

# Optimization of neural network learning via system with automata approach

В. А. Бирюкова<sup>1</sup>

В данной работе представлена автоматная модель обучения нейронных сетей, реализованная с помощью языка программирования высокого уровня python и апробированная на задаче бинарной классификации. Также внутри рассматриваемой системы реализована процедура автоматизации обучения нейронных сетей с помощью выбора стратегии обучения с точки зрения «лучших практик».

**Ключевые слова:** искусственные нейронные сети, конечный автомат, бесконечный автомат, система для работы с нейронными сетями, автоматизация обучения нейронных сетей, оптимизация выбора гиперпараметров с использованием «лучших практик», компьютерное зрение, бинарная классификация.

В последнее время нейронные сети получили весьма широкое распространение, поскольку данные модели могут решать совершенно разные задачи и поэтому находят применение во многих областях, таких как научные исследования (моделирование биологических нейронных сетей)[1], финансовый сектор (прогнозирование котировок на основе статистических данных и новостных сообщений)[2], медицина (распознавание аномалий по снимкам пациентов) [3], проектирование беспилотных автомобилей [4] и т.д.

Искусственные нейронные сети (англ. artificial neural network, ANN) [5] — это математические модели и их программные и аппаратные реализации. Как математическая модель, нейронная сеть зависит от огромного количества параметров, процесс настраивания которых принято называть обучением нейронной сети. Помимо параметров, значения которых находятся во время обучения, нейронная сеть и сам процесс обучения имеют ряд параметров, требующих ручной настройки, т. н. гиперпараметров. Как набор гиперпараметров, так и их значения имеют многочисленные вариации даже в рамках решения очень узкого круга задач, поэтому выбор оптимальной стратегии обучения нейронной сети, которая бы после обучения удовлетворяла поставленным в задаче требованиям, остается процессом весьма трудным и требующим в существенной степени индивидуального подхода.

---

<sup>1</sup>*Бирюкова Вероника Андреевна* — студент кафедры Математической теории интеллектуальных систем механико-математического факультета МГУ, e-mail: biryukovaveronika@mail.ru.

*Biryukova Veronika Andreevna* — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Ввиду вышеописанных причин системы, позволяющие автоматизировать процесс подбора гиперпараметров сети, и, тем более, дающие возможность использовать такой мощный инструмент для решения задач из других научных сфер, весьма актуальны и востребованы. К решению данной проблемы существуют различные подходы, как чисто технические (программные), так и более концептуально-содержательные методы.

Например, библиотеки языка программирования python `sklearn` и `keras` имеют свои модули для автоматизации подбора гиперпараметров нейронной сети. О качестве полученных результатов и о возможностях улучшения процесса обучения нейронных сетей уже может судить эксперт.

Концептуально-содержательные подходы к автоматизации обучения нейронных сетей существуют, однако большинство из них имеют ограничения на тип используемой архитектуры. Данную проблему решают, например, с помощью алгоритмов «каскадной корреляции» [6] и «дальновидного мышления» [7] и их вариаций.

Кроме того, поскольку обучение нейронной сети можно рассматривать как некоторую оптимизационную задачу, то для решения вопроса об автоматизации обучения нейронных сетей можно использовать системы, изучающие моделирование логических процессов, такие как «автоматический решатель математических задач» А.С. Подколзина [8].

Предлагаемая в данной работе интеллектуальная система [9] и её реализация позволяют автоматизировать обучение нейронных сетей для решения различных задач, также в рамках этой системы возможно исполнение вспомогательных функций, таких как работа с имеющимися базами данных, обученными ранее моделями и накопленным в данной системе опытом обучения нейронных сетей. При создании данной системы используется автоматный подход [10]. Рассматриваемая система представляет собой взаимодействие двух абстрактных автоматов. Один автомат отвечает за управление системой – «управляющий» автомат (далее УА), другой – «исполняющий» автомат (далее ИА), который решает поставленные перед ним управляющим автоматом подзадачи.

Пользователь задает некоторый начальный набор параметров – предлагаемая системе задача и её характеристики, – и тем самым задавая начальные состояния обоих автоматов. Рассматриваются задачи из области компьютерного зрения – задачи машинного обучения, где объектами выступают изображения [11]. В процессе работы системы УА дает ИА указания, какую локальную задачу следует выполнить, и в зависимости от результата (выполнена подзадача или нет) переходит в соответствующее состояние. Если подзадача не выполнена, но система имеет вари-

анты решения этой проблемы, например предложение изменить условия поставленной перед системой задачи, тогда УА переходит в состояния, со второй компонентой «userdecision», где происходит взаимодействие с пользователем (с внешней средой), иначе система заканчивает свою работу.

Управляющий автомат – это конечный автомат без входа и выхода с несколькими начальными и заключительными состояниями. Состояние этого автомата задается двумя компонентами как прямое произведение двух конечных алфавитов  $(T, N)$ :

$T = \{ 'training', 'testing', 'database', 'history' \} = \{ t_0, t_1, t_2, t_3 \}$  – тип решаемой системой задачи (обучение, тестирование нейронной сети, работа с базами изображений, работа с сохраненной в этой системе «историей» обучения нейронных сетей соответственно).

$N = \{ 'check', 'database', 'userdecision', 'train', 'test', 'data', 'history', 'done' \} = \{ q_0, \dots, q_7 \}$  – название локально решаемой подзадачи.

Множество начальных состояний –  $\{ (t, 'check'), t \in T \}$ .

Множество заключительных состояний –  $\{ (t, 'done'), t \in T \}$ .

Функция переходов  $Q(q, t) = \tilde{Q}(\psi(q, t), q, t)$ , где  $\psi(q, t) \in \{0, 1\}$  – выход ИА, который интерпретируется как выполнена ли локальная задача или нет.

На рисунке 1 представлена обобщенная диаграмма переходов УА.

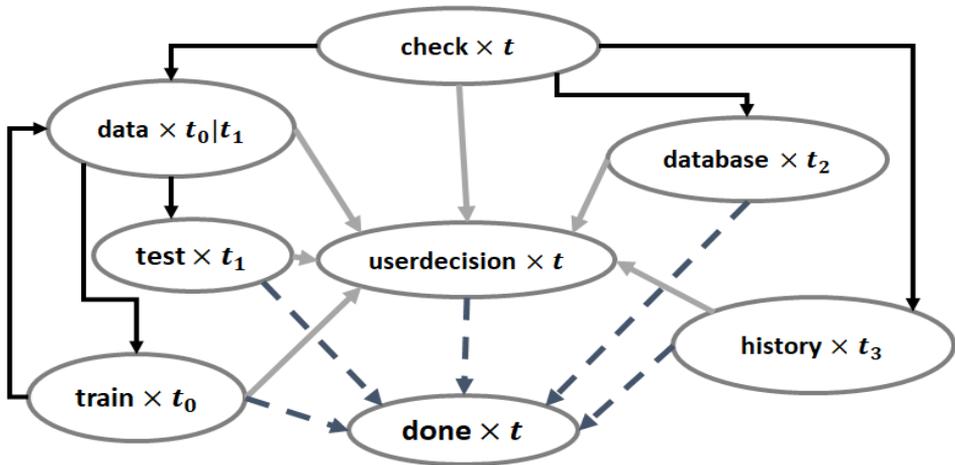


Рис. 1. Обобщенная диаграмма переходов управляющего автомата,  $t \in T$

Компонента состояния УА, отвечающая блоку решения локальной подзадачи:

- *'check'* – любое начальное состояние имеет это значение второй компоненты, в данном блоке происходит проверка функционирования ИА, при  $t \in \{'training', 'testing'\}$  ИА проводит поиск в своей внутренней базе моделей по соответствующему запросу;
- *'data'* – такое значение компоненты можем получить при  $t \in \{'training', 'testing'\}$ , в этом блоке ИА формирует множества для обучения и\или тестирования нейронной сети;
- *'userdecision'* – система взаимодействует с пользователем;
- *'train'* – в этом блоке происходит обучение нейронной сети ( $t = 'training'$ );
- *'test'* – проводится тестирование нейронной сети ( $t = 'testing'$ );
- *'database'* – блок работы с базами изображений ( $t = 'database'$ );
- *'history'* – блок работы с сохраненной в системе историей обучения нейронных сетей ( $t = 'history'$ );
- *'done'* – заключительное состояние УА, в него попадаем как при успешном выполнении поставленной задачи, так и при невозможности её решения.

Исполняющий автомат – это бесконечный автомат, представляющий собой часть системы, которая непосредственно занимается решением подзадач, поступающих от УА. Начальное состояние ИА задается по характеристикам задачи, поступающим из внешней среды. Состояние ИА  $s$  ( $s \in S$ ) задается как прямое произведение нескольких конечных и бесконечных алфавитов  $S$ . Заключительное состояние ИА – одно, в него попадаем, если поставленная задача была решена или решить задачу не представляется возможным (УА переходит в одно из своих заключительных состояний).

В роли входного значения ИА выступает состояние УА  $(q, t)$  ( $q$  – конечный входной алфавит), которое указывает какую подзадачу требуется решить. Система выполняет соответствующие действия, выдает результирующее значение функции выходов  $\psi(q, t) = \tilde{\psi}(q, t, s_i) = r$ ,  $r \in \{0, 1\}$ , которое интерпретируется УА как «решена» или «не решена» данная подзадача, а ИА переходит в новое состояние  $s_j$  ( $\varphi(q, t, s_i) = s_j$ ,  $s_j \in S$ ).

На рисунке 2 представлена обобщенная схема системы для работы с нейронными сетями.

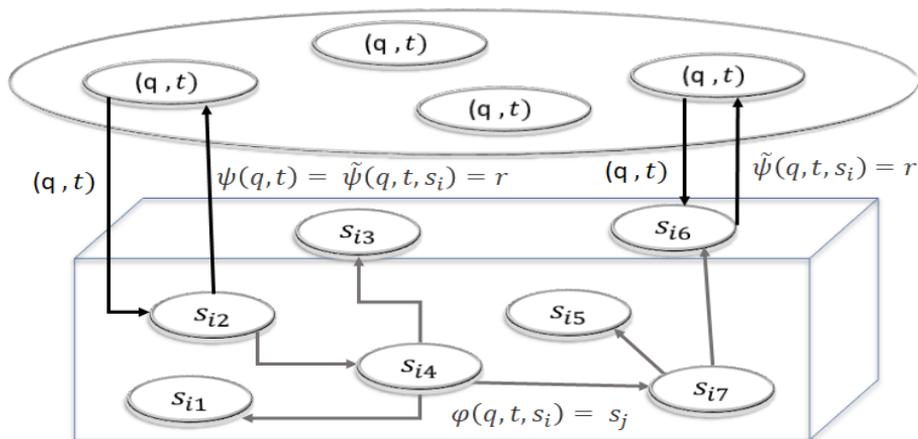


Рис. 2. Схема системы для работы с нейронными сетями

Для любого типа задачи в начальный момент времени в ИА передается специальный параметр с формализованным описанием задачи.

При ( $t = 'database'$ ) в начальный момент времени ИА получает описание запроса к базе изображений. Подобным запросом может быть «проверить количество датасетов изображений в общей базе» или «предъявить список изображений, на которых есть объекты определенных категорий» и т. п.

При ( $t = 'history'$ ) ИА получает из внешней среды содержательную информацию о задаче, связанной с накопленной в системе историей обучения моделей. Например, задача может состоять в просмотре статистики по нейронным сетям, которые обучались классифицировать 2 класса – «кошки» и «собаки».

При ( $t = 'testing'$ ) задача состоит в тестировании предъявленной модели на объектах определенной категории.

При ( $t = 'training'$ ) описание задачи – это категория задачи компьютерного зрения ('classification', 'detection', 'segmentation'), список классов объектов, которые будут распознаваться нейронной сетью ('cat', 'dog' и т. д.) и цель, например, распознавать с точностью 0.9 объекты из соответствующего списка. Уже в состоянии ('check', 'training') (далее компонента 'training' опущена) ИА проводит поиск в базе обученных нейронных сетей на предмет наличия уже обученной модели, соответствующей предъявляемым требованиям. Если такая сеть найдена, то она предъявляется пользователю с предложением закончить работу системы, в случае отказа система переходит в состояние 'data', где по списку категорий изображений создаются 3 множества для обучения и последующего тестирования модели. Затем начинается процесс обучения нейронной сети, в котором система самостоятельно определяет, как будет проходить

процесс обучения нейронной сети, как необходимо изменить гиперпараметры при обучении сети и т. д. Обучение продолжается, пока не будет достигнута требуемая цель, или не будут исчерпаны все возможности системы.

Автоматизация обучения нейронных сетей происходит за счет механизма с использованием «лучших практик». В системе хранится статистика обучения нейронных сетей, именуемая «историей», с описанием процедуры обучения нейронной сети – какая была категория задачи, какие гиперпараметры использовались при обучении, на каких изображениях сеть обучалась, где хранится эта сеть, какие результаты были достигнуты при обучении, какие объекты распознаются сетью и т. д. Первые примеры были записаны экспертом. Последующие записи – это уже результаты работы самой системы.

Из этой истории выбираются практики, соответствующие текущей задаче, затем одна запись определяется в качестве первоначальной стратегии обучения. После чего проводится обучение по этой стратегии. Затем, если требуется скорректировать процедуру обучения, происходит изменение параметров обучения и процесс обучения продолжается.

При необходимости после работы в рамках одной стратегии выбирается другая стратегия и процедура обучения проводится ещё раз. По результатам решения задачи статистическая база пополняется новой информацией о проведенном эксперименте, и уже эта информация может быть использована при решении других задач. Следовательно, рассматриваемая система является самообучающейся.

Таким образом, представленная система решает круг задач, связанных с обучением нейронных сетей. Сама система может быть рассмотрена с точки зрения теории автоматов как композиция двух автоматов – управляющего и исполняющего. Поставленная перед системой задача разбивается на подзадачи, которые последовательно подаются управляющим автоматом исполняющему для их последующего решения, и в зависимости от результата этого процесса определяется следующая подзадача.

В системе используется механизм для автоматизации обучения нейронных сетей – использование при обучении «лучших практик», накопленных в соответствующей базе системы. По результатам проведенного эксперимента в системе сохраняется информация о полученном опыте, который позднее будет ею использоваться при решении схожих задач. Другими словами, сама система является самообучающейся моделью.

Помимо обучения система предполагает выполнение вспомогательных функций, возникающих при работе с нейронными сетями, таких как

тестирование уже обученных моделей, работа с базами данных и анализ накопленного системой опыта.

Проводились эксперименты по обучению нейронных сетей внутри данной системы. Были обучены модели, способные решать задачу бинарной классификации с точностью более 90%. По результатам этих экспериментов система начала накапливать опыт по обучению нейросетевых моделей, который в дальнейшем использовался для решения других задач.

Разрабатываемая система предполагает взаимодействие с пользователем, и может рассматриваться как рекомендательная система для специалистов в области искусственных нейронных сетей, так и как инструмент для обучения нейронных сетей с возможностью последующего использования этих искусственных нейронных сетей в других областях, что делает разработку подобных систем весьма актуальной и востребованной задачей.

Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета «Мозг, когнитивные системы, искусственный интеллект»

## Список литературы

- [1] P.R.Patnaik, “Applications of neural networks to recovery of biological products”, *Biotechnology Advances*, **1:6** (1999), 477-488.
- [2] Victor Skuratov, Konstantin Kuzmin, Igor Nelin and Mikhail K. Sedankin, “Creation of a neural network algorithm for automated collection and analysis of statistics of exchange quotes graphics”, *EUREKA: Physics and Engineering*, **3** (2020), 22-29.
- [3] Jerry A Darsey, William O Griffin, Sravanthi Joginipelli, Venkata Kiran Melapu, “Architecture and biological applications of artificial neural networks: a tuberculosis perspective”, *Methods in molecular biology*, 2015, 269-283.
- [4] Luca Caltagirone, Mauro Bellone, Lennart Svensson, Mattias Wahde, “LIDAR-camera fusion for road detection using fully convolutional neural networks”, *Robotics and Autonomous Systems*, **111** (2019), 125-131.
- [5] Саймон Хайкин, *Нейронные сети: полный курс*, 2-е издание, Издательский дом «Вильямс», Москва, 2006, 1104 с.
- [6] Yang, J. and Honavar, V., “Experiments with the Cascade-Correlation Algorithm”, *Microcomputer Applications*, **17:2** (1998), 40-46
- [7] Steven Abreu, *Automated Architecture Design for Deep Neural Networks*, arXiv: arxiv.org/abs/1908.10714.
- [8] А.С. Подколзин, *Компьютерное моделирование логических процессов. Архитектура и языки решателя задач*, ФИЗМАТЛИТ, Москва, 2008, 1024 с.
- [9] В.Б.Кудрявцев, Э.Э. Гасанов, А.С. Подколзин, *Основы теории интеллектуальных систем*, МАКС Пресс, Москва, 2016, 612 с.
- [10] В.Б. Кудрявцев, С.В. Алёшин, А.С. Подколзин, *Введение в теорию автоматов: Монография.*, 2-е издание, Издательство Московского университета, Москва, 2019, 436 с.

- [11] Ян Эрик Солем, *Программирование компьютерного зрения на языке Python*, ДМК Пресс, Москва, 2016, 312 с.

**Viruykova V.A.**

**Автоматный подход для оптимизации работы системы обучения нейронных сетей**

This paper presents an automata model for learning neural networks, implemented using the high-level programming language python and tested on the problem of binary classification. Also, within the system under consideration, a procedure for automating the neural network training by choosing a training strategy from the point of view of "best practices" is implemented.

*Keywords:* artificial neural networks, finite automata, non-finite automata, system for working with neural networks, automated machine learning, autoML, hyperparameter optimization via best practices, computer vision, CV, binary classification.

## References

- [1] P.R.Patnaik, "Applications of neural networks to recovery of biological products", *Biotechnology Advances*, 1:6 (1999), 477-488
- [2] Victor Skuratov, Konstantin Kuzmin, Igor Nelin and Mikhail K. Sedankin, "Creation of a neural network algorithm for automated collection and analysis of statistics of exchange quotes graphics", *EUREKA: Physics and Engineering*, 3 (2020), 22-29
- [3] Jerry A Darsey, William O Griffin, Sravanthi Joginipelli, Venkata Kiran Melapu, "Architecture and biological applications of artificial neural networks: a tuberculosis perspective", *Methods in molecular biology*, 2015, 269-283
- [4] Luca Caltagirone, Mauro Bellone, Lennart Svensson, Mattias Wahde, "LIDAR-camera fusion for road detection using fully convolutional neural networks", *Robotics and Autonomous Systems*, 111 (2019), 125-131
- [5] Haykin Simon, *Neural Networks. A comprehensive foundation*, the second edition, «Williams», Moscow, 2006 (In Russian), 1104 с.
- [6] Yang, J. and Honavar, V., "Experiments with the Cascade-Correlation Algorithm", *Microcomputer Applications*, 17:2 (1998), 40-46
- [7] Steven Abreu, *Automated Architecture Design for Deep Neural Networks*, arXiv: arxiv.org/abs/1908.10714.
- [8] Podkolzin A.S., *Computer Simulation of Logical Processes. Solver Architecture and Task Solver Languages*, Fizmatlit, Moscow, 2008 (In Russian), 1024 с.
- [9] Kudryavtsev V.B., Gasanov E.E., Podkolzin A.S., *Fundamentals of Theory of Intelligent Systems*, MAKS Press, Moscow, 2016 (In Russian), 612 с.
- [10] Kudryavtsev V.B., Aleshin S.V., Podkolzin A.S., *Introduction to automata theory*, the second addition, MSU Press, Moscow, 2019 (In Russian), 436 с.
- [11] Jan Erik Solem, *Programming Computer Vision with Python*, ДМК Пресс, Moscow, 2016 (In Russian), 312 с.