



MachineT: НОВАЯ АРХИТЕКТУРА ДЛЯ ЭНЕРГОЭФФЕКТИВНОГО МАШИННОГО ОБУЧЕНИЯ

Аннотация

Современные алгоритмы машинного обучения, особенно нейронные сети, обеспечивают высокую точность, но требуют значительных вычислительных ресурсов и энергии, а также больших объемов данных для обучения. Это ограничивает их применение на устройствах с низкой производительностью и вызывает экологические вопросы. В данной статье представлена новая архитектура MachineT, основанная на логических правилах, которая обеспечивает высокую точность и интерпретируемость при существенно меньшем энергопотреблении. Мы описываем структуру алгоритма, процесс обучения и предсказания, а также демонстрируем его эффективность на стандартных наборах данных.

Ключевые слова:

машинное обучение, искусственный интеллект, логические модели,
энергоэффективность, нейронные сети

Annotation

Modern machine learning algorithms, especially neural networks, provide high accuracy but require significant computational resources and energy, as well as large amounts of training data. This limits their use on low-performance devices and raises environmental concerns. This paper presents a new rule-based architecture, MachineT, that provides high accuracy and interpretability with significantly lower energy consumption. We describe the algorithm's structure, training and prediction process, and demonstrate its effectiveness on standard datasets.

Keywords:

machine learning, artificial intelligence, logical models, energy efficiency, neural networks



1 Введение

В последние годы нейронные сети, особенно архитектуры на основе Transformers [4], стали стандартом в задачах машинного обучения, таких как обработка естественного языка, компьютерное зрение и классификация данных. Однако их успех сопряжен с серьезными недостатками: обучение моделей, таких как GPT-4, требует десятков гигаватт-часов электроэнергии и кластеров высокопроизводительных серверов, что делает их недоступными для локальных вычислений и экологически затратными. По оценкам, обучение одной крупной модели может генерировать углеродный след, сравнимый с несколькими трансатлантическими перелетами [1].

Кроме того, нейросети, основанные на векторных представлениях, создают пространства высокой размерности, что увеличивает вычислительную сложность и затрудняет интерпретацию решений. В прошлом исследователи предлагали использовать логические подходы для создания более прозрачных и эффективных систем [2]. Основываясь на этих идеях, мы разработали новую архитектуру

MachineT, которая использует логические правила вместо традиционных векторных операций. MachineT позволяет проводить обучение и предсказания на локальных устройствах, обеспечивая высокую точность, энергоэффективность и интерпретируемость.

Цель статьи — представить архитектуру MachineT, описать её ключевые механизмы и продемонстрировать её преимущества по сравнению с существующими решениями.

2 Архитектура алгоритма

MachineT — это модульная архитектура для задач классификации, которая заменяет традиционные векторные вычисления набором логических правил (клауз). Каждая клауза анализирует входные признаки и их отрицания, формируя внутреннее состояние, которое отражает их значимость для конкретного класса.

2.1 Базовая модель

Базовая MachineT предназначена для бинарной классификации (один класс против остальных). Она состоит из следующих компонентов:

- **Состояние машины:** Структура, хранящая степени уверенности в значимости каждого признака и его отрицания.



- **Клаузы:** Логические правила, которые сопоставляют входные данные с состоянием машины. Например, клауза может указывать, что признак $x_i = 1$ увеличивает вероятность принадлежности к классу.

- **Обновление состояния:** Вероятностный механизм, усиливающий или ослабляющий значимость признаков на основе входных данных.

Обучение: На этапе обучения MachineT итеративно обновляет своё состояние, анализируя входные образцы. Для каждого образца модель вычисляет, какие признаки способствуют правильной классификации, и корректирует их веса. Процесс напоминает адаптивное обучение с подкреплением, но основан на логике, а не на градиентном спуске.

Пример работы алгоритма и вычисления клауз:

На начальном этапе все состояния (синие точки) находятся на границе памяти (красная линия, уровень 2), что отражает отсутствие знаний автомата о данных. На этом этапе MachineT ещё не обучена и не может делать предсказания, так как веса всех признаков ("there is a tail", "meows", "barks", "there is wool", "white" и их отрицания) равны нулю относительно принятия решения. Рисунок 1.

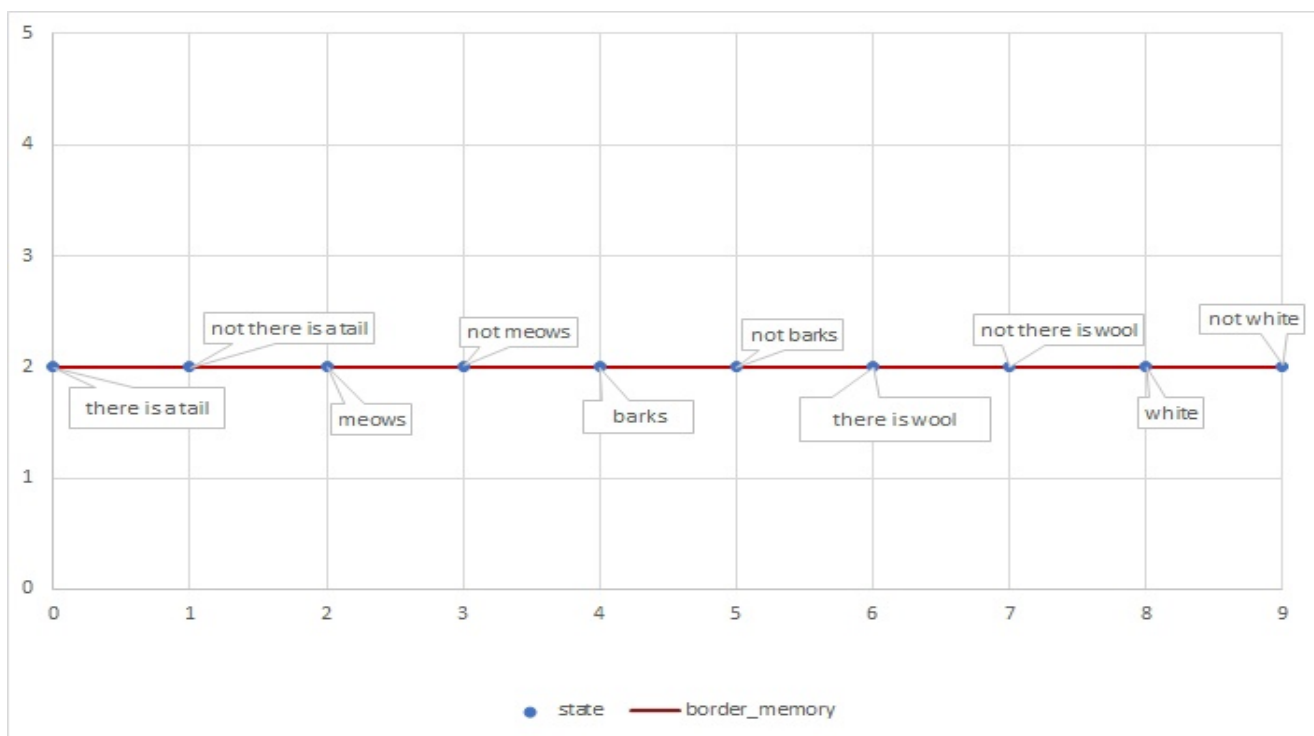


Рисунок 1- Инициализация клаузы

Обучение на первом наборе данных: ["not there is a tail" (нет хвоста), "not meows" (не мяукает), "barks" (лает), "there is wool" (есть шерсть), "not white" (не белый)]. На этом



этапе автомат обновляет состояния на основе обучающего правила: если признак присутствует в данных, его состояние повышается на +1 (выше границы памяти), если отсутствует — понижается на -1 (ниже границы памяти). В данном случае: "there is a tail" (-1), "not there is a tail" (+1), "meows" (-1), "not meows" (+1), "barks" (+1), "not barks" (-1), "there is wool" (+1), "not there is wool" (-1), "white" (-1), "not white" (+1). Таким образом, состояния для признаков barks, there is wool, not there is a tail, not meows и not white поднимаются выше границы памяти (уровень 3), а признаки there is a tail, meows, white, not barks и not there is wool опускаются ниже границы памяти (уровень 1). В этом примере не используется вероятностная модель для обновления состояния; вместо этого применяется детерминированное правило, при котором каждое состояние строго увеличивается или уменьшается на 1 в зависимости от наличия или отсутствия признака.

Рисунок 2.

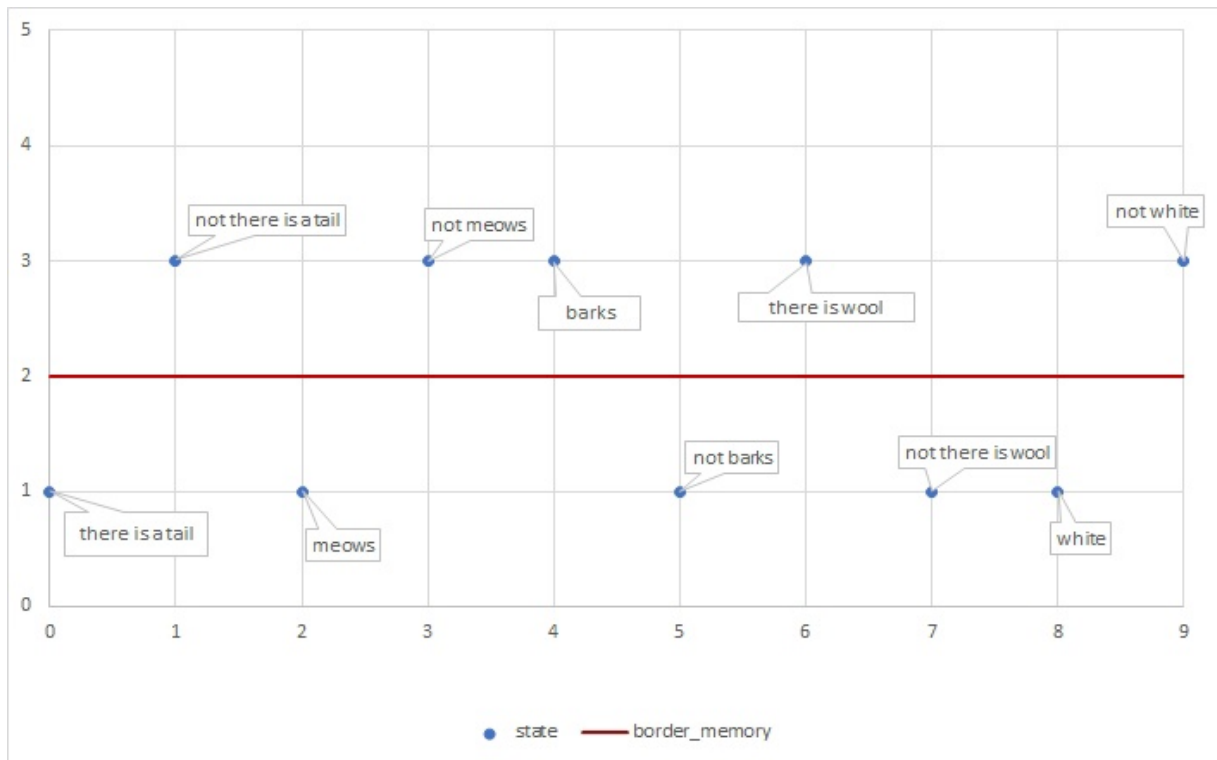


Рисунок 2 - Обучение на первом наборе данных

Обучение на втором наборе данных: ["there is a tail" (есть хвост), "not meows" (не мяукает), "barks" (лает), "there is wool" (есть шерсть), "white" (белый)]. Автомат продолжает обновлять состояния, применяя те же детерминированные правила. Для новых данных: "there is a tail" (+1), "not there is a tail" (-1), "meows" (+1), "not meows" (-1), "barks" (+1), "not barks" (-1), "there is wool" (+1), "not there is wool" (-1), "white" (+1), "not white" (-1). После этого шага состояние "there is a tail" поднимается с уровня 1 до уровня

2, "not there is a tail" опускается с уровня 3 до уровня 2, "meows" остается на уровне 1, "not meows" опускается с уровня 3 до уровня 2, "barks" и "there is wool" укрепляются на уровне 3, "not barks" и "not there is wool" остаются на уровне 1, "white" поднимается с уровня 1 до уровня 2, а "not white" опускается с уровня 3 до уровня 2. Это отражает накопление знаний о признаках, характерных для класса (например, "собака"). Здесь также используется детерминированное обновление, без применения вероятностной модели. Рисунок 3.

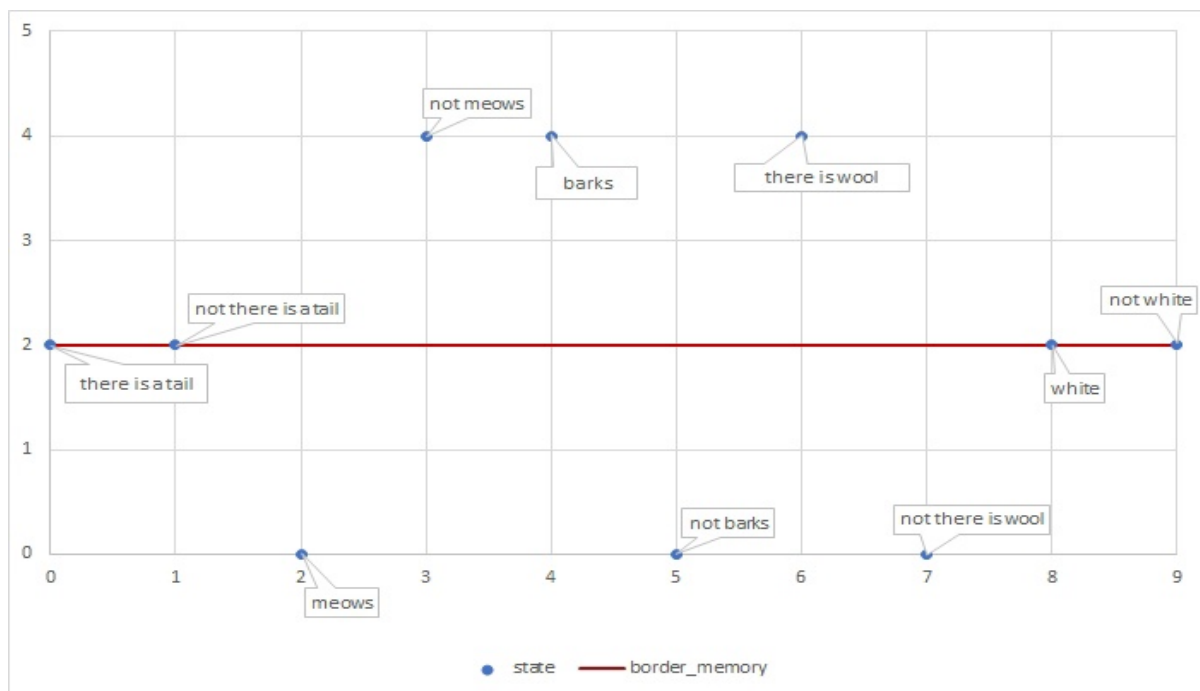


Рисунок 3 - обучение на втором наборе данных

Прогнозирование на новых данных:

Тестирование на данных: ["not there is a tail" (нет хвоста), "meows" (мяукает), "not barks" (не лает), "there is wool" (есть шерсть), "not white" (не белый)]. На этапе предсказания MachineT считает количество позитивных и негативных правил. Позитивные правила — это признаки, чьи состояния выше границы памяти (уровень 3), негативные — ниже границы памяти (уровень 1). На основе предыдущего обучения: "there is a tail" (уровень 2, нейтрально), "meows" (уровень 1, -1), "barks" (уровень 3, +1), "there is wool" (уровень 3, +1), "white" (уровень 2, нейтрально). Для тестовых данных: "not there is a tail" (нейтрально), "meows" (-1), "not barks" (-1), "there is wool" (+1), "not white" (нейтрально). Итог: 1 позитивное правило ("there is wool") против 4 негативных ("meows", "not barks", "not there is a tail", "not white"). Так как негативных правил больше,

автомат не может уверенно классифицировать объект и сигнализирует о неопределённости. Рисунок 4.

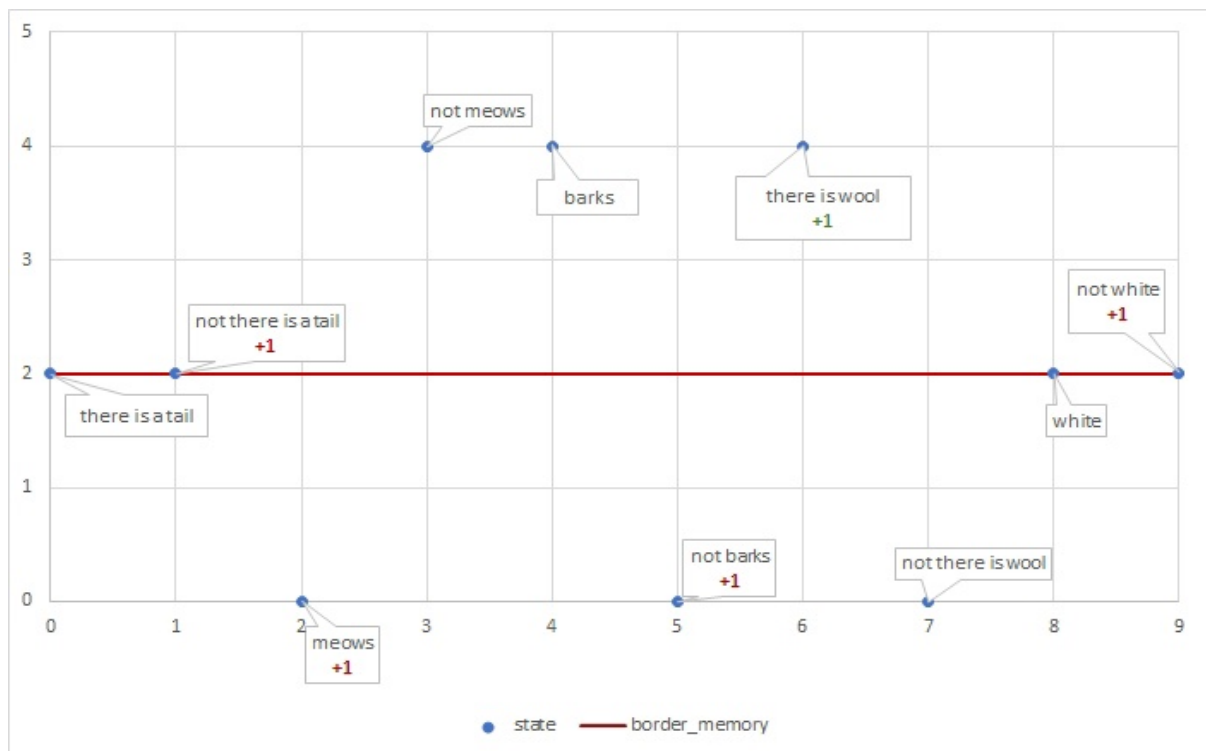


Рисунок 4 - ["not there is a tail", "meows", "not barks", "there is wool", "not white"]

Тестирование на данных: ["there is a tail" (есть хвост), "not meows" (не мяукает), "barks" (лает), "there is wool" (есть шерсть), "not white" (не белый)]. Применяя те же состояния: "there is a tail" (уровень 2, нейтрально), "not meows" (+1, так как "meows" на уровне 1), "barks" (+1), "there is wool" (+1), "not white" (нейтрально). Итог: 3 позитивных правила ("not meows", "barks", "there is wool") против 2 негативных ("there is a tail" и "white" нейтральны). Так как позитивных правил больше, автомат классифицирует объект как принадлежащий к классу "собака". Рисунок 5.



Рисунок 5 - ["there is a tail", "not meows", "barks", "there is wool", "not white"]

2.2 Многоклассовая модель

Для задач с несколькими классами MachineT использует подход "один против всех". Модель состоит из набора независимых базовых машин, каждая из которых отвечает за свой класс.

- **Обучение:** Входные образцы направляются в соответствующую базовую машину в зависимости от их метки класса. Обучение проводится итеративно в течение нескольких эпох, что позволяет каждой машине адаптироваться к своему классу.
- **Предсказание:** Все базовые машины оценивают входной образец и выдают степени уверенности. Класс с наибольшей суммой голосов выбирается как результат. Если уверенность ниже заданного порога, модель может отказаться от предсказания, сигнализируя о неопределённости.

2.3 Преимущества архитектуры

- **Энергоэффективность:** Отсутствие операций с высокой размерностью снижает вычислительные затраты.
- **Интерпретируемость:** Логические правила позволяют объяснить, почему модель приняла то или иное решение.



- **Модульность:** Независимые базовые машины упрощают параллельную обработку и масштабирование.

3 Сравнение с существующими методами

Для оценки преимуществ предложенного алгоритма MachineT мы провели сравнительный анализ с традиционными нейронными сетями, в частности с многослойным перцептроном (MLP) и деревьями решений. Сравнение проводилось по ключевым критериям, включая эффективность обучения, вычислительную сложность, интерпретируемость, способность выявлять недостатки в знаниях и устойчивость к переобучению. Эксперименты были выполнены на идентичных условиях, чтобы обеспечить объективность результатов.

3.1 Качественное сравнение характеристик

Таблица 1 – Сравнение характеристик нейронных сетей и MachineT

Критерий	Нейросети	MachineT
Осознает ли незнание?	Нет, нейросети это перебор вероятностей и ответ будет всегда вне зависимости от знаний	Да, машина понимает свои пробелы в знаниях благодаря порогу знаний T
Эффективность обучения	Низкая, обучение сводится к матричным операциям и подбору весов	Высокая, обучение основано на построении логических правил
Сложность вычислений	Сложные вычислительные операции (масштабирование, параллелизм)	Легкие вычислительные операции (логические)
Эффективность интерпретации	Низкая, модель действует как «черный ящик» [3]	Высокая для простых правил, снижается с ростом сложности



Тип представления данных	Векторы чисел с плавающей запятой (float)	Бинарные данные (0 и 1)
Риск переобучения	Есть риск переобучения при наличии шума или неточностей	Есть риск переобучения при наличии шума или неточностей

3.2 Экспериментальная методология

Для количественного сравнения мы реализовали алгоритм MachineT и многослойный перцептрон на языке C++. В качестве эталонного метода использовалась реализация MLP из библиотеки `tiny_dnn`. Архитектура MLP включала один скрытый слой (16 нейронов), функцию активации ReLU и выходной слой с функцией softmax. Оптимизация проводилась с помощью стохастического градиентного спуска с фиксированной скоростью обучения $\eta=0.001$ $\eta = 0.001$ $\eta=0.001$.

Эксперименты выполнялись на вычислительной системе с процессором Intel Xeon Cooperlake (2 ядра, 3.0 ГГц), оперативной памятью 3.8 ГБ DDR4, накопителем 60 ГБ SSD и операционной системой Ubuntu 6.8.0. Для сравнения использовался синтетический набор данных для бинарной классификации, содержащий 12 объектов с 10 признаками.

Обучение проводилось в двух режимах: с одной эпохой и с 50 эпохами, чтобы оценить производительность алгоритмов на разных стадиях обучения. Измерялись время обучения, время предсказания и точность на обучающем наборе.

3.3 Результаты экспериментов

Результаты экспериментов представлены в таблицах 2 и 3.

Таблица 2 – Результаты сравнения обучения для одной эпохи

Критерий	Время обучения (секунды)	Время предсказания (секунды)	Точность предсказаний на обучающем наборе	Потребление памяти (КБ)
MachineT	2.857e-06	6.42e-07	100%	90.80

Многослойный перцептрон	0,002	0,001	66.67%	98.95
-------------------------	-------	-------	--------	-------

Таблица 3 – Результаты сравнения для 50 эпох

Критерий	Время обучения (секунды)	Время предсказания (секунды)	Точность предсказаний на обучающем наборе	Потребление памяти (КБ)
MachineT	9.4541e-05	6.49e-07	100%	90.75
Многослойный перцептрон	0,11	0,001	100%	102.32

Таблица 4 – Сравнение с решающим деревом

Критерий	Время обучения (секунды)	Время предсказания (секунды)	Точность предсказаний на обучающем наборе	Потребление памяти (КБ)
Решающие деревья	0.011791	4.4e-05	100%	75.50

3.4 Анализ результатов

MachineT демонстрирует значительные преимущества в скорости и эффективности обучения, что делает его перспективным для задач с ограниченными вычислительными ресурсами, таких как встраиваемые системы или устройства для работы в реальном времени. Способность алгоритма выявлять пробелы в знаниях благодаря порогу T позволяет повысить надежность предсказаний в условиях неполных данных.

4 Выводы

В данной статье представлена новая архитектура MachineT, разработанная для энергоэффективного машинного обучения. Основанная на логических правилах и бинарных данных, MachineT предлагает альтернативу традиционным нейронным сетям, снижая вычислительную сложность и повышая интерпретируемость решений. Ключевая особенность алгоритма — порог уверенности T , позволяющий выявлять пробелы в знаниях и повышать надежность предсказаний в условиях ограниченных данных.

Экспериментальное сравнение с многослойным перцептроном (MLP) на синтетическом наборе данных для бинарной классификации подтвердило преимущества MachineT. Алгоритм достигает 100% точности на обучающем наборе уже после одной эпохи, демонстрируя время обучения и предсказания на несколько порядков ниже, чем у MLP (2.857×10^{-6} с против 0.002 с для обучения и 6.42×10^{-7} с против 0.001 с для предсказания). Эти результаты подчеркивают потенциал MachineT для приложений, требующих высокой скорости и низкого энергопотребления, таких как встраиваемые системы, устройства для работы в реальном времени и локальные вычисления.

Научный вклад работы заключается в разработке модульной и интерпретируемой архитектуры, которая сочетает энергоэффективность с точностью, сравнимой с нейронными сетями. Практическая значимость MachineT связана с её применимостью в сценариях с ограниченными ресурсами, где традиционные модели оказываются неэффективными.

В будущем мы планируем протестировать MachineT на устройствах Raspberry Pi, чтобы оценить её производительность и энергопотребление в реальных условиях встраиваемых систем. Также предполагается развивать алгоритм для применения в задачах обработки естественного языка (NLP), распознавания изображений, генеративных моделей искусственного интеллекта и обучения с подкреплением. Эти направления позволят расширить функциональность MachineT, сохраняя её энергоэффективность. Дополнительные усилия будут направлены на автоматизацию настройки порога T и оптимизацию алгоритма для задач с большим числом классов и сложными признаками. Эти шаги укрепят позицию MachineT среди современных решений для устойчивого машинного обучения.



Список литературы

1. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171–4186. <https://aclanthology.org/P19-1355/>
2. Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79(8), 2554–2558. <https://doi.org/10.1007/BF03037089>
3. Benítez, J. M., Castro, J. L., & Requena, I. (1997). Are artificial neural networks black boxes? IEEE Transactions on Neural Networks, 8(5), 1156–1164. http://150.214.190.154/sites/default/files/ficherosPublicaciones/0818_1997-benitez-IEEETNN.pdf
4. Transformer (deep learning architecture). (n.d.). In Wikipedia. Retrieved April 13, 2025, from [https://en.wikipedia.org/wiki/Transformer_\(deep_learning_architecture\)](https://en.wikipedia.org/wiki/Transformer_(deep_learning_architecture))