Создание искусственных нейронных сетей для работы в распределенных вычислительных средах

Николай Вершков [1], Наталья Кучукова [1] и Владислав Луценко [1]

[1] *Северо-Кавказский федеральный университет, ул. Пушкина, 1, г. Ставрополь, 355017, Россия*

Аннотация

В статье рассматриваются аспекты обучения и применения по назначению модульных искусственных нейронных сетей с целью их использования в распределенных вычислительных средах. Для достижения поставленной цели предлагается метод проектирования входной информации на пространство ортогональных полиномов с целью получения подпространств, имеющих меньшую размерность и позволяющих осуществлять раздельную обработку с помощью модулей, расположенных на разных вычислительных узлах распределенной среды. Предлагаемый подход имеет преимущество перед методом послойного разделения в объемах информации, передаваемой между модулями и в гибкости вычислений, позволяющей оптимизировать соотношение «объем вычислительных ресурсов/качество».

Ключевые слова

Искусственные нейронные сети, оптимизация нейронных сетей, ортогональные преобразования, модульные нейронные сети, модульное обучение, распределенные вычисления.

1. Введение

Распределенные вычисления становятся все более важной частью современных интеллектуальных технических систем таких как туманные и периферийные вычисления интернета вещей, управление группой беспилотных устройств, группой спутников на орбите и т.п. При этом все больше задач таких как компьютерное зрение и распознавание объектов, работающих в контуре управления объектами, требуют использования искусственных нейронных сетей (ИНС). Высокоточные модели ИНС имеют значительные объемы и требуют больших вычислительных ресурсов. При наличии вычислительных узлов высокой мощности проблем с использованием ИНС нет, вычислительных ресурсов достаточно. Однако смещение акцента на распределенные вычисления часто требует размещения ИНС на узлах, вычислительная мощность которых ограничена. В связи с этим для поддержания модели высокоточной ИНС требуется использовать несколько вычислительных узлов одновременно.

Для решения подобной проблемы наиболее часто используют модели распределенного программирования, например, MapReduce [1]. Такой подход позволяет разделить ИНС на слои (по тем или иным признакам) и выполнять один или несколько слоев на отдельном узле распределенных вычислений [2, 3]. Значительным недостатком этого подхода является большая нагрузка на каналы передачи данных, т.к. объем информации, передаваемый между слоями, очень велик. Другим, также значительным недостатком, является отсутствие гибкости, т.е. приспосабливаемости ИНС, к неопределенностям в наличии необходимых вычислительных ресурсов. Например, в среде туманных вычислений нет достаточного количества свободных узлов, поломка или потеря нескольких беспилотных устройств в группе может привести к отсутствию возможности использования ИНС. В этом случае потребуется

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: vernick61@yandex.ru (A. 1); nkuchukova@ncfu.ru (A. 2); vvlutcenko@ncfu.ru (A. 3)
ORCID: 0000-0001-5756-7612 (A. 1); 0000-0002-8070-0829 (A. 2); 0000-0003-4648-8286 (A. 3)

ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.07

резервирование вычислительных мощностей, что, в свою очередь, приводит к неэффективному использованию вычислительных узлов распределенной среды. Предлагаемый метод обладает свойством эластичности, т.е. при отсутствии свободных вычислительных узлов ИНС в состоянии выполнять свои функции с помощью такого количества модулей, сколько свободных вычислительных узлов имеется в наличии, хотя и с несколько низким качеством, при этом не требует переобучения. При освобождении вычислительных мощностей в распределенной среде количество модулей может увеличиваться, повышая качество работы ИНС.

2. Реализация модульных ИНС

Теоретической основой ИНС считается доказательство теоремы Колмогорова-Арнольда, которая показала возможность представления непрерывной действительной функции n переменных $f(x_1, x_2, \dots, x_n)$ в виде суперпозиции функций меньшего числа переменных.

В своей работе А.Н. Горбань делает вывод [4] о том, что несмотря на то, что теорема Колмогорова-Арнольда говорит о точном представлении функций многих переменных в классе непрерывных функций, практическое вычисление большинства функций производится приближенно даже при наличии точных формул. А подобное решение уже лежит в плоскости приближения функции $f(x_1, x_2, \dots, x_n)$ на замкнутом ограниченном множестве Q последовательностью полиномов (теоремы Вейерштрасса, Стоуна).

Из обобщения Горбанем теоремы Стоуна следует, что любая нелинейность обладает универсальными аппроксимационными свойствами, т.е. из многочлена p степени $m > 1$ и многочленов первой степени с помощью операций суперпозиции, сложения и умножения на число можно получить все элементы пространства $\boldsymbol{C}[x_1, x_2, \dots, x_n]$. Таким образом, если существуют ИНС S_1, S_2,…, S_k, которые вычисляют функции y_1, y_2, …, y_K от вектора входных сигналов x, то с помощью оператора $f(w_0 + w_1y_1 + \cdots + w_ky_k)$ можно получить результат аппроксимации с заданной точностью. В соответствии с теоремой Стоуна-Горбаня множество функций, вычислимых нейронными сетями с заданной непрерывной нелинейной характеристической функцией, плотно в пространстве непрерывных функций от входных сигналов [4].

Приближенное решение исходной задачи можно получить в виде разложения x по некоторому множеству базисных элементов $\{u_i(t)\}_{i=1}^{k}$

$$x_k = \sum_{i=1}^{k} a_i u_i(t) \quad (1)$$

Базисные функции в (2) могут быть заданы аналитически, алгоритмически или численно. Одними из первых, кто предложил использовать ортогональные преобразования для выделения признаков, являются Н. Ахмед и К.Р. Рао [5]. Используя стандартную архитектуру ИНС авторы [5] показали, что с помощью, например, дискретного косинусного преобразования (ДКП) можно снизить размерность классификатора изображений, что стало первым шагом в оптимизации ИНС, используя приближенное представление входного сигнала.

2.1. Преобразование Фурье как способ приближенного представления входного сигнала ИНС

Наиболее известным и широко используемым оператором в теории передачи информации является преобразование Фурье [6]. Поскольку входными значениями ИНС являются дискретные значения, то можно использовать дискретное преобразование Фурье (ДПФ):

$$Ax(t) = F(\omega) = \sum_{i=0}^{n} x_i e^{-\frac{2\pi j \omega i}{n}}$$

Здесь $x \in L^1(R)$ – входные значения оператора A, $\omega = 2\pi f$ – круговая частота. Для функций, интегрируемых с квадратом на R, можно определить гильбертово пространство

функций $L^2(R)$, на котором преобразование Фурье может быть определено как оператор $A: L^2(R) \to L^2(R)$.

Коэффициент разложения a_i из выражения (1) представляет собой проекцию функции на i-тую ортогональную функцию выбранного базиса на всем промежутке $[0, T]$. Воспользуемся понятием точности приближения для замкнутых систем ортогональных функций [7]. На основании равенства Парсеваля:

$$B = \sum_{i=0}^{\infty} x_I^2 = \sum_{n=0}^{\infty} a_n^2 \quad (2)$$

Тогда относительную интегральную точность приближения с учетом (4) можно оценить как

$$\gamma = \frac{\sum_{n=0}^{k} a_n^2}{B} = \frac{a_0^2}{B} + \frac{a_1^2}{B} + \cdots + \frac{a_n^2}{B} = \frac{B_1}{B} + \frac{B_2}{B} + \cdots + \frac{B_d}{B} \quad (3)$$

Из свойств преобразования Фурье известно, что чем выше степень гладкости функции $x(t)$, тем быстрее убывает ее преобразование Фурье $F(\omega)$. Следовательно, частичные суммы из ряда (2) будут убывать. Тогда интегральная точность приближения (3) не будет значительно убывать при отбрасывании последних членов ряда, а частичные суммы коэффициентов уменьшаются по мере роста номеров гармоник. Таким образом, разделив спектр на две равные части, получим следующие значения интегральной точности приближения для каждой половины спектра:

$$\gamma_1 = \frac{\sum_{n=0}^{k/2} a_n^2}{B} \quad (4)$$

$$\gamma_2 = \frac{\sum_{n=k/2+1}^{k} a_n^2}{B} \quad (5)$$

Поскольку частичная сумма в выражении (5) будет меньше, чем в выражении (4), то точность приближения ИНС, обученной первой частью спектра будет выше, чем второй. В дальнейшем это будет показано экспериментально. В зависимости от решаемой задачи, количество частей, на которые делится спектр входной функции, может быть различным. Кроме того, части спектра не обязательно должны быть одинаковы. Если имеется несколько вычислительных узлов повышенной мощности, для них может быть выделены большая часть спектра с использованием более объемной ИНС.

Преобразование Фурье является эффективным инструментом исследования функций, но имеет существенный недостаток: имея высокую степень частотной локализации обладает очень слабой временной (пространственной) локализацией.

2.2. Использование вейвлет-преобразования для создания модульной структуры входного сигнала ИНС

Если же вместо Фурье-подобных преобразований использовать вейвлет-преобразование [8], то можно совместить операции генерации и отбора признаков и получить выигрыш в количестве значимых признаков без проведения дополнительных исследований [9]. Вейвлеты в общем виде представляют собой систему функций вида

$$\varphi_{a,b}(x) = \sqrt{2^a}\varphi(2^a x - b) \quad (6)$$

Если V_a – пространство, порожденное системой функций (6), то имеют место следующие включения [9] $V_0 \subset V_1 \subset \cdots \subset V_a$. Т.е. получается система вложенных подпространств $V_i \subset L^2(R)$, в каждом из которых выделен ортонормированный базис $\{\varphi_{i,b}(x)\}$. Полученная последовательность подпространств может быть использована для того, чтобы перейти от некоторой функции $f(x)$ из $L^2(R)$ к ее приближению с помощью оператора ортогонального проектирования:

$$P_a: L^2(R) \to V_a, P_a(f) = \sum_{b \in Z} (f, \varphi_{a,b})\varphi_{a,b}(x)$$

Проекции P_i являются приближениями $f(x)$ все более точными с ростом i. Возвращаясь к нейронным сетям, можно провести следующую аналогию. Набор входных векторов $\{f_i(x_j)\}$ в пространстве $L^2(R)$ может быть спроектирован на набор подпространств $V_0 \subset V_1 \subset \cdots \subset V_a$ так,

что каждая проекция P_i является приближением входных данных. Выполнив обучение нейронной сети на проекции P_0 получим максимально грубое приближение ожидаемого результата. Однако, за счет децимации это будет самое «короткое» приближение и для функционирования потребуется минимум вычислительных ресурсов.

Широко известный вейвлет Хаара [5, 10, 11] позволяет разбить пространство $L^2(R)$ на 2 подпространства V_0 и V_1. Используя подпространство V_0 в качестве базового, можно получить модульную архитектуру ИНС, которая позволит для маломощных устройств использовать базовый модуль [11] и, таким образом, получить оптимизированную ИНС для маломощного устройства. Фактически вейвлет Хаара выполняет разбиение пространства коэффициентов на две равные части [10]. Благодаря этому возможно решение задачи, когда базовый модуль позволит решить задачу применения ИНС по назначению на устройствах с невысокой вычислительной мощностью с незначительной потерей точности и без дополнительного обучения. Решение задачи распределенных вычислений требует размещения модулей на узлах вычислительной сети, причем в условиях высокой неопределенности наличия вычислительной мощности изначально может быть использован базовый модуль, а по мере освобождения вычислительных узлов на них могут размещаться дополнительные модули, повышая качество работы ИНС.

При практическом применении предлагаемых теоретических положений нужно учитывать, что подобное преобразование можно выполнить еще раз для каждой половины коэффициентов. В этом случае получится разбить все пространство на 4 части и сформировать 4-хмодульную структуру и двухуровневый вейвлет-спектр, и т.д. Кроме вейвлетов Хаара для решения подобной задачи могут быть использованы вейвлеты Добеши [12], обучение модульных ИНС может вестись одновременно на нескольких уровнях вейвлет-спектра. В зависимости от состава и размера признаков наилучший результат обучения может быть получен на любом уровне вейвлет-спектра.

2.3. Варианты реализации параллельных высокопроизводительных ИНС

Практическая реализация модульных ИНС осуществлялась с помощью библиотеки PyTorch [13] с использованием сета MNIST [14].

При построении модульных ИНС рассматривались 2 основных варианта: 1. в наличии имеется 1 вычислительный узел и на него может быть помещено столько модулей, сколько позволяет его вычислительная мощность; 2. в наличии имеются несколько вычислительных узлов, при этом точное количество узлов и их вычислительные возможности заранее неизвестны.

Задача оптимизации возможностей одного вычислительного узла (вариант 1) решается путем размещения максимально возможного количества модулей с учетом его вычислительных возможностей. В качестве первого примера представляется 4-хмодульная ИНС, построенная с помощью ДКП. ДКП выполняется в 1-м слое прямосвязной ИНС, при этом веса первого слоя установлены с помощью выражения () и запрещено их изменение. Полученные коэффициенты ДКП разделены на 4 равных части и обрабатываются 4-мя отдельными модулями. Модули представляют собой монолитную ИНС, размер которой уменьшен в 4 раза. Результаты обработки всех модулей поступают на вход последнего объединяющего слоя. ИНС была обучена на сете MNIST до достижения качества распознавания 98%.

Таблица 1

Показатели ИНС при подключении нескольких модулей

	1 модуль	2 модуля	3 модуля	4 модуля
Качество распознавания, %	78.8	94	97.2	98
Среднее время 1 цикла, с	0.01	0.02	0.02	0.03

Таблица 1 иллюстрирует качество распознавания цифр при подключении дополнительных модулей. Если в работу включены не все модули, то вместо отсутствующих данных на соответствующий вход объединяющего слоя подается нулевой тензор. Так, 3 модуля дают вполне приемлемый по качеству результат и более эффективное использование вычислительных возможностей узла в плане его быстродействия.

Аналогичный эксперимент был проведен с использованием вейвлет-преобразования Хаара. Первый слой является сверточным, в ядро которого занесены значения фильтра [15]. Значения ядра также зафиксированы – их изменение запрещено. Показатели ИНС с вейвлет-преобразованием приведены в таблице 2.

Таблица 2

Показатели ИНС при подключении нескольких модулей с вейвлет-преобразованием

	1 модуль	2 модуля	3 модуля	4 модуля
Качество распознавания, %	88.1	96.4	97	98.2
Среднее время 1 цикла, с	0.01	0.02	0.02	0.02

Из таблицы 2 видно, что при использовании вейвлет-преобразования качество распознавания модулей несколько выше.

2-й вариант требует несколько иного подхода. Целью проводимого эксперимента было сравнение временных характеристик модульной ИНС с монолитной. Дистрибутивный пакет, включенный в PyTorch [11] позволяет распараллеливать вычисления по процессам и кластерам машины. Для этого он использует семантику передачи сообщений, позволяя каждому процессу передавать данные любому другому процессу. Поддержка параллельных вычислений осуществляется только под управлением Linux-подобных операционных систем. При работе с бэкендом «gloo» вся вычислительная среда разделяется на процессы с возможностью установить необходимое количество потоков в каждом процессе. Т.е. каждый процесс является виртуальной вычислительной средой, в котором выполняется указанный отрезок кода.

Для разделения ИНС на процессы использовался следующий подход:

1. В первом слое выполняется вейвлет-преобразование путем помещения в ядро сверточного слоя фильтра из банка фильтров, полученного с помощью библиотеки PyWavelets [15]. Эксперимент проводился с использованием вейвлета Хаара с размером фильтра 2x2. Вейвлет-преобразование выполняется в отдельном модуле.

2. Полученный результат в виде двух или более последовательностей с размерностью в 2 раза меньшей, чем исходный сигнал поступает на 2 ИНС. Каждая ИНС выполняется в своем процессе размером 16 потоков. Передача данных на входы ИНС и их синхронизация осуществляются сервером данных, который выполняется в отдельном процессе.

3. Результаты обработки модульных ИНС передаются в объединяющий слой, также выполняющийся в отдельном процессе. В процессе обучения используется версия оптимизатора, которая позволяет осуществлять обучение распределенных ИНС.

Временные характеристики монолитной и 2-хмодульной ИНС представлены в таблице 3.

Таблица 3

Временные показатели монолитной и модульной ИНС

	Макс.	Мин.	Среднее
Монолитная ИНС, время выполнения, с	0.536	0.128	0.202
Модульная ИНС, время выполнения, с	1.416	0.065	0.12

Из таблицы 3 видно явное преимущество модульной ИНС по времени выполнения, при том, что у нее на 2 слоя больше (вейвлет-преобразование и объединяющий). Превышение максимального времени выполнения модульной ИНС объясняется тем, что модульная архитектура работает под управлением сервера данных, который управляет запуском и завершением модулей, и передачей данных для обработки.

Таким образом, возможность построения параллельных архитектур для выполнения распределенной обработки данных искусственными нейронными сетями позволяет проектировать высокоточные ИНС на вычислительных узлах в распределенных средах,

создавать модульные структуры, которые превосходят по скорости обработки монолитные структуры, а также имеют возможность адаптироваться к имеющимся свободным вычислительным ресурсам.

3. Благодарности

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00149, https://rscf.ru/project/24-21-00149/.

4. Ссылки

[1] Dean J, Ghemawat S MapReduce: simplified data processing on large clusters. Communications of the ACM 2008; 51(1): 107-113. DOI: 10.1145/1327452.1327492.
[2] Mao J, Chen X, Nixon KW, Krieger C, Chen Y. MoDNN: Local distributed mobile computing system for Deep Neural Network. Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE 2017; 1396-1401. DOI: 10.23919/DATE.2017.7927211.
[3] Bakhtin VV. Algorithm for decomposition of a monolithic neural network for implemented fog computing in devices based on programmable logic. PNRPU Bulletin. Electrical engineering, information technology, control systems 2022; 41: 123-145. DOI: 10.15593/2224-9397/2022.1.06
[4] Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей, Сиб. журн. вычисл. матем., 1:1 (1998), 11–24
[5] Ахмед Н., Рао К.Р. Ортогональные преобразования при обработке цифровых сигналов: Пер. с англ./Под ред. И.Б. Фоменко. – М.: Связь, 1980
[6] А.В. Солодов. Теория информации и ее применение к задачам автоматического управления и контроля. – М.: Издательство «Наука» Главная редакция физико-математической литературы, 1967
[7] Дедус Ф.Ф., Куликова Л.И., Панкратов А.Н., Тетуев Р.К. Классические ортогональные базисы в задачах аналитического описания и обработки информационных сигналов: учеб. пособие к спецкурсу "Обобщ. спектр.-аналит. метод обраб. информ. массивов" // под ред. Ф. Ф. Дедуса; Моск. гос. ун-т им. М. В. Ломоносова, Фак. вычисл. математики и кибернетики (МГУ, ВМК). – М.: Издат. отд. Фак. вычисл. математики и кибернетики МГУ им. М.В. Ломоносова, 2004.
[8] Н. А. Вершков, М. Г. Бабенко, Н. Н. Кучукова, В. А. Кучуков, Н. Н. Кучеров, “Поперечнослойное разделение искусственных нейронных сетей для классификации изображений”, Компьютерная оптика, 48:2 (2024), 312–320.
[9] Смоленцев Н.К. Основы теории вейвлетов. Вейвлеты в MATLAB. – М.: ДМК Пресс, 2019 г.
[10] Haar A. Zur theorie der orthogonalen funktionensysteme. – GeorgAugustUniversitat, Gottingen; 1909.
[11] Vershkov N., Babenko M., Tchernykh A. et al. Optimization of Artificial Neural Networks using Wavelet Transforms. Program Comput Soft 48, 376–384 (2022). https://doi.org/10.1134/S036176882206007X
[12] Добеши И. Десять лекций по вейвлетам. — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001.
[13] PyTorch (англ.) Дата обращения 29 мая 2025.
[14] Qiao, Yu THE MNIST DATABASE of handwritten digits (2007). Accessed 29.05.2025.
[15] https://pypi.org/project/PyWavelets/. Дата обращения 26 февраля 2024.