**Министерство науки и высшего образования РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Уфимский университет науки и технологий»**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 100 |  |  |  |  |  |  |  |  |  |  |
| 90 |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |
| 70 |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |  |
| 30 |  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |
| 0 |  |  |  |  |  |  |  |  |  |  |

**СРАВНЕНИЕ И АНАЛИЗ АЛГОРИТМОВ ГЕНЕРАЦИИ ИЗОБРАЖЕНИЙ НЕЙРОСЕТЕВЫМИ МЕТОДАМИ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе по дисциплине

«Теория разностных схем»

**~~1507.333113.000 ПЗ~~**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа  ПМ-355 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Султанов Р.Р |  |  |  |
| Консультант | Ямилева А.М. |  |  |  |
| Принял | Лукащук В.О. |  |  |  |

Уфа 2023

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Уфимский университет науки и технологий»

Кафедра Высокопроизводительных вычислительных технологий и систем

**ЗАДАНИЕ**

на курсовую работу по дисциплине

**«Сравнение и анализ алгоритмов генерации изображений нейросетевыми методами»**

Студент: Султанов Равиль Радикович Группа: ПМ-355

Консультант: Ямилева Альфия Маратовна

1. Тема курсовой работы

Сравнение и анализ алгоритмов генерации изображений.

2. Основное содержание

~~2.1.~~ ~~Изучить существующие математические модели биологических нейронных сетей, симулирующих сигналы локального полевого потенциала.~~

~~2.2. Продемонстрировать эффективность полученных систем.~~

~~2.3. Изучить влияние параметров на поведение нейронной сети.~~

~~2.4. Продемонстрировать когерентность двух связанных нейронных сетей.~~

~~2.4. Оформить пояснительную записку к курсовой работе.~~

3. Требования к оформлению материалов работы

Требования к оформлению пояснительной записки

Пояснительная записка к курсовой работе должна быть оформлена в соответствии с требованиями ГОСТ и содержать

• титульный лист,

• задание на курсовую работу,

• содержание,

• введение,

• заключение,

• список литературы,

• приложение, содержащее листинг разработанной программы, если таковая имеется.

|  |  |
| --- | --- |
| Дата выдачи задания  "\_\_" \_\_\_\_\_\_\_\_ 202\_ г. | Дата окончания работы  "\_\_" \_\_\_\_\_\_\_\_\_\_ 202\_ г. |

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ямилева А.М.

**Содержание**

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc150027712)

[1.1. Генеративные модели 6](#_Toc150027713)

[1.2. Модель шума и дуффузии 7](#_Toc150027714)

[1.3. Оценка качества генеративных моделей 9](#_Toc150027715)

[1.4. Математическая постановка задачи 10](#_Toc150027716)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 12](#_Toc150027717)

[2.1 Программное обеспечение 12](#_Toc150027718)

[2.2. Подготовка данных 12](#_Toc150027719)

[2.3. Модель 13](#_Toc150027720)

[2.4. Эксперименты и Результаты 13](#_Toc150027721)

[3. ЗАКЛЮЧЕНИЕ 14](#_Toc150027722)

[СПИСОК ЛИТЕРАТУРЫ 15](#_Toc150027723)

[ПРИЛОЖЕНИЕ А 16](#_Toc150027724)

ВВЕДЕНИЕ

Цель: Исследование и реализация генеративной модели для генерации реалистичных изображении.

Актуальность: Генеративные модели представляют собой класс алгоритмов машинного обучения, целью которых является моделирование и генерация данных, которые похожи на обучающие данные. Важной характеристикой генеративных моделей является их способность создавать новые образцы данных на основе структуры и закономерностей, выявленных в исходных данных. Эти модели имеют широкий спектр применений в таких областях, как компьютерное зрение, обработка естественного языка и искусственный интеллект. Генеративные модели являются ключевым инструментом в сферах, где требуется создание новых реалистичных синтетических данных.

Задачи данной работы:

1. Изучить существующие генеративные модели, использующиеся для генерации изображений.
2. Реализация архитектуры генеративной модели с использованием библиотек глубокого обучения.
3. Обучение модели на наборе данных с изображениями.
4. Оценка качества сгенерированных изображений.
5. Визуализация результатов обучения и сгенерированных изображений.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Генеративные модели

Генеративные модели представляют собой класс алгоритмов машинного обучения, целью которых является моделирование и генерация данных, которые похожи на обучающие данные. Важной характеристикой генеративных моделей является их способность создавать новые образцы данных на основе структуры и закономерностей, выявленных в исходных данных. Эти модели имеют широкий спектр применений в таких областях, как компьютерное зрение, обработка естественного языка и искусственный интеллект. Важно отметить, что генеративные модели являются ключевым инструментом в сферах, где требуется создание новых данных, которые могли бы быть реалистичными и полезными.

Примеры генеративных моделей:

* Генеративные состязательные сети (GAN): GAN - это один из наиболее известных и распространнёных классов генеративных моделей. Он состоит из двух нейронных сетей, генератора и дискриминатора, которые соревнуются друг с другом. Генератор создает данные, а дискриминатор оценивает, насколько они реалистичны. GAN широко используются в генерации изображений, видео и в обработке звука. Например, они могут использоваться для генерации картин, создания реалистичных фотографий лиц или для улучшения качества фотографий.
* Вариационные автокодировщики (VAE): VAE - это другой класс генеративных моделей, который работает на основе автокодировщиков. Они способны не только генерировать данные, но и выполнять семантическую интерполяцию между образцами. Например, VAE могут быть использованы для генерации новых лиц, преобразования стилей изображений или генерации текстовых описаний изображений.

Генеративные модели имеют широкий спектр практических применений. Вот некоторые из них:

* Генерация контента: Генеративные модели могут использоваться для создания контента, такого как изображения, музыка, тексты и видео. Например, они могут генерировать изображения для веб-дизайна, создавать музыку, адаптированную к настроению слушателя, писать тексты.
* Увеличение объёма данных: Генеративные модели могут быть использованы для увеличения объема обучающих данных. Например, в области компьютерного зрения, они могут генерировать дополнительные изображения для обучения моделей распознавания объектов.
* Улучшение данных: Модели, такие как GAN, могут использоваться для улучшения качества существующих данных. Это может включать в себя увеличение разрешения изображений, устранение шума или улучшение деталей.
* Семантическая интерполяция: Генеративные модели могут выполнять семантическую интерполяцию между образцами. Например, они могут создавать плавные переходы между изображениями разных стилей или содержания.
* Генерация данных для исследований: Генеративные модели могут использоваться для создания данных в научных исследованиях. Например, они могут генерировать синтетические данные для исследований в области медицины, физики или других дисциплин.

**Denoising Diffusion Implicit Model (DDIM)**

Denoising Diffusion Implicit Model (DDIM) - это относительно новая генеративная модель. DDIM основан на диффузии, статистическом процессе, который моделирует распространение шума в данных. Основная идея DDIM заключается в обучении модели постепенно очищать (декодировать) данные от шума, создавая реалистичные образцы. DDIM может быть использован для генерации изображений, видео, звуков и других типов данных.

1.2. Модель шума и дуффузии

Шум - представляет собой случайные колебания в сигнале или измерении, которые могут оказывать влияние на точность и качество данных. Математически шум описывается как случайный сигнал, который добавляется к основному сигналу.

**Получение зашумленных изображений**

В случае Denoising Diffusion Implicit Model (DDIM), зашумление изображений происходит путем применения процесса диффузии. Процесс диффузии в данном контексте можно представить как последовательное увеличение уровня шума в изображении. Шум в данной работе имеет случайную природу и подчиняется нормальному закону распределения.

(1)

Предварительно все изображаения нормализованны так, чтобы среднее значение пикселей равно 0, а дисперсия равна 1.

(2)

Зашумленное изображение представляет собой взвешенную сумму исходного сигнала (изображения) и шума :

(3)

где

(4)

- уровень исходного сигнала, - уровень шума.

Так как случайные шумы (стандартное нормальное распределение) и изображения (нормализованные) оба имеют нулевое среднее и единичное стандартное отклонение, зашумленные изображения всегда имеют единичную дисперсию, так же, как и их нешумные компоненты. Таким образом:

(5)

**Диффузионное расписание (Diffusion schedule)**

Предполагается, что диффузия происходит за единичное время в нескольких этапов. В нулевой момент времени уровень исходного сигнала равен ноль, а уровень шума - 0: , . В конце () - наоборот: , . Для вычисления уровня каждого сигнала в зашумленном изображении в промежуточные моменты времени используется специальная функция - *диффузионное расписание* ( - diffusion schedule).

выдает два значения: скорость шума () и скорость сигнала ().

В данной работе используется упрощенная непрерывная версия косинусного расписания, которое широко применяется в литературе [ССЫЛКА]. В этой модели уровни сингалов определяются следующим образом:

,

, (6)

где

,

. (7)

- количество шагов диффузии. Параметры и выбираются близкими к нулю и единице соответсвенно и подбираются экспериментальным путём.

1.3. Оценка качества генеративных моделей

**Kernel Inception Distance (KID)**

Метрика *Kernel Inception Distance* (KID) представляет собой меру различия между сгенерированными изображениями и обучающими данными в пространстве представлений, созданном моделью InceptionV3, предварительно обученной на наборе данных ImageNet. Метрика KID является альтернативой популярной метрике *Frechet Inception Distance* (FID) и обеспечивает оценку качества генерации изображений. Преимущество метрики KID заключается в более простой реализации, меньшей вычислительной сложности, также она лучше подходит для небольших датасетов [СТАТЬЯ], которые используются в нашем случае.

Метрика KID между двумя множествами векторов , определяется следующей формулой:

где

(9)

- функция ядра, - размерность векторов.

**Maximum Mean Discrepancy (MMD)**

KID является эмпирической оценкой *максимального среднего несоответствия* (MMD - maximum mean discrepancy), которое измеряет расстояние между двумя распределениями в пространстве RKHS. RKHS — это гильбертово пространство функций [СТАТЬЯ], связанных с определенной функцией ядра (kernel function). Понятие функции ядра играет важную роль в оценке KID, так как ядро определяет, как функции представления связаны в RKHS. KID в свою очередь является эмпирической оценкой на основе экземпляров двух распределений.

MMD и KID вычисляются с использованием kernel functions, которые представляют собой функции, оценивающие сходство между элементами двух распределений. Основная идея заключается в том, что MMD будет равно нулю, только если оба распределения совпадают. Это свойство делает MMD полезным инструментом для оценки качества генерации изображений.

Таким образом, KID обеспечивает оценку качества генерации изображений, используя математические концепции MMD, kernel functions и RKHS. Эти инструменты позволяют количественно измерить различие между сгенерированными и реальными данными и оценить качество генеративных моделей.

1.4. Математическая постановка задачи

Целью данной курсовой работы является создание генеративной модели, способной генерировать изображения, которые могли бы быть непроизвольными, но при этом сохраняли бы натуральный стиль и качество. Для этого формулируется задача обучения глубокой модели, которая обучается находить две компоненты в каждом сгенерированном изображении: шум и сигнал. Входной шум представлен как случайное изображение, а сигнал - как реальное изображение. Обучаемая модель разделяет шум и сигнал, что позволяет ей генерировать изображения.

В процессе обучения модель корректируется так, чтобы минимизировать ошибку между исходными изображениями и сгенерированными изображениями, где ошибка измеряется в виде средней абсолютной разницы между пиксельными значениями.

Качество генерации оценивается с использованием метрики KID. Метрика KID измеряет разницу между сгенерированными и обучающими распределениями в пространстве представлений нейронной сети InceptionV3, предварительно обученной на ImageNet. Для метрики используется представление именно в таком пространстве, потому что InceptionV3 - универсальная сеть, предварительно обученная на крупном и разнообразном наборе данных ImageNet. Она имеет обширные знания о различных объектах и стилях визуального мира. Это делает ее хорошим выбором для извлечения признаков из изображений.

Вместо того, чтобы сравнивать изображения непосредственно, KID сравнивает представления этих изображений в пространстве InceptionV3. Это позволяет учесть сходство между изображениями на более абстрактном уровне, что часто более информативно для задачи оценки качества генерации.

Пусть:

- - преобразование, накладывающее шум на изображение

- - преобразование, восстанавливающее зашумлённое изображение . - набор параметров модели.

- - множество реальных изображений,

- - множество реальных изображений, после наложения шума.

- - множество сгенерированных изображений, т.е. восстановленных зашумленных изображений

- - функция, отображающая изображения в представления в пространстве InceptionV3,

- - функция ядра между представлениями и , где

и - размерность представлений.

Тогда метрика KID между и определяется следующим образом:

где:

- - количество элементов в каждом из множеств и ,

- и - представления реальных изображений,

- и - представления сгенерированных изображений,

- - функция ядра, как описано выше.

Теперь задача формулируется следующим образом: найти параметры модели Denoise, при которых метрика KID будет минимальна:

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Программное обеспечение

Код программы был реализован на языке Python, с использованием специализированной библиотеки Tensorflow. Она предоставляет инструменты для определения сложных нейронных сетей и эффективного обучения.

2.2. Подготовка данных

Для проведения экспериментов и обучения генеративной модели были использован наборы данных Oxford Flowers 102 и Caltech-UCSD Birds 200 (CUB-200). Oxford Flowers 102 представляет собой разнообразный собранный изображений природы и содержит около 8,000 изображений цветов. CUB-200 - набор данных изображений с фотографиями 200 видов птиц (в основном североамериканских), в наборе данных 2011 года содержится 11 788 изображений.

Для обеспечения адекватного и сбалансированного набора данных для обучения модели, были выполнены следующие этапы:

1. Разбиение данных:

Исходный набор данных был подвергнут разделению на тренировочную (80% данных) и валидационную (20% данных) выборки. Этот процесс разделения данных был выполнен с использованием Tensorflow Datasets slicing API. Это позволило устранить проблему несбалансированности и обеспечить адекватное распределение данных для обучения и оценки модели.

1. Препроцессинг с использованием центральных обрезок:

В качестве части процесса предобработки данных были выполнены центральные обрезки изображений. Отбросывание неинформативной области изображения, может быть полезным для улучшения качества обучения модели.

1. Нормализация:

Для обеспечения стабильности обучения и улучшения результатов модели, данные были нормализованы. Также этого требует используемая модель диффузии. Вместо перевода значений в диапазон [-1, 1], как это делается обычно, было проведено нормализация данных, чтобы они имели среднее значение 0 и единичное стандартное отклонение.

2.3. Модель

В качестве архитектуры нейросети была использована U-Net c 3'901'254 параметрами.

Модель имеет следующий алгоритм обучения:

1. Модель принимает на вход чистое изображение из датасета и нормализует его.
2. Генерация накладываемого шума
3. Генерация уровней сигнала и шума с помощью DS
4. *Noise*: смешивание исходного изображения с шумами
5. *Denoise*: модель пытается отделить исходный сигнал и шум в зашумлённом изображении
6. Расчёт ошибки: насколько полученное изображение, отличается от исходного
7. Корректирование параметров модели для уменьшения ошибки

2.4. Эксперименты и Результаты

Модель обучалась 50 эпох. Важно иметь в виду, что обучение модели является итеративным процессом, и каждая эпоха представляет собой одну полную итерацию через весь тренировочный набор данных.

Функция потерь для изображений (Image Loss) уменьшались с каждой последующей эпохой. Это указывает на то, что модель эффективно училась генерировать изображения и приближать их к данным из обучающего набора.

В начале обучения был виден значительный прогресс в уменьшении ошибки как на обучающем, так и на валидационном наборе. Начиная примерно с 6-й эпохи, обучение стабилизировалось, и дополнительные эпохи обучения не привели к значительному улучшению модели.

KID-оценка на валидационном наборе также остается относительно постоянной, что может указывать на то, что модель достигла своего предельного качества генерации.

Из этой информации можно сделать вывод, что модель обучена до некоторой точки, и, возможно, для дальнейшего улучшения качества генерации изображений следует провести дополнительные исследования и эксперименты, возможно, изменить архитектуру или параметры обучения.

3. ЗАКЛЮЧЕНИЕ

Была проведена серия экспериментов по обучению модели генерации изображений цветов и птиц. Эти эксперименты включали в себя настройку архитектуры модели, выбор параметров обучения и анализ метрик. Исходя из результатов, можно заключить, что модель обучена до определенной степени, и она способна генерировать изображения, которые близки к данным из обучающего набора. Однако некоторые аспекты могут потребовать улучшения, такие как сходство с валидационными данными, выраженное в KID-оценке. Результаты служат хорошей отправной точкой для будущих исследований и доработок. Следующие шаги могут включать в себя эксперименты с более сложными архитектурами, оптимизацию параметров обучения или использование более крупных и разнообразных наборов данных для улучшения качества генерации изображений.

СПИСОК ЛИТЕРАТУРЫ

~~1. Introduction to Theoretical Neurobiology. / Tuckwell HC. // Cambridge, UK: Cambridge Univ. Press, 1988~~

~~2. Arnd Roth, Mark C. W. van Rossum. Modeling Synapses // Computational Modeling Methods for Neuroscientists / MIT Press, 2009 – P. 139-159~~

~~3. Hasselmo, M. E., Bodelon, C., & Wyble, B. P. A proposed function for hippocampal theta rhythm: Separate phases of encoding and retrieval enhance reversal of prior learning. // Neural Computation, 14 – 2002. – P. 793–817~~

~~4. Mysin, I., & Shubina, L. (2022). From mechanisms to functions: The role of theta and gamma coherence in the intrahippocampal circuits. Hippocampus, 1–17.~~

~~5. Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration // Northwestern University, Department of Neurobiology & Physiology, – 2008.~~

~~6. Brunel, Nicolas and Xiao-Jing Wang. What Determines the Frequency of Fast Network Oscillations With Irregular Neural Discharges? I. Synaptic Dynamics and Excitation-Inhibition Balance // Centre National de la Recherche Scientifique-Neurophysique et Physiologie du Syste`me Moteur-Universite´ Paris Rene´ Descartes, Brandeis University, Waltham, Massachusetts – 2003~~

~~7. Brian 2 documentation [Электронный ресурс] Brian 2 2.5.1 documentation — Электрон. дан. — 2021 — Режим доступа: https://brian2.readthedocs.io/en/stable/ Загл. с экрана. — Англ.~~

ПРИЛОЖЕНИЕ А

**Листинг программы**

**fast\_oscill.py**

**ПЛАН-ГРАФИК**

**выполнения курсовой работы**

обучающегося Султанова Р.Р.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапа работ | Трудоемкость выполнения, час. | Процент к общей трудоемкости выполнения | Срок предъявления консультанту |
| Получение и согласование задания | 0,3 | 0,8 | 6 неделя |
| Знакомство с литературой по теме курсовой работы | 2,7 | 7,5 | 7 неделя |
| ~~Теоретический анализ моделей нейрон~~ | ~~3~~ | ~~8,3~~ | 8 неделя |
| ~~Поиск программной среды и библиотек, подходящих для~~ | ~~2~~ | ~~5,6~~ | 9 неделя |
| ~~Программирование моделей одиночного нейрона~~ | ~~3~~ | ~~8,3~~ | 10 неделя |
| ~~Программирование моделей совокупности нейронов~~ | ~~4~~ | ~~11,2~~ | 11 неделя |
| ~~Отладка моделей~~ | ~~5~~ | ~~13,9~~ | 12 неделя |
| ~~Программирование элементов отображения состояния модели~~ | ~~3~~ | ~~8,3~~ | 13 неделя |
| ~~Отладка программы~~ | ~~5~~ | ~~13,9~~ | 14 неделя |
| ~~Изучение моделей с помощью программы~~ | ~~5~~ | ~~13,9~~ | 15 неделя |
| Составление и оформление пояснительной записки и подготовка к защите | 2,7 | 7,5 | 16 неделя |
| Защита | 0,3 | 0,8 | 17 неделя |
| Итого | 36 | 100 | - |