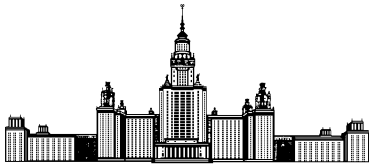
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

Магистерская курсовая работа

Из GAN до WGAN-GP: генерировать образцы с помощью нейросетей

Работу выполнил:

Цинь Вэй

Научный руководитель:

Кропотов Дмитрий Александрович

Москва, 2018

Содержание

Аннотация

Генеративо-состязательная сеть (GAN) уже стала популярной темой исследований в области искусственного интеллекта. В данной работе описывается изучение генеративно-состязательной сети(GAN), его вариаций глубокой сверточной генеративно-состязательной сети (DCGAN), условной генеративно-состязательной сети (CGAN), генеративно-состязательной сети Вассерштейна(WGAN) и генеративно-состязательной сети Вассерштейна со штрафами градиентов(WGAN-GP). Проведен анализ их преимуществ и недостатков теоретически и по результатам экспериментов. Проведено сравнение между этими алгоритмами. Анализ показал, что упорядоченные генеративно-состязательные сети WGAN и WGAN-GP более устойчивы для обучения, обеспечивают значимую метрику потерь, которая коррелирует с конвергенцией генератора и качеством выборки. Тем не менее, очевидного улучшения качества изображений с одинаковой вычислительной мощностью не наблюдалось.

1 Введение

В последние годы глубокое обучение сделало прорыв во многих областях, но все, похоже, обнаружили, что работа по прорыву в глубоком обучении в основном связана с дискриминантными моделями до 2014 года, Ian Goodfellow и его команда предложили генеративную модель вдохновлено антагонистической игрой в игровой теории: генеративно-состязательную сеть[5]. Генеративно-состязательная сеть содержит генеративную модель и дискриминантную модель(обычно искусственная нейронная сеть - математическая модель, описывающая систему соединенных между собой искусственных нейронов, а также реализации этой модели). Среди них модель генерации отвечает за захват распределения выборочных данных, а дискриминантная модель обычно является двоичным классификатором, который различает, являются ли входные данные реальными данными или сгенерированным образцом.Процесс оптимизации этой модели является проблемой «двоичной минимаксной игры», при обучении (генеративной модель и дискриминантной модель) одна из сторон фиксируется, параметры другой модели обновляются, это процесс повторяется до равновесия Нэша[15], и генеративная модель в это время может оценить распределение выборочных данных.

Конечно в современном искусственном интеллекте можно выделить два основных подхода к генерациям образцов: вариационному автокодировщику(VAE)[10] и генеративно-состязательной сети(GAN), но генеративно-состязательная сеть (GAN) показала лучшие результаты во многих генеративных задачах чем вариационный автокодировщик(VAE), поэтому приложения GAN уже были расширены от генерации изображений до различных областей компьютерного зрения и обработки естественного языка и мы будем рассматривать генеративно-состязательная сеть.

В данной работе впервые представили GAN и ее математику. Затем представили два конкретных примера:глубокую сверточную генеративно-состязательную сеть(DCGAN)[16] и условную генеративно-состязательную сеть(CGAN)[14], а затем мы провели эксперименты по набору данных MNIST[12] и проанализировали результаты.

Затем мы представили другую новую модель: генеративно-состязательную сеть Вассерштейна(WGAN)[2], объяснили рациональность, представили метод реализации и провели эксперименты. Результаты экспериментов показали эффективность.

Наконец, на основе WGAN оптимизировали алгоритм реализации, называемый генеративно-состязательной сетей Вассерштейна со штрафами градиентов(WGAN-GP)[6].

В работе было выяснено, что генеративно-состязательная сеть Вассерштейна со штрафами градиентов(WGAN-GP) дает хороший результат и самый устойчивый процесс обучения по сравнению с другими алгоритмами.

2 Генеративно-состязательная сеть(GAN) и ее вариации

2.1 Генеративно-состязательная сеть(GAN)

С тех пор, как Alexnet[11] возобновил глубокое обучение на историческом этапе, самым заметным достижением в области глубокого обучения является дискриминантная модель, известная как VGG[19], GoogleNet[20], ResNet[7], DenseNet[9] и т.д. Эти удивительные успехи в основном основаны на обратном распространении и трюках, таких как исключение и увеличение данных. В то же время глубокие генеративные модели имеют меньшее влияние, из-за сложности аппроксимирования многих трудноразрешимых вероятностных вычислений, которые возникают при оценке максимального правдоподобия и смежных стратегиях, чтобы обойти эти трудности. Ian Goodfellow и его команда предложил генеративную модель GAN [5] в 2014 году.

В предлагаемой генеративной модели противопоказана противнику: дискриминантная модель, которая учится определять, является ли образец из распределения генеративной модели или распределения данных, генеративную модель можно рассматривать как создателя поддельных денег и научиться обманывать дискриминантную модель. дискриминантную модель можно рассматривать как сотрудника полиции, который учится отличать контрафактные деньги.

Повторяя этот процесс, поскольку дискриминантные навыки дискриминантной модели становятся все более и более мощными, мошеннические методы генеративных моделей становятся все более изощренными, и первоклассный поддельный денежный производитель - это то, что нам нужно. Хотя идеи GAN очень интуитивны и просты, нам нужно еще больше понять доказательства и выводы, лежащие в основе этой теории.

В общем, в модели GAN нам необходимо одновременно тренировать две модели - генеративную модель G, которая может захватывать распределение данных и дискриминантную модель D, которая может оценить вероятность того, что данные взяты из реального образца. Процесс обучения генератора G заключается в том, чтобы максимизировать вероятность того, что дискриминатор совершит ошибки, то есть дискриминатор ошибочно предполагает, что данные являются реальной выборкой, а не ложным образцом, сгенерированным генератором. Таким образом, эта структура соответствует минимаксной игре двух участников. Во всех возможных функциях G и D мы можем найти единственное равновесное решение, т. е. G может генерировать такое же распределение, что и обучающий образец, а вероятность суждения D всюду равна 1/2. Другими словами, D и G играют следующую минимаксную игру двух участников со значением $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_z(x)}[\log(1 - D(G(z)))]$$

Кратко объяснить это уравнение. Теперь у нас есть дискриминатор D и генератор G. Для D мы пытаемся максимизировать формулу (сильную способность распознавания), а для G мы хотим минимизировать ее (сгенерированные данные близки к фактическим данным, и дискриминатор считает это - реальные данные). Весь тренинг - это итеративный процесс. На самом деле, минимальная игра максимизации может быть понята отдельно. То есть, учитывая G, мы сначала оптимизируем D (максимизируем V), затем фиксируем D и минимизируем V, чтобы получить G. Математически мы можем рассматривать истинные данные и сгенерированные данные как распределение вероятности, а оптимальная ситуация означает $P_{data}(x) = P_G(x)$.

2.1.1 Расстояние Кульбака-Лейблера и дивергенция Дженсена Шеннона

В теории информации мы можем использовать энтропию Шеннона[17, 13, 18] для количественной оценки общей неопределенности во всем распределении вероятности:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

Если для одной и той же случайной величины x существуют два отдельных распределения вероятностей $P(x)$ и $Q(x)$, мы можем использовать расстояние Кульбака-Лейблера(KLD)[18, 13] для измерения разницы между этими двумя распределениями:

$$D(p \parallel q) = \sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)}$$

Расстояние Кульбака-Лейблера(KLD) имеет много полезных свойств. Самое главное, что оно неотрицательно. Расстояние Кульбака-Лейблера(KLD) равно 0 тогда и только тогда, когда P и Q являются одинаковым распределением в случае дискретных переменных, или в случае непрерывных переменных - это то же самое, что и «почти всюду». Поскольку KLD неотрицательно и измеряет разницу между двумя распределениями, оно часто используется как расстояние между распределениями. Однако на самом деле это не настоящее расстояние, поскольку оно не симметрично: для некоторых P и Q, $KLD(P \parallel Q)$ не равно $KLD(Q \parallel P)$. Аналогично, существуют два распределения P и Q, а среднее распределение этих двух распределений $M = (P + Q)/2$, то дивергенция Дженсена Шеннона(JSD)[13] между этими двумя распределениями равна половине суммы KLD между P и M и KLD между Q и M.

$$JS(P_1 \parallel P_2) = \frac{1}{2} KL(P_1 \parallel \frac{P_1 + P_2}{2}) + \frac{1}{2} KL(P_2 \parallel \frac{P_1 + P_2}{2})$$

Аналогично, существуют два распределения P и Q, а среднее распределение этих двух распределений $M = (P + Q)/2$, то дивергенция Дженсена Шеннона(JSD)[13] между этими двумя распределениями равна половине суммы KLD между P и M и KLD между Q и M.

2.1.2 Глобальная оптимальность $P_G = P_{data}$

Сначала мы можем изменить математическое ожидание интегральной формой:

$$V(G, D) = \int_x P_{data}(x) \log(D(x)) + P_g(x) \log(1 - D(x)) dx$$

Фиксировавши G, с простым доказательством, мы можем получить оптимальный дискриминатор D:

$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

Теперь наша формула может быть переформулирована как:

$$\begin{aligned} C(G) &= \max_D V(G, D) \\ &= \mathbb{E}_{x \sim P_{data}}[\log D_G^*(x)] + \mathbb{E}_{x \sim P_z}[\log(1 - D_G^*(G(z)))] \\ &= \mathbb{E}_{x \sim P_{data}}[\log D_G^*(x)] + \mathbb{E}_{x \sim P_g}[\log(1 - D_G^*(x))] \\ &= \mathbb{E}_{x \sim P_{data}}\left[\log \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}\right] + \mathbb{E}_{x \sim P_g}\left[\log \frac{P_g(x)}{P_{data}(x) + P_g(x)}\right] \end{aligned}$$

Далее докажем, что глобальный минимум критерия виртуального обучения $C(G)$ достигается тогда и только тогда, когда $P_G = P_{data}$. В этот момент $C(G)$ достигает значения $-\log 4$. Чтобы доказать это, нам нужно объяснить это в двух направлениях. Первое,когда $P_G = P_{data}$,получаем $-\log 4$, это значение является кандидатом на глобальный минимум, поскольку оно появляется только тогда, когда $P_G = P_{data}$.

$$V(G, D_G^*) = -\log 2 \int_x P_G(x) dx - \log 2 \int_x P_{data}(x) dx = -2\log 2 = -\log 4$$

Второе, добавляя нулевые элементы в $C(G)$ и выводя, мы можем получить:

$$\begin{aligned} C(G) = \int_x &(\log 2 - \log 2) P_{data}(x) + P_{data}(x) \log\left(\frac{P_{data}(x)}{P_G(x) + P_{data}(x)}\right) \\ &+ (\log 2 - \log 2) P_G(x) + P_G(x) \log\left(\frac{P_G(x)}{P_G(x) + P_{data}(x)}\right) dx \end{aligned}$$

$$C(G) = -\log 4 + KL(P_{data} \mid \frac{P_{data} + P_G}{2}) + KL(P_G \mid \frac{P_{data} + P_G}{2})$$

Как мы обсуждали ранее дивергенцию Дженсена Шеннона, это равно:

$$C(G) = -\log 4 + 2 * JSD(P_{data} \mid P_G),$$

и мы узнали тогда и только тогда, когда два распределения точно совпадают, JSD принимает минимальное значение 0, поэтому минимальное значение $C(G)$ равно -log4, тогда и только тогда, когда $P_G = P_{data}$, таким образом, -log4 это глобальный минимум $C(G)$, и $P_G = P_{data}$ это единственное решение.

2.1.3 Конвергенция модели

Рассмотрим $V(G, D) = U(P_G, D)$ как функцию P_G, и как в статьи предполагается, $U(P_G, D)$ является выпуклой функцией, тогда субдифференциал супремумы выпуклых функций включают производную функции в точке, где достигается максимум, поэтому при достаточно малых обновлениях P_G, P_G сходится к P_{data}. но $U(P_G, D)$ может быть невыпуклым, поэтому это еще открытый вопрос, фактически в экспериментах мы не всегда можем наблюдать сходимость.

2.2 Условная генеративно-состязательная сеть(CGAN)

С момента внедрения GAN, его отличная производительность привлекла внимание многих исследователей. Поэтому на основе GAN существуют много других вариаций, и следующие обсуждаемые модели в этой статье могут быть считать как разные вариации исходных GAN.

Мы узнали основные принципы GAN, но в реализации GAN слишком произволен из-за отсутствия предварительного моделирования, например, если есть больше пикселей или более крупного изображения, эта модель будет неконтролируемой, поэтому для решения этой проблемы исследователи добавляют некоторые условные ограничения в модель GAN, которая называется условной генеративно-состязательной сетей (CGAN)[14]. В модели генерации G и дискриминантной модели D условие ограничения добавляется одновременно, чтобы направлять процесс генерации данных. Условиями могут быть любая дополнительная информация, такая как метки классов, метки других модальных и т.д. Это помогает GAN лучше применяться к кросс-модальным проблемам, таким как автоматическая аннотация изображения. Следующая - объективная функция и структура CGAN:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)}[\log D(x \mid y)] + \mathbb{E}_{z \sim P_z(x)}[\log(1 - D(G(z \mid y)))]$$

Возьми MNIST[12] в качестве набора данных, вход в генеративную модель представляет собой 100-мерный однородный вектор шума, а условие y - унитарный код метки класса.

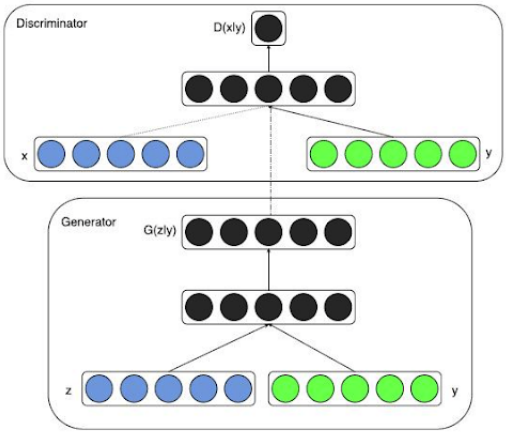


Рис. 1: Принцип CGAN

2.3 Глубокая сверточная генеративно-состязательная сеть(DCGAN)

В исходном GAN структура генератора и дискриминатора, принятая моделью, представляет собой многослойный персептрон, но эта модель очень неустойчива в обучении, она часто производит бессмысленный выход. В этой ситуации исследователи предложили новую структуру - глубокую сверточную генеративно-состязательную сеть(DCGAN), а также некоторые рекомендации по повышению стабильности модели. Вот структура и рекомендации DCGAN:

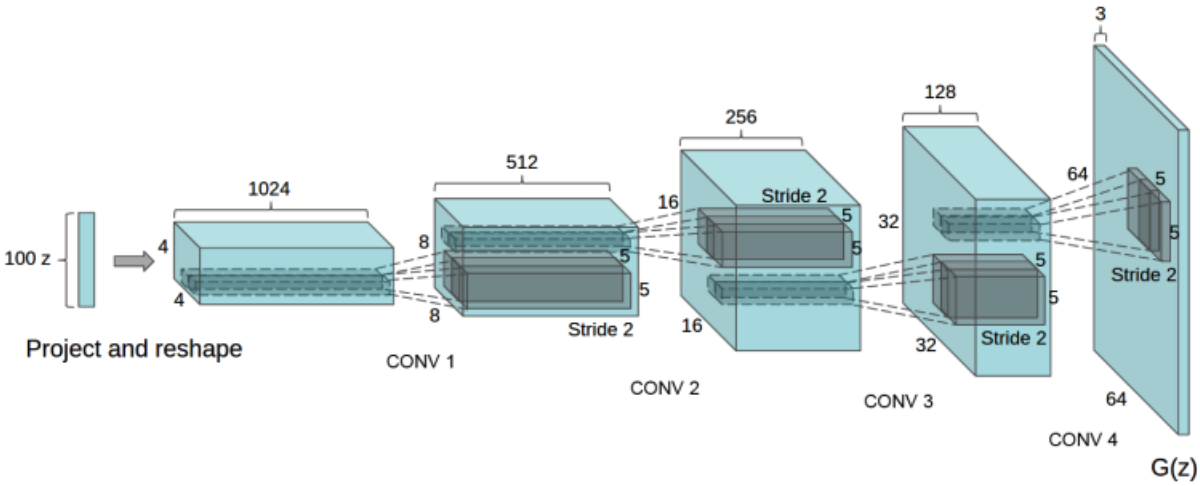


Рис. 2: архитектура DCGAN

Рекомендации по архитектуре: Заменяют все пулинг с свертками (дискриминатор) и транспонированными свертками(генератор). Используют нормализацию пар-

тии как в генераторе, так и в дискриминаторе. Удаляют полностью подключенные скрытые слои для более глубоких архитектур. Используют активацию ReLU в генераторе для всех слоев, за исключением вывода, в котором используется Tanh. Используют активацию LeakyReLU в дискриминаторе для всех слоев.

2.4 Реализация

Чтобы проверить эффективность GAN, я сделал несколько экспериментов. Фреймворк, которую я принял, была MXNet[4], поддерживаемая Amazon. По сравнению с Tensorflow, у которого больше всего пользователей, MXNet имеет почти такую же производительность и его проще развернуть. И наборами данных, которые я использовал, были MNIST и еще один сам сделанный набор данных лица девушки, вот несколько примеров.

вот алгоритм тренировки генеративно-состязательных сетей:

Algorithm 1 Мы устанавливаем скорость обучения как 0,0002, эпохи - 260, кроме того, число шагов, применяемых к дискриминатору, k, является гиперпараметром. В наших экспериментах мы использовали $k = 1$, наименее дорогостоящий вариант.

1: **for** количество обучающих итераций **do**
2: **for** k шагов **do**
3: Пример мини-партии образцов шума $\{z^{(1)}, \cdots, z^{(m)}\}$ от априорное распределение $P_g(z)$ шума.
4: Пример мини-партии m примеров $\{x^{(1)}, \cdots, x^{(m)}\}$ из распределения данных $P_{data}(x)$.
5: Обновите дискриминатор, подняв его стохастический градиент:

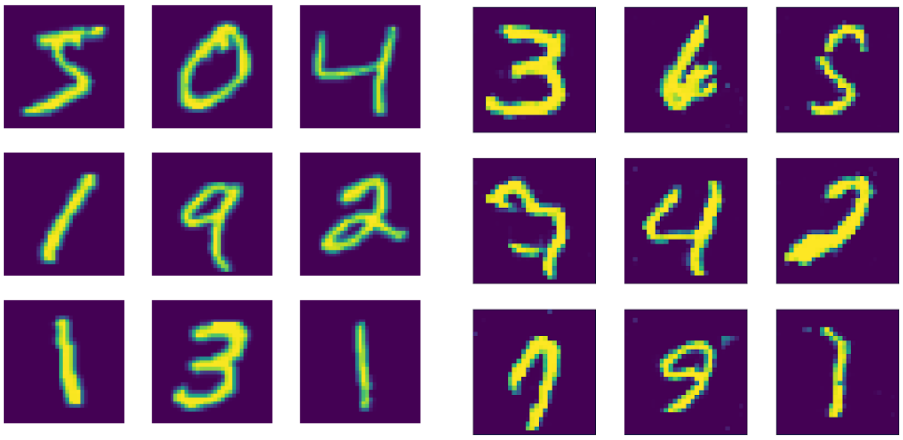
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))].$$

6: **end for**
7: Пример мини-партии образцов шума $\{z^{(1)}, \cdots, z^{(m)}\}$ от априорное распределение $P_g(z)$ шума.
8: Обновите генератор, подняв его стохастический градиент:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))).$$

9: **end for**
10: Обновления на основе градиента могут использовать любое стандартное правило обучения на основе градиента. Мы использовали момент(momentum) в наших экспериментах.

Ниже приведены некоторые из сгенерированных изображений с DCGAN. Рисунок 3 - экспериментальные результаты по набору данных mnist.Рисунок 4 - самодельный набор данных, содержащий более 8 000 изображений лица.



(a) оригинальные цифровые изображения

(b) сгенерированные цифровые изображения с DCGAN

Рис. 3: сгенерированные цифровые изображения с помощью DCGAN

(a) оригинальные изображения человеческих лиц

(b) сгенерированные изображения с DCGAN человеческих лиц

Рис. 4: сгенерированные цифровые изображения с помощью DCGAN

Кроме того, эта статья реализует CGAN[14], как показано на рисунке 5, из-за времени, аппаратных и других ограничений, у нас нет большого количества тестов, но мы видим, что добавленная переменная y может действительно управлять сгенери-

рованным изображением. Здесь, на входе дискриминатора, y транслируется до 28 * 28, а затем складывается с данными x для операции свертки. В генераторе y расширяется до 100 измерений, а затем подключается к шуму z для выполнения операции деконволюции.

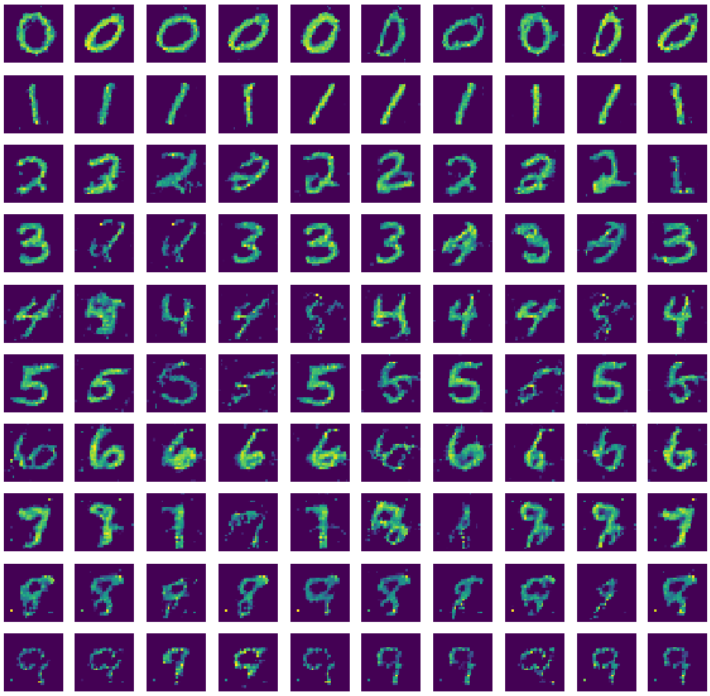


Рис. 5: изображение, которое содержит цифры 0-9

2.5 Анализ экспериментов

Как мы видели, хотя качество изображений не очень стабильно, мы действительно можем генерировать некоторые образцы с помощью DCGAN, здесь мы оцениваем образцы своим чувством из-за ограничения времени, на самом деле существуют более разумные метрики для оценки мощности генератора, например inception score[3] и FID[8].

2.5.1 Метрики inception score и Frechet Inception Distance (FID)

Inception Score[3] предлагает способ количественной оценки качества сгенерированных выборок. Мы применяем модель сначала к каждому сгенерированному изображению, чтобы получить условное распределение меток $p(y \mid x)$. Изображения, содержащие содержательные объекты, должны иметь условное распределение меток $p(y \mid x)$ с низкой энтропией. Более того, мы ожидаем, что модель будет генерировать разнообразные изображения, поэтому частное распределение $p(y \mid x = G(z))$ dz должно иметь высокую энтропию.

Наконец, эти соображения объединяются в один балл:

$$\mathrm{IS} = \exp(\mathbb{E}_{x \sim G}[d_{KL}(p(y \mid x), p(y))])$$

Frechet Inception Distance (FID) предлагает альтернативный подход. Чтобы количественно оценить качество сгенерированных выборок, они сначала встроены в пространство функций, предоставленное Inception Net [20]. Затем, рассматривая слой внедрения как непрерывный многомерный гауссов, среднее и ковариация оцениваются как для сгенерированных данных, так и для реальных данных. Расстояние Фреше между(Frechet distance) этими двумя гауссианами затем используется для количественного определения качества образцов, т. е.

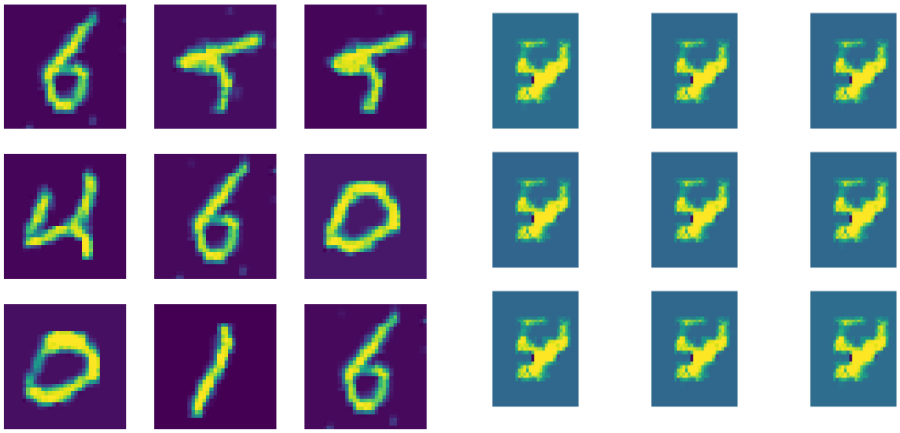
$$\mathrm{FID}(x, g) = \|\mu_x - \mu_g\|_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{\frac{1}{2}})$$

(Меньшее значения FID означают лучшее качество изображения и разнообразие) где (μ_x, Σ_x) и (μ_g, Σ_g) - это средняя и ковариация от распределения реальных изображений и генерируемых изображений , Tr - суммирует все диагональные элементы.

2.5.2 Сбой режима(mode collapse)

Почти все распределения данных интересного реального мира очень сложны и мультимодальны. Но когда мы тренируем GAN, иногда мы получаем генератор, который учится производить образцы с чрезвычайно низким разнообразием. Мы называем эту проблему сбоем режима:

На рисунке 6 показана сбой режима. На рисунке 6(a) мы можем заметить, что три числа одинаковы. На рисунке 6(b) все числа имеют одинаковую форму.



(a) сгенерированные цифровые изображения с DCGAN

(b) сгенерированные цифровые изображения с DCGAN,

Рис. 6: некоторые результаты, которые могут показать сбой режима

3 Генеративно-состязательная сеть Вассерштейна(WGAN)

С тех пор как Ian Goodfellow предложил GAN в 2014 году, исследователи проделали большую работу и обнаружили много недостатков[1] идеальный дискриминатор, отсутствие различий в образованных изображениях, функция потерь дискриминатора и генератора не может указывать процесс обучения.Несмотря на успех GAN, существует мало теорий, объясняющих неустойчивое поведение обучения GAN. Из-за этого чрезвычайно сложно реализовать его с новыми вариантами или использовать их в новых доменах, чтобы изменить эту ситуацию, некоторые исследователи много пробовали и получали некоторый прогресс.

3.1 Проблемы оригинальной генеративно-состязательной сети и решение

В одной статье дается конкретное доказательство проблемы, из-за ограниченности пространства, эта статья даст лишь интуитивное представление о том, почему предлагается WGAN.

Как видно из рисунка 7, после двух или трех итераций точность классификации дискриминатора равна 1, а в большинстве случаев учебного процесса это значение

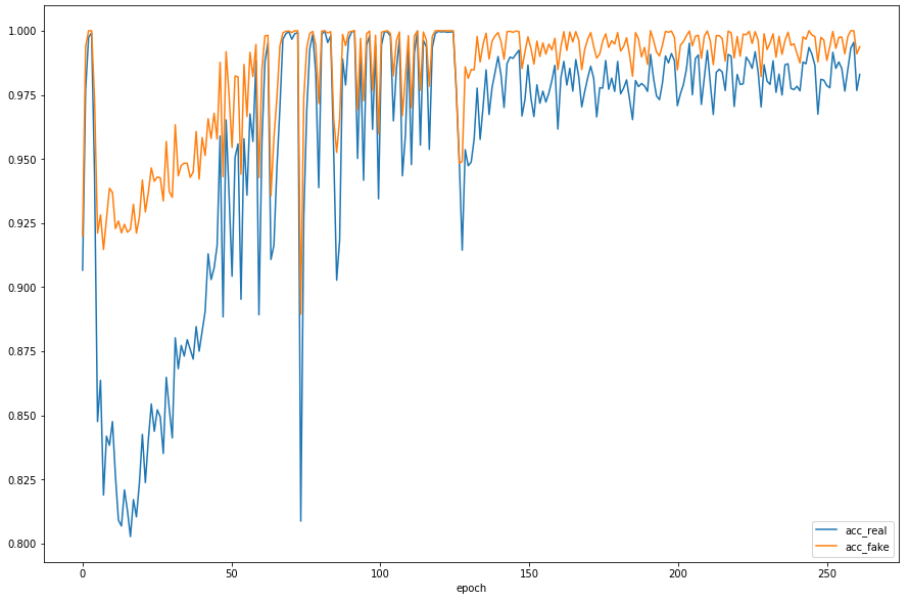


Рис. 7: график точности классификации дискриминатора по мере изменения эпохи

стабильно около 1, что означает, что существует идеальный дискриминатор, который отлично отличает все реальные изображения и сгенерированные изображения. Затем мы объясним эту проблему с двух сторон. Во-первых, эта причина этой ситуации и какие проблемы вызовет этот идеальный дискриминатор. Во-вторых, как мы можем решить эту проблему.

В случае GAN, P_g определяется путем выборки из простого предыдущего $z \sim p(z)$, а затем применяя функцию $g : Z \to X$, поэтому носитель P_g должен содержаться в $g(Z)$. Если размерность Z меньше размерности X (как это обычно бывает), то для Pg невозможно быть непрерывным. Это связано с тем, что в большинстве случаев $g(Z)$ будет содержаться в объединении маломерных многообразий и, следовательно, имеет меру 0 в X. Заметим, что хотя это и интуитивно, это очень нетривиально, так как наличие n-мерной параметризации абсолютно не означает, что изображение будет лежать на n-мерном многообразии.

Другими словами, почти нет пренебрежимого перекрытия между P_g и P_r, поэтому независимо от того, насколько узким является “промежуток” между ними, должна быть оптимальная поверхность сегментации для их разделения, кроме перекрытия, которое можно игнорировать. Поскольку дискриминатор представляет собой нейронную сеть с очень сильной фиктивной способностью, мы можем найти оптимальный дискриминатор и дать вероятность 1 почти всем реальным выборкам, а вероятность 0

задана для почти всех сгенерированных выборок. Хотя есть образцы, которые трудно классифицировать дискриминатором, мы уже говорили, что мера этого перекрытия почти равна 0, что можно игнорировать, в этом случае мы получим исчезновение градиента.

Давайте посмотрим на это по-другому. Мы уже знаем, что процесс оптимизации генератора фактически приближает JSD между двумя распределениями P_r и P_g. Затем мы проанализируем что произойдет, когда перекрытие между P_r и P_g может быть проигнорировано. Для любой переменной x мы имеем четыре случая:

1. $P_r(x) = 0, P_g(x) \neq 0$
2. $P_r(x) \neq 0, P_g(x) = 0$
3. $P_r(x) = 0, P_g(x) = 0$
4. $P_r(x) \neq 0, P_g(x) \neq 0$

в первом и втором случаях JSD равна log2, третий случай не способствует расчёте JSD. Четвертый случай также не способствует тому, что мера перекрытия равна нулю. Таким образом, до тех пор, пока между этими двумя распределениями не будет пренебрежимо малая перекрываемость, JSD между ними почти постоянное log2, поэтому наш учебный процесс, который хочет приближает JSD будет бессмысленным.

Коренная причина исходной проблемы GAN может быть уменьшена до двух точек. Один из них заключается в том, что измерение расстояния для оптимизации эквивалентности (JSD) необоснованно, а другое заключается в том, что перекрытие между двумя распределениями можно игнорировать.

Одна статья предложила решение для второй точки, которая заключается в добавлении шума к сгенерированному образцу и реальному образцу. Интуитивно, исходные два низкоразмерных многообразия рассеяны во всем высокоразмерном пространстве, заставляя их иметь не пренебрежимое перекрытие. Как только происходит перекрытие, JSD может действительно работать, и тогда проблема исчезновения градиента решена. Во время учебного процесса мы можем отжигать добавленный шум и медленно уменьшать его дисперсию. Когда онтологии двух низкоразмерных многообразий перекрываются, даже если шум полностью удален, JSD также может работать и продолжать создавать значимые градиенты, которые приближают два низкоразмерных многообразия.

В рамках этого решения мы можем безопасно обучать дискриминатора оптимальному, не беспокоясь о том, что градиент исчезает. Когда дискриминатор оптимален,

минимальную потерю дискриминатора можно получить по формуле:

$$\min_D L(P_{r+\epsilon}, P_{g+\epsilon}) = -\mathbb{E}_{x\sim P_{r+\epsilon}}[\log D^*(x)] - \mathbb{E}_{x\sim P_{g+\epsilon}}[\log(1 - D^*(x))]$$
$$= 2\log 2 - 2JS(P_{r+\epsilon} \parallel P_{g+\epsilon})$$

Как мы видим, конкретное значение шумной JSD зависит от дисперсии шума. Поскольку шум отжигается, значения до и после не могут сравниваться, поэтому он не может быть существенной мерой расстояния между P_r и P_g. Это означает, что это решение делает два распределения перекрывающимися, но все же не может обеспечить значимого индикатор учебного процесса.Поскольку этот метод может решить только одну проблему, исследователи предложили другое решение: использование расстояния Вассерштейна, которое учитывает оба эти проблемы.

3.2 Расстояния Вассерштейна

Расстояние Земного Двигателя (EM) или Расстояние Вассерштейна:

$$W(P_r, P_g) = \inf_{\gamma\sim\prod(P_r,P_g)} \mathbb{E}_{(P_r,P_g)} [\|x - y\|]$$

где $\prod(P_r, P_g)$ обозначает множество всех совместных распределений $\gamma(x, y)$, маргиналами которых являются соответственно P_r и P_g. Интуитивно, $\gamma(x, y)$ указывает, сколько "массы" необходимо транспортировать из x в y, чтобы преобразовать распределения P_r в распределение P_g. Дальность EM - это "стоимость" оптимального транспортного плана.

Превосходство расстояния Вассерштейна по сравнению с KLD и JSD состоит в том, что даже если два распределения не перекрываются, расстояние Вассерштейна все же может отражать их расстояние.

Похоже, что нижняя грань в этой формуле очень трудноразрешима. Но повезло, что двойственность Канторовича-Рубинштейна говорит нам, что

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L\leq K} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

Здесь нам нужно ввести понятие - липшицево непрерывное. Это на самом деле накладывает ограничение на непрерывную функцию f, требующую постоянной K,

чтобы сделать любые два элемента x_1 и x_2 в области определения удовлетворяли условию:

$$\|f(x_1) - f(x_2)\| \leq K\|x_1 - x_2\|$$

В это время K называется константой Липшица функции f. Поэтому если мы имеем параметризованное семейство функций $f(w)$, удовлетворяющих K-липшицу для некоторого K, мы могли бы рассмотреть решение задачи

$$\max_{\omega \in W} \mathbb{E}_{x \sim P_r}[f_\omega(x)] - \mathbb{E}_{z \sim p(z)}[f_\omega(g_\theta(z)]$$

Здесь мы можем использовать нейронную сеть с параметром w для представления f, который является нашим дискриминатором. Мы можем достичь расстояния между этими двумя распределениями путем непрерывного обучения из-за мощных возможностей приближения нейронной сети. Наконец, что касается ограничения непрерывности непрерывное условие Липшица, мы не очень заботимся о конкретном , если это не положительная бесконечность, поскольку он только делает градиент больше раз и не влияет на направление градиента, поэтому мы взяли очень простой подход, ограничив все параметры нейронной сети $f(w)$ не более чем на некоторый диапазон $[-c, c]$, мы называем эту операцию “подрез веса”(weight clipping). В это время производная входного образца x не превышает некоторого диапазона, и выполняется непрерывное условие Липшица. В частности, в реализации алгоритма его нужно только обрезать до этого диапазона после каждого обновления ω.

До сих пор мы можем построить сеть дискриминаторов $f(w)$ с параметром w, когда предел w не превышает некоторого диапазона, мы максимизируем

$$L = \mathbb{E}_{x \sim P_r}[f_\omega(x)] - \mathbb{E}_{x \sim P_g}[f_\omega(x)]$$

насколько это возможно, при этом точка L аппроксимирует расстояние Вассерштейна между истинным распределением и порожденным распределением (игнорируя постоянное кратное K). Обратите внимание, что исходный дискриминатор GAN выполняет истинную и ложную двухклассическую классификационную задачу, поэтому последний слой является сигмовидным, но теперь дискриминатор $f(w)$ в WGAN делает приблизительное расстояние Вассерштейна, которое является задачей регрессии, поэтому последний слой сигмоида удаляется.

Тогда нам нужно минимизировать расстояние Вассерштейна чтобы приближать зти два распределения, благодаря отличному характеру расстояния Вассерштейна, нам не нужно беспокоиться о исчезновении градиента генератора. Тогда мы можем получить две функции потерь WGAN:

функция потерь гениратора WGAN: $-\mathbb{E}_{x\sim P_g}[f_\omega(x)$

функция потерь дискриминатора WGAN: $\mathbb{E}_{x\sim P_g}[f_\omega(x) - \mathbb{E}_{x\sim P_r}[f_\omega(x)]$

(Приближение противоположности расстояния Вассерштейна, чем меньше этот член, тем ближе значение, которое мы получаем к расстоянию Вассерштейна между двумя распределениями).

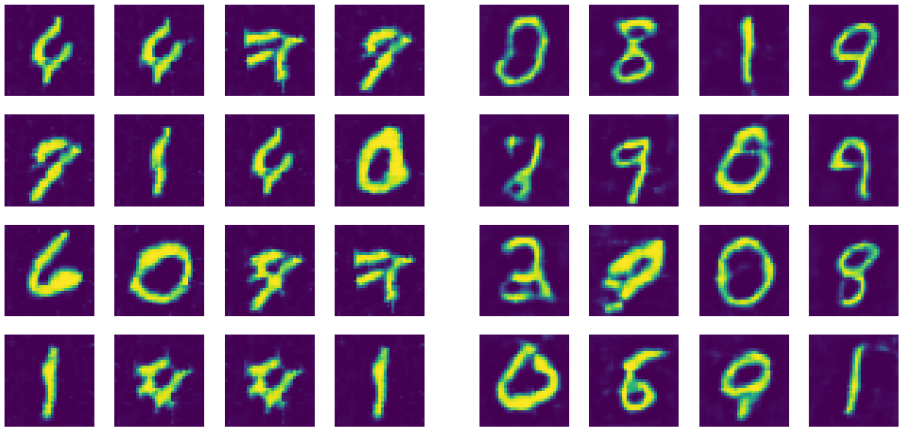
3.3 Реализация

Это конкретный алгоритм:

Algorithm 2 Мы устанавливаем скорость обучения $\alpha = 0.003$, параметр отсечения $c = 0,03$, число итераций критика на итерацию генератора $n_{critic} = 5$, кроме того, мы использовали ту же архитектуру, что и в 1.

while θ не сходится **do**
  for $t = 0, \cdots, n_{critic}$ **do**
    Пример мини-партии $\{x^{(i)}{}_{i=1}\}^m \sim P_r$ из реальных данных..
    Пример мини-партии $\{z^{(i)}\}_{i=1}^m \sim P_z$ из из предыдущих выборок.
    $g_w \leftarrow \nabla_w[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$
    $w \leftarrow w + \alpha \cdot \mathrm{RMSProp}(w, g_w)$
    $w \leftarrow \mathrm{clip}(w, -c, c)$
  end for
  Пример мини-партии $\{z^{(i)}\}_{i=1}^m \sim P_z$ из из предыдущих выборок.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \mathrm{RMSProp}(\theta, g_\theta)$
end while

Здесь рисунок 8 представлены образцы, сгенерированные с помощью WGAN.



(a) сгенерированные цифровые изображения с помощью DCGAN

(b) сгенерированные цифровые изображения с WGAN

Рис. 8: сгенерированные цифровые изображения с помощью DCGAN и WGAN с той же архитектурой и гиперпараметрами

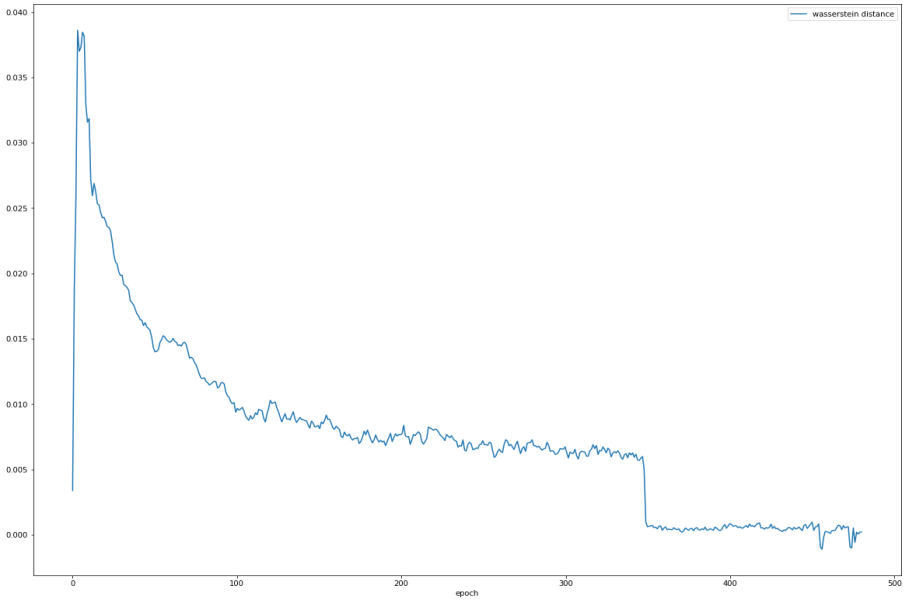
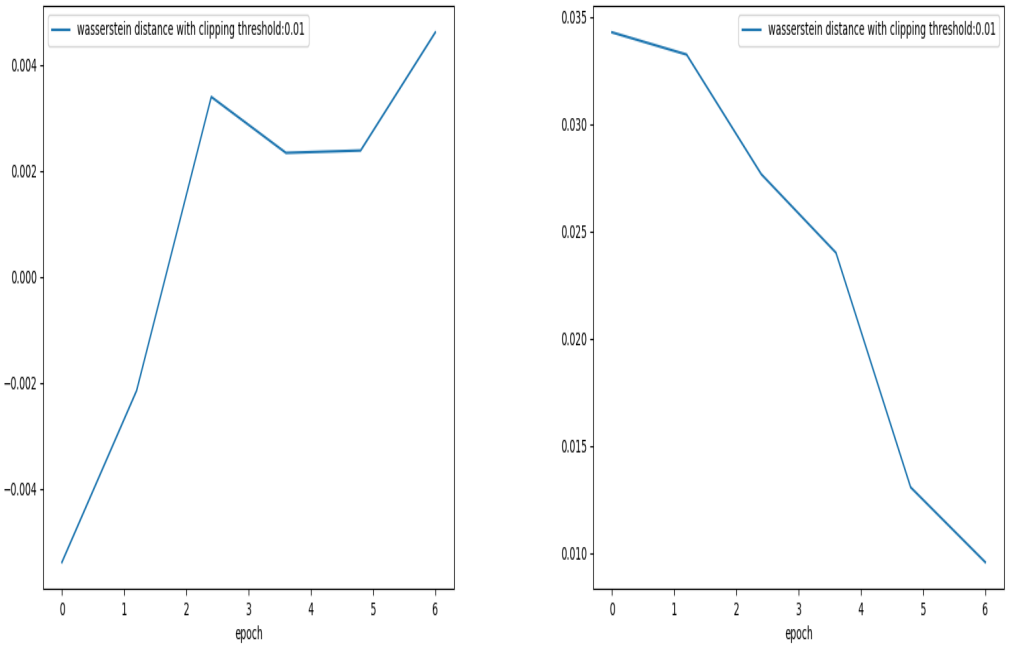


Рис. 9: график расстояния Вассерштейна как изменение эпохи

Как видно из рисунка 9, расстояние Вассерштейна также уменьшается по мере прогрессирования обучения. Но мы также заметили, что в начале учебного процесса расстояние Вассерштейна увеличилось, а затем уменьшилось, на самом деле это может быть отрицательным значением, если мы не тренировари критику не полностью или у нас разные величины порога подреза веса, что означает ценность, которую мы

получили, не является расстоянием Вассерштейна, и это ничего не значит, поэтому мы провели несколько экспериментов чтобы оптимизировать процесс обучения, вот результат. Из рисунка мы можем видеть, что с большим количеством критиков обучения ценность, которую мы получаем, будет более значимой. (мы проводили несколько раз, и это действительно полезный трюк)



(a) с 5 раз все время

(b) с 30 раз в первые 2 эпохи, 15 раз следующие 2 эпохи и 5 раз для остальных

Рис. 10: расстояние Вассерштейна в первые 6 эпох с различным числом критиков

3.4 Генеративно-состязательная сеть Вассерштейна с штрафами градиентов(WGAN-GP)

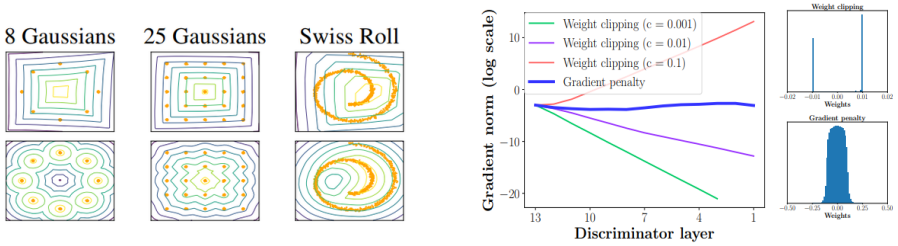
Возможно, без каких-либо доказательств вы можете почувствовать, что липшицево непрерывное с помощью подрез веса - не изящный метод. Фактически, этот метод вызовет некоторые проблемы.

Таким образом, в этом случае исследователи предложили новый метод чтобы удовлетворить липшицево непрерывное, который заключается в том, чтобы добавить

новый термин к функции потерь:

$$L(D) = \mathbb{E}_{x\sim P_g}[D(x)] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda \mathbb{E}_{x\sim X}[\|\nabla_x D(x)\|_p - 1]^2$$

Как показано на рисунке 11, мы непосредственно устанавливаем дополнительный термин потерь, чтобы отразить, что “градиент дискриминатора не превышает "K” с WGAN-GP мы видим, что мы можем получить генератор, который может захватить более высокий момент распространения данных.



(a) Ценностные поверхности критиков WGAN обучаются оптимальности на наборах игрушек с использованием (верхнего) подреза веса и (нижнего) градиентного штрафа. Критики, обученные с подрезом веса, не могут захватить более высокие моменты распространения данных. «Генератор» фиксируется на реальных данных плюс гауссовский шум.

(b) Градиентные нормы глубоких критиков WGAN во время обучения по набору данных Swiss Roll либо взрываются, либо исчезают при использовании подрезе веса, но не при использовании градиентного штрафа. (справа) Обрезание по весу (вверху) подталкивает грузы к двум значениям (крайности диапазона отсечения), в отличие от штрафа градиента (снизу)

Рис. 11: Градиентный штраф в WGAN не проявляет нежелательного поведения как подрез веса

Как видно из рисунка 12 в WGAN, расстояние Вассерштейна в WGAN-GP также является метрикой здесь.

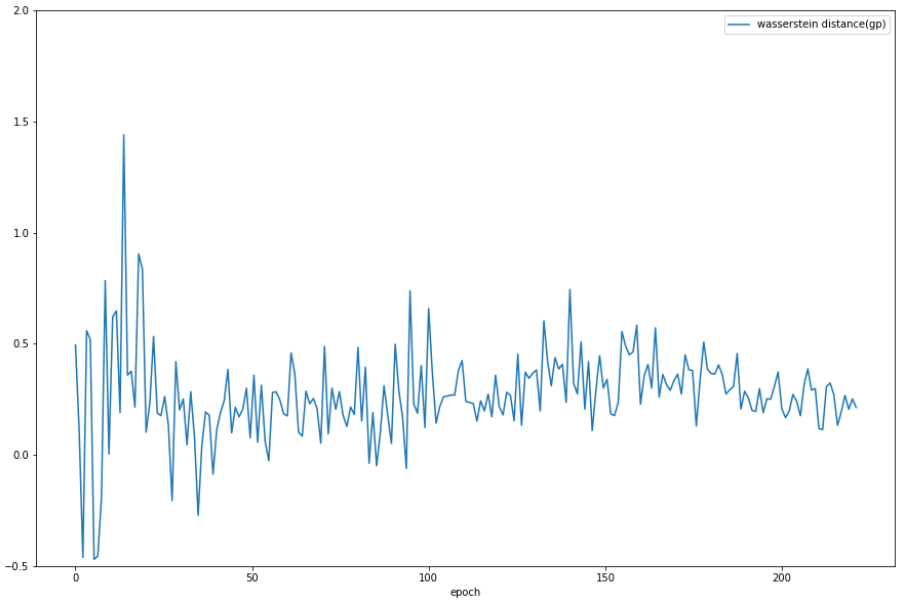
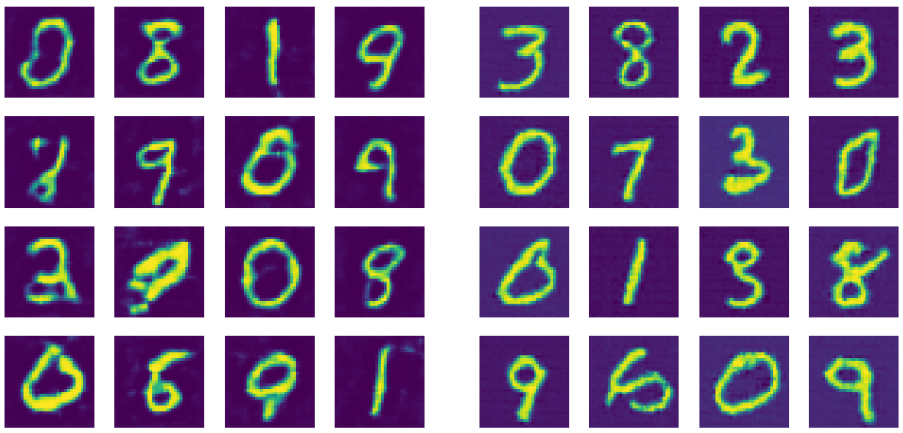


Рис. 12: график расстояния Вассерштейна как изменение эпохи(WGAN-GP)



(a) сгенерированные цифровые изображения с помощью WGAN

(b) сгенерированные цифровые изображения с WGAN-GP

Рис. 13: сгенерированные цифровые изображения с помощью wGAN и WGAN-GP с той же архитектурой и гиперпараметрами

Как показано на рисунке 13 и рисунке 8, с помощью WGAN-GP мы можем сгенерировать более качественные изображения с большей стабильностью и лучшим качеством.

Чтобы количественно сравнить различные структуры, мы использовали Inception Score[3] и FID[8], о котором мы упоминали ранее, в качестве показателей для сравнения DCGAN, WGAN и WGAN. Ниже приведены результаты сравнения(по MNIST).

Таблица 1: сравнение разных структур

	Inception Score	FID
DCGAN	1.483697	203.75015
WGAN	1.5933928	179.75687
WGAN-gp	1.8670504	171.6774

Как мы обсуждали ранее, меньшее значения FID и более значение Inception Score означают лучшее качество изображения и разнообразие

4 Заключение и будущая работа

В рамках курсовой работы были исследованы и опробованы разные модели генеративно-состязательных сетей для генеративной задачи изображений, подведем итоги:

- были объяснены и проведены эксперименты модели GAN, DCGAN, CGAN, WGAN и WGAN-GP.
- Анализировали преимущества и недостатки этих моделей, обнаружили, в случае одной и той же архитектуры и гиперпараметров, WGAN и WGAN-GP займет больше времени, но сгенерированные изображения будут иметь более высокое качество и быть более стабильными. Таким образом, для простых наборов данных WGAN-GP может использоваться для создания высококачественных изображений. Для сложных наборов данных WGAN или даже DCGAN могут использоваться для сокращения требуемого времени (хотя для корректировки гиперпараметров может потребоваться больше времени).
- По сравнению с другими моделями, WGAN-GP более устойчив.

Несмотря на то, что GAN хорошо выполняет задачу генерации изображений, все еще остается нерешенными проблемами:

- Можно легко найти, что упорядоченная модель например WGAN и WGAN-GP работает лучше, чем нерегулируемый сеть как экспериментально, так и теоретически. Тогда возникает проблема, кроме условий Lipschitz, есть ли другие условия регуляризации??
- Существуют метрики inception score и FID для сравнения различных моделей GAN, нам нужно сравнить разные модели с этими метриками.
- Когда я читал статьи о WGAN, я видел как исследователи объясняли это с точки зрения геометрии и параметризации многообразий, хотя моих знаний было недостаточно для понимания, оно дало направление для моих будущих исследований.

Список литературы

[1] M. Arjovsky and L. Bottou. Towards Principled Methods for Training Generative Adversarial Networks. https://arxiv.org/abs/1701.04862, January 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. https://arxiv.org/abs/1701.07875, January 2017.

[3] S. Barratt and R. Sharma. A Note on the Inception Score. https://arxiv.org/abs/1801.01973, January 2018.

[4] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. https://arxiv.org/abs/1512.01274,https://mxnet.incubator.apache.org/, December 2015.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. https://arxiv.org/abs/1406.2661, June 2014.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. https://arxiv.org/abs/1704.00028, March 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. https://arxiv.org/abs/1512.03385, December 2015.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. https://arxiv.org/abs/1706.08500, 2017.

[9] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. https://arxiv.org/abs/1608.06993/, August 2016.

[10] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. https://arxiv.org/abs/1312.6114, December 2013.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf, 2012.

[12] Yann Lecun, L?on Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. https://ieeexplore.ieee.org/document/726791, 1998.

[13] A. P. Majtey, P. W. Lamberti, and D. P. Prato. Jensen-Shannon divergence as a measure of distinguishability between mixed quantum states. https://arxiv.org/abs/quant-ph/0508138/, November 2005.

[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. https://arxiv.org/abs/1411.1784, 2014.

[15] John F. Nash. Equilibrium points in n-person games. http://www.pnas.org/content/36/1/48/, 1950.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. https://arxiv.org/abs/1511.06434, 2015.

[17] Luis Serrano. Shannon entropy, information gain, and picking balls from buckets. https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4/. Accessed Nov 6, 2017.

[18] J. Shlens. Notes on Kullback-Leibler Divergence and Likelihood. https://arxiv.org/abs/1404.2000/, April 2014.

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556/, 2014.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. https://arxiv.org/abs/1409.4842/, 2014.