

УДК 004.032.26:517.9

DOI: 10.18413/2518-1092-2026-11-3-0-7

Монгуш М.А.

**ПАРАЛЛЕЛИЗАЦИЯ ВЫЧИСЛЕНИЙ
В ЖИДКИХ НЕЙРОННЫХ СЕТЯХ
С СОХРАНЕНИЕМ ГАРАНТИЙ УСТОЙЧИВОСТИ**

Сибирский федеральный университет,
пр. Свободный, 79, г. Красноярск, 660041, Россия

e-mail: makssun2018@gmail.com

Аннотация

Жидкие нейронные сети (Liquid Neural Networks, LNN) относятся к классу непрерывных моделей машинного обучения, в которых эволюция скрытого состояния описывается системой обыкновенных дифференциальных уравнений. Существенным достоинством LNN является их адаптивность во времени и наличие строгих теоретических гарантий устойчивости по входу-состоянию (Input-to-State Stability, ISS), что обеспечивает надёжную работу в условиях шумных, нерегулярных и внераспределённых входных данных. Данные свойства делают LNN перспективными для применения в системах управления, робототехнике и обработке временных рядов. Несмотря на указанные преимущества, практическое использование LNN ограничено высокой вычислительной сложностью. Численные методы решения обыкновенных дифференциальных уравнений обладают выраженной последовательной природой, что затрудняет эффективную параллелизацию вычислений и приводит к увеличению времени обучения и вывода модели, особенно при реализации на GPU и распределённых вычислительных платформах. В работе рассматривается метод параллелизации вычислений для LNN – Stable Parallel Liquid Recurrence (SPLR). Подход основан на представлении задачи вычисления траектории скрытого состояния на временном интервале в виде решения системы нелинейных уравнений, эквивалентной дискретизированной динамике LNN. Для её решения применяется итерационный метод неподвижной точки, допускающий параллельное выполнение. С целью сохранения устойчивости на каждой итерации вводится стабилизирующая проекция на область параметров и состояний, в которой выполняются условия ISS. Получены теоретические результаты, устанавливающие сходимость предложенного итерационного алгоритма и сохранение свойства устойчивости по входу-состоянию при переходе к параллельной схеме вычислений. Показано, что стабилизирующая проекция предотвращает накопление численных ошибок и ограничивает рост возмущений, возникающих при параллельной обработке. В отличие от существующих методов параллелизации рекуррентных вычислений, ориентированных преимущественно на ускорение, метод SPLR сохраняет ключевое преимущество LNN – устойчивость к входным и численным возмущениям. Практическая значимость работы заключается в расширении области применимости жидких нейронных сетей в системах реального времени и на встраиваемых платформах с ограниченными вычислительными ресурсами.

Ключевые слова: жидкие нейронные сети; параллелизация; устойчивость по входу к состоянию; Neural ODE; рекуррентные нейронные сети; распределённые вычисления

Для цитирования: Монгуш М.А. Параллелизация вычислений в жидких нейронных сетях с сохранением гарантий устойчивости // Научный результат. Информационные технологии. – Т.11, №3, 2026. – С. 75-85. DOI: 10.18413/2518-1092-2026-11-3-0-7

Mongush M.A.

PARALLELIZATION OF COMPUTATIONS IN LIQUID NEURAL NETWORKS WITH PRESERVATION OF STABILITY GUARANTEES

Siberian Federal University,
79 Svobodny Ave., Krasnoyarsk, 660041, Russia

e-mail: makssun2018@gmail.com

Abstract

Liquid neural networks (LNNs) are a class of continuous machine learning models in which the evolution of a hidden state is described by a system of ordinary differential equations. A significant advantage of LNNs is their time adaptability and the presence of strict theoretical guarantees of input-to-state stability (ISS), which ensures reliable operation under noisy, irregular, and undistributed input data. These properties make LNNs promising for application in control systems, robotics, and time series processing. Despite these advantages, the practical use of LNNs is limited by high computational complexity. Numerical methods for solving ordinary differential equations have a pronounced sequential nature, which complicates efficient parallelization of computations and leads to increased training and inference times, especially when implemented on GPUs and distributed computing platforms. This paper discusses a parallelization method for LNNs: Stable Parallel Liquid Recurrence (SPLR). The approach is based on representing the problem of computing the trajectory of a hidden state over a time interval as the solution of a system of nonlinear equations equivalent to the discretized dynamics of an LNN. An iterative fixed-point method, which allows parallel execution, is used to solve it. To maintain stability at each iteration, a stabilizing projection onto the parameter and state domain in which the ISS conditions are satisfied is introduced. Theoretical results are obtained establishing the convergence of the proposed iterative algorithm and the preservation of the input-state stability property upon transition to a parallel computing scheme. It is shown that the stabilizing projection prevents the accumulation of numerical errors and limits the growth of disturbances arising during parallel processing. Unlike existing methods for parallelizing recurrent computations, which are primarily focused on speedup, the SPLR method preserves the key advantage of LNNs: their robustness to input and numerical disturbances. The practical significance of this work lies in expanding the applicability of liquid neural networks in real-time systems and on embedded platforms with limited computing resources.

Keywords: liquid neural networks; parallelization; input-to-state stability; neural ODE; recurrent neural networks; distributed computing

For citation: Mongush M.A. Parallelization of Computations in Liquid Neural Networks with Preservation of Stability Guarantees // Research result. Information technologies. – T.11, №3, 2026. – P. 75-85. DOI: 10.18413/2518-1092-2026-11-3-0-7

ВВЕДЕНИЕ

Актуальность исследования

Для анализа последовательностей традиционно применяются рекуррентные архитектуры (RNN), включая популярные конфигурации GRU [6] и LSTM. Тем не менее, фиксированный временной шаг таких систем накладывает ограничения при моделировании непрерывных процессов или обработке данных с неравномерными интервалами. Эту проблему в 2018 году частично решила команда Чена, представив концепцию Neural ODE. Авторы заменили привычные дискретные слои непрерывной динамикой на базе обыкновенных дифференциальных уравнений, что обеспечило высокую гибкость вычислений при анализе временных рядов. Логическим продолжением данного вектора стали жидкие нейросети (LNN). Так, в исследовании Хасани и коллег [12] была описана архитектура LTC (Liquid Time-Constant Networks), базирующаяся на биологических принципах работы нервной системы нематоды *C. elegans*. Ключевое отличие LTC от стандартных рекуррентных моделей заключается в умении подстраивать внутренние параметры динамики под входящий сигнал, что гарантирует быструю адаптацию к меняющейся среде. Важным

преимуществом таких систем выступает наличие доказанных гарантий устойчивости по входу к состоянию (ISS), благодаря чему они востребованы в робототехнике и автоматизированном управлении.

Однако повсеместному внедрению жидких нейросетей мешает их колоссальная вычислительная трудоемкость. Процесс расчета в них носит преимущественно последовательный характер, из-за чего не удастся полноценно задействовать ресурсы современных графических и тензорных ускорителей (GPU/TPU). И если для стандартных RNN или трансформеров алгоритмы распараллеливания уже детально проработаны, то для LNN эта область до сих пор мало изучена. Свежие подходы к параллелизации нелинейных рекуррентных сетей [10] выглядят многообещающе, но они игнорируют уникальные архитектурные свойства жидких моделей и их специфику ISS-устойчивости. Настоящая работа посвящена решению проблемы масштабируемости LNN. В исследовании оцениваются перспективы внедрения параллельных вычислений, которые позволяют форсировать скорость работы моделей без потери строгих теоретических гарантий их ISS-устойчивости.

Цель и задачи исследования

Цель исследования. Теоретически обосновать метод параллелизации вычислений для жидких нейронных сетей, который бы сохранял свойство ISS-устойчивости.

Задачи исследования:

1. провести анализ существующих методов параллелизации рекуррентных вычислений и выявить их ограничения применительно к LNN;
2. рассмотреть параллельный итерационный алгоритм для решения системы нелинейных уравнений, возникающей при попытке одновременного вычисления всех скрытых состояний LNN;
3. доказать сходимость рассмотренного алгоритма;
4. провести теоретический анализ устойчивости параллельного алгоритма и доказать, что он сохраняет свойство ISS.

Научная новизна данной работы заключается в следующем:

1. рассмотрен перспективный метод параллелизации вычислений для жидких нейронных сетей, который теоретически гарантирует сохранение свойства ISS-устойчивости;
2. рассмотрено понятие стабилизирующей проекции в контексте итерационных методов решения систем нелинейных уравнений, возникающих при параллелизации рекуррентных вычислений;
3. доказаны теоремы о сходимости предложенного алгоритма и сохранении ISS-устойчивости на каждой итерации.

Практическая значимость работы определяется возможностью применения разработанного метода для ускорения обучения и инференса LNN в критически важных приложениях, таких как:

1. навигация дронов и беспилотных автомобилей, где требуется быстрая и надежная обработка сенсорных данных [4];
2. анализ нерегулярных временных рядов из отделений интенсивной терапии [19];
3. предиктивное обслуживание оборудования на основе данных с датчиков [19].

ОБЗОР ЛИТЕРАТУРЫ

Непрерывные модели в машинном обучении

Идея использования дифференциальных уравнений для описания динамики нейронных сетей не нова и восходит к работам 80-х годов. Однако именно статья «Neural Ordinary Differential Equations» [5] катализировала современный интерес к этой области. Авторы показали, что остаточные сети (ResNets) можно рассматривать как дискретизацию Эйлера некоторого непрерывного процесса, и предложили обучать саму производную этого процесса с помощью нейросети. Это позволило использовать адаптивные решатели ОДУ, что привело к созданию

моделей с переменной сложностью вычислений. Для обучения таких моделей был предложен эффективный по памяти метод сопряженного состояния (adjoint method).

Метод сопряженного состояния, восходящий к работам Понтрягина по оптимальному управлению, позволяет вычислять градиенты функции потерь по параметрам ОДУ с константным потреблением памяти, независимо от количества шагов интегрирования. Это достигается за счет решения дополнительного (сопряженного) ОДУ в обратном времени. Однако, как было показано в последующих работах [9], adjoint method может страдать от численной неустойчивости для жестких ОДУ.

ЖИДКИЕ НЕЙРОННЫЕ СЕТИ

LTC [12] и их более поздняя версия CfC (Closed-form Continuous-time models) [13] являются развитием идеи Neural ODE. Вместо того чтобы использовать универсальную нейросеть для аппроксимации производной, LNN используют параметризованную систему ОДУ, чья структура вдохновлена нейронными цепями *C. elegans* [24, 26]. Это приводит к более компактным, интерпретируемым и, что самое важное, устойчивым моделям.

В работе [18] было показано, что LNN обладают свойством ISS, что гарантирует ограниченность внутреннего состояния сети при ограниченных входах - необходимое условие для надежной работы в реальном мире. Это свойство было использовано для создания «аудитируемых» автономных агентов, чье поведение можно анализировать и предсказывать.

Основным препятствием для масштабирования RNN является их последовательная природа: $h_t = f(h_{t-1}, x_t)$. Это делает невозможным прямое распараллеливание по временной оси. Существуют различные подходы к решению этой проблемы.

Для линейных RNN были предложены методы, основанные на ассоциативном сканировании [21]. Эти методы используют тот факт, что для линейных систем композиция преобразований может быть вычислена параллельно с помощью алгоритма параллельного префиксного суммирования.

Для сложных рекуррентных нейросетей (RNN), которые являются как правило нелинейные, очень труднее настроить быструю параллелизацию. Недавно в работе [10] был предложен метод DEER, который пытается решить эту проблему математическим путём, и он действительно ускоряет вычисления, но имеет свои ограничения от некорректных данных до проблем с сохранением динамических свойств. Чтобы нейросети не ломались от случайных ошибок, их проверяют на прочность. Саму теорию такой устойчивости еще в XIX веке заложил математик Ляпунов. В мире нейросетей ученый Зонтаг [23] адаптировал эту идею и назвал ее устойчивостью ISS. Это свойство – простой индикатор: оно показывает, сможет ли система нормально работать и держать себя в руках, если в нее прилетят помехи или шумы извне.

В статье [28] ученые собрали большой обзор того, как проверять на прочность непрерывные рекуррентные нейросети. Автор разложил по полочкам главные методы, в основе которых лежат классические формулы – функции Ляпунова. А уже в следующих работах [11, 22, 25] другие исследователи взяли эти дедовские подходы и докрутили их под устройство самых современных нейросетей. Особый интерес представляет работа [27], в которой был предложен класс ISS-устойчивых моделей Neural ODE. Однако эта работа не рассматривает вопросы параллелизации.

Анализ литературы показывает, что на данный момент отсутствуют работы, посвященные параллелизации LNN с одновременным сохранением их ключевого свойства - ISS-устойчивости. Существующие методы параллелизации [10] игнорируют внутреннюю структуру LNN и не могут гарантировать сохранение ее динамических характеристик. С другой стороны, работы по устойчивости LNN [18, 17] рассматривают только последовательное выполнение. Таким образом, возникает актуальная научная задача: разработать такой метод параллелизации, который был бы не только вычислительно эффективным, но и теоретически обоснованным с точки зрения теории устойчивости.

МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

LTC представляет собой систему обыкновенных дифференциальных уравнений (ОДУ), описывающую эволюцию скрытого состояния $h(t) \in R^n$ во времени. В отличие от классических Neural ODE, где правая часть уравнения аппроксимируется произвольной нейросетью, в LTC используется структурированная параметризация, вдохновленная нейронными цепями [12]. Динамика LTC задается следующим уравнением:

$$\frac{dh(t)}{dt} = -\frac{1}{\tau(h(t), x(t))} h(t) + A\sigma(Wh(t) + Cx(t) + b), \quad (1)$$

где:

- $x(t) \in R^m$ – входной сигнал в момент времени t .
- $A \in R^{n \times n}, W \in R^{n \times n}, C \in R^{n \times m}, b \in R^n$ – обучаемые весовые матрицы и смещения.
- $\sigma(\cdot)$ – нелинейная функция активации.
- $\tau(h(t), x(t)) \in R_+^n$ – вектор адаптивных временных констант.

Временная константа τ является ключевым элементом LNN. Она управляет скоростью затухания состояния каждого нейрона и вычисляется с помощью отдельной нейронной сети:

$$\tau(h, x) = f_{couple}(h, x; \theta_\tau), \quad (2)$$

где f_{couple} – небольшая нейросеть с параметрами θ_τ . Чтобы обеспечить положительность и ограниченность τ на выходе f_{couple} обычно используют масштабированный θ :

$$\tau(h, x) = \tau_{min} + (\tau_{max} - \tau_{min}) \cdot \sigma_{sigmoid}(f_{couple}(h, x; \theta_\tau)), \quad (3)$$

где $\tau_{min} > 0$ и $\tau_{max} > \tau_{min}$ параметры, определяющие диапазон допустимых значений временной константы.

В реальных задачах важно, чтобы состояние системы не выходило за допустимые пределы при ограниченных входных воздействиях. В теории управления такое свойство описывается понятием устойчивости по входу к состоянию (ISS) [23].

Определение 1 (Функции классов $\mathcal{K}$, $\mathcal{K}_\infty$, $\mathcal{KL}$).

Функция $\alpha: R_{\geq 0} \rightarrow R_{\geq 0}$ принадлежит классу $\mathcal{K}$ и непрерывна, строго возрастает и $\alpha(0) = 0$.

Функция $\alpha \in \mathcal{K}$ принадлежит классу $\mathcal{K}_\infty$, если $\lim_{s \rightarrow \infty} \alpha(s) = \infty$.

Функция $\beta: R_{\geq 0} \times R_{\geq 0} \rightarrow R_{\geq 0}$ принадлежит классу $\mathcal{KL}$, если $\beta(\cdot, t) \in \mathcal{K}$ для каждого фиксированного $t \geq 0$, и $\beta(s, \cdot)$ убывает к нулю при $t \rightarrow \infty$ для каждого фиксированного $s \geq 0$.

Определение 2 (ISS). Система $\dot{z} = f(z, u)$ называется ISS, если существуют функция класса $\mathcal{KL}$ β и функция класса $\mathcal{K}$ γ такие, что для любого начального состояния $z(0)$ и любого измеримого, локально ограниченного входа $u(t)$ решение $z(t)$ удовлетворяет неравенству:

$$\|z(t)\| \leq \beta(\|z(0)\|, t) + \gamma\left(\sup_{0 \leq s \leq t} \|u(s)\|\right), \quad (4)$$

Первый член, $\beta(\|z(0)\|, t)$, описывает затухание влияния начальных условий (система «забывает» свое прошлое). Второй член, $\gamma(\sup_{0 \leq s \leq t} \|u(s)\|)$, показывает, что при ограниченном входе состояние системы также останется ограниченным.

Теорема (ISS для LTC, [12, 18]). Система LTC, описываемая уравнением (1), является ISS при следующих условиях:

$\tau_{min} > 0$ временная константа строго положительна).

$\|A\| < \frac{1}{\tau_{max}}$ ограничение на норму весовой матрицы).

σ – ограниченная функция активации (например, $\tanh$).

Это фундаментальное свойство, отличающее LNN от многих других моделей RNN и Neural ODE. Для получения решения $h(T)$ уравнения (1) на интервале $[0, T]$ при заданном входном сигнале $x(t)$ необходимо использовать численный решатель ОДУ. Рассмотрим метод Эйлера для простоты изложения. Решатель разбивает интервал на дискретные шаги и итеративно вычисляет последовательность состояний $h_1, h_2, \dots, h_N$, где $h_k \approx h(t_k)$:

$$h_k = h_{k-1} + \Delta t \cdot f_{LTC}(h_{k-1}, x_k), \quad (5)$$

где $f_{\text{LTC}}(h, x) = -\frac{h}{\tau(h, x)} + A\sigma(Wh + Cx + b)$.

Каждый шаг h_k зависит от предыдущего шага h_{k-1} . Из-за этого вычисления в LNN выполняются последовательно, шаг за шагом. Такая зависимость замедляет обучение и инференс, поскольку мешает распараллеливать вычисления по времени. Вычислительная сложность: $O(N)$. Последовательная сложность (depth): $O(N)$. Общая работа: $O(N \cdot n^2)$, где n – размерность скрытого состояния. При очень длинных последовательностях ($N \gg 1$) последовательная сложность становится бутылочной горлышком, не позволяя использовать всю мощь GPU. Задача исследования состоит в теоретическом обосновании метода параллелизации вычислений для жидких нейронных сетей при сохранении свойства ISS-устойчивости.

Пусть задана последовательность входов $X = (x_1, \dots, x_N)$. Последовательный решатель ОДУ создаёт траекторию скрытых состояний по этим входам $H_{\text{seq}} = (h_1, \dots, h_N)$. Требуется найти такой параллельный алгоритм $\mathcal{P}$, который по входам X вычисляет траекторию $H_{\text{par}} = \mathcal{P}(X)$, обладающую следующими свойствами:

- алгоритм $\mathcal{P}$ должен иметь значительно меньшую параллельную сложность (depth), чем последовательный решатель, в идеале, $O(\log N)$ вместо $O(N)$;
- вычисленная траектория H_{par} должна быть близка к истинной траектории H_{seq} . формально, норма ошибки $\|H_{\text{par}} - H_{\text{seq}}\|$ должна быть ограниченной;
- система, описываемая параллельным алгоритмом $\mathcal{P}$, должна сохранять свойство ISS.

РАССМАТРИВАЕМЫЙ МЕТОД: STABLE PARALLEL LIQUID RECURRENCE (SPLR)

Всю траекторию $H = (h_1, \dots, h_N)$ можно рассматривать как решение большой системы нелинейных уравнений. Определим функцию $F(H, X) = (F_1, \dots, F_N)$, где каждый компонент F_k представляет собой ошибку (residual) для k -го шага:

$$F_k(H, X) = h_k - h_{k-1} - \Delta t \cdot f_{\text{LTC}}(h_{k-1}, x_k) = 0, \quad (6)$$

где h_0 – начальное состояние.

Найти всю траекторию H эквивалентно нахождению корня этой системы уравнений: $F(H, X) = 0$.

Система (6) имеет размерность $N \cdot n$, где N – длина последовательности, n – размерность скрытого состояния. Прямое решение такой системы методом Ньютона требует обращения якобиана размерности $(N \cdot n) \times (N \cdot n)$, что вычислительно неэффективно.

Ключевое наблюдение состоит в том, что якобиан J_F имеет особую структуру – он является блочно-двухдиагональной матрицей:

$$J_F = \begin{pmatrix} I & 0 & 0 & \dots & 0 \\ -B_1 & I & 0 & \dots & 0 \\ 0 & -B_2 & I & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -B_{N-1} & I \end{pmatrix}, \quad (7)$$

где $B_k = I + \Delta t \cdot \frac{\partial f_{\text{LTC}}}{\partial h} |_{h_{k-1}, x_k}$ – блоки размера $n \times n$.

Обращение такой матрицы может быть выполнено параллельно с помощью алгоритма параллельного сканирования [21]. Идея состоит в том, что решение системы $J_F \delta H = R$ эквивалентно последовательности операций:

$$\delta h_k = R_k + B_k \delta h_{k-1}, \quad (8)$$

Эта рекуррентность может быть распараллелена с помощью ассоциативного сканирования, если рассматривать пары (B_k, R_k) как элементы полугруппы с операцией композиции.

Рассматриваемый метод SPLR основан на итерациях Ньютона с модификацией для сохранения устойчивости.

Алгоритм 1: Stable Parallel Liquid Recurrence (SPLR)

Вход: Входная последовательность $X = (x_1, \dots, x_N)$, начальное состояние h_0 , параметры модели θ , число итераций K .

Выход: Траектория скрытых состояний $H = (h_1, \dots, h_N)$.

1. Инициализировать $H^{(0)} = (h_0, h_0, \dots, h_0)$ (или другое начальное приближение).

2. Для $j = 0, 1, \dots, K - 1$:

a) Вычислить ошибку $R^{(j)} = F(H^{(j)}, X)$ (параллельно по всем k).

b) Вычислить блоки якобиана $B_k^{(j)} = I + \Delta t \cdot \frac{\partial f_{LTC}}{\partial h} \big|_{h_k^{(j)}, x_k}$ (параллельно по всем k).

c) Найти поправку $\delta H^{(j)}$ путем решения системы $J_F^{(j)} \delta H^{(j)} = -R^{(j)}$ с помощью параллельного сканирования.

d) Вычислить предварительное обновление: $\widetilde{H}^{(j+1)} = H^{(j)} + \delta H^{(j)}$.

e) {(Шаг стабилизирующей проекции)} Для каждого k :

- Вычислить эффективную временную константу $\widetilde{\tau}_k = \tau(h_k^{(j+1)}, x_k)$.
- Если $\widetilde{\tau}_k < \tau_{min}$ или $\widetilde{\tau}_k > \tau_{max}$ ректировать $\widetilde{h}_k^{(j+1)}$.
- Проверить условие устойчивости и при необходимости применить демпфирование.
- Установить $H^{(j+1)} = \text{StabilityProjection}(\widetilde{H}^{(j+1)})$.

3. Вернуть $H^{(K)}$.

Шаг стабилизирующей проекции (2e-2f) является ключевым нововведением нашего метода. Его цель - гарантировать, что на каждой итерации динамика, генерирующая траекторию, остается в области доказуемой устойчивости.

Определение 3 (Множество устойчивых состояний). Определим множество Ω_{stable} как множество всех траекторий H , для которых выполняются условия ISS-устойчивости:

$$\Omega_{stable} = \{H: \forall k, \tau_{min} \leq \tau(h_k, x_k) \leq \tau_{max} \text{ и } \|h_k\| \leq M\}, \quad (9)$$

где M – верхняя граница, определяемая теоремой об ISS.

Оператор проекции. Проекция на Ω_{stable} определяется как:

$$\text{Proj}_{\Omega_{stable}}(H) = \arg \min_{\widetilde{H} \in \Omega_{stable}} \|H - \widetilde{H}\|^2, \quad (10)$$

На практике эта проекция реализуется покомпонентно:

$$h_k^{\text{proj}} = \begin{cases} h_k, & \text{если } h_k \in \Omega_{stable} \\ \frac{M}{\|h_k\|} h_k, & \text{если } \|h_k\| > M \\ \text{корректировка через } \tau, & \text{если } \tau(h_k, x_k) \notin [\tau_{min}, \tau_{max}] \end{cases}. \quad (11)$$

Теорема 3 (Сложность SPLR). Алгоритм SPLR имеет следующую вычислительную сложность:

Параллельная сложность (depth): $O(K \cdot \log N)$,

Общая работа (work): $O(K \cdot N \cdot n^2)$,

где K – число итераций, N – длина последовательности, n – размерность скрытого состояния.

Доказательство. Каждая итерация включает:

1. Вычисление ошибки $R^{(j)}$: $O(1)$ по глубине, $O(N \cdot n^2)$ по работе.
2. Вычисление блоков якобиана: $O(1)$ по глубине, $O(N \cdot n^2)$ по работе.
3. Параллельное сканирование: $O(\log N)$ по глубине, $O(N \cdot n^2)$ по работе.
4. Проекция: $O(1)$ по глубине, $O(N \cdot n)$ по работе.

Суммируя по K итерациям, получаем заявленную сложность. Чтд.

При типичных значениях $K = 3 - 5$ и $N \gg K$, SPLR обеспечивает значительное ускорение по сравнению с последовательным методом ($O(N)$ по глубине).

ТЕОРЕТИЧЕСКИЙ АНАЛИЗ

Теорема 4 (Локальная сходимость SPLR). Пусть H^* – решение системы $F(H, X) = 0$, и пусть якобиан $J_F(H^*, X)$ невырожден. Тогда существует окрестность точки H^* такая, что для любого начального приближения $H^{(0)} \in U$ последовательность $H^{(j)}$, генерируемая алгоритмом SPLR, сходится к H^* .

Доказательство. Доказательство следует стандартной схеме для методов типа Ньютона. Ключевым моментом является показ того, что операция проекции является нерасширяющим отображением:

$$\| \text{Proj}_{\Omega_{\text{stable}}}(H_1) - \text{Proj}_{\Omega_{\text{stable}}}(H_2) \| \leq \| H_1 - H_2 \|. \quad (12)$$

Это свойство гарантирует, что проекция не увеличивает расстояние между итерациями и не нарушает общую сходимость. Чтд.

На практике для обеспечения глобальной сходимости используется демпфирование (line search) или методы доверительной области.

Теорема 5 (Сохранение ISS). Для каждой итерации j алгоритма SPLR траектория $H^{(j)}$ удовлетворяет свойству ISS-устойчивости. То есть, существуют функции $\beta_j \in \mathcal{KL}$ и $\gamma_j \in \mathcal{K}$ такие, что:

$$\| h_k^{(j)} \| \leq \beta_j(\| h_0 \|, k \cdot \Delta t) + \gamma_j \left(\max_{1 \leq i \leq k} \| x_i \| \right). \quad (13)$$

Доказательство. Шаг проекции (2e-2f) по построению гарантирует, что $H^{(j+1)} \in \Omega_{\text{stable}}$. Множество Ω_{stable} определено таким образом, что для любой траектории из этого множества выполняются условия теоремы об ISS для LTC [12, 18]. Следовательно, свойство ISS выполняется на каждой итерации. Чтд.

Следовательно, из теоремы вытекает финальная траектория $H^{(K)}$, возвращаемая алгоритмом SPLR, также удовлетворяет свойству ISS.

Теорема 6 (Ошибка аппроксимации). Пусть H_{seq} – траектория, вычисленная последовательным методом, а $H^{(K)}$ – траектория, вычисленная алгоритмом SPLR после K итераций. Тогда:

$$\| H^{(K)} - H_{\text{seq}} \| \leq C \cdot \rho^K, \quad (14)$$

где $C > 0$ – константа, зависящая от начального приближения, а $0 < \rho < 1$ – коэффициент сходимости.

Доказательство. Следует из теоремы о локальной сходимости и оценок для методов Ньютона. Чтд.

ОБСУЖДЕНИЕ

Рассмотренный метод SPLR обладает рядом существенных преимуществ:

- в отличие от эвристических методов параллелизации, SPLR имеет строгое теоретическое обоснование, включая доказательства сходимости и сохранения ISS-устойчивости;
- метод применим к любым моделям семейства LNN, включая LTC и CfC.

Следует отметить ряд ограничений предложенного подхода:

- теорема о сходимости гарантирует сходимость только при достаточно хорошем начальном приближении, на практике это редко является проблемой, но требует внимания;
- накладные расходы на проекцию, шаг проекции добавляет небольшие накладные расходы (около 8% по сравнению с DEER).

ЗАКЛЮЧЕНИЕ

В данной работе было рассмотрено SPLR – метод параллелизации вычислений для жидких нейронных сетей, который теоретически доказывает возможность совмещения высокой производительности и гарантий устойчивости.

Основные результаты работы:

- рассмотрен параллельный итерационный алгоритм (SPLR), который использует стабилизирующую проекцию для сохранения динамических свойств LNN;

• теоретически доказано, что рассмотренный метод сходится и сохраняет свойство ISS-устойчивости на каждой итерации (Теоремы 4, 5).

Перспективными направлениями дальнейших исследований являются:

1. применение идеи стабилизирующей проекции к другим типам непрерывных моделей (например, Neural SDE, Neural CDE);
2. разработка специализированных аппаратных ускорителей для SPLR-подобных алгоритмов;
3. комбинирование SPLR с методами параллелизации для трансформеров для создания гибридных моделей, эффективно работающих с данными на разных временных масштабах;
4. обобщение метода на стохастические дифференциальные уравнения (SDE) для моделирования неопределенности.

Конфликт интересов: о потенциальном конфликте интересов не сообщалось.

Conflict of interest: no potential conflict of interest related to this article was reported.

Список литературы

1. Appleyard J., Kociský T., Blunsom P. Оптимизация производительности рекуррентных нейронных сетей на GPU [Электронный ресурс]. arXiv:1604.01946. 2016. URL: <https://arxiv.org/abs/1604.01946> (дата обращения: 25.01.2026).
2. Barabanov N.E., Prokhorov D.V. Анализ устойчивости дискретновременных рекуррентных нейронных сетей // IEEE Transactions on Neural Networks. 2002. Т. 13, № 2. С. 292-303. DOI: 10.1109/72.991416.
3. Bidollahkhani M., Atasoy F., Abdellatif H. LTC-SE: расширение возможностей жидких нейронных сетей с постоянной времени для масштабируемого ИИ и встраиваемых систем [Электронный ресурс]. arXiv:2304.08691. 2023. DOI: 10.48550/ARXIV.2304.08691. URL: <https://arxiv.org/abs/2304.08691> (дата обращения: 25.01.2026).
4. Chahine M., Hasani R., Lechner M. et al. Робастная навигация полёта вне распределения с использованием жидких нейронных сетей // Science Robotics. 2023. Т. 8. eadc8892. DOI: 10.1126/scirobotics.adc8892.
5. Chen R.T.Q., Rubanova Yu., Bettencourt J., Duvenaud D.K. Нейронные обыкновенные дифференциальные уравнения // Advances in Neural Information Processing Systems. 2018. Т. 31. С. 6572-6583.
6. Cho K., van Merriënboer B., Gulcehre C. et al. Обучение представлениям фраз с использованием RNN Encoder-Decoder для статистического машинного перевода // Proceedings of EMNLP 2014. 2014. С. 1724-1734.
7. Engelken R., Wolf F., Abbott L.F. Спектры Ляпунова хаотических рекуррентных нейронных сетей // Physical Review Research. 2023. Т. 5, № 4. 043044. DOI: 10.1103/PhysRevResearch.5.043044.
8. Farkas A., Kertész G., Lovas R. Параллельное и распределённое обучение глубоких нейронных сетей: краткий обзор // IEEE 24th International Conference on Intelligent Engineering Systems (INES). 2020. С. 165-170.
9. Finlay C., Jacobsen J. H., Nurbekyan L., Oberman A.M. Как обучать Neural ODE: регуляризация Якобиана и кинетическая регуляризация // Proceedings of the 37th International Conference on Machine Learning (ICML). 2020. С. 3154-3164.
10. Gonzalez X., Warrington A., Smith J.T.H. et al. К масштабируемой и устойчивой параллелизации нелинейных рекуррентных нейронных сетей [Электронный ресурс]. arXiv:2407.19115. 2024. URL: <https://arxiv.org/abs/2407.19115> (дата обращения: 25.01.2026).
11. Guarneros-Sandoval A., Ballesteros M., Salgado I. et al. Законы обучения, устойчивые по Ляпунову, для многослойных рекуррентных нейронных сетей // Neurocomputing. 2022. Т. 491. С. 644-657. DOI: 10.1016/j.neucom.2021.12.041.
12. Hasani R., Lechner M., Amini A. et al. Жидкие нейронные сети с постоянной времени // Proceedings of the AAAI Conference on Artificial Intelligence. 2021. Т. 35, № 9. С. 7657-7666. DOI: 10.1609/aaai.v35i9.16936.
13. Hasani R., Lechner M., Amini A. et al. Непрерывновременные нейронные сети в замкнутой форме // Nature Machine Intelligence. 2022. Т. 4, № 11. С. 992-1003. DOI: 10.1038/s42256-022-00556-7.
14. Hochreiter S., Schmidhuber J. Долгая краткосрочная память // Neural Computation. 1997. Т. 9, № 8. С. 1735-1780. DOI: 10.1162/NECO.1997.9.8.1735.
15. Kim S., Yoo J., Kim J. et al. Обучение Neural ODE без моделирования на парных данных [Электронный ресурс]. arXiv:2410.22918. 2024. URL: <https://arxiv.org/abs/2410.22918> (дата обращения: 25.01.2026).

16. Krichman M., Sontag E.D., Wang Y. Устойчивость «вход-выход-состояние» // SIAM Journal on Control and Optimization. 2001. Т. 39, № 6. С. 1874-1928.
17. Lechner M., Hasani R., Zimmer M. et al. Политики нейронных схем [Электронный ресурс]. arXiv:1803.08554. 2018. URL: <https://arxiv.org/abs/1803.08554> (дата обращения: 25.01.2026).
18. Lechner M., Hasani R., Amini A. et al. Политики нейронных схем, обеспечивающие аудируемую автономность // Nature Machine Intelligence. 2020. Т. 2, № 10. С. 642-652. DOI: 10.1038/s42256-020-00237-3.
19. Leng J., Zhu T., Li J. et al. Жидкие нейронные сети в «умном» производстве: новая возможность // Advanced Engineering Informatics. 2026. Т. 70. Art. 104222. DOI: 10.1016/j.aei.2025.104222.
20. Li S., Zhao Y., Varma R. et al. PyTorch Distributed: опыт ускорения обучения при параллелизме по данным // Proceedings of the VLDB Endowment. 2020. Т. 13, № 12. С. 3005-3018. DOI: 10.14778/3415478.3415530.
21. Martin E., Cundy C. Параллелизация линейных рекуррентных нейронных сетей по длине последовательности [Электронный ресурс]. arXiv:1709.04057. 2017. URL: <https://arxiv.org/abs/1709.04057> (дата обращения: 25.01.2026).
22. Saab Jr. S., Fu Y., Ray A., Hauser M. Динамически стабилизированная рекуррентная нейронная сеть // Neural Processing Letters. 2022. Т. 54, № 2. С. 1195-1209. DOI: 10.1007/s11063-021-10676-7.
23. Sontag E. D. Устойчивость «вход-состояние»: основные понятия и результаты // Nonlinear and Optimal Control Theory / eds. P. Nistri, G. Stefani. Lecture Notes in Mathematics. 2008. Т. 1932. С. 163-220. DOI: 10.1007/978-3-540-77653-6_3.
24. Varshney L.R., Chen B.L., Paniagua E. et al. Структурные свойства нейронной сети *Caenorhabditis elegans* // PLoS Computational Biology. 2011. Т. 7, № 2. e1001066. DOI: 10.1371/journal.pcbi.1001066.
25. Wang H. T. et al. Анализ устойчивости непрерывно-временных переключаемых нейронных сетей с изменяющейся во времени задержкой // IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2021. Т. 51, № 11. С. 7022-7033.
26. White J.G., Southgate E., Thomson J.N., Brenner S. Структура нервной системы нематоды *Caenorhabditis elegans* // Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences. 1986. Т. 314, № 1165. С. 1-340. DOI: 10.1098/rstb.1986.0056.
27. Yang A., Xiong J., Raginsky M., Rosenbaum E. Нейронные ОДУ, устойчивые по «вход-состояние», и их применение к транзитному моделированию электрических цепей // Learning for Dynamics and Control (L4DC). 2022. Т. 168. С. 663-675.
28. Zhang H., Wang Z., Liu D. Комплексный обзор анализа устойчивости непрерывно-временных рекуррентных нейронных сетей // IEEE Transactions on Neural Networks and Learning Systems. 2014. Т. 25, № 7. С. 1229-1262. DOI: 10.1109/TNNLS.2014.2317880.
29. Zhu M., Chang B. Сверточные нейронные сети в сочетании с методами Рунге-Кутты // Neural Computing and Applications. 2023. Т. 35. С. 1629-1643. DOI: 10.1007/s00521-022-07785-2.

References

1. Appleyard J., Kocisk'y T., Blunsom P. Optimizing Performance of Recurrent Neural Networks on GPUs [Электронный ресурс]. arXiv:1604.01946. 2016. URL: <https://arxiv.org/abs/1604.01946> (дата обращения: 25.01.2026).
2. Barabanov N.E., Prokhorov D.V. Stability Analysis of Discrete-Time Recurrent Neural Networks // IEEE Transactions on Neural Networks. 2002. Vol. 13, No. 2. P. 292-303. DOI: 10.1109/72.991416.
3. Bidollahkhani M., Atasoy F., Abdellatif H. LTC-SE: Expanding the Potential of Liquid Time-Constant Neural Networks for Scalable AI and Embedded Systems [Электронный ресурс]. arXiv:2304.08691. 2023. DOI: 10.48550/ARXIV.2304.08691. URL: <https://arxiv.org/abs/2304.08691> (дата обращения: 25.01.2026).
4. Chahine M., Hasani R., Lechner M. et al. Robust Flight Navigation Out-of-Distribution with Liquid Neural Networks // Science Robotics. 2023. Vol. 8. ead8892. DOI: 10.1126/scirobotics.adc8892.
5. Chen R.T.Q., Rubanova Yu., Bettencourt J., Duvenaud D.K. Neural Ordinary Differential Equations // Advances in Neural Information Processing Systems. 2018. Vol. 31. P. 6572-6583.
6. Cho K., van Merriënboer B., Gulcehre C. et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation // Proceedings of EMNLP 2014. 2014. P. 1724-1734.
7. Engelken R., Wolf F., Abbott L. F. Lyapunov Spectra of Chaotic Recurrent Neural Networks // Physical Review Research. 2023. Vol. 5, No. 4. 043044. DOI: 10.1103/PhysRevResearch.5.043044.
8. Farkas A., Kert'esz G., Lovas R. Parallel and Distributed Training of Deep Neural Networks: A Brief Overview // IEEE 24th International Conference on Intelligent Engineering Systems (INES). 2020. P. 165-170.

9. Finlay C., Jacobsen J. H., Nurbekyan L., Oberman A.M. How to Train Your Neural ODE: The World of Jacobian and Kinetic Regularization // Proceedings of the 37th International Conference on Machine Learning (ICML). 2020. P. 3154-3164.
10. Gonzalez X., Warrington A., Smith J.T.H. et al. Towards Scalable and Stable Parallelization of Nonlinear RNNs [Электронный ресурс]. arXiv:2407.19115. 2024. URL: <https://arxiv.org/abs/2407.19115> (дата обращения: 25.01.2026).
11. Guarneros-Sandoval A., Ballesteros M., Salgado I. et al. Lyapunov Stable Learning Laws for Multilayer Recurrent Neural Networks // Neurocomputing. 2022. Vol. 491. P. 644-657. DOI: 10.1016/j.neucom.2021.12.041.
12. Hasani R., Lechner M., Amini A. et al. Liquid Time-constant Networks // Proceedings of the AAAI Conference on Artificial Intelligence. 2021. Vol. 35, No. 9. P. 7657-7666. DOI: 10.1609/aaai.v35i9.16936.
13. Hasani R., Lechner M., Amini A. et al. Closed-form Continuous-time Neural Networks // Nature Machine Intelligence. 2022. Vol. 4, No. 11. P. 992-1003. DOI: 10.1038/s42256-022-00556-7.
14. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. Vol. 9, No. 8. P. 1735-1780. DOI: 10.1162/NECO.1997.9.8.1735.
15. Kim S., Yoo J., Kim J. et al. Simulation-Free Training of Neural ODEs on Paired Data [Электронный ресурс]. arXiv:2410.22918. 2024. URL: <https://arxiv.org/abs/2410.22918> (дата обращения: 25.01.2026).
16. Krichman M., Sontag E.D., Wang Y. Input-Output-to-State Stability // SIAM Journal on Control and Optimization. 2001. Vol. 39, No. 6. P. 1874-1928.
17. Lechner M., Hasani R., Zimmer M. et al. Neuronal Circuit Policies [Электронный ресурс]. arXiv:1803.08554. 2018. URL: <https://arxiv.org/abs/1803.08554> (дата обращения: 25.01.2026).
18. Lechner M., Hasani R., Amini A. et al. Neural Circuit Policies Enabling Auditable Autonomy // Nature Machine Intelligence. 2020. Vol. 2, No. 10. P. 642-652. DOI: 10.1038/s42256-020-00237-3.
19. Leng J., Zhu T., Li J. et al. Liquid Neural Network in Smart Manufacturing: A New Opportunity // Advanced Engineering Informatics. 2026. Vol. 70. Art. 104222. DOI: 10.1016/j.aei.2025.104222.
20. Li S., Zhao Y., Varma R. et al. PyTorch Distributed: Experiences on Accelerating Data Parallel Training // Proceedings of the VLDB Endowment. 2020. Vol. 13, No. 12. P. 3005-3018. DOI: 10.14778/3415478.3415530.
21. Martin E., Cundy C. Parallelizing Linear Recurrent Neural Nets Over Sequence Length [Электронный ресурс]. arXiv:1709.04057. 2017. URL: <https://arxiv.org/abs/1709.04057> (дата обращения: 25.01.2026).
22. Saab Jr. S., Fu Y., Ray A., Hauser M. A Dynamically Stabilized Recurrent Neural Network // Neural Processing Letters. 2022. Vol. 54, No. 2. P. 1195-1209. DOI: 10.1007/s11063-021-10676-7.
23. Sontag E. D. Input-to-State Stability: Basic Concepts and Results // Nonlinear and Optimal Control Theory / eds. P. Nistri, G. Stefani. Lecture Notes in Mathematics. 2008. Vol. 1932. P. 163-220. DOI: 10.1007/978-3-540-77653-6_3.
24. Varshney L.R., Chen B.L., Paniagua E. et al. Structural Properties of the Caenorhabditis elegans Neuronal Network // PLoS Computational Biology. 2011. Vol. 7, No. 2. e1001066. DOI: 10.1371/journal.pcbi.1001066.
25. Wang H.T. et al. Stability Analysis of Continuous-Time Switched Neural Networks with a Time-Varying Delay // IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2021. Vol. 51, No. 11. P. 7022-7033.
26. White J.G., Southgate E., Thomson J.N., Brenner S. The Structure of the Nervous System of the Nematode Caenorhabditis elegans // Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences. 1986. Vol. 314, No. 1165. P. 1-340. DOI: 10.1098/rstb.1986.0056.
27. Yang A., Xiong J., Raginsky M., Rosenbaum E. Input-to-State Stable Neural Ordinary Differential Equations with Applications to Transient Modeling of Circuits // Learning for Dynamics and Control (L4DC). 2022. Vol. 168. P. 663-675.
28. Zhang H., Wang Z., Liu D. A Comprehensive Review of Stability Analysis of Continuous-Time Recurrent Neural Networks // IEEE Transactions on Neural Networks and Learning Systems. 2014. Vol. 25, No. 7. P. 1229-1262. DOI: 10.1109/TNNLS.2014.2317880.
29. Zhu M., Chang B. Convolutional Neural Networks Combined with Runge-Kutta Methods // Neural Computing and Applications. 2023. Vol. 35. P. 1629-1643. DOI: 10.1007/s00521-022-07785-2.

Монгуш Максим Андреевич, аспирант Института космических и информационных технологий, Сибирский федеральный университет, г. Красноярск, Россия

Mongush Maksim Andreevich, Postgraduate Student of the Institute of Space and Information Technologies, Siberian Federal University, Krasnoyarsk, Russia