Generative Adversarial Networks

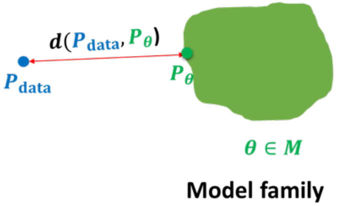
Stefano Ermon, Aditya Grover

Stanford University

Lecture 9

Recap

$\mathbf{x}_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$



- Model families
  - Autoregressive Models: $p_\theta(\mathbf{x}) = \prod_{i=1}^{n} p_\theta(x_i|\mathbf{x}_{<i})$
  - Variational Autoencoders: $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}$
  - Normalizing Flow Models: $p_X(\mathbf{x}; \theta) = p_Z\left(\mathbf{f}_\theta^{-1}(\mathbf{x})\right)\left|\det\left(\frac{\partial \mathbf{f}_\theta^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$
- All the above families are based on maximizing likelihoods (or approximations)
- Is the likelihood a good indicator of the quality of samples generated by the model?

Towards likelihood-free learning

- **Case 1:** Optimal generative model will give best **sample quality** and highest test **log-likelihood**
- For imperfect models, achieving high log-likelihoods might not always imply good sample quality, and vice-versa (Theis et al., 2016)

Towards likelihood-free learning

- **Case 2:** Great test log-likelihoods, poor samples. E.g., For a discrete noise mixture model $p_\theta(\mathbf{x}) = 0.01 p_{\text{data}}(\mathbf{x}) + 0.99 p_{\text{noise}}(\mathbf{x})$
  - 99% of the samples are just noise
  - Taking logs, we get a lower bound

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log[0.01 p_{\text{data}}(\mathbf{x}) + 0.99 p_{\text{noise}}(\mathbf{x})] \\ &\geq \log 0.01 p_{\text{data}}(\mathbf{x}) = \log p_{\text{data}}(\mathbf{x}) - \log 100\end{aligned}$$

  - For expected likelihoods, we know that
    - Lower bound

$$E_{p_{\text{data}}}[\log p_\theta(\mathbf{x})] \geq E_{p_{\text{data}}}[\log p_{\text{data}}(\mathbf{x})] - \log 100$$

    - Upper bound (via non-negativity of KL)

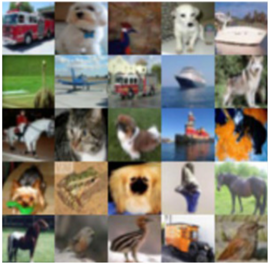
$$E_{p_{\text{data}}}[\log p_{\text{data}}(\mathbf{x}))] \geq E_{p_{\text{data}}}[\log p_\theta(\mathbf{x})]$$

  - As we increase the dimension of $\mathbf{x}$, absolute value of $\log p_{\text{data}}(\mathbf{x})$ increases proportionally but $\log 100$ remains constant. Hence, $E_{p_{\text{data}}}[\log p_\theta(\mathbf{x})] \approx E_{p_{\text{data}}}[\log p_{\text{data}}(\mathbf{x})]$ in very high dimensions

Towards likelihood-free learning

- **Case 3:** Great samples, poor test log-likelihoods. E.g., Memorizing training set
  - Samples look exactly like the training set (cannot do better!)
  - Test set will have zero probability assigned (cannot do worse!)
- The above cases suggest that it might be useful to disentangle likelihoods and samples
- **Likelihood-free learning** consider objectives that do not depend directly on a likelihood function

Comparing distributions via samples



$S_1 = \{\mathbf{x} \sim P\}$

vs.

$S_2 = \{\mathbf{x} \sim Q\}$

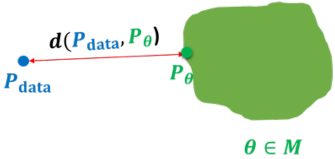
Given a finite set of samples from two distributions $S_1 = \{\mathbf{x} \sim P\}$ and $S_2 = \{\mathbf{x} \sim Q\}$, how can we tell if these samples are from the same distribution? (i.e., $P = Q$?)

Two-sample tests

- Given $S_1 = \{\mathbf{x} \sim P\}$ and $S_2 = \{\mathbf{x} \sim Q\}$, a **two-sample test** considers the following hypotheses
  - Null hypothesis H_0: $P = Q$
  - Alternate hypothesis H_1: $P \neq Q$
- Test statistic T compares S_1 and S_2 e.g., difference in means, variances of the two sets of samples
- If T is less than a threshold α, then accept H_0 else reject it
- **Key observation:** Test statistic is **likelihood-free** since it does not involve the densities P or Q (only samples)

Generative modeling and two-sample tests

$\mathbf{x}_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$

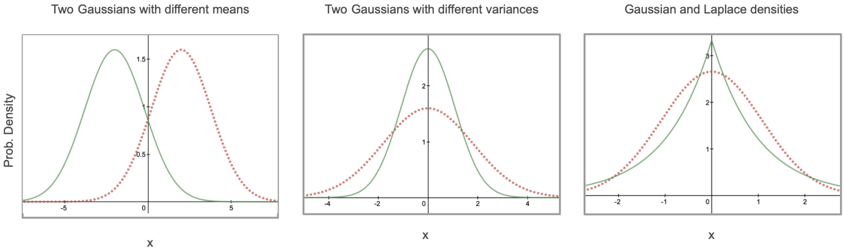


Model family

- Apriori we assume direct access to $S_1 = \mathcal{D} = \{\mathbf{x} \sim p_{\text{data}}\}$
- In addition, we have a model distribution p_θ
- Assume that the model distribution permits efficient sampling (e.g., directed models). Let $S_2 = \{\mathbf{x} \sim p_\theta\}$
- **Alternate notion of distance between distributions:** Train the generative model to minimize a two-sample test objective between S_1 and S_2

Two-Sample Test via a Discriminator

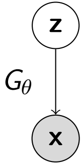
- Finding a two-sample test objective in high dimensions is hard



- In the generative model setup, we know that S_1 and S_2 come from different distributions p_{data} and p_θ respectively
- **Key idea: Learn** a statistic that **maximizes** a suitable notion of distance between the two sets of samples S_1 and S_2

Generative Adversarial Networks

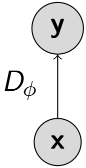
- A two player minimax game between a **generator** and a **discriminator**



- **Generator**
  - Directed, latent variable model with a deterministic mapping between $\mathbf{z}$ and $\mathbf{x}$ given by G_θ
  - Minimizes a two-sample test objective (in support of the null hypothesis $p_{\text{data}} = p_\theta$)

Generative Adversarial Networks

- A two player minimax game between a generator and a discriminator



- **Discriminator**
  - Any function (e.g., neural network) which tries to distinguish "real" samples from the dataset and "fake" samples generated from the model
  - Maximizes the two-sample test objective (in support of the alternate hypothesis $p_{\mathrm{data}} \neq p_\theta$)

Example of GAN objective

- **Training objective for discriminator**:

$$\max_{D} V(G, D) = E_{\mathbf{x} \sim p_{\mathrm{data}}}[\log D(\mathbf{x})] + E_{\mathbf{x} \sim p_G}[\log(1 - D(\mathbf{x}))]$$

- For a fixed generator G, the discriminator is performing binary classification with the cross entropy objective
  - Assign probability 1 to true data points $\mathbf{x} \sim p_{\mathrm{data}}$
  - Assing probability 0 to fake samples $\mathbf{x} \sim p_G$
- Optimal discriminator

$$D_G^*(\mathbf{x}) = \frac{p_{\mathrm{data}}(\mathbf{x})}{p_{\mathrm{data}}(\mathbf{x}) + p_G(\mathbf{x})}$$

Example of GAN objective

- **Training objective for generator**:

$$\min_G V(G, D) = E_{\mathbf{x} \sim p_{\text{data}}}[\log D(\mathbf{x})] + E_{\mathbf{x} \sim p_G}[\log(1 - D(\mathbf{x}))]$$

- For the optimal discriminator $D^*_G(\cdot)$, we have

$$\begin{aligned}
& V(G, D^*_G(\mathbf{x})) \\
&= E_{\mathbf{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}\right] + E_{\mathbf{x} \sim p_G}\left[\log \frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}\right] \\
&= E_{\mathbf{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{\frac{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}{2}}\right] + E_{\mathbf{x} \sim p_G}\left[\log \frac{p_G(\mathbf{x})}{\frac{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}{2}}\right] - \log 4 \\
&= \underbrace{D_{KL}\left[p_{\text{data}}, \frac{p_{\text{data}} + p_G}{2}\right] + D_{KL}\left[p_G, \frac{p_{\text{data}} + p_G}{2}\right]}_{2 \times \text{Jenson-Shannon Divergence (JSD)}} - \log 4 \\
&= 2 D_{JSD}[p_{\text{data}}, p_G] - \log 4
\end{aligned}$$

Jenson-Shannon Divergence

- Also called as the symmetric KL divergence

$$D_{JSD}[p, q] = \frac{1}{2}\left(D_{KL}\left[p, \frac{p+q}{2}\right] + D_{KL}\left[q, \frac{p+q}{2}\right]\right)$$

- Properties
  - $D_{JSD}[p, q] \geq 0$
  - $D_{JSD}[p, q] = 0$ iff $p = q$
  - $D_{JSD}[p, q] = D_{JSD}[q, p]$
  - $\sqrt{D_{JSD}[p, q]}$ satisfies triangle inequality $\rightarrow$ Jenson-Shannon Distance
- Optimal generator for the JSD/Negative Cross Entropy GAN

$$p_G = p_{\mathrm{data}}$$

- For the optimal discriminator $D^*_{G^*}(\cdot)$ and generator $G^*(\cdot)$, we have

$$V(G^*, D^*_{G^*}(\mathbf{x})) = -\log 4$$

The GAN training algorithm

- Sample minibatch of m training points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}$ from $\mathcal{D}$
- Sample minibatch of m noise vectors $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(m)}$ from p_z
- Update the generator parameters θ by stochastic gradient **descent**

$$\nabla_\theta V(G_\theta, D_\phi) = \frac{1}{m} \nabla_\theta \sum_{i=1}^{m} \log(1 - D_\phi(G_\theta(\mathbf{z}^{(i)})))$$

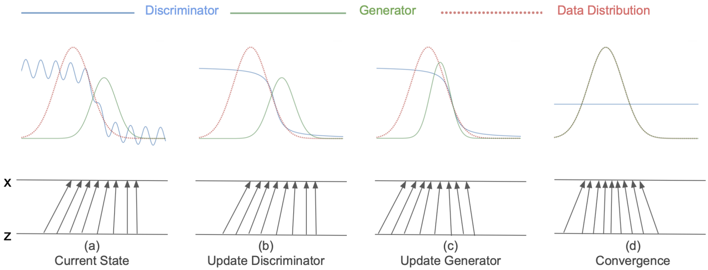
- Update the discriminator parameters ϕ by stochastic gradient **ascent**

$$\nabla_\phi V(G_\theta, D_\phi) = \frac{1}{m} \nabla_\phi \sum_{i=1}^{m} [\log D_\phi(\mathbf{x}^{(i)}) + \log(1 - D_\phi(G_\theta(\mathbf{z}^{(i)})))]$$

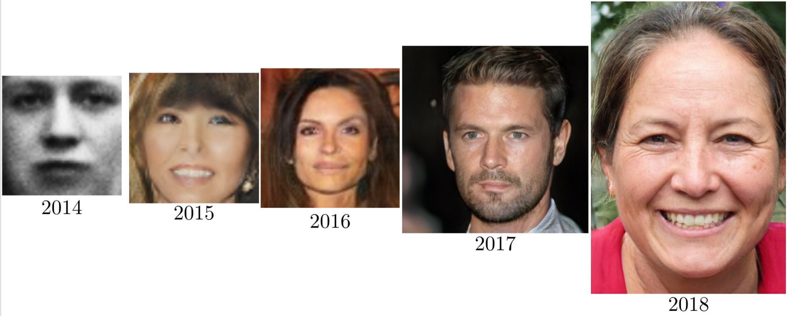
- Repeat for fixed number of epochs

Alternating optimization in GANs

$$\min_{\theta} \max_{\phi} V(G_\theta, D_\phi) = E_{\mathbf{x} \sim p_{\mathrm{data}}}[\log D_\phi(\mathbf{x})] + E_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - D_\phi(G_\theta(\mathbf{z})))]$$



Frontiers in GAN research

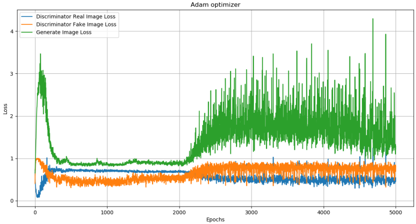


- GANs have been successfully applied to several domains and tasks
- However, working with GANs can be very challenging in practice
    - Unstable optimization
    - Mode collapse
    - Evaluation
- Many bag of tricks applied to train GANs successfully

Image Source: Ian Goodfellow. Samples from Goodfellow et al., 2014, Radford et al., 2015, Liu et al., 2016, Karras et al., 2017, Karras et al., 2018

Optimization challenges

- **Theorem (informal):** If the generator updates are made in function space and discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution
- **Unrealistic assumptions**!
- In practice, the generator and discriminator loss keeps oscillating during GAN training

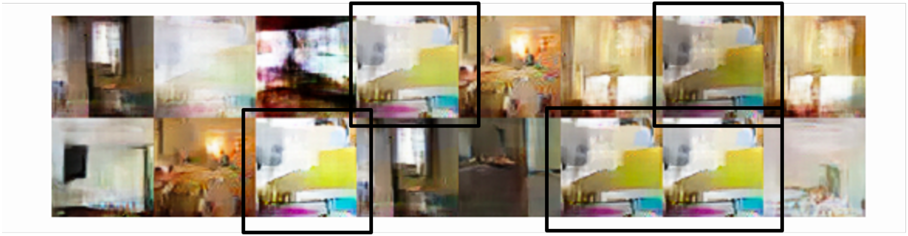


Source: Mirantha Jayathilaka

- No robust stopping criteria in practice (unlike MLE)

Mode Collapse

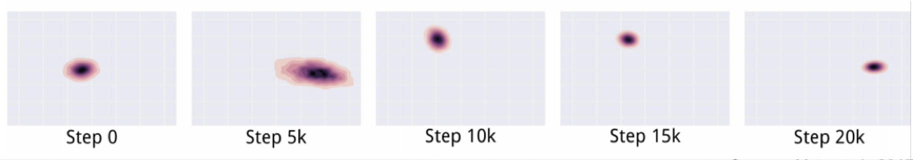
- GANs are notorious for suffering from **mode collapse**
- Intuitively, this refers to the phenomena where the generator of a GAN collapses to one or few samples (dubbed as "modes")



Arjovsky et al., 2017

Mode Collapse

- True distribution is a mixture of Gaussians



Source: Metz et al., 2017

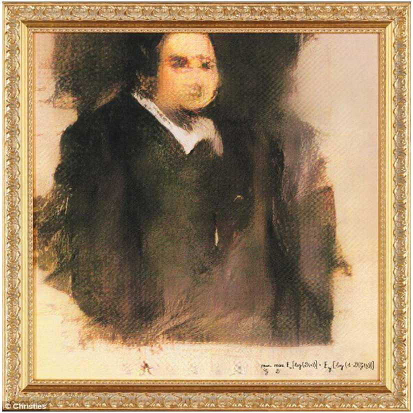
- The generator distribution keeps oscillating between different modes

Mode Collapse

Source: Metz et al., 2017

- Fixes to mode collapse are mostly empirically driven: alternate architectures, adding regularization terms, injecting small noise perturbations etc.
- `https://github.com/soumith/ganhacks`
  How to Train a GAN? Tips and tricks to make GANs work by Soumith Chintala

Beauty lies in the eyes of the discriminator



Source: Robbie Barrat, Obvious

GAN generated art auctioned at Christie's.
Expected Price: $\$7,000 - \$10,000$
True Price: $\$432,500$