Deep Generative Models

Chapter 4: Generative Adversarial Networks

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Summer 2025

Questioning Validity of GAN

+ *Though intuitive, is there any ways to show that this training approach is indeed leading to sampling from data distribution?*
– Well! We can claim this, if we can show the following

> Let $G_{\mathbf{w}^\star}$ and $D_{\phi^\star}$ be solutions to the min-max game; then, $x = G_{\mathbf{w}^\star}(z)$ for $z \sim Q(z)$ is *approximately* distributed by *data distribution*

To show this arguments let's consider the following idealistic assumptions

- *Discriminator D_ϕ can realize any function D exactly*
- *Generator $G_{\mathbf{w}}$ can realize any mapping G exactly*
  - ↳ *Discriminator and generator are infinitely-deep NNs*
  - ↳ *In practice, we can get approximately close with very deep NNs*
- *Dataset $\mathbb{D}$ has infinitely large number of samples*
  - ↳ *Our estimated average is accurate, i.e., $\hat{\mathbb{E}} \rightsquigarrow \mathbb{E}$*
  - ↳ *In practice, we can get approximately close with large batches of data*

Min-Max Game in *Idealistic Setting*

With an ideal setting: *the objective of the min-max problem becomes*

$$\mathcal{L}(G, D) = \mathbb{E}_{x \sim P} \{\log D(x)\} + \mathbb{E}_{z \sim Q} \{\log(1 - D(G(z)))\}$$

Since we assume G is any generator: we can say

$x = G(z)$ *can be distributed by any* $\hat{P}$ on $\mathbb{X}$

↳ *By changing G $\rightsquigarrow$ $\hat{P}$ is changed*[a]

[a] *Recall that $\hat{P}$ is not necessarily computable!*

With these definitions, we could say

$$\begin{aligned}\mathcal{L}(G, D) &= \mathbb{E}_{x \sim P} \{\log D(x)\} + \mathbb{E}_{x \sim \hat{P}} \{\log(1 - D(x))\} \\ &\equiv \mathcal{L}\left(\hat{P}, D\right)\end{aligned}$$

Min-Max Game in *Idealistic Setting*

Ideal GAN

In the ideal case with infinitely deep generator and discriminator, as well as infinitely large dataset, the GAN finds explicitly $G^\star$ and $D^\star$ as

$$G^\star, D^\star = \min_G \max_D \mathcal{L}\left(G, D\right)$$

which implicitly finds $\hat{P}^\star$ as

$$\hat{P}^\star, D^\star = \min_{\hat{P}} \max_D \mathcal{L}(\hat{P}, D)$$

where $\hat{P}^\star$ is the distribution of $\hat{x} = G^\star\left(z\right)$ with $z \sim Q$

+ *Does $\hat{P}^\star = P$ in this ideal case?*
  - ↳ *If so, then in practice GAN approximates data distribution, i.e., $P_{\mathbf{w}} \approx P$*

Optimal Players: *Discriminator*

Let's start with finding *theoretically-optimal generator and discriminator: we could expand the objective as*

$$\begin{aligned}\mathcal{L}\left(\hat{P}, D\right) &= \mathbb{E}_{x\sim P}\left\{\log D\left(x\right)\right\} + \mathbb{E}_{x\sim \hat{P}}\left\{\log\left(1 - D\left(x\right)\right)\right\} \\ &= \int_{\mathbb{X}} \log D\left(x\right) P\left(x\right) \mathrm{d}x + \int_{\mathbb{X}} \log\left(1 - D\left(x\right)\right) \hat{P}\left(x\right) \mathrm{d}x \\ &= \int_{\mathbb{X}} \log D\left(x\right) P\left(x\right) + \log\left(1 - D\left(x\right)\right) \hat{P}\left(x\right) \mathrm{d}x\end{aligned}$$

The min-max game first maximizes $\mathcal{L}$ over D: we can find $D^\star$ by setting

$$\frac{\mathrm{d}}{\mathrm{d}D}\mathcal{L}\left(\hat{P}, D\right) \overset{!}{=} 0$$

+ *How can we compute derivative with respect to a function?!*
– *We should use Radon-Nikodym derivative, but let us do it naively!*

Optimal Discriminator: *Naive Analysis*

! Attention

Following analysis is super-simplified and not accurate, but reflects key idea

Assume a tiny vicinity $V(x) \subset \mathbb{X}$ around $x \in \mathbb{R}^d$, where

↳ P, $\hat{P}$ *and* D *are roughly constant*

we can then decompose the objective as

$$\mathcal{L}\left(\hat{P}, D\right) = |V(x)| \left[\log D(x) P(x) \log(1 - D(x)) \hat{P}(x)\right] + \mathcal{R}$$

where $\mathcal{R}$ is the integral over the remaining part of the data space,i.e.,

$$\mathcal{R} = \int_{\mathbb{X} - V(x)} \log D(u) P(u) + \log(1 - D(u)) \hat{P}(u) \, \mathrm{d}u$$

not depending on $D(x)$

Optimal Discriminator: *Naive Analysis*

We want to know the optimal discriminator $D^\star$

↳ *at any x, we look for optimal value $\mu^\star = D^\star(x)$ that maximizes objective*

↳ *This is like thinking of $D(x)$ as an infinitely-large vector*

  ↳ *$D^\star$ is then the optimal vector*

By this intuitive formulation, we could say

$$\begin{aligned}\mathcal{L}\left(\hat{P}, D\right) &= |V(x)| \left[\log D(x) P(x) \log(1 - D(x)) \hat{P}(x)\right] + \mathcal{R} \\ &= |V(x)| \left[\log \mu P(x) \log(1 - \mu) \hat{P}(x)\right] + \mathcal{R}\end{aligned}$$

and $\mu^\star$ is found by setting

$$\frac{\mathrm{d}}{\mathrm{d}\mu} \mathcal{L}\left(\hat{P}, D\right) = 0$$

Optimal Discriminator: *Naive Analysis*

Since μ shows up only in the first expression, we can write

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\mu}\mathcal{L}\left(\hat{P}, D\right) &= |V(x)| \frac{\mathrm{d}}{\mathrm{d}\mu}\left[\log \mu P(x) \log(1-\mu)\hat{P}(x)\right] \\ &= |V(x)| \left[\frac{P(x)}{\mu} - \frac{\hat{P}(x)}{1-\mu}\right]\end{aligned}$$

which setting to zero concludes

$$\mu^\star = D^\star(x) = \frac{P(x)}{P(x) + \hat{P}(x)}$$

Optimal Discriminator: *Naive Analysis*

If we do this analysis more concretely using Radon-Nikodym derivative: we can conclude that

$$D^\star(x) = \frac{P(x)}{P(x) + \hat{P}(x)}$$

is the optimal discriminator that maximizes the objective

Attention

Note that $D^\star(x)$ is not computable!

- ↳ *We have no access to the data distribution*
- ↳ *We cannot compute explicitly $\hat{P}(x)$*

But, if the discriminator model is deep enough and we have a large number of samples $\rightsquigarrow$ we can approximate it by a computational model D_ϕ

Optimal Generator

To find optimal generator: *we can use $D^\star$ and solve outer minimization, i.e.,*

$$\min_G \max_D \mathcal{L}(G, D) = \min_G \mathcal{L}(G, D^\star)$$

This means that the is the solution to

$$\min_G \mathbb{E}_{x\sim P}\left\{\log \frac{P(x)}{P(x) + \hat{P}(x)}\right\} + \mathbb{E}_{x\sim \hat{P}}\left\{\log\left(1 - \frac{P(x)}{P(x) + \hat{P}(x)}\right)\right\}$$

We can alternatively say: distribution of $\hat{x} = G^\star(z)$ is

$$\hat{P}^\star = \underset{\hat{P}}{\operatorname{argmin}}\ \mathbb{E}_{x\sim P}\left\{\log \frac{P(x)}{P(x) + \hat{P}(x)}\right\} + \mathbb{E}_{x\sim \hat{P}}\left\{\log \frac{\hat{P}(x)}{P(x) + \hat{P}(x)}\right\}$$

Optimal Generator: *Divergence Minimizer*

Average Dist

Let us define the average model and data distribution as

$$A_{P,\hat{P}}(x) = 0.5 \left(P(x) + \hat{P}(x) \right)$$

which is still a distribution defined on $\mathbb{X}$

With this definition: *we could say*

$$\begin{aligned}
\hat{P}^\star &= \operatorname*{argmin}_{\hat{P}} \mathbb{E}_{x\sim P} \left\{ \log \frac{P(x)}{2A_{P,\hat{P}}(x)} \right\} + \mathbb{E}_{x\sim \hat{P}} \left\{ \log \frac{\hat{P}(x)}{2A_{P,\hat{P}}(x)} \right\} \\
&= \operatorname*{argmin}_{\hat{P}} \mathbb{E}_{x\sim P} \left\{ \log \frac{P(x)}{A_{P,\hat{P}}(x)} \right\} + \mathbb{E}_{x\sim \hat{P}} \left\{ \log \frac{\hat{P}(x)}{A_{P,\hat{P}}(x)} \right\} - 2\log 2
\end{aligned}$$

Optimal Generator: *Divergence Minimizer*

Recall: *KL Divergence*

The KL divergence between P and Q is defined as

$$D_{\mathrm{KL}}\left(P \| Q\right) = \mathbb{E}_{x \sim P}\left\{\log \frac{P\left(x\right)}{Q\left(x\right)}\right\}$$

We can thus say that

$$\begin{aligned}\hat{P}^\star &= \operatorname*{argmin}_{\hat{P}} \mathbb{E}_{x \sim P}\left\{\log \frac{P\left(x\right)}{A_{P,\hat{P}}\left(x\right)}\right\} + \mathbb{E}_{x \sim \hat{P}}\left\{\log \frac{\hat{P}\left(x\right)}{A_{P,\hat{P}}\left(x\right)}\right\} - 2\log 2 \\ &= \operatorname*{argmin}_{\hat{P}} \boxed{D_{\mathrm{KL}}\left(P \| A_{P,\hat{P}}\right) + D_{\mathrm{KL}}\left(\hat{P} \| A_{P,\hat{P}}\right)}\end{aligned}$$

divergence metric

Jensen-Shannon Divergence

Jensen-Shannon Divergence

The Jensen-Shannon divergence between P and $\hat{P}$ is defined as

$$D_{\text{JS}}\left(P \| \hat{P}\right) = D_{\text{KL}}\left(P \| A_{P,\hat{P}}\right) + D_{\text{KL}}\left(\hat{P} \| A_{P,\hat{P}}\right)$$

In Assignment 2: *we play around with JS divergence to see the following*

KL Divergence vs JS Divergence

Under some regularity conditions on P and $\hat{P}$, we can say that

$$D_{\text{JS}}\left(P \| \hat{P}\right) = 0 \leftrightsquigarrow D_{\text{KL}}\left(P \| \hat{P}\right) = 0 \leftrightsquigarrow P = \hat{P}$$

Optimal Generator: *Divergence Minimizer*

Assuming those mild conditions fulfilled: *we can say*

$$\hat{P}^\star = \operatorname*{argmin}_{\hat{P}} D_{\mathrm{JS}} \left(P \| \hat{P} \right) \leftrightsquigarrow \hat{P}^\star = \operatorname*{argmin}_{\hat{P}} D_{\mathrm{KL}} \left(P \| \hat{P} \right)$$

In other words, as we train using min-max game

distribution of generated sample $x = G(z)$ tends to data distribution

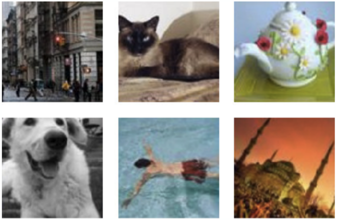
Moral of Story: *Implicit MLE via GAN*

Using min-max training strategy of GAN we implicitly train a generator with minimal KL divergence $\equiv$ maximal likelihood at its output

Sample Outputs of GAN

Trained on the dataset

GAN is able to sample



Vanilla GAN: *Stability Challenge*

Vanilla GAN was one of first models that could generate realistic samples: *it however suffers from various issues*

- *Training loop is unstable*
  - ↳ *Since we use gradient descent for min-max it could easily diverge*
  - ↳ *It is very sensitive to hyperparameters*
- *Mode collapse could happen*
  - ↳ *The generator gives a few good samples all the time to fool the discriminator*
  - ↳ *This results in lack of diversity*
  - ↳ *GAN learns distribution in a very tiny part of the data manifold*
- *Computational vanilla GANs are vulnerable to vanishing gradient*
  - ↳ *Objective function returns small gradients*

To overcome these issues, Wasserstein GAN was proposed which

replaces the JS divergence with the Wasserstein distance

We learn Wasserstein GANs next!