



Известия высших учебных заведений. Прикладная нелинейная динамика. 2025. Т. 33, № 6
Izvestiya Vysshikh Uchebnykh Zavedeniy. Applied Nonlinear Dynamics. 2025;33(6)

Научная статья
УДК 004.032.26, 530.152.2

DOI: 10.18500/0869-6632-003177
EDN: VYXWAN

Влияние внутреннего шума на точность работы свёрточной нейронной сети

Н. И. Семенова

Саратовский национальный исследовательский государственный университет
имени Н. Г. Чернышевского, Россия
E-mail: semenovani@sgu.ru

Поступила в редакцию 28.03.2025, принята к публикации 26.05.2025,
опубликована онлайн 30.05.2025, опубликована 28.11.2025

Аннотация. Цель настоящего исследования — установить особенности распространения и накопления шумов в свёрточной нейронной сети. В данной статье рассматривается, как меняется точность работы обученной свёрточной сети в зависимости от типа шумового воздействия и его интенсивности. **Методы.** В качестве источников шумов рассматриваются источники белого гауссова шума. В зависимости от способа воздействия шума на искусственные нейроны использовалось аддитивное и мультиплекативное шумовое воздействие, а в зависимости от влияния на слой нейронов рассматривалось коррелированное и некоррелированное воздействие. **Результаты.** Было показано, что аддитивные шумы (как коррелированный, так и некоррелированный) в сетях со свёрточным слоем накапливаются сильнее, чем в сети без него. Зависимости точности работы сети от интенсивности мультиплекативного коррелированного шума выглядят одинаково для сетей со свёрточным слоем и без, а аналогичные зависимости для мультиплекативного некоррелированного шума выглядят гораздо лучше для сети со свёрточным слоем. Также были рассмотрены слои дискретизации с MaxPooling и MeanPooling. Их использование позволяет существенно поднять точность при наличии аддитивного шума в свёрточном слое. Спадание точности при увеличении интенсивности мультиплекативного коррелированного шума практически одинаковое для сетей со слоем субдискретизации и без него. Что же касается некоррелированного мультиплекативного шума, то сети с MaxPooling менее устойчивы к его воздействию. **Заключение.** Показано, что аддитивный шум сильно ухудшает работу сети при наличии свёрточного слоя, однако его негативное воздействие можно уменьшить при наличии слоя субдискретизации сразу после слоя свёртки. Для мультиплекативного шума выводы не такие однозначные. В большинстве случаев его влияние одинаково независимо от наличия слоев свёртки и субдискретизации. Однако использование MaxPooling в слое субдискретизации делает сеть менее устойчивой к мультиплекативному некоррелированному шумовому воздействию.

Ключевые слова: нейронные сети, сверточные нейронные сети, шумовое воздействие, белый шум, уменьшение шума.

Благодарности. Исследование выполнено за счет гранта Российского научного фонда № 23-72-01094, <https://rscf.ru/project/23-72-01094/>.

Для цитирования: Семенова Н. И. Влияние внутреннего шума на точность работы свёрточной нейронной сети // Известия вузов. ПНД. 2025. Т. 33, № 6. С. 898–916. DOI: 10.18500/0869-6632-003177. EDN: VYXWAN

Статья опубликована на условиях Creative Commons Attribution License (CC-BY 4.0).

The impact of internal noise on the performance of convolutional neural network

N. I. Semenova

Saratov State University, Russia

E-mail: semenovani@info.sgu.ru

Received 28.03.2025, accepted 26.05.2025,
available online 30.05.2025, published 28.11.2025

Abstract. Purpose. This study aims to establish the characteristics of noise propagation and accumulation in convolutional neural networks. The article investigates how the accuracy of a trained convolutional network varies depending on the type and intensity of noise exposure. Methods. White Gaussian noise sources were used as the basis for noise exposure. Two types of noise exposure were applied to artificial neurons: additive and multiplicative. Additionally, the effects of correlated and uncorrelated noise on the layers of neurons were examined. Results. The findings indicate that additive noise (both correlated and uncorrelated) accumulates more significantly in networks with convolutional layers compared to those without. The relationship between network accuracy and the intensity of multiplicative correlated noise is similar for both types of networks. However, the impact of multiplicative uncorrelated noise is more favorable for networks with convolutional layers. The study also considered pooling layers, specifically MaxPooling and MeanPooling, which significantly enhance accuracy in the presence of additive noise within the convolutional layer. The decline in accuracy due to increasing intensity of multiplicative correlated noise is nearly identical for networks with and without pooling layers. Conversely, networks employing MaxPooling demonstrate reduced resilience to uncorrelated multiplicative noise. Conclusion. The study demonstrates that additive noise severely degrades network performance when a convolutional layer is present, though this negative effect can be mitigated by including a pooling layer immediately following the convolutional layer. In contrast, the effects of multiplicative noise are less clear-cut. In most cases, its impact remains consistent regardless of the presence of convolution and pooling layers. However, the use of MaxPooling in the pooling layer may compromise the network's robustness against multiplicative uncorrelated noise.

Keywords: neural networks, convolutional neural networks, noise influence, white noise, noise reduction.

Acknowledgements. The study was supported by the Russian Science Foundation grant No. 23-72-01094, <https://rscf.ru/project/23-72-01094/>.

For citation: Semenova NI. The impact of internal noise on the performance of convolutional neural network. Izvestiya VUZ. Applied Nonlinear Dynamics. 2025;33(6):898–916. DOI: 10.18500/0869-6632-003177

This is an open access article distributed under the terms of Creative Commons Attribution License (CC-BY 4.0).

Введение

Искусственные нейронные сети (ИНС) за последние несколько лет стали мощным инструментом, позволяющим решать задачи, которые ранее считались неразрешимыми без привлечения естественного интеллекта [1]. На данный момент нейронные сети активно применяются в системах диагностики, для распознавания образов [2, 3], решения задач классификации, распознавания речи [4], предсказания климатических явлений [5] и многоного другого. Изначально общая структура ИНС была вдохновлена работой человеческого мозга, но в современных ИНС свойства нейронов и особенности связи между ними зависят от решаемой задачи и довольно далеки от изначального биологического примера.

С точки зрения использования ИНС их моделирование и проведение вычислений на компьютерах или вычислительных кластерах является весьма ресурсоемкой задачей. Согласно статистике компании OpenAI [6], за последние 20 лет рост производительности ИНС реализуется гораздо быстрее, чем ранее. Если до 2012 рост производительности (в Petaflop/s-days) происходил с удвоением каждые 2 года, то после 2012 года период удвоения составляет 3–4 месяца.

С точки зрения увеличения мощности современного компьютерного оборудования рост не такой стремительный. Таким образом, моделирование нейронной сети на цифровом оборудовании является узким горлышком при масштабировании сети, скорости получения, обработке информации и энергоэффективности [7, 8]. В последние годы все больше исследователей в области нейронных сетей заинтересованы в создании аппаратных нейронных сетей [9], в которых нейроны и связь между ними представляют собой реальное устройство, способное обучаться и решать задачи. В зарубежной литературе можно встретить названия “analog neural network” и “hardware neural network”. В рамках этого направления нейронные сети не моделируются на компьютере, а представляют собой реальное устройство, в котором сами нейроны сети и связь между ними реализованы на физическом уровне, что позволяет получить существенный прирост в скорости и энергоэффективности [10, 11]. В рамках этого направления наблюдается экспоненциальный рост числа работ с аппаратными реализациями ИНС. Наибольшую эффективность на данный момент показывают аппаратные ИНС, в основе которых лежат лазеры [12], мемристоры [13], спин-трансферные осцилляторы [14]. Связь между нейронами в оптических реализациях ИНС основывается на принципах голограмии [15], дифракции [16, 17], интегрированных сетей модуляторов Маха–Цендера [18], спектрального уплотнения каналов [19], оптических связей, реализованных при помощи 3D-принтера [20–22].

В аппаратных ИНС исчезает проблема обращения к памяти и математических операций над большим количеством данных, так как каждый нейрон соответствует аппаратному нелинейному компоненту, а каждое соединение – физическому каналу связи. Это дает существенный прирост в скорости обработки информации и энергоэффективности, однако аппаратные ИНС подвержены внутренним шумам, исходящим из составляющих компонент таких устройств. В случае цифровой компьютерной реализации ИНС шум может попасть в систему исключительно с входным сигналом, тогда как в аппаратной ИНС существует множество внутренних источников шума с различными свойствами. Поэтому исследование влияния различных шумов на работу таких сетей, а также поиск топологий, при которых сеть сама сможет подавлять внутренние шумы, является актуальной и важной задачей. В предыдущих работах мы рассматривали влияние внутренних шумов на работу обученных глубоких сетей прямого распространения [23] и рекуррентных сетей [24], а также были предложены универсальные стратегии уменьшения внутренних шумов сети [25, 26].

В данной статье рассматривается еще один принципиально важный тип ИНС – свёрточные нейронные сети. По аналогии с нашими предыдущими статьями будут рассмотрены аддитивный и мультиплектический шумы в зависимости от воздействия на конкретные нейроны, а также коррелированный и некоррелированный шумы в зависимости от влияние на слой сети, состоящий из нейронов. Рассматриваемые типы шумового воздействия были вдохновлены аппаратной сетью из [16], реализованной в оптическом эксперименте. При анализе выходного сигнала этой сети было получено, что внутренние шумы в установке, как правило, бывают аддитивными и мультиплектическими, и вместе с тем коррелированными и некоррелированными. При моделировании можно использовать источники белого гауссова шума с нулевым средним. Также были установлены интенсивности шумовых воздействий, характерных для этой экспериментальной установки [16, 27]. Однако в данной статье будут рассматриваться разные интенсивности шума, чтобы полученные результаты охватывали не одну отдельно взятую аппаратную ИНС, а могли быть адаптированы и для других аппаратных реализаций ИНС.

Свёрточные нейронные сети (в англоязычной литературе “convolutional neural networks”) являются особым подвидом глубоких нейронных сетей. Работа свёрточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям и далее к ещё более абстрактным деталям вплоть до выделения понятий высокого уровня. При этом в процессе обучения сеть самонастраивается и вырабатывает сама необходимую

иерархию абстрактных признаков (последовательности карт признаков), фильтруя маловажные детали и выделяя существенное. С точки зрения построения сети это выливается в чередование свёрточных слоев (convolution layers) и слоев субдискретизации (pooling layers). Подобные особенности функционирования делают свёрточные сети весьма эффективными в распознавании изображений, и их часто используют во многих областях, включая компьютерное зрение и обработку естественного языка [28]. Особый тип связи и наличие слоев с чередующейся топологией позволяет взглянуть на вопросы шумового воздействия и накопления шума с принципиально новой стороны.

1. Исследуемые системы

1.1. Свёрточные сети. В данной статье рассматривается влияние шума на свёрточную нейронную сеть (СНС). Для того чтобы исключить сложное влияние статистических характеристик матриц связи, будет рассмотрена упрощенная обученная сеть, состоящая из основных компонент, присущих именно свёрточным сетям. Ранее нами были рассмотрены глубокие сети прямого распространения [23], где были сделаны выводы о распространении шума исходя из дисперсии шумового воздействия и статистических свойств матриц связи в обученной сети. В данной статье мы будем отталкиваться от сделанных ранее выводов, но их необходимо скорректировать исходя из особенностей слоев свёртки и субдискретизации, которые используются в СНС.

Для обучения сети будет использоваться довольно стандартная задача распознавания рукописных цифр из базы данных MNIST [29]. В этой базе данных содержатся 70000 изображений в черно-белом формате размером 28×28 пикселей. Часть этих изображений используется для обучения сети (60000), а оставшаяся часть — для тестирования. При работе с базой данных MNIST определенные условия накладываются на входной и выходной слой сети. Входной слой должен быть составлен таким образом, чтобы каждый входной нейрон принимал на вход значение соответствующего пикселя изображения. Так как изображения имеют размер 28×28 пикселей, то входной слой должен состоять из 784 нейронов. Для удобства работы значения, полученные из входного изображения, будут отнормированы на 255, чтобы входные значения сети принадлежали диапазону $[0, 1]$.

Нейронная сеть должна быть обучена для решения задачи классификации изображений, то есть каждое входное изображение необходимо отнести к одному из 10 возможных классов (цифры 0–9). Тогда выходной слой должен состоять из 10 нейронов, причем каждый выходной нейрон отвечает за свою цифру. Выходным сигналом сети считается не сам выходной сигнал выходных нейронов, а то, какой именно выходной нейрон имеет максимальное значение. Например, если на вход сети передается изображение с цифрой 0, тогда максимальным значением должен обладать 0-й выходной нейрон. Такая операция называется softmax.

В последующих разделах в СНС будет добавляться шум, а её структура будет постепенно усложняться введением новых компонент (рис. 1). Сначала будет рассмотрено влияние шума на максимально упрощенную сеть, в которой нет скрытых слоев (рис. 1, a) и которая состоит из входного и выходного полно связанных слоев. Шум будет вводиться в последний выходной слой, чтобы было понятно, как сильно влияет шум на точность работы сети, когда после слоя с шумовым воздействием нет никаких дополнительных слоев, так как матрицы связи между слоями могут оказывать дополнительное влияние на накопление или уменьшение внутренних шумов [23].

На следующем этапе (рис. 1, b) будет рассмотрена сеть, в которой добавлен слой свёртки (в англоязычной литературе “convolutional layer”). В этой сети шум уже будет добавляться в слой свёртки. Смысл слоя свёртки заключается в фильтрации изображения. Слой свёртки включает в себя фильтр с ядром определенного размера. В данной работе мы рассмотрим размеры 3×3 и 5×5 . По сути, свёрточный слой представляет собой матрицу или несколько матриц

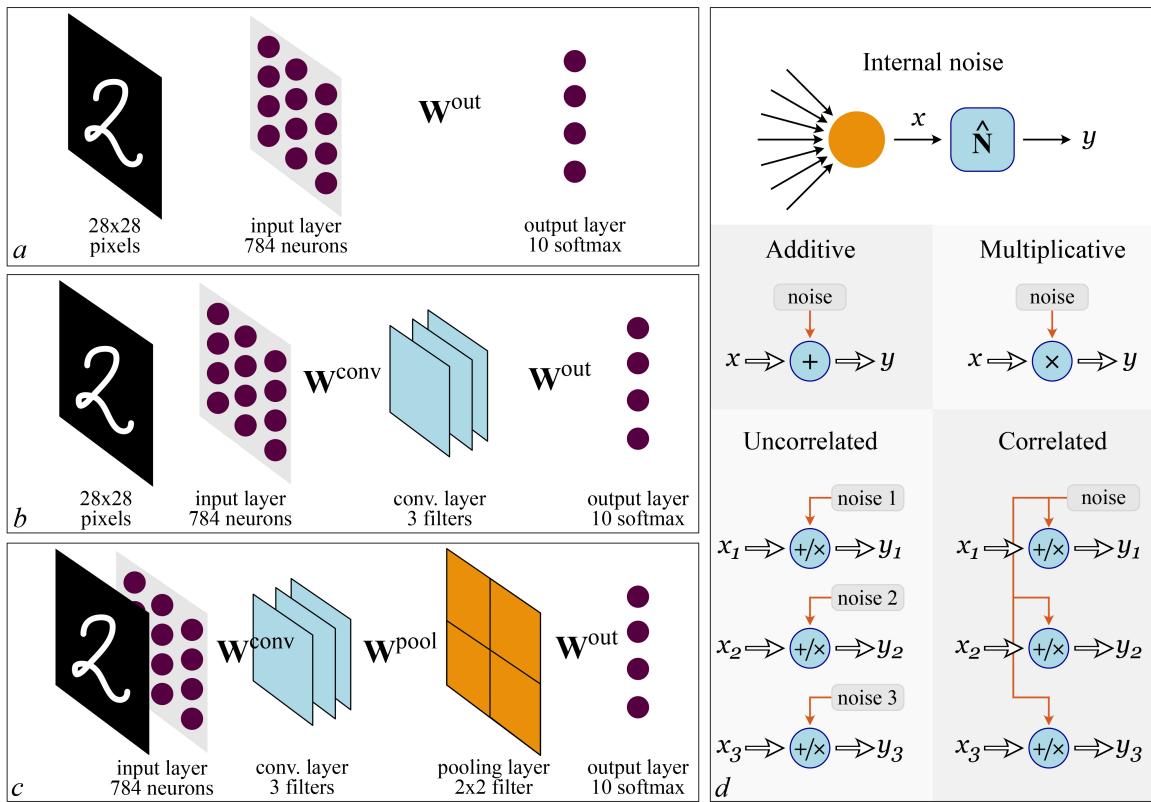


Рис. 1. Схематичное представление рассматриваемых сетей (a–c) и способов введения шумового воздействия (d) (цвет онлайн)

Fig. 1. Schematic representation of considered neural networks (a–c) and methods of introducing the noise exposure (d) (color online)

(если фильтров несколько) определенного размера, значения которой подбираются в процессе обучения. Применение фильтра размером 3×3 на этапе свёртки заключается в следующем. Например, из входного слоя мы получили матрицу значений, состоящую 784 значений. Для удобства интерпретации расположим их так, чтобы они имели вид матрицы размером 28×28 . Фильтр будет применен к левому верхнему углу матрицы, захватывая 9 элементов, далее происходит поэлементное перемножение этих 9 элементов матрицы и элементов фильтра, и их суммарное значение записывается уже в новую матрицу. Затем фильтр сдвигается на один столбец вправо, и процедура повторяется, пока фильтр не дойдет до конца строки матрицы, после этого фильтр сдвигается на одну строку вниз, и вся процедура снова повторяется. В конце после применения фильтра мы получаем новую матрицу, но ее размер уже составляет 26×26 (то есть 676 нейронов понадобится для обработки этих значений). Если фильтр имеет размер 5×5 , тогда несложно получить, что входное изображение сжимается до 24×24 (соответствует 576 нейронам).

На этапе свёртки таких фильтров может быть несколько. Если фильтр 3×3 всего один, то на этапе свёртки задействовано 676 нейронов, и выходная матрица \mathbf{W}^{out} , связывающая скрытый слой с выходным слоем, должна иметь размер 676×10 . Если используется два фильтра, тогда нейронов становится 1352, и размер \mathbf{W}^{out} составляет 1352×10 . Использование 5 фильтров приводит уже к $676 \times 5 = 3380$ нейронам и так далее. Для топологии сети, приведенной на рис. 1, b, шум будет вводиться в нейроны, полученные в результате применения слоя свёртки.

Также важно отметить, что применение свёртки можно представить в матричном виде, характерном для двух полно связанных слоев, \mathbf{W}^{conv} . Процедура представления свёрточного слоя

в виде двух полносвязных слоев и матричного умножения детально описана в книге [30] и заключается в особом заполнении матрицы \mathbf{W}^{conv} значениями из фильтров свёртки. Применяемые фильтры можно представить в виде трехмерной матрицы \mathbf{K} размерностью $m \times n \times n$, где m — число фильтров размером $n \times n$. Тогда для входных изображений размером $N \times N$ после применения свёртки получается m новых изображений размером $(N - n + 1) \times (N - n + 1)$, а элементы двумерной матрицы \mathbf{W}^{conv} определяются как

$$W_{i_1 \cdot N + j_1, k \cdot (N - n + 1)^2 + i_2 \cdot (N - n + 1) + j_2}^{\text{conv}} = K_{k, i_1 - i_2, j_1 - j_2}, \quad (1)$$

если элементы матрицы \mathbf{K} существуют. В остальных случаях элементы матрицы \mathbf{W}^{conv} задаются равными нулю. Уравнение (1) задает связь между элементами (i_1, j_1) исходного изображения $N \times N$ с элементами нового изображения (i_2, j_2) размером $(N - n + 1) \times (N - n + 1)$ после применения k -го фильтра.

Слой субдискретизации (иначе подвыборки или пулинга, “pooling” в англоязычной литературе) применяется для уменьшения размерности, которая создается после свёртки (рис. 1, c). Роль субдискретизации заключается в выделении входного значения одного нейрона из некоторой группы нейронов. Преобразование имеет вид непересекающихся прямоугольников или квадратов одинакового размера, каждый из которых захватывает определенную группу нейронов для последующей трансформации их значений в одно. Наиболее часто применяется выделение максимального значения (MaxPooling) или среднего значения (MeanPooling). Субдискретизация интерпретируется так: если на предыдущей операции свёртки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение уже не нужно, и оно уплотняется до менее подробного. Например, после применения свёртки с одним фильтром 3×3 образовалось 676 нейронов, которые можно расположить в виде квадрата 26×26 . Применение субдискретизации с фильтром 2×2 приводит к тому, что матрица значений 26×26 разбивается на ячейки размером 2×2 . Из каждой ячейки выбирается только одно значение (максимум или среднее), и, таким образом, размерность матрицы уменьшается до 13×13 , что соответствует 169 нейронам, тогда размер выходной матрицы \mathbf{W}^{out} уже становится 169×10 . На рис. 1, c вся процедура субдискретизации представлена в виде матрицы связи \mathbf{W}^{pool} , которая в случае фильтра 2×2 и использования MeanPooling заполняется значениями $1/4$. В случае MaxPooling ситуация немного сложнее и требует использования функции наподобие softmax. Подробно этот переход описан в книге [30].

Для всех типов сетей обучение проводилось при помощи свободно распространяемой библиотеки Keras [31] на языке программирования Python, используя оптимизатор “adam” и категориальную кросс-энтропию (categorical cross-entropy). Суть обучения сети заключается в корректном определении принадлежности каждого изображения из базы данных MNIST к одному из 10 возможных классов (10 цифр). Категориальная кросс-энтропия — это функция потерь, которая используется в задачах многоклассовой классификации:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{i=1}^{Cl} y_i \log(\hat{y}_i), \quad (2)$$

где \mathbf{y} — истинное распределение вероятностей на выходе сети, $\hat{\mathbf{y}}$ — предсказанное распределение вероятностей, Cl — общее количество классов. В данном случае их 10. Adam (Adaptive Moment Estimation) — это широко распространенный алгоритм оптимизации градиентного спуска, который активно используется в глубоком обучении. Он предназначен для эффективного обновления весов сети в процессе обучения путем адаптации скорости обучения для каждого параметра в отдельности. Благодаря этому алгоритму удается в несколько раз ускорить обучения и использовать

меньшее количество эпох. Обучение проводилось в течение 100 эпох. Ниже приведена часть кода, которая задает свёрточную сеть с пятью фильтрами размером 3×3 в свёрточном слое и фильтром 2×2 MaxPooling в слое субдискретизации.

```

kernel_size=3
filters_number = 5
# Задание структуры нейросети
model = tf.keras.models.Sequential([
    tf.keras.layers.Input(shape=(28, 28, 1)),
    tf.keras.layers.Conv2D(filters_number, kernel_size=kernel_size,
                          activation=None, use_bias=False),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(10, activation='softmax', use_bias=False)
])
# Компиляция и обучение сети
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(x_train, y_train, epochs=100)

```

1.2. Типы внутренних шумов. Свойства и особенности введения внутренних шумовых воздействий аналогично нашим предыдущим работам [23, 25]. Исходные типы внутренних шумов, их интенсивности и методы введения были получены из аппаратной реализации ИНС в оптическом эксперименте, предложенном в работе [16]. Но здесь будут рассмотрены различные интенсивности шума, чтобы сделать результаты более общими и применимыми к другим аппаратным сетям.

Схематично все рассматриваемые типы шумов приведены на рис. 1, d. В зависимости от того, как шум влияет на сигнал одного отдельно взятого нейрона, будут рассмотрены аддитивный и мультипликативный шум. Будем считать, что каждый нейрон обладает своим свободным от шума выходным сигналом x_i , в который включено влияние матриц связи, всевозможные дополнительные операции типа свёртки или субдискретизации. Затем в этот сигнал добавляется шумовое воздействие:

$$y_i(t) = x_i(t) \cdot \left(1 + \sqrt{2D_M} \xi_M(t, i)\right) + \sqrt{2D_A} \xi_A(t, i). \quad (3)$$

Аддитивный шум ξ_A добавляется к выходному незашумленному сигналу, а мультипликативный шум (с индексами “M”) умножается на него. Обозначение ξ соответствует белому гауссовскому шуму с нулевым средним значением и дисперсией, равной единице. Его множитель $\sqrt{2D}$ определяет общую дисперсию шумового воздействия, равную $2D$. Величину D часто называют интенсивностью шумового воздействия. В уравнении (3) индекс i соответствует номеру нейрона в пределах одного слоя, а t — номеру входного изображения.

В экспериментальных реализациях нейронных сетей часто имеют место шумы, которые являются одинаковыми для групп нейронов. Поэтому наряду с классификацией шумов в зависимости от воздействия на отдельно взятый нейрон необходимо еще ввести классификацию шумового воздействия в зависимости от влияния на группу нейронов (в этой статье это один слой). Шумовое воздействие, значения которого для каждого нового входного изображения разные, но они одинаковые для всех нейронов в пределах одного слоя, мы будем называть коррелированным (correlated, $\sqrt{2D_A^C} \xi_A^C(t)$, $\sqrt{2D_M^C} \xi_M^C(t)$), а шумовое воздействие, значения которого разные для этих нейронов, будем называть некоррелированным воздействием (uncorrelated, $\sqrt{2D_A^U} \xi_A^U(t, i)$, $\sqrt{2D_M^U} \xi_M^U(t, i)$).

Таким образом, всего в статье будет рассматриваться четыре типа шумового воздействия:

- аддитивный некоррелированный шум: $y_i(t) = x_i(t) + \sqrt{2D_A^U} \xi_A^U(t, i);$
- аддитивный коррелированный шум: $y_i(t) = x_i(t) + \sqrt{2D_A^C} \xi_A^C(t);$
- мультипликативный некоррелированный шум: $y_i(t) = x_i(t) \cdot (1 + \sqrt{2D_M^U} \xi_M^U(t, i));$
- мультипликативный коррелированный шум: $y_i(t) = x_i(t) \cdot (1 + \sqrt{2D_M^C} \xi_M^C(t));$

Таким образом, оператор, отвечающий за общее шумовое воздействие, $\hat{\mathbf{N}}$ может быть введен следующим образом:

$$\vec{y}(t) = \hat{\mathbf{N}} \vec{x}(t),$$

$$y_i(t) = x_i(t) \cdot (1 + \sqrt{2D_M^C} \xi_M^C(t)) (1 + \sqrt{2D_M^U} \xi_M^U(t, i)) + \sqrt{2D_A^C} \xi_A^C(t) + \sqrt{2D_A^U} \xi_A^U(t, i). \quad (4)$$

При этом получается, что шум добавляется к выходному сигналу нейронов, поэтому функция активации при этом не играет никакой роли. Распространение шума будет зависеть от следующей за зашумленным слоем матрицы связи. Аддитивный шум при этом можно интерпретировать как шум, исходящий из нейрона, а мультипликативный — как шум, встроенный в связь между зашумленным слоем и следующим.

В дальнейшем оценка влияния шумового воздействия будет проводиться с точки зрения изменения точности распознавания изображений в зависимости от интенсивности шума. Так как рассматриваются естественные шумы, характерные для физических систем, оценка влияния шума будет проводиться следующим образом. Один и тот же входной сигнал повторяется несколько раз (в данной статье 100 раз), затем для каждого повторения вычисляется точность работы зашумленной сети, а затем считается средняя точность, характерная для обученной зашумленной сети при определенной интенсивности шумового воздействия.

2. Шум в выходном слое сети

Для начала рассмотрим наиболее простой случай сети, состоящей из входного и выходного полносвязных слоев (см. рис. 1, a). Далее топология сети будет усложняться добавлением слоя свёртки и субдискретизации.

Выходной сигнал такой сети будет определяться как

$$\vec{x}^{\text{out}} = \vec{x}^{\text{in}} \cdot \mathbf{W}^{\text{out}}, \quad \vec{y}^{\text{out}} = \hat{\mathbf{N}} \vec{x}^{\text{out}}, \quad (5)$$

где \vec{x}^{in} определяет вектор состояний нейронов входного слоя. По сути, это значения, соответствующие входному изображению, представленные в виде вектора длиной 784. Оператор $\hat{\mathbf{N}}$ отвечает за введение шумового воздействия, упомянутого в разделе 1.2. Настолько упрощенная сеть, по сути, является линейным отображением, и её использование здесь вызвано необходимостью сравнения влияния шума в выходном слое со свёрточным слоем в последующих разделах.

Нами было обучено четыре сети. Обучение проводилось при помощи свободно распространяемой библиотеки Keras [31] на языке программирования Python, используя оптимизатор “adam” и категориальную кросс-энтропию. Обучение проводилось в течение 100 эпох. Для четырех сетей конечная точность работы на обучающих данных составила 93.85%, 93.92%, 93.93%, 93.54%, а на проверочных — 92.34%, 92.55%, 92.49%, 92.51%. Шумовое воздействие вводилось в уже обученную сеть в выходной слой. На рис. 2 приведены графики того, как меняется точность работы этих сетей на проверочных данных при увеличении интенсивности шума.

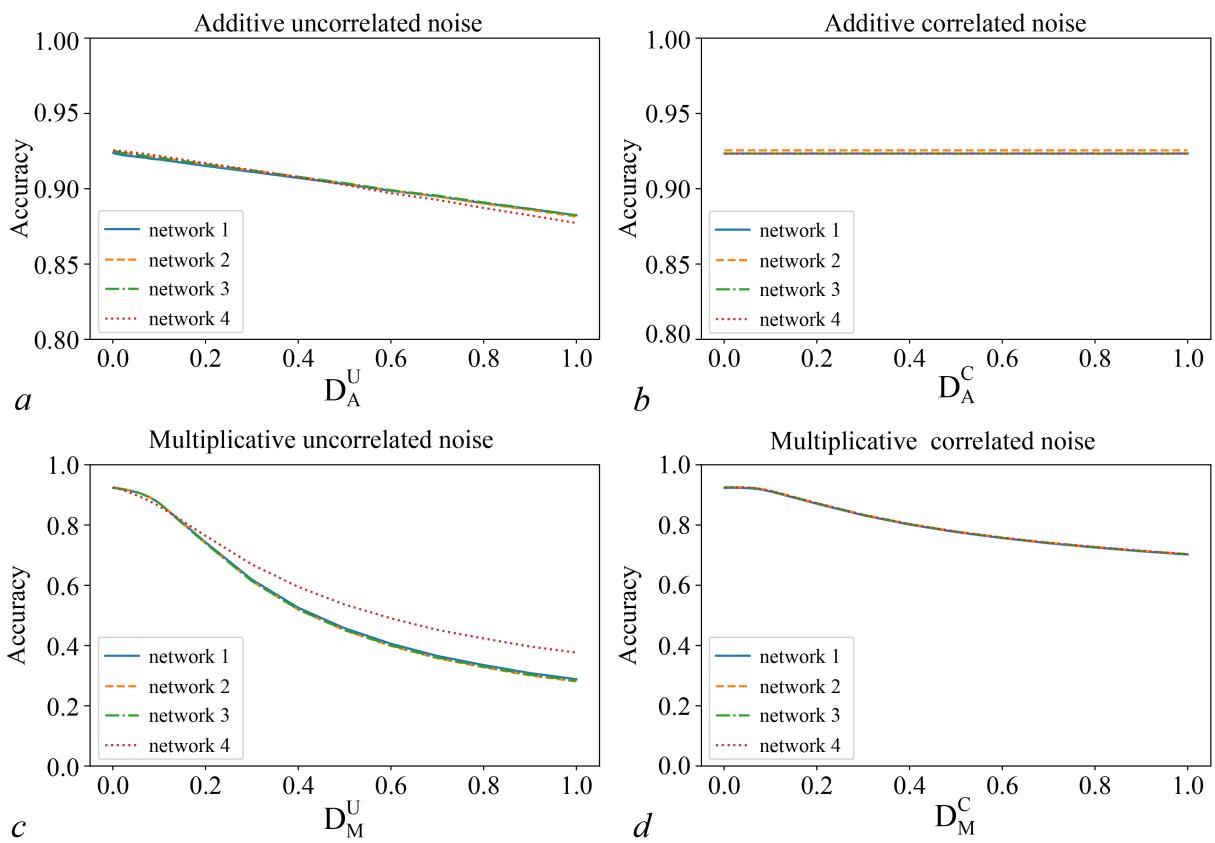


Рис. 2. Изменение точности работы сети, приведенной на рис. 1, *a*, в зависимости от интенсивности шумового воздействия, введенного в выходной слой сети. Были рассмотрены следующие способы введения шумового воздействия: *a* — аддитивный некоррелированный шум, *b* — аддитивный коррелированный шум, *c* — мультипликативный некоррелированный шум, *d* — мультипликативный коррелированный шум. Цветом показаны графики, полученные для разных нейронных сетей, которые были обучены до введения шумового воздействия (цвет онлайн)

Fig. 2. Change in the accuracy of the network shown in Fig. 1, *a*, depending on the intensity of the noise exposure introduced into the output layer of the network. The following methods of introducing noise were considered: *a* — additive uncorrelated noise, *b* — additive correlated noise, *c* — multiplicative uncorrelated noise, *d* — multiplicative correlated noise. The color shows the graphs obtained for different neural networks that were trained before introducing noise exposure (color online)

Для всех четырех обученных сетей зависимости практически совпадают. Аддитивный некоррелированный шум (рис. 2, *a*) незначительно влияет на точность работы сети. Точность линейно падает с увеличением интенсивности шума. Даже при интенсивности шума $D_A^U = 1$, которая сравнительно большая для рассматриваемого диапазона входных значений сети $[0, 1]$, точность падает примерно до 88%. Из-за решения задачи классификации и того, что не так сильно важен сам выходной сигнал нейронов, сколько то, у какого именно нейрона максимальный выходной сигнал, аддитивный коррелированный шум (рис. 2, *b*) в выходном слое никак не влияет на конечную точность. По этой же причине влияние мультипликативного коррелированного шума (рис. 2, *d*) не так критично, как мультипликативный некоррелированный шум (рис. 2, *c*). Самым критичным в выходном слое является мультипликативный некоррелированный шум. Точность при увеличении интенсивности этого шума спадает нелинейно, и при $D_M^U = 1$ точность составила примерно 30%.

На графиках приведены результаты для четырех обученных сетей. В результате их обучения статистика матриц связи \mathbf{W}^{out} в сетях 1–3 оказалась примерно похожей, а для четвертой сети

Таблица 1. Статистические характеристики матриц связи \mathbf{W}^{out} для четырех сетей, состоящих из двух полносвязанных слоев (рис. 1, a)
Table 1. Statistical characteristics of connection matrices \mathbf{W}^{out} for all four trained networks containing two fully connected layers (Fig. 1, a)

	сеть 1	сеть 2	сеть 3	сеть 4
Точность (обуч.)	93.85%	93.92%	93.93%	93.54%
Точность (проверки)	92.34%	92.55%	92.49%	92.51%
Среднее \mathbf{W}^{out}	-0.6036	-0.6022	-0.6038	-0.2257
Квадрат среднего \mathbf{W}^{out}	0.3643	0.3627	0.3646	0.0509
Среднее квадратическое \mathbf{W}^{out}	1.7682	1.7656	1.7688	0.6238

статистические характеристики существенно отличаются (см. табл. 1). В рассмотренном типе сетей шум вводился уже в выходной слой после применения всех матрицы связи, поэтому статистика матриц связи может отразиться только на диапазоне выходных значений сети. Поэтому на рис. 2 все соответствующие зависимости для всех обученных сетей качественно одинаковые, а небольшие количественные отличия можно увидеть только для четвертой сети.

3. Шум в свёрточном слое сети

В данном разделе будет рассмотрено, как внутренний шум в свёрточном слое влияет на точность работы сети. Для этого будет обучено несколько сетей, похожих на рис. 1, b. Они отличаются от сетей, рассмотренных в предыдущем разделе тем, что добавлен один свёрточный слой. Для сравнения рассматриваются 4 типа сетей: 1 фильтр в свёрточном слое размером 3×3 , 1 фильтр размером 5×5 , 2 фильтра размером 3×3 , 5 фильтров размером 3×3 . Шум вводился в уже обученные сети в нейроны после этапа свёртки.

Применение шумового воздействия перед выходным слоем аналогично случаю, когда в аппаратной нейронной сети функция softmax реализуется уже после считывания сигнала с аппаратных нейронов, то есть связь между нейронами и сами нейроны реализованы физически, а финальный ответ сети получается уже после анализа полученных 10 выходных сигналов сети.

В сети, изображенной на рис. 1, b, выходной сигнал нейронов свёрточного слоя в общем виде определяется как

$$\bar{x}^{\text{conv}} = f(\bar{x}^{\text{in}} \cdot \mathbf{W}^{\text{conv}}), \quad \bar{y}^{\text{conv}} = \hat{\mathbf{N}} \bar{x}^{\text{conv}}. \quad (6)$$

Здесь функция $f(\cdot)$ — это функция активации. В данной статье рассматривается линейная (тождественная) функция активации $f(x) = x$. В наших предыдущих работах было показано, что нелинейная функция активации может оказывать существенное влияние на накопление шума [23, 27], поэтому в данной работе рассматриваются только упрощенные сети с линейной функцией активации. Влияние нелинейности, разных функций активации и чередование свёрточных слоев на накопление шума будет рассмотрено в следующих работах. Тем не менее в экспериментах с фотонными нейронными сетями часто используются кусочно-линейные функции типа ReLU [32, 33]. Матрица \mathbf{W}^{conv} представляет собой матрицу, связывающую нейроны входного слоя с нейронами свёрточного слоя. Содержимое этой матрицы соответствует процедуре, описанной в разделе 1.1 для сети со свёрточным слоем. Процедура представления операции свёртки в виде особой связи между двумя полносвязанными слоями сети и матричного умножения детально описана в книге [30] и заключается в особом заполнении матрицы связи значениями фильтров свёртки. В данном разделе рассматриваются 4 типа свёрточного слоя: 1 фильтр 3×3 ,

Таблица 2. Различные комбинации фильтров 3×3 для рассмотренных обученных сверточных сетей рис. 1, b

Table 2. Different combinations of filters of size 3×3 in considered trained convolutional networks from Fig. 1, b

1 фильтр 3×3			2 фильтра 3×3					
-0.60	0.11	0.35	-0.13	-0.28	0.35	0.23	-0.10	0.05
1.24	0.55	-0.33	1.06	0.09	-0.07	-0.84	0.69	-0.28
-1.34	0.85	-0.15	-0.58	-0.20	0.43	0.12	-0.36	0.45
5 фильтров 3×3								
0.11	0.12	-0.04	0.04	-0.07	0.10	0.03	0.06	0.03
0.05	-0.33	0.45	0.06	-0.02	-0.004	-0.18	0.26	-0.21
0.17	0.22	-0.28	0.06	-0.13	0.08	-0.06	-0.02	0.16
0.11	-0.34	0.31	0.29	-0.18	0.06			
0.06	0.24	-0.54	-0.34	-0.16	-0.12			
0.04	-0.25	0.36	-0.16	0.45	-0.28			

Таблица 3. Фильтр 5×5 одной из обученных сверточных сетей рис. 1, b

Table 3. Filter of size 5×5 in one of trained convolutional networks from Fig. 1, b

1 фильтр 5×5				
-0.10	-0.25	-0.03	0.36	-0.07
0.21	0.44	-0.61	-0.09	0.31
-0.29	0.20	-0.67	0.57	-0.36
0.35	-0.59	0.48	-0.80	-0.50
0.18	0.01	-0.50	0.65	0.36

2 фильтра 3×3 , 5 фильтров 3×3 и 1 фильтр 5×5 . Сами значения, полученные для этих фильтров после обучения сетей, приведены в табл. 2 и 3.

После применения свёртки общий выходной сигнал сети определяется как

$$\bar{y}^{\text{out}} = \bar{y}^{\text{conv}} \cdot \mathbf{W}^{\text{out}}. \quad (7)$$

На рис. 3 приведены зависимости точности от интенсивности 4 типов шума для перечисленных выше четырех обученных сетей. Из графиков видно, что нет однозначной корреляции между снижением точности работы сети и количеством фильтров или их размером.

Исходя из наших предыдущих результатов для глубоких сетей, опубликованных в [23], наибольшее влияние на распространение различных шумов оказывают статистические характеристики матрицы связи, которая идет после слоя с внутренним шумом. В нашем случае это матрица \mathbf{W}^{out} . Влияние коррелированного шума определялось квадратом среднего матрицы связи, а влияние некоррелированного определялось средним квадратическим этой матрицы. Для всех рассмотренных сетей эта информация приведена в табл. 4.

Аддитивный некоррелированный шум (рис. 3, a) в свёрточном слое приводит к более выраженному падению точности, чем в выходном слое. То же самое можно сказать и про аддитивный коррелированный шум (рис. 3, b). Если в выходном слое даже большие интенсивности не приводили к ухудшению работы сети, то аналогичный шум в свёрточном слое приводит к существенному падению точности вплоть до полной потери возможности распознавания цифр.

Мультипликативный некоррелированный шум (рис. 3, c) линейно уменьшает точность работы сети при увеличении интенсивности шума. Для шума в выходном слое аналогичная зависимость была нелинейной. Более того, из сравнения рис. 3, c и рис. 2, c видно, что графики, полученные для сетей со свёрточным слоем, находятся существенно выше, что говорит о том, что сеть сама может подавлять мультипликативный некоррелированный шум. Ранее для глубоких

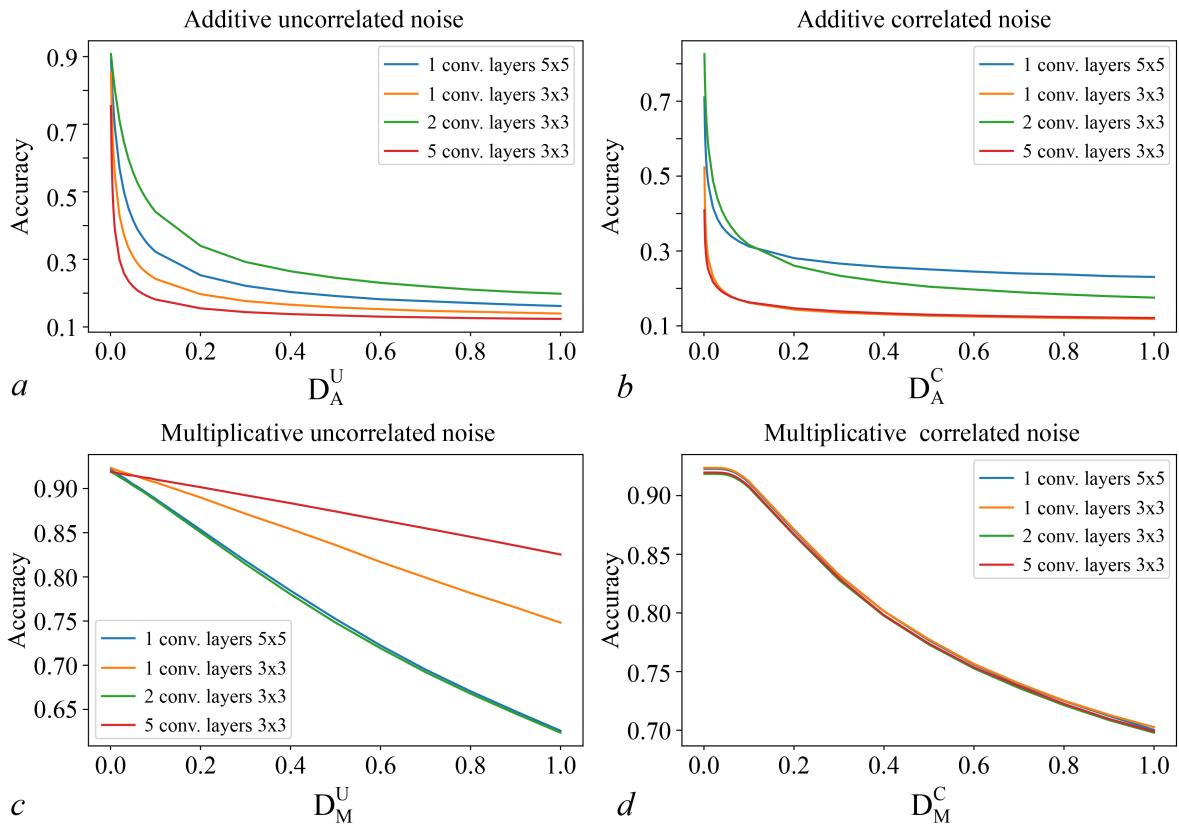


Рис. 3. Изменение точности работы сети, приведенной на рис. 1, b, в зависимости от интенсивности шумового воздействия, введенного в свёрточный слой сети. Были рассмотрены следующие конфигурации свёрточного слоя: 1 фильтр 5×5 (синий), 1 фильтр 3×3 (оранжевый), 2 фильтра 3×3 (зеленый), 5 фильтров 3×3 (красный). Способы введения шумового воздействия: a — аддитивный некоррелированный шум, b — аддитивный коррелированный шум, c — мультипликативный некоррелированный шум, d — мультипликативный коррелированный шум (цвет онлайн)

Fig. 3. Change in the accuracy of the network shown in Fig. 1, b, depending on the intensity of the noise exposure introduced into the convolutional layer of the network. The following configurations of the convolutional layer were considered: 1 filter 5×5 (blue), 1 filter 3×3 (orange), 2 filters 3×3 (green), 5 filters 3×3 (red). Methods of introducing noise exposure: a — additive uncorrelated noise, b — additive correlated noise, c — multiplicative uncorrelated noise, d — multiplicative correlated noise (color online)

Таблица 4. Статистические характеристики матриц связи \mathbf{W}^{out} для сетей со свёрточным слоем (рис. 1, b)

Table 4. Statistical characteristics of connection matrices \mathbf{W}^{out} for networks with convolutional layer (Fig. 1, b)

	1 фильтр 5×5	1 фильтр 3×3	2 фильтра 3×3	5 фильтров 3×3
Точность (обуч.)	93.73%	93.96%	94.09%	93.85%
Точность (пров.)	91.85%	92.28%	92.37%	91.97%
Среднее \mathbf{W}^{out}	0.0913	-0.1565	-0.1863	0.0316
Квадрат среднего \mathbf{W}^{out}	0.0083	0.0245	0.0347	0.0010
Среднее квадратическое \mathbf{W}^{out}	0.7273	1.1017	1.2873	1.2322

сетей было обнаружено, что мультиплексивный некоррелированный шум распространяется меньше, если у следующей матрицы связи меньше среднее квадратическое. Сравнивая значения последних строк в табл. 1 и 4, видно, что при использовании свёрточного слоя значения среднего квадратического матрицы \mathbf{W}^{out} на порядок меньше, что подтверждает наше предположение.

Из графиков рис. 3, *d* видно, что независимо от матриц связи и того, какая сеть рассматривается, мультиплексивный коррелированный шум влияет на сети одинаково. Полученная зависимость точности работы сети от интенсивности коррелированного мультиплексивного шума качественно и количественно похожа на то, что было получено для сети без свёрточного слоя рис. 2, *d*.

Таким образом, наиболее критичным шумом можно считать аддитивный шум в свёрточном слое сети, он приводит к существенному и наиболее стремительному падению точности работы сети. Это касается как коррелированного, так и некоррелированного аддитивного шума.

4. Шум в свёрточном слое при наличии слоя субдискретизации

В данном разделе рассматривается свёрточная сеть, состоящая из слоя свёртки и слоя субдискретизации, схематично показанная на рис. 1, *c*. В сети, изображенной на рис. 1, *c*, выходной сигнал свёрточного слоя определяется так же, как для предыдущей сети (6), и шум вводится также — в свёрточный слой. Операция субдискретизации может быть представлена в виде матричного умножения, и тогда общий выходной сигнал сети:

$$\bar{y}^{\text{out}} = \left(\bar{y}^{\text{conv}} \cdot \mathbf{W}^{\text{pool}} \right) \cdot \mathbf{W}^{\text{out}}. \quad (8)$$

Здесь процедура субдискретизации заменена матричным умножением с использованием матрицы \mathbf{W}^{pool} . Подробное описание процедуры субдискретизации было приведено ранее в разделе 1.1. В данном же разделе будут рассмотрены результаты численного моделирования для обученной сети с фильтром 2×2 в слое субдискретизации с использованием MeanPooling и MaxPooling.

Из рис. 4 видно, что аддитивный некоррелированный шум удается существенно уменьшить при помощи использования MeanPooling в слое субдискретизации (рис. 4, *a*). Это довольно закономерный результат, так как в статье [26] мы предлагали технику пулов для уменьшения влияния некоррелированных шумов. Суть метода заключалась в создании дубликатов нейронов и последующем усреднении. По сути, это очень похоже на процедуру, которая происходит при применении слоя субдискретизации с усреднением (MeanPooling). В случае с MaxPooling ситуация не такая однозначная. На рис. 4, *a* голубой график, который соответствует использованию MaxPooling, находится выше красного графика, который соответствует сети без слоя субдискретизации. Однако в некоторых случаях графики могут пересекаться, и в целом использование MaxPooling при аддитивном некоррелированном шуме нельзя считать способом уменьшения шума.

В случае с аддитивным коррелированным шумом (рис. 4, *b*) видно, что оба способа введения слоя субдискретизации позволяют уменьшить влияние шума. Этот результат был получен много раз для разных обученных сетей.

При наличии в свёрточном слое мультиплексивного коррелированного шума (рис. 4, *d*) значительного улучшения работы сети можно добиться, только если используется MaxPooling. Если же шум мультиплексивный некоррелированный, то MaxPooling улучшает работу сети только при малых значениях интенсивности шума. При увеличении интенсивности шума больше 0.1 точность работы сети существенно снижается и становится хуже, чем в сетях без слоя субдискретизации. Для обоих мультиплексивных шумов применение MeanPooling в слое субдискретизации приводит к тем же значениям точности, что и в сетях без этого слоя.

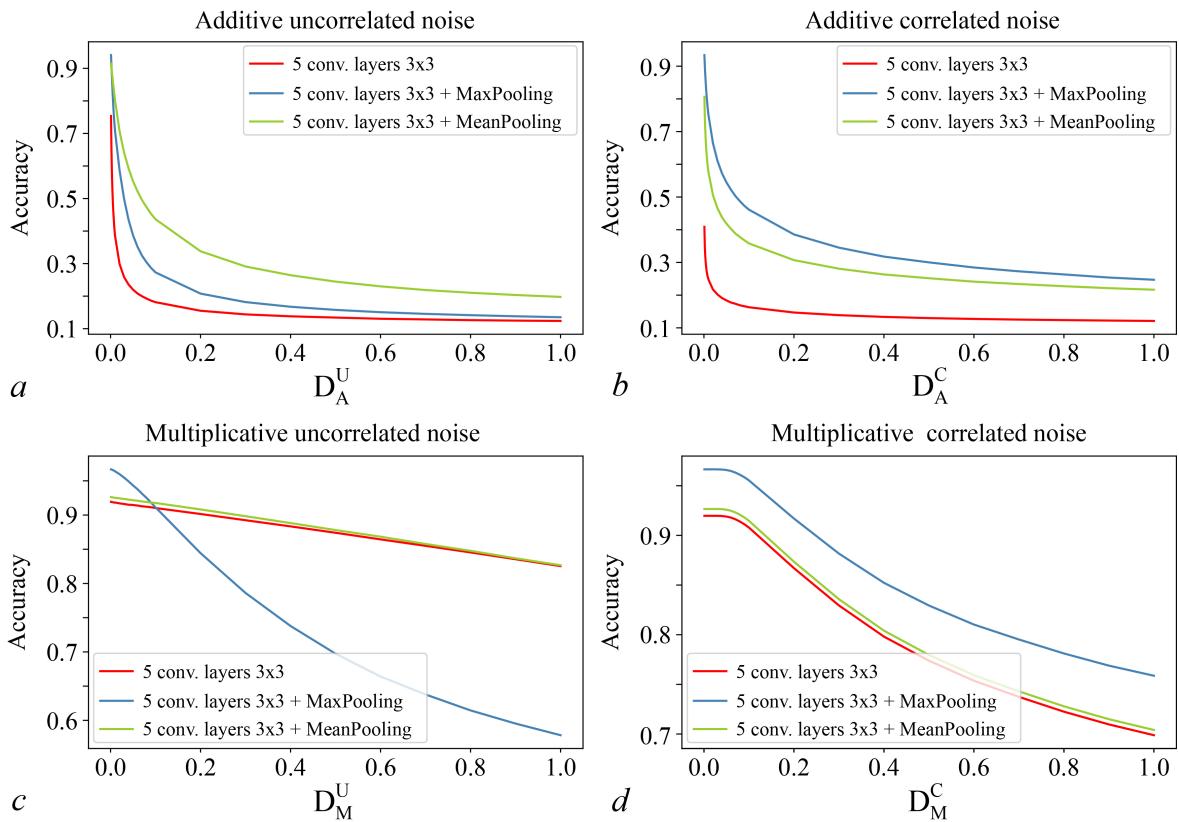


Рис. 4. Изменение точности работы сети, приведенной на рис. 1, *c*, в зависимости от интенсивности шумового воздействия, введенного в свёрточный слой сети. На данном рисунке приведен случай, когда в свёрточном слое находится 5 фильтров размером 3×3 . Красным цветом показаны зависимости, полученные без использования слоя субдискретизации после слоя свёртки. Синие и зеленые графики соответствуют слою субдискретизации с использованием MaxPooling и MeanPooling соответственно. Способы введения шумового воздействия: *a* — аддитивный некоррелированный шум, *b* — аддитивный коррелированный шум, *c* — мультипликативный некоррелированный шум, *d* — мультипликативный коррелированный шум (цвет онлайн)

Fig. 4. Change in the accuracy of the network shown in Fig. 1, *c*, depending on the intensity of the noise exposure introduced into the convolutional layer of the network. This figure shows the case when the convolutional layer contains 5 filters of size 3×3 . The dependencies obtained without using a pooling layer after the convolution layer are shown in red. Blue and green graphs correspond to the pooling layer using MaxPooling and MeanPooling, respectively. Methods of introducing noise effect: *a* — additive uncorrelated noise, *b* — additive correlated noise, *c* — multiplicative uncorrelated noise, *d* — multiplicative correlated noise (color online)

Заключение

В данной статье было рассмотрено влияние внутренних шумов на работу свёрточной нейронной сети. Были рассмотрены сети с разными компонентами, которые часто встречаются именно в свёрточных сетях. В разделе 2 была рассмотрена простая полносвязная сеть, и шум вводился в выходной слой. Далее в сеть был добавлен свёрточный слой с различным числом фильтров и их размерами (раздел 3). При этом шум вводился в скрытый слой сети, который получался в результате применения свёртки. Было показано, что аддитивные шумы (как коррелированный, так и некоррелированный) при этом накапливаются сильнее, чем в сети без свёрточного слоя. Зависимости точности работы сети от интенсивности мультипликативного коррелированного шума выглядят одинаково для сетей со свёрточным слоем и без, а аналогичные зависимости для мультипликативного некоррелированного шума выглядят гораздо лучше для сети со свёрточным

слоем. Было обучено несколько сетей со свёрточными слоями, и качественно результаты для них похожи. Однако количественные результаты, а именно кривые зависимости точности работы сети от интенсивности шумового воздействия для каких фильтров располагаются выше или ниже, чаще всего вызвано другими статистическими характеристиками матрицы связи \mathbf{W}^{out} , которая применяется к нейронам с внутренними шумами.

Использование слоя субдискретизации после слоя свёртки позволяет несколько улучшить работу сети и ее устойчивость к внутренним шумам (раздел 4). Так, использование MeanPooling и MaxPooling позволяет существенно поднять точность при наличии аддитивного шума в свёрточном слое. Что же касается мультиплекативного шума, то здесь ситуация не такая однозначная. Использование MeanPooling приводит примерно к тому же результату, что и отсутствие слоя субдискретизации. Для мультиплекативного коррелированного шума сеть с MaxPooling обладает изначально большей точностью, а скорость спадания точности при увеличении интенсивности шума примерно такая же, как для сети с MeanPooling или для сети без слоя субдискретизации. Что же касается мультиплекативного некоррелированного шума, использование MaxPooling в этом случае делает сеть менее устойчивой к шуму.

Список литературы

1. *LeCun Y., Bengio Y., Hinton G.* Deep learning // *Nature*. 2015. Vol. 521, no. 7553. P. 436–444. DOI: 10.1038/nature14539.
2. *Krizhevsky A., Sutskever I., Hinton G. E.* ImageNet classification with deep convolutional neural networks // *Commun. ACM*. 2017. Vol. 60, no. 6. P. 84–90. DOI: 10.1145/3065386.
3. *Maturana D., Scherer S.* VoxNet: A 3D Convolutional Neural Network for real-time object recognition // In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2015, Hamburg, Germany. 2015. P. 922–928. DOI: 10.1109/IROS.2015.7353481.
4. *Graves A., Mohamed A., Hinton G.* Speech recognition with deep recurrent neural networks // In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013, Vancouver, BC, Canada. 2013. P. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.
5. *Kar S., Moura J. M. F.* Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise // *IEEE Transactions on Signal Processing*. 2009. Vol. 57, no. 1. P. 355–369. DOI: 10.1109/TSP.2008.2007111.
6. AI and compute. [Electronic resource]. 2018. Available from: <https://openai.com/index/ai-and-compute/>.
7. *Hasler J., Marr H. B.* Finding a roadmap to achieve large neuromorphic hardware systems // *Front. Neurosci.* 2013. Vol. 7. P. 118. DOI: 10.3389/fnins.2013.00118.
8. *Gupta S., Agrawal A., Gopalakrishnan K., Narayanan P.* Deep Learning with Limited Numerical Precision // In: Proceedings of the 32nd International Conference on International Conference on Machine Learning. 2015. Vol. 37. P. 1737–1746. DOI: 10.1109/72.80206.
9. *Karniadakis G. E., Kevrekidis I. G., Lu L., Perdikaris P., Wang S., Yang L.* Physics-informed machine learning // *Nat. Rev. Phys.* 2021. Vol. 3. P. 422–440. DOI: 10.1038/s42254-021-00314-5.
10. *Aguirre F., Sebastian A., Le Gallo M., Song W., Wang T., Yang J. J., Lu W., Chang M.-F., Ielmini D., Yang Y., Mehonic A., Kenyon A., Villena M. A., Roldán J. B., Wu Y., Hsu Hu.-H., Raghavan N., Suñé J., Miranda E., Eltawil A., Setti G., Smagulova K., Salama K. N., Krestinskaya O., Yan X., Ang K.-W., Jain S., Li S., Alharbi O., Pazos S., Lanza M.* Hardware implementation of memristor-based artificial neural networks // *Nat. Commun.* 2024. Vol. 15. P. 1974. DOI: 10.1038/s41467-024-45670-9.
11. *Chen Y., Nazhamaiti M., Xu H., Meng Y., Zhou T., Li G., Fan J., Wei Q., Wu J., Qiao F., Fang L., Dai Q.* All-analog photoelectronic chip for high-speed vision tasks // *Nature*. 2023. Vol. 623. P. 48–57. DOI: 10.1038/s41586-023-06558-8.

12. *Brunner D., Soriano M. C., Mirasso C. R., Fischer I.* Parallel photonic information processing at gigabyte per second data rates using transient states // *Nat. Commun.* 2023. Vol. 4. P. 1364. DOI: 10.1038/ncomms2368.
13. *Tuma T., Pantazi A., Le Gallo M., Sebastian A., Eleftheriou E.* Stochastic phase-change neurons // *Nature Nanotech.* 2016. Vol. 11. P. 693–699. DOI: 10.1038/nnano.2016.70.
14. *Torrejon J., Riou M., Araujo F., Tsunegi S., Khalsa G., Querlioz D., Bortolotti P., Cros V., Yakushiji K., Fukushima A., Kubota H., Yuasa Sh., Stiles M. D., Grollier J.* Neuromorphic computing with nanoscale spintronic oscillators // *Nature.* 2017. Vol. 547. P. 428–431. DOI: 10.1038/nature23011.
15. *Psaltis D., Brady D., Gu X. G., Lin S.* Holography in artificial neural networks // *Nature.* 1990. Vol. 343. P. 325–330. DOI: 10.1038/343325a0.
16. *Bueno J., Maktoobi S., Froehly L., Fischer I., Jacquot M., Larger L., Brunner D.* Reinforcement learning in a large-scale photonic recurrent neural network // *Optica.* 2018. Vol. 5, no. 6. P. 756–760. DOI: 10.1364/OPTICA.5.000756.
17. *Lin X., Rivenson Y., Yardimci N. T., Veli M., Jarrahi M., Ozcan A.* All-optical machine learning using diffractive deep neural networks // *Science.* 2018. Vol. 361. P. 1004–1008. DOI: 10.1126/science.aat8084.
18. *Shen Y., Harris N. C., Skirlo S., Prabhu M., Baehr-Jones T., Hochberg M., Sun X., Zhao S., Larochelle H., Englund D., Soljacic M.* Deep learning with coherent nanophotonic circuits // *Nature Photonics.* 2017. Vol. 11. P. 441–446. DOI: 10.1038/nphoton.2017.93.
19. *Tait A. N., De Lima T. F., Zhou E., Wu A. X., Nahmias M. A., Shastri B. J., Prucnal P. R.* Neuromorphic photonic networks using silicon photonic weight banks // *Sci. Rep.* 2017. Vol. 7, no. 1. P. 7430. DOI: 10.1038/s41598-017-07754-z.
20. *Moughames J., Porte X., Thiel M., Ulliac G., Larger L., Jacquot M., Kadic M., Brunner D.* Three-dimensional waveguide interconnects for scalable integration of photonic neural networks // *Optica.* 2020. Vol. 7, no. 6. P. 640–646. DOI: 10.1364/OPTICA.388205.
21. *Dinc N. U., Psaltis D., Brunner D.* Optical neural networks: The 3D connection // *Photoniques.* 2020. Vol. 104. P. 34–38. DOI: 10.1051/photon/202010434.
22. *Moughames J., Porte X., Larger L., Jacquot M., Kadic M., Brunner D.* 3D printed multimode-splitters for photonic interconnects // *Optical Materials Express.* 2020. Vol. 10, no. 11. P. 2952–2961. DOI: 10.1364/OME.402974.
23. *Semenova N., Larger L., Brunner D.* Understanding and mitigating noise in trained deep neural networks // *Neural Netw.* 2022. Vol. 146. P. 151–160. DOI: 10.1016/j.neunet.2021.11.008.
24. *Semenova N.* Impact of white Gaussian internal noise on analog echo-state neural networks // arXiv:2405.07670. arXiv Preprint, 2024. 10 p. DOI: 10.48550/arXiv.2405.07670.
25. *Semenova N., Brunner D.* Noise-mitigation strategies in physical feedforward neural networks // *Chaos.* 2022. Vol. 32, no. 6. P. 061106. DOI: 10.1063/5.0096637.
26. *Semenova N., Brunner D.* Impact of white noise in artificial neural networks trained for classification: Performance and noise mitigation strategies // *Chaos.* 2024. Vol. 34, no. 5. P. 051101. DOI: 10.1063/5.0206807.
27. *Semenova N., Porte X., Andreoli L., Jacquot M., Larger L., Brunner D.* Fundamental aspects of noise in analog-hardware neural networks // *Chaos.* 2019. Vol. 29, no. 20. P. 103128. DOI: 10.1063/1.5120824.
28. *Li Z., Liu F., Yang W., Peng S., Zhou J.* A survey of convolutional neural networks: Analysis, applications, and prospects // *IEEE Transactions on Neural Networks and Learning Systems.* 2022. Vol. 33, no. 12. P. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.
29. *LeCun Y.* The MNIST database of handwritten digits [Electronic resource] // 1998. Available from: <http://yann.lecun.com/exdb/mnist/>.

30. Гудфеллоу Я., Бенджио И., Курвиль А. Глубокое обучение. М.: ДМК Пресс, 2018. 652 с.
31. Chollet F. et al. Keras [Electronic resource]. 2015. Available from: <https://github.com/fchollet/keras>.
32. Stephanie M. V., Pham L., Schindler A., Grasser T., Waltl M., Schrenk B. Photonic neuron with on frequency-domain ReLU activation function // Journal of Lightwave Technology. 2024. Vol. 42, no. 22. P. 7919–7928. DOI: 10.1109/JLT.2024.3413976.
33. Li G. H. Y., Sekine R., Nehra R., Gray R. M., Ledezma L., Guo Q., Marandi A. All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning // Nanophotonics. 2023. Vol. 12, no. 5. P. 847–855. DOI: 10.1515/nanoph-2022-0137.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436–444. DOI: 10.1038/nature14539.
2. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun. ACM*. 2017;60(6):84–90. DOI: 10.1145/3065386.
3. Maturana D, Scherer S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2015, Hamburg, Germany. 2015. P. 922–928. DOI: 10.1109/IROS.2015.7353481.
4. Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013, Vancouver, BC, Canada. 2013. P. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.
5. Kar S, Moura JMF. Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Transactions on Signal Processing*. 2009;57(1):355–369. DOI: 10.1109/TSP.2008.2007111.
6. AI and compute. [Electronic resource]. 2018. Available from: <https://openai.com/index/ai-and-compute/>.
7. Hasler J, Marr HB. Finding a roadmap to achieve large neuromorphic hardware systems. *Front. Neurosci*. 2013;7:118. DOI: 10.3389/fnins.2013.00118.
8. Gupta S, Agrawal A, Gopalakrishnan K, Narayanan P. Deep Learning with Limited Numerical Precision. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning. 2015. Vol. 37. P. 1737–1746. DOI: 10.1109/72.80206.
9. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nat. Rev. Phys.* 2021;3:422–440. DOI: 10.1038/s42254-021-00314-5.
10. Aguirre F, Sebastian A, Le Gallo M, Song W, Wang T, Yang JJ, Lu W, Chang M-F, Ielmini D, Yang Y, Mehonic A, Kenyon A, Villena MA, Roldán JB, Wu Y, Hsu Hu-H, Raghavan N, Suñé J, Miranda E, Eltawil A, Setti G, Smagulova K, Salama KN, Krestinskaya O, Yan X, Ang K-W, Jain S, Li S, Alharbi O, Pazos S, Lanza M. Hardware implementation of memristor-based artificial neural networks. *Nat. Commun.* 2024;15:1974. DOI: 10.1038/s41467-024-45670-9.
11. Chen Y, Nazhamaiti M, Xu H, Meng Y, Zhou T, Li G, Fan J, Wei Q, Wu J, Qiao F, Fang L, Dai Q. All-analog photoelectronic chip for high-speed vision tasks. *Nature*. 2023;623:48–57. DOI: 10.1038/s41586-023-06558-8.
12. Brunner D, Soriano MC, Mirasso CR, Fischer I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* 2023;4:1364. DOI: 10.1038/ncomms2368.
13. Tuma T, Pantazi A, Le Gallo M, Sebastian A, Eleftheriou E. Stochastic phase-change neurons. *Nature Nanotech*. 2016;11:693–699. DOI: 10.1038/nnano.2016.70.
14. Torrejon J, Riou M, Araujo F, Tsunegi S, Khalsa G, Querlioz D, Bortolotti P, Cros V, Yakushiji K,

- Fukushima A, Kubota H, Yuasa Sh, Stiles MD, Grollier J. Neuromorphic computing with nanoscale spintronic oscillators. *Nature*. 2017;547:428–431. DOI: 10.1038/nature23011.
15. Psaltis D, Brady D, Gu XG, Lin S. Holography in artificial neural networks. *Nature*. 1990;343:325–330. DOI: 10.1038/343325a0.
 16. Bueno J, Maktoobi S, Froehly L, Fischer I, Jacquot M, Larger L, Brunner D. Reinforcement learning in a large-scale photonic recurrent neural network. *Optica*. 2018;5(6):756–760. DOI: 10.1364/OPTICA.5.000756
 17. Lin X, Rivenson Y, Yardimci NT, Veli M, Jarrahi M, Ozcan A. All-optical machine learning using diffractive deep neural networks. *Science*. 2018;361:1004–1008. DOI: 10.1126/science.aat8084.
 18. Shen Y, Harris NC, Skirlo S, Prabhu M, Baehr-Jones T, Hochberg M, Sun X, Zhao S, Larochelle H, Englund D, Soljacic M. Deep learning with coherent nanophotonic circuits. *Nature Photonics*. 2017;11:441–446. DOI: 10.1038/nphoton.2017.93.
 19. Tait AN, De Lima TF, Zhou E, Wu AX, Nahmias MA, Shastri BJ, Prucnal PR. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* 2017;7(1):7430. DOI: 10.1038/s41598-017-07754-z.
 20. Moughames J, Porte X, Thiel M, Ulliac G, Larger L, Jacquot M, Kadic M, Brunner D. Three-dimensional waveguide interconnects for scalable integration of photonic neural networks. *Optica*. 2020;7(6):640–646. DOI: 10.1364/OPTICA.388205.
 21. Dinc NU, Psaltis D, Brunner D. Optical neural networks: The 3D connection. *Photoniques*. 2020;104:34–38. DOI: 10.1051/photon/202010434.
 22. Moughames J, Porte X, Larger L, Jacquot M, Kadic M, Brunner D. 3D printed multimode-splitters for photonic interconnects. *Optical Materials Express*. 2020;10(11):2952–2961. DOI: 10.1364/OME.402974.
 23. Semenova N, Larger L, Brunner D. Understanding and mitigating noise in trained deep neural networks. *Neural Netw*. 2022;146:151–160. DOI: 10.1016/j.neunet.2021.11.008.
 24. Semenova N. Impact of white Gaussian internal noise on analog echo-state neural networks. *arXiv:2405.07670*. arXiv Preprint; 2024. 10 p. DOI: 10.48550/arXiv.2405.07670.
 25. Semenova N, Brunner D. Noise-mitigation strategies in physical feedforward neural networks. *Chaos*. 2022;32(6):061106. DOI: 10.1063/5.0096637.
 26. Semenova N, Brunner D. Impact of white noise in artificial neural networks trained for classification: Performance and noise mitigation strategies. *Chaos*. 2024;34(5):051101. DOI: 10.1063/5.0206807.
 27. Semenova N, Porte X, Andreoli L, Jacquot M, Larger L, Brunner D. Fundamental aspects of noise in analog-hardware neural networks. *Chaos*. 2019;29(20):103128. DOI: 10.1063/1.5120824.
 28. Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*. 2022;33(12):6999–7019. DOI: 10.1109/TNNLS.2021.3084827.
 29. LeCun Y. The MNIST database of handwritten digits. [Electronic resource] // 1998. Available from: <http://yann.lecun.com/exdb/mnist/>.
 30. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016. 800 p.
 31. Chollet F et al. Keras. [Electronic resource]. 2015. Available from: <https://github.com/fchollet/keras>.
 32. Stephanie MV, Pham L, Schindler A, Grasser T, Waltl M, Schrenk B. Photonic neuron with on frequency-domain ReLU activation function. *Journal of Lightwave Technology*. 2024;42(22):7919–7928. DOI: 10.1109/JLT.2024.3413976.
 33. Li GHY, Sekine R, Nehra R, Gray RM, Ledezma L, Guo Q, Marandi A. All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning. *Nanophotonics*. 2023;12(5):847–855. DOI: 10.1515/nanoph-2022-0137.

Семенова Надежда Игоревна — родилась в Саратове (1992). Окончила с отличием физический факультет Саратовского государственного университета по специальности «Радиофизика и электроника» (2014). Защищила диссертацию на соискание учёной степени кандидата физико-математических наук по специальности «Радиофизика» (2017, СГУ). Получила PhD степень по специальности «Оптика» в университете Бургундии Франши-Конте (Франция, 2021). С 2021 года работает на кафедре радиофизики и нелинейной динамики Саратовского государственного университета в должности доцента и старшего научного сотрудника. Научные интересы — нейронавка, математическое моделирование, нейронные сети, машинное обучение, влияние шумов, пространственно-временные режимы в ансамблях и сетях, химерные состояния, уединенные состояния, запаздывающая обратная связь. Опубликовала свыше 50 научных статей по указанным направлениям.



Россия, 410012 Саратов, ул. Астраханская, 83
Саратовский национальный исследовательский
государственный университет имени Н. Г. Чернышевского
E-mail: semenovani@sgu.ru
ORCID: 0000-0002-9180-3030
AuthorID (eLibrary.Ru): 850757