

УДК 004.89

doi:10.21685/2072-3059-2021-3-2

## Проектирование нейросетевых моделей посредством методов визуального программирования

С. А. Ямашкин<sup>1</sup>, А. А. Камаева<sup>2</sup>

<sup>1,2</sup>Национальный исследовательский Мордовский государственный университет имени Н. П. Огарева, Саранск, Россия

<sup>1</sup>yamashkinsa@mail.ru, <sup>2</sup>aakamaeva@yandex.ru

**Аннотация.** *Актуальность и цели.* В современном мире все большую популярность приобретают методы компонентной разработки. Они позволяют не только быстрее решать поставленные задачи, но и соответствовать высоким требованиям к производительности и надежности создаваемых программных продуктов. Применение блочного подхода в машинном обучении является прорывным методом, позволяющим в несколько раз облегчить разработку сложных нейросетевых архитектур. Целью исследования является изучение методов компонентной разработки и технологии графосимволического программирования для решения задачи визуального программирования нейронных сетей. *Материалы и методы.* Развиваются методы и алгоритмы конфигурирования нейросетевых моделей на основе программных комплексов, реализующих графический интерфейс создания нейронных сетей с помощью графосимволического программирования на основе языка программирования JavaScript. *Результаты.* Проведены комплексные исследования в области визуального программирования нейронных сетей с применением методов компонентной разработки. Разработан графический интерфейс для конфигурирования архитектур нейронных сетей. *Выводы.* Представленный подход визуального программирования нейронных сетей упрощает процесс разработки, позволяет избегать ошибок и создавать более производительные и надежные системы.

**Ключевые слова:** визуальное программирование, нейронная сеть, машинное обучение, графы, архитектура нейронных сетей, топологический граф, графосимволическое программирование, компонентная разработка, модульное программирование

**Финансирование:** работа выполнена при финансовой поддержке гранта Президента Российской Федерации (грант № МК-199.2021.1.6).

**Для цитирования:** Ямашкин С. А., Камаева А. А. Проектирование нейросетевых моделей посредством методов визуального программирования // Известия высших учебных заведений. Поволжский регион. Технические науки. 2021. № 3. С. 14–24. doi:10.21685/2072-3059-2021-3-2

## Designing neural network models using visual programming methods

S.A. Yamashkin<sup>1</sup>, A.A. Kamaeva<sup>2</sup>

<sup>1,2</sup>Ogarev Mordovia State University, Saransk, Russia

<sup>1</sup>yamashkinsa@mail.ru, <sup>2</sup>aakamaeva@yandex.ru

**Abstract.** *Background.* In the modern world, component development methods are becoming increasingly popular. They allow not only to quickly solve the assigned tasks, but also to meet the high requirements for the performance and reliability of the created software products. The use of the block approach in machine learning is a breakthrough method that

makes the development of complex neural network architectures several times easier. The purpose of this work is to study the methods of component development and the technology of graph-symbolic programming for solving the problem of visual programming of neural networks. *Materials and methods.* The work develops methods and algorithms for configuring neural network models based on software systems that implement a graphical interface for creating neural networks using graph-symbolic programming based on the JavaScript programming language. *Results.* Comprehensive research has been carried out in the field of visual programming of neural networks using component development methods. A graphical interface has been developed for configuring neural network architectures. *Conclusions.* The presented approach to visual programming of neural networks simplifies the development process, avoids errors and creates more efficient and reliable systems.

**Keywords:** visual programming, neural network, machine learning, graphs, neural network architecture, topological graph, graphic-symbolic programming, component development, modular programming

**Acknowledgments:** the research was financed by the grant of the President of the Russian Federation (grant № МК-199.2021.1.6).

**For citation:** Yamashkin S.A., Kamaeva A.A. Designing neural network models using visual programming methods. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences.* 2021;(3):14–24. (In Russ.). doi:10.21685/2072-3059-2021-3-2

## Введение

Краеугольное значение в решении задачи усиления связанности территории Российской Федерации играет формирование, внедрение и эффективное использование цифровых инфраструктур пространственных данных (ИПД) регионов страны, нацеленных на оперативное и долгосрочное прогнозирование стихийных процессов, а также диагностику состояния природно-социально-производственных систем (ПСПС). Вследствие того, что данные, интегрируемые в цифровых ИПД часто характеризуются большим объемом и наличием в них сложных нелинейных закономерностей, ядром систем данного класса в настоящее время часто становятся методы и алгоритмы автоматизированного машинного анализа, позволяющие решать комплекс прикладных и проектно-ориентированных задач – классификация и сегментация данных, обнаружение аномалий, обучение признакам, слияние данных.

Предметом анализа при этом могут выступать цифровые ландшафтные карты, данные дистанционного зондирования Земли (ДЗЗ), материалы аэрофотосъемки, структурированные массивы информации об природных, социальных и экономических объектах, имеющих распределенную геопрограммированную организацию [1]. При этом области применения результатов такого анализа в народном хозяйстве являются весьма разнообразными: от повышения эффективности сельского хозяйства посредством развертывания систем точечного земледелия до оценки последствий стихийных природных и природно-техногенных процессов. В рамках обозначенного направления следует выделить ряд задач, для которых актуальны разработка и внедрение информационных систем с использованием искусственных нейронных сетей [2]: классификация, кластеризация, распознавание образов, принятие решений, анализ данных и прогнозирование.

Решение научной проблемы проектирования, обучения и использования нейронных сетей для решения задачи автоматизированного анализа пространственных и пространственно-временных данных встречает множество

открытых вопросов [3]. Данная статья посвящена решению задачи разработки эффективного средства проектирования нейронных сетей посредством методов визуального программирования. Обозначенный модуль представляет собой компонент репозитория глубоких нейросетевых моделей, создаваемый для решения научной проблемы накопления и систематизации моделей и алгоритмов машинного обучения посредством разработки для анализа и прогнозирования развития пространственных процессов с целью поддержки процесса принятия управленческих решений в области обеспечения условий устойчивого развития регионов России.

## 1. Материалы и методы исследования

Чтобы разобраться в сложностях проектирования нейронных сетей, целесообразно выделить и охарактеризовать этапы жизненного цикла данного процесса:

1. *Постановка задачи и выбор архитектуры нейронной сети.* Анализ поставленной задачи с обоснованием применения методов искусственного интеллекта для ее решения. Выявление преимуществ, которые метод нейросетевого моделирования должен дать. Под конкретную постановку задачи подбирается архитектура нейронной сети.

2. *Определение количественного и качественного составов входов и выходов.* Качественный состав входных и выходных переменных нейросетевой модели определяется главным образом задачей, поставленной на предыдущем этапе. Количественный состав может зависеть от природы входных и выходных переменных и от описываемых ими процессов и свойств.

3. *Формирование исходной выборки данных.* Исходная выборка данных может быть сформирована на основе практического или вычислительного эксперимента. Чем больше объем проведенного эксперимента, тем более точную нейросетевую модель можно получить и использовать на практике.

4. *Предварительная обработка исходной выборки и ее нормализация.* Данный этап определяется выбранной архитектурой нейронной сети, составом входных и выходных переменных и, собственно, фактическими данными, полученными в результате эксперимента.

5. *Разделение исходной выборки на обучающую и тестовую составляющие.* Имеющаяся исходная выборка делится на обучающую и тестовую подвыборки с учетом используемой архитектуры нейронной сети. Обычно объем обучающей подвыборки в несколько раз больше, чем тестовой. Первая используется строго для настройки весовых коэффициентов. Вторая – для проверки корректности настроенной нейросетевой модели. В случае ограниченного объема исходной выборки она вся может быть задействована для обучения, а оценка корректности работы сети проверяется на этапе практической эксплуатации.

6. *Определение структуры нейронной сети.* Помимо ранее установленного состава входных и выходных переменных, на данном этапе задается количество скрытых слоев и нейронов в каждом слое. Для многих архитектур нейронных сетей выбор структуры зависит в том числе от объема обучающих данных.

7. *Настройка параметров нейронной сети и алгоритма ее обучения.* На данном этапе задаются вид и параметры активационных функций скры-

тых и выходных нейронов [4]. Для итерационного алгоритма выбираются условия окончания обучения, коэффициент скорости, порядок предъявления примеров обучающей выборки.

8. *Обучение нейронной сети.* На этом этапе осуществляется расчет весовых коэффициентов. Для обучения используется только обучающая выборка.

9. *Контрастирование нейронной сети.* Если алгоритм обучения нейронной сети подразумевал итерационную подстройку весовых коэффициентов на протяжении всех эпох обучения, нередки случаи, когда некоторые из весов по абсолютному значению настолько близки к нулю, что не оказывают никакого практического влияния на результаты расчетов. Тем не менее на такие расчеты тратится довольно много вычислительных ресурсов, что заметно замедляет процесс обучения. Для решения этой проблемы такие весовые коэффициенты полностью обнуляются, т.е. соответствующая связь между входами и нейронами или между двумя нейронами исключается из структуры сети.

10. *Тестирование нейронной сети.* На этапе тестирования производится оценка ошибки работы нейронной сети на тестовой выборке. Этапы с седьмого по девятый могут выполняться в цикле неоднократно – до тех пор, пока не будет получена настроенная нейронная сеть, с помощью которой можно решать задачу с требуемым уровнем ошибки.

11. *Практическое использование и дообучение нейронной сети.* Получаемые в ходе практической эксплуатации нейронной сети пары входных и выходных векторов могут использоваться для дальнейшей подстройки весовых коэффициентов. Это особенно важно, если изначально объем выборки данных был небольшим [5]. Часть архитектур нейронных сетей, использующих принцип самоорганизации, в ходе дообучения может не только изменять значения весовых коэффициентов, но и менять свою структуру.

Следует заметить, что перечисленная последовательность этапов характерна в общем для работы с искусственными нейронными сетями. Однако для ряда архитектур могут иметь место свойственные только им отличия. Данная структура позволяет увидеть четкий подход к проектированию, концепцию которого можно реализовать при помощи средств блочного подхода в программировании.

Так как существует большое количество решений, помогающих не тратить драгоценное время на изобретение велосипеда, обратимся к парадигме модульного программирования, где функционально самодостаточный фрагмент кода называется модулем [6]. При взаимодействии параметры и функции каждого из модулей должны оставаться неизменными. Плюсы модульного программирования очевидны: ускорение разработки, повышение надежности, упрощение тестирования, переносимость и взаимозаменяемость.

Как уже было сказано, в области нейронных сетей существует огромное количество библиотек, модулей и паттернов, которые можно внедрять, комбинировать и использовать под широкий набор решаемых задач [7]. Для наглядного отображения разрабатываемых блоков с целью их эффективного применения целесообразно обратиться к парадигме визуального проектирования.

Визуальная разработка программного обеспечения – это процесс образного представления программы при помощи стандартного набора графических элементов [8]. Методы визуальной разработки позволяют повысить

качество разрабатываемого программного продукта за счет того, что при разработке значительно снижается количество вынужденных ошибок управления.

Согласно общепринятой классификации к системам визуализации программного обеспечения относят графические языки программирования и визуальные средства разработки (рис. 1). Языки визуального программирования позволяют разрабатывать программное обеспечение, не печатая текстовые команды, а манипулируя с графическими элементами. В визуальных средствах разработки наиболее распространенные блоки программного кода представлены в виде графических объектов [9]. В некоторых случаях эти понятия линейно связаны между собой. Так, G – визуальный язык программирования, доступный посредством его родной среды разработки LabView.

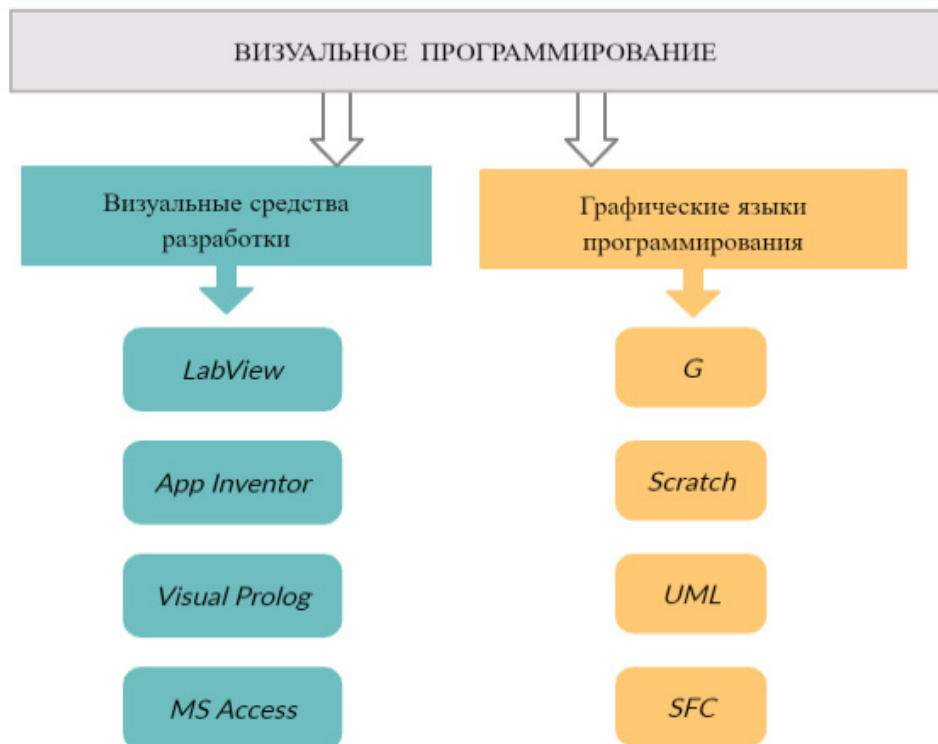


Рис. 1. Классификация средств визуального программирования

Существует несколько распространенных классификаций языков визуального программирования по разным признакам [10]. Так, по основному моделируемому аспекту следует выделить: языки описания структуры программной системы, потоков управления и данных, поведения (процессов), а также средства построения графического интерфейса [11]. Редакторы визуального программирования часто используются для проектирования пользовательского интерфейса. Они позволяют значительно упростить разработку приложений путем замены текстовой формы записи кода на метод конструирования. Однако визуальное представление алгоритмов программ они не затрагивают.

При разработке визуальных языков программирования в качестве основного подхода используются графовые формы, позволяющие реализовывать сложные взаимосвязи объектов [12]. Информативность графового представления алгоритма достигается за счет варьирования цветов, размеров и форм вершин графа, преобразования толщины и внешнего вида дуг, изменения взаимного расположения объектов без изменения топологии графа.

Создание визуального (образного) стиля разработки программного обеспечения является основным мотивом разработки технологии *графосимволического программирования*. Кроме того, эта технология преследует цель автоматизации процессов проектирования, кодирования и тестирования программного обеспечения [13]. Эти концепции отражают требования к разрабатываемой визуальной среде для создания и конфигурирования нейронных сетей.

Технологии графосимволического программирования повышают наглядность разрабатываемых программных кодов, значительно уменьшают количество ошибок, допускаемых на этапах проектирования и кодирования, и за счет этого ускоряют процесс разработки и повышают надежность разрабатываемых программных кодов. В совокупности с применением блочного подхода методы визуального программирования позволят не только ускорять разработку за счет простой графической взаимозаменяемости логических блоков, но и обеспечат простую переносимость подобного рода программ.

## 2. Визуальное программирование нейросетевых моделей

Практическое использование глубоких нейронных сетей, интегрируемых в репозитории нейросетевых моделей, требует декомпозиции онтологической модели хранилища на системообразующие домены: моделей машинного обучения, данных и задач (рис. 2). Благодаря этому становится возможным создание платформенного решения для консолидации, подбора и эффективного использования новых методов и алгоритмов для решения проблемно-ориентированных задач, связанных с анализом больших пространственных и пространственно-временных данных [1].

Консолидируемые в репозитории модели машинного обучения и решаемые с их использованием проектные задачи всегда опираются на наборы данных различного типа. Домен моделей машинного обучения при этом описывается понятиями, определяющими глубокие нейросетевые модели различной топологии, и методами, методиками и алгоритмами их обучения. Поскольку машинное обучение не ограничивается искусственными нейронными сетями, в данном исследовании на них сделан акцент вследствие того, что алгоритмы данного класса имеют важное значение для эффективного анализа больших пространственных данных, нацеленного на выявление сложных нелинейных зависимостей в них.

Модель глубокой искусственной нейронной сети (ИНС) определяется ее топологией, в том числе использованием слоев различного типа (сверточные, полносвязные, рекуррентные), применением активационных и регуляризационных функций. Архитектуру модели при этом определяют паттерны связи слоев, формат представления входных данных модели и особенности ее выходного сигнала. Глубокая нейросетевая модель характеризуется также алгоритмами инициализации и оптимизации, стратегией обучения.

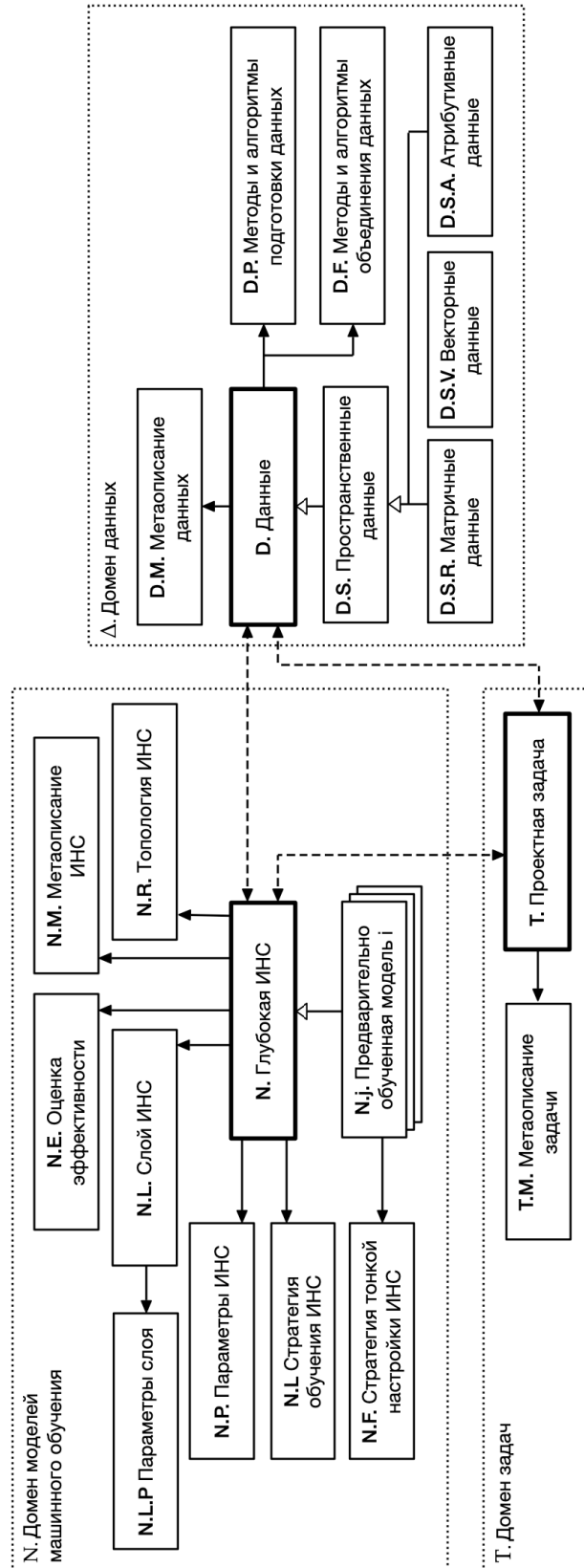


Рис. 2. Предлагаемая онтологическая модель репозитория глубоких нейронных сетей

Отметим также, что практический интерес представляют не только модели искусственных нейронных сетей, но и их обученные экземпляры, готовые к решению проектных задач. Каждой модели репозитория при этом целесообразно сопоставить системную категорию, метаописание, руководство для использования, характеристику решаемых проектных задач, описание анализируемых данных и форм их представления.

Для решения задачи эффективного проектирования глубоких нейросетевых моделей был разработан программный комплекс, позволяющий решать обозначенную задачу, посредством методов визуального программирования. Созданный модуль представляет собой компонент репозитория глубоких нейросетевых моделей, создаваемого для решения научной проблемы накопления и систематизации моделей и алгоритмов машинного обучения посредством разработки для анализа и прогнозирования. Для того чтобы оценить удобство визуального программирования нейронных сетей, обратимся к моделям с несколькими входами и выходами.

Например, если необходимо создать систему для ранжирования полученной геоинформации по приоритету и для дальнейшего направления информации в нужный отдел, то разрабатываемая модель будет иметь три входа: графические данные с устройств IoT (данные дистанционного зондирования); данные геолокации (картографические материалы); автоматически генерируемые теги, описывающие события (категориальные сведения). В результате модель должна иметь два выхода: оценка приоритета от 0 до 1 (скалярный сигмовидный выход); отдел, который должен обрабатывать полученную геоинформацию (вывод softmax по набору отделов). Результат проектирования нейронной сети ранжирования заявок представлен на рис. 3,а.

В дополнение к моделям с несколькими входами и выходами рассмотрим манипулирование топологиями нелинейной связи, в которых слои нейронной сети не связаны последовательно. Примером такой архитектуры является ResNet, использующая глубокую «остаточную» структуру обучения. Модель ResNet для CIFAR10 представлена на рис. 3,б.

Для реализации графического интерфейса модуля проектирования нейросетевых моделей посредством методов визуального программирования использован язык программирования JavaScript. Модуль апробирован при формировании решения задачи формирования базовых моделей нейросетевых архитектур, представленных в открытой нейросетевой библиотеке Keras, в частности Layers API. По результатам исследования получено свидетельство о государственной регистрации программы для ЭВМ «Программный модуль для графической визуализации нейросетевых моделей» (№ 2021619571).

### **Заключение**

Важным компонентом репозитория глубоких нейросетевых моделей, обладающего функциями навигации, поиска релевантной модели машинного обучения для решения практико-ориентированных задач анализа пространственных данных, является подсистема динамической визуализации моделей на основе адаптивных веб-интерфейсов в виде динамически выстраиваемых графов с возможностью интерактивного редактирования архитектуры и топологии нейросетевой модели посредством веб-клиента.



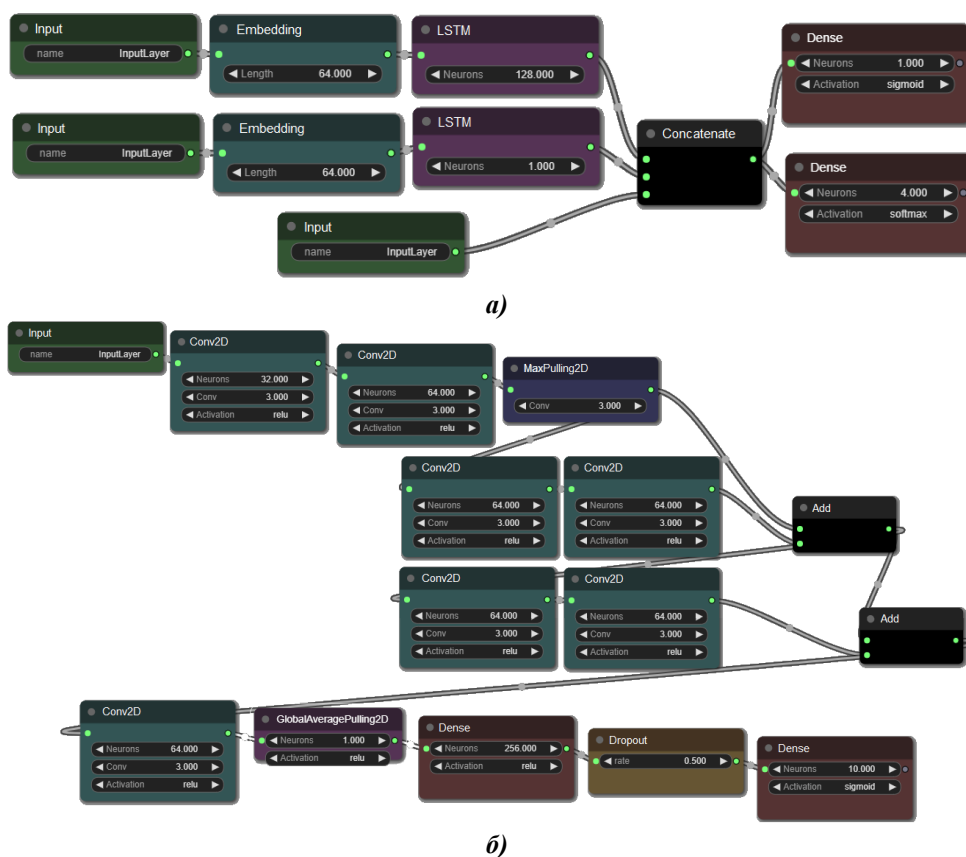


Рис. 3. Описание глубоких нейросетевых моделей посредством визуального программирования: *a* – модель с тремя входами и двумя выходами; *б* – архитектура ResNet

Графические интерфейсы репозитория нейронных сетей должны проектироваться исходя из необходимости эффективного решения задачи предоставления информации о конкретной нейросетевой модели (в том числе систематизированное описание, класс архитектуры, формат анализируемых и выходных данных, сведения о топологии, субъективные и объективные показатели эффективности, рекомендации по тонкой настройке модели, примеры практико-ориентированного использования).

Применение методов визуального программирования для описания сложных архитектур нейронных сетей позволяет значительно ускорить процесс их разработки и уменьшает количество ошибок за счет повышения наглядности.

В совокупности с применением блочного подхода методы визуального программирования позволят не только ускорять разработку за счет простой графической взаимозаменяемости логических блоков, но и обеспечат простую переносимость как целых программ, так и отдельных модулей.

### Список литературы

1. Ямашкин С. А. Ямашкин А. А., Занозин В. В. Формирование репозитория глубоких нейронных сетей в системе цифровой инфраструктуры пространственных данных // Потенциал интеллектуально одаренной молодежи – развитию науки и

- образования : материалы IX Междунар. науч. форума молодых ученых, инноваторов, студентов и школьников / под общ. ред. Т. В. Золиной. Астрахань, 2020. С. 370–375.
2. Камаева А. А. Современное состояние искусственных нейронных сетей // *Инновации. Наука. Образование*. 2020. № 16 С. 377–387.
  3. Коварцев А. Н., Жидченко В. В., Попова-Коварцева Д. А., Аболмасов П. В. Принципы построения технологии графосимволического // *Открытые семантические технологии проектирования интеллектуальных систем*. 2013. №. 3. С. 195–204.
  4. Морозова Т. Ю., Бурлаченко Т. Б. Решение задачи прогнозирования в системах с большой степенью неопределенности // *Известия ТРТУ*. 2006. № 9-2 (64). С. 169.
  5. Гафаров Ф. М., Галимьянов А. Ф. Искусственные нейронные сети и приложения : учеб. пособие. Казань : Изд-во Казан. ун-та, 2018. 121 с.
  6. Бубнов И. Что такое модульное программирование и кому оно нужно // *Geek-Brains*. URL: [https://geekbrains.ru/posts/module\\_programming/](https://geekbrains.ru/posts/module_programming/)
  7. Кириченко А. А. Нейропакеты – современный интеллектуальный инструмент исследователя : учеб. пособие. М., 2013. 297 с.
  8. Коварцев А. Н., Жидченко В. В., Попова-Коварцева Д. А. Методы и технологии визуального программирования : учеб. пособие. Самара : Офорт, 2017. 197 с.
  9. Ломакин В. В. Программирование и программное обеспечение информационных технологий : учеб. пособие. Белгород : Изд-во БелГУ, 2010. 114 с.
  10. Тюгашев А. А. Графические языки программирования и их применение в системах управления реального времени. Самара : Изд-во Самарского научного центра РАН, 2009. 98 с.
  11. Ломакин В. В. Программирование и программное обеспечение информационных технологий : учеб. пособие. Белгород : Изд-во БелГУ, 2010. 114 с.
  12. Жидченко В. В. Программный комплекс моделирования и анализа алгоритмов параллельных вычислений : дис. ... канд. техн. наук : 05.13.18. Самара, 2007. 189 с.
  13. Власов А. И. Системный анализ технологических процессов производства сложных технических систем с использованием визуальных моделей // *Международный научно-исследовательский журнал*. 2013. № 10-2. С. 17.

### References

1. Yamashkin S.A., Yamashkin A.A., Zanozin V.V. Formation of a repository of deep neural networks in the system of digital spatial data infrastructure. *Potentsial intellektual'no odarennoy molodezhi – razvitiyu nauki i obrazovaniya: materialy IX Mezhdunar. nauch. foruma molodykh uchenykh, innovatorov, studentov i shkol'nikov = The potential of intellectually gifted youth for the development of science and education: proceedings of the 9<sup>th</sup> International scientific forum of young scientists, innovators, students and pupils*. Astrakhan, 2020:370–375. (In Russ.)
2. Kamaeva A.A. The current state of artificial neural networks. *Innovatsii. Nauka. Obrazovanie = Innovation. Science. Education*. 2020;(16):377–387. (In Russ.)
3. Kovartsev A.N., Zhidchenko V.V., Popova-Kovartseva D.A., Abolmasov P.V. Principles of constructing graph-symbolic technology. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system = Open semantic technologies for the design of intelligent systems*. 2013;(3):195–204. (In Russ.)
4. Morozova T.Yu., Burlachenko T.B. A solution of the forecasting problem in systems with a high degree of uncertainty. *Izvestiya TRTU = Proceedings of TRTU*. 2006; (9-2):169. (In Russ.)
5. Gafarov F.M., Galimyanov A.F. *Iskusstvennye neyronnye seti i prilozheniya: ucheb. posobie = Artificial neural networks and applications: textbook*. Kazan: Izd-vo Kazan. un-ta, 2018:121. (In Russ.)
6. Bubnov I. What is modular programming and who needs it. *Geek-Brains*. (In Russ.). Available at: [https://geekbrains.ru/posts/module\\_programming/](https://geekbrains.ru/posts/module_programming/)

7. Kirichenko A.A. *Neuropakety – sovremennyy intellektual'nyy instrument issledovatelya: ucheb. posobie = Neuropackages – a modern intellectual tool for the researcher: textbook*. Moscow, 2013:297. (In Russ.). ISBN 978-5-9904911-1-3.
8. Kovartsev A.N., Zhidchenko V.V., Popova-Kovartseva D.A. *Metody i tekhnologii vizual'nogo programmirovaniya: ucheb. posobie = Methods and technologies of visual programming: textbook*. Samara: Ofort, 2017:197. (In Russ.)
9. Lomakin V.V. *Programmirovaniye i programmnoye obespecheniye informatsionnykh tekhnologiy: ucheb. posobie = Information technology programming and software: textbook*. Belgorod: Izd-vo BelGU, 2010:114. (In Russ.)
10. Tyugashev A.A. *Graficheskie yazyki programmirovaniya i ikh primeneniye v sistemakh upravleniya real'nogo vremeni = Graphic programming languages and their application in real-time control systems*. Samara: Izd-vo Samarskogo nauchnogo tsentra RAN, 2009:98. (In Russ.)
11. Lomakin V.V. *Programmirovaniye i programmnoye obespecheniye informatsionnykh tekhnologiy: ucheb. posobie = Information technology programming and software: textbook*. Belgorod: Izd-vo BelGU, 2010:114. (In Russ.)
12. Zhidchenko V.V. Software complex for modeling and analysis of parallel computing algorithms. PhD dissertation. Samara, 2007:189. (In Russ.)
13. Vlasov A.I. System analysis of technological processes for the production of complex technical systems using visual models. *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal = International scientific and research journal*. 2013;(10-2):17. (In Russ.)

#### Информация об авторах / Information about the authors

**Станислав Анатольевич Ямашкин**  
кандидат технических наук, доцент  
кафедры автоматизированных систем  
обработки информации и управления,  
Национальный исследовательский  
Мордовский государственный  
университет имени Н. П. Огарева  
(Россия, г. Саранск,  
ул. Большевикская, 68/1)  
E-mail: yamashkinsa@mail.ru

**Stanislav A. Yamashkin**  
Candidate of engineering sciences,  
associate professor of the sub-department  
of automated data processing and control  
systems, Ogarev Mordovia State University  
(68/1 Bolshevistskaya street,  
Saransk, Russia)

**Анастасия Андреевна Камаева**  
студентка, Национальный  
исследовательский Мордовский  
государственный университет  
имени Н. П. Огарева (Россия,  
г. Саранск, ул. Большевикская, 68/1)  
E-mail: aakamaeva@yandex.ru

**Anastasiya A. Kamaeva**  
Student, Ogarev Mordovia State  
University (68/1 Bolshevitskaya  
street, Saransk, Russia)

**Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflicts of interests.**

**Поступила в редакцию / Received 23.06.2021**

**Поступила после рецензирования и доработки / Revised 03.08.2021**

**Принята к публикации / Accepted 20.09.2021**