

doi: 10.17586/2226-1494-2023-23-6-1162-1170

УДК 004.94

Реализация нейронных сетей в методе многоуровневых компонентных цепей Максим Игоревич Кочергин✉

Томский государственный университет систем управления и радиоэлектроники, Томск, 634050, Российская Федерация

maksim.i.kochergin@tusur.ru✉, <https://orcid.org/0000-0001-7404-8718>

Аннотация

Введение. Проанализированы способы представления искусственных нейронных сетей в пакетах моделирования Simulink и SimInTech. Приведены примеры визуальных схем (моделей), построенных в данных пакетах моделирования с использованием блоков нейронных сетей. Показано, что этим моделям присущи такие недостатки, как отсутствие механизмов проведения структурной оптимизации искусственных нейронных сетей, отсутствие возможности их объединения в ансамбли и обучения нейронных сетей одновременно с работой модели объекта управления. При построении нейросетевых моделей управления затруднено использование специализированных библиотек Python (Keras, PyTorch и др.) и среды NeuroGenetic Optimizer (BioCompSystems). **Метод.** Показан способ реализации моделей искусственных нейронных сетей в формализме метода многоуровневых компонентных цепей. Согласно предложенному способу, построение моделей объекта и системы управления осуществляется на визуальном языке из готовых блоков (компонентов) с направленными и ненаправленными связями. Методика многоуровневого представления нейросетевых моделей управления связывает модели с другими инструментами метода компонентных цепей. **Основные результаты.** Предложены варианты моделей нейронных сетей с инкапсулированной и компонентной структурами. Первый вариант характеризуется компактностью модели управления, возможностью автоматизированной вариации и оптимизации структуры нейронной сети, возможностью изменения структуры сети во время работы модели при вычислительном эксперименте (сценарии). Второй вариант обладает возможностью детальной отладки, исследования процесса обучения сети и конструирования сети любой структурной сложности. Представлено описание связей основных разработанных компонентов — нейронной сети, блоков обучения, объединения нейросетей в ансамбль (бэггинг), чтения данных из файла, формирования выборок и слоев нейронной сети (входной, скрытой, выходной). Работа компонентов проиллюстрирована на реализации многоуровневой компьютерной модели неуправляемого полета тела (цели) и управляемого полета снаряда для решения задачи управления траекторией снаряда с целью поражения цели. **Обсуждение.** Разработанные библиотеки компонентов могут быть использованы в составе среды моделирования MAPS для построения многоуровневых систем управления объектами мультифизической природы.

Ключевые слова

нейронные сети, моделирование, метод компонентных цепей, машинное обучение, Simulink, SimInTech, среда моделирования MAPS

Ссылка для цитирования: Кочергин М.И. Реализация нейронных сетей в методе многоуровневых компонентных цепей // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 6. С. 1162–1170. doi: 10.17586/2226-1494-2023-23-6-1162-1170

Implementation of neural networks in the method of multilevel component circuits Maksim I. Kochergin✉

Tomsk State University of Control Systems and Radioelectronics, Tomsk, 634050, Russian Federation

maksim.i.kochergin@tusur.ru✉, <https://orcid.org/0000-0001-7404-8718>

Abstract

The paper analyzes the features of representing artificial neural networks in Simulink and SimInTech. Examples of visual schemes (models) built in these modeling environments using neural network blocks are given. The following

shortcomings of such representations are the lack of mechanisms: for carrying out structural optimization of neural networks, for combining them into ensembles, for training them synchronously with the simulation of the object model. It was noted that there are difficulties in using other tools, such as specialized Python libraries (Keras, PyTorch, etc.), the NeuroGenetic Optimizer (BioCompSystems) for building neural network control models. A method is shown to implement the representation of neural networks in the formalism of the method of multilevel component circuits, according to which the construction of models of an object and a control system is carried out in a visual language from ready-made blocks (components) with directional and non-directional connections. A technique has been developed for multilevel representation of neural network control models, which allows them to be combined with other tools of the component circuit method. Two options for representing neural networks are proposed: with an encapsulated structure and with a component structure. The first version of the representation is characterized by the compactness of the representation of the control model, the possibility of automated variation and optimization of the structure of the neural network, and the possibility of changing the structure of the network during the executing of the model within a computational experiment (scenario). The second option has the ability to perform detailed debugging and research of the network learning process, and the ability to construct a network of any structural complexity. The paper describes the main developed components with their connections: a neural network, a training block, an ensemble unit (bagging), a block for reading data from a file, a sampling block, a neural network layer (input, hidden, output). A multilevel computer model of the uncontrolled flight of a body (target) and the controlled flight of a projectile is presented as an example to illustrate the operation of the developed components to solve the problem of controlling a projectile to hit the target. The developed component libraries can be used as part of the MARS modeling environment to build multilevel control systems for objects of a multiphysics nature.

Keywords

neural networks, modeling, component circuits method, machine learning, Simulink, SimInTech, simulation environment MARS

For citation: Kochergin M.I. Implementation of neural networks in the method of multilevel component circuits. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 6, pp. 1162–1170 (in Russian). doi: 10.17586/2226-1494-2023-23-6-1162-1170

Введение

Большую роль в моделировании систем управления технологическими процессами и роботизированными системами играют интеллектуальные системы управления (ИСУ) [1]. ИСУ способны моделировать рассуждения человека-оператора во время принятия решения. Существующие в настоящее время комплексы программ не позволяют в полной мере разрабатывать сложные ИСУ. Универсальные пакеты моделирования (Simulink, SimInTech, Wolfram SystemModeler и др.) не дают возможности проектировать и разрабатывать иерархические системы управления [2], осуществляющие интеллектуальное и адаптивное [3] управление сложными динамическими системами. Это связано с тем, что универсальные пакеты предлагают только ограниченный набор методов и решают узкий класс задач. В свою очередь, специализированное программное обеспечение для построения моделей машинного обучения не позволяет моделировать объекты управления с достаточной точностью и не дает возможность отработки проектируемой интеллектуальной системы на уже имеющихся моделях технических объектов или процессов. Например, среда NeuroGenetic Optimizer компании BioCompSystems может по сформированной обучающей выборке «параметры системы — управляющее воздействие (или коэффициенты регулятора)» получить модель нейросетевой системы управления, но имеет сложности с интеграцией полученной модели управления с моделью исследуемого объекта в какой-либо среде или на языке программирования, которая может функционировать одновременно с реальным объектом при построении моделей типа цифрового двойника. С аналогичной проблемой можно столкнуться, используя библиотеки обучения искусственных

нейронных сетей (ИНС) типа Keras, PyTorch языка Python, так как Python не имеет библиотек, требуемых для построения моделей технических, физических или других систем из готовых блоков, и его использование потребует программирования дополнительных процедур (чтение, предобработка данных и др.).

Косвенным подтверждением (кроме непосредственного анализа функциональных возможностей систем) недостатков существующих программно-инструментальных средств является потребность исследователей из разных предметных областей в разработке собственных решений [4–6]. Разработка таких ИСУ «с нуля» приводит к созданию узкоспециализированных систем, направленных на решение конкретной задачи, а их модификация в случае изменения объекта управления или масштабирование являются трудоемкими задачами, вплоть до разработки новых ИСУ, что делает актуальной разработку универсальных средств моделирования ИСУ, в том числе на базе ИНС.

Цель настоящей работы — разработка методики построения многоуровневых нейросетевых систем управления и реализация представления ИНС в формализме метода многоуровневых компонентных цепей (ММКЦ) [7], позволяющего составлять архитектуру ИНС из готовых блоков (компонентов) с направленными связями.

Обзор аналогов

В качестве основных аналогов среди пакетов моделирования с возможностью проектирования ИНС рассмотрим пакеты Simulink (MATLAB) [8] и SimInTech [9].

Расчетная схема нейронной сети в SimInTech представляет собой направленный граф из блоков библиотеки «Нейронные сети». Каждый блок (рис. 1) относится к одному из трех типов: один из слоев ИНС (IN, D и

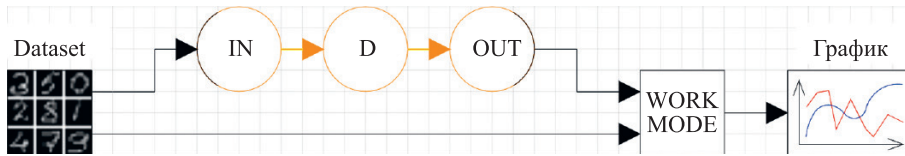


Рис. 1. Нейронная сеть на схеме в SimInTech
 Fig. 1. Neural network on the scheme in SimInTech

OUT — «Входной слой», «Скрытый слой» и «Выходной слой»); Dataset — блок «Источник данных»; WORK MODE — блок «Режим работы», блок «График» для визуализации результатов обучения. Топология ИНС определена связями блоков. Расчетная схема ИНС должна начинаться с блока «Входной слой» и заканчиваться блоком «Выходной слой», но между ними могут следовать любые блоки-слои. По межблочным связям между блоками-слоями передаются не сами значения сигналов, а свойства слоя для подключения следующего слоя к предыдущему.

Блок WORK MODE позволяет выбрать пользователю один из режимов работы нейронной сети («Обучение», «Сравнение», «Определение») и может подавать на выход показатели точности обучения ИНС или непосредственно ее отклик. Так как между слоями ИНС в SimInTech передаются не данные, а свойства сети, такое представление не позволяет отслеживать значения сигналов на выходе какого-либо из слоев ИНС, в связи с чем делает невозможным использование алгоритмов обучения, отсутствующих в библиотеке разработчика, и может затруднять организацию работы нескольких ИНС при объединении их в ансамбли.

Из-за отсутствия блоков для обучения пакет Simulink не позволяет синхронно обучить ИНС с работой модели объекта управления. При этом Simulink доступен импорт в виде subsystems-блоков (рис. 2) любых ИНС, обученных в других модулях MATLAB: Statistics and Machine Learning Toolbox, Deep Learning Toolbox или Neural Network Fitting.

В отличие от пакета SimInTech, данная модель получает данные с каждого слоя, а подключение собственных m-функций на языке MATLAB расширяет возмож-

ности использования ИНС в Simulink в соответствии с задачами исследователя.

Общие недостатки рассмотренных подходов заключаются в отсутствии: механизмов структурной оптимизации ИНС (т. е. автоматизированного подбора конфигурации ИНС (число слоев, количество нейронов в них, вид функции активации) в сторону уменьшения/упрощения с сохранением заданной точности работы (или скорости отклика) или в сторону увеличения для достижения указанной точности); инструментов отбора лучших моделей из числа обученных или объединения результатов их работы в ансамбли; возможности обучения ИНС синхронно с анализом модели объекта управления.

Далее кратко рассмотрим предлагаемое многоуровневое представление ИНС, направленное на устранение перечисленных недостатков.

Многоуровневые компьютерные модели

Приведем основные положения ММКЦ – основы предлагаемого представления ИНС, и его программной реализации в среде моделирования MARC¹ [10].

Каждую модель представим компонентной цепью (КЦ) $C = (K, B, N)$, состоящей из множеств: компонент K , узлов N и ветвей B . Каждый K соответствует объекту, процессу или параметру решаемой задачи, обладает свойствами и описывается математической (на объектном слое) или алгоритмической (на логическом слое) моделью. N в модели соответствуют параметрам и

¹ Ознакомительная версия доступна по ссылке: https://github.com/mikochergin/SE_MARS_demo_doc

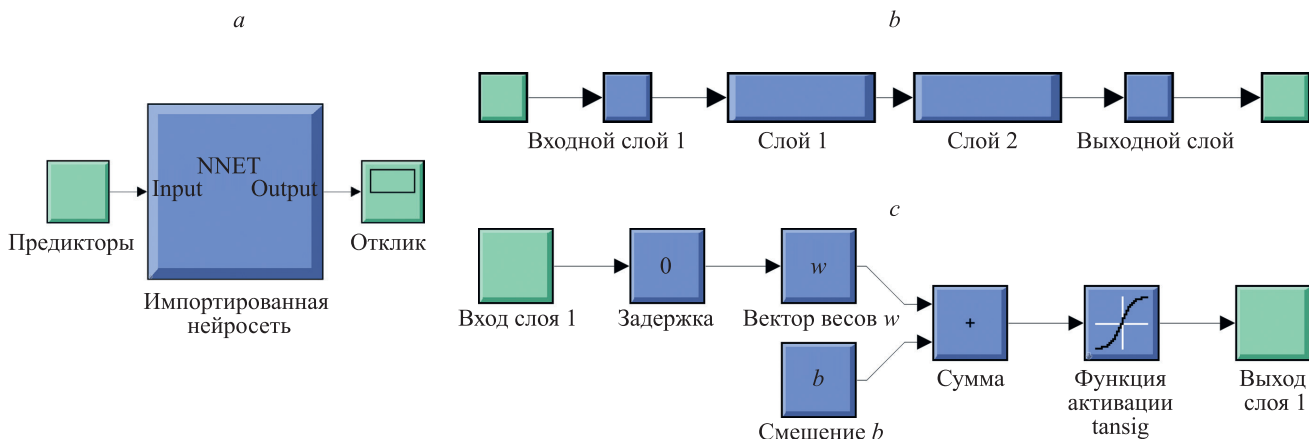


Рис. 2. Импортированный блок нейросети в Simulink (a) и его декомпозиции на слои (b) и на элементы (c):
 input — связь входа, output — связь выхода, NNET — блок нейронной сети

Fig. 2. Imported neural network block in Simulink (a) and its decomposition into layers (b) and elements (c)

переменным. В каждом из N выполняются два топологических закона: закон равенства потенциальных переменных всех связей, подсоединенных к одному узлу, и закон равенства нулю алгебраической суммы потоковых переменных, входящих в один узел. Топологические связи K определяются множеством B . При анализе КЦ вычислительное ядро автоматически составляет системы уравнений на основе математических моделей K и топологических уравнений их связей [7].

Структура многоуровневой компьютерной модели, каждый слой которой содержит свою отдельную КЦ, представлена на рис. 3.

Структуру многоуровневой компьютерной модели определим слоями трех типов.

- *Объектный слой* содержит КЦ, описывающую модель исследуемого объекта, например, электрической цепи, кинематики и динамики робота-манипулятора. КЦ этого слоя имеют ненаправленные связи и состоят из: схематехнических блоков (инерционных звеньев, интеграторов и др.), структурно-физических блоков (труба, насос, сепаратор и др.) или математических панелей для ввода модели в аналитическом виде.
- *Логический слой* содержит КЦ с направленными связями, определяющие сценарий проведения вычислительного эксперимента или обработки его результатов, а также алгоритм поведения объекта, чья модель в виде КЦ располагается на объектном слое. На этом слое осуществляются вспомогательные расчеты и операции.
- *Визуальный слой* содержит интерактивные средства управления моделью (регуляторы, цифровые табло и др.).

Компоненты *логического слоя* относятся к языку моделирования алгоритмических конструкций [11] и взаимодействуют согласно его механизму передачи сообщений: информационные сигналы передаются от источников к приемникам, инициируя поведение компонентов на каждой итерации расчета модели. Таким образом, включение ИНС в этот слой позволило организовать обучение ИНС одновременно с работой модели управления.

Пример многоуровневой компьютерной модели представлен на рис. 4. На *объектном слое* с применением математических панелей заданы модели движения двух объектов, где силы F_x , F_y имитируют воздействие двигателей вдоль осей OX , OY .

Методика построения многоуровневых нейросетевых систем управления

В рамках формализма ММКЦ предложены два варианта моделей ИНС: с инкапсулированной и компонентной структурами, которые позволяют включать вычислительные модели ИНС в контур КЦ управления исследуемым объектом [12, 13].

Первый вариант модели ИНС подразумевает, что исследователь задает структуру ИНС с помощью параметризации одного блока (рис. 5), а второй — формирует структуру ИНС послойно (из нескольких блоков) с заданием их топологии.

Компонент ИНС располагается на *логическом слое № 2*, где он соединен с регулятором [14, 15] (или напрямую с узлом объекта управления) и различными измерителями и средствами визуализации. На этом слое располагаются переключательные модели типа диаграмм состояний [16], определяющие поведение подсистемы ИСУ на данном уровне иерархии, что позволяет переключаться по условию с одной управляющей ИНС на другую. Основная задача слоя — обеспечение работы ИСУ в режиме эксперимента (анализа режимов эксплуатации исследуемого объекта).

Инфраструктура для обучения ИНС, генерации выборок и оценки качества обучения располагается на *логическом слое № 1*. Слой может отсутствовать (или быть не активным), если сеть была обучена заранее и не предполагается ее «дообучение» [17, 18].

Недостаток первой модели — ограничение архитектуры проектируемой ИНС структурой сети прямого распространения. Для другой архитектуры необходима разработка нового типового компонента. Преимущества данной модели ИНС в ММКЦ: компактное представление КЦ управления единым блоком; возможность динамического (во время работы модели) изменения

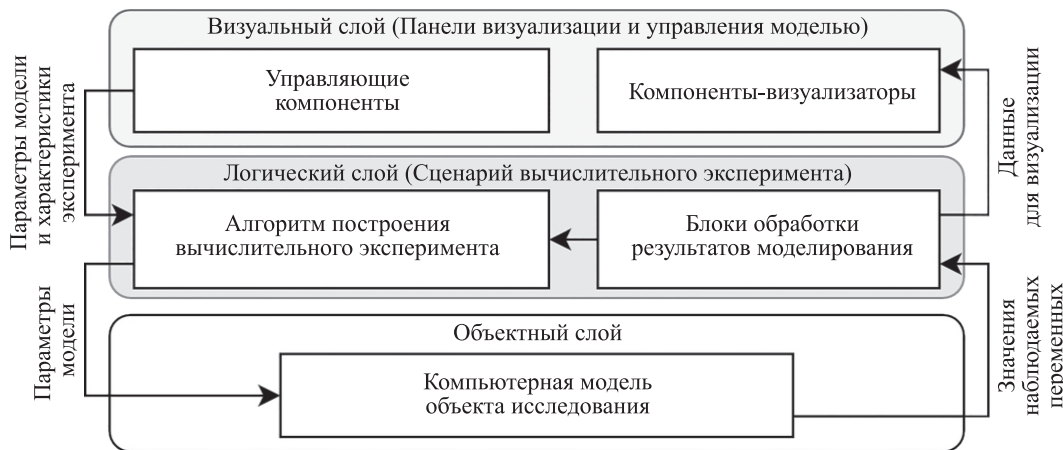


Рис. 3. Структура многоуровневой компьютерной модели

Fig. 3. The structure of a multilevel computer model

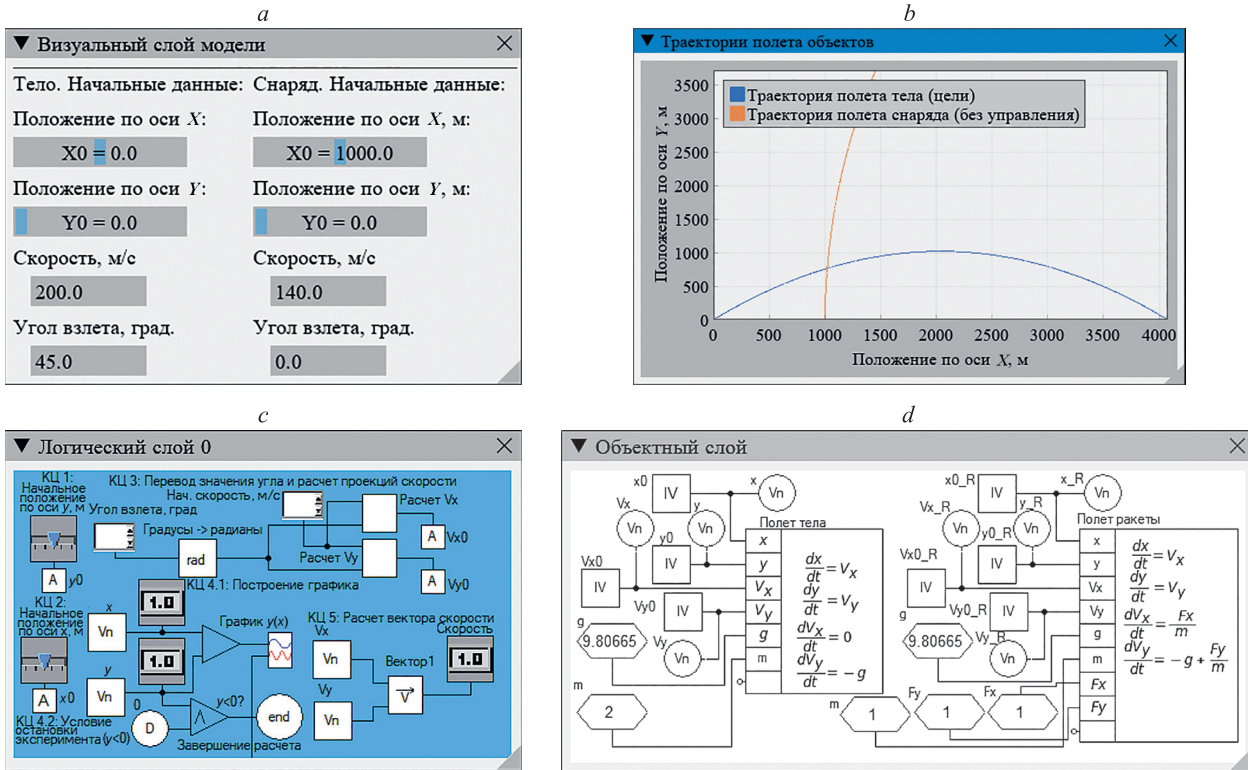


Рис. 4. Многоуровневая компьютерная модель полета тела (цель) и снаряда: интерфейс управления моделью (а); визуализация результатов моделирования (б); вспомогательные расчеты и преобразования (в); модели объектов (д)

Fig. 4. Multilevel computer model of the flight of the body (target) and projectile: model control interface (a); visualization of modeling results (b); supporting calculations and transformations (c); object models (d)

структуры ИНС при добавлении/удалении признаков в выборке данных или в сценариях исследования результатов работы ИНС с различными параметрами; возможность автоматизированной вариации и оптимизации структуры нейронной сети [19, 20].

Второй вариант модели ИНС в ММКЦ — с компонентной структурой, заключается в том, что нейросеть (рис. 5) представлена в декомпозированном виде и составлена исследователем из компонентов типа «Входной слой», «Скрытый слой», «Выходной слой».

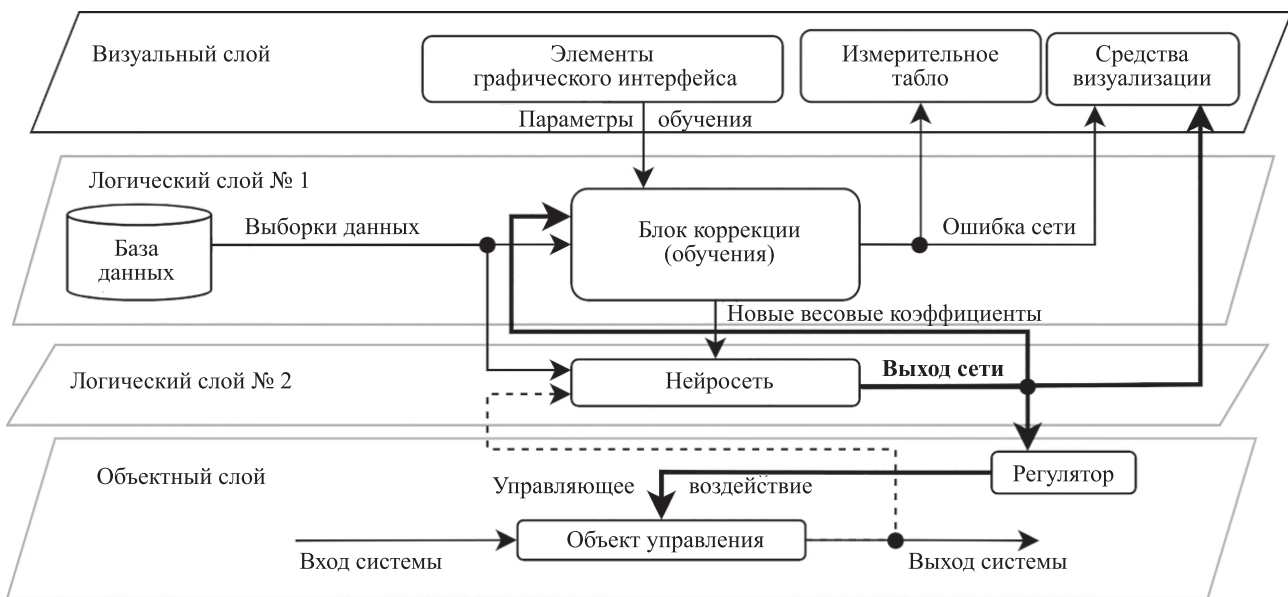


Рис. 5. Концептуальное представление нейросетевой модели управления объектом

Fig. 5. Conceptual representation of a neural network model of object control

Таблица. Описание реализованных компонентов искусственных нейронных сетей
 Table. Description of the implemented components of neural networks

Название	Изображение	Описание
Нейронная сеть (инкапсулированная структура)		Используется для задания архитектуры ИНС и получения отклика на входные данные. Входы: S_0 — для обновленных в процессе обучения весовых коэффициентов (от компонента «Блок обучения»); S_1 — для управляющего сигнала, переводящего ИНС из состояния простоя в состояние работы и наоборот; $S_2 \dots S_n$ — динамически изменяемое количество узлов для входных данных, на которые ИНС формирует отклик. Выход: $S_{n+1} \dots S_{n+m}$ — для сформированного отклика. Параметры: количество слоев и нейронов, список функций активации.
Блок обучения		Используется для обучения ИНС на выборке данных. Входы: S_0, S_1 — для обучающей выборки, S_2, S_3 — для тестовой выборки, S_4 — для сигнала управляющего режимом обучения; $S_7 \dots S_n$ — для реального отклика ИНС. Выход: S_5 — для вывода ошибки; S_6 — для рассчитанных весов ИНС. Параметры: метрика точности, метод оптимизации, количество эпох обучения, batch size, learning rate.
Бэггинг		Используется для построения ансамблей ИНС. Входы: $S_1 \dots S_n$ — для откликов ИНС. Выход: S_0 — для рассчитанного усредненного отклика. Параметры: веса ИНС, режим усреднения (методом большинства голосов — для дискретной величины, среднее арифметическое — для непрерывной величины).
Чтение из файла		Используется для чтения наборов данных из файла. Выход: S_0 — для массива считанных признаков/меток. Параметры: тип переменной (признак — матрица, метка — вектор), номера столбцов для чтения, тип файла.
Формирование выборки		Используется для разбиения наборов данных на обучающую и тестовую выборки. Входы: S_0, S_1 — для считанных значений (признаков и меток). Выходы: S_2, S_3 — для обучающей выборки (признаки и отклик), S_4, S_5 — для тестовой выборки. Параметры: доля обучающей выборки, константа для воспроизведения случайного распределения выборки.
Входной, скрытый, выходные слои (компонентная структура)		Используется для задания архитектуры ИНС, топологии связей и получения отклика на входные данные. Входы: S_0 — для входных данных (признаков), на которые ИНС формирует отклик; S_1 — для обновленных в процессе обучения весовых коэффициентов (от компонента «Блок обучения»); S_3 — для управляющего сигнала, переводящего ИНС из состояния простоя в состояние работы и наоборот (только у компонента «Входной слой»). Выход: S_2 — для сформированного отклика. Параметры: количество нейронов в слое, функции активации.

Такая модель позволяет: строить ИНС произвольной структуры (добавлять обратные связи, параллельно работающие слои); рассматривать ИНС как объект исследования, а не инструмент, что дает возможность оценить ход вычисления отклика, ход ее обучения, выходные значения отдельных слоев и т. д., при этом сохраняя функцию ИНС как системы управления объектом.

Описание разработанных компонентов ИНС в формализме ММКЦ приведено в таблице.

Внешний вид графического интерфейса для настройки компонентов типа «Блок обучения» и «Нейронная сеть» показан на рис. 6.

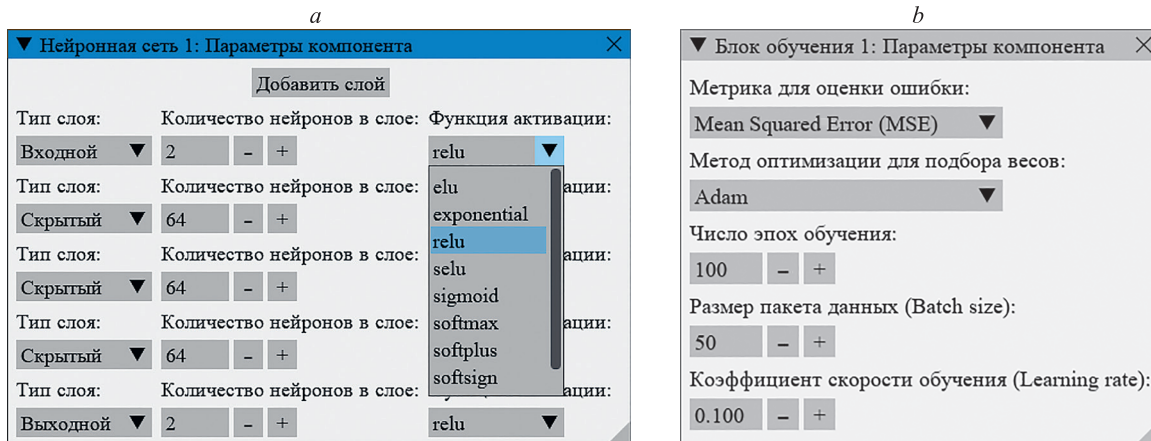


Рис. 6. Окна настройки архитектуры (a) и параметров обучения (b) нейронной сети
 Fig. 6. GUI for setting up the architecture (a) and training parameters (b) of the neural network

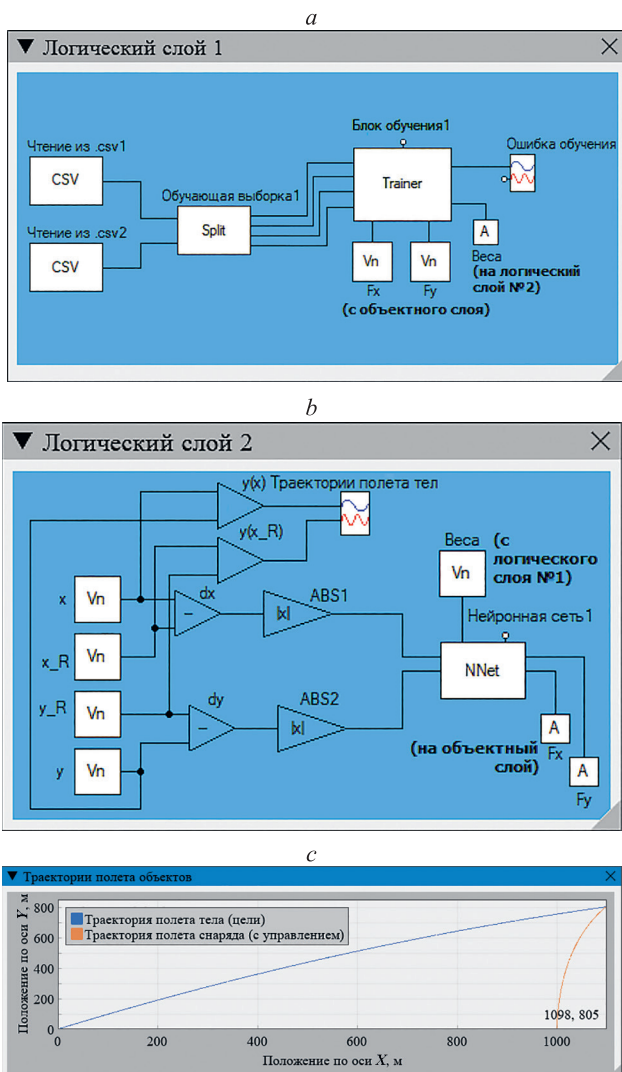


Рис. 7. Фрагмент многоуровневой модели для обучения нейросети: инфраструктура для обучения сети (a); нейросетевая модель управления (b); визуализация результатов моделирования (c)

Fig. 7. Fragment of a multilevel model for training a neural network: infrastructure for network training (a); neural network control model (b); visualization of simulation results (c)

Пример многоуровневой нейросетевой модели управления

На рис. 7 представлено дополнение к многоуровневой модели на рис. 4. Компоненты «Чтение из CSV» (CSV — формат текстового файла с данными) считывают значения независимых переменных (признаков dx , dy — проекций расстояния между телом и снарядом) и зависимых переменных (прогнозируемых сил F_x , F_y) из файлов. Далее компонентом «Выборка» формируются тестовая и обучающая выборки, подаваемые на вход «Блоку обучения», обновляющему веса ИНС. Компонент «Нейронная сеть 1» регулирует значения F_x , F_y , которые на этом слое являются приемниками, а на объектном — источниками и корректируют движение снаряда вдоль осей.

Заключение

Анализ особенностей представления нейронных сетей в современных пакетах моделирования выявил в качестве недостатков отсутствие механизмов: структурной оптимизации нейронных сетей; отбора лучших моделей из числа обученных или объединения их результатов работы в ансамбли; обучения нейросетей синхронно с анализом модели объекта управления.

Предложенная методика многоуровневого моделирования нейросетевых моделей управления позволила включить их компонентные модели в многоуровневый контур управления объектом, в том числе в иерархические системы управления.

В рамках методики разработаны два варианта моделей нейронных сетей: с инкапсулированной структурой, задаваемой параметрами одного блока, и компонентной структурой, задаваемой послойно (из нескольких блоков) с указанием их топологии.

Преимуществами предложенных вариантов моделей являются: наличие возможностей для составления произвольной архитектуры нейронной сети из визуальных блоков; автоматизированные вариации и оптимизации структуры нейронной сети; обучение нейросети на результатах моделирования объекта управления во время расчета его модели (что актуально при взаимодействии

компьютерной модели с реальным объектом, например, при реализации цифровых двойников); построение сложных нейросетевых моделей управления, включая

их объединение в ансамбли; отладка и анализ нейросетевых моделей как объекта исследования.

Литература

1. Корилов А.М. Интеллектуальное управление в технических системах // Научный вестник Новосибирского государственного технического университета. 2014. № 1(54). С. 18–26.
2. Онуфрей А.Ю., Разумов А.В., Какаев В.В. Метод оптимизации структуры в иерархических распределенных системах управления // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23. № 1. С. 44–53. <https://doi.org/10.17586/2226-1494-2023-23-1-44-53>
3. Колесников А.А., Колесников А.А., Кузьменко А.А. Метод АКАР и теория адаптивного управления в задачах синтеза нелинейных систем управления // Мехатроника, автоматизация, управление. 2017. Т. 18. № 9. С. 579–589. <https://doi.org/10.17587/mau.18.579-589>
4. Dayev Z.A., Kairakbayev A.K., Sultanov N.Z. Development of humidity of natural gas control system based on linear programming methods // Energy Systems. 2021. V. 12. N 1. P. 233–245. <https://doi.org/10.1007/s12667-019-00353-z>
5. Matveev M., Kabulova E., Kanisheva O. Fuzzy modeling for steel making processes // ARPN Journal of Engineering and Applied Sciences. 2015. V. 10. N 18. P. 8271–8277.
6. Евсюткин И.В., Марков Н.Г. Глубокие искусственные нейронные сети для прогноза значений дебитов добывающих скважин // Известия Томского политехнического университета. Инжиниринг георесурсов. 2020. Т. 331. № 11. С. 88–95. <https://doi.org/10.1879/9/24131830/2020/11/2888>
7. Дмитриев В.М., Ганджа Т.В., Кочергин М.И. Матрично-топологический анализ компонентных цепей // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2023. № 62. С. 25–35. <https://doi.org/10.17223/19988605/62/3>
8. Pantelic V., Postma S., Lawford M., Jaskolka M., Mackenzie B., Korobkine A., Bender M., Ong J., Marks G., Wassying A. Software engineering practices and simulink: bridging the gap // International Journal on Software Tools for Technology Transfer. 2018. V. 20. N 1. P. 95–117. <https://doi.org/10.1007/s10009-017-0450-9>
9. Мызникова В.А., Устименко В.В., Чубарь А.В. Построение нечётких регуляторов для систем управления автономных объектов в среде SimInTech // Космические аппараты и технологии. 2019. Т. 3. № 1(27). С. 22–27. <https://doi.org/10.26732/2618-7957-2019-1-22-27>
10. Дмитриев В.М., Ганджа Т.В. Методика построения многоуровневых компонентных цепей для моделирования химико-технологических систем // Доклады Томского государственного университета систем управления и радиоэлектроники. 2017. Т. 20. № 3. С. 98–102. <https://doi.org/10.21293/1818-0442-2017-20-3-82-86>
11. Дмитриев В.М., Ганджа Т.В., Ганджа В.В., Мальцев Ю.И. СВИП — система виртуальных инструментов и приборов. Томск: В-Спектр, 2014. 216 с.
12. Gandzha T.V., Dmitriev V.M., Zaichenko T.N., Aksenova N.V. Computer modeling of chemical process systems with distributed parameters by means of component circuit method with non-uniform vector coupling // Petroleum and Coal. 2021. V. 63. N 1. P. 68–77.
13. Karelin A.E., Maystrenko A.V., Svetlakov A.A., Dmitriev V.M., Gandzha T.V., Aksenova N.V. Synthesis of an automatic control method for major oil pipelines based on inverse dynamics problem concept // Petroleum and Coal. 2018. V. 60. N 1. P. 152–156.
14. Цвенгер И.Г., Низамов И.Р. Применение нейросетевых регуляторов в системах управления электроприводами // Вестник Технологического университета. 2017. Т. 20. № 8. С. 111–114.
15. Суханова Н.В. Разработка и применение нейросетевых моделей в автоматизации управления оборудованием и технологическими процессами // Автоматизация и моделирование в проектировании и управлении. 2022. № 1(15). С. 24–32. <https://doi.org/10.30987/2658-6436-2022-1-24-32>
16. Kochergin M.I. Interpretation of the statechart diagram into a multilevel simulation language // Доклады Томского государственного университета систем управления и радиоэлектроники. 2017. Т. 20. № 4. С. 122–125. <https://doi.org/10.21293/1818-0442-2017-20-4-122-125>

References

1. Korikov A.M. Intellectual management in technical systems. *Scientific Bulletin of NSTU*, 2014, no. 1(54), pp. 18–26. (in Russian)
2. Onufrey A.Yu., Razumov A.V., Kakaev V.V. A method of optimizing the structure of hierarchical distributed control systems. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 1, pp. 44–53. (in Russian). <https://doi.org/10.17586/2226-1494-2023-23-1-44-53>
3. Kolesnikov A.A., Kolesnikov A.A., Kuzmenko A.A. ADAR method and theory of adaptive control in the tasks of synthesis of the nonlinear control systems. *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18, no. 9, pp. 579–589. (in Russian). <https://doi.org/10.17587/mau.18.579-589>
4. Dayev Z.A., Kairakbayev A.K., Sultanov N.Z. Development of humidity of natural gas control system based on linear programming methods. *Energy Systems*, 2021, vol. 12, no. 1, pp. 233–245. <https://doi.org/10.1007/s12667-019-00353-z>
5. Matveev M., Kabulova E., Kanisheva O. Fuzzy modeling for steel making processes. *ARPN Journal of Engineering and Applied Sciences*, 2015, vol. 10, no. 18, pp. 8271–8277.
6. Evsyutkin I.V., Markov N.G. Deep artificial neural networks for forecasting debit values for production wells. *Bulletin of the Tomsk Polytechnic University. Geo Assets Engineering*, 2020, vol. 331, no. 11, pp. 88–95. (in Russian). <https://doi.org/10.18799/24131830/2020/11/2888>
7. Dmitriev V.M., Gandzha T.V., Kochergin M.I. Matrix topological analysis of component circuits. *Tomsk State University Journal of Control and Computer Science*, 2023, no. 62, pp. 25–35. (in Russian). <https://doi.org/10.17223/19988605/62/3>
8. Pantelic V., Postma S., Lawford M., Jaskolka M., Mackenzie B., Korobkine A., Bender M., Ong J., Marks G., Wassying A. Software engineering practices and simulink: bridging the gap. *International Journal on Software Tools for Technology Transfer*, 2018, vol. 20, no. 1, pp. 95–117. <https://doi.org/10.1007/s10009-017-0450-9>
9. Myznikova V.A., Ustimenko V.V., Chubar A.V. Fuzzy controllers construction in the simintech environment. *Spacecrafts & Technologies*, 2019, vol. 3, no. 1(27), pp. 22–27. (in Russian). <https://doi.org/10.26732/2618-7957-2019-1-22-27>
10. Dmitriev V.M., Gandzha T.V. The method of constructing multilevel component chains for modeling chemical-technological systems. *Proceedings of TUSUR University*, 2017, vol. 20, no. 3, pp. 98–102. (in Russian). <https://doi.org/10.21293/1818-0442-2017-20-3-82-86>
11. Dmitriev V.M., Gandzha T.V., Gandzha V.V., Malcev Ju.I. *SVID — System of Virtual Instruments and Devices*. Tomsk, V-Spektr, 2014, 216 p. (in Russian)
12. Gandzha T.V., Dmitriev V.M., Zaichenko T.N., Aksenova N.V. Computer modeling of chemical process systems with distributed parameters by means of component circuit method with non-uniform vector coupling. *Petroleum and Coal*, 2021, vol. 63, no. 1, pp. 68–77.
13. Karelin A.E., Maystrenko A.V., Svetlakov A.A., Dmitriev V.M., Gandzha T.V., Aksenova N.V. Synthesis of an automatic control method for major oil pipelines based on inverse dynamics problem concept. *Petroleum and Coal*, 2018, vol. 60, no. 1, pp. 152–156.
14. Tsvenger I.G., Nizamov I.R. Application of neural network controllers in the electric drive control systems. *Bulletin of the Technological University*, 2017, vol. 20, no. 8, pp. 111–114. (in Russian)
15. Sukhanova N. Developing and applying neural network models in automating equipment control and technological processes. *Automation and Modelling in Design and Management*, 2022, no. 1(15), pp. 24–32. (in Russian). <https://doi.org/10.30987/2658-6436-2022-1-24-32>
16. Kochergin M.I. Interpretation of the statechart diagram into a multilevel simulation language. *Proceedings of TUSUR University*, 2017, vol. 20, no. 4, pp. 122–125. <https://doi.org/10.21293/1818-0442-2017-20-4-122-125>
17. Ezhov A.A., Cherepnev A.S. The uplearning of the Hopfield neural network: the search for a global minimum of a functional and the model of rapid eye movement. *Matematicheskoe modelirovanie*, 2009, vol. 21, no. 5, pp. 10–20. (in Russian)

17. Ежов А.А., Черепнев А.С. Дообучение нейронной сети Хопфилда: поиск глобального минимума функционала и модель быстрого сна // Математическое моделирование. 2009. Т. 21. № 5. С. 10–20.
18. Дмитриенко В.Д., Заковоротный А.Ю. Решение проблемы дообучения классических нейронных сетей // Автоматизированные технологии и производства. 2015. № 4 (10). С. 32–40.
19. Федотов Д.В., Попов Е.А., Охорзин В.А. Оптимизация структуры сверточной нейронной сети с помощью самоконфигурируемого эволюционного алгоритма в одной задаче идентификации // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2015. Т. 16. № 4. С. 857–863.
20. Рыбин П.К., Горин Р.В. Метод оптимизации структуры нейронной сети для задачи планирования подвода поездов к предпортовым станциям // Вестник транспорта Поволжья. 2017. № 5(65). С. 55–63.
18. Dmitrienko V.D., Zakovorotnyi A.Y. Solving the problem of additional training of classical neural networks. *Automation of technologies and production*, 2015, no. 4 (10), pp. 32–40. (in Russian).
19. Fedotov D.V., Popov E.A., Okhorzin V.A. Optimisation of convolutional neural network structure with self-configuring evolutionary algorithm in one identification problem. *Vestnik SibGAU*, 2015, vol. 16, no. 4, pp. 857–863. (in Russian)
20. Rybin P.K., Gorin R.V. Method of optimizing of the neural network structure for planning train arrival at pre-port stations. *Vestnik transporta Povolzhya*, 2017, no. 5 (65), pp. 55–63. (in Russian)

Автор

Кочергин Максим Игоревич — кандидат технических наук, доцент, Томский государственный университет систем управления и радиоэлектроники, Томск, 634050, Российская Федерация, [sc 57211392793](https://orcid.org/0000-0001-7404-8718), [https://orcid.org/0000-0001-7404-8718](mailto:maksim.i.kochergin@tusur.ru), maksim.i.kochergin@tusur.ru

Статья поступила в редакцию 12.07.2023
Одобрена после рецензирования 19.10.2023
Принята к печати 10.11.2023

Author

Maksim I. Kochergin — PhD, Associate Professor, Tomsk State University of Control Systems and Radioelectronics, Tomsk, 634050, Russian Federation, [sc 57211392793](https://orcid.org/0000-0001-7404-8718), <https://orcid.org/0000-0001-7404-8718>, maksim.i.kochergin@tusur.ru

Received 12.07.2023
Approved after reviewing 19.10.2023
Accepted 10.11.2023



Работа доступна по лицензии
 Creative Commons
 «Attribution-NonCommercial»