

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра информатики и программирования

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАБОТЫ СПАЙКОВЫХ НЕЙРОННЫХ
СЕТЕЙ ДЛЯ ЗАДАЧ РЕГРЕССИИ И КЛАССИФИКАЦИИ**
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Фролова Святослава Александровича

Научный руководитель:

старший преподаватель

Лапшева Е.Е.

подпись, дата

Зав. кафедрой:

к.ф.-м.н., доцент

Огнева М. В.

подпись, дата

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы. В последние годы наблюдается стремительное развитие области нейронных вычислений, в том числе и альтернативных подходов к архитектуре и принципам работы нейронных сетей. Спайковые нейронные сети, в отличие от классических искусственных нейронных сетей (Artificial Neural Networks, ANN), обрабатывают данные во временной области и реагируют на входные импульсы (спайки), что делает их потенциально более энергоэффективными и адаптированными к асинхронной обработке информации. Интерес к SNN обусловлен не только их биологической правдоподобностью, но и возможностью создания вычислительных систем нового поколения, обладающих высокой скоростью обработки, низким энергопотреблением и способностью к непрерывному обучению. Эти характеристики особенно важны при внедрении интеллектуальных алгоритмов в устройства с ограниченными ресурсами, такие как сенсоры интернета вещей, автономные роботы и нейроморфные процессоры.

Цель бакалаврской работы – исследование и программная реализация методов обучения спайковых нейронных сетей для решения задач регрессии и классификации, а также анализ их эффективности.

Поставленная цель определила **следующие задачи**:

1. Провести обзор используемых источников.
2. Описать математические основы спайковых нейронных сетей.
3. Описать используемые инструменты.
4. Подобрать наборы данных для проведения эксперимента.
5. Исследовать задачи регрессии на спайковых нейронных сетях.
6. Исследовать задачи классификации на спайковых нейронных сетях.
7. Проанализировать результаты и сформулировать выводы.

Методологические основы исследования в области спайковых нейронных сетей представлены в работах Герстнера [1], Тавани [2], Индивери [3], Бинга [4], Пфайффера [5] и Роя [6].

Практическая значимость бакалаврской работы заключается в исследовании преимуществ и недостатков спайковых нейронных сетей, реализации методов машинного обучения SNN моделей для задач регрессии и классификации, оценки их качества путем сравнительного анализа результатов обучения с моделями классических нейронных сетей на датасетах с разным количеством данных.

Структура и объём работы. Бакалаврская работа состоит из введения, 5 разделов, заключения, списка использованных источников и 15 приложений. Общий объем работы – 159 страниц, из них 101 страница – основное содержание, включая 27 рисунков и 2 таблицы, список использованных источников информации.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Нейронные сети. Спайковые нейронные сети (SNN)» посвящен теоретическому описанию классических и спайковых нейронных сетей в машинном обучении.

Нейронные сети — это вычислительные модели, основанные на работе биологических нейронов. Они лежат в основе современных методов машинного и глубокого обучения, успешно справляются с такими задачами, как распознавание образов, классификация и прогнозирование. В основе их работы — имитация обработки информации в мозге.

Спайковые нейронные сети (SNN) представляют собой третье поколение нейронных сетей, которые моделируют биологические нейроны более точно, чем традиционные искусственные нейронные сети.

Основной моделью спайкового нейрона является модель Хьюджена-Хаксли (Hodgkin-Huxley model), однако для упрощения часто используется модель Леки-Интегрейт-энд-Файр (Leaky Integrate-and-Fire, LIF).

Второй раздел «Используемые инструменты» посвящен обзору фреймворков и библиотек, которые были использованы в практической части работы.

1. Реализация спайковых нейронных сетей в PyTorch.

PyTorch — это удобный и мощный инструмент (фреймворк) для машинного обучения, который позволяет строить самые разные нейросетевые архитектуры. Чтобы работать со спайковыми нейронными сетями (SNN), в PyTorch есть специальные библиотеки.

2. Snntorch.

Snntorch — это специализированная библиотека для спайковых моделей, интегрированная с PyTorch. Основные принципы её работы:

- Модели нейронов как модули PyTorch.

Каждая разновидность спайкового нейрона (например, LIF — leaky integrate-and-fire) оформлена в виде подкласса `torch.nn.Module`. Внутри модуля

хранится «мембранный потенциал» (тензор), который обновляется на каждом временном шаге по правилам интеграции и утечки.

3. Torch.

Torch предоставляет базовые механизмы, на которых строится snntorch:

- Тензоры и вычислительный граф.

`torch.Tensor` — основной объект для хранения данных и параметров, поддерживает все стандартные операции над многомерными массивами. Динамический граф (`define-by-run`) даёт гибкость при построении сложных моделей, в том числе временных расчётов спайковых сетей.

- Автоматическое дифференцирование.

Модуль `torch.autograd` автоматически отслеживает все операции над тензорами с флагом `requires_grad=True`. Пользователь может определять собственные функции прямого и обратного прохода (`torch.autograd.Function`), что необходимо для реализации суррогатных градиентов в SNN.

- Оптимизаторы и обучение.

Встроенные оптимизаторы (`torch.optim.SGD`, Adam и др.) используются без изменений — достаточно передать параметры спайковой сети. Механизм лосс-функций (`torch.nn.functional`) работает со спайковыми выходами точно так же, как и в классических сетях.

4. Torchvision.

Хотя пакет `torchvision` напрямую не содержит спайковых компонентов, он играет важную роль в подготовке и загрузке данных:

- Датасеты.

`torchvision.datasets` предоставляет стандартные наборы изображений (MNIST, CIFAR-10 и т. д.), которые часто используются в качестве источника временных последовательностей для SNN (после предварительной обработки).

- Трансформации.

`torchvision.transforms` облегчает аугментацию и нормализацию изображений: изменение размера, случайные обрезки, повороты, перевод в

тензор и т. д. Дополнительно можно написать собственный трансформер, разбивающий статическое изображение на последовательность бинарных фреймов (event-based encoding) для подачи в SNN.

- Интеграция с DataLoader.

Класс `torch.utils.data.DataLoader` автоматически формирует батчи и обеспечивает эффективную загрузку данных в процесс обучения, включая многопоточность и предварительную выборку.

Третий раздел «Реализация спайковых нейронных сетей и классических нейронных сетей на задаче регрессии» содержит подробное описание этапов практической части работы для задачи регрессии на 3 наборах данных.

В качестве среды выполнения использовалась интерактивная облачная среда Google Colaboratory.

Описание синтетического набора данных.

В этом коде используется искусственный набор данных, созданный с помощью функции `make_regression` из библиотеки `sklearn.datasets`. Он помогает создать точки на плоскости. Этот набор данных нужен для задачи регрессии, где цель — предсказать непрерывное значение на основании одного или нескольких признаков.

Параметры генерации датасета:

- `n_features=1`. Количество признаков в датасете.
- `noise=20.0`. Уровень шума, добавляемого к данным.
- `random_state=57`. Означает, что при каждом запуске кода с этим параметром будут генерироваться одни и те же данные.

Структура датасета:

- `X`: Массив признаков (в данном случае одномерный).
- `y`: Массив целевых значений, которые нужно предсказать.

Датасет разделяется на обучающую и тестовую выборки с помощью функции `train_test_split`, где 70% данных используются для обучения модели, а 30% — для тестирования.

Описание набора данных Fuel Economy.

В оригинальном датасете присутствует свыше 80 столбцов и более 80 000 записей, каждая из которых представляет собой отдельную модификацию конкретного транспортного средства. В рамках данной работы были отобраны следующие четыре признака, наиболее релевантные поставленной задаче предсказания расхода топлива:

- `comb08` – комбинированный расход топлива в милях на галлон (MPG), рассчитанный как взвешенное среднее между городским и шоссе́йным расходом. Это является целевой переменной в задаче регрессии.
- `displ` – объем двигателя автомобиля в литрах. Данный параметр отражает размер двигателя, что напрямую влияет на его мощность и потребление топлива.
- `cylinders` – количество цилиндров двигателя. Этот признак также оказывает влияние на производительность и топливную эффективность автомобиля.
- `year` – год выпуска транспортного средства. Он важен как индикатор технологического уровня модели: в более новых автомобилях, как правило, применяются более эффективные решения по расходу топлива.

Описание набора данных California Housing.

Для построения и обучения спайковой нейронной сети в задаче регрессии был использован открытый и хорошо структурированный набор данных California Housing, предоставляемый библиотекой `sklearn.datasets`. Этот датасет представляет собой результат агрегации демографической и экономической информации, собранной в рамках переписи населения в штате Калифорния (США). Он предназначен для задач машинного обучения, в которых необходимо предсказать стоимость жилья на основе ряда социально-экономических и географических признаков.

В наборе содержится 20640 наблюдений, каждое из которых представляет собой агрегированную информацию по отдельному блоку жилой застройки. Это не индивидуальные дома, а усреднённые значения по

определённой территориальной единице. Структура данных включает восемь числовых признаков (средний доход домохозяйства в районе (в десятках тысяч долларов США), средний возраст зданий в районе (в годах), среднее количество комнат в жилых единицах и т.д.) и одну целевую переменную. Целевая переменная (MedHouseVal) представляет собой медианную стоимость жилого помещения в данном районе, выраженную в сотнях тысяч долларов США. Именно это значение используется в качестве целевой переменной в задачах регрессии, в том числе и в данной работе.

Четвертый раздел «Реализация спайковых нейронных сетей и классических нейронных сетей на задаче классификации» содержит подробное описание этапов практической части работы для задачи классификации на 4 наборов данных.

Описание набора данных Palmer Pinguis.

Структура датасета: каждая запись в датасете соответствует одному представителю вида пингвинов. В данной работе использовались только числовые признаки, наиболее информативные с точки зрения биологической идентификации (длина клюва(мм), глубина клюва(мм), длина плавника(мм), масса тела пингвина (г)).

Целевая переменная species содержит название вида пингвина. В наборе представлены три категории: Adelie, Chinstrap и Gentoo. Для корректной обработки категориальных данных при обучении модели была применена техника one-hot кодирования, при которой каждая категория представляется в виде бинарного вектора.

Описание набора данных N-MNIST.

Датасет N-MNIST (Neuromorphic MNIST) представляет собой модифицированную версию классического MNIST, специально адаптированную под задачи в области нейроморфных вычислений и спайковых нейронных сетей (SNN). Его разработали исследователи из группы Garrick Orchard et al. для оценки производительности алгоритмов, способных

обрабатывать событийные данные, а не традиционные статические изображения.

Вместо привычных двумерных изображений рукописных цифр размером 28×28 пикселей, как в оригинальном MNIST, в N-MNIST используются потоки событий, сгенерированные динамическим зрительным сенсором (Dynamic Vision Sensor, DVS). Такой сенсор регистрирует изменения яркости на каждом пикселе независимо и асинхронно, что делает данные более реалистичными для задач, приближённых к работе человеческого зрения.

Для создания набора данных DVS-камера была закреплена на движущейся установке, которая воспроизводила микродвижения глаз, направляя сенсор на каждое изображение из MNIST. В результате вместо одного изображения каждая цифра представлена как последовательность событий, содержащих координаты (x , y), временную метку (timestamp) и полярность (увеличение или уменьшение яркости). Это превращает классическую задачу классификации изображений в задачу обработки временных спайковых паттернов, что идеально подходит для тестирования спайковых нейросетей.

Ключевые особенности N-MNIST:

Размерность: каждый пример — это поток событий, происходящих на сенсоре размером 34×34 пикселя.

Формат данных: каждое событие представлено как кортеж (x , y , t , p), где t — время, а p — бинарная полярность (ON/OFF).

Количество классов: 10, соответствуют цифрам от 0 до 9, аналогично MNIST.

Объём:

- 60 000 примеров для обучения.
- 10 000 примеров для тестирования.

Описание набора данных Fashion MNIST.

Вместо цифр Fashion MNIST включает изображения предметов одежды, что делает задачу классификации более приближенной к реальным задачам компьютерного зрения.

Каждое изображение в датасете представляет собой монохромную (одноканальную) картинку размером 28x28 пикселей, в градациях серого, на которой изображён один из 10 классов одежды. Классы включают такие категории, как:

- 0 — T-shirt/Топ (футболка/топ),
- 1 — Trouser (брюки),
- 2 — Pullover (свитер),
- 3 — Dress (платье),
- 4 — Coat (пальто),
- 5 — Sandal (босоножка),
- 6 — Shirt (рубашка),
- 7 — Sneaker (кроссовки),
- 8 — Bag (сумка),
- 9 — Ankle boot (ботинок).

ЗАКЛЮЧЕНИЕ

Цель выпускной квалификационной работы успешно выполнена: проведено исследование преимуществ спайковых нейронных сетей над традиционными нейронными сетями на задачах регрессии и классификации.

В процессе выпускной квалификационной работы проведен обзор используемых источников, описаны математические основы спайковых нейронных сетей, описаны используемые инструменты: PyTorch, Torch, Snntorch и Torchvision, подобраны наборы данных для проведения эксперимента, реализованы модели SNN и MLP для проведения эксперимента на задачах регрессии и классификации, также проанализированы результаты и сформулированы выводы по полученным значениям. Было использовано 3 набора данных для задачи регрессии: синтетический набор данных для точек, Fuel Economy и California Housing и 4 набора данных для задачи классификации: Palmer Penguins, N-MNIST, Fashion MNIST, Fashion MNIST с добавлением Гауссовского шума, для каждого из них построены модели SNN и ANN на основе MLP и проанализированы метрики: MSE, R^2 , точность и время обучения модели. Метрики регрессии были лучше у ANN на основе MLP, а метрики классификации – у SNN на шумовых данных. Работа представлена в студенческой конференции.

Основные источники информации:

1. Gerstner W., Kistler W.M. Spiking neuron models: single neurons, populations, plasticity. – Cambridge: Cambridge University Press, 2002. – 496 с.
2. Tavanaei A., Ghodrati A., Kheradpisheh S.R., Masquelier T., Maida A.S. Deep learning in spiking neural networks // Neural Networks. – 2019. – Vol. 115. – P. 100–112.
3. Indiveri G., Liu S. Neuromorphic engineering: merging neuroscience with microelectronics // Proceedings of the IEEE. – 2015. – Vol. 103, No. 10. – P. 1712–1735.
4. Bing J., Amir M., Bohte S.M. Survey of spiking neural network models and their applications to neuromorphic systems // Frontiers in Neuroscience. – 2018. – Vol. 12. – P. 331.
5. Pfeiffer M., Pfeil T. Deep learning with spiking neurons: opportunities and challenges // Frontiers in Neuroscience. — 2018. — Vol. 12. — P. 952.
6. Roy N., Jaiswal A., Sengupta N. Towards spike-based machine intelligence with neuromorphic computing // Nature Machine Intelligence. — 2019. — Vol. 1. — P. 33–44.