Поперечнослойное разделение искусственных нейронных сетей для классификации изображений

Н.А. Вершков [1], М.Г. Бабенко [1], Н.Н. Кучукова [1], В.А. Кучуков [1], Н.Н. Кучеров [1]
[1] Северо-Кавказский центр математических исследований, Северо-Кавказский федеральный университет, 355017, Россия, г. Ставрополь, ул. Пушкина, д. 1

Аннотация

В статье рассматриваются задачи модульного обучения искусственных нейронных сетей, а также исследуются возможности частичного использования модулей в условиях ограниченных вычислительных ресурсов. Предлагаемый метод основывается на свойствах вейвлет-преобразования по разделению информации на высокочастотную и низкочастотную части. Используя наработки по вейвлет-преобразованию на основе сверточного слоя, авторы осуществляют поперечнослойное разделение сети на модули для дальнейшего частичного использования их на устройствах с малой вычислительной мощностью. Теоретическое обоснование такого подхода в статье подкрепляется экспериментальным разделением базы MNIST на 2 и 4 модуля и их последовательным использованием с замером точности и производительности. Выигрыш в производительности составил 2 и более раза при использовании отдельных модулей. Также с помощью AlexNet-подобной сети с использованием набора данных GTSRB проверены предложенные теоретические положения, при этом выигрыш производительности одного модуля составил 33 % без потери точности.

Ключевые слова: вейвлет-преобразование, искусственные нейронные сети, сверточный слой, ортогональные преобразования, модульное обучение, оптимизация нейронных сетей.

Цитирование: **Вершков, Н.А**. Поперечнослойное разделение искусственных нейронных сетей для классификации изображений / Н.А. Вершков, М.Г. Бабенко, Н.Н. Кучукова, В.А. Кучуков, Н.Н. Кучеров // Компьютерная оптика. – 2024. – Т. 48, № 2. – С. 312-320. – DOI: 10.18287/2412-6179-CO-1278.

Citation: Vershkov NA, Babenko MG, Kuchukova NN, Kuchukov VA, Kucherov NN. Transverse-layer partitioning of artificial neural networks for image classification. Computer Optics 2024; 48(2): 312-320. DOI: 10.18287/2412-6179-CO-1278.

Введение

Интерес к модульной организации искусственных нейронных сетей (ИНС) значительно возрос в последние годы. Это связано с ростом интереса к ИНС в самых разных областях, в том числе и для решения задач в условиях ограниченных вычислительных ресурсов. В настоящее время стали развиваться периферийные вычисления, позволяющие анализировать и фильтровать данные с использованием конечных устройств, т.е. ближе к пользователю, к источнику данных. Такой подход позволяет построить распределенную систему обработки данных, адаптированную к различным вычислительным устройствам. По прогнозам компании «Juniper Research», к 2025 году 59 % данных, создаваемых устройствами Интернета вещей, будет обрабатываться при помощи технологии периферийных вычислений.

Обеспечение работоспособности нейронных сетей на мобильных и маломощных вычислительных устройствах требует создания программного обеспечения с низким потреблением вычислительных ресурсов и адаптированного к использованию на маломощных устройствах. Для выполнения этого требования возникает необходимость создания быстро обучаемых нейронных сетей, масштабирования обученных сетей, т.е. значительного (в разы) уменьшения объема ИНС при некотором снижении точности. Подобный подход позволит быстро распространять нейронные сети на любую вычислительную платформу без дополнительного обучения.

Одна из первых попыток использовать обобщенный модульный подход к описанию ИНС была сделана в [1]. Однако при исследовании авторы углубились в особенности архитектуры ИНС, которые не позволяют использовать предлагаемый метод без введения определенных правил и ограничений, а это требует дополнительных исследований. В работе [2] автор провел значительный анализ зависимости скорости обучения ИНС от количества содержащихся в ней нейронов, однако упустил требования по декомпозиции нейросети на модули, в связи с чем предлагаемый подход не позволяет получить аппроксимационную оценку для обучения одного модуля. Интересный подход продемонстрировали авторы в [3], однако он предложен для решения очень специфической задачи классификации объектов, состоящих из текста и цветного изображения, причем цвет используется для сосредоточения внимания на определенном объекте. Модульный подход продемонстрирован в [4] с последующим объединением результатов с помощью голосования по Нэшу [5]. Авторы получили

компромиссный подход к использованию вычислительных ресурсов, однако он не позволил получить значительный выигрыш для маломощных устройств.

Таким образом, задача обеспечения модульности нейронных сетей для их применения на мобильных и маломощных вычислительных устройствах остается открытой [8 – 10]. Пусть существует ИНС, обученная для решения определенного класса задач, например, для классификации изображений, предназначенная для работы на сервере или в облачной среде. Возникает необходимость использовать эту ИНС на маломощном устройстве для решения такой же задачи, но при этом вычислительные возможности устройства не позволяют решать задачу распознавания изображения за определенное время. Требуется осуществить трансформацию существующей ИНС таким образом, чтобы она выполняла функциональную задачу за определенное время на устройстве за счет незначительного ухудшения качества распознавания. При этом архитектура исходной ИНС может быть различной: например, прямосвязной или сверточной.

1. Теоретическое обоснование возможности модульного обучения ИНС

Как было показано во введении, попытки разделить нейронную сеть предпринимались, однако авторы подобных решений переходили к анализу архитектуры сети, и полученное решение получалось «однобоким», т.е. только для конкретной архитектуры, или стремились обосновать решение проблемы без практических выводов для получения конкретных результатов.

Авторы [21] склоняются к идее «вертикальной модульности», т.е. послойного разделения ИНС для обучения или применения по назначению на маломощных вычислительных устройствах. Т.е. объем ИНС не изменяется, а нехватка вычислительных ресурсов компенсируется применением нескольких вычислительных устройств, соединенных последовательно, и размещением на них нескольких слоев сети. При этом авторы приводят тезис о невозможности разделения одного слоя на два и более вычислительных устройства [21].

В результате анализа становится очевидным, что универсальное решение для различных видов архитектур ИНС может быть получено только путем внесения модульности в исследуемую информацию, а не в аппаратную или программную архитектуру ИНС. Самый простой способ – разделить входную информацию на блоки. Но учитывая, что количество признаков в каждом модуле и их взаимосвязь (корреляция) с признаками в других модулях априори не известны, такой подход неэффективен [12]. Мы предлагаем следующую модульную архитектуру: один из модулей (который будем называть в дальнейшем базовым) должен содержать максимальное количество признаков (значащей информации) для обучения ИНС, а дополнительные модули можно будет использовать для повышения точности работы нейросети за счет увеличения затрат вычислительных ресурсов (при их наличии). Рассмотрим пример.

Пусть существует набор данных для обучения ИНС, длина каждого значения из набора (вектора) равна n. В [6] показано, что сети шириной $n+4$ с функциями активации ReLU могут аппроксимировать любую интегрируемую функцию Лебега на n-мерном входном пространстве относительно L^2 расстояния, если разрешен рост глубины сети, т.е. для любой интегрируемой функции Бохнера–Лебега $f: \mathbb{R}^n \to \mathbb{R}^m$ и любой $\varepsilon > 0$, существует полносвязная сеть F точной ширины $d_m = \max\{n+1, m\}$, удовлетворяющая условию

$$\int_{\mathbb{R}^n} \left\| f(x) - F(x) \right\|^p \mathrm{d}x < \varepsilon.$$

В [7] доказано, что пространство прямосвязанных нейронных сетей является плотным относительно топологии равномерной сходимости. Исходя их этого можно сделать вывод о том, что необходимо выполнить над входными данными такое преобразование $g: \mathbb{R}^n \to \mathbb{R}^n$, чтобы максимальное количество значимых признаков оказались с одной стороны. Тогда, разделив входные данные на две равные части, получим базовый модуль и дополнительный. Одним из преобразований такого вида является преобразование Фурье, т.к. низкочастотная часть спектра обычно имеет большую амплитуду и, соответственно, содержит большую часть входной информации. Данное утверждение справедливо для любых, необязательно однородных данных [22].

Из свойств преобразования Фурье известно [11], что чем выше степень гладкости функции, тем быстрее убывают коэффициенты преобразования Фурье. Т.е. для гладких функций на входе ИНС достаточно выполнить преобразование Фурье вида

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-jx\omega} \mathrm{d}x, \qquad (1)$$

где $f(x)$ – исследуемая функция, $e^{-jx\omega}$ – ядро преобразования Фурье, ω – круговая частота, и получить в первой половине спектра длиной $n/2$ базовый модуль, а во второй половине – дополнительный модуль такой же длины. Учитывая всё вышесказанное, а также то, что преобразование Фурье является ортогональным преобразованием, т.е. выполняется условие

$$\int u_m(t) u_n(t) \mathrm{d}t = \begin{cases} c, \text{ если } m = n \\ 0, \text{ если } m \neq n \end{cases},$$

можно разделить спектр исследуемой функции на модули и тем самым перейти к параллельной обработке модулей входной информации. Учитывая гладкость функции, обучение ИНС можно осуществлять на первой половине спектра, при этом качество обучения будет ненамного хуже, чем если бы обучение производилось на всем спектре. Это возможно благо-

даря тому, что большая часть энергии функции сосредоточена в низкочастотном модуле. Однако проблемность такого решения заключается в том, что, во-первых, как отмечалось ранее, исследуемые функции на входе ИНС должны быть гладкими (что достижимо далеко не всегда) и, во-вторых, преобразование Фурье является комплекснозначным. В-третьих, преобразование Фурье обладает локализацией в частотной области в силу свойства ортогональности и полным ее отсутствием во временной области [11]. В условиях нестационарности входных значений ИНС этот недостаток будет иметь существенное значение. Для улучшения временной локализации используют дополнительную функцию $g(t-t_0)$, которая позволит преобразовать (1) к виду:

$$F(\omega)=\int_{-\infty}^{\infty} f(x)\,g(x-x_0)\,e^{-jx\omega}\mathrm{d}x.$$

Такое преобразование называют оконным преобразованием Фурье. Оконное преобразование частично решает вопрос временной локализации, однако обладает собственными недостатками, к важнейшему из которых относят принцип неопределенности Гейзенберга. В этом случае говорят не о конкретной частоте, а о диапазоне частот, присутствующих в сигнале в определенном диапазоне времени.

В теории передачи информации преобразование Фурье используют для анализа сигналов в частотной области. Например, при передаче сигнала в канале связи помехи носят, как правило, высокочастотный характер, поэтому для фильтрации полезного сигнала используют низкочастотный фильтр. При разложении сигнала на две части в частотной области применяют низкочастотный и высокочастотный фильтры с передаточными характеристиками $G(\omega)$ и $H(\omega)$ соответственно, при этом для идеальных фильтров $G(\omega)+H(\omega)=1$. После прохождения низкочастотного фильтра сигнал на выходе имеет вид $X^g(\omega)=G(\omega)X(\omega)$, а после высокочастотного – $X^h(\omega)=H(\omega)X(\omega)$. При разложении дискретного сигнала (входной сигнал ИНС имеет дискретный характер) на две компоненты $\{x_n\}\to\{x_n^h\},\{x_n^g\}$ получается, что количество данных на выходе фильтров $\{x_n^h\}$, $\{x_n^g\}$ вдвое больше, чем у первоначального сигнала $\{x_n\}$. На основании теоремы Котельникова лишние элементы можно удалить [11], этот процесс называется диадической децимацией. Несложно показать [11], что разделенный таким образом сигнал может быть восстановлен с помощью обратных фильтров.

Использование вместо преобразования Фурье таких ортогональных преобразований, как, например, дискретное косинусное преобразование (ДКП), позволяет снизить количество нейронов в последующих слоях ИНС за счет сжатия и отсутствия комплексных чисел [12]. Однако отбор значимых признаков все равно осуществляется с помощью дисперсионного критерия и требует отдельного исследования в процессе обучения. Существуют и другие методы выделения признаков в ИНС [23, 24], однако все они требуют проведения дополнительных исследований и не могут быть выполнены в реальном масштабе времени.

Использование Фурье-подобных преобразований осложняется еще тем, что частотное представление сигнала не всегда позволяет выделить лучший вектор признаков по сравнению с временным представлением. Например, когда входной сигнал ИНС не является стационарным (в этом случае шум при представлении кластера не является Гауссовым), тогда частотный спектр также не будет стационарным в силу влияния каждого временного отсчета на каждую частотную составляющую.

Если же вместо Фурье-подобных преобразований использовать вейвлет-преобразование, то можно получить выигрыш в количестве значимых признаков в первом модуле за счет деления спектра исследуемой функции пополам. Вейвлеты в общем виде представляют собой систему функций вида

$$\varphi_{a,b}(x)=\sqrt{2^a}\,\varphi\left(2^a x-b\right). \qquad (2)$$

Если V_a – пространство, порожденное системой функций (2), то имеют место следующие включения [11] $V_0\subset V_1\subset V_a$. Т.е. получается система вложенных подпространств $V_1\subset L^2(R)$, 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 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 $\{f_i(x_i)\}$ в пространстве $L^2(R)$ может быть спроектирован на набор подпространств $V_0\subset V_1\subset\ldots\subset V_a$ 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 по сравнению с другими проекциями.

Широко известный вейвлет Хаара [11, 12, 16] 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. Благодаря этому возможно решение задачи в вышеприведенной постановке, когда базовый модуль позволит решить задачу применения ИНС по назначению на устрой-

ствах с невысокой вычислительной мощностью с незначительной потерей точности и без дополнительного обучения.

2. Практическое решение задачи применения вейвлет-преобразования для создания модульных архитектур ИНС

Благодаря модели нейрона МакКаллока–Питса [13] стало возможным реализовать ортогональные преобразования непосредственно в нейронной сети. Авторы рассмотрели подобные реализации в работах [14 – 15]. Вейвлет-преобразование представляет на выходе аппроксимацию, т.е. низкочастотную часть сигнала и отдельно – высокочастотную. Подробно реализация вейвлет-преобразования в ИНС прямого распространения и сверточных сетях рассмотрена в [15].

В этой работе рассматривается модульная архитектура ИНС с использованием вейвлет-преобразования Хаара в первом слое сети. В качестве преобразователя используется первый сверточный слой ИНС с ядрами преобразования Хаара. ИНС реализована с использованием библиотеки PyTorch [17], ядра вейвлет-преобразования сформированы с помощью библиотеки PyWavelets [18], а в качестве данных для обучения использована база данных MNIST [19].

Как отмечалось в параграфе 1, процедура вейвлет-преобразования может быть описана как операция прохождения входного сигнала через полуполосный цифровой фильтр с частотной характеристикой $h(n)$ (высокочастотный фильтр) или $g(n)$ (низкочастотный фильтр):

$$\begin{cases} x(n)*h(n)=\sum_k x(k)h(n-k) \\ x(n)*g(n)=\sum_k x(k)g(n-k). \end{cases}$$

Если сигнал на входе ИНС представляет собой одномерную последовательность длиной n, то, используя одномерный сверточный слой с ядром $h(n)$ или $g(n)$, на выходе получим коэффициенты вейвлет-преобразования. Чтобы сократить количество слоев ИНС, можно использовать один слой с двумя (или более) различными ядрами. Для этого создается сверточный слой с одним входом, двумя (или более) выходами и с шагом, равным размерности ядра вейвлета. В этом случае будет создан сверточный слой с двумя ядрами, в которые заносятся значения $h(n)$ и $g(n)$.

Таким образом, модульная архитектура ИНС представляет собой две (или более, в зависимости от вида вейвлет-преобразования) ИНС, каждая из которых обучается высокочастотной и низкочастотной частью вейвлет-преобразования. Результаты классификации модулей ИНС объединяются последним слоем, который обучается совместно с модулями и не требует дополнительного времени на дообучение.

2.1. Исследование модульной архитектуры с использованием набора данных MNIST

Для проведения эксперимента с использованием набора данных MNIST [19] были использованы следующие библиотеки для компилятора Python 3.7 (64 bit): Numpy (v. 1.21.4), PyWavelets (v. 1.3.0), PyTorch (v. 1.12.1). Эксперимент проводился на персональном компьютере под управлением операционной системы «Windows 10 Домашняя», процессор Intel© Core™ i7-10510U, 16 ГБ оперативной памяти.

Перед началом исследования произведем сравнение обычной прямосвязной ИНС с модульной архитектурой. Модульная архитектура представлена двумя прямосвязными ИНС с длиной слоя, равной половине длины слоя обычной прямосвязной сети. Для этого обучим предлагаемые архитектуры на одном компьютере одинаковыми данными и сравним продолжительность и затраты времени на один цикл обучения. Для того, чтобы временные показатели исследуемой ИНС были сопоставимы с показателями ИНС с вейвлет-преобразованием, в состав прямосвязной ИНС включены первый сверточный слой (аналог вейвлет-преобразователя, но с разрешением обучения ядра) и последний, аналогичный объединяющему слою. В состав прямосвязной ИНС включены 4 основных слоя: $fc1\rightarrow(768, 1000)$, $fc2\rightarrow(1000, 1000)$, $fc3\rightarrow(1000, 1000)$, $fc4\rightarrow(1000, 10)$ с использованием нелинейного слоя ReLU. В качестве дополнительных использованы сверточный слой с размером ядра 2, а также $fc5\rightarrow(10, 10)$ (рис. 1*а*). Для модульной ИНС в ядро сверточного слоя были занесены значения `g(n) = wv.dec_lo` (низкочастотный фильтр) и `h(n) = wv.dec_hi` (высокочастотный фильтр), а также запрещено изменение значений ядра (`for param in self.conv1.parameters(): param.requires_grad = False`). Последний слой представлен в виде $fc5\rightarrow(20, 10)$ (рис. 1*б*). Решение от разделенной на модули сети выносится последним слоем сети $fc5$, который может быть как линейным (прямосвязным), так и сверточным. Объединяющий слой обучается совместно с модулями и поэтому не требует отдельного времени на обучение.

На рис. 2. представлены результаты обучения. Обе ИНС были обучены до достижения точности распознавания 0,98.

Из проведенного исследования видно, что модульная архитектура проигрывает в количестве циклов обучения в 1,29 раза, но выигрывает в среднем времени одного цикла обучения 1,68 раза.

Затем была произведена оценка производительности прямосвязной и модульной ИНС при использовании их по назначению. В качестве входных данных была использована тестовая часть базы MNIST [19]. Результаты приведены в табл. 1.

Из табл. 1 видно, что модульная ИНС имеет незначительный выигрыш в производительности в 4-м знаке

после запятой. Теперь оценим точность и производительность каждого из модулей. Для этого создадим ИНС, эквивалентную по размерам и архитектуре одному модулю. Осуществим загрузку весов из модулей обученной модели по очереди: сначала из низкочастотного модуля, затем – из высокочастотного.

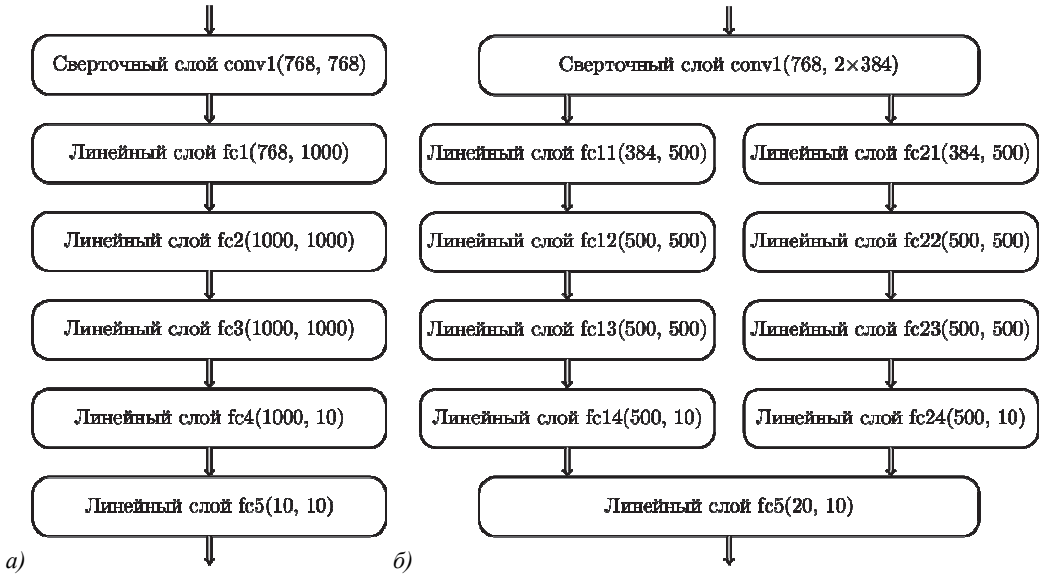

Рис. 1. Архитектуры ИНС для сравнения. (а) Архитектура прямосвязной ИНС. (б) Архитектура модульной ИНС

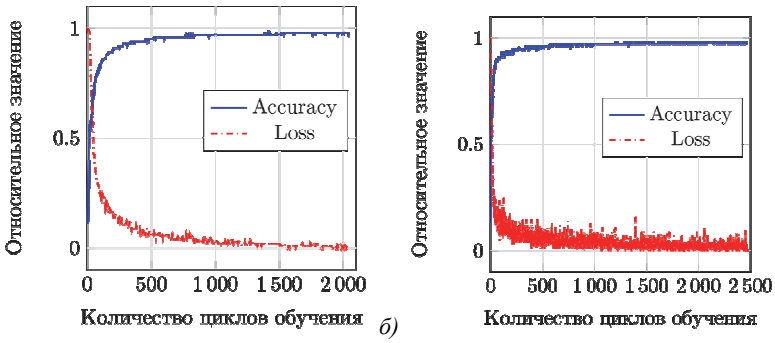

Рис. 2. Результаты обучения ИНС различной архитектуры. (а) Результаты обучения прямосвязной ИНС. (б) Результаты обучения модульной архитектуры ИНС

Табл. 1. Временные оценки применения по назначению прямосвязной и модульной ИНС

ИНС	Макс. время, с	Мин. время, с	Среднее время, с	СКО
Прямосвязная	0,0057	0,0029	0,0038	0,00056
Модульная	0,0050	0,0023	0,0030	0,00050

Поскольку модель, сохраненная в файл с помощью библиотеки PyTorch [17], имеет тип OrderedDict, можно удалить из нее ненужные слои (например, высокочастотный модуль) и загрузить в созданную ИНС. При выполнении распознавания тестовой части MNIST [19] ИНС, включающая в себя низкочастотную часть модульной архитектуры, показала точность распознавания 0,97, включающую только высокочастотную часть – 0,53 (табл. 2).

Временные оценки работы такой ИНС приведены в табл. 3.

Табл. 2. Качество распознавания изображений ИНС модульной архитектуры

Точность двух модулей	0,98
Точность низкочастотного модуля	0,97
Точность высокочастотного модуля	0,53

Табл. 3. Временные оценки применения по назначению низкочастотного и высокочастотного модуля ИНС

ИНС	Макс. время, с	Мин. время, с	Среднее время, с	СКО
Низкочастотный модуль	0,0053	0,0014	0,0018	0,00042
Высокочастотный модуль	0,0038	0,0013	0,0018	0,00030

Сравнение табл. 1 и 3 показывает более чем двукратный выигрыш в производительности по сравнению с прямосвязной ИНС и в 1,7 раза по сравнению с модульной архитектурой по среднему времени одного цикла. При этом потеря точности при использовании низкочастотного модуля составляет всего 1% (0,01). При этом сохраняется возможность восстановления ИНС модульного типа в полном объеме. Такой вариант может быть использован при обновлении

ИНС по каналам связи с ограниченной пропускной способностью. Такой вариант особенно интересен, если ИНС будет разделена на значительное количество модулей: 4, 8 и более.

Вейвлет-преобразование, в отличие от Фурье-подобных преобразований, имеет двумерную природу. В каждом подпространстве коэффициенты преобразования представляют изначальную функцию со своей степенью приближения. Так, полученные низкочастотный и высокочастотный модули ИНС могут быть разбиты с помощью преобразования Хаара еще на 2 модуля каждый. Таким образом, можно обучить модульную ИНС из 4 модулей.

В качестве следующего примера рассмотрим ИНС свёрточного типа (рис. 3). Здесь в качестве модуля используется нейронная сеть из двух сверточных слоев и двух прямосвязных. Все четыре модуля объединяются посредством линейного слоя $fc5 \to (40, 10)$.

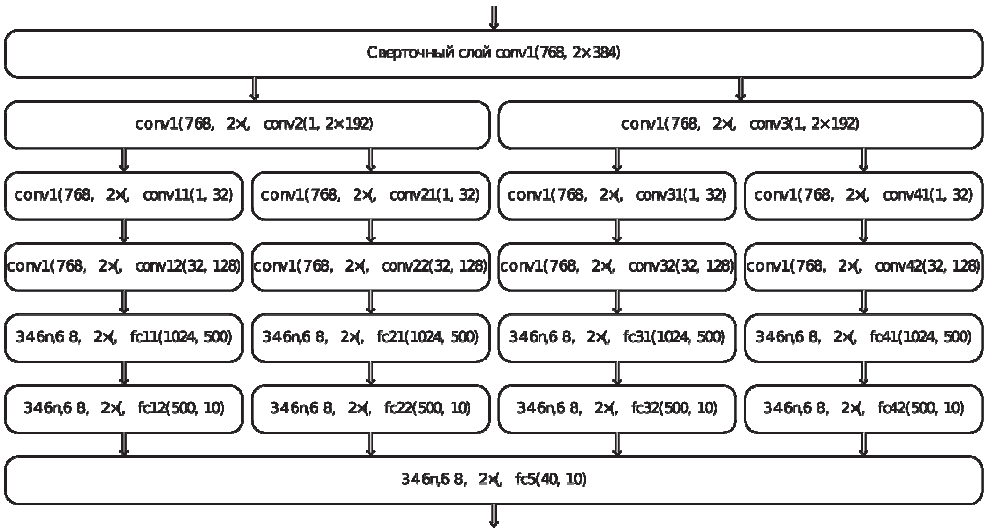

Рис. 3. Архитектуры четырехмодульной ИНС

Смысл эксперимента заключается в сравнении качества обучения модулей с использованием вейвлет-преобразования и без него. Входная информация с помощью сверточного слоя с одним входом и двумя выходами разбивается на 2 информационных модуля по 384 бита каждый. Два последующих сверточных слоя разбивают каждый из этих модулей еще на 2 по 192 бита. Таким образом, после прохождения трех сверточных слоев образуется 4 информационных модуля.

Нейронная сеть состоит из 4 модулей, каждый из которых содержит 2 сверточных слоя и 2 линейных. Линейные слои каждого модуля уменьшены в 4 раза по сравнению с исходной ИНС. Выходы модулей поступают на вход последнего линейного слоя размером (40×10). Обучение описанной ИНС осуществлялось 2 раза: один раз без вейвлет-преобразования, т.е. ядра первых 3 сверточных слоев не задавались, и второй, когда в первые 3 слоя было занесено ядро вейвлет-преобразования Хаара и запрещено его изменение. При обучении без вейвлет-преобразования ИНС удалось обучить только до качества распознавания 97 %, с вейвлет-преобразованием – до 98 %. При этом в первом случае обучение длилось 10 эпох, а во втором случае – всего 6.

После обучения ИНС было произведено 2 серии экспериментов: для ИНС с ядром вейвлет-преобразования и без него. Сравнительные характеристики этих 2 серий представлены в табл. 4. Во вновь созданную ИНС, повторяющую архитектурой описанную выше, заносились значения из сохраненной обученной модели. Эксперименты проводились следующим образом: сначала в составе ИНС оставили 1 модуль, занесли значения из обученной модели, вместо отсутствующих модулей на вход последнего (объединяющего) слоя подали нулевые значения и провели проверку применения ИНС на контрольных данных. В следующем эксперименте оставили 2 модуля, затем 3 и, наконец, 4.

Табл. 4. Оценки применения по назначению 1–4-х модулей ИНС

Количество модулей в ИНС	Точность распознавания без вейвлет-преобразования	Точность распознавания с вейвлет-преобразованием
1 модуль	26 %	88 %
2 модуля	74 %	96 %
3 модуля	96 %	97 %
4 модуля	97 %	98 %

Из табл. 4 наглядно видно преимущество ИНС с использованием вейвлет-преобразования. 1 модуль такой модели осуществляет распознавание с потерей точности всего 10 % при уменьшении количества операций в 4 раза. При этом 1 модуль ИНС, обученной без использования вейвлет-преобразования, дает точность распознавания на 60 % меньше. Близкие значения качества распознавания становятся при использовании 3-х и более модулей.

Проведенный эксперимент показывает преимущество использования вейвлет-преобразования при

модульном обучении при прочих равных параметрах ИНС.

2.2. Исследование модульной архитектуры с использованием набора данных GTSRB

Для проведения эксперимента с использованием набора данных GTSRB [20] были использованы следующие библиотеки для компилятора Python 3.7 (64 bit): Numpy (v. 1.21.4), PyWavelets (v. 1.3.0), PyTorch (v. 1.12.1). Эксперимент проводился на сервере под управлением операционной системы Linux, 2 процессора CPU E5-2690V4, 1 ТБ оперативной памяти.

Хотя база MNIST и является стандартом, предложенным Национальным институтом стандартов и технологий США с целью калибрации и сопоставления методов распознавания изображений с помощью машинного обучения в первую очередь на основе нейронных сетей, рассмотрим применение предложенного подхода к архитектурам, используемым в практических задачах. Для подтверждения полученных результатов авторы использовали AlexNet-подобную архитектуру нейронной сети для обучения на основе набора данных GTSRB (German Traffic Sign Recognition Benchmark) [21]. Первый слой осуществляет вейвлет-преобразование Хаара на основе ядра, полученного с помощью библиотеки PyWavlete [18]. Обучение высокочастотной и низкочастотной половины осуществляется отдельными модулями, причем в связи с уменьшением разрядности данных в результате вейвлет-преобразования линейные слои классификатора были уменьшены в 2 раза. Объединение результатов обучения осуществляется линейным слоем 86×43 (43 – количество знаков дорожного движения в базе). Для полноценного сравнения было произведено обучение сети немодульной архитектуры. Результаты обучения представлены на рис. 4.

Из данных, представленных на рис. 4, видно, что скорость обучения модульной архитектуры несколько выше, чем у обычной. Сравнительные временные характеристики представлены в табл. 5.

Табл. 5. Временные оценки применения по назначению ИНС обычной и модульной архитектуры

ИНС	Макс. время, с	Мин. время, с	Среднее время, с	СКО
Обычная архитектура	26,3	25,4	25,6	0,25
Модульная архитектура	28,4	27,6	28,0	0,30

Из данных, представленных на рис. 4 и в табл. 5, можно сделать вывод о том, что, в отличие от случая с прямосвязной ИНС, модульная архитектура проигрывает в скорости обучения, зато выигрывает в количестве циклов обучения. При обучении модульной архитектуры график качества распознавания очень быстро идет вверх и достигает приемлемых величин (более 90 %) уже после 5-й эпохи. Обычной сети для этого требуется 7 – 8 эпох.

Затраты времени одного модуля на выполнение распознавания тестовой (валидационной) части теста приведены в табл. 6.

Табл. 6. Временные оценки применения по назначению низкочастотного и высокочастотного модуля ИНС

ИНС	Макс. время, с	Мин. время, с	Среднее время, с	СКО
Низкочастотный модуль	20,3	19,6	19,8	0,18
Высокочастотный модуль	20,3	19,7	19,9	0,16

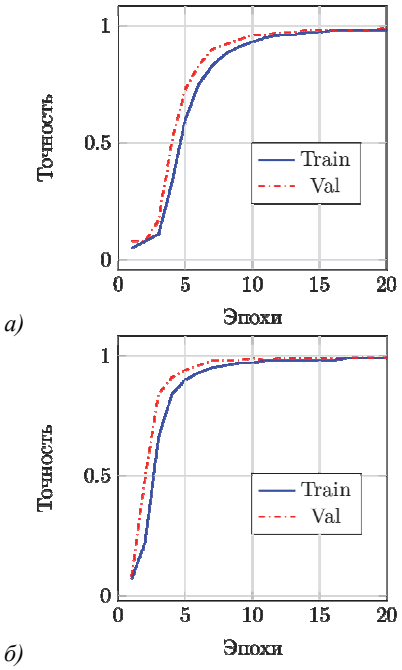

Рис. 4. Результаты обучения ИНС обычной и модульной архитектуры. (а) Результаты обучения AlexNet-подобной ИНС. (б) Результаты обучения модульной архитектуры ИНС

Из данных табл. 6 можно сделать вывод о том, что выигрыш от разделения ИНС не столь значителен, как в случае с прямосвязной ИНС, но все равно составляет 33 % по сравнению с обычной архитектурой, при этом точность распознавания (0,992) остается выше, чем ИНС с обычной архитектурой (0,987) за счет более высокой скорости обучения при одинаковом количестве эпох обучения.

Заключение

В процессе решения задачи модульного обучения ИНС возникает целый ряд вопросов: от решения обратной задачи (раздельного обучения модулей с целью их дальнейшего объединения) до взаимосвязи архитектуры ИНС со скоростью обучения. Конечно, все эти вопросы требуют систематического исследования. Тем не менее, в рамках представленного исследования решена задача масштабирования ИНС: при необходимости снижения затрат вычислительных ресурсов появилась возможность использования одного — двух или более модулей за счет адекватного

снижения точности ИНС (6 – 4 %). И это возможно в рамках существующих архитектур ИНС, что предполагает использование типовых библиотек для разработки программного обеспечения.

На наш взгляд, предлагаемая модульная архитектура ИНС может найти применение при переносе ИНС на маломощные вычислительные устройства, а также везде, где требуется оптимизация вычислительных ресурсов за счет точности распознавания. Кроме того, предлагаемый подход может быть использован в Интернете вещей для периферийных вычислений при ограниченных вычислительных ресурсах, распределенных вычислениях и т.п. В качестве дополнительных направлений для дальнейших исследований предполагается применение предлагаемого метода для других архитектур нейронных сетей, таких как ResNet, VGG, EfficientNet, MobileNet. Кроме того, предполагается сравнение различных видов вейвлетов для модульного обучения ИНС.

Благодарности

Работа выполнена при поддержке Российского научного фонда (проект № 22-71-10046), https://rscf.ru/en/project/22-71-10046/.

References

[1] Kussul ME. A modular representation of neural networks [In Russian]. Mathematical Machines and Systems 2006; 4: 51-62.

[2] Rykov VP. The modular principle of artificial neural network training using known neural network topologies as an example [In Russian]. Bulletin of Tambov State University 2014; 19(2): 583-586.

[3] Andreas J, Rohrbach M, Darrell T, Klein D. Neural module networks. Proc IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016: 39-48.

[4] Auda G, Kamel M, Raafat H. Modular neural network architectures for classification. Proc Int Conf on Neural Networks (ICNN'96) 1996; 2: 1279-1284. DOI: 10.1109/ICNN.1996.549082.

[5] Auda G, Kamel M, Raafat H. Voting schemes for cooperative neural network classifiers. IEEE Conf on Neural Networks (ICNN'95) 1995; 3: 1240-1243.

[6] Lu Z, Pu H, Wang F, Hu Z, Wang L. The expressive power of neural networks: A view from the width. 31st Conf on Neural Information Processing Systems (NIPS 2017) 2017: 6232-6240.

[7] Kidger P, Lyons T. Universal approximation with deep narrow networks. 33rd Annual Conf on Learning Theory 2020: 1-22.

[8] Kim JS, Cho Y, Lim TH. Prediction of locations in medical images using orthogonal neural networks. Eur J Radiol Open 2021; 8: 100388.

[9] Jamal A, Ashour M, Helmi R, Fong S. A wavelet–neural networks model for time series. 11th IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE) 2021: 325-330. DOI: 10.1109/ISCAIE51753.2021.9431777.

[10] D'Amario V, Sasaki T, Boix X. How modular should neural module networks be for systematic generalization? arXiv Preprint. 2022. Source: <arXiv:2106.08170v2>.

[11] Smolencev NK. Basics of wavelet theory. Wavelets in MATLAB [In Russian]. Moscow: "DMK Press" Publisher; 2019. ISBN: 5-94074-415-X.

[12] Ahmed N, Rao KR. Orthogonal transfarms for digital signal processing. Springer-Verlag; 1975.

[13] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943; 5(4): 115-133.

[14] Vershkov NA, Kuchukov VA, Kuchukova NN, Babenko M. The wave model of artificial neural network. Proc 2020 IEEE Conf of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) 2020: 542-547.

[15] Vershkov N, Babenko M, Tchernykh A, et al. Optimization of artificial neural networks using wavelet transforms. Program Comput Soft 2022; 48: 376-384. DOI: 10.1134/S036176882206007X.

[16] Haar A. Zur theorie der orthogonalen funktionensysteme. Gottingen: Georg-August Universitat; 1909.

[17] PyTorch. Source: <https://pytorch.org/>.

[18] PyWavelets. Source: <https://pypi.org/project/PyWavelets/>.

[19] Qiao Y. THE MNIST DATABASE of handwritten digits. 2007. Source: <http://www.gavo.t.u-tokyo.ac.jp/~qiao/database.html>.

[20] GTSRB - German traffic sign recognition benchmark. 2023. Source: <https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>.

[21] Ushakov YA, Polezhaev PN, Shukhman AE, Ushakova MV. Distribution of the neural network between mobile device and cloud infrastructure services [In Russian]. Modern Information Technology and IT-education 2018; 14(4): 903-910. DOI: 10.25559/SITITO.14.201804.903-910.

[22] Rytov SM, Kravtsov YuA, Tatarsky VI. Introduction to statistical radiophysics. Part 2. Random fields [In Russian]. Moscow: "Nauka" Publisher; 1978.

[23] Minkin AS, Nikolaeva OV, Russkov AA. Hyperspectral data compression based upon the principal component analysis. Computer Optics 2021; 45(2): 235-244. DOI: 10.18287/2412-6179-CO-806.

[24] Zenkov IV, Lapko AV, Lapko VA, Kiryushina EV, Vokin VN, Bakhtina AV. A method of sequentially generating a set of components of a multidimensional random variable using a nonparametric pattern recognition algorithm. Computer Optics 2021; 45(6): 926-933. DOI: 10.18287/2412-6179-CO-902.

Сведения об авторах

Вершков Николай Анатольевич, 1961 года рождения, в 1983 году окончил Ставропольское высшее военное инженерное училище связи по специальности «Автоматизированные системы управления войсками и оружием», кандидат технических наук, старший научный сотрудник Северо-Кавказского федерального университета. Сфера научных интересов: модулярная арифметика, нейрокомпьютерные технологии, цифровая обработка сигналов. E-mail: *nvershkov@ncfu.ru*

Бабенко Михаил Григорьевич, 1985 года рождения, в 2007 году окончил Ставропольский государственный университет по специальности «Математика», работает начальником отдела теоретико-числовых систем Северо-Кавказского центра математических исследований, заведующим кафедрой вычислительной математики и кибернетики Северо-Кавказского федерального университета. Область научных исследований: система остаточных классов, искусственные нейронные сети, безопасность, облачные вычисления, интернет вещи.
E-mail: *mgbabenko@ncfu.ru*

Кучукова Наталья Николаевна, 1990 года рождения, в 2018 году окончила Северо-Кавказский федеральный университет по направлению «09.06.01 Информатика и вычислительная техника», ведущий специалист Северо-Кавказского федерального университета. Сфера научных интересов: нейросетевые технологии, распознавание речи, цифровая обработка сигналов. E-mail: *nkuchukova@ncfu.ru*

Кучуков Виктор Андреевич, 1990 года рождения, в 2018 году окончил Северо-Кавказский федеральный университет по направлению «09.06.01 Информатика и вычислительная техника», младший научный сотрудник отдела теоретико-числовых систем Северо-Кавказского центра математических исследований. Сфера научных интересов: модулярная арифметика, машинное обучение, проектирование СБИС. E-mail: *vkuchukov@ncfu.ru*

Кучеров Николай Николаевич, 1991 года рождения, учился в Ставропольском государственном университете (ныне – Северо-Кавказский федеральный университет). Работает старшим научным сотрудником. Область научных интересов: модулярная арифметика, гомоморфное шифрование, матричное исчисление и нейронные сети. E-mail: *nkucherov@ncfu.ru*

ГРНТИ: 28.23.15
Поступила в редакцию 16 января 2023 г. Окончательный вариант – 20 июля 2023 г.

Transverse-layer partitioning of artificial neural networks for image classification

N.A. Vershkov [1], M.G. Babenko [1], N.N. Kuchukova [1], V.A. Kuchukov [1], N.N. Kucherov [1]
[1] North-Caucasus Center for Mathematical Research, North Caucasus Federal University, 355017, Russia, Stavropol, st. Pushkin 1

Abstract

We discuss issues of modular learning in artificial neural networks and explore possibilities of the partial use of modules when the computational resources are limited. The proposed method is based on the ability of a wavelet transform to separate information into high- and low-frequency parts. Using the expertise gained in developing convolutional wavelet neural networks, the authors perform a transverse-layer partitioning of the network into modules for the further partial use on devices with low computational capability. The theoretical justification of this approach in the paper is supported by experimentally dividing the MNIST database into 2 and 4 modules before using them sequentially and measuring the respective accuracy and performance. When using the individual modules, a two-fold (or higher) performance gain is achieved. The theoretical statements are verified using an AlexNet-like network on the GTSRB dataset, with a performance gain of 33% per module with no loss of accuracy.

Keywords: wavelet transform, artificial neural networks, convolutional layer, orthogonal transforms, modular learning, neural network optimization.

Citation: Vershkov NA, Babenko MG, Kuchukova NN, Kuchukov VA, Kucherov NN. Transverse-layer partitioning of artificial neural networks for image classification. Computer Optics 2024; 48(2): 312-320. DOI: 10.18287/2412-6179-CO-1278.

Acknowledgements: the research was financially supported by the Russian Science Foundation under grant No. 22-71-10046, https://rscf.ru/en/project/22-71-10046/.

Authors' information

Nikolay Anatolyevich Vershkov, (b. 1961), graduated from Stavropol Higher Military Engineering School of Communications, majoring in Automated Control Systems for Troops and Weapons in 1985, Ph.D. in Engineering Sciences. Currently he works as Senior Researcher at the North Caucasus Federal University. Research interests are modular arithmetic, neuro-computer technologies, digital signal processing. E-mail: *nvershkov@ncfu.ru*

Mikhail Grigorievich Babenko, (b. 1985) graduated from Stavropol State University in 2007, majoring in Mathematica. Currently he works as the head of Number-Theoretic Systems department of the North Caucasian Center for Mathematical Research, the head of Computational Mathematics and Cybernetics department at the North Caucasus Federal University. Research interests are residue number system, neural networks, security, cloud computing, Internet of Things. E-mail: *mgbabenko@ncfu.ru*

Natalya Nikolaevna Kuchukova (b. 1990), graduated from North Caucasus Federal University in 2018, majoring in Informatics and Computer Engineering. She works as leading specialist at the North Caucasus Federal University. Research interests are neural network technologies, speech recognition, digital signal processing. E-mail: *nkuchukova@ncfu.ru*

Viktor Andreevich Kuchukov, (b. 1990), graduated from North Caucasus Federal University in 2018, majoring in Informatics and Computer Engineering. He works as the junior researcher of Number-Theoretic Systems department of the North Caucasian Center for Mathematical Research. Research interests are modular arithmetic, machine learning, VLSI design. E-mail: *vkuchukov@ncfu.ru*

Nikolay Nikolaevich Kucherov, (b. 1991), studied at the Stavropol State University (now - North Caucasus Federal University). Works as Senior Scientist. Research interests: modular arithmetic, homomorphic encryption, matrix calculus and neural networks. E-mail: *nkucherov@ncfu.ru*

Received January 16, 2023. The final version – July 20, 2023.