ENM531: Data-driven modeling and probabilistic scientific computing

Lecture #23: Generative adversarial networks

Paris Perdikaris
April 28, 2020

Penn
University of Pennsylvania

Tricks of the trade

- Variational bounds
- Density re-parametrizations
- Density ratio estimation
- Variational optimization/evolution strategies
- Adversarial games

The density ratio trick

The central task in the above five statistical quantities is to efficiently compute the ratio $r(x)$. In simple problems, we can compute the numerator and the denominator separately, and then compute their ratio. Direct estimation like this will not often be possible: each part of the ratio may itself involve intractable integrals; we will often deal with high-dimensional quantities; and we may only have samples drawn from the two distributions, not their analytical forms.

This is where the ***density ratio trick*** *or formally, density ratio estimation*, enters: it tells us to construct a binary classifier $\mathcal{S}(x)$ that distinguishes between samples from the two distributions. We can then compute the density ratio using the probability given by this classifier:

$$r(x) = \frac{\rho(x)}{q(x)} = \frac{\mathcal{S}(x)}{1 - \mathcal{S}(x)}$$

To show this, imagine creating a data set of $2N$ elements consisting of pairs (data x, label y):

- → N data points are drawn from the distribution ρ and assigned a label +1.
- → The remaining N data points are drawn from distribution q and assigned label −1.

http://blog.shakirm.com/2018/01/machine-learning-trick-of-the-day-7-density-ratio-trick/

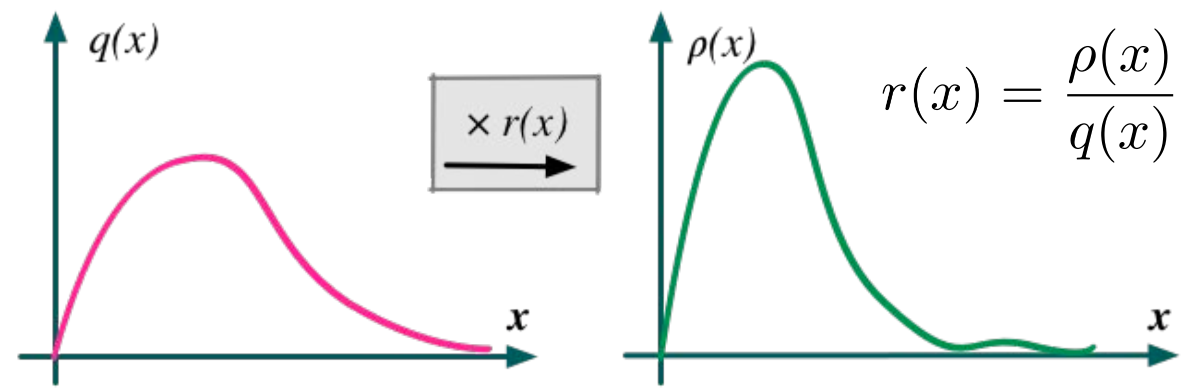
Density ratio estimation by probabilistic classification

$$\mathbb{KL}[p(\boldsymbol{x})||q(\boldsymbol{x})] := \int \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} p(\boldsymbol{x}) d\boldsymbol{x} = \mathbb{E}_{p(\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]$$

Estimating density ratios is a challenging task:

- Each part of the ratio may itself involve intractable integrals
- We often deal with high-dimensional quantities.
- We may only have samples drawn from the two distributions, not their analytical forms.

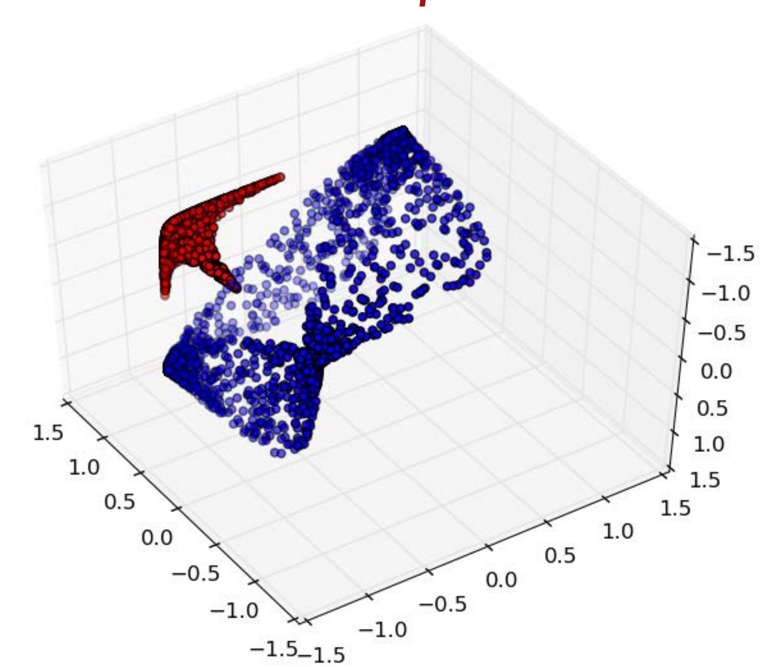
This is where the ***density ratio trick*** enters: it allows us to construct a binary classifier that distinguishes between samples from the two distributions.



$$r(x) = \frac{\rho(x)}{q(x)}$$

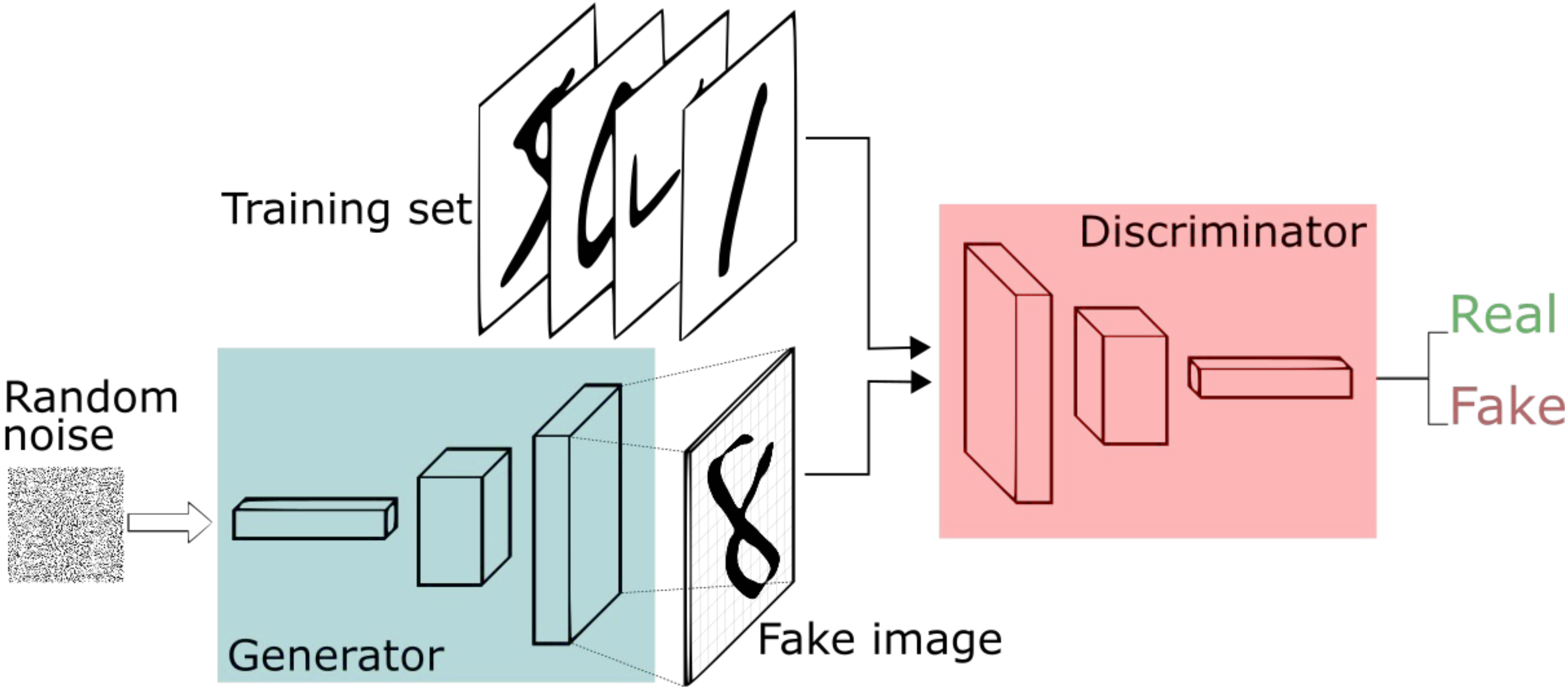
The density ratio gives the correction factor needed to make two distributions equal.

$$\begin{aligned} r(x) = \frac{\rho(x)}{q(x)} &= \frac{p(x|y=+1)}{p(x|y=-1)} \\ &= \frac{p(y=+1|x)p(x)}{p(y=+1)} \Big/ \frac{p(y=-1|x)p(x)}{p(y=-1)} \\ &= \frac{p(y=+1|x)}{p(y=-1|x)} = \frac{p(y=+1|x)}{1-p(y=+1|x)} = \frac{\mathcal{S}(x)}{1-\mathcal{S}(x)} \end{aligned}$$



http://blog.shakirm.com/2018/01/machine-learning-trick-of-the-day-7-density-ratio-trick/

Generative adversarial networks



$$\min_G \max_D V(D,G) = \mathbb{E}_{q(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

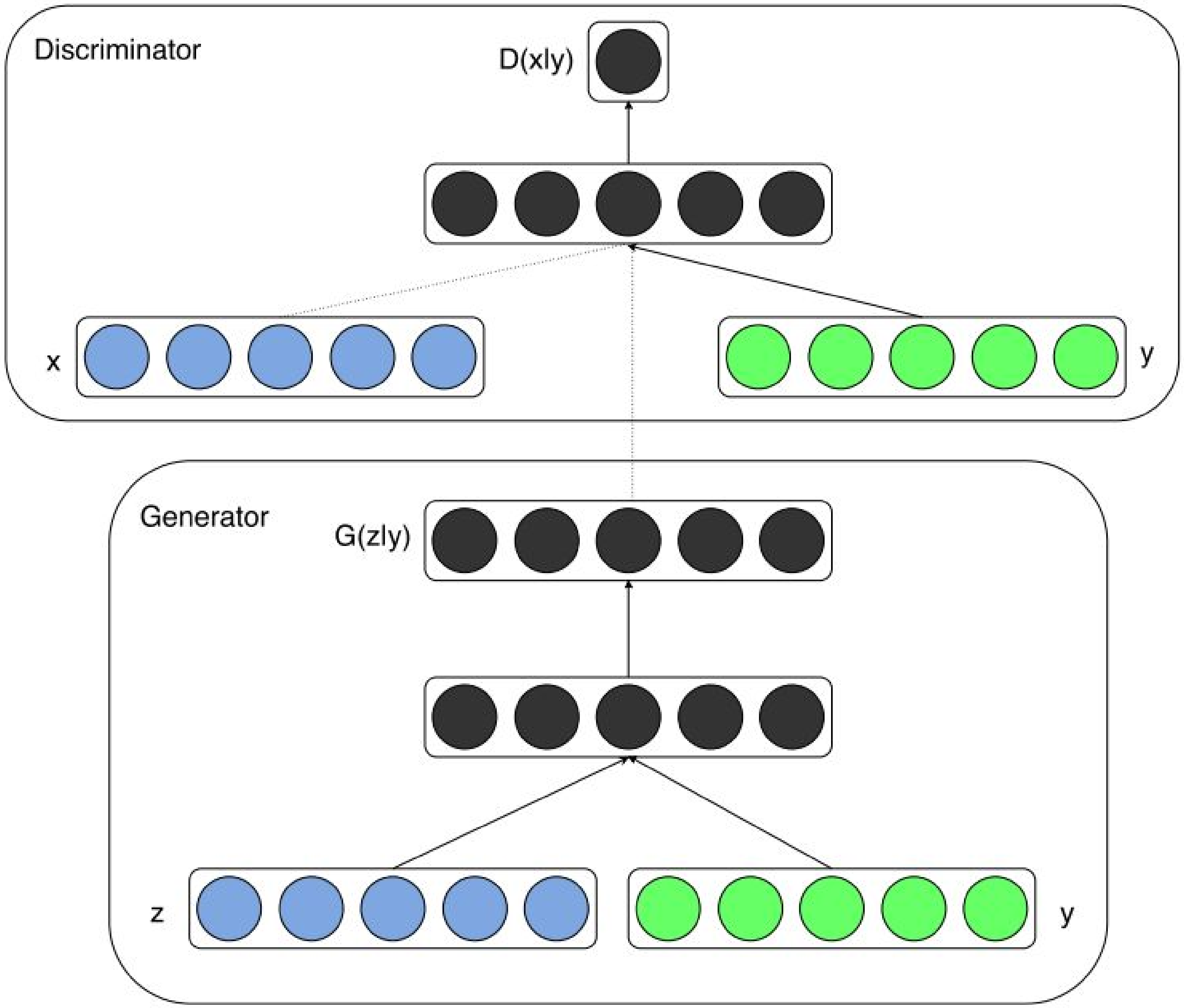
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

Generative adversarial networks

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

Conditional generative adversarial networks

Mirza and Osindero (2014)



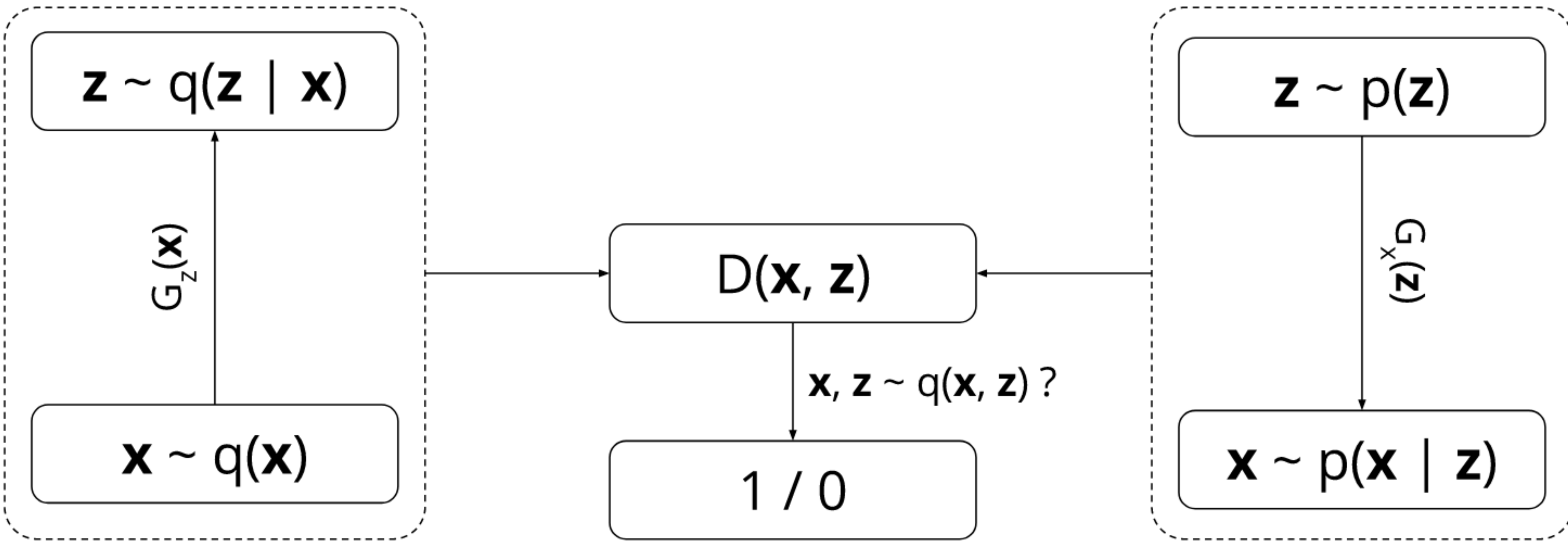
GAN $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$

CGAN $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$

Conditional generative adversarial networks

Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8798-8807).

Adversarially learned inference

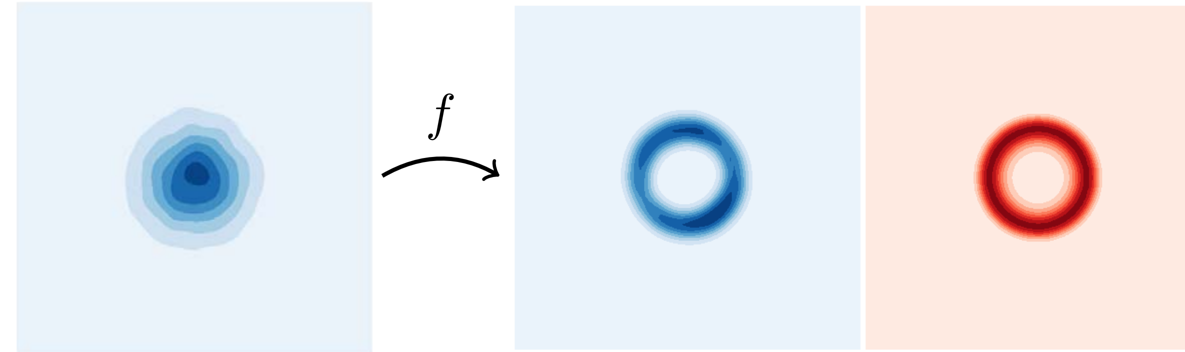


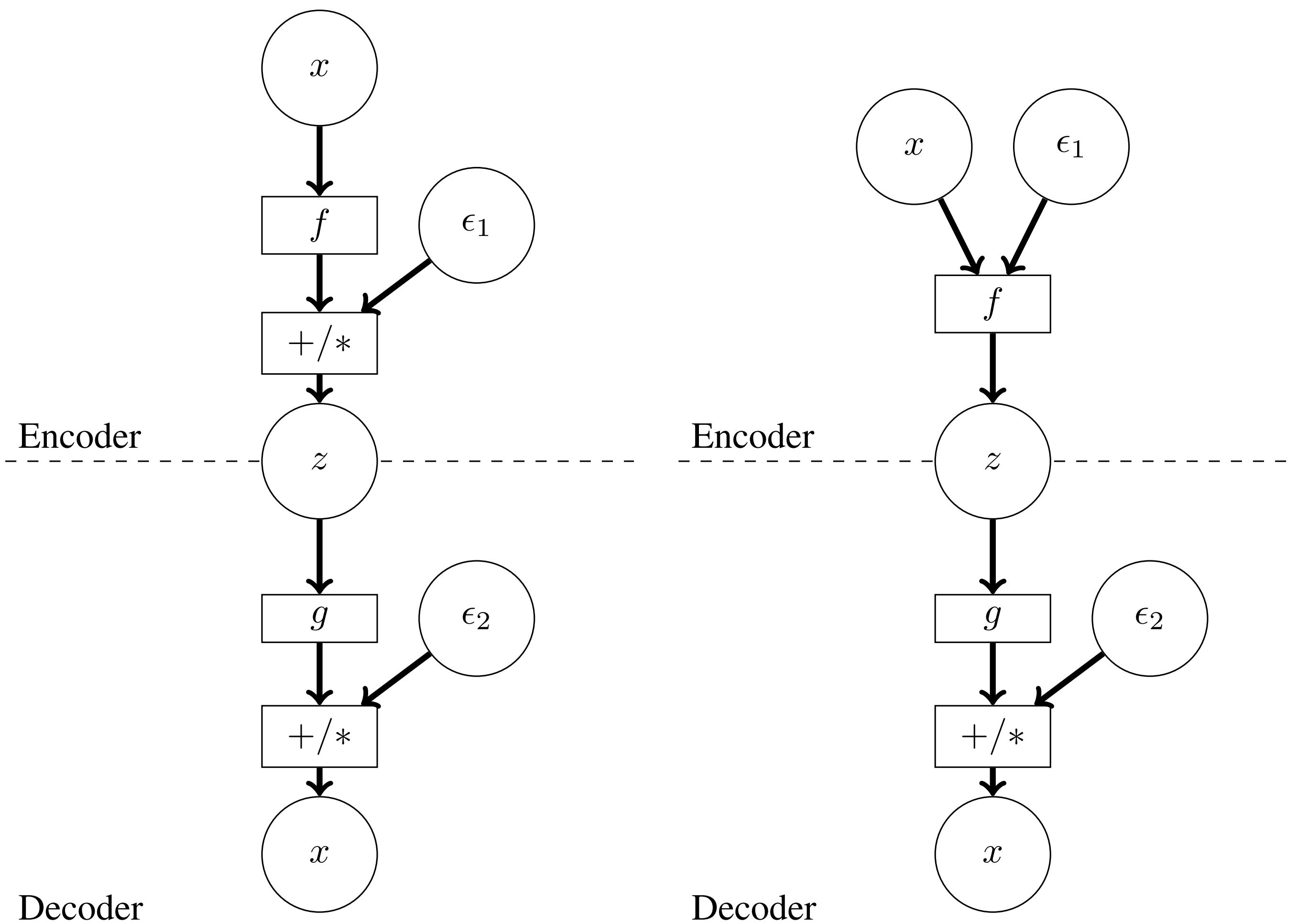
$$\min_G \max_D V(D,G) = \mathbb{E}_{q(\mathbf{x})}[\log(D(\mathbf{x}, G_z(\mathbf{x})))] + \mathbb{E}_{p(\mathbf{z})}[\log(1 - D(G_x(\mathbf{z}), \mathbf{z}))]$$

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., & Courville, A. (2016). Adversarially learned inference. arXiv preprint arXiv:1606.00704.

Adversarial Variational Bayes

Instead of using approximating distributions of a given pre-defined form (e.g. Gaussian) we can implicitly parametrize them using deep neural networks.





(a) Standard VAE (b) Our model

Algorithm 1 Adversarial Variational Bayes (AVB)

$i \leftarrow 0$

while not converged **do**

Sample $\{x^{(1)}, \dots, x^{(m)}\}$ from data distrib. $p_{\mathcal{D}}(x)$

Sample $\{z^{(1)}, \dots, z^{(m)}\}$ from prior $p(z)$

Sample $\{\epsilon^{(1)}, \dots, \epsilon^{(m)}\}$ from $\mathcal{N}(0,1)$

Compute θ-gradient (eq. 3.7):

$$g_\theta \leftarrow \frac{1}{m}\sum_{k=1}^{m} \nabla_\theta \log p_\theta \left(x^{(k)} \mid z_\phi\left(x^{(k)}, \epsilon^{(k)}\right)\right)$$

Compute ϕ-gradient (eq. 3.7):

$$g_\phi \leftarrow \frac{1}{m}\sum_{k=1}^{m} \nabla_\phi \Big[-T_\psi\left(x^{(k)}, z_\phi(x^{(k)}, \epsilon^{(k)})\right) + \log p_\theta\left(x^{(k)} \mid z_\phi(x^{(k)}, \epsilon^{(k)})\right)\Big]$$

Compute ψ-gradient (eq. 3.3) :

$$g_\psi \leftarrow \frac{1}{m}\sum_{k=1}^{m} \nabla_\psi \Big[\log\left(\sigma(T_\psi(x^{(k)}, z_\phi(x^{(k)}, \epsilon^{(k)})))\right) + \log\left(1 - \sigma(T_\psi(x^{(k)}, z^{(k)}))\right)\Big]$$

Perform SGD-updates for θ, ϕ and ψ:

$\theta \leftarrow \theta + h_i\, g_\theta, \quad \phi \leftarrow \phi + h_i\, g_\phi, \quad \psi \leftarrow \psi + h_i\, g_\psi$

$i \leftarrow i + 1$

end while

Huszár, F. (2017). Variational inference using implicit distributions. arXiv preprint arXiv:1702.08235.

Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. arXiv preprint arXiv:1701.04722.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.