# 1 Outline

In this lecture, we cover

- Wasserstein Generative Adversarial Networks,
- adversarial training.
- sharpness-aware minimization.

# 2 Review of Generative Adversarial Network (GAN) and $f$-GAN

In the last lecture, we studied **Generative Adversarial Networks (GANs)** [GPAM+14]. Recall that a GAN consists of the generator network $G$ and the discriminator network $D$. First, the generator $G$ receives a random vector $z \in \mathbb{R}^k$ drawn from a fixed distribution $\gamma$ such as the standard Guassian distribution $N(0, I_k)$. Then it produces a sample

$$x = G(z).$$

Note that the generator network corresponds to the distribution $\nu$ with the density function

$$q(x) = \gamma\left(G^{-1}(x)\right).$$

Hence, the generator network produces a sample from the distribution $\nu$ with density function $q$ without explicitly modeling $q$. Then, after the generator $G$ produces a sample $x = G(z)$, the discriminator $D$ classifies whether $x$ comes from the training data set or were given by $G$. The way the discriminator $D$ works is that it takes a sample $x$ and returns

$$D(x) \in [0, 1]$$

that measures the probability of $x$ coming from the true data. Remember that we use the following loss function to train a GAN.

$$V(G, D) = \mathbb{E}_{x\sim\mu}\left[\log D(x)\right] + \mathbb{E}_{z\sim\gamma}\left[\log(1 - D(G(z)))\right].$$

For a fixed generator $G$, maximizing $V(G, D)$ with respect to $D$ requires making $D(x)$ high for $x$ coming from the true data while making $D(G(z))$ small for $z \sim \gamma$. Hence, by considering

$$\max_{D} \quad V(G, D),$$

one can train the discriminator so that it can distinguish the generated samples from the true data. In response to the discriminator, the generator solves

$$\min_{G}\max_{D} \quad V(G, D).$$

As a result, the generator attempts to fool the discriminator by making it end up assigning a high value to $G(z)$.

We also derived the following equivalent representations of the loss function $V(G, D)$.

$$\begin{aligned} V(G,D) &= \mathbb{E}_{x\sim\mu}\left[\log D(x)\right] + \mathbb{E}_{z\sim\gamma}\left[\log(1-D(G(z)))\right] \\ &= \mathbb{E}_{x\sim\mu}\left[\log D(x)\right] + \mathbb{E}_{x\sim\nu}\left[\log(1-D(x))\right] \\ &= \int_{\mathbb{R}^d} \log D(x) d\mu(x) + \int_{\mathbb{R}^d} \log(1-D(x)) d\nu(x) \\ &= \int_{\mathbb{R}^d} \left(p(x)\log D(x) + q(x)\log(1-D(x))\right) dx. \end{aligned}$$

Defining the Kullback-Leibler(KL) divergence of two distributions $p$ and $q$ as

$$D_{KL}(p||q) = \int_{\mathbb{R}^d} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

and the Jensen-Shannon divergence of $p$ and $q$ as

$$D_{JS}(p||q) = \frac{1}{2} D_{KL}\left(p \middle|\middle| \frac{p+q}{2}\right) + \frac{1}{2} D_{KL}\left(q \middle|\middle| \frac{p+q}{2}\right),$$

we proved that

$$\min_G \max_D \quad V(G,D) \quad = \quad \min_p \quad 2D_{JS}(p||q) - \log 4.$$

Here, the Jensen-Shannon divergence can be generalized to the so-called $f$-divergence. The $f$-divergence of two distributions $p$ and $q$ is defined as

$$D_f(p||q) = \int_{\mathbb{R}^d} q(x) f\left(\frac{p(x)}{q(x)}\right) dx.$$

With this, one may consider a generalization of the GAN framework by taking

$$\min_q \quad D_f(p||q).$$

This is the $f$-GAN framework proposed by Nowozin et al. [NCT16].

## 3 Wasserstein GAN

The **Wasserstein GAN** [ACB17] is another variant of GAN, extending the idea of $f$-GAN. The important component of $f$-GAN is that the generator attempts to mimick the true distribution $p$ by minimizing the $f$-divergence of $p$ and $q$. The Wasserstein GAN is basically that the $f$-divergence is replaced by the so-called **Wasserstein distance**. For $p \geq 1$, the $p$-Wasserstein distance between two distributions $p$ and $q$ with respect to a norm $\|\cdot\|$ is defined as

$$W(p,q) := \inf_{\Pi} \left\{ \left(\mathbb{E}_{(\xi,\xi')\sim\Pi}\left[\left\|\xi-\xi'\right\|^p\right]\right)^{\frac{1}{p}} : \Pi \text{ has marginal distributions } p, q \right\}.$$

What is commonly used in practice is the 1-Wasserstein distance with respect to the $\ell_2$-norm, given by

$$W(p,q) := \inf_{\Pi} \left\{ \mathbb{E}_{(\xi,\xi')\sim\Pi}\left[\left\|\xi-\xi'\right\|_2\right] : \Pi \text{ has marginal distributions } p, q \right\}.$$

Throughout the section, we stick to the 1-Wasserstein distance with respect to the $\ell_2$-norm, and we refer to it simply by the Wasserstein distance. The Wasserstein distance is also called the **earth mover's distance**, as it can be interpreted as the minimum transportation cost to move some

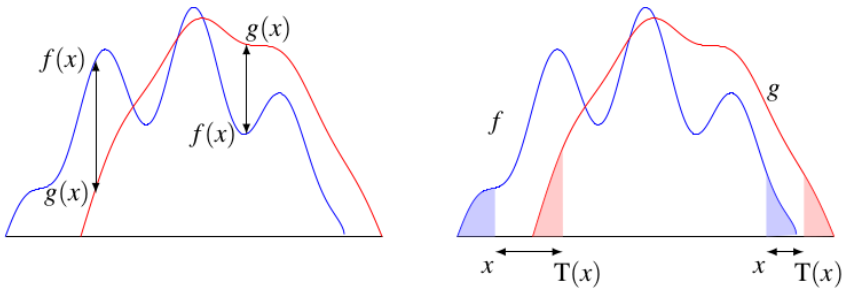


Figure 19.1: Illustrating the Wasserstein distance as optimal transport

probability mass (dirt) from one distribution to form the other distribution. The Wasserstein GAN is simply the GAN framework trained by

$$\min_q \quad W(p,q).$$

By the Kantorovich-Rubinstein duality theorem, the Wasserstein distance can be rewritten as follows.

$$W(p,q) = \sup\left\{\mathbb{E}_{x\sim p}\left[h(x)\right] - \mathbb{E}_{x\sim q}\left[h(x)\right] : \ h \text{ is 1-Lipschitz continuous}\right\}.$$

In fact, we have

$$\sup\left\{\mathbb{E}_{x\sim p}\left[h(x)\right] - \mathbb{E}_{x\sim q}\left[h(x)\right] : \ h \text{ is } L\text{-Lipschitz continuous}\right\} = L\cdot W(p,q).$$

Then we may consider

$$\min_q \max_{h:\text{Lipschitz}} \quad \mathbb{E}_{x\sim p}\left[h(x)\right] - \mathbb{E}_{x\sim q}\left[h(x)\right].$$

One may take a neural network parameterized by $\omega$ for $h$. Then

$$\min_q \max_\omega \quad \mathbb{E}_{x\sim p}\left[h_\omega(x)\right] - \mathbb{E}_{x\sim q}\left[h_\omega(x)\right].$$

As the distribution $q$ is obtained based on the generator network $G_\theta$ parameterized by $\theta$, we get

$$\min_\theta \max_\omega \quad \mathbb{E}_{x\sim p}\left[h_\omega(x)\right] - \mathbb{E}_{z\sim\gamma}\left[h_\omega(G_\theta(z))\right].$$

# 4 Adversarial Training

Consider a set of data $(x_1,y_1),\ldots,(x_n,y_n)$ where $x_i$ denotes the *feature* and $y_i$ is the corresponding *label*. We are given a model $f_\theta$ parametrized by $\theta$ that receives a feature vector $x$ and predicts its label as $f_\theta(x)$. For the set of $n$ data, we can consider the mean squared error given by

$$\frac{1}{n}\sum_{i=1}^n \left(y_i - f_\theta(x_i)\right)^2.$$

For a general loss function $\ell$, we may also consider

$$\frac{1}{n}\sum_{i=1}^n \ell\left(f_\theta(x_i), y_i\right).$$

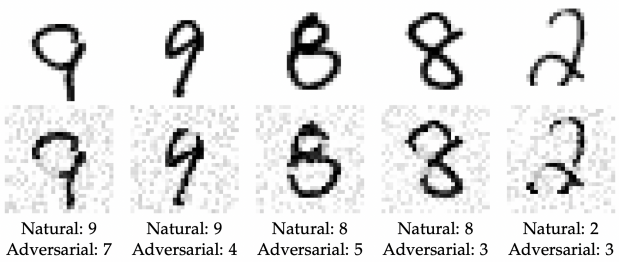


Figure 19.2: Adversarial examples after data perturbations

In this section, we consider the **adversarial training** framework which aims to train the model so that it is robust to some noise and perturbations present in data. Given a data point $(x, y)$, one may create an **adversarial example** by solving

$$\max \left\{ \ell \left( f_\theta(x+\delta), y \right) : \ \|\delta\| \leq \epsilon \right\}.$$

Here, the feature vector $x$ is corrupted by some small noise $\delta$, so we would still predict the same label $y$ with $x+\delta$. However, the noise term $\delta$ is chosen so that the loss associated with the current model $f_\theta$ is maximized.

Let us consider the formulation with the $\ell_\infty$-norm:

$$\max \left\{ \ell \left( f_\theta(x+\delta), y \right) : \ \|\delta\|_\infty \leq \epsilon \right\}.$$

The **fast gradient sign method (FGSM)** [GSS15] sets

$$\delta = \epsilon \cdot \operatorname{sign} \left( \nabla_x \ell \left( f_\theta(x), y \right) \right) = \epsilon \cdot \operatorname{sign} \left( \nabla_x f_\theta(x) \nabla_{f_\theta} \ell \left( f_\theta(x), y \right) \right).$$

Although the FGSM provides a heuristic solution, it is widely used in practice to create adversarial


Figure 19.3: Adversarial examples by the FGSM

examples. One may attempt to directly solving the problem by considering the following composite optimization formulation.

$$\max_{\delta} \quad f(\delta) + g(\delta)$$

where

$$f(\delta) = \ell \left( f_\theta(x), y \right) \quad \text{and} \quad g(\delta) = \mathbf{1} \left( \|\delta\|_\infty \leq \epsilon \right).$$

A natural approach is to apply the **proximal gradient ascent (PGA)**, which proceeds with

$$\delta_{t+1} = \mathrm{prox}_{\eta g}\left(\delta_t + \eta \nabla f(\delta_t)\right).$$

Here, the proximal operator is given by

$$\left(\mathrm{prox}_{\eta g}(\delta)\right)_i = \begin{cases} \epsilon, & \text{if } \delta_i \geq \epsilon, \\ \delta_i, & \text{if } -\epsilon \leq \delta_i < \epsilon, \\ -\epsilon, & \text{if } \delta_i < -\epsilon. \end{cases}.$$

Indeed, adversarial perturbations alter the outcomes of a model drastically. Then the question is about how to train a model that is robust to adversarial perturbations. Given the set of $n$ data $(x_1, y_1), \ldots, (x_n, y_n)$, we discussed the loss minimization given by

$$\min_{\theta} \quad \frac{1}{n}\sum_{i=1}^{n} \ell\left(f_\theta(x_i), y_i\right).$$

Taking adversarial perturbations into account, one may consider

$$\min_{\theta} \quad \frac{1}{n}\sum_{i=1}^{n} \max\left\{\ell\left(f_\theta(x_i+\delta), y_i\right) : \|\delta\|_\infty \leq \epsilon\right\}.$$

To optimization this formulation by a gradient-based method, one needs to compute

$$\nabla_\theta \left(\max\left\{\ell\left(f_\theta(x_i+\delta), y_i\right) : \|\delta\|_\infty \leq \epsilon\right\}\right).$$

In general, the function

$$\max\left\{\ell\left(f_\theta(x_i+\delta), y_i\right) : \|\delta\|_\infty \leq \epsilon\right\}$$

is not differentiable with respect to $\theta$. Nevertheless, we may take

$$\delta^* \in \mathrm{argmax}\left\{\ell\left(f_\theta(x_i+\delta), y_i\right) : \|\delta\|_\infty \leq \epsilon\right\}$$

and use the gradient with respect to $\delta = \delta^*$:

$$\nabla_\theta \ell\left(f_\theta(x_i+\delta^*), y_i\right).$$

## 5 Sharpness-Aware Minimization

The next question that we consider is about generalization. It has been observed that flat minima tend to generalize better than sharp minima. Rather than finding a single $\theta$ that has a low loss, we look for $\theta$ such that parameter $\theta'$ contained in the neighborhood of $\theta$ uniformly has a low loss value. To achieve this goal, we consider the following formulation.

$$\min_{\theta} \quad \frac{1}{n}\sum_{i=1}^{n} \max_{\delta:\|\delta\|_2\leq\epsilon} \quad \ell\left(f_{\theta+\delta}(x_i), y_i\right).$$

This framework is **sharpness-aware minimization (SAM)** [FKMN21]. Here, the inner maximization can be approximated using the first-order Taylor approximation.

$$\max_{\delta:\|\delta\|_2\leq\epsilon} \ell\left(f_{\theta+\delta}(x_i), y_i\right) \simeq \max_{\delta:\|\delta\|_2\leq\epsilon} \left\{\ell\left(f_\theta(x_i), y_i\right) + \delta^\top \nabla_\theta \ell\left(f_\theta(x_i), y_i\right)\right\}.$$

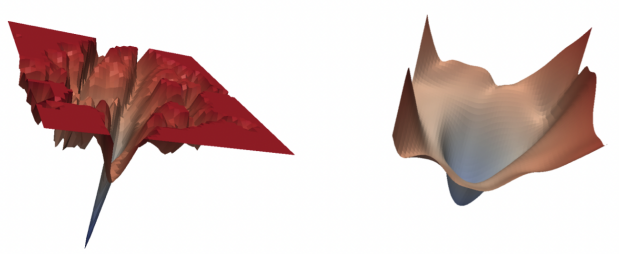

Figure 19.4: Sharp and flat minima

The maximizer of the approximation is simply given by

$$\delta^* = \frac{\epsilon}{\|\nabla_\theta \ell\left(f_\theta(x_i), y_i\right)\|_2} \nabla_\theta \ell\left(f_\theta(x_i), y_i\right).$$

Then we substitute $\delta = \delta^*$ and consider $f_{\theta+\delta^*}$:

$$\ell\left(f_{\theta+\delta^*}(x_i), y_i\right).$$

Here,

$$\nabla_\theta\left(\ell\left(f_{\theta+\delta^*}(x_i), y_i\right)\right) = \left(1 + \frac{d\delta^*}{d\theta}\right) \nabla_\theta \ell\left(f_{\theta+\delta^*}(x_i), y_i\right) \simeq \nabla_\theta \ell\left(f_{\theta+\delta^*}(x_i), y_i\right).$$

## References


[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. 3

[FKMN21] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 5

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[NCT16] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 2