

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра информатики и программирования

НЕЙРОННЫЕ СЕТИ С НЕОБУЧАЕМЫМИ СЛУЧАЙНЫМИ СЛОЯМИ
АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 02.04.03 — Математическое обеспечение и администрирование
информационных систем
факультета КНиИТ
Ткачева Владимира Владимировича

Научный руководитель
профессор, д. ф.-м. н., доцент _____

П. В. Купцов

Заведующий кафедрой
доцент, к. ф.-м. н. _____

М. В. Огнева

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. При решении самых разных вычислительных задач в настоящее время применяют модели машинного обучения. Используя накопленные данные, модели машинного обучения помогают анализировать сложные процессы, обработка которых другими средствами затрудняется неточностью, нестабильностью результатов.

Среди всех существующих моделей машинного обучения выделяется модель нейронной сети. Нейросети, поэтапно конструируя разные представления входных данных, показывают высокую вычислительную способность и находят применение в задачах, которые ранее решались только вручную человеком: обработка текста, распознавание образов, предсказание будущих показателей временных рядов и другие.

Сложные нейронные сети, содержащие десятки слоев из тысячи и более нейронов, требуют значительных ресурсов от вычислительных систем в процессе обучения для подготовки своих внутренних параметров и, впоследствии, на этапе эксплуатации. Передовые языковые модели задействуют для своей работы связки из множества вычислительных узлов с дорогостоящими аппаратными средствами. Такие системы используют количество энергии, не сравнимое с потреблением, к примеру, умных часов, и, тем более, человеческого мозга.

В 2000-х годах начало развиваться направление резервуарных вычислений. Исследования по данному направлению базируются на идее создания модели искусственного интеллекта, эффективной по затрачиваемым времени и энергии, а также быстро адаптируемой к разным условиям.

Метод резервуарных вычислений основывается на использовании «резервуара» — некой нелинейной динамической системы, видимое поведение которой характеризуется непредсказуемостью и случайностью. Путем взаимодействия входных данных задачи с данной системой получают показатели, на которых производится дальнейшая подстройка параметров модели машинного обучения с простой структурой. Такая система из неизменяемого «резервуара» и адаптируемой к конкретной задаче модели машинного обучения требует значительно меньших затрат в применении по сравнению с большими нейросетями.

Цель магистерской работы — на примере генеративных языковых моделей построить и исследовать работу нейронных сетей, в которых необучаемые случайные слои (резервуарные сети) дополнены традиционными слоями глу-

бокого обучения, выполнить сравнительный анализ различных конфигураций, сделать выводы о качестве работы полученных гибридных сетей.

Поставленная цель определила **следующие задачи**:

1. Проанализировать современные технологии построения нейронных сетей.
2. Познакомиться с методами резервуарных вычислений.
3. Изучить основные способы выбора гиперпараметров.
4. Разобрать программные инструменты для создания нейронных сетей.
5. Рассмотреть программные инструменты для подбора гиперпараметров моделей и мониторинга процесса обучения.
6. Выполнить сбор и подготовку текстовых данных на русском языке для обучения языковой модели.
7. Реализовать на практике модели, использующие необучаемые случайные слои, и проанализировать результаты.

Методологические основы нейронных сетей с необучаемыми случайными слоями представлены в работах Yan, M., Huang, C. и Bienstman, P., Jaeger, H., Nakajima, K, Schomaker, L., а также Köster, F. и Uchida, A.

Теоретическая значимость магистерской работы. В работе выделены особенности сетей с необучаемыми случайными слоями. Реализованы и обучены ряд генеративных языковых моделей с необучаемыми слоями на наборах данных с разными естественными языками. По результатам обучения и генерации текста оценены показатели производительности, которые указали на более качественную работу гибридных моделей. Сделаны выводы о текстах, генерируемых моделями.

Структура и объём работы. Магистерская работа состоит из введения, 5 разделов, заключения, списка использованных источников и 14 приложений. Общий объём работы – 74 страниц, из них 50 страниц – основное содержание, включая 20 рисунков и 4 таблицы, список использованных источников информации – 37 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Нейронные сети и их классификация» посвящен теоретическим основам нейросетевых моделей.

Нейронная сеть (нейросеть, Neural Network, NN) — модель машинного обучения, в основе которой заложены механизмы работы мозга. Цель нейронной сети состоит в преобразовании некоторых входных данных определенного вида в результат желаемого вида. Это могут быть преобразования классификации, где выходной результат имеет вид дискретных значений, зависящих от класса входа; преобразование регрессии, где входы конвертируются в некоторые непрерывные выходные значения.

Одна из ранних моделей нейронных сетей носит названия «персептрон». Подобная модель разделена на три блока, каждый из которых имеет своё назначение. Персептрон в своем первоначальном образе обладает относительно слабой предсказывательной способностью. Например, он не способен разделять входные объекты по сложному правилу вида XOR.

Современная модель нейросети преодолевает недостатков персептрона. Для того, чтобы обновить свое значение, каждый отдельный нейрон умножает входы на коэффициенты-веса и суммирует их, а также добавляет некоторое постоянное значение-смещение. Далее, функция активации принимает свой аргумент — сумму и возвращает результат — выход нейрона. Полученное значение далее может передаваться другим нейронам или использоваться как выходное значение в сети.

Нейроны могут быть связаны в слои и представлять нейронные сети прямого распространения (Feedforward Neural Network, FNN). В сетях данного вида нейроны входного слоя принимают внешние данные и поставляют их в скрытый слой. Данные «протекают» в одном направлении, от входа к выходу. Выходы от скрытых нейронов связаны или с выходными нейронами сети, или со входами другого скрытого слоя. Одним из ограничений FNN является то, что данный вид сетей решает задачи только в случаях, когда экземпляры входных данных независимы друг от друга, например, как в задаче распознавания изображений.

Рекуррентные нейронные сети (Recurrent Neural Network, RNN) имеют похожую на структуру сетей прямого распространения, за исключением того, что в них присутствуют внутренние обратные связи. В рекуррентных нейросетях учитываются временные зависимости во входных данных. Дополнительные

соединения нейронов позволяют входным данным влиять на выходные данные в дальнейшем.

Один из общепринятых методов обучения RNN называется методом обратного распространения во времени и является адаптацией метода для FNN. В отличие от метода обратного распространения для FNN, метод нестабилен и не гарантирует сходимости к локальному минимуму функции потерь, а затраты на вычисление весов соединений относительно высоки. В силу этого требуется много времени для обучения и ограничивается практический размер RNN. Многообещающее применение RNN в области нелинейных временных рядов побудило к разработке моделей, которые используют преимущества рекуррентной обратной связи RNN, но с более простыми методами обучения.

В процессе обучения многослойных сетей наибольшую коррекцию получают именно слои, близкие к выходу сети, а входные слои часто не изменяют веса в силу эффекта затухания градиентов. Однако после обучения это обстоятельство обычно не мешает и нейронная сеть функционирует достаточно хорошо. На основе этого наблюдения была предложена идея резервуарных сетей и осуществляемых с их помощью резервуарных вычислений.

Нейросеть Echo-State Network (ESN) состоит из трех компонентов — входного слоя, динамического резервуара и выходного слоя (считывателя). Резервуар представляет собой набор нелинейных нейронов (обычно с сигмоидальной функцией активации), случайным образом соединенных друг с другом. Каждый нейрон сети имеет сетевое входное соединение и сетевое выходное соединение. Выходные данные представляют собой линейную комбинацию всех значений состояний нейронов резервуара.

Резервуар управляется согласно формуле:

$$\vec{x}(t) = f(\mathbf{W}_{in}u(t) + \mathbf{W}\vec{x}(t - 1)),$$

где $u(T)$ — функция, зависящая от времени, которая обновляет резервуар,

f — нелинейная функция активации,

W_{in} и W — входная и внутренняя матрицы связей соответственно, которые инициализируются случайным образом и фиксируются.

Как и во всех моделях, основанных на резервуарных вычислениях, в ESN обучается только выходной слой. Выходные связи можно настроить при помощи метода наименьших квадратов для регрессии.

Второй раздел «Гиперпараметры моделей» посвящен обзору разных методов автоматического подбора гиперпараметров моделей машинного обучения.

Метод поиска по сетке (Grid Search) исполняет полный перебор всех комбинаций выбранных гиперпараметров, выбранных разработчиком модели. Главный недостаток метода поиска по сетке — он очень плохо масштабируется: если условиться, что у каждого гиперпараметра n возможных значений, то число операций, требуемых для полного перебора возрастает по закону $O(n^k)$ при росте количества гиперпараметров k .

Метод случайного поиска является развитием идеи Grid Search. Данный метод случайно выбирает разные конфигурации гиперпараметров и дает возможность приближенно решить ту же задачу за наперед заданное количество построений модели, тем самым сократив время и число операций перебора для особенно больших конфигураций множеств гиперпараметров.

Суть алгоритма Successive halving состоит в последовательном обучении, определении производительности экземпляров моделей и отсеивании лучших из кандидатов из текущего набора. Обычно на этапе отсеивания выбирается половина от общего числа моделей в наборе. Данный алгоритм обычно превосходит по скорости алгоритмы случайного перебора и перебора по сетке на одинаковых наборах данных. Однако Successive halving может отсеять потенциально лучших кандидатов, производительность которых раскрывается только на поздних этапах обучения.

Автоматический подбор параметров через метод Hyperband комбинирует случайный поиск и Successive halving. Данный способ несколько раз случайным образом выбирает наборы из всего пространства сочетаний гиперпараметров, а затем путем последовательного сокращения добивается выбора лучшей модели-кандидата. Hyperband на каждый этап Successive halving выделяет приблизительно одинаковое число ресурсов.

Третий раздел «Программные средства реализации нейронных сетей» посвящен описанию языка Python и надстроек для работы с нейронными сетями разных видов, архитектур и масштабов.

TensorFlow — бесплатная платформа машинного обучения на Python с открытым исходным кодом, разработанная в основном в Google. Основная цель TensorFlow — дать возможность манипулировать математическими выражения-

ми с числовыми тензорами. TensorFlow может автоматически вычислять градиент любого дифференцируемого выражения; работать не только на обычных, но также на высокопараллельных аппаратных ускорителях; распределять вычисления между множеством компьютеров.

Keras — это библиотека глубокого обучения для Python, основанная на TensorFlow, которая обеспечивает удобный способ определения и тренировки моделей глубокого обучения.

Четвертый раздел «Программные средства обучения нейронных сетей» посвящен описанию вспомогательных библиотек для обучения моделей машинного обучения.

KerasTuner — библиотека, дополняющая возможности пакета глубокого обучения Keras функциями подбора гиперпараметров моделей. KerasTuner работает с функциями-конструкторами моделей Keras. Код функции должен выполнять создание модели привычным способом, например, через класс `Sequential`. Или же, можно использовать функциональный стиль объявления. Вместо фиксированных значений гиперпараметров подставляются специальные вызовы методов класса `HyperParameters`.

TensorBoard предназначен для визуализации разных аспектов того, как протекает процесс обучения моделей. Разработчик ML-решений может применять TensorBoard вместе с KerasTuner для того, чтобы составить себе более полную картину и помочь выбрать конкретный экземпляр модели.

Пятый раздел «Практическая часть: применение резервуарных вычислений для генерации текста» содержит описание решаемой задачи генерации текста. Также в этом разделе описываются вариации моделей такие, как модель с механизмом внимания, гибридная глубокая модель и модель с вниманием к последовательности векторов. Здесь же приведены сведения о программной реализации моделей и скриптов обучения с подбором гиперпараметров, обсуждаются результаты применения моделей.

В рамках практической части работы выполнены реализация и анализ применения подходов к генерации символьных последовательностей на основе двух корпусов текстов: пьесе Вильяма Шекспира и баснях Ивана Андреевича Крылова. Были проведены численные эксперименты, в которых производилось обучение языковых моделей с несколькими видами нейронных сетей с необучаемыми слоями — классической с резервуаром, резервуарной сети с применением меха-

низма внимания (гибридного метода) и других. Реализация моделей генерации текста была произведена с использованием библиотеки Keras и инструментария TensorFlow.

Источником набора русских текстов стал вебсайт-библиотека lib.ru Максима Мошкова. С данного ресурса был извлечен документ с баснями И.А. Крылова. Извлечение произведено с помощью библиотек requests и BeautifulSoup языка Python. В процессе извлечения набор текстов также был обработан. Версия творчества И.А. Крылова, представленная в lib.ru, имела в ряде мест искажения, например, в лице лишних символов и латинских символов, которые присутствовали на месте кириллических (например, латинская буква «o» вместо кириллической «о»). Недостатки набора были исправлены.

Первый вариант модели реализован в формате классической модели Echo-State Network. Второй вариант дополнял резервуарные вычисления механизмом внимания (Attention-enhanced Reservoir Computing, AERC): вместо использования фиксированных выходных весов как в обычных моделях RC используют веса внимания, которые меняются в зависимости от входных данных.

Также были реализованы глубокие варианты обоих ранее рассмотренных вариантов моделей. Число дополнительных скрытых слоев менялось от 2 до 6: они добавлялись в виде сочетания пар модулей Keras, SimpleRNN и Dense. Скрытые слои SimpleRNN настраивались так, чтобы на выходе они один вектор для каждого входного экземпляра. Размерность состояния модулей SimpleRNN равномерно возрастала от первого до последнего слоя. Размерность Dense-слоев с каждым разом становилась в два раза больше. Начальный размер обучаемого слоя Dense и конечная размерность состояния SimpleRNN выделялись как гиперпараметры модели и подбирались автоматическими методами.

В исходном виде механизм внимания принимает на вход результат работы резервуара. Четвертый вариант модели менял работу данного механизма так, чтобы он обрабатывал все промежуточные состояния резервуара. Это было реализовано через аргумент Keras `return_sequences=True`, также был применен слой Flatten, который компонует данные нескольких векторов в один; с этим вектором может работать механизм внимания без изменений в вычислительных формулах.

Параметры моделей подстраивались для предсказания следующего символа в последовательности методом градиентного спуска по формуле (0.1), ми-

минимизируя потерю кросс-энтропии:

$$\mathbf{W}_{net}^{(s+1)} = \mathbf{W}_{net}^{(s)} - \gamma \nabla F(\mathbf{W}_{net}^{(s)}, \mathbf{R}), \quad (0.1)$$

где s — индекс подмножества обучающих данных, используемого на каждом шаге обновления (обучающий пакет),

γ — скорость обучения, вещественное число,

∇F — градиент функции потерь кросс-энтропии,

W_{net} — матрица с весами модели,

\mathbf{R} — значения экземпляров обучающих данных модели.

Во всех конфигурациях моделей обучаемые веса оптимизировались с помощью алгоритма обратного распространения ошибки и оптимизатора Adam, распространенного варианта стохастического градиентного спуска. Оптимизация выполнялась с шагом 10^{-3} в течении нескольких проходов по обучающему набору.

Гиперпараметры перебирались в режиме Hyperband с значением параметра бюджета ресурсов $R = 12$. В процессе подбора ранжирование моделей с разными значениями опиралось на их значение функции потерь (кросс-энтропию).

У моделей с классической архитектурой выбирались значения размерности векторов-эмбеддингов из дискретного множества $\{4, 8, 16\}$ и размерность необучаемого резервуарного слоя из множества $\{300, 750, 1200\}$. Также в качестве гиперпараметра был выбран параметр обучаемости самого слоя подготовки векторов-эмбеддингов. У «глубоких» моделей вынесен параметр числа скрытых слоев, этих слоев могло быть от 1 до 3. Для моделей архитектуры с механизмом внимания помимо вышеописанных параметров слоев эмбеддингов и резервуара также подбирался гиперпараметр N_{hid} размерности нейросети, осуществляющей подсчет весов внимания. Это 10, 50 или 100 нейронов.

Чтобы проверить модели, была измерена кросс-энтропия в качестве основной метрики производительности. Также оценивалась генерация в замкнутом цикле: входная последовательность обрабатывалась слоями модели, далее предсказанный символ переходил во входные данные и процесс повторялся.

Для объективной оценки качества генерации моделей также использовалась метрика редакционного расстояния (расстояние Левенштейна): для сгенерированных слов производился поиск по словарю русского языка слов с наименьшим показателем метрики расстояния. Полученные метрики усреднялись

по тексту. В качестве структуры данных, по которой велся поиск близких слов, было выбрано ВК-дерево.

При ранжировании моделей по возрастанию функции потерь первыми по качеству генерации оказались модели с механизмом внимания, далее идут модели с классической архитектурой. В обоих случаях глубокие модели производительнее и показывают более низкие показатели. Самый высокий показатель функции потерь имеет вариант модели с механизмом внимания, обрабатывающим всю последовательность выходов резервуара.

Обнаруженные особенности параметров лучших моделей:

- размерность необучаемого слоя у лучших моделей ниже;
- у лучших моделей слой эмбеддингов заморожен (не обучается);
- у лучших глубоких моделей было два скрытых слоя (промежуточное значение между граничными 1 и 3).

Рассмотренные модели оказались не в состоянии точно генерировать слова. Тем не менее структура сгенерированных слов формально близка к словарной, о чём свидетельствуют сравнительно небольшие редакционные расстояния для лучших протестированных моделей. Для достижения более качественной генерации потребовались бы более ресурсоёмкие модели со значительно большим числом параметров. Кроме того, сгенерированный текст в целом удовлетворительно воспроизводит специфику авторского стиля текста. Несмотря на неточность генерации слов, полученные образцы текста позволили нам выполнить сравнительный анализ рассмотренных нейросетевых моделей, что и являлось основной целью данной работы.

Выполненное в настоящей работе исследование позволяет сделать вывод, что дополнение резервуарных сетей слоями, обучаемыми при помощи традиционных методов глубокого обучения, позволяет получить гибридные сети с более высоким качеством работы по сравнению с базовыми резервуарными моделями. Это выражается в возможности достижения более низких значений функции потерь и более высокой степени соответствия структуры генерируемых текстов структуре наборов обучающих данных.

ЗАКЛЮЧЕНИЕ

В работе были рассмотрены различные нейросети: сети прямого распространения, рекуррентные нейросети, а также нейросети со случайными слоями в лице Echo-State Network, изучены принципы их работы. Проведено ознакомление с резервуарными вычислениями. Уделено внимание технике и особенностям работы с нейросетями на языке Python и, таким образом, были подобраны программные средства для дальнейшей работы с нейросетями.

При изучении теоретических принципов выбора гиперпараметров моделей машинного обучения были рассмотрены методы перебора по сетке, случайного поиска, градиентного спуска и байесовской оптимизации вместе с их отличиями и особенностями. Были разобраны программные инструменты для подбора гиперпараметров моделей и мониторинга процесса обучения на примере компонентов KerasTuner и TensorBoard.

В качестве базы для дальнейшего обучения и подбора параметров на практике был извлечен набор текстовых данных творчества И. А. Крылова, находящихся в интернет-источнике lib.ru. Сбор и обработка были произведены с применением библиотек nfcusts и BeautifulSoup4.

В рамках практического эксперимента с данными произведений В. Шекспира и И. А. Крылова были реализованы модели классических резервуарных вычислений и резервуара с улучшенным вниманием на уровне символов, а также модели с дополнительными скрытыми слоями. Проведен анализ производительности в задаче генерации текста на естественном языке. Подобраны гиперпараметры моделей с помощью средств KerasTuner, сделаны выводы относительно показателей производительности, цель работы достигнута.

Исследования в области РС могут сыграть важную роль в нескольких важных прикладных областях. Классические резервуарные сети, хотя и ограничены в своей способности распознавать сложные зависимости, показывают весьма хорошие результаты при относительно простой архитектуре. Сети с механизмом внимания, в свою очередь, обеспечивают баланс между эффективностью и производительностью.

Отдельные части магистерской работы были опубликованы / представлены на конференциях:

1. Публикация «Резервуарные вычисления и нейросети» в сборнике 11-той научно-практической конференции «Прикладная математика и информа-

тика: современные исследования в области естественных и технических наук» от 2025 года.

2. Доклад «Модель машинного обучения с элементами резервуарных вычислений для генерации текста» на студенческой научной конференции факультета СГУ КНиИТ, 13.04.2026.

Основные источники информации:

1. Summary of ChatGPT-Related research and perspective towards the future of large language models [Text] / Yiheng, L., Tianle, H., Siyuan, M., [et al.] // Meta-Radiology. — 2023. — Vol. 1, no. 2. — P. 100017.
2. Emerging opportunities and challenges for the future of reservoir computing [Text] / Yan, M., Huang, C., Bienstman, P., [et al.] // Nature Communications. — 2024. — Vol. 15, no. 1. — P. 2056.
3. Шолле, Ф. Глубокое обучение на Python [Текст]. — Санкт-Петербург : Питер, 2023.
4. Jaeger, H. Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "echo State Network" Approach [Text]. — Bonn : GMD Forschungszentrum Informationstechnik, 2002.
5. Nakajima, K. Reservoir Computing Theory, Physical Implementations, and Applications: Theory, Physical Implementations, and Applications [Text] / Nakajima, K. and Fischer, I. — Singapore : Springer, 2021.
6. Schomaker, L. A neural oscillator-network model of temporal pattern generation [Text] // Human Movement Science. — 1992. — Vol. 11, no. 1. — P. 181–192.
7. Yang, L. On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice [Text] / Yang, L. and Shami, A. // arXiv e-prints. — 2022. — P. arXiv:2007.15745.
8. Köster, F. Reservoir Computing as a Language Model [Text] / Köster, F. and Uchida, A. // arXiv e-prints. — 2025. — P. arXiv:2507.15779.