МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

ПОСТРОЕНИЕ ФОТОРЕАЛИСТИЧНЫХ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ПОРОЖДАЮЩИХ СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 02.04.03 — Математическое обеспечение и администрирование
информационных систем
факультета КНиИТ
Филатова Михаила Дмитриевича

Научный руководитель
к. ф.-м. н., доцент _____ А. А. Кузнецов

Заведующий кафедрой
к. ф.-м. н., доцент _____ А. С. Иванов

Саратов 2020

ВВЕДЕНИЕ

Программное обеспечение является неотъемлемой частью современного мира. Его разработка ведется для создания, обработки и отображения информации, относящейся к различным сферам человеческой деятельности. В ходе решения той или иной задачи разрабатывается некий алгоритм, позволяющий эффективно и корректно выдавать результат, согласно полученным входным данным. Во многих случаях для решения конкретной прикладной задачи уже существует некоторый алгоритм, либо возможна его модификация. Такие задачи достаточно легко автоматизировать. Но зачастую задача требует творческого подхода для выделения закономерностей и построение соответствующей системы становится нетривиальной задачей. С целью оптимизации ресурсов и ускорения работы по разработке алгоритмов для новых задач была начата работа над так называемыми «системами искусственного интеллекта».

Искусственный интеллект позволяет компьютерам обучаться на собственном опыте, адаптироваться к задаваемым параметрам и выполнять те задачи, которые раньше были под силу только человеку. В большинстве случаев реализации ИИ (от компьютерных шахматистов до беспилотных автомобилей) крайне важна возможность глубокого обучения и обработки естественного языка. Благодаря этим технологиям компьютеры можно «научить» выполнению определенных задач с помощью обработки большого объема данных и выявления в них закономерностей.

Основой разработки систем искусственного интеллекта на данный момент являются нейронные сети. Эта математическая модель создавалась на основе биологических нейронных сетей — сетей нервных клеток живого организма. Модель возникла в процессе изучения работы мозга и протекающих в нем процессов, при попытку симулировать работу мозга для обучения компьютера. Однако данная теория нашла широкое применение в области разработки и реализации языков программирования. Использование технологий, основанных на синтаксических методах, стало общепринятым. При этом эффективное и грамотное применение этих средств требует от разработчика знаний математических основ теории формальных языков и трансляций.

На заре глубоких нейронных сетей основным направлением развития было распознавание голоса. На данный момент нейронные сети используются для работы с текстовыми, графическими, звуковыми и прочими представлени-

ями данных. Использование глубокого машинного обучения позволяет анализировать данные, выявлять в них неочевидные закономерности, генерировать данные с требуемыми характеристиками, не выделяя какие-либо зависимости явным образом.

Одним из частных применений является генерация изображений с заданными свойствами. Выделяются различные подходы к построению нейронных сетей для работы с графическими данными, но для всех архитектур общим является процесс выявления требуемых характеристик из исходного набора данных и перенесение их на целевые изображения.

Целью настоящей работы является создание нейронной сети для генерации фотореалистичных изображений. Для достижения поставленной цели сформулированы следующие задачи:

— изучить аналогичные задачи и существующие решения;
— проанализировать преимущества и недостатки существующих решений, выделить наиболее эффективные методики;
— рассмотреть процедуру подготовки обучающих датасетов и ее возможные оптимизации;
— реализовать нейронную сеть, подготовить тестовые данные и обучить сеть на заданном наборе;
— проверить корректность работы и оценить результативность реализованной модели.

Описываемая работа носит практическую значимость с точки зрения подготовки тренировочных наборов данных для обучения нейронных сетей. В ходе работы выдвинуты предложения по оптимизации процесса сбора обучающих датасетов, а также проведены эксперименты с целью определения оптимальных параметров нейронной сети для получения лучших результатов при меньших затратах.

Магистерская работа состоит из введения, трех разделов, заключения, списка использованных источников и одного приложения. Общий объем работы — 70 страниц, из них 49 страниц — основное содержание, включая 33 рисунка и 2 таблицы, цифровой носитель в качестве приложения, список использованных источников информации — 20 наименований.

1 Теоретические материалы

Первый раздел посвящен теоретическим основам нейронных сетей, используемым в дальнейшей практической работе.

Нейронные сети — одно из направлений в разработке систем искусственного интеллекта. Идея заключается в том, чтобы максимально близко смоделировать работу человеческой нервной системы — а именно, её способности к обучению и исправлению ошибок. В этом состоит главная особенность любой нейронной сети — она способна самостоятельно обучаться и действовать на основании предыдущего опыта, с каждым разом делая всё меньше ошибок [1].

В основе теории искусственных нейронных сетей лежит идея создания интеллектуального устройства по образу и подобию человеческого мозга. Дело в том, что способ обработки информации человеческим мозгом принципиально отличается от способов, используемых в обычных – неймановских компьютерах. Основное преимущество биологического мозга заключается в параллельности производимых им вычислений.

Использование нейронных сетей в области обработки изображений может позволить снизить затраты времени и ресурсов на получение графических данных с заданными характеристиками. Состязательные сети используются при анимации, позволяя построить промежуточные кадры по заданным опорным. В качестве опорных кадров могут использоваться низко детализированные изображения при наличии базы данных искомых изображений. По заданной выборке нейронная сеть способна обработать кадры и придать им необходимые признаки, такие как реалистичность или особенный графический стиль.

1.1 Порождающие состязательные сети

Впервые порождающие состязательные сети (GAN) были рассмотрены и подробно представлены в 2014 году Яном Гудфеллоу [2]. В его статье вводится принцип соперничества двух подсетей: генератора и дискриминатора, направленный на их совместное обучение. Также в статье приводится сравнение с другими классами нейронных сетей по точности, обучаемости и другим характеристикам.

Имеется множество образцов X из распределения p_{data}, заданного на R_n, а также некоторое пространство латентных факторов Z из распределения pz,

например, случайные вектора из равномерного распределения $U^t(0,1)$.

Рассмотрим две нейронные сети: первая — генератор $G : Z \to R^n$ с параметрами θ, цель которой сгенерировать похожий образец из p_{data}, и вторая — дискриминатор $D : R^n \to [0,1]$ с параметрами γ, цель которой выдавать максимальную оценку на образцах из X и минимальную на сгенерированных образцах из G. Распределение, порождаемое генератором будем обозначать p_{gen}. Так же заметим, что в текущем изложении не принципиальны архитектуры нейронных сетей, поэтому можно считать, что параметры θ и γ являются просто параметрами многослойных персептронов [3].

В качестве примера можно рассматривать генерацию реалистичных фотографий: в этом случае, входом для генератора может быть случайный многомерный шум, а выходом генератора (и входом для дискриминатора) RGB-изображение; выходом же для дискриминатора будет вероятность, что фотография настоящая, т.е число от 0 до 1.

Наша задача выучить распределение p_{gen} так, чтобы оно как можно лучше описывало p_{data}. Зададим функцию ошибки для получившейся модели. Со стороны дискриминатора мы хотим распознавать образцы из X как правильные, т.е в сторону единицы, и образцы из G как неправильные, т.е в сторону нуля, таким образом нужно максимизировать следующую величину:

$\underset{x\sim p_{data}}{E}[logD(x)] + \underset{x\sim p_{gen}}{E}[log(1-D(x))]$, где $\underset{x\sim p_{gen}}{E}[log(1-D(x))] = \underset{z\sim p_z}{E}[log(1-D(G(z))]$,

Со стороны генератора требуется научиться "обманывать"дискриминатор, т.е минимизировать по p_{gen} второе слагаемое предыдущего выражения. Другими словами, G и D играют в так называемую минимаксную игру, решая следующую задачу оптимизации:

$$\min_G \max_D \underset{x\sim p_{data}}{E}[logD(x)] + \underset{z\sim p_z}{E}[log(1 - D(G(z))].$$

1.2 Cycle-Consistent Generative Adversarial Networks

Модель Cycle-Consistent Generative Adversarial Networks (cycle-GAN) [4] представляет из себя две сети-генератора со своими дискриминаторами. Первая осуществляет отображение $F : X \to Y$, а вторая — обратное отображение $G : Y \to X$. Таким образом, получаем сеть, способную выполнять операции между двумя доменами, причем отображение стремится к биективному. Идея обучения cycle-GAN заключается в распознавании, является ли полу-

ченное изображение из каждого домена реальным или создано генератором. Подобный подход позволяет выявлять и закреплять за моделью характерные признаки обоих доменов, в связи с чем растет общая эффективность модели.

Преимуществом модели CycleGAN является то, что она может тренирована без попарно соответствующих изображений в тестовом наборе данных. Таким образом для каждого входного значения не нужно иметь соответствующий ожидаемый выходной результат (например фотография дня и ночи, зимы и лета и т.д.). Достаточно иметь 2 набора данных, объединяющих в себе изображения по некоему общему признаку, выделение которого и является поставленной задачей.

Архитектура модели разделена на две вложенные модели генераторов: первый генератор (Generator-A) отвечает за создание изображений из первого набора данных, или точнее первого домена (Domain-A), а второй генератор (Generator-B) отвечает за получение изображений из второго домена (Domain-B) [5].

$GeneratorA \rightarrow DomainA$

$GeneratorB \rightarrow DomainB$

Модель генератора скрывает под собой некоторое преобразование изображения. Это значит, что получаемый результат работы генератора зависим от поданного на вход генератору значения. Входным значением является изображение из домена противоположного целевому (Generator-A получает на вход значение из Domain-B, а Generator-B - значение из DomainA).

$DomainB \rightarrow GeneratorA \rightarrow DomainA$

$DomainA \rightarrow GeneratorB \rightarrow DomainB$

Каждому генератору соответствует свой дискриминатор. Первый дискриминатор (Discriminator-A) получает на вход реальное значение из первого домена и результат работы первого генератора. Задача дискриминатора в том, чтобы определить, является ли поданное на вход реальным или сгенерированным. Аналогичное утверждение касается второго дискриминатора (Discriminator-B) и генератора соответственно.

$DomainA \rightarrow DiscriminatorA \rightarrow [Real/Fake]$

$DomainB \rightarrow GeneratorA \rightarrow DiscriminatorA \rightarrow [Real/Fake]$

$DomainB \rightarrow DiscriminatorB \rightarrow [Real/Fake]$

$DomainA \rightarrow GeneratorB \rightarrow DiscriminatorB \rightarrow [Real/Fake]$

Помимо обучения сети двум противоположным отображениям между доменами ставится задачи минимизации потерь от последовательного применения обоих преобразований к исходному изображению. Так на каждое тестовое значение из первого домена применяется преобразование Generator-A, после чего к полученный результат подается на вход второму генератору. В силу ожидаемой взаимной однозначности преобразований, ожидается получение результата равного исходному ($f : A \to B, g : B \to A, g(f(x)) = x$). Такое консистентное преобразование называется циклом (cycle), откуда и берется название алгоритма. На практике получается изображение, соответствующее исходному лишь с некоторой точностью. На основании степени различия между исходным значение и результатом вычисляется потеря консистентности цикла (cycle-consistency loss). Вычисление данного значения проводится на каждой итерации обучения порождающих состязательных сетей с целью итогового снижения значения потерь.

1.3 Differentiable Interpolation-based Renderer

В 2019 году NVIDIA [6] впервые представила систему, позволяющую преобразовать любое 2D-изображение в 3D-модель. В выпущенной статье описывается архитектура нейросети, способной по одной фотографии построить 3D модель объекта и полную его развертку. Отличительными особенностями по сравнению с предыдущими решениями является учет особенностей позиции света и текстуры объекта. В ходе работы над исходной фотографией предполагается выделять из изображения объекта его текстуру, параметры освещения и отделять их от искомой 3D модели.

NVIDIA использовала технологию рендеринга под названием DIB-R (Differentiable Interpolation-based Renderer). Принцип работы сравнительно простой и понятный, поскольку человек видит двумерное изображение, а уже мозг преобразует его в 3D.

Представленная в статье структура рендеринга аналитически градиенты всех пикселей изображения. Рассматривая свойства растеризиации переднего плана и фона, предлагается выведение общих свойств геометрии объекта и отделения их от прочих свойств. Данный подход позволяет точно оптимизировать положения вершин, цвета, нормали, направления света и координаты текстуры с помощью различных моделей освещения. По выделенным свойствам строится новый рендер, который в свою очередь сравнивается с исходным с

целью уменьшения потерь.

В ходе работы DIB-R проводится прогнозирование трехмерных объектов, построение развертки изучаемого объекта и изображений объекта с ракурсов, отличных от исходного.

За основу берется полигональная сфера, которую нейронная сеть подгоняет под форму реального объекта с фотографии. Сложность обучения подобной сети заключается в необходимости подбора достаточного числа тестовых данных, включающих в себя различные трехмерные объекты. Тренировочный датасет пополняется новыми объектами, что в ближайшее время позволит строить трехмерные модели произвольных объектов.

2 Практическая реализация

Второй раздел посвящен практической реализации приложения. В качестве основы были рассмотрены CycleGAN и Differentiable Interpolation-based Renderer (DIB-R) для работы с трехмерными объектами.

На практике рассмотрены следующие практические задачи: обучение нейросети на частично размеченных данных для обработки двухмерных изображений; обучение нейросети на полностью неразмеченных данных для обработки изображений трехмерных объектов.

В качестве первого набора данных используется датасет фотографий фасадов зданий. Для данного тестового набора достаточно просто подготовить размеченный набор данных для первичного обучения. Набор данных не учитывает объемные характеристики изображенных объектов. Рассматривается исключительно фасад здания, что значительно упрощает построение парных шаблонов для второго домена. Все элементы фасадов зданий могут быть представлены в виде двухмерных графических примитивов.

В качестве второго набора данных используется набор фотографий домов в проекции. Исходным доменом являются рендеры трехмерных моделей загородных домов. Построение моделей проведено в среде 3dsMAX.

2.1 Использованное окружение

Обучение порождающий состязательных сетей проводилось при использовании следующего аппаратного обеспечения:

- Процессор - Intel Core i5-8600 - 3.10GHz;
- Графический процессор - NVIDIA GeForce 1060 - 6GB;
- 16 ГБ оперативной памяти.

Несмотря на возможность обучения непосредственно процессором, обучение с использованием графического ускорителя показывает повышенную эффективность работы. В ходе практической работы обучение проводилось при помощи графического процессора с применением сжатия результирующего изображения. Исходные элементы датасетов были сжаты до размер 256x256 пикселей с целью уменьшения длительности обработки.

Реализация порождающей состязательной сети проведена на языке Python [7]. На данный момент язык программирования Python является стандартом для работы с глубокими сетями. Большинство вспомогательных библиотек

написано для языка Python.

TensorFlow [8] — открытая библиотека для машинного обучения, разработанная для решения задач построения и тренировки нейронной сети. Целью работы TensorFlow является автоматическое нахождение и классификация образов, подобное человеческому восприятию. TensorFlow хорошо подходит для автоматизированной аннотации изображений с высокой степенью ранжирования.

Visdom — инструментарий для графического отображения статуса работы и тренировки нейронных сетей. Является вспомогательным модулем фреймворка PyTorch. В ходе работы Visdom используется для отображения промежуточных результатов обучения сети, мониторинга графика функции потерь в ходе обучения.

Autodesk 3ds Max [9]— профессиональное программное обеспечение для 3D-моделирования, анимации и визуализации при создании игр и проектировании. В настоящее время разрабатывается и издается компанией Autodesk.

2.2 Подготовка тестовых наборов данных

Основной проблемой обучения нейронных сетей на сегодняшний день остается недостаточность данных. Для различных практических задач скорость обучения и требуемое число тестовых экземпляров для получения минимального ожидаемого показателя точности остается индивидуальным. С целью оптимизации процесса генерации тренировочного датасета были рассмотрены следующие подходы:

Первый реализованный подход заключается в подготовке упрощенного, но полноценного набора данных. Данный подход является наивным и требует выявления всех возможных характеристик рассматриваемых объектов. Использование минимального набора данных чувствительно к применению на образцах, отличающихся от основного набора. При этом задача выявления всех отличительных признаков и возможных значений является нетривиальной.

Второй подход предполагает подготовку полноценного набора данных с использованием конечных трехмерных моделей. Данная методика является наиболее эффективной с точки зрения конечного результата, при этом остается наиболее трудоемкой на этапе подготовки тренировочных данных. Увеличение числа обучающих наборов приводит к наибольшему разнообразию рассмот-

ренных объектов, что позволяет избегать артефактов в процессе применения сети на тестовых данных и в ходе практического применения.

Третий подход подразумевает построение дополнительных кадров за счет DIB-R сети. Данная сеть генерирует трехмерную модель объекта по одному изображению. Сеть предугадывает геометрические свойства объекта на основании имеющихся данных.

2.3 Анализ эффективности

В ходе работы проводился анализ параметров, необходимых для эффективной работы порождающей состязательной сети. В качестве рассматриваемых параметров были изучены количество слоев нейронной сети, количество эпох, на которых проводилось обучение, размер обучающего датасета. Эффективность обучения оценивалась согласно времени, затраченному на обучение, результатам функций потерь, а также наличию явных артефактов.

Для определения оптимального числа эпох были проведены измерения на трех типах датасетов:

— Малый набор (25 объектов);
— Полный набор (100 объектов);
— Дополненный набор (25 исходных изображений, 75 сгенерированных экзмемляра).

На указанных наборах данных проводилось обучение сети до 500 эпох. Значения функций потери cycle-consistent 1 приведены в таблицах:

Набор данных	50	100	150	200	300	400	500
Малый набор	1.948	1.625	1.532	1.110	1.102	1.097	1.089
Полный набор	1.544	1.327	1.107	1.102	1.094	1.090	1.087
Дополненный набор	1.521	1.378	1.111	1.104	1.092	1.097	1.092

Таблица 1 – Функции cycle-consistent loss

Результаты показали, что для датасета домов достаточным является обучение на 200 эпох. При этом значение функции ошибки для полного набора и дополненного начиная со 150 эпохи различаются незначительно, что говорит о целесообразности применения сгенерированных данных для дополнения исходного датасета.

3 Анализ полученных результатов

В третьем разделе приводится анализ полученных результатов и вывод об эффективности различных подходов к подготовке данных.

В ходе проведенного исследования показано, что обучение сети на датасете с использованием автоматически сгенерированных дополнительных данных имеет схожие показатели функций потерь с полноценным датасетом, подготовленным вручную. Несмотря на лучшие результаты работы с полным тренировочным набором, для подготовки дополненного датасета требуется значительно меньше исходных данных. Обучение на малом наборе дает наихудшие результаты и приводит к появлению различных артефактов на данных, отличных от исходных.

Так как сгенерированные изображения не исключают использования исходных изображений или рендеров, а только дополняют наборы, увеличивая число итераций в рамках одной эпохи обучения, общий уровень потерь при генерации изображений остается в области допустимых значений.

Использование готовых 3D моделей является наиболее результативным с точки зрения общего качества изображения и фотореалистичности. Механизм Differentiable Interpolation-based Renderer дает выигрыш во времени, уступая полноценному датасету в качестве. С учетом значительного выигрыша во времени и приемлемого уровня качества полученных изображений применение DIB-R сети для расширения датасета считается целесообразным.

ЗАКЛЮЧЕНИЕ

В настоящей работе реализована порождающая состязательная сеть для генерации фотореалистичных изображений на основе простых макетов.

В ходе работы проведено исследование существующих моделей нейронных сетей, проанализированы преимущества и недостатки конкретных реализаций, выбраны наиболее эффективные архитектуры. Подготовлены тестовые наборы данных для дальнейшего обучения сетей. На практике реализована сеть на основе архитектуры Cycle GAN. Проведено тестирование на различных наборах данных: частично размеченных и неразмеченных. Для работы с трехмерными объектами задействован модуль для обработки различных ракурсов одного объекта. Проведено сравнение с существующими реализациями Cycle GAN и альтернативными архитектурами для работы с изображениями. Подтверждена эффективность работы выбранной архитектуры для задачи генерации фотореалистичных изображений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Deng, Li*. Deep Learning. Methods and Applications [Электронный ресурс] / Li Deng, Dong Yu. — 2014. — URL: https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/ (Дата обращения 05.03.2019). Загл. с экр. Яз. англ.

2 *Goodfellow, Ian*. Deep Learning [Электронный ресурс] / Ian Goodfellow, Yoshua Bengio, Aaron Courville. — MIT Press, 2016. — URL: http://www.deeplearningbook.org (Дата обращения 13.08.2019). Загл. с экр. Яз. англ.

3 Generative adversarial nets [Электронный ресурс] / Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. — 2014. — URL: https://arxiv.org/abs/1406.2661 (Дата обращения 17.10.2018). Загл. с экр. Яз. англ.

4 *Zhu, Jun-Yan*. Unpaired image-to-image translation using cycle-consistent adversarial networkss // Computer Vision (ICCV), 2017 IEEE International Conference on. — 2017.

5 *Radford, Alec*. Unsupervised representation learning with deep convolutional generative adversarial networks / Alec Radford, Luke Metz, Soumith Chintala. — 2015.

6 *Chen, Wenzheng*. Learning to predict 3d objects with an interpolation-based differentiable renderer // Advances In Neural Information Processing Systems. — 2019.

7 *Лутц, М.* Программирование на Python / М. Лутц. — Символ-плюс, 2011.

8 TensorFlow. API Documentation [Электронный ресурс]. — URL: https://www.tensorflow.org/api_docs/ (Дата обращения 21.04.2019). Загл. с экр. Яз. англ.

9 Autodesk 3ds Max / Autodesk 3ds Max Design 2013 : Installation Help [Электронный ресурс]. — URL: http://download.autodesk.com/us/3dsmax/2013/faq/3dsMax2013_FAQ.pdf (Дата обращения 14.10.2019). Загл. с экр. Яз. англ.