Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Научно–образовательный центр «Высшая ИТ школа»

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
д–р. физ.–мат. наук, профессор
_____ О.А.Змеев
подпись
«11» июня 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ИСПОЛЬЗОВАНИЕ ГРАФОВЫХ СВЁРТОК И ГЕНЕРАТИВНОГО ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ АДАПТАЦИИ ГЕОЛОГИЧЕСКИХ МОДЕЛЕЙ

по направлению подготовки 09.03.04 Программная инженерия
направленность (профиль) «Программная инженерия»

Выгон Роман Соломонович

Руководитель ВКР
Д–р. Физ.–мат. Наук, профессор
_____О.А. Змеев
подпись
«10» июня 2022 г.

Консультант ВКР
аспирант
_____Г.Ю. Шишаев
подпись
«08» июня 2022 г.

Автор работы
студент группы № 971810
_____ Р.С. Выгон
подпись
«08» июня 2022 г.

Томск – 2022

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
НОЦ «Высшая ИТ школа»

УТВЕРЖДАЮ
руководитель ООП
д-р.физ.-мат. наук, профессор
_____ О.А.Змеев
«21» февраля 2022 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающемуся
Выгону Роману Соломоновичу
(Ф.И.О. обучающегося)

по направлению подготовки Программная инженерия, направленность «Программная инженерия»

1. Тема выпускной квалификационной работы бакалавра
Использование графовых свёрток и генеративного глубокого обучения для адаптации геологических моделей

2. Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) в учебный офис – «09» июня 2022г.
б) в ГЭК – «11» июня 2022г.

3. Исходные данные к работе:
Цель работы – разработка системы алгоритмов машинного обучения, способной сжимать информацию о графах для решения задачи адаптации геологических моделей.
(цели и задачи ВКР, ожидаемые результаты)

Задачи работы:
1) сгенерировать датасет геологических карт; 2) подобрать архитектуру графовой сети; 3) обучить полученную нейросеть; 4) модифицировать архитектуру для оценки плотности скрытого пространства автокодировщика; 5) подобрать алгоритм для адаптации геологических моделей.

Ожидаемые результаты – проведен анализ задачи, представлено решение на основе нейросетей с графовыми свёртками, а также алгоритма эволюционной стратегии для адаптации геологических моделей

Организация, по тематике которой выполняется работа
Томский Политехнический Университет

Руководитель выпускной квалификационной работы
д-р физ.-мат. наук, профессор кафедры программной инженерии _____ / О.А.Змеев
(должность, место работы) (подпись) (И.О. Фамилия)

Консультант выпускной квалификационной работы
инженер, ТПУ _____ / Г.Ю. Шипаков
(должность, место работы) (подпись) (И.О. Фамилия)

Задание принял к исполнению
«18» февраля 2022 г _____ / Р.С. Выгон
(дата) (подпись) (И.О. Фамилия)

АННОТАЦИЯ

Выпускная квалификационная работа: 40 стр., 17 рис., 2 табл., 1 лист., 36 источников.

МАШИННОЕ ОБУЧЕНИЕ, ML, ГЕНЕРАТИВНЫЕ НЕЙРОННЫЕ СЕТИ, ЭВОЛЮЦИОННЫЕ СТРАТЕГИИ, РЕФАКТОРИНГ, CI/CD

Объект разработки: система алгоритмов автоматической адаптации геологических моделей.

Цель работы: разработка системы алгоритмов машинного обучения, способной сжимать информацию о графах для решения задачи адаптации геологических моделей.

Результаты работы: в ходе работы над исследованием был предложен новый способ для преобразования геологических моделей в графы и наоборот. Была разработана архитектура модели глубокого обучения, способная сжимать информацию об исходном графе в 80 раз для получения более компактного представления, а также оценивать плотность пространства таких представлений. Данная модель была обучена и успешно протестирована на собранном наборе данных, что подтвердило исходную гипотезу исследования.

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ 5
ВВЕДЕНИЕ 7
1 Описание задачи 9
1.1 Существующие методы 10
1.2 Графовые сети 12
2 Датасет 14
3 Машинное обучение и используемые модели 19
3.1 Генеративная сеть 19
3.1.1 Графовые свёртки 19
3.1.2 Автокодировщики 19
3.2 Метрики и функции потерь 23
3.2.1 Среднеквадратичное отклонение (англ. Mean Squared Error, MSE) . 23
3.2.2 Коэффициент Жаккара (англ. Intersection over Union, IoU) 23
3.3 Оценка плотности пространства автокодировщика 25
3.3.1 Риманова геометрия 25
3.4 CMA–ES 27
3.4.1 Эволюционные стратегии 27
4 Эксперименты 29
4.1 Сравнение графовых свёрток 29
4.2 Влияние WMSE на качество восстановления гридов 31
4.3 Методика обучения 31
4.4 Используемые инструменты 32
4.4.1 Машинное обучение 32
4.4.2 tNavigator 32
5 Результаты 34

ПЕРЕЧЕНЬ РИСУНКОВ

Рисунок 1 – Пористость 9
Рисунок 2 – Проницаемость 9
Рисунок 3 – Различные формы тел в объектном моделировании 11
Рисунок 4 – Пример графа из датасета 14
Рисунок 5 – Гистограмма распределения пористости в датасете 16
Рисунок 6 – Распределение проницаемости в активных ячейках 17
Рисунок 7 – Распределение пористости в активных ячейках 17
Рисунок 8 – Распределение проницаемости активных ячеек после преобразования 18
Рисунок 9 – Архитектура кодировщика 21
Рисунок 10 – Архитектура декодировщика 22
Рисунок 11 – Гистограммы проницаемости исходных и восстановленных гридов 25
Рисунок 12 – Двумерная проекция скрытого пространства автокодировщика 26
Рисунок 13 – Метрика AMSE для сетей с GCNConv и GraphConv 29
Рисунок 14 – Метрика ELBO для сетей с GCNConv и GraphConv 30
Рисунок 15 – Распределение проницаемости, восстановленной с помощью GCNConv 30
Рисунок 16 – Влияние WMSE на метрику AMSE 31
Рисунок 17 – Интерфейс tNavigator 33
Рисунок 18 – Гистограммы пористости ячеек из датасета и восстановленных гридов 34

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

Геологическая карта (грид) – матрица, содержащая числовые значения геологических свойств местности.

Свёртка – операция, заключающаяся в перемножении карты признаков на двигающееся ядро (матрицу весов).

Свёрточная сеть – нейронная сеть, слои которой являются свёртками.

Адаптация модели – процесс построения грида по известным свойствам из нескольких точек на местности, а также по историческим данным по добыче нефти.

Гидродинамика – динамические свойства месторождений: темп добычи нефти или воды, давление и т.д.

Датасет (англ. dataset, набор данных) – выборка данных. В данной работе под данными понимаются аудиофайлы человеческой речи.

Валидация (англ. validation) – процесс верификации модели на тестовом наборе данных.

Графовая нейронная сеть (ГНН) – особый тип нейронных сетей, позволяющий работать с графовыми структурами без предварительной обработки данных.

Фации –пласт почвы, отличающийся на всём протяжении одинаковыми литологическими свойствами и включающий одинаковые органические ископаемые осадки.

Граф – структура данных, моделирующая множество объектов (вершин) и связей между ними (рёбер). Формально, граф G есть упорядоченная пара $(V(G), E(G))$ где $V(G)$ – множество вершин графа G, а $E(G)$) – множество ребер графа G [36]. Все графы в данной работе следует считать неориентированными и простыми. Это означает, что все рёбра имеют два неупорядоченных конца, не равных друг другу.

Эпоха – одна итерация машинного обучения, за которую нейросеть обучается на всех примерах из обучающей выборки по одному разу.

Автокодировщик (англ. Autoencoder, автоэнкодер**) –** нейронная сеть, состоящая из двух компонентов – кодировщика и декодировщика. Основной задачей таких сетей является воспроизведение входных данных. При этом на сеть автокодировщик накладываются дополнительные ограничения – размер промежуточного слоя между кодировщиком и декодировщиком должен быть меньше размера входного слоя.

Скрытое (латентное) пространство – векторное пространство, в которое сеть-кодировщик проецирует входные данные. Проекции входных данных в скрытое пространство называются скрытыми (латентными) кодами или представлениями.

ВВЕДЕНИЕ

Адаптация геологической модели (англ. History matching, АГМ) – процесс, в котором входные параметры симулятора месторождений (пористость, проницаемость, насыщенность почвы) изменяются по отдельности или вместе, чтобы минимизировать разницу между результатами симуляции и наблюдаемыми историческими данными месторождения, такими как дебиты воды, нефти или газа. Пространственная неоднородность и анизотропность пород–коллекторов приводят к большой размерности геологической модели, что усложняет задачу адаптации.

Все чаще для автоматического подбора лучших параметров используются методы генеративного машинного обучения. Существуют различные решения на основе генеративных состязательных сетей [12] и вариационных автоэнкодеров [15], однако они состоят из конвенциональных свёрточных сетей, где ядра свёрток являются многомерными матрицами, из–за чего данные приходится приводить к прямоугольной форме, что является ограничением.

В данной работе геологические карты являются графами, что позволяет анализировать месторождения со сложной геометрией – сдвигами пород, большими непроницаемыми участками. Это достигается за счёт использования графовых свёрток в архитектуре генеративной модели.

Помимо обученной генеративной сети для задачи адаптации также необходим алгоритм, интерпретирующий скрытое пространство автокодировщика для нахождения грида, наиболее подходящего под статические и динамические данные месторождения. Однако АГМ – некорректно поставленная обратная задача, ведь одни и те же результаты симуляцию достигаются бесконечным множеством различных гридов, многие из которых нереалистичны. Чтобы избежать риска генерации неправдоподобных карт, в оптимизируемую метрику была добавлена оценка плотности скрытого пространства автокодировщика.

Таким образом, в данной работе описываются двухэтапный процесс адаптации геологических моделей. На первом этапе обучается генеративная сеть–автокодировщик. Затем, данная сеть используется вкупе с эволюционным алгоритмом для получения грида, соответствующего требуемым гидродинамическим свойствам. Для построения такого процесса были поставлены следующие задачи:

Задачи работы

1. Сгенерировать датасет геологических карт стандартными методами моделирования.
2. Подобрать архитектуру модели–автокодировщика, основанную на графовых операциях.
3. Обучить полученную модель на датасете гридов.
4. Модифицировать архитектуру для оценки плотности скрытого пространства автокодировщика.
5. Подобрать алгоритм оптимизации и метрики качества адаптации геологических моделей.
6. Оценить возможность использования разработанной системы алгоритмов для решения задачи АГМ.

1 Описание задачи

В данной работе адаптация геологической модели производится с помощью подбора двух ключевых свойств породы – пористости и проницаемости. Пористость (Рисунок 1) – отношение порового пространства к объему грунта. Проницаемость (Рисунок 2) определяет способность порового пространства проводить жидкости и газы [8].

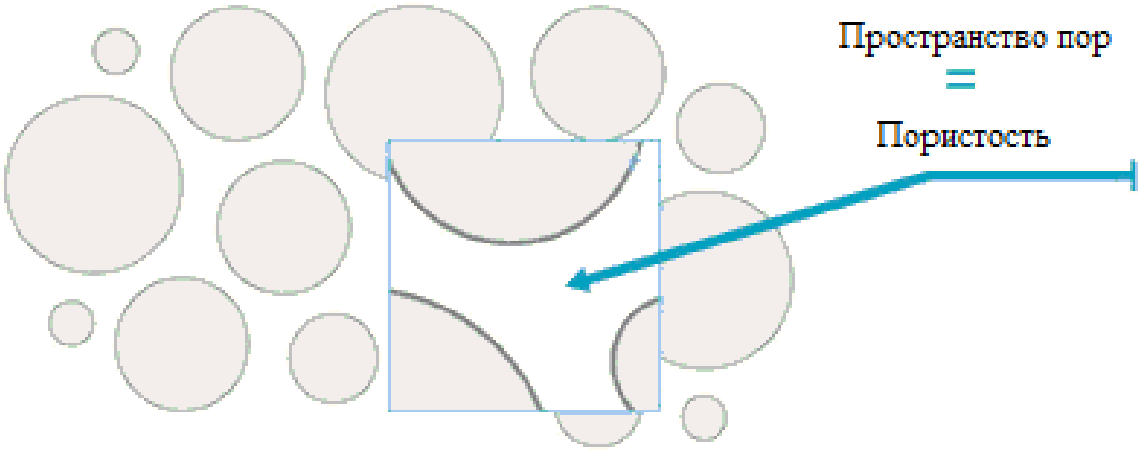


Рисунок 1 – Пористость

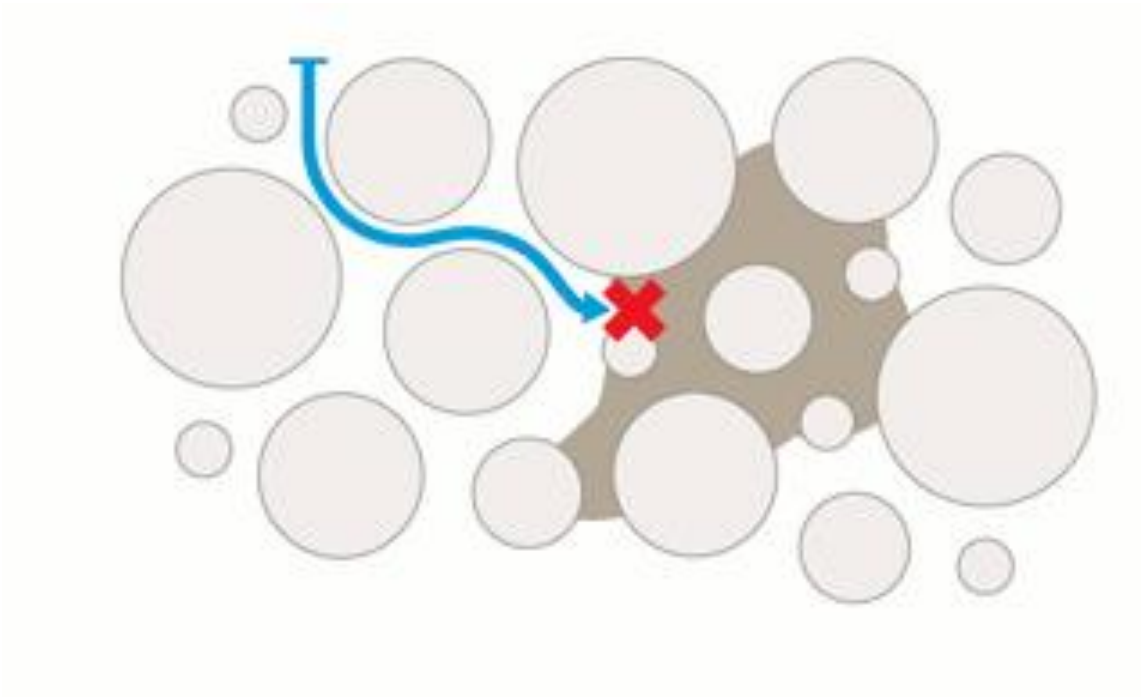


Рисунок 2 – Проницаемость

1.1 Существующие методы

Почти с самого их появления компьютеры использовались для решения задачи адаптации геологических моделей [34]. Зачастую задача сводилась к решению системы дифференциальных уравнений [7]. Решив такую систему, можно было получить значения чувствительности (англ. sensitivity) для пористости и проницаемости, которые использовались подобно градиентам для подсчета новых значений свойств. В машинном обучении такой метод пока удалось реализовать только на синтетических месторождениях [23].

Пиксельные методы моделирования, появившиеся ещё в двадцатом веке [18], до сих пор являются стандартным выбором алгоритма для АГМ. Они основываются на предположении о том, что распределение свойства по всему пространству обладает стационарностью. Стационарный процесс – это стохастический процесс, у которого не изменяется распределение вероятности при смещении во времени, то есть корреляция между значениями параметра в различных точках пространства зависит лишь от расстояния между точками, но не от их координат. Более продвинутый аналог пиксельных методов – объектное моделирование, использовался в данной работе для генерации обучающего набора данных. В процессе данного моделирования некие объекты (тела, фигуры) распределяются в пространстве случайным образом. Объекты характеризуются некоторыми параметрами, как стохастическими (к ним относится направление, ориентация или размеры), так и детерминированными (формы) (Рисунок 3). Все параметры характеризуются средними значениями и среднеквадратичными отклонениями. Процесс такого моделирования можно описать следующими шагами:

1. определение центра нового объекта, в соответствии с трендом интенсивности фации случайным образом;
2. генерирование непосредственного объекта. Для этого генерируются параметры этого объекта (например, размер и ориентация) в соответствии с заданными законами распределения;

3. проверка согласованности полученного объекта с скважинными данными. Если противоречий не обнаружено, то объект добавляется в грид. Если обнаружены противоречия, то либо объект не включают в модель и переходят к моделированию следующего, либо удаляют из уже существующей модели те объекты, которым противоречит вновь созданный, и включают его в модель;

4. проверка условий согласованности моделей, после включения нового объекта в модель.

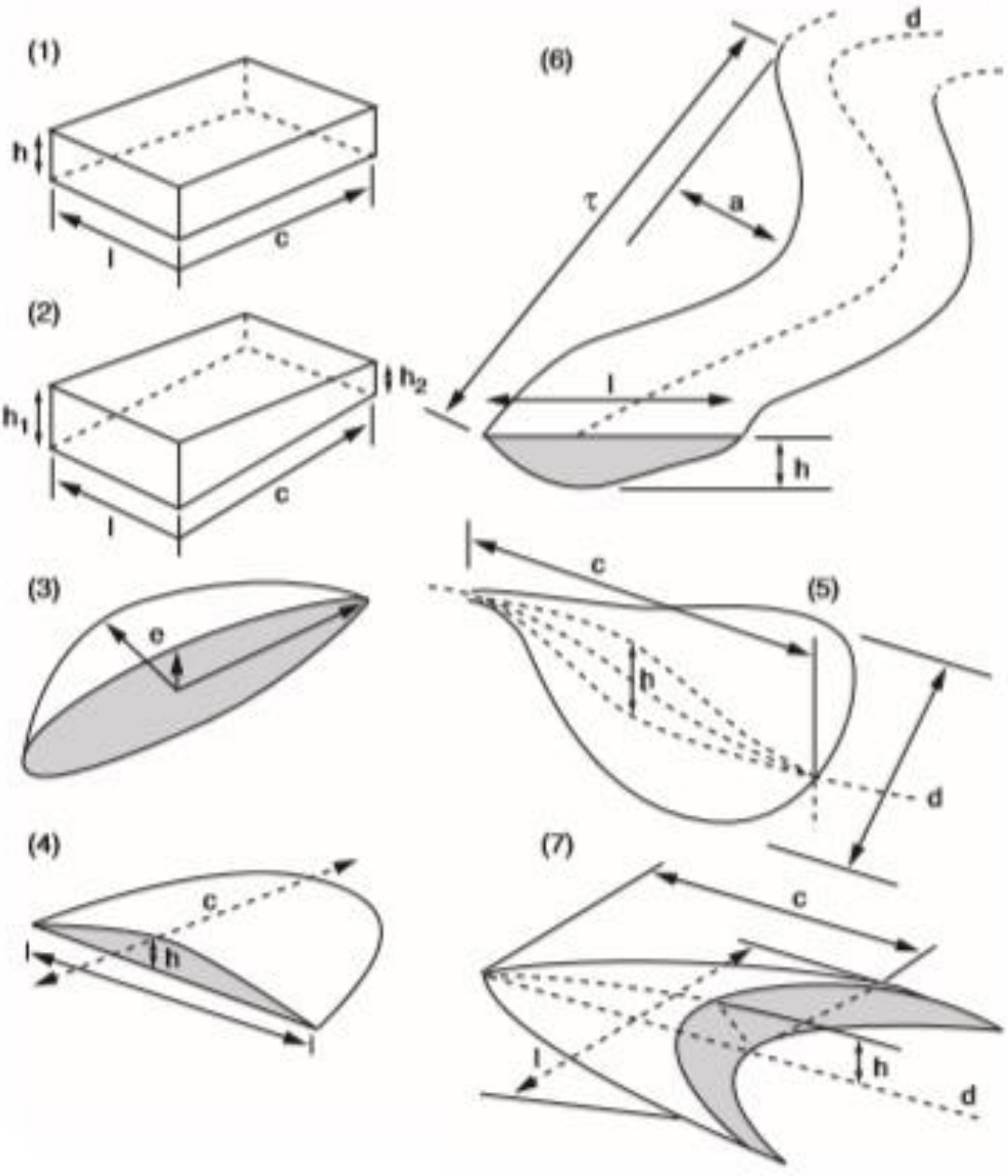


Рисунок 3 – Различные формы тел в объектном моделировании

Значительный прогресс в сфере генеративных моделей способствовал появлению методов моделирования тех или иных геологических карт, использующих глубокие свёрточные нейронные сети [10]. Первые такие работы фокусировались на решении двумерного варианта задачи [6]. В дальнейшем было показано [5], что полученные генеративные сети можно эффективно использовать для условной реконструкции геологических карт по статическим данным с месторождения.

Такой же развитие произошло и с генерацией трехмерных гридов. Моссер и др. обучил [24] генеративную состязательную сеть на объёмном датасете, а затем применил [25] похожую архитектуру для условной генерации томографических изображений известняка.

В данной работе используется архитектура сетей-автокодировщиков. Для задачи АГМ она применяется реже [17,27], но достигает сравнимых результатов с генеративными состязательными сетями.

1.2 Графовые сети

Особенно популярны графы в задачах, связанных с социальной наукой, графами знаний и взаимодействиями белков.

Первые работы, применяющие машинное обучение к графам, использовали рекурсивные нейронные сети [32] для создания модели состояний и переходов, что ограничивало возможности расширения таких алгоритмов. ГНН обрели новую жизнь с популяризацией свёрточных сетей, стали появляться публичные датасеты, такие как Cora [21] и PubMed [30], которые состоят из одного большого графа, в котором вершинами являются научные статьи, а ребра между ними показывают цитирования. Такие датасеты не подходят для апробирования нейросетей из данной работы, т.к. содержат лишь один граф, к тому же публичные результаты по этим наборам данных опубликованы лишь для задач кластеризации, классификации вершин и восстановления рёбер, в данной же работе решается задача восстановления свойств вершин.

Задачи, решаемые в сфере графовых сетей, можно разделить на несколько классов:

Задачи графового уровня

В таких задачах требуется предсказать одно значение для всего графа – например, класс, к которому этот граф относится.

Задачи вершинного уровня

В задачах этого типа нужно предсказать некоторую величину для каждой вершины графа. К этому типу и относится нейросеть, описанная в данной работе, т.к. она предсказывает значения исходных свойств каждой вершины.

Задачи реберного уровня

В задачах реберного уровня необходимо либо предсказать наличие ребра между некоторыми вершинами, либо посчитать величину, описывающую существующие ребра.

2 Датасет

Обучающий набор данных был сгенерирован методом объектного моделирования в программной среде Petrel [26]. Всего было получено 5000 различных гридов. Каждый такой грид состоит из 1919 ячеек, обозначающих куб почвы размером $50м^3$. Для всех ячеек известны численные значения пористости и проницаемости, а также их трехмерные координаты. Координаты по осям x и y дискретны и принимают значения от 0 до 15 и от 0 до 11 соответственно. Значения высоты заданы для всех 8 вершин куба, непрерывны и варьируются от -2480 до -2420. При построении грид напоминает по форме эллиптический параболоид.

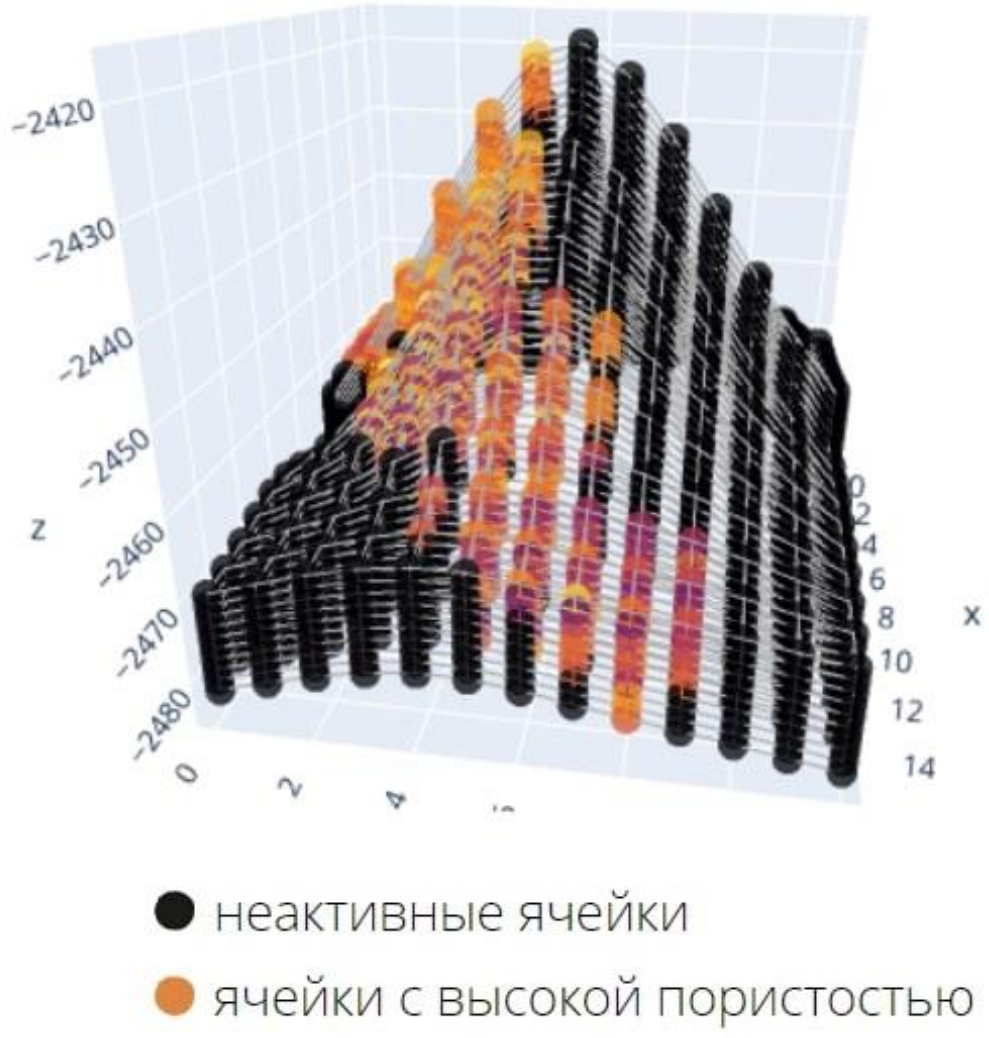


Рисунок 4 – Пример графа из датасета

Гриды датасета были сконвертированы в графы путём соединения соседних ячеек – вершины u и v считаются соседними, если выполняются условия (1) и (2):

$$|x_v - x_u| \leq 1, |y_v - y_u| \leq 1, \quad (1)$$

$$\exists (S_v, S_u), R_v = \left\{x \in \mathbb{R}: Z_{S_v}^1 \leq x \leq Z_{S_v}^2\right\},$$
$$R_u = \left\{x \in \mathbb{R}: Z_{S_v}^1 \leq x \leq Z_{S_v}^2\right\}, R_v \cap R_u \neq \emptyset, \quad (2)$$

где S – грань куба ячейки, а Z_S^1 и Z_S^2 меньшая и большая из высот вершин грани, соответственно.

Таким образом, графы в датасете имеют 1919 вершин и 5285 рёбер.

В результате анализа данных было выяснено (Рисунок 5), что распределения пористости и проницаемости в датасете являются бимодальными. Это объясняется тем, что лишь 18,9% ячеек датасета являются активными, т.е. имеют пористость и проницаемость, отличную от нуля. Распределение пористости в активных ячейках нормально (Рисунок 7), а распределение проницаемости напоминает экспоненциальное или логнормальное (Рисунок 6).

Для того, чтобы превратить распределение проницаемости в нормальное, был использована трансформация Йео-Джонсона [35] из семейства преобразований Бокса–Кокса [4]. Формально, к свойствам была применена следующая функция:

$$y_i^{(\lambda)} = \begin{cases} (\frac{y_i + 1)^\lambda - 1}{\lambda}, & \lambda \neq 0, y \geq 0 \\ \log(y_i + 1), & \lambda = 0, y \geq 0 \\ -\frac{(-y_i + 1)^{(2-\lambda)} - 1}{2 - \lambda}, & \lambda \neq 2, y < 0 \\ -\log(-y_i + 1), & \lambda = 2, y < 0 \end{cases}, \quad (3)$$

где y_i – исходные значения проницаемости, $y_i^{(\lambda)}$ – преобразованные значения проницаемости, а λ – параметр, максимизирующий функцию правдоподобия:

$$f(\lambda) = -\frac{N}{2\log(\hat{\sigma}^2)} + (\lambda - 1)\sum_i sign(x_i)\log(|x_i| + 1)\,, \qquad (4)$$

где $\hat{\sigma}$ – стандартное отклонение преобразованной проницаемости.

Результат преобразования можно увидеть на рисунке 8.

Для облегчения задачи восстановления свойств пористость и проницаемость были нормализованы. Формально, к ним было применено следующее преобразование:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \qquad (5)$$

где x исходные значения признака, а x' – преобразованные.

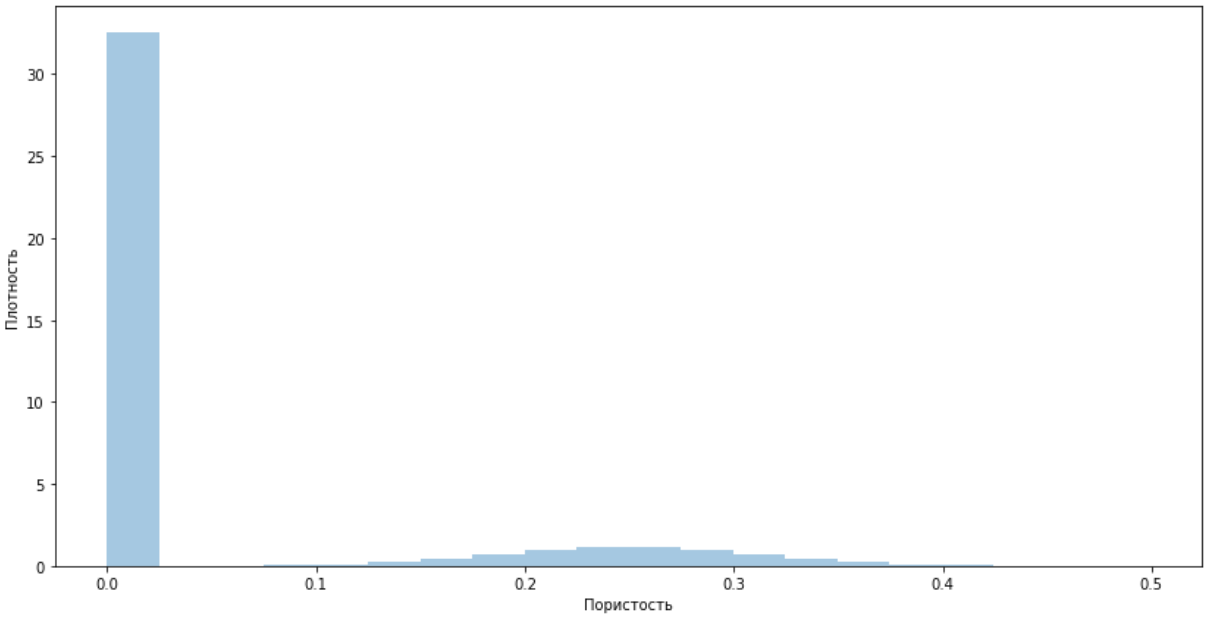


Рисунок 5 – Гистограмма распределения пористости в датасете

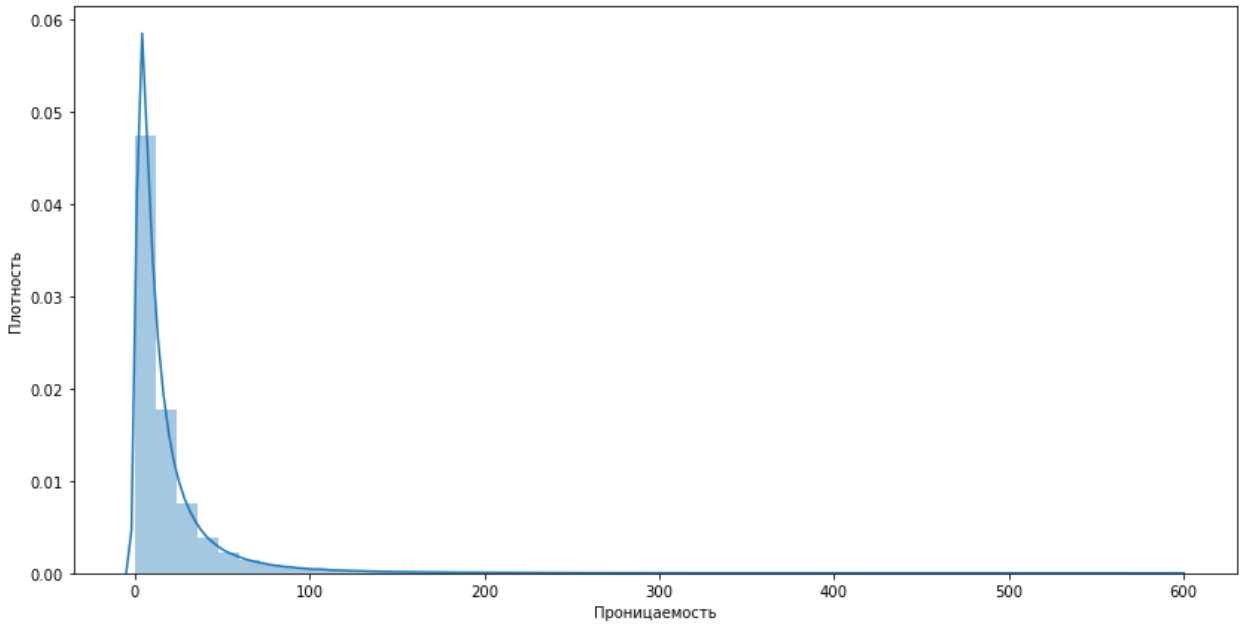


Рисунок 6 – Распределение проницаемости в активных ячейках

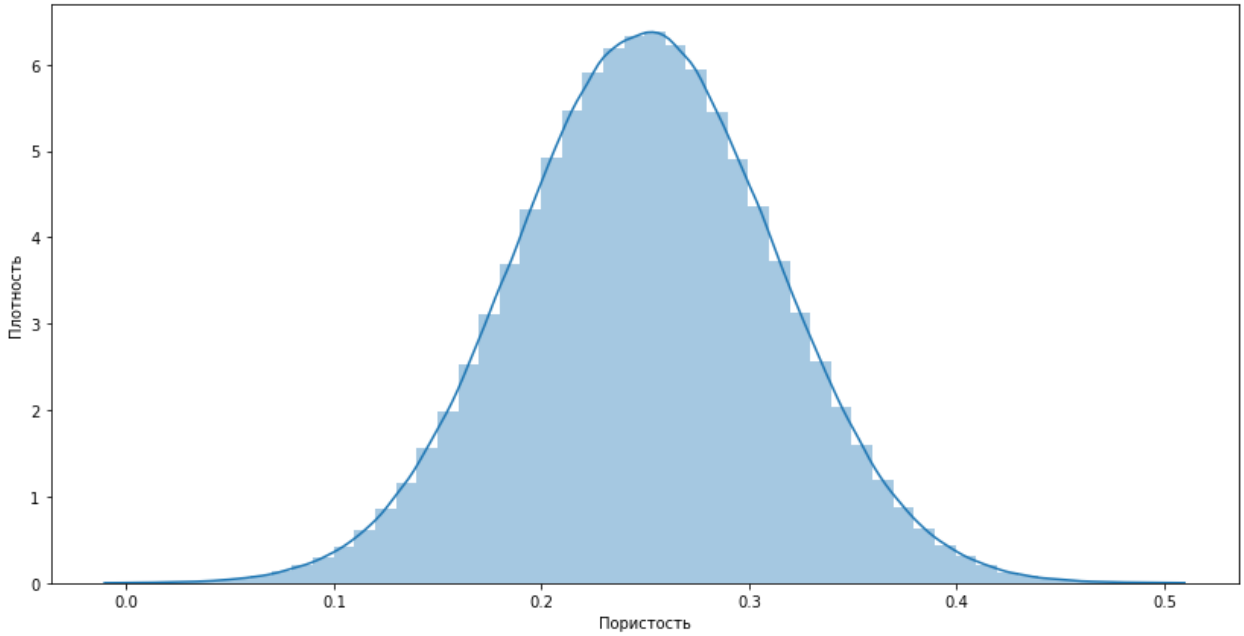


Рисунок 7 – Распределение пористости в активных ячейках

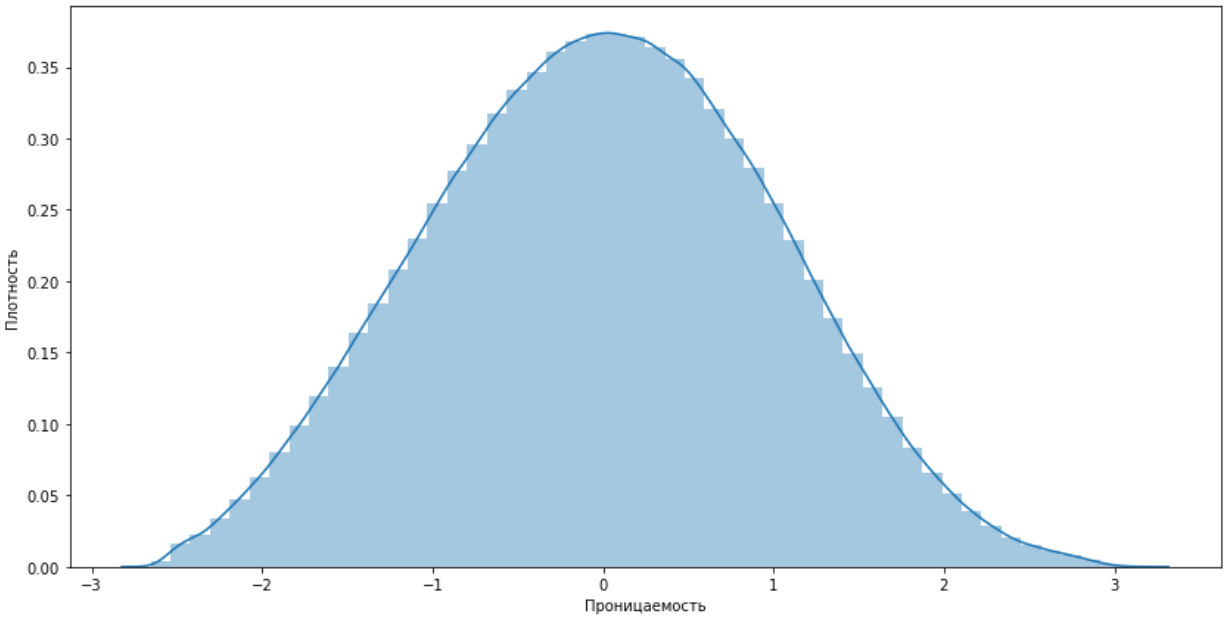


Рисунок 8 – Распределение проницаемости активных ячеек после преобразования

3 Машинное обучение и используемые модели

3.1 Генеративная сеть

3.1.1 Графовые свёртки

Для работы с графовыми данными не подходят обычные свёрточные операции – они работают лишь с данными прямоугольной формы. Графовые свёрточные слои позволяют вычислять активации нейронов сети на основе матрицы свойств вершин, а также списка рёбер, соединяющих данные вершины. В данной работе проводились эксперименты с двумя видами графовых свёрток – GraphConv [22] и GCNConv [16], т.к. они наиболее часто применяются в смежных задачах классификации вершин и восстановления рёбер графа. Слои GraphConv и GCNConv описаны в (6) и (7), соответственно.

$$x_i' = \theta_1 x_i + \theta_2 \sum_{j \in N(i)} e_{j,i} \cdot x_j \,, \tag{6}$$

где x_i – активации предыдущего слоя, x_i' – активации следующего слоя, $N(i)$ – множество вершин, соседних с i, $e_{j,i}$ – вес ребра (j, i), θ_1 и θ_2 – полносвязные слои.

$$x_i' = \theta^T \sum_{j \in N(i) \cup \{i\}} \frac{e_{j,i}}{\sqrt{\widehat{d}_j \widehat{d}_i}} x_j \,, \tag{7}$$

где $\widehat{d}_i = 1 + \sum_{j \in N(i)} e_{j,i}$, остальная нотация аналогична GraphConv. Веса всех рёбер в данной работе равны 1, т.к. рёбра не отличаются друг от друга.

3.1.2 Автокодировщики

Автокодировщик (англ. autoencoder, автоэнкодер) – архитектура нейросетей, основной принцип которой – получить на выходном слое отклик, максимально повторяющий входные данные. Эта особенность позволяет автоэнкодерам обучаться на неразмеченных данных (обучение без учителя). Впервые такая архитектура была использована в работе Дэвида Румельхарта и др. [29], в которой использовалась сеть прямого распространения, содержащая три слоя: входной, промежуточный и выходной. При этом для того, чтобы

полученная сеть была полезной, на промежуточный слой накладывается дополнительное ограничение – он должен быть меньше, чем входной и выходной слои. За счёт такого ограничения сеть обучается скрытым зависимостям во входных данных, которые зашифровываются в матрицах весов сети-автокодировщика, и сжимает информацию о датасете в более компактное представление.

Формально, автокодировщик состоит из двух частей: кодировщика g и декодировщика f. Энкодер преобразует входные данные в вектор $z = g(x)$, который можно трактовать как точку в скрытом пространстве кодировщика. Декодер, в свою очередь, восстанавливает данные из скрытого пространства обратно в пространство признаков $x = f(z)$.

Вариационные автокодировщики учат генеративную функцию $p_\theta(x, z)$ определяя вероятность скрытых кодов $p_\theta(\mathrm{x}|\mathrm{z})$ и априорное распределение скрытых кодов $p(z)$. Они обучаются с помощью максимизации особой метрики Evidence Lower Bound (ELBO), которая состоит из двух частей – ошибки реконструкции и расхождения Кульбака-Лейблера (D_z)

Архитектуры полученных сетей кодировщика и декодировщика представлены на рисунках 9 и 10.

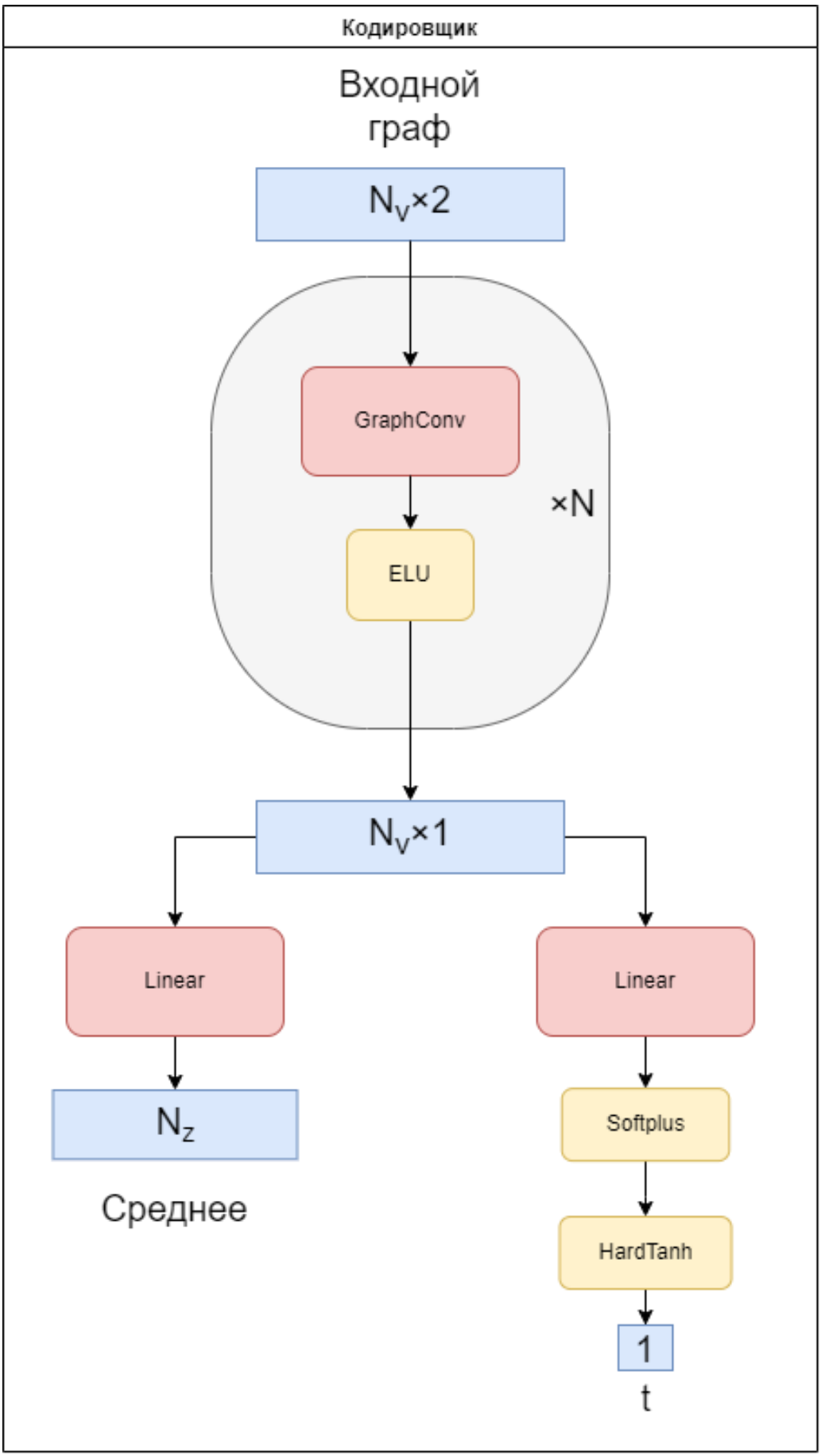


Рисунок 9 – Архитектура кодировщика

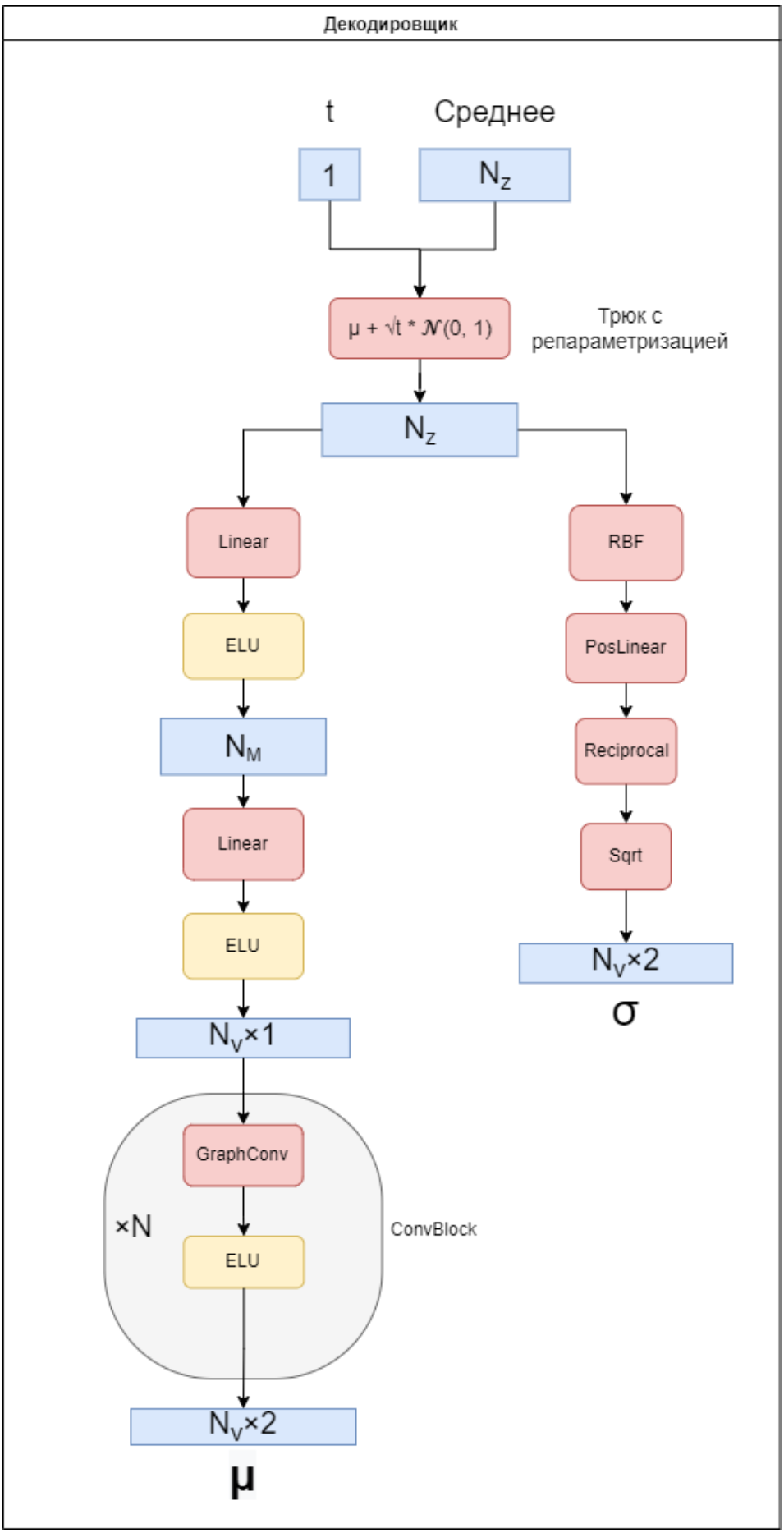


Рисунок 10 – Архитектура декодировщика

3.2 Метрики и функции потерь

3.2.1 Среднеквадратичное отклонение (англ. Mean Squared Error, MSE)

Значение ELBO сложно использовать для оценки качества восстановленных гридов из-за слагаемого D_{KL}. Кроме того, на качество генерируемых карт гораздо больше влияет ошибка восстановления именно активных ячеек. Поэтому для обучения и оценки моделей использовались метрики AMSE и WMSE. Оригинальная функция MSE выглядит следующим образом:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\,, \tag{8}$$

где N – количество элементов, $y, \hat{y}$ – исходные и восстановленные свойства, соответственно.

WMSE (Weighted Mean Squared Error) – взвешенное среднеквадратичное отклонение, рассчитываемое по формуле:

$$WMSE = \alpha MSE(X_{active}) + \beta MSE(X_{non-active}), \tag{9}$$

где X_{active}, $X_{non-active}$ – множество свойств активных и неактивных ячеек, соответственно. α и β – параметры, регулирующие отношение важности ошибки.

Таким образом, при $\alpha > \beta$ нейросеть должна лучше восстанавливать активные ячейки грида.

AMSE (Active Mean Squared Error) – метрика качества сети, показывающая среднеквадратичную ошибку лишь для активных ячеек:

$$AMSE = MSE(X_{active}), \tag{10}$$

3.2.2 Коэффициент Жаккара (англ. Intersection over Union, IoU)

Кроме минимизации AMSE для улучшения качества гридов также необходимо, чтобы сеть-автокодировщик точно воспроизводила

распределения свойств активных ячеек всего грида. Для более легкого сравнения обученных сетей было предложено использовать вариацию коэффициента Жаккара [11], подсчитанного по гистограммам распределений свойств. Метрика вычисляется по отдельности для каждого из свойств следующим образом:

1. Промежуток $[0; 1]$ делится на 100 равных отрезков, для каждого отрезка i вычисляется значение c_i – количество ячеек, свойства которых попадает в отрезок i.
2. Данная процедура повторяется для восстановленного грида, чтобы получить c_i' аналогично c_i.
3. Конечная метрика вычисляется по формуле:

$$IoU = \frac{\sum_i^N \min(c_i, c_i')}{\sum_i^N \max(c_i, c_i')}, \tag{11}$$

где N – общее количество сравниваемых ячеек.

Значения IoU для пористости и проницаемости складываются, чтобы получить общую метрику IoU_S. Можно заметить, что $0 \leq IoU_S \leq 2$. Пример гистограмм, которые сравнивает предложенная метрика можно увидеть на рисунке 11.

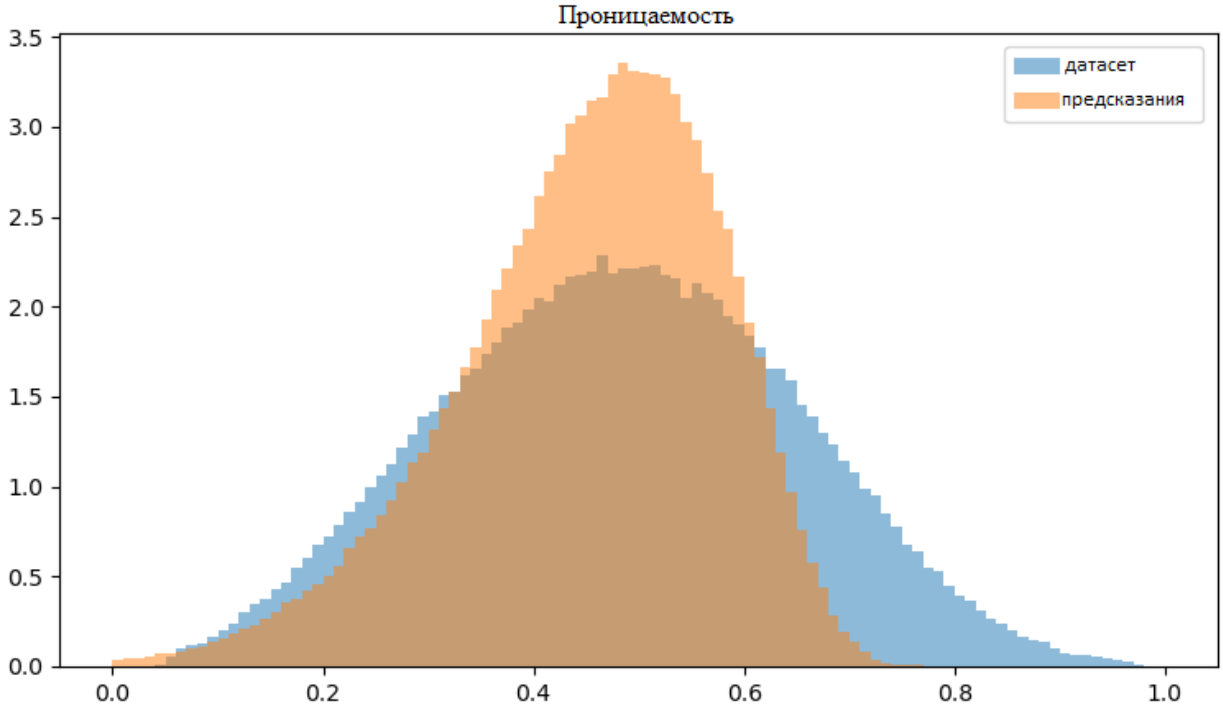


Рисунок 11 – Гистограммы проницаемости исходных и восстановленных гридов

3.3 Оценка плотности пространства автокодировщика

В данной работе дан лишь краткий обзор методики, используемой для оценки плотности скрытого пространства автоэнкодера более подробное описание дано в работах Г. Арванитидиса [1] и Д. Калатциса [13].

3.3.1 Риманова геометрия

Стандартное предположение о том, что априорное распределение декодировщика является гауссовым основывается на мере Лебега [19], которая, в свою очередь, предполагает, что скрытое пространство подчиняется законом евклидовой геометрии. Исследования показали, что такое предположение не всегда верно [2] и использование римановой [28] геометрии даёт больше информации о скрытых кодах промежуточного слоя автокодировщика. Для определения плотности скрытого пространства предлагается [3] использовать следующую метрику:

$$G_z = J_\mu(z)^T J_\mu(z) + J_\sigma(z)^T J_\sigma(z), \quad (12)$$

где μ – стандартный вариационный декодировщик, а σ – дополнительная ветвь декодировщика, вычисляющая следующую меру, оценивающую, насколько плотно скрытое пространство в данной точке:

$$Precision(z) = \frac{1}{\sqrt{PosLinear(RBF(z))}}, \quad (13)$$

Слой PosLinear – особая вариация полносвязного слоя:

$$PosLinear(x) = \log\left(1 + e^{\theta}\right) x, \quad (14)$$

где θ – матрица весов слоя.

Слой RBF использует расстояние между вектором скрытого пространства и скрытых векторов из обучающего датасета:

$$\mathrm{RBF}(z) = \max\left(\sum z^2 + \sum z_d^2 - 2 * zz_d\,, 0\right), \quad (15)$$

где z_d – матрица значений скрытых кодов гридов из обучающего датасета.

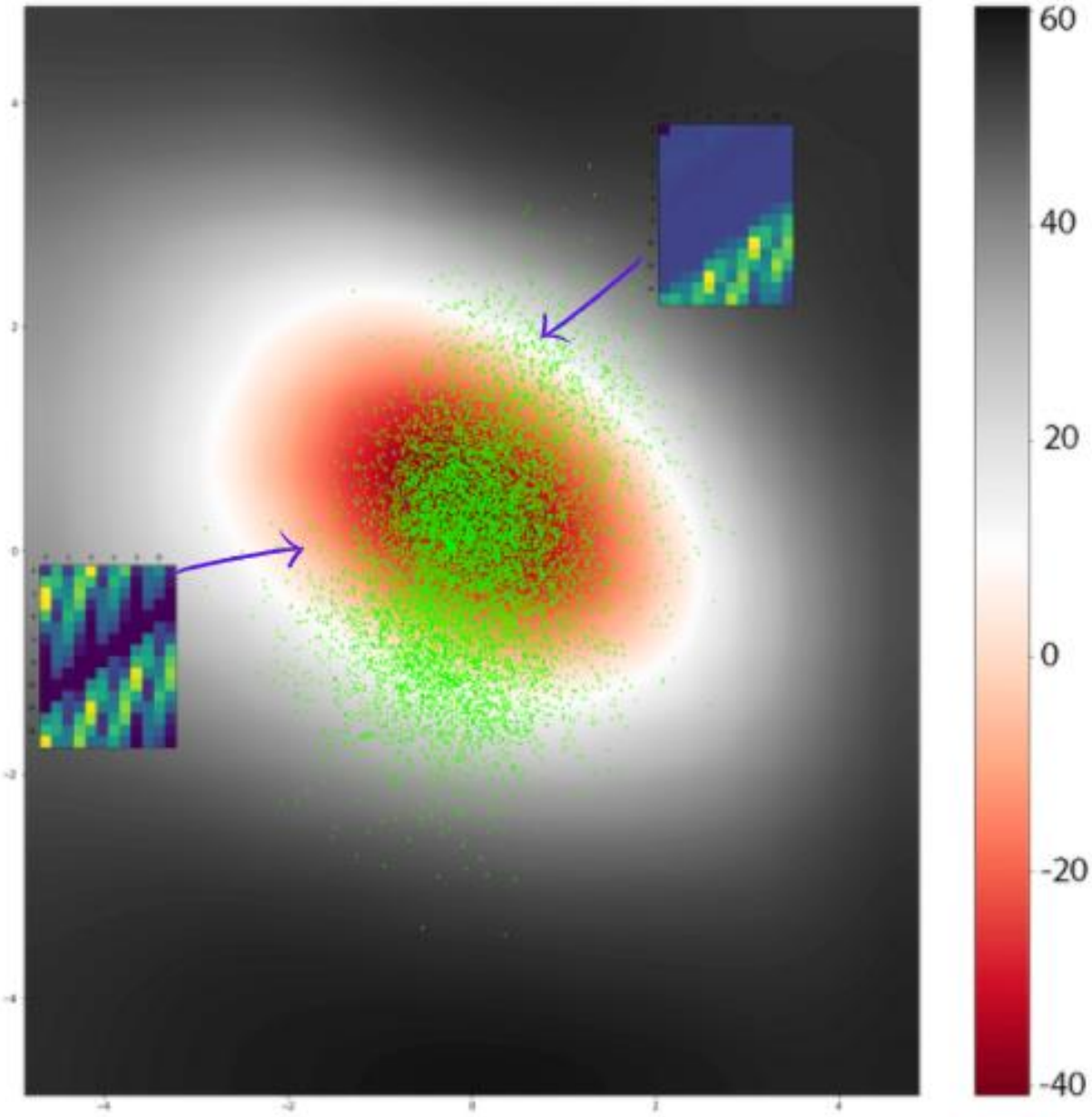


Рисунок 12 – Двумерная проекция скрытого пространства автокодировщика

На рисунке 12 визуализирована метрика скрытого пространства обученного автокодировщика. Области, где метрика низкая (обозначены красным) соответствуют более качественным гридам, чем те, где метрика высокая (обозначены белым и черным).

Для подсчета метрики G_z необходим эффективный метод подсчета Якобианов всех слоев нейронной сети. Для слоя GraphConv подсчет производной пришлось реализовывать вручную:

$$\frac{\partial x_i'}{\partial x_i} = \theta_1, \tag{16}$$

$$\frac{\partial x'_j}{\partial x_i} = \theta_2, \tag{17}$$

```
class GraphConv:

    def _jacobian(self):
        J_analytical = torch.zeros(self.num_vertices, self.out_channels,
self.num_vertices,
                                   self.in_channels, device=torch.device('cuda'))
        idx = torch.arange(self.num_vertices, device=torch.device('cuda'))

        # значения на диагонали проходят через обычный полносвязный слой,
        # поэтому их производная равна весу этого слоя
        J_analytical[idx, :, idx, :] = self.lin_r.weight

        J_analytical[self.edge_index[0], :, self.edge_index[1],
        :] = self.lin_l.weight

        J = torch.reshape(J_analytical,
                          (1, self.num_vertices * self.out_channels,
                           self.num_vertices * self.in_channels))
        return J, JacType.FULL


    def _jac_mul(self, Jseq, JseqType):
        """
        :Jseq: значение Якобиана предыдущего слоя
        :JseqType: тип Якобиана предыдущего слоя
        """
        J, _ = self._jacobian(x, val)  # (out)x(in) -- размер Якобиана функции
        J = J[0]
        # Нужно совершить матричное умножение текущего Якобиана на:
        #  1) вектор размера (B)x(in)
        #       Такой вектор считается диагональной матрицей размера
        #       (B)x(in)x(in), и нужно посчитать J * diag(Jseq)
        #  2) матрицу размера (B)x(in)x(M)
        #       В таком случае нужно посчитать J * Jseq
        if JseqType is JacType.FULL:
            Jseq = torch.einsum("oi,bim->bom", J, Jseq)  # J * Jseq: (out)x(in) *
(B)x(in)x(M)-> (B)x(out)x(M)
        elif JseqType is JacType.DIAG:
            Jseq = torch.einsum("oi,bi->boi", J, Jseq)  # J * diag(Jseq): (out)x(in) *
(B)x(in) -> (B)x(out)x(in)
        return Jseq, JacType.FULL
```

Листинг 1 – Класс для подсчета Якобиана GraphConv

3.4 CMA–ES

3.4.1 Эволюционные стратегии

Эволюционные стратегии – семейство алгоритмов оптимизации, не требующие градиента функции и основанные на симуляции некоторых правил

на наборе особей. Каждая особь – предполагаемой решение задачи (в данном случае – скрытый код некоторого грида). На каждой итерации эволюционной стратегии для каждой особи высчитывается метрика. На основе этих значений определяется новое поколение особей, которое должно улучшить результат.

Эволюционная стратегия с адаптацией матрицы ковариации (англ. Covariance Matrix Adaptation Evolution Strategy, CMA-ES) – алгоритм из семейства эволюционных стратегий, разработанный для решения сложных, нелинейных, невыпуклых задач типа «черный ящик» в непрерывном пространстве поиска. Особенность CMA-ES заключается в том, что на каждом поколении происходит перерасчет матрицы ковариации, которая непосредственно влияет на размер и форму области, где будут генерироваться будущие потенциальные решения. Этот метод был выбран среди других, так как он показывает наилучшие результаты на множестве публичных наборов данных [20].

Метрика, которая оптимизируется эволюционной стратегией в данной работе состоит из трех слагаемых:

1. среднеквадратичное отклонение свойств пористости и проницаемости в известных ячейках (скважинах).
2. метрика R^2, вычисляемая по отклонениям дебитов воды и нефти от известных:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)}, \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i\,, \qquad (18)$$

где y, $\hat{y}$ – исходные и предсказанные, значения дебитов соответственно.

3. метрика G_z, разделенная на -100 для нормализации.

4 Эксперименты

4.1 Сравнение графовых свёрток

Для определения наиболее подходящего графового слоя были проведены эксперименты, сравнивающие автокодировщики, построенные с использованием разных графовых свёрток – GraphConv и GCNConv. Сети обучались в течение 50 эпох, оценивались метрики AMSE и ELBO. На рисунках 13, 14 видно, что сеть с GCNConv уступает аналогичной с GraphConv.

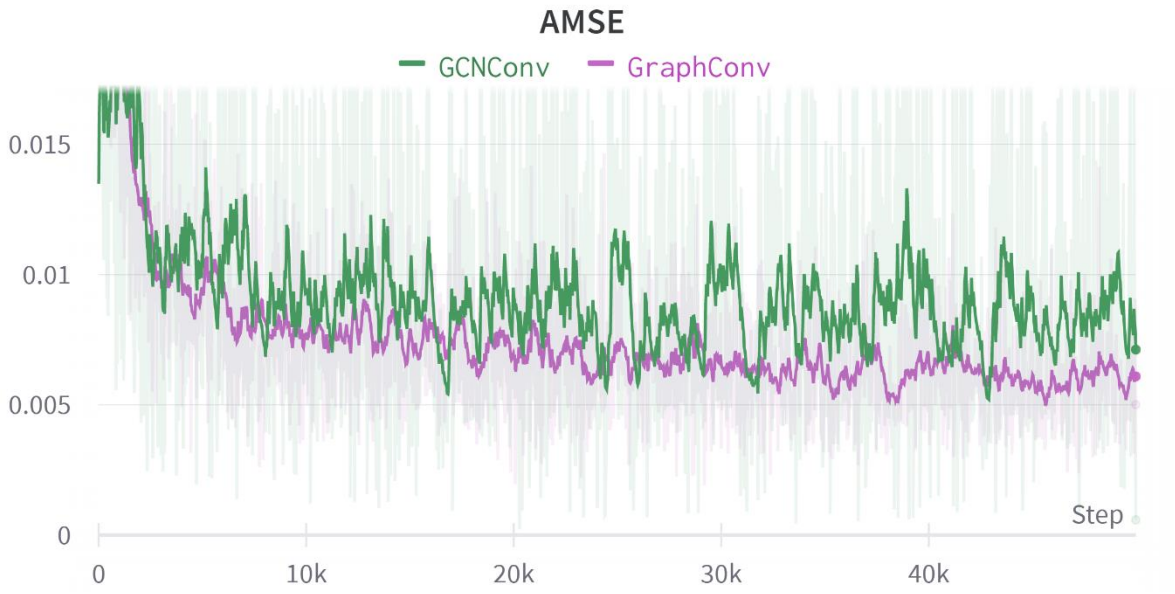


Рисунок 13 – Метрика AMSE для сетей с GCNConv и GraphConv

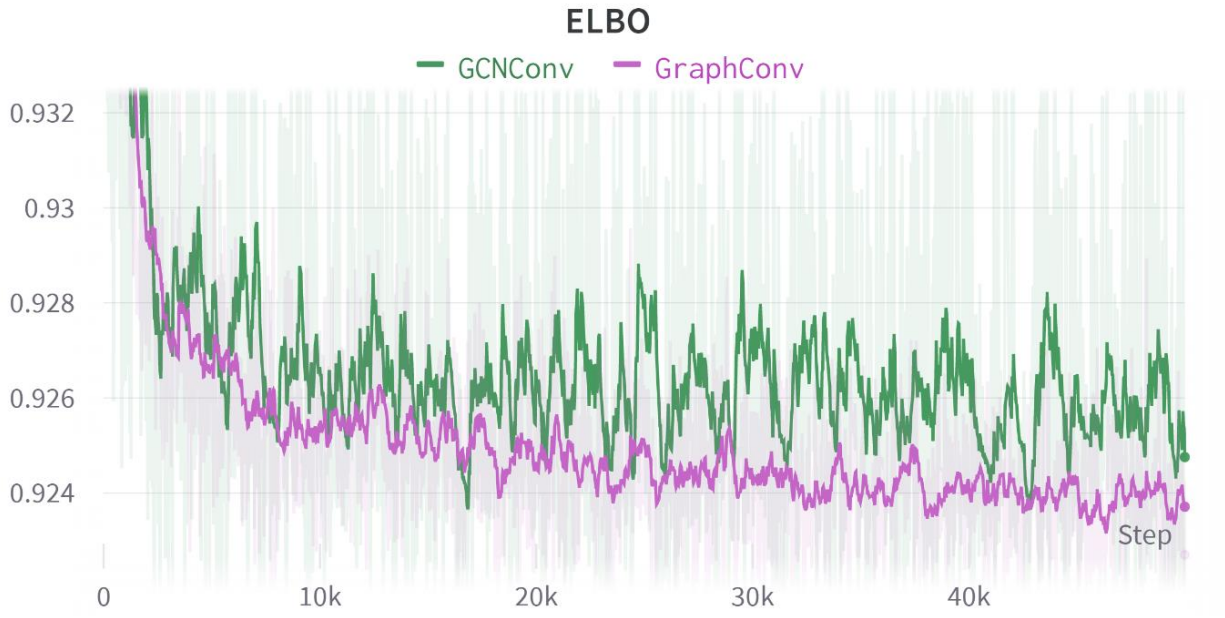


Рисунок 14 – Метрика ELBO для сетей с GCNConv и GraphConv

Это объясняется тем, что в GCNConv отсутствует дополнительный слой, подобный θ_1 из (6), не использующий агрегацию по соседним вершинам. Из-за этого сеть с GCNConv не способна воспроизводить высокие значения свойств (Рисунок 15).

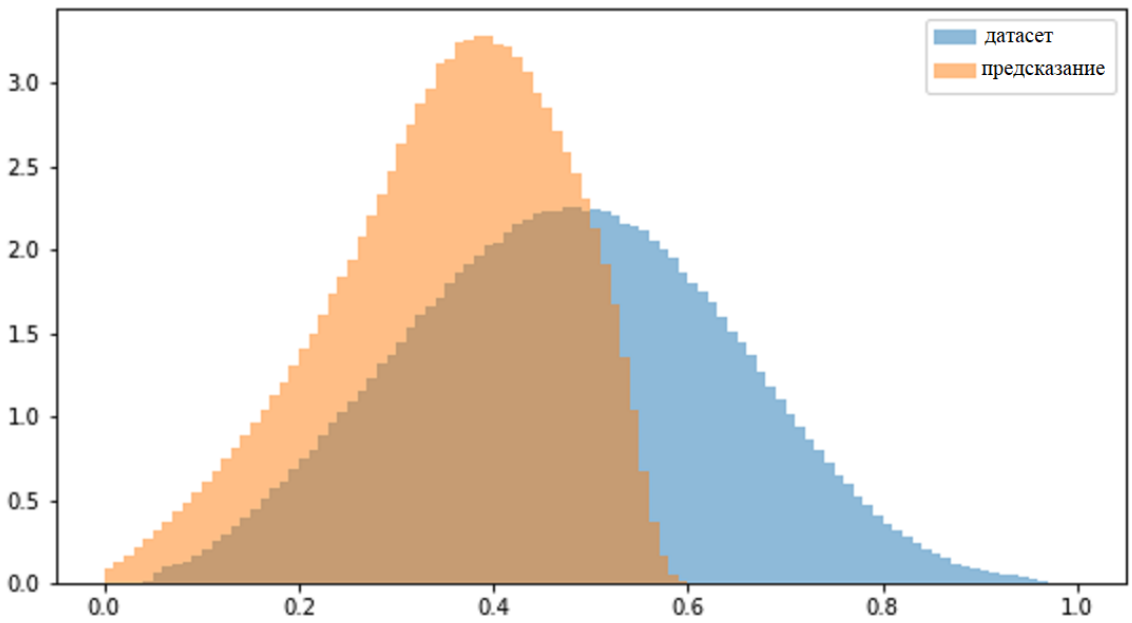


Рисунок 15 – Распределение проницаемости, восстановленной с помощью GCNConv

4.2 Влияние WMSE на качество восстановления гридов

Для определения влияния использования функции потерь WMSE на качество восстановления активных ячеек был проведён эксперимент, в котором сравнивались сети, обучаемый с использованием WMSE и без него. Сети обучались в течение десяти эпох. Общая функция потерь в первом случае получалась сложением ELBO и WMSE. Параметры α и β в (9) равнялись 10 и 1 соответственно. На рисунке 16 изображен график изменения метрики AMSE в ходе обучения моделей. Эксперимент показал, что использование WMSE уменьшает ошибку в восстановлении активных ячеек более чем в 2 раза.

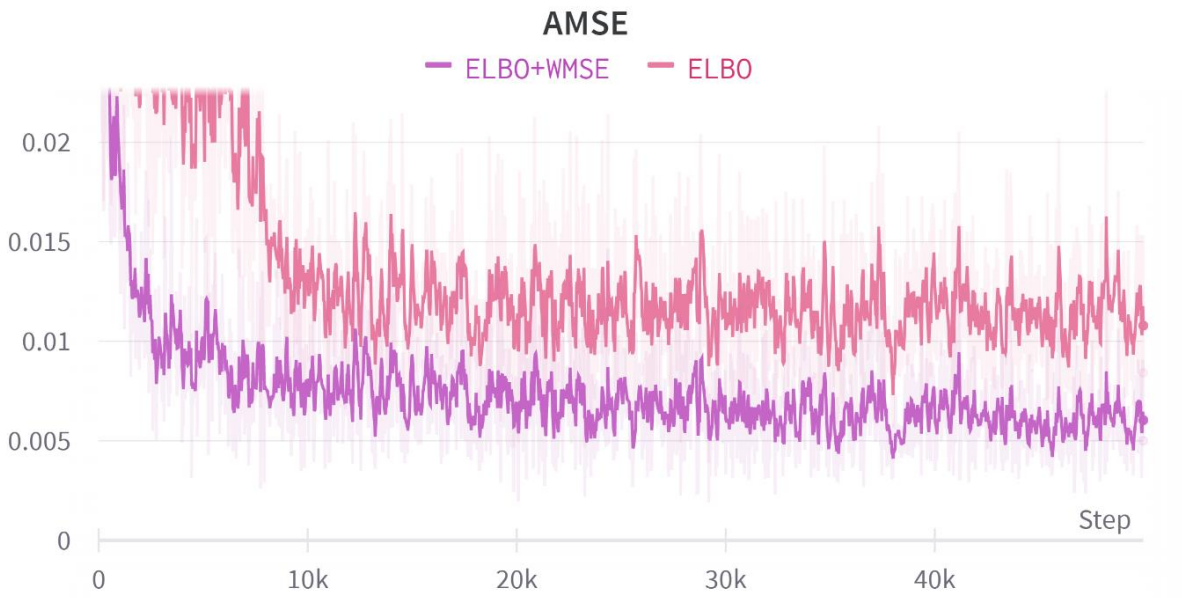


Рисунок 16 – Влияние WMSE на метрику AMSE

4.3 Методика обучения

Для обучения всех нейросетей в качестве оптимизатора использовался оптимизатор Adam [14].

Все вычисления производились на персональном компьютере со следующими характеристиками:

- процессор: Intel® Core™ i7–8700K CPU @ 3.70GHz,
- графический процессор: NVIDIA GeForce RTX 3060 12Gb VRAM,
- оперативная память: 24Gb DDR4.

4.4 Используемые инструменты

4.4.1 Машинное обучение

Скрипты для обучения сетей в данной работе написаны на языке программирования Python. Для проведения экспериментов использовалась среда разработки Jupyter Notebook, позволяющая выполнять отдельные сегменты кода (клетки).

Сеть–автокодировщик была создана с использованием PyTorch, для реализации графовых слоёв использовалась библиотека PyTorch Geometric [9]. Реализация CMA–ES взята из библиотеки pycma.

Для визуализации данных применялись библиотеки matplotlib и seaborn.

4.4.2 tNavigator

В ходе адаптации геологической модели необходимо было рассчитывать гидродинамические свойства сгенерированных гридов. Для этой задачи использовалась проприетарная программная среда tNavigator [33]. Взаимодействие алгоритмов с tNavigator осуществлялось с помощью консольной утилиты. Основной интерфейс программы представлен на рисунке 17.

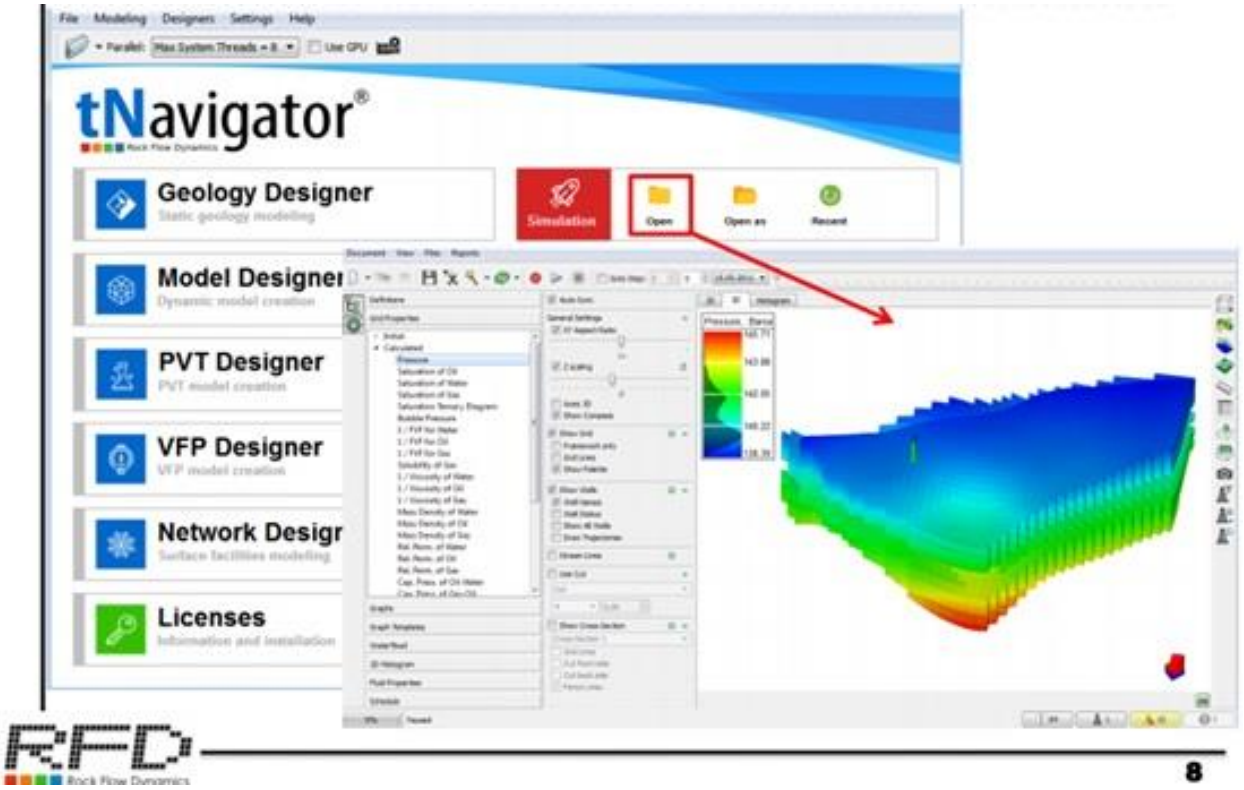


Рисунок 17 – Интерфейс tNavigator

5 Результаты

Обученная сеть-автокодировщик была протестирована на контрольной выборке из датасета, её показатели приведены в таблице 1. Сравнение исходного и генерируемого распределения пористости показано на рисунке 18.

Таблица 1 – Значения метрик лучшей обученной модели

Метрика	**Результат**
AMSE	0.005
MSE	0.01
ELBO	0.9227
IoU_S	1.5

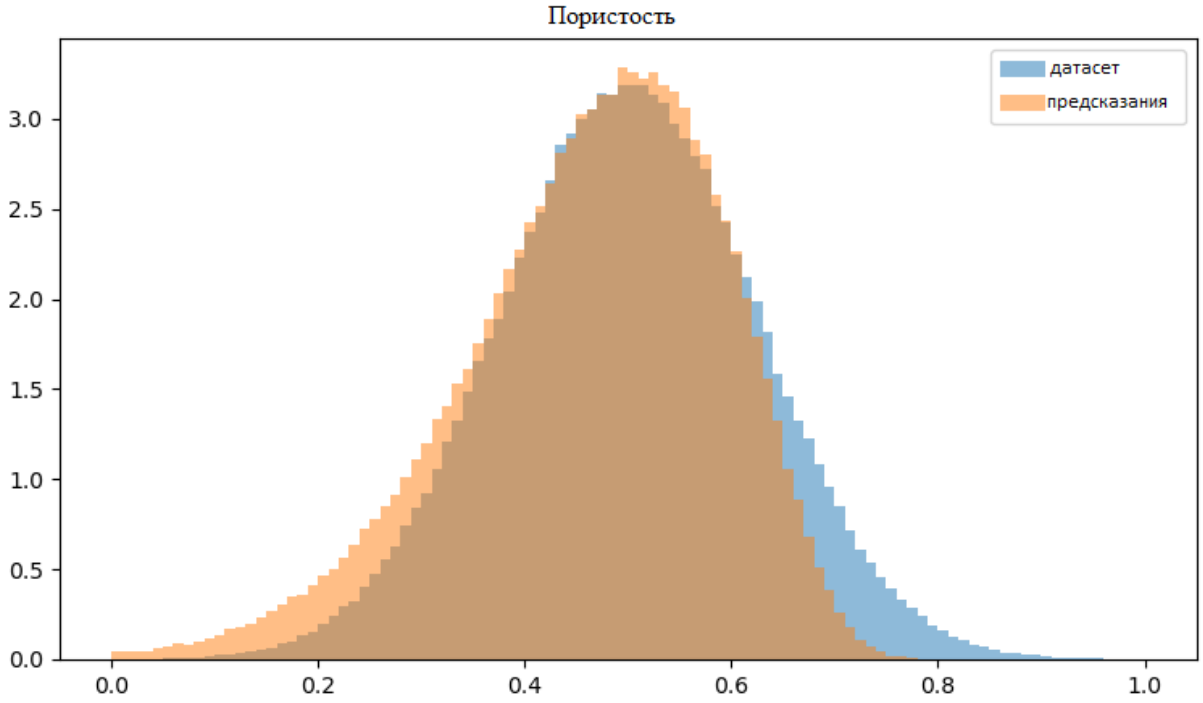


Рисунок 18 – Гистограммы пористости ячеек из датасета и восстановленных гридов

Обучение можно считать успешным, т.к. нейросеть с точностью до 10^{-2} восстанавливает все ячейки гридов, а активные ячейки восстанавливаются с точностью в два раза больше.

В ходе дальнейших экспериментов по адаптации геологической модели с использованием алгоритма CMAES были также получены удовлетворительные результаты, что позволяет сделать вывод об эффективности представленных методов. Финальные значения метрик приведены в таблице 2. Стоит заметить, что метрика MSE считалась после преобразования свойств пористости и проницаемости обратно в их исходные распределения, поэтому значение среднеквадратичной ошибки выше, чем в Таблице 1.

Таблица 2 – Значения метрик АГМ

Метрика	**Результат**
MSE	0.1465
R^2	0.00982
G_z	34.83

ЗАКЛЮЧЕНИЕ

В данной работе был представлен новый набор алгоритмов, состоящий из сети-автокодировщика и оптимизатора на основе CMAES, который может в дальнейшем быть использован для решения задачи адаптации геологических моделей. Таким образом, выполнены все поставленные задачи и успешно достигнута цель работы.

Графовые архитектуры нейронных сетей могут стать новым стандартом в сфере генерации геологических карт, т.к. они позволяют работать с месторождениями любой формы и геометрии.

Оценка плотности скрытого пространства высокой размерности позволяет гарантировать генерацию только корректных гридов, что является критичной проблемой других методов.

Часть результатов работы опубликована в научной статье [31] для конференции International Petroleum Technology Conference.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Arvanitidis G. et al. Geometrical Aspects of Manifold Learning. – 2019.

2. Arvanitidis G., Hansen L. K., Hauberg S. A locally adaptive normal distribution //Advances in Neural Information Processing Systems. – 2016. – Т. 29.

3. Arvanitidis G., Hansen L. K., Hauberg S. Latent space oddity: on the curvature of deep generative models //arXiv preprint arXiv:1710.11379. – 2017.

4. Box G. E. P., Cox D. R. An analysis of transformations //Journal of the Royal Statistical Society: Series B (Methodological). – 1964. – Т. 26. – №. 2. – С. 211–243.

5. Chan S., Elsheikh A. H. Parametric generation of conditional geological realizations using generative neural networks //Computational Geosciences. – 2019. – Т. 23. – №. 5. – С. 925-952.

6. Chan S., Elsheikh A. H. Parametrization and generation of geological models with generative adversarial networks //arXiv preprint arXiv:1708.01810. – 2017.

7. Chen W. H. et al. A new algorithm for automatic history matching //Society of Petroleum Engineers Journal. – 1974. – Т. 14. – №. 06. – С. 593–608.

8. Darcy H. Les fontaines publiques de la ville de Dijon: exposition et application.. – Victor Dalmont, 1856.

9. Fey M., Lenssen J. E. Fast graph representation learning with PyTorch Geometric //arXiv preprint arXiv:1903.02428. – 2019.

10. Fukushima K., Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition //Competition and cooperation in neural nets. – Springer, Berlin, Heidelberg, 1982. – С. 267-285.

11. Gilbert G. K. Finley's tornado predictions //American Meteorological Journal. A Monthly Review of Meteorology and Allied Branches of Study (1884-1896). – 1884. – Т. 1. – №. 5. – С. 166.

12. Goodfellow I. et al. Generative adversarial nets //Advances in neural information processing systems. – 2014. – Т. 27.

13. Kalatzis D. et al. Variational autoencoders with riemannian brownian motion priors //arXiv preprint arXiv:2002.05227. – 2020.

14. Kingma D. P., Ba J. Adam: A method for stochastic optimization //arXiv preprint arXiv:1412.6980. – 2014.

15. Kingma D. P., Welling M. Auto–encoding variational bayes //arXiv preprint arXiv:1312.6114. – 2013.

16. Kipf T. N., Welling M. Semi–supervised classification with graph convolutional networks //arXiv preprint arXiv:1609.02907. – 2016.

17. Laloy E. et al. Inversion using a new low–dimensional representation of complex binary geological media based on a deep neural network //Advances in water resources. – 2017. – Т. 110. – С. 387-405.

18. Lanzarini W. L. et al. Stochastic modeling of geometric objects and reservoir heterogeneities //Latin American and Caribbean Petroleum Engineering Conference. – OnePetro, 1997.

19. Lebesgue H. Intégrale, longueur, aire //Annali di Matematica Pura ed Applicata (1898-1922). – 1902. – Т. 7. – №. 1. – С. 231-359.

20. Loshchilov I., Schoenauer M., Sebag M. Bi-population CMA-ES agorithms with surrogate models and line searches //Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. – 2013. – С. 1177-1184.

21. McCallum A. K. et al. Automating the construction of internet portals with machine learning //Information Retrieval. – 2000. – Т. 3. – №. 2. – С. 127–163.

22. Morris C. et al. Weisfeiler and leman go neural: Higher-order graph neural networks //Proceedings of the AAAI conference on artificial intelligence. – 2019. – Т. 33. – №. 01. – С. 4602-4609.

23. Mosser L., Dubrule O., Blunt M. J. Deepflow: history matching in the space of deep generative models //arXiv preprint arXiv:1905.05749. – 2019.

24. Mosser L., Dubrule O., Blunt M. J. Reconstruction of three-dimensional porous media using generative adversarial neural networks //Physical Review E. – 2017. – Т. 96. – №. 4. – С. 043309.

25. Mosser, L., Dubrule, O., & Blunt, M. J. (2018). Conditioning of three–dimensional generative adversarial networks for pore and reservoir–scale models. arXiv preprint arXiv:1802.05622.

26. Petrel [Электронный ресурс] //software.slb.ru [сайт]. [2022]. URL: https://software.slb.ru/products/petrel/ (дата обращения: 08.05.2022).

27. Potratz J. et al. Large Dimension Parameterization with Convolutional Variational Autoencoder: An Application in the History Matching of Channelized Geological Facies Models //2020 20th International Conference on Computational Science and Its Applications (ICCSA). – IEEE, 2020. – С. 23-32.

28. Riemann B. The hypotheses on which geometry is based. – 1854.

29. Rumelhart D. E., Hinton G. E., Williams R. J. Learning internal representations by error propagation. – California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

30. Sen P. et al. Collective classification in network data //AI magazine. – 2008. – Т. 29. – №. 3. – С. 93-93.

31. Shishaev G. et al. History Matching and Uncertainty Quantification of Reservoir Performance with Generative Deep Learning and Graph Convolutions //International Petroleum Technology Conference. – OnePetro, 2022.

32. Sperduti A., Starita A. Supervised neural networks for the classification of structures //IEEE Transactions on Neural Networks. – 1997. – Т. 8. – №. 3. – С. 714–735.

33. tNavigator [Электронный ресурс] //rfdyn.ru [сайт]. [2022]. URL: https://rfdyn.ru/integrated–modeling/gidrodinamicheskoe–modelirovanie/ (дата обращения: 08.05.2022).

34. Wahl W. L. et al. Matching the performance of saudi arabian oil fields with an electrical model //Journal of Petroleum Technology. – 1962. – Т. 14. – №. 11. – С. 1275–1282.

35. Yeo I. K., Johnson R. A. A new family of power transformations to improve normality or symmetry //Biometrika. – 2000. – Т. 87. – №. 4. – С. 954–959.

36. Карпов Д. В. Теория графов //СПб.: Санкт-Петербургское отделение Мат. института им. ВА Стеклова РАН. – 2017. С.16

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Змеев О.А.
10.06.2022

Автор: Выгон Роман
Проверяющий: Выгон Роман

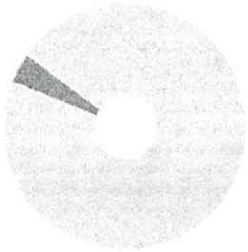
Отчет предоставлен сервисом «Антиплагиат» - http://users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 4
Начало загрузки: 04.06.2022 18:37:26
Длительность загрузки: 00:00:01
Имя исходного файла: _____.pdf
Название документа: _____
Размер текста: 39 кБ
Символов в тексте: 40117
Слов в тексте: 4520
Число предложений: 387

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 04.06.2022 18:37:28
Длительность проверки: 00:00:02
Комментарии: не указано
Модули поиска: Интернет Free



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
4,33%	0%	0%	95,67%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,23%	DeepFlow: History Matching in the Space of Deep Generative Models http://arxiv.org	19 Мар 2020	Интернет Free
[02]	0,96%	Towards a Robust Parameterization for Conditioning Facies Models Using Deep Variational Autoencoders and Ensemble Smoother http://arxiv.org	19 Окт 2019	Интернет Free
[03]	0,8%	Understanding Deep Learning Techniques for Image Segmentation http://arxiv.org	19 Мар 2020	Интернет Free

Еще источников: 4
Еще заимствований: 1,34%