DEEP LEARNING

Generative adversarial networks

Святослав Елизаров, Борис Коваленко, Артем Грачев

2 июня 2018

Высшая школа экономики

ОЦЕНКА ПЛОТНОСТИ РАСПРЕДЕЛЕНИЯ

Рис. 1: Оценка плотности распределения на основе данных

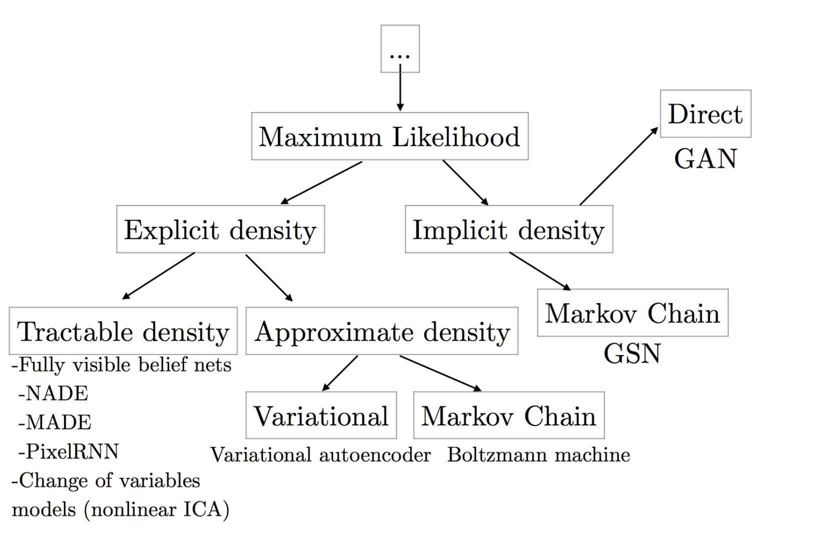
ОЦЕНКА ПЛОТНОСТИ РАСПРЕДЕЛЕНИЯ

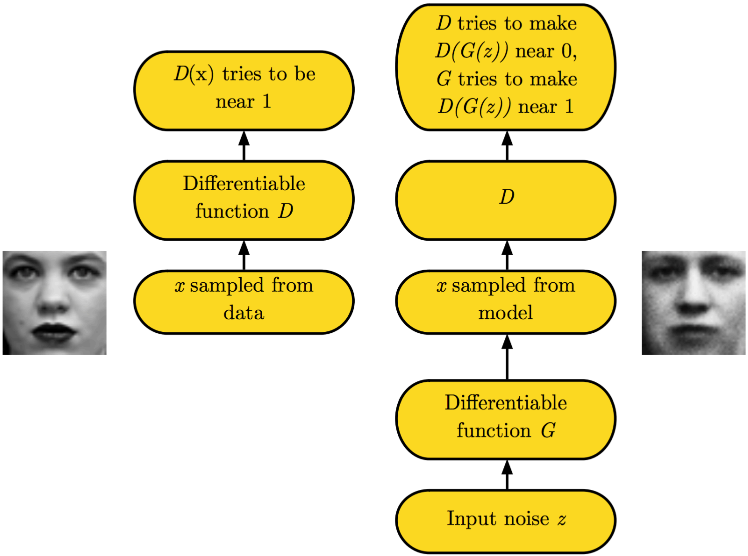
$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right) \\ &= \arg\max_{\boldsymbol{\theta}} \log \prod_{i=1}^{m} p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right).\end{aligned}$$

Максимизация log-likelihood эквивалентна минимизации KL-дивергенции между распределениями $\hat{p}_{data}$ и p_{model}

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} D_{\text{KL}}\left(p_{\text{data}}(\boldsymbol{x}) \| p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})\right).$$

ОЦЕНКА ПЛОТНОСТИ РАСПРЕДЕЛЕНИЯ




D(x) tries to be near 1
Differentiable function D
x sampled from data
D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1
D
x sampled from model
Differentiable function G
Input noise z

Игра с 0 суммой:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Для дискриминатора оптимизируем функционал с помощью градиентного подъема:

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Для генератора оптимизируем функционал с помощью градиентного спуска:

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

При фиксированном генераторе, оптимальные значения достигаются, если дискриминатор имеет вид:

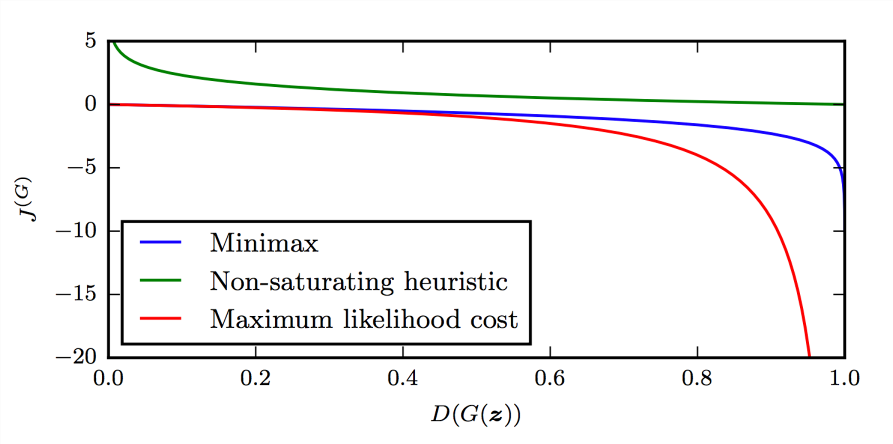
$$D^* = \frac{\mathbb{P}}{\mathbb{P} + \mathbb{Q}}$$

Где $\mathbb{Q}$ – восстановленное распределение

Оптимизация такого min-max функционала в пространстве функций эквивалетно минимизации дивергенции Йенсена-Шеннона между истинным распределением и восстановленным распределением.

$$JS(\mathbb{P}, \mathbb{Q}) = KL(\mathbb{P} \| \frac{\mathbb{P} + \mathbb{Q}}{2}) + KL(\mathbb{Q} \| \frac{\mathbb{P} + \mathbb{Q}}{2})$$

Generative Adversarial Networks Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio


5
0
−5
−10
−15
−20
$J(G)$
0.0
0.2
0.4
0.6
0.8
1.0
$D(G(z))$
Minimax
Non-saturating heuristic
Maximum likelihood cost

Игра с 0 суммой:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Для дискриминатора оптимизируем функционал с помощью градиентного подъема:

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Для генератора оптимизируем функционал с помощью градиентного ~~спуска~~ подъема:

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Общее описание алгоритма:

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:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

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 ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Generative Adversarial Networks Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio

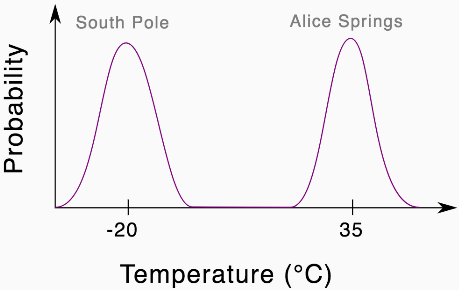
Проблемы GAN:

1. Сходимость алгоритма гарантирована только если мы работаем в пространстве функций. На практике функции представлены нейронными сетями, т.е. мы находимся в пространстве параметров.
2. Осциляции функций потерь при обучении
3. Коллапс моды
4. И др.

Немного о коллапсе моды:

Немного о коллапсе моды:

Немного о коллапсе моды:



Пример задачки с коллапсом моды

Немного о коллапсе моды:

1. Генератор понимает, что если будет выдавать только температуру с Южного Полюса, то дискриминатор ничего не поймет
2. Дискриминатор понимает что его обманывает и начинает подбрасывать монетку, чтобы угадать температура реальная или нет
3. Генератор понимает, что дискриминатор понял что погода с Южного Полюса фейковая и наченает генерировать погоду только из Австралии
4. Дискриминатор снова понял что его обманывают, но теперь фейки идут в Австралийской погоде и он начинает угадывать их там
5. См. пункт 1

Как эмпирически понять, что коллапс моды состоялся?

Как эмпирически понять, что коллапс моды состоялся?

Парадокс дня рождения. Есть суппорт распределения N, то в семпле размера $\sqrt{N}$ наверняка будет дубликат.

Как эмпирически понять, что коллапс моды состоялся?

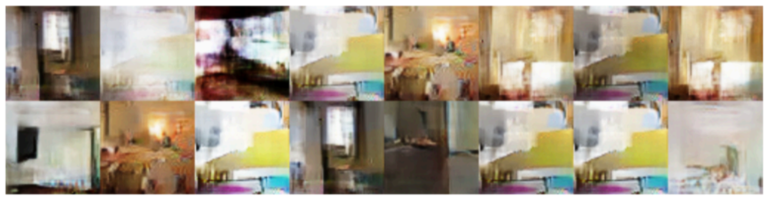
Парадокс дня рождения. Есть суппорт распределения N, то в семпле размера $\sqrt{N}$ наверняка будет дубликат.

1. Сгенерировать семпл размера s
2. Найти дубликаты в семле
3. Повторить до сходимости вероятности дубликата в семпле размера s

Если вероятность дубликата в семпле размера s велика, то суппорт примерно равен s^2

Do GANs actually learn the distribution? An empirical study / Sanjeev Arora, Yi Zhang

Пример колласпа моды для картинок:



Wasserstein GAN / Martin Arjovsky, Soumith Chintala, Léon Bottou

$$\mathbb{E}_{z \sim p(z)} - \log(D_w(G_\theta(z)))$$

Для фиксированного оптимального дискриминатора функция равна:

$$KL(\mathbb{Q} \| \mathbb{P}) - 2JS(\mathbb{P}, \mathbb{Q})$$

Какие выводы мы можем сделать?

Как преодолеть эти проблемы?

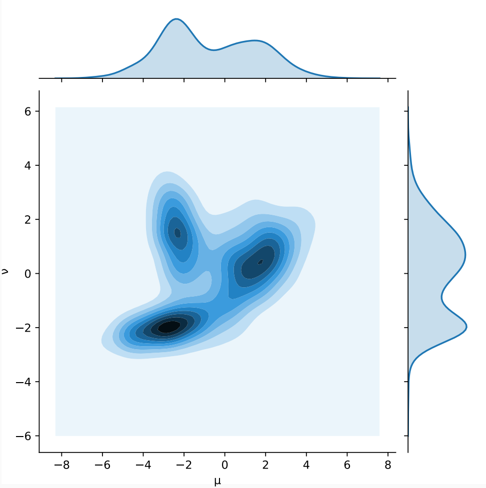
Введем следующую меру близости между распределениями

$$W(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \prod(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

Где $\prod(\mathbb{P}, \mathbb{Q})$ соответсвует множеству всех возможных совместных распределений $\gamma(x, y)$, таких что маргинальными распределениями для них являются $\mathbb{P}, \mathbb{Q}$

Другими словами, $\gamma(x, y)$ показывает как много вероятности надо перенести из x в y чтобы преобразовать $\mathbb{P}$ в $\mathbb{Q}$.

$W(\mathbb{P}, \mathbb{Q})$ задаёт оптимальный план переноса.



Транспортная плоскость

Чем эта мера лучше остальных?

Рассмотрим следующие меры:

1. $KL(\mathbb{P} \| \mathbb{Q}) = \sum\limits_{x \sim \mathbb{P}} \mathbb{P}(x) \log \frac{\mathbb{P}(x)}{\mathbb{Q}(x)}$
2. $JS(\mathbb{P}, \mathbb{Q}) = KL(\mathbb{P} \| \frac{\mathbb{P}+\mathbb{Q}}{2}) + KL(\mathbb{Q} \| \frac{\mathbb{P}+\mathbb{Q}}{2})$
3. $\delta(\mathbb{P}, \mathbb{Q}) = \sup\limits_{A \in \Sigma} |\mathbb{P}(A) - \mathbb{Q}(A)|$
4. $W(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \prod(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(x,y) \sim \gamma} \|x - y\|$

Пусть дана случайная величина распределенная равномерно:

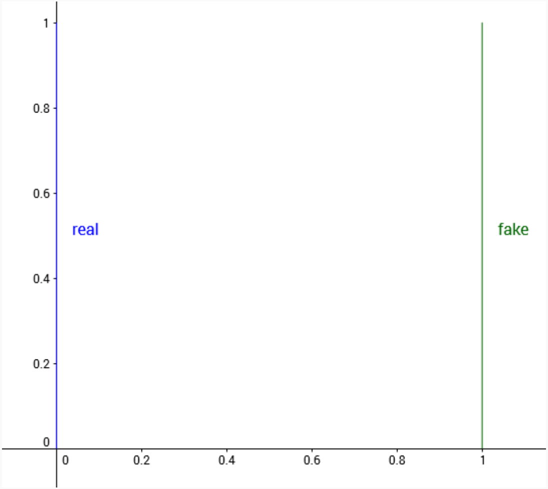
$$Z \sim \mathcal{U}[0, 1]$$

Пусть распределение $\mathbb{P}$ определено на $\mathbb{R}^2$ следующим образом:

$$\mathbb{P} = (0, Z)$$

Теперь зададим параметрическое семейство $\mathbb{Q}$ вида:

$$\mathbb{Q}_\theta = (\theta, Z)$$

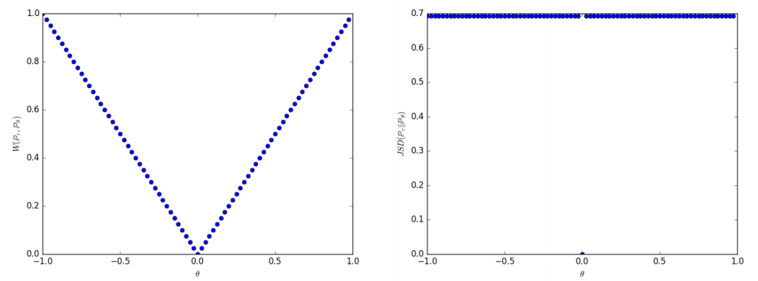


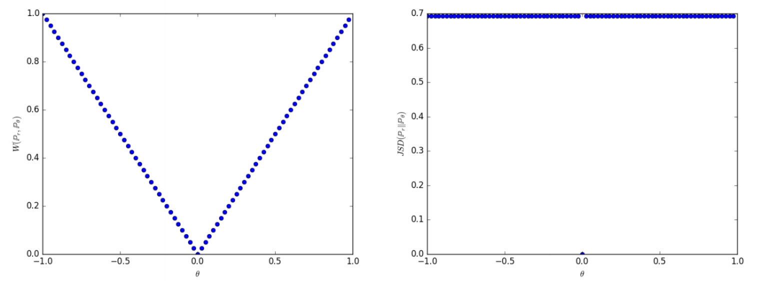
1
0.8
0.6
0.4
0.2
0
real
fake
0
0.2
0.4
0.6
0.8
1

Какие значения примут каждое из расстояний?

Какие значения примут каждое из расстояний?

1. $W(\mathbb{P}, \mathbb{Q}) = |\theta|$
2. $JS(\mathbb{P}, \mathbb{Q}) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{otherwise} \end{cases}$
3. $KL(\mathbb{P} \| \mathbb{Q}) = KL(\mathbb{Q} \| \mathbb{P}) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{otherwise} \end{cases}$
4. $\delta(\mathbb{P}, \mathbb{Q}) = \begin{cases} 1 & \text{if } \theta \neq 0 \\ 0 & \text{otherwise} \end{cases}$


$W(\mathbb{P}_r, \mathbb{P}_\theta)$
θ
$JSD(\mathbb{P}_r \| \mathbb{P}_\theta)$
θ



Что будет с градиентом?

- Существует бесконечное количество транспортных плоскостей, как найти оптимальную?
- Как применять расстояние Вассершнейна для тренировки GAN?
- Как гарантировать что градиент будет существовать?

Пусть $\mathbb{P}$ задаёт истинное (зафиксированное) распределение на пространстве $\mathcal{X}$.

Z – некоторая случайная величина (например $Z \sim \mathcal{N}(0, I)$) на некотором другом пространстве $\mathcal{Z}$

Зададим функцию

$$g : \mathcal{Z} \times \mathcal{R}^d \to \mathcal{X}$$

Будем обозначать её $g_\theta(z)$.

Пусть теперь распределение $\mathbb{Q}_\theta$ соотвествует распределению $g_\theta(Z)$.

Теорема:

1. Если g непрерывно по θ, то и $W(\mathbb{P}, \mathbb{Q}_\theta)$ тоже
2. Если g локально Липшицева функция, то $W(\mathbb{P}, \mathbb{Q}_\theta)$ непрерывна везде и **дифференцируема почти всюду**
3. Эти свойства не выполняются для KL и JS (см. пример).

Wasserstein GAN Martin Arjovsky, et al.

Остановимся подробнее на втором пункте.

Функция $f : \mathbb{R} \to \mathbb{R}$ называется *Липшицевой*, если:

$$\exists K, \forall x_1, x_2 \in \mathbb{R}, |f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

Назовите примеры таких функций!

Остановимся подробнее на втором пункте.

Функция $f : \mathbb{R} \to \mathbb{R}$ называется Липшицевой, если:

$$\exists K, \forall x_1, x_2 \in \mathbb{R}, |f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

Назовите примеры таких функций!

Свойства:

1. Равномерная непрерывность (следует из определения)
2. Функция дифференцируема почти всюду (теорема Радемахера)
3. Функция является Липшицевой тогда и только тогда, когда **её производная (градиент) ограничена (константой Липшица)**. Докажите это используя теорему о среднем!

Более общо: Пусть даны метрические пространства $(\mathcal{X}, d_{\mathcal{X}})$ и $(\mathcal{Z}, d_{\mathcal{Z}})$. Тогда функция $f : \mathcal{X} \to \mathcal{Z}$ Липшицева, если:

$$\exists K, \forall x_1, x_2 \in \mathcal{X}, d_{\mathcal{Z}}(f(x_1), f(x_2)) \leq K d_{\mathcal{X}}(x_1 - x_2)$$

Локально Липшицевой назывется та функция, в которой для точки найдётся такая окрестность в которой выполняется свойство Липшицевости.

Утверждение: **Любая искуственная нейронная сеть прямого распространения локально Липшицева.**

- g_θ – это генератор
- Генератор всегда является локально Липшицевой функцией, расстояние Вассерштейна определено и дифференцируемо почти всюду
- Но всё же, как считать $W(\mathbb{P}, \mathbb{Q})$?

Двойственная форма Канторовича-Рубинштейна:

$$W(\mathbb{P}, \mathbb{Q}_\theta) = \sup_{\|f\|_K \leq 1} \mathbb{E}_{x \sim \mathbb{P}} f(x) - \mathbb{E}_{x \sim \mathbb{Q}_\theta} f(x)$$

Где $\|f\|_K \leq 1$ обозначает семейство всех 1-Липшицевых функций вида $f : \mathcal{X} \to \mathbb{R}$

Двойственная форма Канторовича-Рубинштейна:

$$W(\mathbb{P}, \mathbb{Q}_\theta) = \max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}} f_w(x) - \mathbb{E}_{x \sim \mathbb{Q}_\theta} f_w(x)$$

Где $w \in \mathcal{W}$ параметры функции f, которая должна являться 1-Липшицевой

1. f_w – это дискриминатор, или как предлагается в оригинальной статье, **критик**.
2. Двойственная форма имеет простой вид и легко считается. А благодаря результатам полученным в теореме, $W(\mathbb{P}, \mathbb{Q}_\theta)$ дифференцируема почти всюду, следовательно нам больше не нужно аккуратно выстраивать расписание тренировки дискриминатора. **Мы можем тренировать до сходимости**
3. Осталось понять, как обеспецить 1-Липшицевость для критика f_w

Всё просто: **ограничим норму градиента**

Всё просто: **ограничим норму градиента**

Введем дополнительную фунцию потерь, обеспецивающую:

$$\|\nabla_w f\| \to 1$$

Improved Training of Wasserstein GANs Ishaan Gulrajani, et al.

GENERATIVE ADVERSARIAL NETWORKS

Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.

```
while θ has not converged do
    for t = 1, ..., n_critic do
        for i = 1, ..., m do
            Sample real data x ~ P_r, latent variable z ~ p(z), a random number ε ~ U[0, 1].
            x̃ ← G_θ(z)
            x̂ ← εx + (1 − ε)x̃
            L^(i) ← D_w(x̃) − D_w(x) + λ(‖∇_x̂ D_w(x̂)‖_2 − 1)^2
        end for
        w ← Adam(∇_w (1/m) Σ_{i=1}^m L^(i), w, α, β1, β2)
    end for
    Sample a batch of latent variables {z^(i)}_{i=1}^m ~ p(z).
    θ ← Adam(∇_θ (1/m) Σ_{i=1}^m −D_w(G_θ(z)), θ, α, β1, β2)
end while
```

$$\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z}), \quad \hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}, \quad L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$$