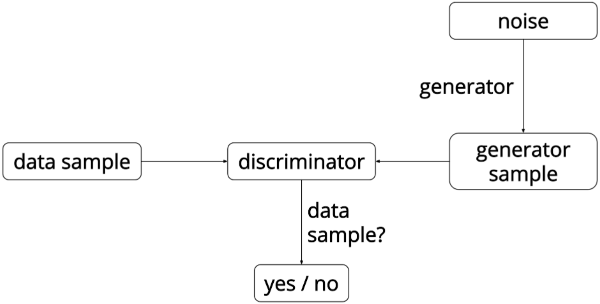
Implicit Generative Models



advocate/penalize samples within the blue/white region.

GAN

Unsupervised learning via supervised learning



GAN

Proposed by [Goodfellow et al., 2014]

A minimax game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim data}\log D(x) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z))) \quad (1)$$
$$J^{(G)} = -J^{(D)} \quad (2)$$

- The optimal discriminator $D^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_{model}(x)}$.
- In this case, $J^{(G)} = 2D_{JS}(p_{data}\|p_{model}) + const.$
- Jenson-Shannon divergence:
  $D_{JS}(p\|q) = \frac{1}{2}D_{KL}(p\|\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q\|\frac{p+q}{2})$.

GAN

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 $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

  • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ 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 $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

  • Update the generator by descending its stochastic gradient:

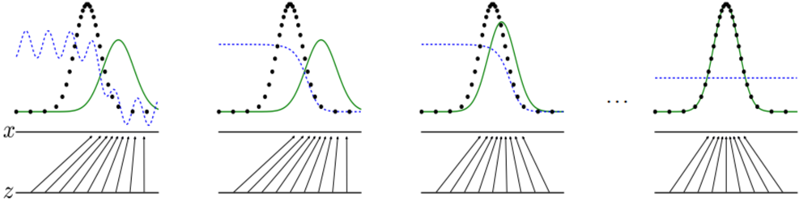
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

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

GAN

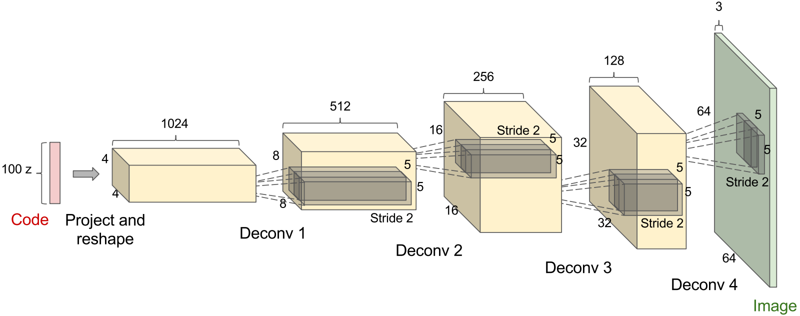
Visualizing the GAN training process



dots: data.
blue: discriminator.
green: generator.

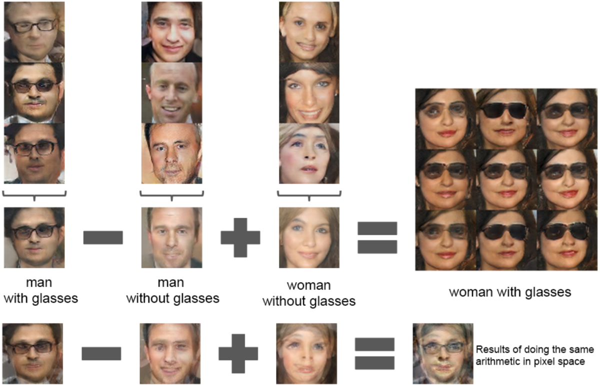
DCGAN

Deep Convolutional GAN [Radford et al., 2015]



Walk around the data manifold

Vector Arithmetic



f-divergence

Can we simply optimize the divergence?

$$D_f(p\|q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) dx \tag{3}$$

- recovers KL divergence when $f(u) = u \log u$.
- optimization $\min_q D_f(p\|q)$ can hardly be carried out with D_f in its original form.

f-GAN [Nowozin et al., 2016]

Fenchel conjugacy: $f(u) = \sup_t tu - f^*(t)$

$$\begin{aligned}
&\min_q \; D_f(P\|Q) && (4)\\
&= \min_q \; \int q(x) \left[\sup_t \left(t\frac{p(x)}{q(x)} - f^*(t)\right)\right] dx && (5)\\
&\geq \min_q \; \sup_T \; \int p(x)T(x) - f^*(T(x))q(x)dx && (6)\\
&= \min_q \; \sup_T \; \mathbb{E}_p T(x) - \mathbb{E}_q f^*(T(x)) && (7)\\
&\approx \min_w \; \max_\theta \; \mathbb{E}_p T_\theta(x) - \mathbb{E}_{q_\omega} f^*(T_\theta(x)) && (8)
\end{aligned}$$

The above includes GAN as its special case.

Training GAN

Training GAN (finding the equilibrium) is hard.

- Gradient for G will vanish when D is very good.
- i.e. when p_{data} and p_{model} are very different.
  - usually true for high-dimensional data.
  - f-divergence may be ill-defined.

Wasserstein GAN

Replace the pointwise f-divergence with Wasserstein distance [Arjovsky et al., 2017].

$$D_w(p\|q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma} dist(x,y) \tag{9}$$

where $\Pi(p,q)$ is the set of all joint distributions $\gamma(x,y)$ which marginal distributions are respectively p and q.

- well defined even if p and q have different support.
- leads to a very simple algorithm using a similar variational trick for the f-divergence.

Wasserstein GAN

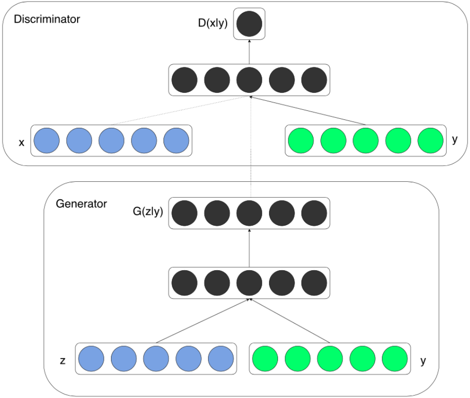
Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

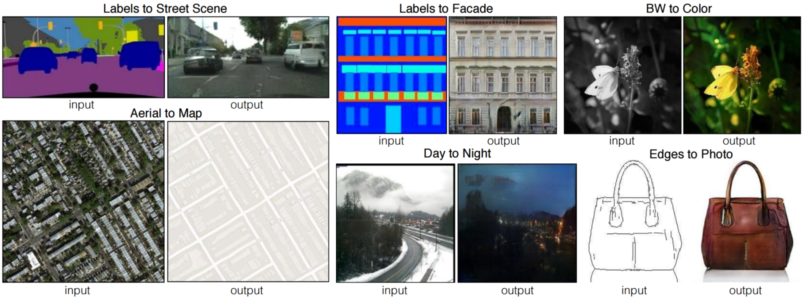
Conditional GAN

[Mirza and Osindero, 2014]



Conditional GAN

Image to image translation [Isola et al., 2016].



Semi-supervised Learning

- Data augmentation
- Regularization [Salimans et al., 2016]
- Categorical GAN [Springenberg, 2015]

Reference I

Arjovsky, M., Chintala, S., and Bottou, L. (2017).
Wasserstein gan.
arXiv preprint arXiv:1701.07875.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016).
Image-to-image translation with conditional adversarial networks.
arXiv preprint arXiv:1611.07004.

Mirza, M. and Osindero, S. (2014).
Conditional generative adversarial nets.
arXiv preprint arXiv:1411.1784.

Mohamed, S. and Lakshminarayanan, B. (2016).
Learning in implicit generative models.
arXiv preprint arXiv:1610.03483.

Reference II

Nowozin, S., Cseke, B., and Tomioka, R. (2016).
f-gan: Training generative neural samplers using variational divergence minimization.
In *Advances in Neural Information Processing Systems*, pages 271–279.

Radford, A., Metz, L., and Chintala, S. (2015).
Unsupervised representation learning with deep convolutional generative adversarial networks.
arXiv preprint arXiv:1511.06434.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016).
Improved techniques for training gans.
In *Advances in Neural Information Processing Systems*, pages 2226–2234.

Springenberg, J. T. (2015).
Unsupervised and semi-supervised learning with categorical generative adversarial networks.
arXiv preprint arXiv:1511.06390.