Обзор статьи «Assessing Generative Models via Precision and Recall»

Авторы статьи: Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, Sylvain Gelly
https://arxiv.org/abs/1806.00035

Google Brain

Автор обзора: Иванин Евгений, 695

Целью данного обзора является познакомить читателя с основными результатами, которых удалось достичь ребятам из Google Brain. Если читателю требуется формализм, формулы и доказательства теорем, то во время чтения ему стоит обращаться к первоисточнику, где авторы статьи подробно расписали свое исследование на привычном нам математическом языке. Приятного чтения!

General Adversarial Networks

Прежде чем погрузиться в изучение описанной в статье проблемы, нужно убедиться, что читатель понимает основные понятия, которые фигурируют на протяжении всего исследования. Изложенный в статье материал намного легче воспринимается на примере, поэтому знакомьтесь - Generative Adversarial Networks!

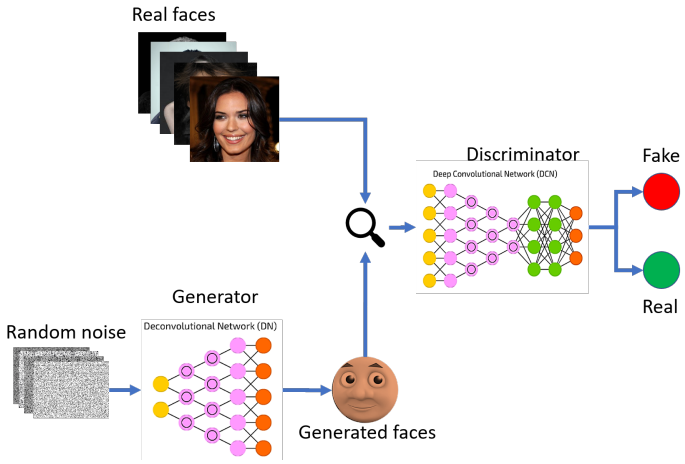


Рис. 1: Архитектура Generative Adversarial Network

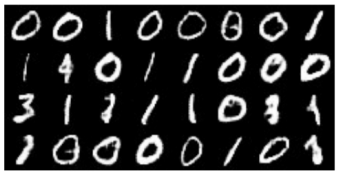
На первый взгляд кажется сложно, хотя на самом деле в базовой архитектуре GAN лежит простая и красивая идея. Для чего вообще людям понадобились GAN, какую задачу они решают? Дело в том, что в определенный момент люди захотели научить компьютер генерировать изображения, которые очень похожи на уже имеющиеся картинки. Например, GAN умеют создавать фотографии несуществующих котиков, только для этого предварительно модели нужно скормить большое число изображений с реальными котиками.

Рис. 2: Изображения котов, созданные генеративной моделью

Давайте разберемся, как работает этот генератор котиков. Архитектура GAN состоит из двух главных компонент: генератор и дискриминатор (Рис.1). Задача генератора заключается в том, чтобы преобразовывать случайный шум в изображение - то самое, которое мы хотим видеть в качестве результата работы модели. Для понимания не очень важно знать, как конкретно реализован генератор - читатель ничего не потеряет, если будет представлять абстрактную нейросеть. Задача дискриминатора заключается в том, чтобы различать фейковые изображения (созданные генератором) от настоящих. Он принимает на вход изображение и возвращает метку класса: 1 - если изображение настоящее, 0 - если изображение создано генератором, таким образом решая задачу бинарной классификации. Компоненты GAN обучаются до тех пор, пока генератор не научится обманывать дискриминатора. В итоге, модель умеет создавать картинки, которые выглядят так же как реальные изображения, используемые при обучении. Если читатель хочет разобраться в деталях работы GAN, то у него есть возможность изучить отличные статьи: оригинальная статья про архитектуру, статья для общего понимания механизма работы, статья про применение GAN в реальной жизни

Популярные метрики

Предположим, что мы обучили GAN. А потом еще одну. И еще одну. Нам нужно научиться сравнивать полученные модели между собой, соответственно, возникает необходимость ввести метрики качества. Наиболее популярными из используемых метрик являются Inception Score (IS) и Fréchet Inception Distance (FID). Разберемся, как они работают. Inception Score оценивает как разнообразие сгенерированных изображений, так и его качество. Разберемся на конкретном примере:



а) Маленькое разнообразие, хорошее качество

б) Большое разнообразие, плохое качество

Рис. 3: Пример моделей, обученных на CIFAR-10

Оба изображения созданы при помощи генеративной модели. Разнообразие представляет собой долю объектов, которые модель умеет генерировать, а качество по сути является мерой отсутствия артефактов, не имеющих отношения к реальным объектам. Для вычисления IS используется Google Inception Classifier - модель, которая принимает изображение и возвращает вероятности принадлежности объекта на изображении к какому-либо классу. Если генератор выдал хорошее качество, то вероятность принадлежности к конкретному классу должна быть высокой, так как этому случаю соответствует бОльшая уверенность в классификации. Если генератор выдает разнообразные изображения, то сумма распределений по каждому объекту должна быть равномерной. Иными словами, классы должны быть равнозначны. Собственно, в этом и заключается идея данной метрики. Статья об Inception Score
FID оценивает реалистичность изображений при помощи измерения «расстояния» между реальными объектами и сгенерированными моделью. Для вычисления метрики множество реальных изображений отображается в пространство признаков при помощи слоя Inception Net. Та же процедура проделывается для сгенерированных изображений. Суть в том, что выходы данного слоя рассматриваются как объекты непрерывного многомерного Гауссовского распределения. Таким образом,

мы можем оценить параметры данного распределения для реальных и фейковых изображений. Формула для подсчета данной метрики довольно интуитивна и представлена на Рис.4. Подробнее: Статья о FID

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

Рис. 4: Формула подсчета FID

Постановка задачи

Теперь читатель знаком с основными понятиями и мы можем перейти непосредственно к содержанию статьи. Последние достижения в области использования генеративных моделей вызывают повышенный интерес к поиску подходящих средств оценки их качества. Хорошим примером служат статистические подходы, основанные на формализации задачи на языке теории вероятностей. Проблема заключается в том, что наиболее популярные метрики, которыми являются FID и IS, не позволяют различать различные типы ошибок, которые совершает модель. Это хорошо видно из Рис.5, изображения сверху и изображения снизу сгенерированы разными моделями, имеющими одинаковые метрики качества IS и FID.

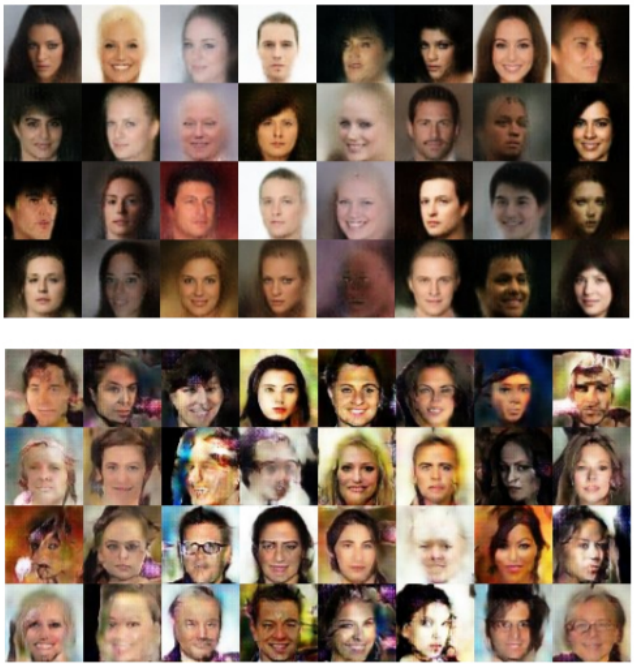


Рис. 5: Пример моделей, обученных на CelebFaces

Тем не менее мы видим, что модель сверху создает гораздо более качественные изображения, но менее разнообразные - наблюдается слишком много темных и размытых артефактов. Модель справа умеет генерировать более вариативные картинки, но их качество оставляет желать лучшего - некоторые лица выглядят слишком нереалистично. Задача, поставленная перед авторами статьи, состоит в том, чтобы научиться различать данные типы ошибок. В качестве решения предложено определить понятия Precision и Recall для контекста генеративных моделей. Использование данного инструмента позволяет сравнивать модели отдельно по каждой из этих метрик, таким образом давая экспериментатору возможность самому выбирать, контроль какого типа ошибок для него приоритетен.

Precision и Recall

Разберемся, как авторы статьи переопределяют известные понятия Precision и Recall. Введем обозначения: P - реальное распределение объектов, Q - распределение, полученное при помощи обучения модели. Иными словами, реальные изображения появляются в жизни из распределения P, а модель (в частности GAN) генерирует их из распределения Q. Ключевая интуиция заключается в том, что Precision должен оценивать долю объектов из Q, которая может быть сгенерирована как часть объектов из P. В свою очередь Recall измеряет, как много объектов из P может быть сгенерировано при помощи Q. Разберем на примере (Рис.6).

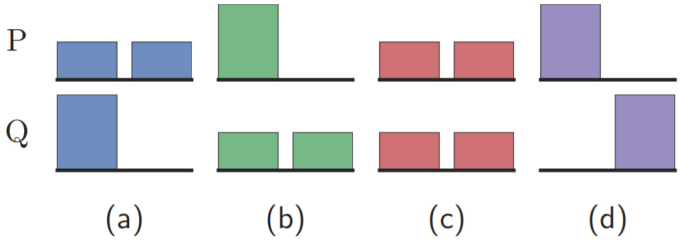


Рис. 6: Пример распределений P и Q

Для простоты будем считать, что P и Q имеют бинарный носитель:

a) если распределение P является бимодальным, а Q охватывает только одно из значений, то мы получим идеальный Precision и низкий Recall.

b) в противоположном случае мы получаем высокий Recall, но низкий Precision, так как модель генерирует значения, которые не входят в реальное распределение (например, нереалистичные изображения).

c) если $P = Q$, то получаем идеальные Precision и Recall.

d) если $P \cap Q = \varnothing$, то получаем отвратительные Precision и Recall. Это значит, что модель уверенно генерирует изображения, не похожие на обучающую выборку.

Рассмотрим вывод формальных определений Precision и Recall, чтобы понять связь между интуитицией, лежащей в основе рассматриваемых метрик, и теоретической моделью. Обозначим $S = supp(P) \cap supp(Q)$ - непустое пересечение носителей распределений P и Q. Тогда распределение P может быть представлено как смесь двух распределений: P_S и $P_{\overline{S}}$, где $\overline{S}$ - дополнение к S. На данном этапе, чтобы проверить себя, читатель может подумать над тем, какой из случаев лучше:

a) S больше, чем $\overline{S}$

b) $\overline{S}$ больше, чем S

b) $\overline{S}$ и S равномощны

Нетрудно понять, что правильным ответом является вариант *a*, ведь все объекты, лежащие в $\overline{S}$, могут быть объяснены лишь одним из распределений. Иными словами, если объект объясняется реальным распределением (например, реальное изображение котика), то он не может быть сгенерирован из Q. Аналогично, если объект объясняется сгенерированным распределением (фейковое изображение котика), то он не может быть сгенерирован из P. В этом случае сгенерированный котик скорее всего и не котик вовсе.
Более формально, для некоторых $\alpha, \beta \in (0, 1]$ представим

$$P = \beta \cdot P_S + (1 - \beta) \cdot P_{\overline{S}} \text{ и } Q = \alpha \cdot Q_S + (1 - \alpha) \cdot Q_{\overline{S}} \qquad (1)$$

Данное разложение имеет естественную интерпретацию. $P_{\overline{S}}$ - часть распределения P, которая не может быть сгенерирована из Q, поэтому β можно рассматривать как Recall. Аналогично, $Q_{\overline{S}}$ - часть Q, которая не может быть в P. Соответственно, α можно рассматривать в качестве Precision. Собственно, в этом и заключается основная математическая интуиция, заложенная в определение данных метрик. Авторы статьи предложили эффективный алгоритм вычисления данных метрик, ознакомиться с которым читатель может тут.

Применение

Для проведения экспериментов авторы рассмотрели три популярных датасета:

- MNIST - изображения с рукописными цифрами
- CIFAR-10 - изображения кораблей, машин, животных и т.д. (10 классов)
- Fashion MNIST - изображения с элементами одежды

Все датасеты содержат 10 сбалансированных классов, мы для примера рассмотрим один из них - CIFAR-10. Положим распределение P основано на первых пяти классах. Для каждого фиксированного $i = 1, ...10$ сгенерируем Q_i, состоящее из объектов первых i классов. Понятно, что с ростом i распределение Q_i должно покрывать больше классов, то есть Recall должен расти. Но когда i будет больше пяти, Q_i начнет содержать классы, которых нет в P. Соответственно, Precision должен начинать падать, в отличие от Recall - ведь мы уже покрыли распределение Q_i все классы из P. Также нетрудно догадаться, что оптимальным распределением является Q_5, ведь оно покрывает все классы из P и только их.

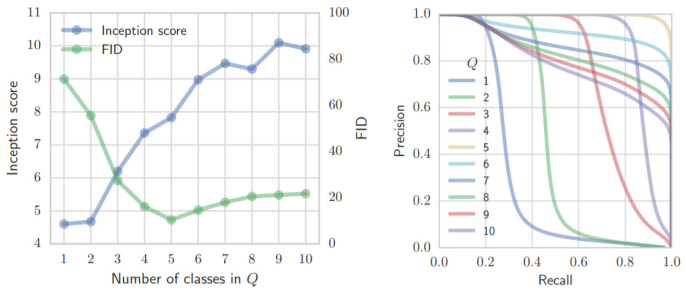


Рис. 7: Сравнение метрик IS, FID и PR

Обратимся к Рис.7. Слева можно увидеть значения метрик Inception Score и FID в зависимости от количество классов в Q (речь идет о процедуре, описанной выше). IS в общем возрастает, то есть наилучше значение метрики будет достигнуто в случае высокого Recall и низкого Precision. В свою очередь, FID оказалась более чувствительной к различным типам ошибок - однако различать их она не умеет. График справа показывает зависимость Precision и Recall от количества классов в Q. Как и планировалось, добавление классов $6, ...10$ привело к ухудшению Precision. При этом недостаток классов (случай $i < 5$) приводит к потерям в значении Recall. Обратим внимание на то, что в случаях $i = 4$ и $i = 6$ значение FID одинаково, при этом значения PR различны. Таким образом, авторы экспериментально подтвердили, что Precision и Recall Способны различать различные типы ошибок.

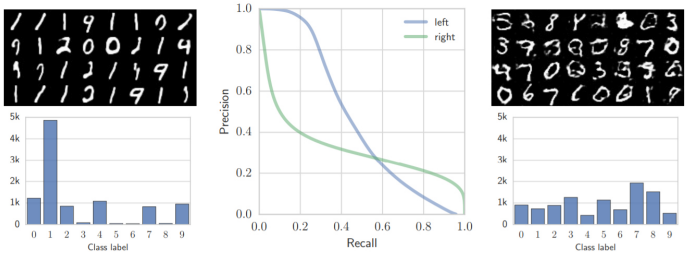


Рис. 8: Сравнение двух моделей, обученных на MNIST

На Рис.8 отображены результаты сравнения двух GAN, обученных на MNIST. Обе модели имеют одинаковый $FID = 49$. Модель слева создает высококачественные объекты, но покрывает лишь подмножество реальных возможных объектов. В свою очередь, модель справа генерирует изображения низкого качества, но зато покрывает все классы. Гистограммы показывают соответствующие распределения на классах. Из графика по центру видно, что Precision ниже для модели, которая генерирует менее качественные картинки, а Recall ниже для модели, которая создает лишь подмножество объектов из истинного распределения. Таким образом, авторы статьи экспериментально показали, что введение предложенных ими метрик действительно имеет смысл для практического применения.

Идеи для дальнейшего исследования

Для дальнейшего исследования можно рассмотреть зависимость метрик Precision и Recall от ресэмплинга обучающей выборки. Часто в задачах, связанных с обучением нейронных сетей, может оказаться недостаточно много данных. Одним из решений данной проблемы является ресэмплинг - повороты, кадрирования и другие преобразования имеющихся в обучающей выборке данных. Интересно было бы узнать, какие конкретно преобразования влияют на значения Precision и Recall и как.
Также можно исследовать влияние генерируемого шума на метрики. Как было отмечено выше, в архитектуре GAN на вход генератору подается случайный шум. Интересно было бы проверить гипотезу о том, что Precision (или Recall) зависит от генерируемого шума. Например, шуму с бОльшим отклонением от среднего соответствовал бы бОльший Recall (это, конечно, может быть не правдой). Если в ходе исследования удалось бы задетектировать интересные зависимости, то это позволило бы контролировать тип ошибки на этапе генерации объектов.

Заключение

Численно оценить качество генеративной модели - сложная задача, которая имеет большую важность на практике. Авторы рассматриваемой статьи показали, что существующие метрики не способы различать типы ошибок, которые делает модель. В качестве альтернативы было предложено ввести новые понятия Precision и Recall для распределений. Авторы продемонстрировали теоретическую и практическую значимость введенных понятий, а также разработали эффективный алгоритм вычисления данных метрик. И это здорово.
Область изучения генеративных моделей имеет огромные просторы для исследований, а значимость выбора подходящих методов измерения качества трудно переоценить. Авторам статьи удалось внести свой вклад в эту область. Возможно, именно эти результаты станут ключом к решению важных задач, над которыми исследователи ломают голову.