

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет «Информатика и системы управления»  
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Домашнее задание**  
по дисциплине «Методы машинного обучения»

Тема: «Анализ современных методов машинного обучения и их применение  
для решения практических задач»

ИСПОЛНИТЕЛЬ:

группа ИУ5-24М

\_\_\_\_\_  
Молева А.А.

ФИО

\_\_\_\_\_  
подпись

"\_\_" \_\_\_\_\_ 2022 г.

ПРЕПОДАВАТЕЛЬ:

\_\_\_\_\_  
Гапанюк Ю.И.

ФИО

\_\_\_\_\_  
подпись

"\_\_" \_\_\_\_\_ 2022 г.

Москва - 2022

---

## Задание

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач.

Домашнее задание включает три основных этапа:

- выбор задачи;
- теоретический этап;
- практический этап.

Этап выбора задачи предполагает анализ ресурса `paperswithcode`. Данный ресурс включает описание нескольких тысяч современных задач в области машинного обучения. Каждое описание задачи содержит ссылки на наиболее современные и актуальные научные статьи, предназначенные для решения задачи (список статей регулярно обновляется авторами ресурса). Каждое описание статьи содержит ссылку на репозиторий с открытым исходным кодом, реализующим представленные в статье эксперименты. На этапе выбора задачи обучающийся выбирает одну из задач машинного обучения, описание которой содержит ссылки на статьи и репозитории с исходным кодом.

Теоретический этап включает проработку как минимум двух статей, относящихся к выбранной задаче. Результаты проработки обучающийся излагает в теоретической части отчета по домашнему заданию, которая может включать:

- описание общих подходов к решению задачи;
- конкретные топологии нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения, предназначенных для решения задачи;
- математическое описание, алгоритмы функционирования, особенности обучения используемых для решения задачи нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения;
- описание наборов данных, используемых для обучения моделей;
- оценка качества решения задачи, описание метрик качества и их

значений;

- предложения обучающегося по улучшению качества решения задачи.

Практический этап включает повторение экспериментов авторов статей на основе представленных авторами репозитория с исходным кодом и возможное улучшение обучающимися полученных результатов. Результаты проработки обучающийся излагает в практической части отчета по домашнему заданию, которая может включать:

- исходные коды программ, представленные авторами статей, результаты документирования программ обучающимися с использованием диаграмм UML, путем визуализации топологий нейронных сетей и другими способами;
- результаты выполнения программ, вычисление значений для описанных в статьях метрик качества, выводы обучающегося о воспроизводимости экспериментов авторов статей и соответствии практических экспериментов теоретическим материалам статей;
- предложения обучающегося по возможным улучшениям решения задачи, результаты практических экспериментов (исходные коды, документация) по возможному улучшению решения задачи.

**Выбранная задача: «Генерация изображений».**

**Выбранные статьи для анализа:**

- Wasserstein GAN;
- Улучшенное обучение Wasserstein GANs.

**Теоретическая часть**

**Статья 1:**

Вводится новый алгоритм под названием WGAN, альтернативный традиционному обучению GAN. В этой новой модели отображается, что можно улучшить стабильность обучения, избавиться от таких проблем, как коллапс режима, и обеспечить значимые кривые обучения, полезные для

отладки и поиска гиперпараметров. Кроме того, показывается, что соответствующая задача оптимизации является обоснованной, и предоставляем обширную теоретическую работу, подчеркивающую глубокие связи с другими расстояниями между распределениями.

Проблема, которой посвящена эта статья, заключается в неконтролируемом обучении. Главным образом, что значит изучить распределение вероятностей? Классический ответ на этот вопрос заключается в изучении плотности вероятности. Это часто делается путем определения параметрического семейства плотностей и нахождения той, которая максимизировала вероятность для данных.

Если реальное распределение данных допускает плотность, а  $P_\theta$  является распределением параметризованной плотности  $P_\theta$ , то асимптотически это сводится к минимизации расхождения Кулбека-Лейблера  $KL(p_{\text{r}} \| p_\theta)$ .

Чтобы это имело смысл, нужно, чтобы плотность модели  $P_\theta$  существовала. Это не относится к довольно распространенной ситуации, когда мы имеем дело с распределениями, поддерживаемыми низкоразмерными многообразиями. Тогда маловероятно, что модельное многообразие и поддержка истинного распределения имеют незначительное пересечение, и это означает, что расстояние  $KL$  не определено (или просто бесконечно).

Типичным решением проблемы является добавление шумового члена к распределению модели. Вот почему практически все генеративные модели, описанные в классической литературе по машинному обучению, включают компонент шума. В простейшем случае предполагается наличие гауссова шума с относительно высокой полосой пропускания, чтобы охватить все примеры. Хорошо известно, например, что в случае моделей генерации изображений этот шум ухудшает качество выборок и делает их размытыми. . Другими словами, добавленный шумовой термин явно неверен для проблемы, но необходим для того, чтобы подход с максимальным правдоподобием работал.

Вместо того, чтобы оценивать плотность  $P_r$ , которая может не существовать, мы можем определить случайную величину  $Z$  с фиксированным распределением  $p(z)$  и передать ее через параметрическую функцию (обычно это какая-то нейронная сеть), которая непосредственно генерирует выборки после определенного распределения  $p_\theta$ . Изменяя  $\theta$ , можно изменить это распределение и сделать его близким к реальному распределению данных  $P_r$ . Это полезно в двух отношениях. Прежде всего, в отличие от плотностей, этот подход может представлять распределения, ограниченные низкоразмерным многообразием. Во-вторых, способность легко генерировать выборки часто более полезна, чем знание числового значения плотности.

Вариационные автокодеры (VAEs) и генеративные состязательные сети (GAN) являются хорошо известными примерами этого подхода. Поскольку мы фокусируемся на приблизительной вероятности примеров, они разделяют ограничения стандартных моделей и нуждаются в дополнительных шумовых терминах. GAN предлагают гораздо большую гибкость в определении целевой функции, включая Дженсена-Шеннона и все  $f$ -дивергенции, а также некоторые экзотические комбинации. С другой стороны, обучающие GAN хорошо известны своей хрупкостью и нестабильностью.

В этой статье мы обращаем наше внимание на различные способы измерения того, насколько близко модельное распределение к реальному распределению, или, что эквивалентно, на различные способы определения расстояния или расхождения.

Вкладом в этот документ являются:

В разделе 2 приводится всесторонний теоретический анализ того, как ведет себя расстояние Earth Mover (EM) по сравнению с популярными вероятностными расстояниями и расхождениями, используемыми в контексте обучающих распределений.

В разделе 3 определяется форма GAN, называемая Wasserstein-GAN, которая минимизирует разумную и эффективную аппроксимацию расстояния EM, и теоретически показывается, что соответствующая задача оптимизации

является обоснованной.

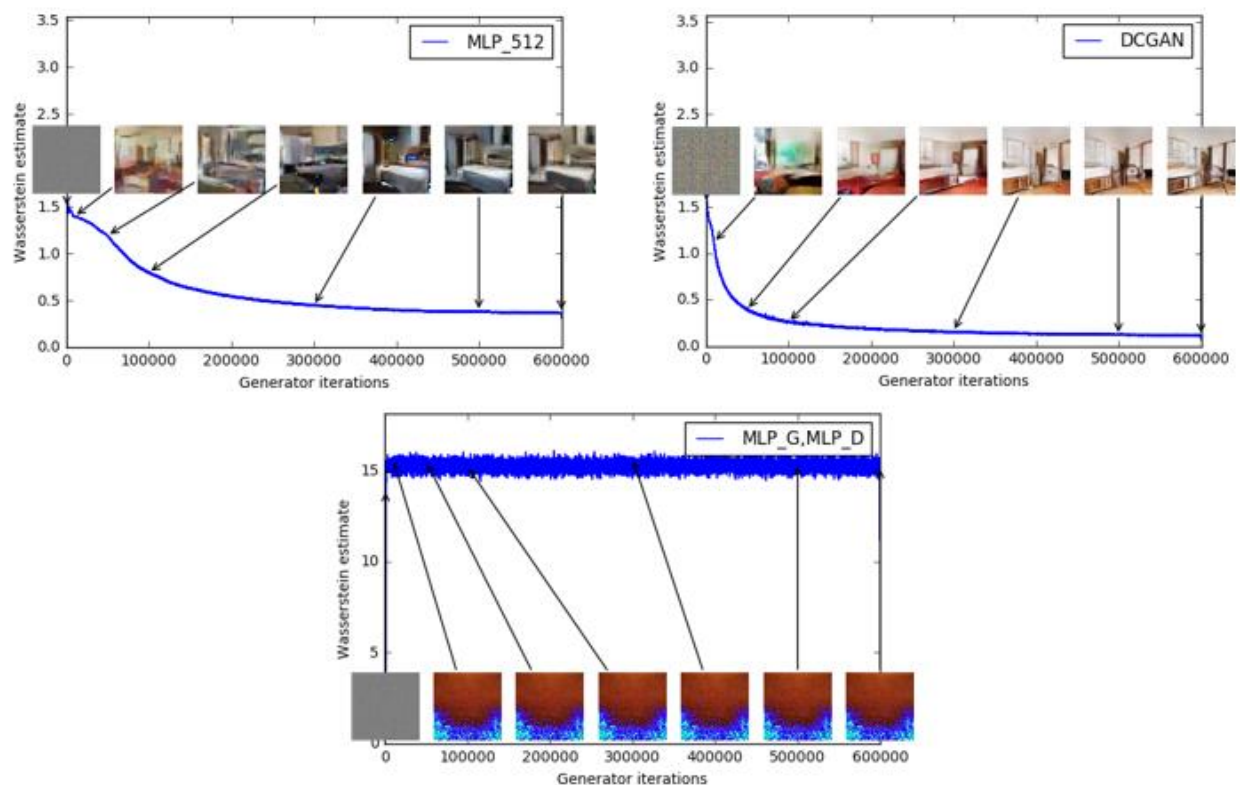
В разделе 4 эмпирически показывается, что WGANs решают основные проблемы обучения GANs. В частности, обучение WGANs не требует поддержания тщательного баланса в обучении дискриминатора и генератора, а также не требует тщательного проектирования сетевой архитектуры. Явление падения режима, типичное для GANs, также резко снижается. Одним из наиболее убедительных практических преимуществ WGANs является возможность непрерывной оценки расстояния до электромагнитного излучения путем обучения дискриминатора к оперативности. Построение этих кривых обучения не только полезно для отладки и гиперпараметрического поиска, но и замечательно коррелирует с наблюдаемым качеством выборки.

Проводятся эксперименты по генерации изображений с использованием алгоритма Вассерштейна-GAN и показывается, что его использование имеет значительные практические преимущества по сравнению с формулировкой, используемой в стандартных GAN.

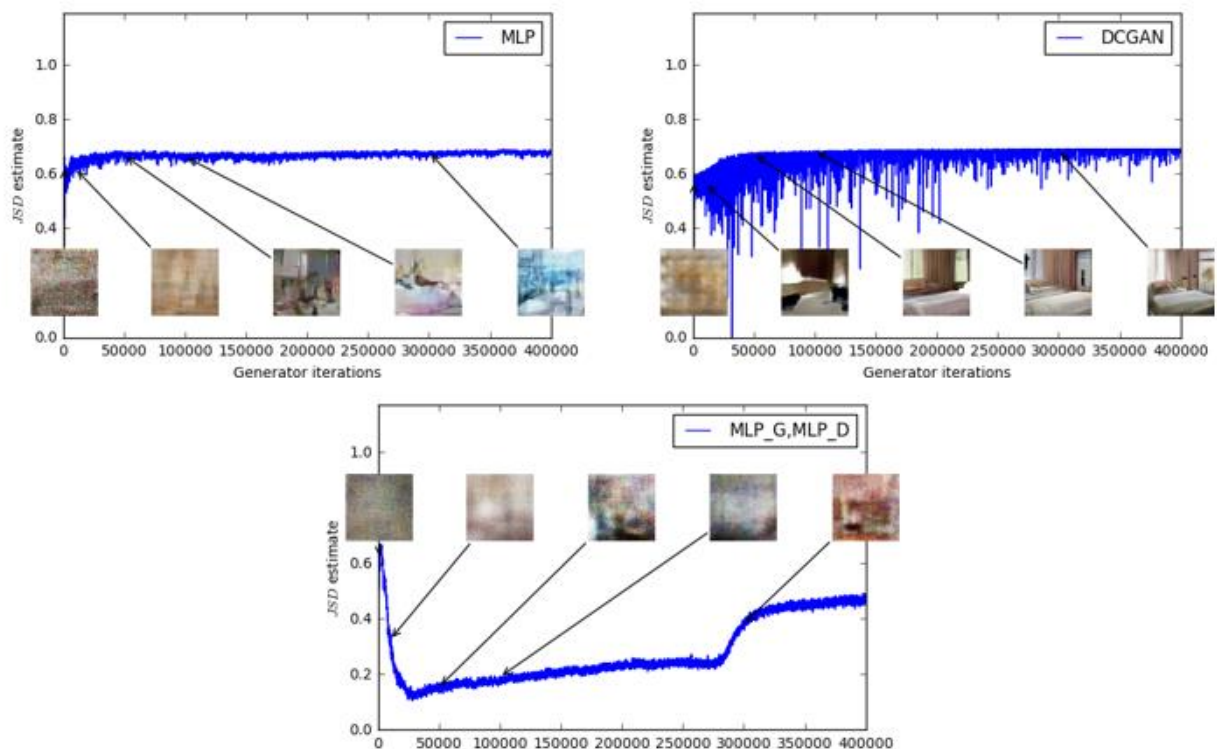
Заявляется о двух основных преимуществах:

- значимый показатель потерь, который коррелирует со сходимостью генератора и качеством выборки;
- улучшена стабильность процесса оптимизации.

Проводятся эксперименты по созданию изображений. Целевым распределением для изучения является набор данных LSUN-Bedrooms dataset – коллекция естественных изображений комнат внутри помещений. Базовым сравнением является DCGAN, GAN со сверточной архитектурой, обученный стандартной процедуре GAN с использованием трюка  $\log D$ . Сгенерированные образцы представляют собой 3-канальные изображения размером 64x64 пикселя.



Обучающие кривые и выборки на разных этапах обучения. Видно четкую корреляцию между меньшей погрешностью и лучшим качеством выборки. Вверху слева: генератор представляет собой MLP с 4 скрытыми слоями и 512 блоками на каждом уровне. Потери последовательно уменьшаются по мере прохождения обучения и повышения качества выборки. Вверху справа: генератор представляет собой стандартный генератор постоянного тока. Потери быстро уменьшаются, а качество образца также повышается. На обоих верхних графиках критиком является DCGAN без сигмовидной мышцы, поэтому потери могут быть подвергнуты сравнению. Нижняя половина: и генератор, и дискриминатор являются MLP с существенно высокой скоростью обучения (поэтому обучение не удалось). Потери постоянны, и образцы также постоянны. Обучающие кривые были пропущены через медианный фильтр для целей визуализации.



Оценки JS для генератора MLP (вверху слева) и генератора DCGAN (вверху справа), обученные с помощью стандартной процедуры GAN. У обоих был дискриминатор DCGAN. Обе кривые имеют возрастающую погрешность. Выборки улучшаются для DCGAN, но оценка JS увеличивается или остается постоянной, указывая на отсутствие существенной корреляции между качеством выборки и потерями. Внизу: MLP как с генератором, так и с дискриминатором. Кривая идет вверх и вниз независимо от качества образца. Все обучающие кривые были пропущены через тот же медианный фильтр, что и на рис. 3.

Одним из преимуществ WGAN является то, что он позволяет обучать критика до достижения оптимальности. Когда критик обучается до завершения, это просто приводит к потере генератора, который можно обучать как любую другую нейронную сеть. Это говорит о том, что больше не нужно правильно балансировать мощность генератора и дискриминатора. Чем лучше критик, тем более качественные градиенты мы используем для обучения генератора.



Наблюдаем, что WGANs гораздо более надежны, чем GAN, когда меняется архитектурный выбор генератора.

Представлен алгоритм WGAN, который сочли альтернативой традиционному обучению GAN. В этой новой модели показано, что можно повысить стабильность обучения, избавиться от таких проблем, как коллапс режима, и обеспечить значимые кривые обучения, полезные для отладки и гиперпараметрического поиска. Кроме того, показано, что соответствующая задача оптимизации является обоснованной, и предоставлена обширная теоретическая работа, подчеркивающая глубокие связи с другими расстояниями между распределениями.

## **Статья 2:**

Генеративные состязательные сети (GAN) являются мощными генеративными моделями, но страдают от нестабильности обучения. Недавно предложенный Wasserstein GAN (WGAN) продвигается к стабильному обучению GAN, но иногда все еще может генерировать только низкокачественные выборки или не сходиться. Обнаружено, что эти проблемы часто возникают из-за использования отсечения веса в WGAN для навязывания критике ограничения Липшица, что может привести к нежелательному поведению. Предлагается альтернатива отсечению весов: штрафовать норму градиента критика относительно его ввода. Предложенный метод работает лучше стандартного WGAN и позволяет стабильно обучать широкий спектр архитектур GAN практически без настройки гиперпараметров, включая 101-слойные реснеты и языковые модели над дискретными данными.

Генеративные состязательные сети (GAN) представляют собой мощный класс генеративных моделей, которые представляют генеративное моделирование как игру между двумя сетями: генераторная сеть производит синтетические данные с учетом некоторого источника шума, а дискриминаторная сеть различает выходные данные генератора и истинные

данные. GAN могут создавать очень визуально привлекательные образцы, но часто их трудно обучить.

В частности, приводится анализ свойств сходимости функции значений, оптимизируемой GANs. Предложенная ими альтернатива, названная Wasserstein GAN (WGAN), использует расстояние Вассерштейна для получения функции значений, которая обладает лучшими теоретическими свойствами, чем исходная. WGAN требует, чтобы дискриминатор должен находиться в пространстве функций 1-Липшица, которую применяют с помощью отсечения веса.

В работе:

1. На игрушечных наборах данных демонстрируется, как отсечение критического веса может привести к нежелательному поведению.
2. Предлагается градиентный штраф (WGAN-GP), который не страдает от тех же проблем.
3. Демонстрируется стабильное обучение различным архитектурам GAN, улучшение производительности при отсечении избыточного веса, высококачественную генерацию изображений и модель языка GAN на уровне символов без какой-либо дискретной выборки.

Стратегия обучения GAN заключается в определении игры между двумя конкурирующими сетями. Генераторная сеть сопоставляет источник шума с входным пространством. Сеть дискриминатора получает либо сгенерированную выборку, либо истинную выборку данных и должна различать их. Генератор обучен обманывать дискриминатор.

Функция значений WGAN приводит к критической функции, градиент которой по отношению к ее входным данным ведет себя лучше, чем ее аналог GAN, что упрощает оптимизацию генератора. Кроме того, WGAN обладает желательным свойством, заключающимся в том, что его функция значений коррелирует с качеством выборки, чего нельзя сказать о GANs.

Обнаружено, что отсечение веса в WGAN приводит к трудностям оптимизации, и что даже при успешной оптимизации результирующая





























критика может иметь патологическую поверхность значений.

Предлагается альтернативный способ применения ограничения Липшица. Дифференцируемая функция является 1-Липштицем тогда и только тогда, когда она имеет градиенты с нормой не более 1 везде, поэтому рассматривается возможность прямого ограничения нормы градиента выходных данных критика относительно его входных данных.

Коэффициент штрафа во всех экспериментах в этой статье используется  $\lambda = 10$ , который, как обнаружено, хорошо работает в различных архитектурах и наборах данных, начиная от игрушечных задач и заканчивая большими CNN ImageNet.

Из набора выбирается 200 архитектур и обучим каждую на 32x32 ImageNet как с WGAN-GP, так и со стандартными целями GAN. В таблице указано количество случаев, когда: только стандартный GAN преуспел, только WGAN-GP преуспел, оба преуспели, либо оба потерпели неудачу, где успех определяется как начальный балл  $>$  минимальный балл. Для большинства вариантов порога оценки WGAN-GP успешно обучает многие архитектуры из этого набора, которые мы не смогли обучить со стандартной целью GAN.

| Min. score | Only GAN | Only WGAN-GP | Both succeeded | Both failed |
|------------|----------|--------------|----------------|-------------|
| 1.0        | 0        | 8            | 192            | 0           |
| 3.0        | 1        | 88           | 110            | 1           |
| 5.0        | 0        | 147          | 42             | 11          |
| 7.0        | 1        | 104          | 5              | 90          |
| 9.0        | 0        | 0            | 0              | 200         |

| DCGAN   | LSGAN   | WGAN (clipping)  | WGAN-GP (ours)  |
|---|---|--|---|
| Baseline (G: DCGAN, D: DCGAN)   |   |  |   |
|  |  |  |  |
| G: No BN and a constant number of filters, D: DCGAN                               |   |  |   |
|  |  |  |  |
| G: 4-layer 512-dim ReLU MLP, D: DCGAN   |   |  |   |
|  |  |  |  |
| No normalization in either G or D   |   |  |   |
|  |  |  |  |
| Gated multiplicative nonlinearities everywhere in G and D                         |   |  |   |
|  |  |  |  |
| tanh nonlinearities everywhere in G and D   |   |  |   |
|  |  |  |  |
| 101-layer ResNet G and D  |   |  |   |
|  |  |  |  |

Чтобы продемонстрировать способность модели обучать множество архитектур с ее настройками по умолчанию, обучаем шесть различных архитектур GAN на базе набора данных LSUN.dataset. В дополнение к базовой архитектуре DC- GAN выбираем шесть архитектур.

Хотя не подтверждено, что это невозможно без этого метода, но это первый случай, когда очень глубокие остаточные сети были успешно обучены в условиях GAN. Для каждой архитектуры обучаем модели, используя четыре различных метода GAN: WGAN-GP, WGAN с отсечением веса, DCGAN и метод наименьших квадратов GAN. Для каждой цели использовали набор гиперпараметров оптимизатора по умолчанию, рекомендованный в этой работе (за исключением LSGAN, где искали по скорости обучения).

Для WGAN-GP заменяем любую пакетную нормализацию в дискриминаторе нормализацией слоя. Обучаем каждую модель 200 Тыс. итераций. Нам удалось обучить каждую архитектуру только с помощью общего набора гиперпараметров с использованием WGAN-GP. Для любого другого метода обучения некоторые из этих архитектур были нестабильными

или страдали от коллапса режима.

Одним из преимуществ этого метода по сравнению с отсечением веса является улучшенная скорость обучения и качество выборки.

В этой работе были продемонстрированы проблемы с отсечением веса в WGAN и введена альтернатива в виде штрафного срока в потере критика, которая не демонстрирует тех же проблем. Используя этот метод, была продемонстрирована высокая производительность и стабильность моделирования в различных архитектурах. Другим интересным направлением является адаптация штрафного термина к стандартной целевой функции GAN, где он может стабилизировать обучение, поощряя дискриминатора изучать более плавные границы принятия решений.

## **Вывод**

В данной работе был проведен анализ двух статей: Wasserstein GAN; Улучшенное обучение Wasserstein GANs. В первой статье говорится о том, что новый алгоритм WGAN, который сочли альтернативой традиционному обучению GAN, повышает стабильность обучения, избавляется от таких проблем, как коллапс режима, и обеспечивает значимые кривые обучения, полезные для отладки и гиперпараметрического поиска. WGAN — использование метрики Вассерштейна внутри функции ошибки, позволяет дискриминатору быстрее обучаться и выявлять повторяющиеся выходы, на которых стабилизируется генератор. А во второй статье говорится о том, что существуют проблемы с отсечением веса в WGAN и из-за этого вводится альтернатива в виде штрафного срока в потере критика, которая не демонстрирует тех же проблем. Используя этот метод, была продемонстрирована высокая производительность и стабильность моделирования в различных архитектурах.

## **Список литературы**

1. Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville Improved Training of Wasserstein GANs // Cornell University
2. Martin Arjovsky, Soumith Chintala, L'eon Bottou Wasserstein GAN // Institute of Mathematical Sciences