Исследование устойчивости генеративно-состязательных сетей

М. Зубков[a], А. Чубчева[a], А. Филиппова[a], Н. Сергеев[a]

[a]*Московский физико-технический институт (национальный исследовательский университет), Москва, Россия*

Abstract

Генеративно-состязательные сети (GAN) — это разновидность генеративных моделей, способная генерировать такие сложные типы данных, как картинки, звук и видео, однако, такие модели часто демонстрируют нестабильное поведение во время обучения. Целью данного исследования является обзор и применение теоретически обоснованных методов стабилизации процесса обучения GAN в задаче генерации мультимодального распределения.

Keywords: GAN, Устойчивость, Оптимизация, WGAN, Спектральная нормализация

1. Введение

Обучение генеративно-состязательной сети (GAN) представляет собой «состязание» двух нейронных сетей: генератора и дискриминатора. Задача генератора заключается в создании объектов (например, изображения или видео), по правдоподобности конкурирующие с обучающей выборкой, взятой из некоторого распределения μ_0. Задача дискриминатора состоит в оценке вероятности того, что образец семплирован из распредедения μ_0, а не создан генератором. Классическая схема обучения GAN представлена на рисунке 1.

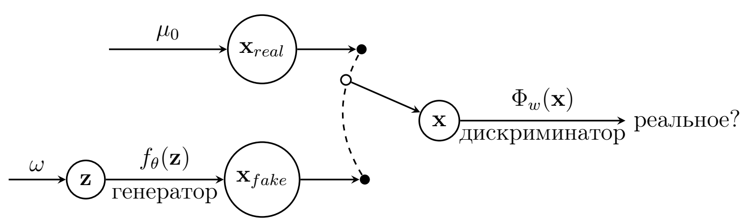


Рис. 1: Структура GAN

Для формулировки нашей задачи определим генератор формально:

Определение 1. *Пусть $X \subset \mathbb{R}^{n\times n}$ – множество исходных данных (например, картинок), $Z \subset \mathbb{R}^{m\times m}$ – некоторое измеримое множество, а ω – фиксированное распределение вероятности над множеством Z. Будем называть генератором параметризованную с параметром $\theta \in \mathbb{R}^d$ нейронную сеть $f_\theta : Z \to X$. Распределением генератора будем называть множество образов вероятностных мер $\{\mu_\theta = f_\theta(\omega),\ где\ \theta \in \mathbb{R}^d\}$.*

Также приведем формально определение дискриминатора:

Email addresses: zubkov.md@phystech.edu (М. Зубков), chubcheva.ad@phystech.edu (А. Чубчева), filippova.am@phystech.edu (А. Филиппова), sergeev.ng@phystech.edu (Н. Сергеев)

Определение 2. *Пусть $Y \subset \mathbb{R}^{n\times n}$ – множество данных, содержащее образцы как из исходной выборки, так и созданные генератором. Будем называть дискриминатором нейронную сеть $\Phi_w : Y \to [0,1]$, параметризованную с параметром $w \in \mathbb{R}^l$, сопоставляющую каждому элементу $y \in Y$ вероятность того, что этот y пришел из исходного распределения μ_0.*

Задача дискриминатора заключается в том, чтобы научиться отличать образцы, предложенные генератором Y_{μ_θ}, от образцов из обучающей выборки Y_{μ_0}. Пусть $\mathcal{J} : (\mu_\theta, \Phi_w) \to \mathbb{R}$ – функция потерь, зависящая от дискриминатора Φ_w и распределения μ_θ, созданного генератором. Чем больше при фиксированном μ_θ значение функции $\mathcal{J}$, тем лучше дискриминатор Φ_w отличает выборку из μ_0 от выборки из μ_θ. Тогда задача определения параметров дискриминатора формулируется следующим образом:

$$\max_{\Phi_w} \mathcal{J}(\mu_\theta, \Phi_w) = J(\mu_0, \mu_\theta) \tag{1}$$

Задача генератора заключается в создании таких объектов, чтобы фиксированный оптимальный дискриминатор как можно хуже отличал их от реальных объектов из распределения μ_0. Таким образом, цель генератора в том, чтобы минимзировать меру отклонения сгенерированной выборки от изначальной $J(\mu_0, \mu_\theta)$.

$$\min_{\mu_\theta} J(\mu_0, \mu_\theta) = \min_{\mu_\theta} \max_{\Phi_w} \mathcal{J}(\mu_\theta, \Phi_w) \tag{2}$$

Существует много различных вариантов определения функции $\mathcal{J}$, мы остановились на функции потерь WGAN [1], ее преимущества будут описаны в разделе 2.

$$J(\mu_0, \mu_\theta) = \max_{\Phi_w} \mathcal{J}(\mu_\theta, \Phi_w) = \max_{\|\Phi_w\|_{Lip} \le 1} \mathbb{E}_{x\sim\mu_0}[\Phi_w(x)] - \mathbb{E}_{x\sim\mu_\theta}[\Phi_w(x)], \tag{3}$$

где $\|\Phi_w\|_{Lip} \le 1$ обозначает, фунция имеет константу Липшица не более единицы.

Использование GAN позволяет с высокой точностью решать такие задачи, как восстановление изображений [2], увеличение разрешения изображения [3], увеличение размеров выборки данных [4], преобразование текста в изображения [5]. Однако, обучение GAN нестабильно, как показано в [1]. Данный вопрос будет подробнее изучен в следующем разделе.

2. Обзор литературы

Одной из наиболее важных проблем, возникающих при обучении GAN, является их нестабильность. Под нестабильностью понимается затухание градиентов функции потерь генератора $J(\mu_0, \mu_\theta)$ по параметрам θ, переобучение дискриминатора и чувствительность к гиперпараметрам, таким как число итераций обучения дискриминатора n_{cr} и размер шага в градиентном методе оптимизации. Для решения перечисленных проблем были предложены новые архитектуры для генератора и дискриминатора [6], целвые функции [1] и регуляризации [7, 8].

Ключевой шаг в решении проблемы нестабильности был сделан в статье [1]. В этой работе предложена функция потерь 2, которая порождена метрикой Вассерштайна [9], являющейся наиболее слабой метрикой в пространстве распределений. Для того чтобы определить наиболее слабую метрику, рассмотрим последовательность нейронных сетей-генераторов f_{θ_n}, где $\theta_n \in \mathbb{R}^d$ - параметризация на n-том шаге обучения генератора. Тогда метрика ρ_0 являеся наиболее слабой, если произвольная последовательность f_{θ_n}, сходящаяся по некоторой метрике ρ, будет также сходиться и по данной метрике ρ_0.

Функция потерь WGAN обладает перечисленными свойствами только в том случае, когда выполнена липшицевость и оптимальность дискриминатора на каждой итерации. Свойство липшицевости было подробно изучено в [7, 10, 11]. Вопрос оптимальности дискриминатора изучался в [12, 13, 14].

Также в статье [1] был предложен алгоритм 1 обучения GAN, который мы будем изучать в данной работе.

Algorithm 1 Используемые гиперпараметры: $\alpha = 9 \cdot 10^{-5}$, $m = 64$.

Require: α, шаг градиентного спуска. m, размер батча. n_{cr}, количество итераций обучения дискриминатора. w_0, начальные параметры дискриминатора. θ_0, начальные параметры генератора.

```
while не выполнен стоп критерий на θ do
    for t = 0, ..., n_cr do
        Сформировать выборку {x^(i)}_{i=1}^m ~ μ_0 батч из реальных данных.
        Сформировать выборку {z^(i)}_{i=1}^m ~ μ_θ батч из данных генератора.
        g_w ← ∇_w [ (1/m) Σ_{i=1}^m Φ_w(x^(i)) − (1/m) Σ_{i=1}^m Φ_w(f_θ(z^(i))) ]
        w ← w + α · Adam(w, g_w)
    end for
    Сформировать выборку {z^(i)}_{i=1}^m ~ μ_θ батч из данных генератора.
    g_θ ← −∇_θ (1/m) Σ_{i=1}^m Φ_w(f_θ(z^(i)))
    θ ← θ − α · Adam(θ, g_θ)
end while
```

Следующий шаг в решении проблемы нестабильности был сделан в статье [15], где была доказана теорема 2, утверждающая, что при выполнении некоторых условий $\|J_k(\mu_\theta)\| \to 0$ при $k \to \infty$, где k соответствует числу итераций обучения генератора. Это гарантирует сходимость функции потерь генератора к стационарной точке при бесконечном числе итераций. Данная теорема играет ключевую роль в нашей статье, сформулируем ее в следующей секции.

3. Постановка задачи

В данной статье мы будем изучать влияние параметра n_{cr} на обучение GAN. Как показано в статье [1], в общем случае GAN не сходится. Для решения данной проблемы в статье [15] рассматривается теорема, которая формулирует достаточное условие сходимости GAN:

Теорема 1 (Bertsekas [16]). *Предположим, что $J : \mu_\theta \to \mathbb{R}$ является ограничена снизу и L-гладкой, то есть $\|\nabla J\|_{Lip} \leq L$. Пусть $\theta_{k+1} = \theta_k - \alpha \nabla J(\mu_{\theta_k})$. Тогда $\|\nabla J(\mu_{\theta_k})\| \to 0$ при $k \to \infty$.*

Данная теорема гарантирует сходимость к стационарной точке при бесконечном числе итераций обновления параметров генератора и дискриминатора. Однако, согласно статье [17] функция J в общем случае не является L-гладкой. Поэтому для обоснования сходимости была доказана следующая теорема, которая гарантирует L-гладкость функции J:

Теорема 2 (Chu et al. [15]). *Пусть $J : \mu_\theta \to \mathbb{R}$ выпуклая функция. Зафиксируем $\mu := \mu_\theta$ и рассмотрим оптимальный дискриминатор: $\Phi_\mu : Y \to [0, 1]$. Пусть он удовлетворяет следующим условиям:*

(D1) $x \mapsto \Phi_\mu(x)$ – *α-липшицева,*

(D2) $x \mapsto \nabla_x \Phi_\mu(x)$ – *β_1-липшицева,*

(D3) $\mu \mapsto \nabla_x \Phi_\mu(x)$ – *β_2-липшицева по метрике 1-Wasserstein [9].*

Также, пусть семейство генераторов $f_\theta(\omega)$ удовлетворяет условиям:

(G1) $\theta \mapsto f_\theta(z)$ *A-липшицева в среднем для $z \sim \omega$, то есть,* $\mathbb{E}_{z\sim\omega}[\|f_{\theta_1}(z) - f_{\theta_2}(z)\|_2] \leq A\|\theta_1 - \theta_2\|_2$, *и*

(G2) $\theta \mapsto D_\theta f_\theta(z)$ *B-липшицева в среднем для $z \sim \omega$, то есть,* $\mathbb{E}_{z\sim\omega}[\|D_{\theta_1} f_{\theta_1}(z) - D_{\theta_2} f_{\theta_2}(z)\|_2] \leq B\|\theta_1 - \theta_2\|_2$.

Тогда $\theta \mapsto J(\mu_\theta)$ является L-гладкой, где $L = \alpha B + A^2(\beta_1 + \beta_2)$.

Одним из ключевых условий теоремы 2 является оптимальность дискриминатора на каждой итерации обучения генератора. Изучение данного вопроса представлено в статье [12]. Автор утверждает, что при оптимальном дискриминаторе процесс обучения генератора сходится с вероятностью 1. С другой стороны, в статье [13] показано, что обучение дискриминатора до оптимальности ведет к затуханию градиента для генератора. В связи с этим противоречием мы решили изучить влияние n_{cr} на процесс обучения GAN.

4. Методы

В статье [15] было доказано, что условия (D1)-(D3), (G1), (G2) теоремы 2 экививаленты уже изученным техникам стабилизации GAN, представленным в таблице 2.

Условие	Методы решения
(D0)	Оптимальность дискриминатора
(D1)	Спектральная нормализация [10] липшицева регуляризация [11]
(D2)	Гладкие функции активации, спектральная нормализация [10]
(D3)	Состязательная атака [8] WGAN-GP [7]
(G1)	Нет метода
(G2)	Нет метода

Рис. 2: Методы

В нашем исследовании мы будем использовать функцию потерь (2) и алгоритм 1, предложенные в статье [1]. При этом из описанных в таблице 2 методов мы будем использовать гладкие функции активации и спектральную нормализацию, предложенную в статье [10]. В нашей задаче дискриминатор является многослойным перцептроном:

$$\Phi_w(x) = W_L(a_L(W_{L-1}(...(W_1 x)...),$$

где $W_l \in \mathbb{R}^{(d_l+1)\times d_{l+1}}$ – l-тый линейный слой, a_l – функция активации после l-того линейного слоя, L – количество слоев дискриминатора. Такая архитектура представлена на рисунке 3(c) для случая $L = 4$.

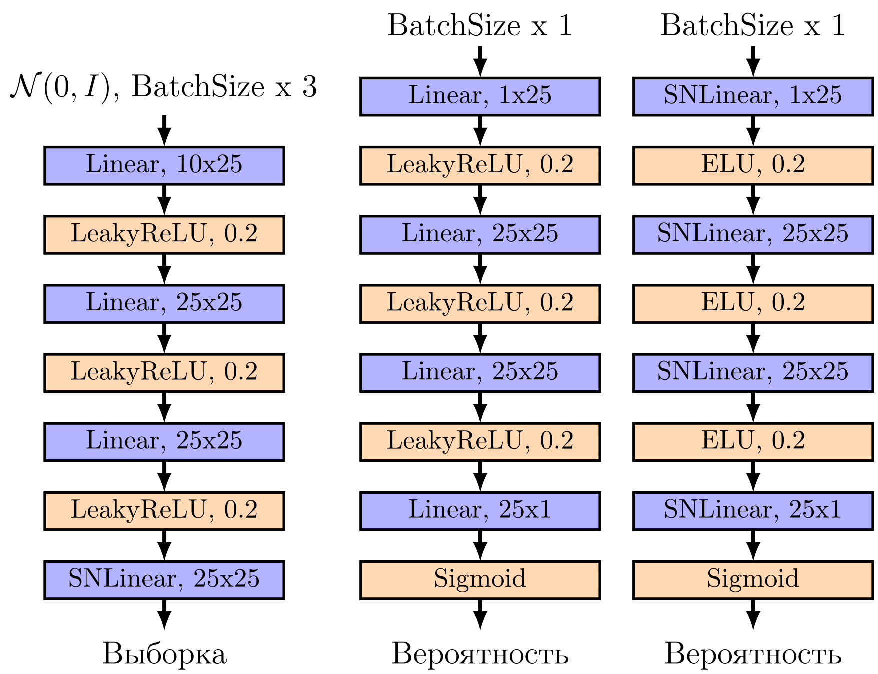
Также в нашей модели дискриминатора мы будем использовать спектральную нормализацию (4). В статье [10] показывается, что:

$$\|\Phi_w\|_{Lip} \leq \prod_{i=1}^{L} \|W_i\|_2$$

l-тый слой можно представить в виде матрицы W_l, для которой $\|W_l\|_2 = \rho(W_l^T W_l)$, где $\rho(W_l) = \max\{|\sigma| :$ где σ - сингулярное число матрицы $W_l^T W_l\}$. Таким образом, спектральной нормализации будет соответствовать преобразование матриц линейных слоев вида:

$$W_l^{SN} = \frac{W_l}{\|W_l\|_2} \tag{4}$$

В экспериментах мы будем использовать как модель со спектральной нормализацией, так и без. Для краткости будем обозначать их SNGAN и noSNGAN соответственно. Архитектуры их дискриминаторов и генераторов представлены на рисунке 3.



(a) Генератор (b) Дискриминатор noSNGAN (c) Дискриминатор SNGAN

Рис. 3: Архитектуры нейронных сетей

В качестве исходных данных мы взяли выборки из линейной комбинации нормальных распределений. Пусть $\xi \sim \mathcal{N}(\mu_1, \sigma_1), \eta \sim \mathcal{N}(\mu_2, \sigma_2), \zeta \sim \mathcal{N}(\mu_3, \sigma_3)$. Задача заключается в том, чтобы научить генератор создавать выборку из целевого распределения $\xi+\eta+\zeta \sim \mu_0$. Пример такого распределения представлен на рисунке 4, где $\mu_1 = 2, \sigma_1 = 0.4, \mu_2 = 0, \sigma_2 = 0.55, \mu_3 = 5, \sigma_3 = 0.25$.

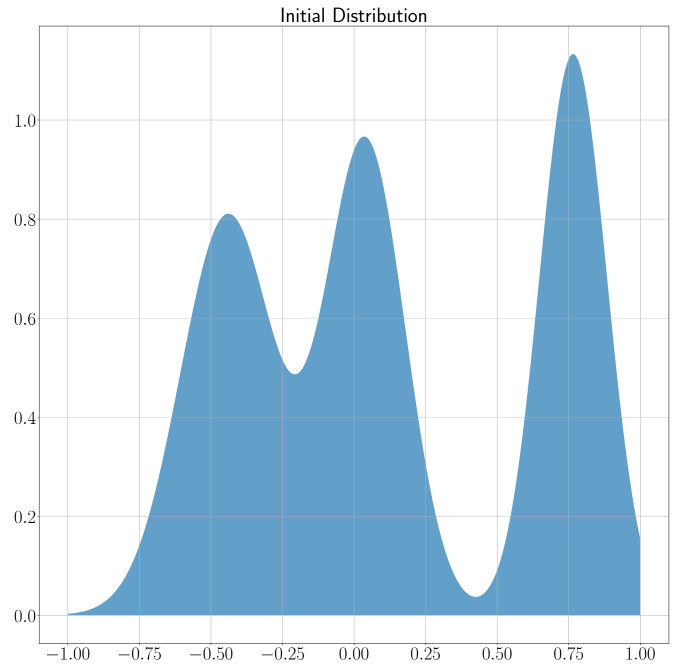


Рис. 4: Исходное мультимодальное распределение μ_0

Далее необходимо определиться с метрикой качества для объектов, созданных генератором. Для сравнения распределений двух выборок существует ряд статистических критериев однородности. Мы воспользовались p-value критерия Колмогорова-Смирнова [18], так как при верности нулевой гипотезы он зависит от μ_0 и μ_θ.

Поставим две гипотезы:

$$H_0 : \mu_\theta = \mu_0$$

$$H_1 : \mu_\theta \neq \mu_0$$

p-value будем называть вероятность получить такое же или более экстремальное значение некоторой статистики, при условии, что гипотеза H_0 верна. В нашем случае такой статистикой будет статистика Колмогорова-Смирнова 5. С помощью p-value мы будем оценивать, насколько похожи исходное и сгенерированное распределения, причем чем больше значение p-value, тем лучше они совпадают. Главное сойство статистики Колмогорова-Смирнова указана в следующей теореме:

Теорема 3. *Пусть $(X_1, ..., X_n)$ – выборка из распределения генератора μ_θ, а выборка $(Y_1, ..., Y_m)$ – выборка из целевого распределения μ_0. Пусть $\hat{F}_n, \hat{G}_m$ - эмпирические функции распределения выборок $\{X_i\}_{i=1}^n$ и $\{Y_i\}_{i=1}^m$ соответственно, что по определению значит: $\hat{F}(x) = \sum_{i=1}^n \mathbb{I}(X_i \leq x)$, где:*

$$\mathbb{I}(X_i \leq x) = \begin{cases} 1, & \text{если } X_i \leq x \\ 0, & \text{иначе} \end{cases}$$

Пусть также:

$$D_{n,m} = \sup_x |\hat{F}_n(x) - \hat{G}_m(x)| \tag{5}$$

При верной гипотезе $H_0 : \mu_0 = \mu_\theta$ статистика Колмогорова-Смирнова имеет распределение, не зависящее от распределений μ_0, μ_θ.

5. Эксперименты

Исходный код для воспроизведения результатов можно найти в репозитории `https://github.com/maximzubkov/opt-project`

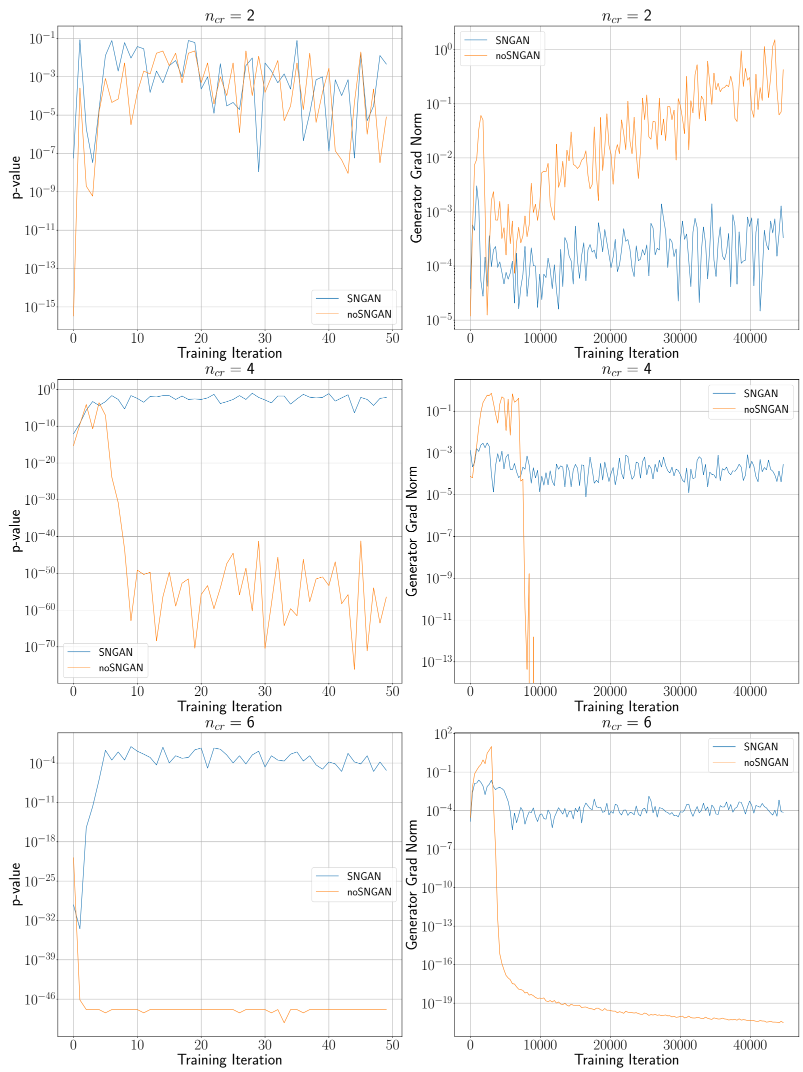


Рис. 5: Сравнение p-value и нормы градиента генератора для noSNGAN и SNGAN при различных значениях n_{cr}

Следуя подходу, предложенному в теореме 2, мы взяли постоянный шаг спуска равный $9 \cdot 10^{-5}$. Процесс вычисления шага представлен в статье [15]. Были получены следующие результаты:

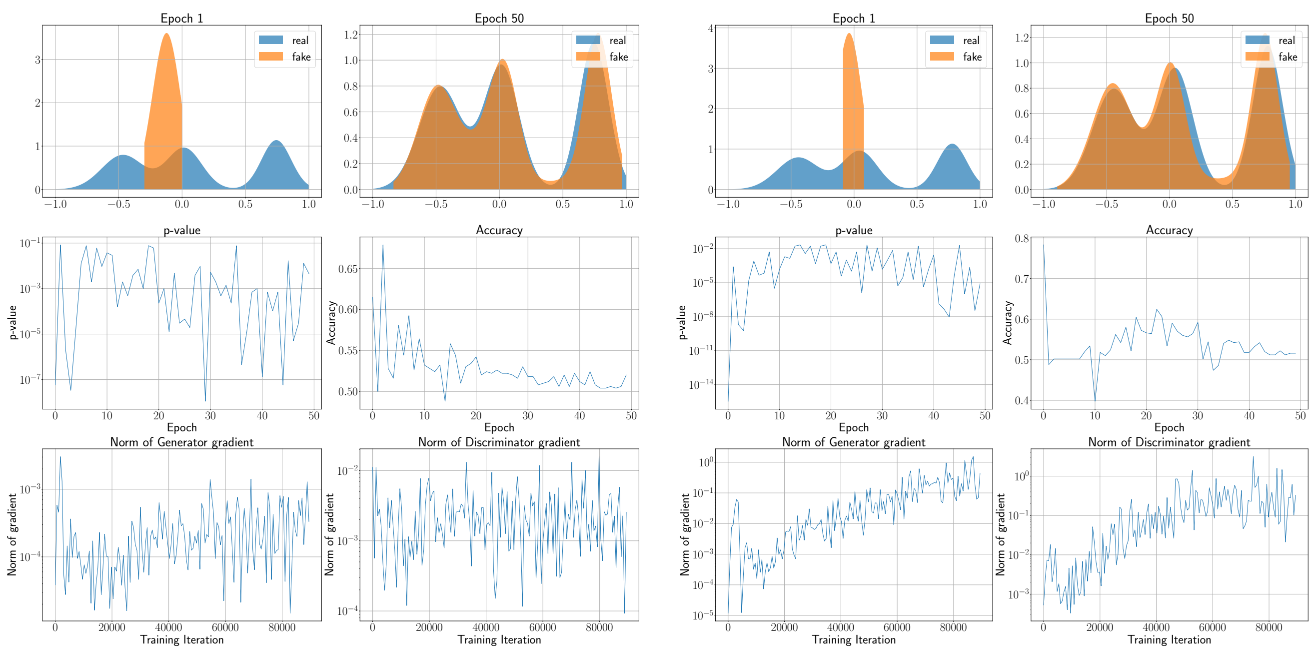
1. При $n_{cr} > 2$ noSNGAN показывает плохие результаты, что можно видеть на рисунках 6(b), 7(b), 8(b). Такой вывод можно сделать по тому, что p-value noSNGAN стремительно убывает к нулю. Также по рисунку 5 можно видеть, что при $n_{cr} > 2$ происходит затухание градиента. SNGAN сходится при любых значениях n_{cr}, сравнение сходимости по p-value представлено на рисунке 5.
2. Зависимость p-value генератора от числа n_{cr} представлена на рисунке 9. Можно видеть, p-value SNGAN почти не зависит от выбранного значения n_{cr}. Однако, по рисунку 8(a) можно заметить, что при $n_{cr} = 6$ мы столкнулись с проблемой коллапса мод. Можно

предположить, что при больших значениях n_{cr} генератору требуется больше эпох, чтобы обучиться, так как на рисунке 8(a) видно, что accuracy достаточно сильно колеблется.

3. Кроме того мы оценили насколько влияет использование спектральной нормализации на время обучения, результаты приведены в таблице 1. Как и ожидалось, SNGAN требует больше времени на итерацию, при этом при увеличении n_{cr} обоим GAN требуется меньше времени, так как необходимо делать меньше обновлений параметров генератора.

SN	n_{cr}	Время одной эпохи, сек.
$-$	2	2.92 ± 0.04
$-$	4	2.77 ± 0.04
$-$	6	2.62 ± 0.03
$+$	2	4.59 ± 0.04
$+$	4	4.26 ± 0.05
$+$	6	4.12 ± 0.02

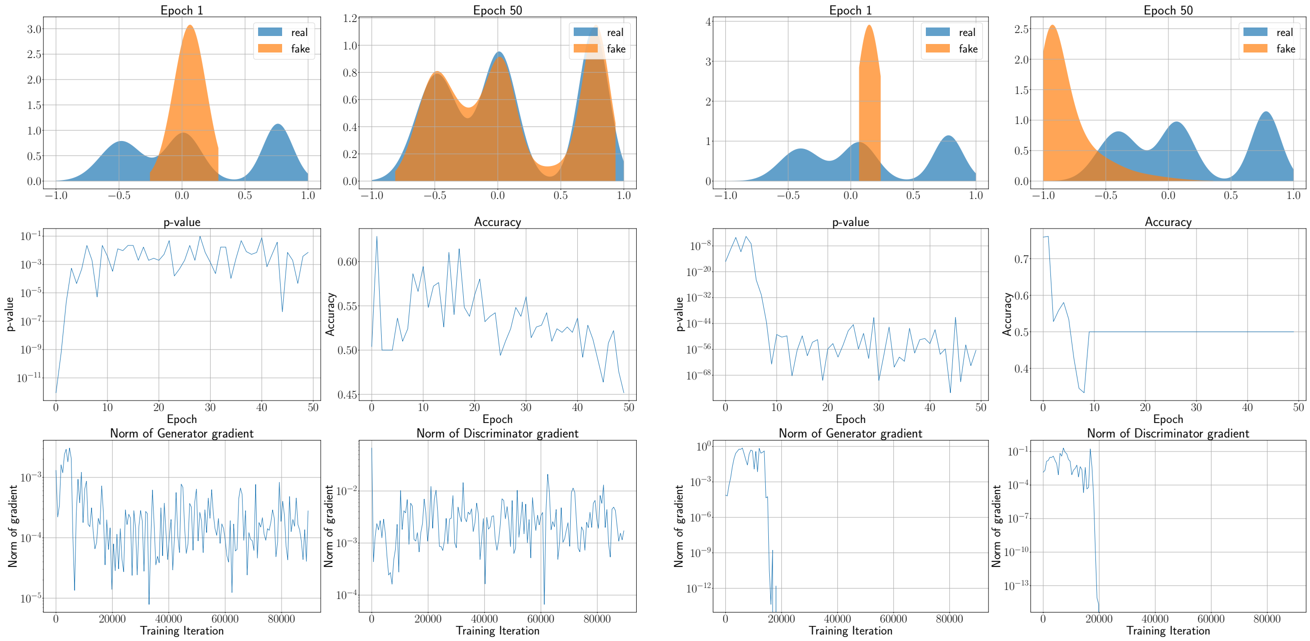
Таблица 1: Размер обучающей выборки для каждого эксперимента составлял $5 \cdot 10^4$. Шаг градиентного спуска $9 \cdot 10^{-5}$. Количество эпох обучения – 50.



(a) Архитектура SNGAN 3(b)

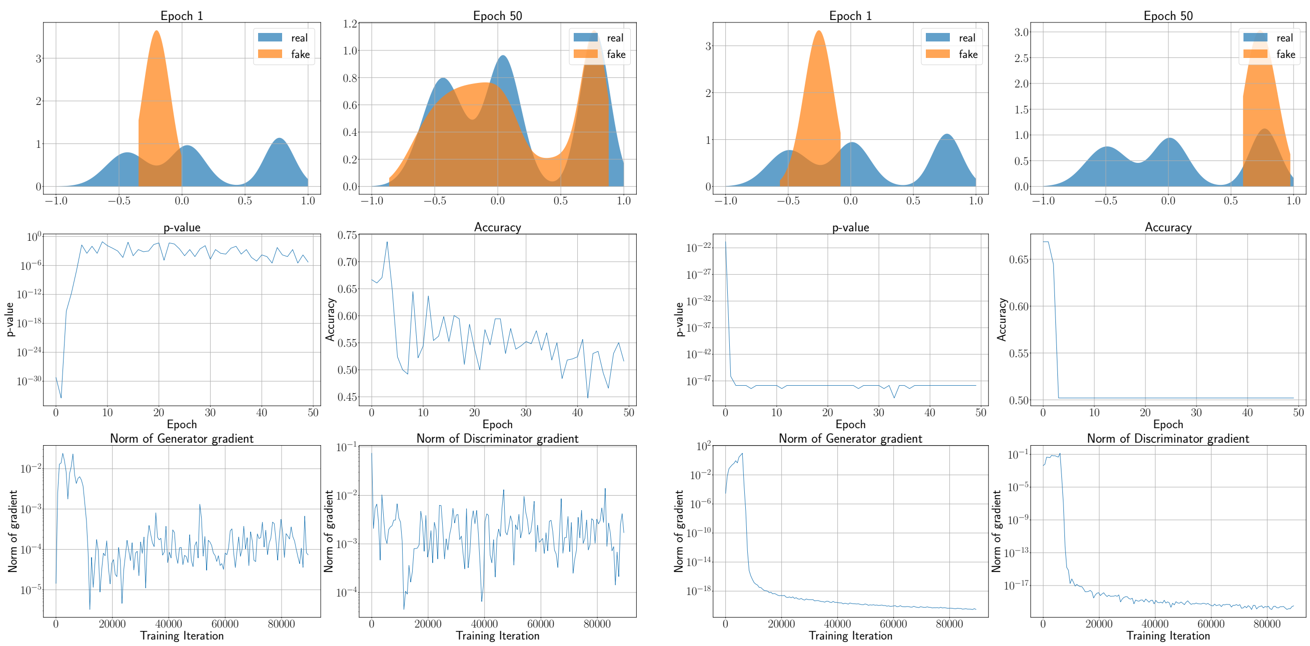
(b) Архитектура noSNGAN 3(c)

Рис. 6: Результаты экспериментов при $n_{cr} = 2$



(a) Архитектура SNGAN 3(b) (b) Архитектура noSNGAN 3(c)

Рис. 7: Результаты экспериментов при $n_{cr} = 4$



(a) Архитектура SNGAN 3(b) (b) Архитектура noSNGAN 3(c)

Рис. 8: Результаты экспериментов при $n_{cr} = 6$

6. Выводы

По полученным результатам были сделаны следующие заключения:

1. Исследуемый параметр n_{cr} оказывает сильное влияние на обучение noSNGAN и SNGAN. Спектральная нормализация ограничивает норму градиента, а в её отсутствие гради-

енты «взрываются», что приводит к переобучению дискриминатора. Как следствие, наблюдается затухание градиента, и остановка процесса обучения. В тех экспериментах, где задействована спектральная нормализация, переобучение дискриминатора контролируется, в результате чего GAN генерирует более правдоподобное распределение.

2. В ходе исследования мы выяснили, что увеличение n_{cr} влечет за собой уменьшение времени на одну эпоху обучения, хотя с другой стороны требуется большее число эпох для обучения генератора.

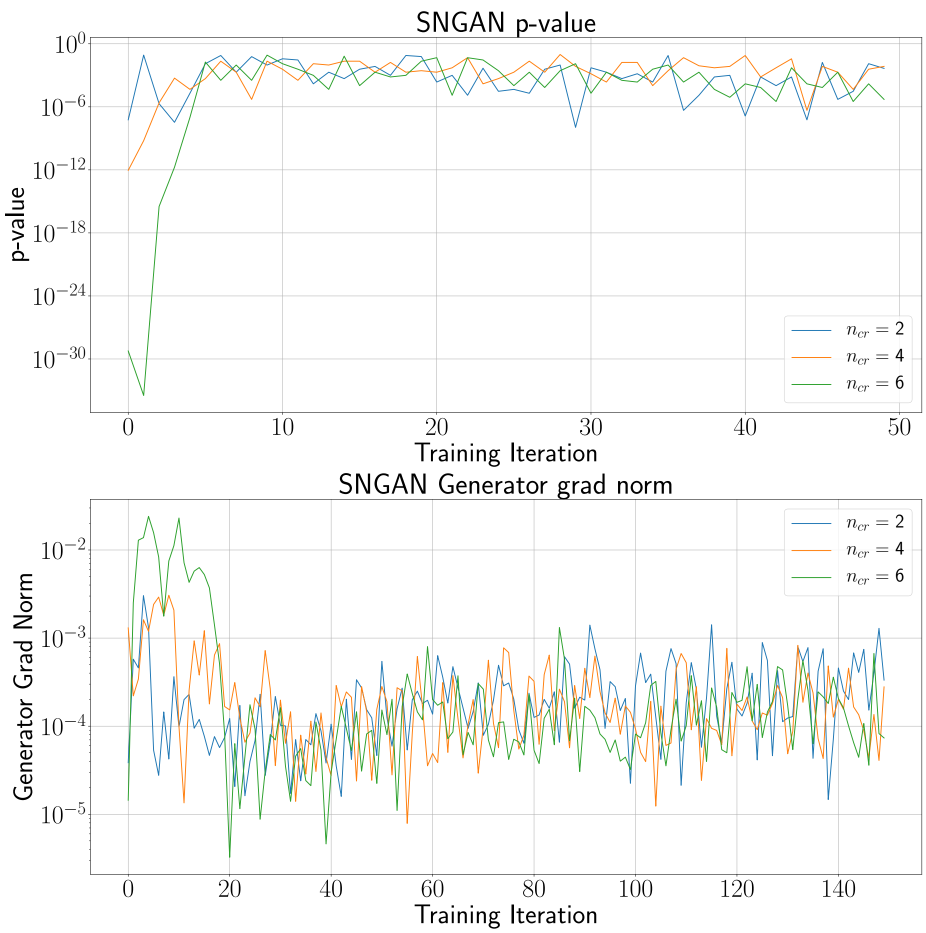


Рис. 9: Сравнение p-value и нормы градиента генератора при различных значениях n_{cr} для SNGAN

7. Результаты и будущая работа

В данной статье мы изучили влияние параметра n_{cr} на процесс обучения генеративно-состязательной нейронной сети, а также убедились в эффективности техник, предложенных теоремой 2.

В будущем планируется применить полученные результаты к обучению генеративных моделей на более сложных данных, таких как изображения, с использованием сетей, имеющих большее число нейронов и слоев.

Вопрос об оценке качества работы генератора остается открытым: для выборки из одномерных данных мы использовали статистический критерий, но такой подход не сработает в задачах, где данные имеют более сложную структуру. Как правило в задаче генерации изображений, используются такие метрики как FID и Inseption Score [19], однако данные метрики несовершенны, как показано в [20].

Для глубоких и сложных архитектур генератора и дискриминатора сложно оценить параметры $A, B, \alpha, \beta_1, \beta_2$ из теоремы 2. Поэтому возникла идея воспользоваться ансамблем простых нейронных стетей. Как утверждается в статье [21], residual-соединения образуют ансамбль из простых моделей, именно поэтому сети, снабженные residual-блоками обучаются легче и быстрее остальных. Данную идею можно применить и в нашей задаче: как было показано ранее, GAN с глубокими сетями генератора и дискриминатора сложнее обучить, однако сравнительно неглубокие архитектуры, как 3, обучаются быстрее и проще. Следовательно использование residual-сетей оправданно.

Список литературы

[1] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875 (2017).

[2] S. A. Hussein, T. Tirer, R. Giryes, Image-adaptive gan based reconstruction, arXiv preprint arXiv:1906.05284 (2019).

[3] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[4] F. H. K. d. S. Tanaka, C. Aranha, Data augmentation using gans, arXiv preprint arXiv:1904.09135 (2019).

[5] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D. Metaxas, Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2016. arXiv:1612.03242.

[6] L. Metz, B. Poole, D. Pfau, J. Sohl-Dickstein, Unrolled generative adversarial networks, arXiv preprint arXiv:1611.02163 (2016).

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of wasserstein gans, in: Advances in neural information processing systems, 2017, pp. 5767–5777.

[8] B. Zhou, P. Krähenbühl, Don't let your discriminator be fooled (2018).

[9] A. Andoni, P. Indyk, R. Krauthgamer, Earth mover distance over high-dimensional spaces., in: SODA, volume 8, 2008, pp. 343–352.

[10] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, arXiv preprint arXiv:1802.05957 (2018).

[11] Y. Yoshida, T. Miyato, Spectral norm regularization for improving the generalizability of deep learning, arXiv preprint arXiv:1705.10941 (2017).

[12] J. Li, A. Madry, J. Peebles, L. Schmidt, On the limitations of first-order approximation in gan dynamics, arXiv preprint arXiv:1706.09884 (2017).

[13] M. Wiatrak, S. V. Albrecht, Stabilizing generative adversarial network training: A survey, arXiv preprint arXiv:1910.00927 (2019).

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in neural information processing systems, 2016, pp. 2234–2242.

[15] C. Chu, K. Minami, K. Fukumizu, Smoothness and stability in gans, arXiv preprint arXiv:2002.04185 (2020).

[16] D. Bertsekas, Nonlinear Programming, Athena scientific optimization and computation series, Athena Scientific, 2016. URL: https://books.google.ru/books?id=TwOujgEACAAJ.

[17] L. Mescheder, S. Nowozin, A. Geiger, The numerics of gans, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 1825–1835. URL: http://papers.nips.cc/paper/6779-the-numerics-of-gans.pdf.

[18] S. Anulova, N. Krylov, R. Liptser, A. Shiryaev, A. Y. Veretennikov, Probability Theory III: Stochastic Calculus, volume 45, Springer Science & Business Media, 2013.

[19] M. J. Chong, D. Forsyth, Effectively unbiased fid and inception score and where to find them, arXiv preprint arXiv:1911.07023 (2019).

[20] M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, Are gans created equal? a large-scale study, in: Advances in neural information processing systems, 2018, pp. 700–709.

[21] A. Veit, M. J. Wilber, S. Belongie, Residual networks behave like ensembles of relatively shallow networks, in: Advances in neural information processing systems, 2016, pp. 550–558.