

ВЕЩЕСТВЕННЫЕ (МЕДИЦИНСКИЕ) РЯДЫ РОДА ВРЕМЕННОГО РЯДА ЦИОН С ПОВТОРЯЮЩИМСЯ УСЛОВНЫМ ГАНСОМ

Stephanie L. Hyland
ETH Zurich, Швейцария

Трехинституциональная учебная программа по вычислительной биологии и медицине, Weill
Cornell Medical stephanie.hyland@inf.ethz.ch

Кристобаль
Эстебан ETH Цюрих,
Швейцария cristobal.esteban@inf.ethz.ch

Gunnar Rätsch
ETH Цюрих, Швейцария
raetsch@inf.ethz.ch

АННОТАЦИЯ

Генеративно-состязательные сети (GAN) продемонстрировали замечательный успех в качестве основы для обучения моделей для получения реалистично выглядящих данных. В этой работе мы предлагаем рекуррентную GAN (RGAN) и рекуррентную условную GAN (RCGAN) для создания реалистичных многомерных временных рядов с действительными значениями с упором на их применение к медицинским данным. RGAN используют рекуррентные нейронные сети (RNN) в генераторе и дискриминаторе. В случае RCGAN обе эти RNN зависят от вспомогательной информации. Мы демонстрируем наши модели в наборе игрушечных наборов данных, где мы показываем визуально и количественно (используя вероятность выборки и максимальное среднее несоответствие), что они могут успешно генерировать реалистичные временные ряды. Мы также описываем новые методы оценки для GAN, где мы создаем синтетический помеченный обучающий набор данных и оцениваем на реальном тестовом наборе производительность модели, обученной на синтетических данных, и наоборот. С помощью этих метрик мы показываем, что RCGAN могут генерировать данные временных рядов, полезные для контролируемого обучения, с лишь незначительным ухудшением производительности на реальных тестовых данных. Это демонстрируется классификацией цифр из «сериализованного» MNIST и обучением системы раннего предупреждения на наборе медицинских данных о 17 000 пациентов из отделения интенсивной терапии. Далее мы обсудим и проанализируем проблемы конфиденциальности, которые могут возникнуть при использовании RCGAN для создания реалистичных синтетических медицинских данных временных рядов, и продемонстрируем результаты дифференцированно частного обучения RCGAN.

1 ВВЕДЕНИЕ

Доступ к данным является одним из узких мест в разработке решений машинного обучения для конкретных проблем предметной области. Доступность стандартных наборов данных (с соответствующими задачами) помогла расширить возможности обучающих систем при выполнении нескольких задач. Однако в других областях, таких как медицина, прогресс, похоже, отстает. Заманчиво предположить, что задачи в медицине просто сложнее — данные более сложные, более зашумленные, проблемы прогнозирования менее четко определены. Несмотря на это, нехватка данных, доступных исследователям, препятствует сравнению моделей, воспроизводимости и, в конечном счете, научному прогрессу. Однако из-за очень конфиденциального характера медицинских данных доступ к ним обычно строго контролируется или требует сложной и, вероятно, несовершенной деидентификации. Таким образом, мотивация этой работы состоит в том, чтобы использовать и развивать структуру генеративно-состязательных сетей (GAN) для создания реалистичных синтетических медицинских данных. Эти данные можно было бы совместно использовать и публиковать, не опасаясь за конфиденциальность, или даже использовать для расширения или обогащения аналогичных наборов данных, собранных в разных или меньших когортах пациентов. Более того, создание системы, способной синтезировать реалистичные медицинские данные, подразумевает моделирование процессов, которые генерируют такую информацию, и, следовательно, может представлять собой первый шаг к разработке нового подхода к созданию прогностических систем в медицинских средах.

Помимо полезности для сообщества исследователей машинного обучения, такой инструмент может принести пользу медицинскому сообществу для использования в учебных симуляторах. В этой работе мы сосредоточимся на синтезе вещественных

* Авторы внесли равный вклад.

данные временного ряда из отделения интенсивной терапии (ОИТ). В отделениях интенсивной терапии врачам приходится принимать поспешные решения в условиях дефицита времени, когда они не могут позволить себе медлить. Использование симуляций для обучения врачей уже стало стандартом в медицинском обучении, но эти симуляции часто основаны на разработанных вручную правилах и физическом реквизите. Таким образом, модель, способная генерировать разнообразные и реалистичные ситуации в отделении интенсивной терапии, может найти немедленное применение, особенно когда она дает возможность обуславливать лежащие в основе «состояния» пациента.

Успех GAN в создании реалистичных изображений (Radford et al., 2015; Ledig et al., 2016; Gauthier, 2014; Reed et al., 2016) предполагает их применимость для этой задачи, однако ограниченная работа использовала их для генерации данных временных рядов. Кроме того, оценка GAN остается в значительной степени нерешенной проблемой, поскольку исследователи часто полагаются на визуальную оценку сгенерированных примеров, что является непрактичным и неподходящим для многомерных медицинских временных рядов.

Основные вклады этой работы:

1. Демонстрация метода генерации последовательностей с действительными значениями с использованием состязательного обучения.
2. Демонстрация новых подходов к оценке GAN.
3. Создание синтетических медицинских временных рядов данных.
4. Эмпирический анализ конфиденциальности как GAN, так и дифференциальных частных GAN.

2 СВЯЗАННЫЕ РАБОТЫ

С момента своего создания в 2014 году (Goodfellow et al., 2014) структура GAN привлекла значительное внимание исследовательского сообщества, и большая часть этой работы была сосредоточена на создании изображений (Radford et al., 2015; Ledig et al., 2016; Готье, 2014; Рид и др., 2016). Примечательно, что (Choi et al., 2017) разработали GAN для создания синтетических наборов данных электронных медицинских карт (EHR). Эти EHR содержат двоичные и счетные переменные, такие как коды счетов по МКБ-9, коды лекарств и процедур. Их внимание к дискретным данным и созданию моментальных снимков пациента дополняет наше внимание к реальным временным рядам. Будущая работа может объединить эти подходы для создания мультимодальных синтетических медицинских временных рядов данных.

Большая часть последовательной генерации данных с помощью GAN была сосредоточена на дискретных токенах, полезных для обработки естественного языка (Yu et al., 2016), где для обучения GAN используется альтернативный подход, основанный на обучении с подкреплением (RL). Нам известна только одна предварительная работа с использованием GAN для генерации последовательностей с непрерывными значениями, целью которой является создание полифонической музыки с использованием GAN с генератором и дискриминатором LSTM (Могрен, 2016). Основные отличия связаны с архитектурой: мы не используем двунаправленный дискриминатор, а выходные данные генератора не возвращаются в качестве входных данных на следующем временном шаге. Кроме того, мы вводим также условную версию этого рекуррентного GAN.

Условные GAN (Mirza & Osindero, 2014; Gauthier, 2014) обуславливают модель дополнительной информацией и, следовательно, позволяют нам управлять процессом генерации данных. Этот подход в основном использовался для задач генерации изображений (Radford et al., 2015; Mirza & Osindero, 2014; Antipov et al., 2017). Недавно архитектуры условной GAN также использовались в обработке естественного языка, включая перевод (Янг и др., 2017) и создание диалогов (Ли и др., 2017), где ни одна из них не использует RNN в качестве предпочтительного выбора для дискриминатора. и, как упоминалось ранее, для обучения моделей используется подход RL из-за дискретного характера данных.

В этой работе мы также представляем некоторые новые подходы к оценке GAN, используя возможности сгенерированных синтетических данных для обучения контролируемых моделей. Подобным образом подход к полуконтролируемому обучению на основе GAN был представлен в (Salimans et al., 2016). Однако наша цель — генерировать данные, которые можно использовать для обучения моделей задачам, неизвестным на момент обучения GAN.

Мы кратко рассмотрим использование дифференциально частного стохастического градиентного спуска (Abadi et al., 2016) для создания RGAN с более сильными гарантиями конфиденциальности, что особенно актуально для конфиденциальных медицинских данных. Альтернативным методом может быть использование подхода PATE (Papernot et al., 2016) для обучения дискриминатора. В этом случае вместо того, чтобы вводить шум в градиенты (как в (Abadi et al., 2016)), классификатор учеников обучается предсказывать зашумленные голоса ансамбля учителей, каждый из которых обучен на непересекающихся наборах данных.

3 МОДЕЛИ: ПОВТОРЯЮЩАЯСЯ ГАН И ПОВТОРЯЮЩАЯСЯ УСЛОВНАЯ ГАН

Модель, представленная в этой работе, следует архитектуре обычной GAN, где и генератор, и дискриминатор заменены рекуррентными нейронными сетями. Поэтому мы представляем рекуррентную GAN (RGAN), которая может генерировать последовательности данных с действительными значениями, и рекуррентную условную GAN (RCGAN), которая может генерировать последовательности данных с действительными значениями при некоторых условных входных данных. Как показано на рисунке 1a, генератор RNN принимает разные случайные начальные значения на каждом временном шаге, а также дополнительные входные данные, если мы хотим обусловить сгенерированную последовательность дополнительными данными. На рисунке 1b показано, как дискриминатор RNN принимает сгенерированную последовательность вместе с дополнительными входными данными, если это RCGAN, и производит классификацию как синтетическую или реальную для каждого временного шага входной последовательности.

В частности, дискриминатор обучен минимизировать среднюю отрицательную кросс-энтропию между его предсказаниями на временной шаг и метками последовательности. Если мы обозначим через $RNN(X)$ вектор или матрицу, содержащую T выходных данных RNN, получающих последовательность T векторов $\{x_t\}_{t=1}^T$ и a через $CE(a, b)$ среднюю кросс-энтропию между последовательностями a и b , тогда потери дискриминатора для пары $\{X_n, y_n\}$ (с $X_n \in \mathbb{R}^{T \times A}$ и $y_n \in \{1, 0\}^T$)

$$\text{есть: } D_{\text{loss}}(X_n, y_n) = -CE(RNND(X_n), y_n)$$

Для реальных последовательностей y_n представляет собой вектор из единиц или нулей для синтетических последовательностей. В каждой обучающей мини-партии дискриминатор видит как настоящие, так и синтетические последовательности.

Таким образом, задача генератора состоит в том, чтобы «обмануть» дискриминатор, чтобы он классифицировал его выходные данные как истинные, т. е. вектор единиц (мы пишем как 1);

$$\text{Блесс } (Z_n) = D_{\text{loss}}(RNNG(Z_n), 1) = -CE(RNND(RNNG(Z_n)), 1)$$

Здесь Z_n — последовательность T точек $\{z_t\}_{t=1}^T$, выбранных независимо из скрытого/шумового пространства Z , таким образом, $Z_n \in \mathbb{R}^{T \times A}$; $RNNG$ и $RNND$ — функции, принимающие на вход последовательности для дальнейшей обработки. Мы не показали каких-либо доказательств того, что эти функции являются функциями потерь.

В этой работе архитектурой, выбранной как для дискриминаторных, так и для генераторных RNN, является LSTM (Hochreiter & Schmidhuber, 1997).

В условном случае (RCGAN) входные данные каждой RNN дополняются некоторой условной информацией c_n (скажем, для выборки n) путем конкатенации на каждом временном шаге; $x_{nt} = [x_{nt}; c_n]$

$$z_{nt} = [z_{nt}; c_n]$$

Таким образом, RNN не может игнорировать условную информацию из-за забывания.

Многообещающие исследования альтернативных целей GAN, таких как GAN Вассерштейна (Arjovsky et al., 2017; Gulrajani et al., 2017), к сожалению, не находят легкого применения в RGAN в наших экспериментах. Применение ограничения Липшица к RNN является темой для дальнейших исследований, но этому может помочь использование унитарных RNN (Arjovsky et al., 2016; Hyland & Rätsch, 2017).

Все модели и эксперименты были реализованы на питоне с помощью scikit-learn (Pedregosa et al., 2011) и Tensorflow (Abadi et al., 2015), а код доступен в общедоступном репозитории git: ANON.

3.1 ОЦЕНКА

Оценка производительности GAN является сложной задачей. Как показано в (Theis et al., 2015) и (Wu et al., 2016), оценка правдоподобия с помощью оценок окна Парзена (Wu et al., 2016) или иным образом может быть обманчивой, а потери генератора и дискриминатора не легко соответствуют «визуальному качеству». Это расплывчатое понятие качества лучше всего оценивается человеческим судьей, но это непрактично и дорого. В области визуализации для помощи в оценке были разработаны такие показатели, как начальный показатель (Salimans et al., 2016), а Mechanical Turk использовался для распределения человеческого труда. Однако в случае последовательных данных с действительным знаком не всегда легко или даже возможно визуально оценить сгенерированные данные. Например, сигналы отделения интенсивной терапии, с которыми мы работаем в этой статье, немедицинскому эксперту могут показаться совершенно случайными.

Поэтому в этой работе мы начнем с демонстрации нашей модели с рядом игрушечных наборов данных, которые можно оценить визуально. Затем мы используем набор количественных методов (описание ниже), которые можно использовать в качестве индикатора качества данных.

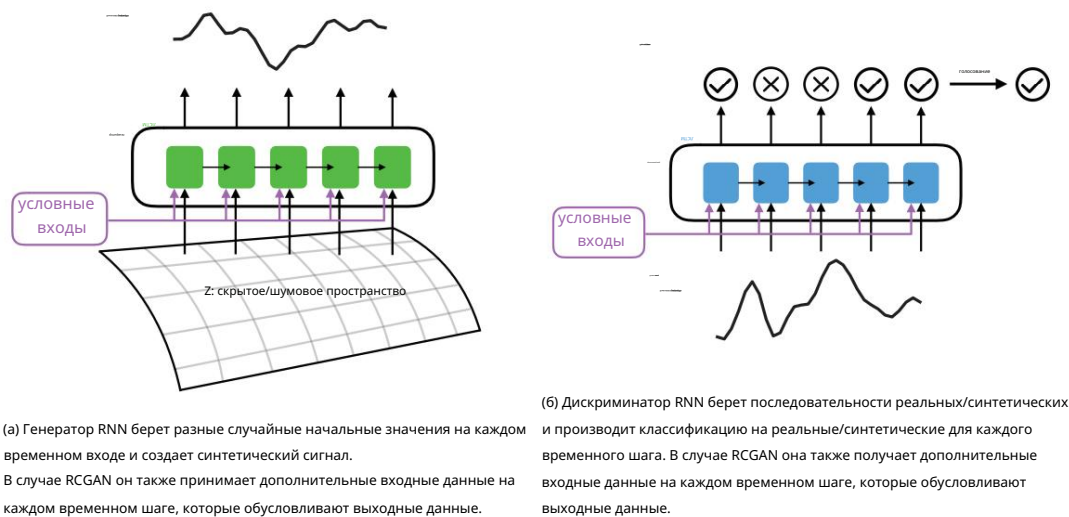


Рисунок 1: Архитектура рекуррентной модели GAN и условно рекуррентной GAN.

3.1.1 МАКСИМАЛЬНОЕ СРЕДНЕЕ НЕСООТВЕТСТВИЕ

Мы считаем GAN успешной, если она неявно изучает распределение истинных данных. Мы оцениваем это, изучая образцы, которые он генерирует. Это идеальная настройка для максимальной средней невязки (MMD) (Gretton et al., 2007), и она использовалась в качестве цели обучения для сетей согласования генеративных моментов (Li et al., 2015). MMD спрашивает, были ли два набора выборок — один из GAN и один из истинного распределения данных, например, — были сгенерированы одним и тем же распределением. Это делается путем сравнения статистики выборок. На практике мы рассматриваем квадрат разности статистики между двумя наборами выборок (MMD2) и заменяем внутренние продукты между (функциями) двух выборок ядром. Для ядра $K: X \times Y \rightarrow \mathbb{R}$ и выборок $\{x_i\}_{i=1}^n$ и $\{y_j\}_{j=1}^m$ несмещенная оценка MMD2 такова:

$$\text{MMD}^2 = \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j=1}^n K(x_i, x_j) - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m K(x_i, y_j) + \frac{1}{m(m-1)} \sum_{i=1}^m \sum_{j=1}^m K(y_i, y_j)$$

Определение соответствующих ядер между временными рядами является областью активных исследований. Однако большая часть проблем возникает из-за необходимости согласования временных рядов. В нашем случае сгенерированные и реальные образцы уже выровнены нашей фиксацией оси «время». Затем мы решили рассматривать наши временные ряды как векторы (или матрицы в многомерном случае) для сравнений и использовать ядро радиальной базисной функции (RBF), используя норму квадрата ℓ^2 или норму Фробениуса между векторами / матрицами; $K(x, y) = \exp(-\frac{\|x - y\|^2}{2\sigma^2})$. Мы выбрали ядро радиальной базисной функции (RBF) для теста MMD между двумя распределениями (Sutherland et al., 2016); $\hat{t} = \text{MMD}$, где V — асимптотическая дисперсия оценки MMD2. Мы делаем это, используя ядро радиальной базисной функции (RBF) для теста MMD2, а также для обучения ядра. Следуя (Sutherland et al., 2016), мы определяем смешанное ядро как сумму ядер RBF с двумя разными σ s, которые мы оптимизируем одновременно. Мы считаем, что MMD2 более информативен, чем потеря генератора или дискриминатора, и хорошо коррелирует с качеством, оцененным визуализацией.

3.1.2 ПОЕЗДКА НА СИНТЕТИКЕ, ТЕСТ НА РЕАЛЬНОМ (TSTR)

Мы предлагаем новый метод оценки выходных данных GAN, когда контролируемая задача может быть определена в области обучающих данных. Мы называем это «Тренируйся на синтетическом, тестируй на реальном» (TSTR). Проще говоря, мы используем набор данных, сгенерированный GAN, для обучения модели, которая затем тестируется на наборе реальных примеров. Это требует, чтобы сгенерированные данные имели метки — мы можем либо предоставить их условной GAN, либо использовать стандартную GAN для их создания в дополнение к функциям данных. В этой работе мы выбрали первое, как мы опишем ниже. За использование GAN для обмена синтетическими «деидентифицированными»

данные, эта оценочная метрика идеальна, потому что она демонстрирует возможность использования синтетических данных для реальных приложений. Мы представляем псевдокод для этой стратегии оценки GAN в алгоритме 1.

Алгоритм 1 (TSTR) Обучение на синтетике, проверка на реальном

```

1: поезд, тест = разделение (данные)
2: дискриминатор, генератор = поезд_ГАН (поезд) 3: с метками из поезда:
4: синтетический = генератор.генерировать_синтетический (метки) 5:
классификатор = поезд_классификатор (синтетический, метки)

6: Если доступен набор проверки, дополнительно оптимизируйте GAN по сравнению с
производительностью классификатора. 7: с метками и функциями из теста: 8: предсказания =
classifier.predict(features)
9: TSTR_score = оценка (прогнозы, метки)

```

Тренировка на реальном, тестирование на синтетическом (TRTS): Подобно методу TSTR, предложенному выше, мы можем рассмотреть обратный случай, называемый «Обучение на реальном, тест на синтетическом» (TRTS). В этом подходе мы используем реальные данные для обучения контролируемой модели набору задач. Затем мы используем RCGAN для создания синтетического набора тестов для оценки. В случае (как для MNIST), когда истинный классификатор достигает высокой точности, это служит оценкой способности RCGAN генерировать убедительные примеры меток и реалистичности генерируемых им признаков. Однако, в отличие от настройки TSTR, если в GAN произойдет сбой режима, производительность TRTS не ухудшится соответствующим образом, поэтому мы считаем TSTR более интересной оценкой.

4 УЧИТЬСЯ СОЗДАВАТЬ РЕАЛИСТИЧНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ

Чтобы продемонстрировать способность модели генерировать «реалистично выглядящие» последовательности в контролируемых условиях, мы рассмотрим несколько экспериментов с синтетическими данными. В следующих экспериментах, если не указано иное, синтетические данные состоят из последовательностей длиной 30. В этом разделе мы сосредоточимся на безусловной модели RGAN.

4.1 СИНУСОУС

Качество сгенерированных синусоидальных волн легко подтверждается визуальным осмотром, но, варьируя амплитуды и частоты реальных данных, мы можем создать набор данных с нелинейными вариациями. Мы генерируем волны с частотами в $[1, 0, 5, 0]$, амплитудами в $[0, 1, 0, 9]$ и случайными фазами между $[-\pi, \pi]$. В левой части рисунка 2а показаны примеры этих сигналов, как реальных, так и сгенерированных (хотя их трудно различить).

Мы обнаружили, что, несмотря на отсутствие ограничений для применения семантики в латентном пространстве (как в (Chen et al., 2016)), мы могли изменять частоту и фазу генерируемых выборок, изменяя латентные размеры, хотя представление не было «распутались», и одно измерение скрытого пространства повлияло на несколько аспектов сигнала.

На этом этапе мы попытались обучить рекуррентную версию вариационного автоэнкодера (VAE) (Kingma & Welling, 2013) с целью сравнения ее производительности с RGAN. Мы попробовали реализацию, предложенную в (Fabius & van Amersfoort, 2014), которая, возможно, является наиболее простым решением для реализации рекуррентного вариационного автоэнкодера (RVAE). Он состоит из замены кодера и декодера VAE на RNN, а затем использования последнего скрытого состояния кодировщика RNN в качестве закодированного представления входной последовательности. После выполнения трюка репараметризации полученное закодированное представление используется для инициализации скрытого состояния декодера RNN. Поскольку в этом простом наборе данных все последовательности имеют одинаковую длину, мы также попробовали альтернативный подход, в котором кодирование входной последовательности вычисляется как конкатенация всех скрытых состояний кодировщика RNN. Используя эти архитектуры, мы могли генерировать только синусоидальные волны с непостоянными амплитудами и частотами, качество которых явно уступало тем, которые производились RGAN. Исходный код для воспроизведения этих экспериментов включен в упомянутый ранее репозиторий git. Мы считаем, что этот подход нуждается в дальнейшем исследовании, особенно для задачи генерации

Точность
Реальная 0,988 ± 0,002
TRTS 0,988 ± 0,005

Таблица 1. Показатели, полученные сверточной нейронной сетью при: а) обучении и тестировании на реальных данных, б) обучении на синтетических данных и тестировании на реальных данных и в) обучении на реальных и тестировании на синтетических данных. Во всех случаях ранняя остановка и (в случае синтетических данных) выбор эпохи определялись с помощью проверочного набора.

помеченные данные, которые будут представлены позже в этой статье, чего нам пока не удалось сделать с помощью RVAE.

4.2 ГЛАДКИЕ ФУНКЦИИ

Синусоиды — это простые сигналы, легко воспроизводимые моделью. В нашем окончательном медицинском приложении мы хотим, чтобы модель воспроизводила сложные физиологические сигналы, которые могут не следовать простой динамике. Поэтому мы рассматриваем более сложную задачу изучения произвольных гладких сигналов. Гауссовские процессы предлагают метод выборки значений таких гладких функций. Мы используем ядро RBF с, чтобы указать GP с функцией среднего значения с нулевым значением. Затем мы рисуем 30 равноотстоящих образцов. Это составляет однократное взятие из многомерного нормального распределения с функцией ковариации, заданной ядром RBF, оцениваемым на сетке равноотстоящих точек. При этом мы указали именно то распределение вероятностей, сгенерированное истинными данными, что позволяет нам оценивать сгенерированные выборки в соответствии с этим распределением. Справа на рис. 2а показаны примеры (реальные и сгенерированные) этого эксперимента. Основная особенность реальных и сгенерированных временных рядов заключается в том, что они демонстрируют гладкость с локальными корреляциями, и это быстро фиксируется RGAN.

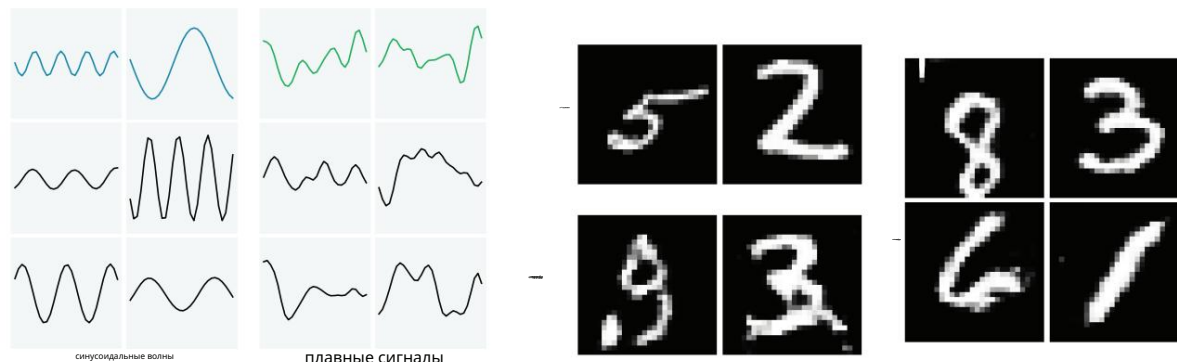
Поскольку у нас есть доступ к распределению данных, на рисунке 3 показано, как увеличивается средняя (логарифмическая) вероятность набора сгенерированных выборок при распределении данных во время обучения. Это несовершенная мера, поскольку она слепа к разнообразию сгенерированных выборок — часто наблюдаемому коллапсу моды или «Сценарию Helvetica» (Goodfellow et al., 2014) GAN — поэтому мы предпочитаем меру MMD2 (см. Рисунок 3). Тем не менее обнадеживает тот факт, что, хотя цель GAN не знает об основном распределении данных, вероятность сгенерированных выборок улучшается с обучением.

4.3 MNIST КАК ВРЕМЕННОЙ РЯД

Набор данных рукописных цифр MNIST повсеместно используется в исследованиях машинного обучения. Точность классификации цифр MNIST достаточно высока, чтобы считать проблему «решенной», а генерация цифр MNIST кажется почти тривиальной задачей для традиционных GAN. Однако последовательное создание MNIST выполняется реже (яркими примерами являются PixelRNN (Oord et al., 2016) и сериализация MNIST в литературе RNN с длинной памятью (Le et al., 2015)). Чтобы сериализовать MNIST, каждая цифра 28×28 образует 784-мерный вектор, который представляет собой последовательность, которую мы можем сгенерировать с помощью RGAN.

Это дает дополнительное преимущество в виде образцов, которые мы можем легко оценить визуально.

Чтобы упростить задачу и изучить способность RGAN генерировать многомерные последовательности, мы рассматриваем каждое изображение 28×28 как последовательность 28, 28-мерных выходных данных. Мы показываем два вида



(а) Примеры реальных (цветные, сверху) и сгенерированных (черные, две нижние линии) образцов.

(б) Слева сверху: реальные цифры MNIST. Слева внизу: нереалистичные цифры, сгенерированные в эпоху 27. Справа: цифры с минимальными искажениями, сгенерированные в эпоху 100.

Рисунок 2: RGAN может генерировать реалистичные примеры.

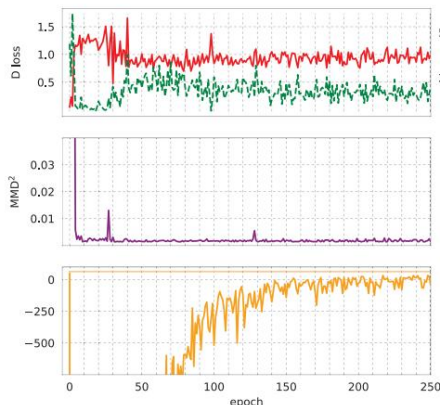


Рис. 3. Трасса потери генератора (пунктирная), дискриминатора (сплошная), оценка MMD2 и логарифмическая вероятность сгенерированных выборок при распределении данных во время обучения для RCGAN, генерирующего гладкие последовательности (результаты на рис. 2а).

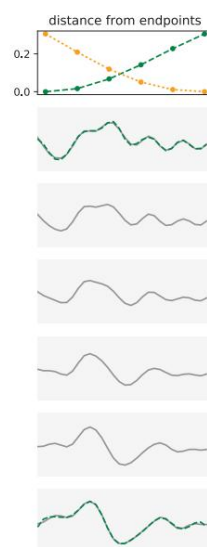


Рисунок 4. Обратное проецирование обучающих примеров в скрытое пространство и их линейная интерполяция дает плавное изменение выборки. свободное пространство. Верхний график показывает расстояние в пространстве выборки от верхнего (зеленый, пунктирный) образца до нижнего (оранжевый, пунктирный). Мерой расстояния является ядро RBF с пропускной способностью, выбранной как медиана попарного расстояния между обучающими выборками. Исходные обучающие примеры показаны пунктирными линиями на нижнем и втором сверху графиках.

поэкспериментируйте с этим набором данных. В первом мы обучаем RCGAN генерировать цифры MNIST таким последовательным образом. Рисунок 2b демонстрирует, насколько реалистично выглядят сгенерированные цифры.

Во втором эксперименте мы уменьшаем цифры MNIST до 14x14 пикселей и рассматриваем первые три цифры (0, 1 и 2). С помощью этих данных мы обучаем RCGAN и впоследствии выполняем оценки TSTR (и TRTS), описанные выше, для задачи классификации цифр. То есть для оценки TSTR мы генерируем синтетический набор данных с использованием GAN, используя в качестве входных данных реальные обучающие метки. Затем мы обучаем классификатор (сверточная нейронная сеть) на этих данных и оцениваем его эффективность на реальном тестовом наборе. И наоборот, для TRTS мы обучаем классификатор на реальных данных и оцениваем его на синтетическом тестовом наборе данных, сгенерированном GAN. Результаты этого эксперимента показаны в Таблице 1. Чтобы получить планки погрешностей для заявленной точности, мы обучали RCGAN пять раз с различными случайными инициализациями. Результат TSTR показывает, что RCGAN генерирует синтетические наборы данных, достаточно реалистичные для обучения классификатора, который затем достигает высокой производительности на реальных тестовых данных. Результат TRTS показывает, что синтетические примеры в тестовом наборе в высокой степени соответствуют своим меткам, учитывая, что точность классификатора, обученного на реальных данных, очень высока.

5 ОБУЧЕНИЕ СОЗДАВАТЬ РЕАЛИСТИЧНЫЕ ДАННЫЕ ОРИТ

Одной из основных целей этой статьи является построение модели, способной генерировать реалистичные наборы медицинских данных, в частности данные из отделений интенсивной терапии. С этой целью мы основывали нашу работу на недавно выпущенной базе данных Philips eICU1. Этот набор данных был собран в рамках программы телездоровоохранения для интенсивной терапии, предоставленной Philips.

Он содержит около 200 000 пациентов из 208 медицинских учреждений США, всего 224 026 866 записей, разделенных на 33 таблицы.

На основе этих данных мы фокусируемся на создании четырех наиболее часто регистрируемых, регулярно измеряемых переменных, измеряемых прикроватными мониторами: насыщение кислородом, измеряемое пульсоксиметром (SpO2), частота сердечных сокращений (ЧСС), частота дыхания (ЧД) и среднее артериальное давление (САД). В наборе данных eICU эти переменные измеряются каждые пять минут. Чтобы уменьшить длину рассматриваемых нами последовательностей, мы снижаем дискретизацию до одного измерения каждые пятнадцать минут, беря медианное значение в каждом окне. Это значительно ускоряет обучение нашей GAN на основе LSTM, сохраняя при этом соответствующую динамику данных.

В следующих экспериментах мы рассматриваем начало пребывания пациента в отделении интенсивной терапии, считая это критическим моментом в их уходе. Мы фокусируемся на первых 4 часах их пребывания, что дает 16 измерений каждой переменной. В то время как медицинские данные обычно содержат пропущенные значения, в этой работе мы обошли эту проблему, отбросив пациентов с пропущенными данными (после субдискретизации). После предварительной обработки данных таким образом мы получаем когорту из 17 693 пациентов. Наиболее ограничительным было требование о непропущенных значениях MAP, поскольку эти измерения проводятся инвазивно.

		SpO2 < 95	ЧСС < 70	ЧСС > 100
АУРОК	настоящий TCT	0,9587 ± 0,0004 0,9908 ± 0,0005 0,9919 ± 0,0002 0,88 ± 0,01 0,96 ± 0,01 0,95 ± 0,01		
АУПРК	настоящий TSTR случайный	0,9059 ± 0,0005 0,9855 ± 0,0002 0,9778 ± 0,0002 0,66 ± 0,02 0,90 ± 0,02 0,84 ± 0,03 0,16 0,26 0,18		

		OP < 13	OP > 20	КАРТА < 70	КАРТА > 110
АУРОК	настоящий TCT	0,9735 ± 0,0001 0,963 ± 0,001 0,9717 ± 0,0001 0,875 ± 0,007		0,86 ± 0,01 0,84 ± 0,02	0,960 ± 0,001 0,87 ± 0,04
АУПРК	настоящий TSTR случайный	0,9557 ± 0,0002 0,891 ± 0,001 0,9653 ± 0,0001 0,73 ± 0,02 0,26	0,8629 ± 0,0007 0,50 ± 0,06 0,1	0,82 ± 0,02 0,39	0,42 ± 0,07 0,05

Таблица 2: Производительность классификатора случайного леса для задач eICU при обучении на реальных данных и при обучении на синтетических данных (набор тестов реальный), включая базовые линии случайного прогнозирования. АУПРК обозначает площадь под кривой точности-отзыва, а AUROC обозначает площадь под кривой ROC. Курсив обозначает те задачи, производительность которых была оптимизирована при перекрестной проверке.

5.1 ЗАДАЧИ TSTR В EICU

Данные, генерируемые в отделении интенсивной терапии, сложны, поэтому немедицинским экспертам сложно выявить закономерности или тенденции на нем. Таким образом, один график, показывающий синтетические данные отделения интенсивной терапии, не даст достаточно информации. оценить его фактическое сходство с реальными данными. Поэтому мы оцениваем работу отделения интенсивной терапии. RCGAN с использованием метода TSTR.

Чтобы выполнить оценку TSTR, нам нужна контролируемая задача (или задачи) для данных. Актуальный вопрос в отделении интенсивной терапии — станет ли пациент «критическим» в ближайшем будущем — своего рода ранняя диагностика. система оповещения. Для модели, генерирующей динамические данные временных рядов, это особенно подходит, поскольку тенденции в данных, вероятно, являются наиболее предсказуемыми. Основываясь на наших четырех переменных (SpO2, ЧСС, ЧДД, САД), мы определить «критические пороги» и сгенерировать двоичные метки того, превысит ли эта переменная порог в ближайший час пребывания больного, то есть между 4 и 5 часом, так как мы считаем первые четыре часа «наблюдаются». Пороги показаны в столбцах Таблицы 2. Верхнего предела нет. пороговое значение SpO2, так как это процентное значение, где 100 % означает идеальные условия.

Что касается MNIST, мы «выбираем» метки, рисуя их из меток реальных данных, и используем их как входы кондиционирования для RCGAN. Это обеспечивает распределение меток в синтетическом наборе данных и реальный набор данных тот же, с учетом того факта, что метки не являются независимыми (пациент вряд ли одновременно страдать от повышенного и пониженного артериального давления).

Следуя алгоритму 1, мы обучаем RCGAN в течение 1000 эпох, сохраняя одну версию набора данных каждые 50 эпох. После этого мы оцениваем синтетические данные с помощью TSTR. Мы используем перекрестную проверку для выбора лучший синтетический набор данных, основанный на производительности классификатора, но поскольку мы предполагаем, что это может быть также используется для неизвестных задач, мы используем только 3 из 7 интересующих задач для выполнения этой перекрестной проверки шаг (обозначен курсивом в табл. 2). Результаты этого эксперимента представлены в таблице 2, которая сравнивает производительность, достигнутую классификатором случайного леса, который был обучен предсказывать 7 задачи, представляющие интерес, в одном эксперименте с реальными данными и в другом эксперименте с синтетическими сгенерированные данные.

6 ЗАПОМНЯЕТ ЛИ GAN ТРЕНИРОВОЧНЫЕ ДАННЫЕ?

Одним из объяснений производительности TSTR в MNIST и eICU может быть то, что GAN просто «запоминание» обучающих данных и их воспроизведение. Если бы это было так, то (потенциально частный) данные, используемые для обучения GAN, могут быть утеряны, что вызовет проблемы с конфиденциальностью при использовании в конфиденциальных медицинских данные. Важно, чтобы данные обучения для модели не могли быть восстановлены противником. В Кроме того, в то время как типичная цель GAN побуждает генератор воспроизводить обучающие примеры, мы надеемся, что он не подходит к тренировочным данным и изучит неявное распределение с пиком на обучающих примерах и пренебрежимо малы в других местах.

Чтобы ответить на этот вопрос, мы проводим три теста - один качественный, два статистических, изложенных в следующие подразделы. Хотя эти оценки носят эмпирический характер, мы по-прежнему считаем, что предлагаемые и протестированные меры оценки конфиденциальности могут быть очень полезны для быстрой проверки свойств конфиденциальности. данных, сгенерированных РГАН, но без надежных гарантий конфиденциальности.

6.1 СРАВНЕНИЕ РАСПРЕДЕЛЕНИЯ ОШИБОК РЕКОНСТРУКЦИИ

Чтобы проверить, не выглядят ли сгенерированные выборки «слишком похожими» на обучающую выборку, мы могли бы сгенерировать большое количество выборок и рассчитать расстояние до ближайшего соседа (в обучающей выборке) для каждой сгенерированной выборки. Мы могли бы сравнить распределение этих расстояний с теми, которые сравнивают сгенерированные образцы и протянутый тестовый набор. Однако, чтобы получить точную оценку расстояний, нам может понадобиться сгенерировать много выборок и, соответственно, вычислить много попарных расстояний.

Вместо этого мы намеренно генерируем ближайшего соседа для каждой тренировочной (или тестовой) заданной точки, а затем сравниваем расстояния.

Мы генерируем этих ближайших соседей, минимизируя ошибку реконструкции между целью y и сгенерированной точкой; $\text{Irecon}(y)(Z) = 1 - K(G(Z), y)$, где K — ядро RBF, описанное в разделе 3.1.1, с полосой пропускания σ , выбранной с использованием эвристики медианы (Bounliphone et al., 2015). Мы находим Z путем минимизации ошибки до приблизительной сходимости (когда норма градиента падает ниже порога).

Затем мы можем спросить, можем ли мы различить распределение ошибок реконструкции для разных входных данных. В частности, мы спрашиваем, можем ли мы различить распределение ошибок между обучающим набором и тестовым набором. Интуиция такова, что если модель «запомнила» обучающие данные, она добьется заметно более низких ошибок реконструкции, чем с тестовым набором. Мы используем двухвыборочный тест Колмогорова-Смирнова, чтобы проверить, различаются ли эти распределения. Для RCGAN, генерирующего синусоидальные волны, p -значение составляет $0,2 \pm 0,1$, для гладких сигналов — $0,09 \pm 0,04$, а для эксперимента MNIST, показанного на рис. 26, — $0,38 \pm 0,06$. Для MNIST, обученного с помощью RCGAN (результаты TSTR в таблице 1), p -значение составляет $0,57 \pm 0,18$. Мы пришли к выводу, что распределение ошибок реконструкции существенно не отличается между обучающими и тестовыми наборами ни в одном из этих случаев, и что модель, по-видимому, не склонна к восстановлению примеров обучающего набора.

6.2 ИНТЕРПОЛЯЦИЯ

Предположим, что модель имеет переоснащение (неявное распределение сильно достигает пика в области обучающих примеров), и большинство точек в скрытом пространстве сопоставляются с обучающими примерами (или рядом с ними). Если мы идем по гладкому пути в скрытом пространстве, мы ожидаем, что в каждой точке соответствующая сгенерированная выборка будет иметь вид «ближайшего» (в скрытом пространстве) обучающего примера с небольшими вариациями, пока мы не достигнем бассейна аттрактора другого обучающий пример, после чего образцы меняются местами.

Мы проверяем это качественно следующим образом: мы отбираем пару обучающих примеров (на глаз подтверждаем, что они не выглядят «слишком похожими»), а затем «обратно проецируем» их в скрытое пространство, чтобы найти ближайшую соответствующую скрытую точку, как описано выше. Затем мы линейно интерполируем между этими скрытыми точками и производим выборки из генератора в каждой точке. На рис. 4 показан пример этой процедуры с использованием набора данных «гладкая функция». Образцы показывают четкую постепенную разницу между начальной и входной последовательностями, в отличие от того, что мы могли бы ожидать, если бы модель просто запоминала данные.

6.3 СРАВНЕНИЕ СОЗДАННЫХ ВЫБОРОК

Вместо использования метода ближайших соседей (как в разделе 6.1) мы можем использовать тест трех выборок MMD (Bounliphone et al., 2015) для сравнения полного набора сгенерированных выборок. Поскольку X — сгенерированные выборки, Y и Z — тестовая и обучающая выборки соответственно, мы спрашиваем, меньше ли MMD между X и Y , чем MMD между X и Z . Тест построен таким образом, потому что мы ожидаем, что если модель запомнила обучающие данные, поэтому MMD между синтетическими данными и обучающими данными будет значительно ниже, чем MMD между синтетическими данными и тестовыми данными. В этом случае гипотеза о том, что $\text{MMD}(\text{синтетика}, \text{тест}) < \text{MMD}(\text{синтетика}, \text{поезд})$, будет ложной.

Поэтому мы проверяем (как в разделе 6.1), можно ли отвергнуть нашу нулевую гипотезу (о том, что модель не запомнила обучающие данные). Средние значения p , которые мы наблюдали, были: для данных eICU в разделе 5.1: $0,40 \pm 0,05$, для данных MNIST в разделе 4.3: $0,47 \pm 0,16$, для синусоидальных волн: $0,41 \pm 0,07$, для гладких сигналов: $0,07 \pm 0,04$ и для эксперименты MNIST RCGAN с более высоким разрешением в разделе 4: $0,59 \pm 0,12$ (до поправки на проверку множественных гипотез). Мы пришли к выводу, что мы не можем отвергнуть нулевую гипотезу о том, что MMD между синтетическим набором и тестовым набором не больше, чем MMD между синтетическим набором и обучающим набором, что указывает на то, что синтетические образцы не выглядят более похожими на обучающий набор, чем они делают с тестовым набором.

7 ТРЕНИРОВОЧНЫХ ОРГАНОВ С ДИФФЕРЕНЦИАЛЬНОЙ КОНФИДЕНЦИАЛЬНОСТЬЮ

Хотя анализ, описанный в Разделе 6, показывает, что GAN не генерирует данные для обучения в первую очередь, мы понимаем, что медицинские данные часто очень конфиденциальны, а нарушения конфиденциальности обходятся дорого. Чтобы перейти к более сильным гарантиям конфиденциальности синтетических медицинских данных, мы исследовали использование дифференциальной частной процедуры обучения для GAN. Дифференциальная конфиденциальность связана с влиянием наличия или отсутствия отдельных записей в базе данных. Интуитивно дифференциальная конфиденциальность накладывает ограничения на вероятность получения одного и того же результата (в нашем случае экземпляра обученной GAN) при небольшом возмущении базового набора данных. Если процедура обучения гарантирует (ϵ, δ) дифференциальную конфиденциальность, то для двух «соседних» наборов данных (отличающихся одной записью) D, D_0

$$P[M(D) \leq S] \leq P[M(D_0) \leq S] + \delta(1) \text{ где } M(D) \text{ — GCC, полученная в результате}$$

обучения на D , S — любое подмножество возможных выходов процедуры обучения (любое подмножество возможных GAN), а вероятность P учитывает случайность в процедуре $M(D)$. Таким образом, дифференциальная конфиденциальность требует, чтобы распределение по GAN, создаваемое M , менялось «медленно» по мере изменения D , где ϵ и δ ограничивают эту «медленность». Вдохновленные недавним препринтом (Beaulieu-Jones et al., 2017), мы применяем алгоритм дифференциального частного стохастического градиентного спуска (DP-SGD) (Abadi et al., 2016) к дискриминатору (поскольку генератор не «видит» личные данные напрямую). Для получения дополнительной информации об алгоритме (и приведенном выше определении дифференциальной конфиденциальности) мы ссылаемся на (Abadi et al., 2016) и (Dwork et al., 2006).

На практике DP-SGD работает путем отсечения градиентов для каждого примера и пакетного добавления шума. Это означает, что сигнал, полученный из любого отдельного примера, ограничен, что обеспечивает дифференциальную конфиденциальность. Некоторый бюджет конфиденциальности «тратится» каждый раз, когда процедура обучения вычисляет градиенты для дискриминатора, что позволяет нам оценивать эффективные значения ϵ и δ во время обучения. Мы используем метод бухгалтера моментов из (Abadi et al., 2016), чтобы отслеживать эти расходы на конфиденциальность. Поиск гиперпараметров, которые обеспечивают как приемлемую конфиденциальность, так и реалистичные образцы GAN, оказался сложной задачей. Мы сосредоточились на задачах MNIST и eICU с RCGAN, используя оценку TSTR.

Для MNIST мы обрезали градиенты до 0,05 и добавили гауссовский шум со средним нулем и стандартным отклонением $0,05 \times 2$. Для $\epsilon = 1$ и $\delta = 1,8 \times 10^{-3}$ мы достигли точности $0,75 \pm 0,025$ при $\epsilon = 0,75$ и $\delta = 9,8 \times 10^{-3}$. Эти результаты намного ниже производительности, о которой сообщает оригинальный GAN (таблица 1), но они обеспечивают конфиденциальность. Для сравнения, в (Abadi et al., 2016) они сообщают о точности 0,95 для обучения классификатора MNIST (по полной задаче) на реальном наборе данных дифференциально частным образом. (Обратите внимание, однако, что наша модель GAN должна была решить более сложную задачу моделирования цифр как временного ряда.)

Для eICU результаты показаны в таблице 3. В этом случае мы обрезали градиенты до 0,1 и добавили шум со стандартным отклонением $0,1 \times 2$. В удивительном контрасте с нашими выводами по MNIST мы наблюдаем, что производительность в задачах eICU остается высокой при дифференцированном частном обучении, даже для более строгой настройки конфиденциальности ($\epsilon = 0,5$ и $\delta = 9,8 \times 10^{-3}$). Визуальная оценка выборок, сгенерированных дифференциально-частной GAN, показывает, что, хотя она склонна к созданию менее реалистичных последовательностей, ошибки, которые она вносит, кажутся неважными для рассматриваемых нами задач. В частности, DP-GAN создает последовательности с более экстремальными значениями, но, поскольку задача заключается в прогнозировании экстремальных значений, может случиться так, что наиболее значимая часть последовательности будет сохранена. Возможность ввести шум, сохраняющий конфиденциальность, который, тем не менее, позволяет обучать нисходящие модели, предлагает интересные направления исследований на пересечении конфиденциальности и GAN.

8 ЗАКЛЮЧЕНИЕ

Мы описали, обучили и оценили рекуррентную архитектуру GAN для генерации последовательных данных с действительным знаком, которую мы называем RGAN. Мы дополнительно разработали условный вариант (RCGAN) для создания синтетических наборов данных, состоящих из данных временных рядов с действительными значениями и соответствующими метками. Поскольку эта задача ставит новые задачи, мы представили новые решения для оценки и вопросов конфиденциальности. Путем создания помеченных обучающих данных — обуславливая метки и создавая соответствующие образцы, мы можем оценить качество модели с помощью «техники TSTR», где мы обучаем модель на синтетических данных и оцениваем ее на реальных, хранимых данных. тестовый набор. Мы продемонстрировали этот подход с использованием «сериализованного» многомерного MNIST и на наборе данных реальных отделений интенсивной терапии.

		SpO2 < 95 ЧСС < 70 AUROC	ЧСС > 100
TSTR (DP)	0,859 ± 0,004 0,86 ± 0,01 0,90 ± 0,01		
АУПРК	TSTR (ДП) 0,582 ± 0,008 0,77 ± 0,03 0,75 ± 0,03 случайный 0,16 0,27		0,16

		OP < 13	OP > 20	КАРТА < 70 КАД > 110
AUROC	TSTR (DP) 0,86 ± 0,01 0,87 ± 0,01			0,78 ± 0,01 0,83 ± 0,06
АУПРК	TSTR (ДП) 0,72 ± 0,02 0,48 ± 0,03 0,705 ± 0,005 0,26 ± 0,06 случайный 0,26 0,09			0,39 0,05

Таблица 3. Производительность классификатора случайного леса, обученного на синтетических данных, сгенерированных дифференциальным методом. частная ГАН, проверенная на реальных данных. Сравните с таблицей 2. Эпоха, из которой генерируются данные был выбран с использованием проверочного набора с учетом производительности в подмножестве задач (SpO2 < 95, ЧСС > 100 и RR < 13, выделены курсивом). В каждом повторе GAN обучался с (δ) дифференциальным конфиденциальностью для = 0,5 и δ 9,8 × 10−3 .

пациентов, где модели, обученные на синтетическом наборе данных, достигли производительности за время, сравнимое с что из реальных данных. В таких областях, как медицина, где соображения конфиденциальности препятствуют обмену данными, это означает, что с усовершенствованием этих методов модели могут быть разработаны на синтетических данных. которые по-прежнему ценны для реальных задач. Это может позволить разработать синтетический «бенчмаркинг». наборы данных для медицины (или других чувствительных областей), которые позволили добиться значительного прогресса в другие области. Мы также показали, что такой синтетический набор данных не представляет серьезной угрозы для конфиденциальности. касаются или представляют собой утечку данных для исходных конфиденциальных данных обучения, а также для более строгой конфиденциальности гарантирует, что дифференциальная конфиденциальность может использоваться при обучении RCGAN с некоторой потерей производительности.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

Мартин Абади, Ашиш Агарвал, Пол Бархэм, Юджин Бревдо, Жифэн Чен, Крэйг Ситро, Грег С. Коррадо, Энди Дэвис, Джеффри Дин, Матьё Девин, Санджай Гемават, Ян Гудфеллоу, Эндрю Харп, Джеффри Ирвинг, Майкл Айсард, Янцин Цзя, Рафал Йозефович, Лукаш Кайзер, Манджунат Кудлур, Джош Левенберг, Дэн Мане, Раджат Монга, Шерри Мур, Дерек Мюррей, Крис Ола, Майк Шустер, Джонатон Шленс, Бенуа Штайнер, Илья Суцкевер, Кунал Талвар, Пол Такер, Винсент Ванхоук, Виджай Васудеван, Фернанда Вьегас, Ориол Виньялс, Пит Уорден, Мартин Ваттенберг, Мартин Вике, Юань Юй и Сюэцян Чжэн. TensorFlow: крупномасштабное машинное обучение на гетерогенные системы, 2015. URL <http://tensorflow.org/>. Программное обеспечение доступно от tensorflow.org.

Мартин Абади, Энди Чу, Ян Гудфеллоу, Х. Брендан МакМахан, Илья Миронов, Кунал Талвар и Ли Чжан. Глубокое обучение с дифференциальной конфиденциальностью. В материалах ACM SIGSAC 2016 г. Конференция по компьютерной и коммуникационной безопасности, стр. 308–318. АСМ, 2016.

Григорий Антипов, Моэз Баккуш и Жан-Люк Дюгеле. Старение лица с условным генеративом противоборствующие сети. Препринт arXiv arXiv: 1702.01983, 2017.

Мартин Арджовски, Амар Шах и Йошуа Бенжио. Унитарная эволюция рекуррентных нейронных сетей. В Международная конференция по машинному обучению, стр. 1120–1128, 2016 г.

Мартин Аржовски, Сумит Чинтала и Леон Ботту. Вассерштейн ГАН. 26 января 2017 г.

Бретт К. Больё-Джонс, Живэй Стивен Ву, Крис Уильямс и Кейси С. Грин. Сохранение конфиденциальности генеративные глубокие нейронные сети поддерживают обмен клиническими данными. bioRxiv, 2017. doi: 10.1101/159756. URL <https://www.biorxiv.org/content/early/2017/07/05/159756>.

Вача Бунлифон, Юджин Белиловский, Мэтью Б. Блашко, Иоаннис Антоноглу и Артур Греттон. Тест относительного сходства для выбора модели в генеративных моделях. 14 ноября 2015 г.

Си Чен, Ян Дуань, Рейн Хутуфт, Джон Шульман, Илья Суцкевер и Питер Аббель. InfoGAN: обучение интерпретируемому представлению с помощью информации, максимизирующей генеративные состязательные сети. 12 июня 2016.

- Эдвард Чой, Сиддхарт Бисвал, Брэдли Малин, Джон Дьюк, Уолтер Ф. Стюарт и Джименг Сан. Генерация дискретных электронных медицинских карт с несколькими метками с использованием генеративно-состязательных сетей. 19 марта 2017 г.
- Синтия Дворк, Кришнарам Кентапади, Фрэнк МакШерри, Илья Миронов и Мони Наор. Наши данные, мы сами: Конфиденциальность за счет распределенной генерации шума. В Eurocrypt, том 4004, стр. 486–503. Спрингер, 2006.
- Отто Фабиус и Йост Р ван Амерсфорт. Вариационные рекуррентные автокодировщики. Препринт arXiv arXiv: 1412.6581, 2014 г.
- Джон Готье. Условно-генеративные состязательные сети для сверточной генерации граней. Классный проект для Стэнфорда CS231N: сверточные нейронные сети для визуального распознавания, зимний семестр, 2014 г. (5): 2, 2014 г.
- Ян Дж. Гудфеллоу, Жан Пуже-Абади, Мехди Мирза, Бинг Сюй, Дэвид Уорд-Фарли, Шерджил Озейр, Аарон Курвиль и Йошуа Бенжиро. Генеративные состязательные сети. 10 июня 2014 г.
- Артур Греттон, Карстен М. Боргвардт, Мальте Раш, Бернхард Шелькопф и Алекс Дж. Смола. Метод ядра для задачи с двумя выборками. В Достижениях в области систем обработки нейронной информации, стр. 513–520, 2007 г.
- Ишаан Гулрадхани, Фарук Ахмед, Мартин Арджовски, Винсент Дюмулен и Аарон Курвиль. Улучшено обучение вассерштейнских GAN. 31 марта 2017 г.
- Зепп Хохрайтер и Юрген Шмидхубер. Длинная кратковременная память. Нейронные вычисления, 9(8): 1735–1780, 1997.
- Стефани Л. Хайланд и Гуннар Рэтч. Изучение унитарных операторов с помощью $u(n)$. В AAAI 2017, 2017.
- Дидерик П. Кингма и Макс Веллинг. Автокодирование вариационного байеса. Препринт arXiv arXiv: 1312.6114, 2013 г.
- Куок В. Ле, Навдип Джейтли и Джеффри Э. Хинтон. Простой способ инициализации рекуррентных сетей выпрямленных линейных единиц. Препринт arXiv arXiv: 1504.00941, 2015 г.
- Кристиан Ледиг, Лукас Тейс, Ференц Хусар, Хосе Кабальеро, Эндрю Каннингем, Алехандро Акоста, Эндрю Эйткен, Алихан Теджани, Йоханнес Тотц, Зехан Ван и др. Фотореалистичное одиночное изображение сверхвысокого разрешения с использованием генеративной состязательной сети. Препринт arXiv arXiv: 1609.04802, 2016 г.
- Дживэй Ли, Уилл Монро, Тяньлинь Ши, Алан Риттер и Дэн Джурафски. Состязательное обучение для генерации нейронных диалогов. Препринт arXiv arXiv: 1701.06547, 2017.
- Юцзя Ли, Кевин Сверски и Ричард Земель. Сети согласования генеративных моментов. 10 февраля 2015.
- Мехди Мирза и Симон Осиндеро. Условно-генеративные состязательные сети. Препринт arXiv arXiv: 1411.1784, 2014 г.
- Олоф Могрен. C-RNN-GAN: непрерывные рекуррентные нейронные сети с состязательным обучением. 29 ноября 2016 г.
- Аарон ван ден Оорд, Нал Калхбреннер и Корай Кавуккуоглу. Пиксельные рекуррентные нейронные сети. Препринт arXiv arXiv: 1601.06759, 2016 г.
- Николас Паперно, Мартин Абади, Ульфэр Эрлингссон, Ян Гудфеллоу и Кунал Талвар. Полуконтролируемая передача знаний для глубокого обучения на основе данных частного обучения. Препринт arXiv arXiv: 1610.05755, 2016 г.
- Ф. Педрегоса, Г. Вароко, А. Грамфор, В. Мишель, Б. Тирион, О. Гризель, М. Блондель, П. Преттенхофер, Р. Вайс, В. Дюбур, Ж. Вандерплас, А. Пассос, Д., Курнапо, М. Брюше, М. Перро и Э. Дюшене. Scikit-learn: Машинное обучение в Python. Журнал исследований машинного обучения, 12:2825–2830, 2011 г.

Алек Рэдфорд, Люк Мец и Сумит Чинтала. Неконтролируемое репрезентативное обучение с помощью глубоких сверточных генеративно-состязательных сетей. Препринт arXiv arXiv: 1511.06434, 2015 г.

Скотт Рид, Зейнеп Аката, Синьчен Ян, Ладжануген Логесваран, Бернт Шиле и Хонглак Ли. Генеративный состязательный текст для синтеза изображений. В материалах 33-й Международной конференции по машинному обучению, том 3, 2016 г.

Тим Салиманс, Ян Гудфеллоу, Войцех Заремба, Вики Чунг, Алек Рэдфорд и Си Чен. Улучшенные методы обучения GAN. 10 июня 2016 г.

Дугал Дж. Сазерленд, Сяо-Ю Тунг, Хайко Стратманн, Сумяджит Де, Адитья Рамдас, Алекс Смола и Артур Греттон. Генеративные модели и критика моделей через оптимизированное максимальное среднее несоответствие. 14 ноября 2016 г.

Лукас Тейс, Аарон ван ден Оорд и Матиас Бетге. Примечание об оценке генеративных модели. 5 ноября 2015 г.

Юхуай Ву, Юрий Бурда, Руслан Салахутдинов и Роджер Гроссе. При количественном анализе Генеративные модели на основе декодера. 14 ноября 2016 г.

Чжэнь Ян, Вэй Чен, Фэн Ван и Бо Сюй. Улучшение нейронного машинного перевода с условным последовательность генерирующих состязательных сетей. Препринт arXiv arXiv: 1703.04887, 2017.

Лантао Юй, Вейнань Чжан, Цзюнь Ван и Юн Юй. SeqGAN: последовательность генерирующих состязательных сетей с градиентом политики. 18 сентября 2016 г.