



САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ «ЛЭТИ»
им. В.И. Ульянова (Ленина)



ПАВЛОВСКИЙ ЦЕНТР
«ИНТЕГРАТИВНАЯ
ФИЗИОЛОГИЯ - МЕДИЦИНЕ,
ВЫСОКОТЕХНОЛОГИЧНОМУ
ЗДРАВООХРАНЕНИЮ
И ТЕХНОЛОГИЯМ
СТРЕССОУСТОЙЧИВОСТИ»



ИНСТИТУТ ФИЗИОЛОГИИ
им. И.П. Павлова
РОССИЙСКОЙ
АКАДЕМИИ НАУК



IEEE

РОССИЙСКАЯ
СЕВЕРО-ЗАПАДНАЯ
СЕКЦИЯ IEEE

NEURONT.ETU.RU

III МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ ПО НЕЙРОННЫМ СЕТЯМ И НЕЙРОТЕХНОЛОГИЯМ

16 июня 2022

NeuroNT'22

СБОРНИК ДОКЛАДОВ



Санкт-Петербург, 2022

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина) (СПбГЭТУ «ЛЭТИ»)

Павловский центр «Интегративная физиология – медицине, высокотехнологичному
здоровоохранению и технологиям стрессоустойчивости»

Институт физиологии им. И.П. Павлова Российской академии наук (ИФ РАН)

Российская Северо-Западная секция Международного института инженеров электротехники
и электроники IEEE

III МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ ПО НЕЙРОННЫМ СЕТЯМ И НЕЙРОТЕХНОЛОГИЯМ (NEURONT'2022)

16 июня 2022 года

Сборник докладов

Санкт-Петербург
2022

УДК 004.032.26

ISBN 978-5-7629-3043-7

III Международная конференция по нейронным сетям и нейротехнологиям (NeuroNT'2022):
сб. докладов. СПб.: СПбГЭТУ «ЛЭТИ», 16 июня 2022. – 79 с.

Международный программный комитет

Почетный председатель: Проф. Ханнес Топфер (Германия)

Сопредседатели:

Проф. М.С. Куприянов (СПбГЭТУ «ЛЭТИ», Россия)

Проф. Ю.Е. Шелепин (ИФ РАН, Россия)

Члены международного программного комитета:

Dr. S.P. Shiva Prakash, Associate Professor, Department of Information Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India

Dr. Balaraman Ravindran, Prof., Computer Science and Engineering & Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, Chennai, Tamil Nadu, India

Dr. Thanh Nghi Nguyen, Prof., Head of the Department of Physics and Mathematics, Hanoi University of Technology, Vietnam

Dr. Nigel Foreman, Prof., Middlesex university, Department of Psychology, London, UK

Проф. Ю.А. Шичкина (СПбГЭТУ «ЛЭТИ», Россия)

Доц. К.В. Кринкин (СПбГЭТУ «ЛЭТИ», Россия)

Проф. В.О. Полякова (СПбГУ, Россия)

Доц. О.И. Буренева (СПбГЭТУ «ЛЭТИ», Россия)

Доц. О.Л. Петросян (СПбГУ, Россия)

Проф. С.А. Вакуленко (ИПМаш РАН, Россия)

Доц. М.П. Чмелевский (ФГБУ «НМИЦ им. В.А. Алмазова», Россия)

Доц. Е.А. Пустозеров (СПбГЭТУ «ЛЭТИ», Россия)

Проф. О.В. Войцеховская (ФГБУН Ботанический институт им. В. Л. Комарова РАН, Россия)

Р.О. Малашин, (ИФ РАН, Россия)

Е.Ю. Шелепин (ИФ РАН, Россия)

Адаптация обучения с подкреплением для библиотеки федеративного обучения FL4J

А. Д. Азаревич¹, М. А. Ефремов²

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

¹azartem@mail.ru, ²jakutenshi@gmail.com

Аннотация. Предметом исследования является возможность реализации и использования федеративного подхода для обучения с подкреплением. Объектом исследования является алгоритм обучения с подкреплением DQN. Данная работа, в частности, рассматривает реализацию взаимодействия библиотеки федеративного обучения на Java – FL4J – с одной стороны, и симулятора среды для обучения с подкреплением OpenAI Gym – с другой. В результате были рассмотрены непосредственно обучение с подкреплением и федеративный метод, а также предоставлена реализация алгоритма обучения с подкреплением DQN на базе FL4J.

Ключевые слова: распределённые вычисления, машинное обучение, федеративное обучение

I. ВВЕДЕНИЕ

В современном мире возникает множество различных задач, которые возможно решить с использованием машинного обучения (МО) – классификация данных, распознавание лиц, голоса, определение объектов и их типов на изображениях. Однако существует ряд задач, в которых необходимо выбрать лучший вариант среди многих или достичь сложной цели за множество ходов. В таких случаях применяется обучение с подкреплением – вид машинного обучения, в котором агент – обучаемая модель – учится тому, что делать, как соотносить ситуации с действиями, чтобы максимизировать числовой сигнал вознаграждения. Агенту не говорят, какие действия предпринять, но вместо этого он должен выяснить, какие действия приносят наибольшую награду, пробуя их [1].

Так же с усложнением задач, к которым стараются применить машинное обучение, появляется проблема с ростом объёма обучающих данных, а, следовательно, и со временем, требуемым для обучения. Данную проблему может решить федеративное обучение (ФО) – разновидность распределённого машинного обучения, в ходе которого в обучении участвуют несколько агентов, которые обучаются на собственных наборах данных, а в передаче данных между ними участвуют только параметры обученных моделей, что позволяет повысить надёжность системы обучения и снизить нагрузку на канал передачи данных, в отличие от случаев, когда пересылаются обучаемые или тестовые данные [2].

II. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

A. Обучение с подкреплением

Обучение с подкреплением – это подмножество методов МО, позволяющих оптимизировать некоторую функцию вознаграждения для агента обучения, чтобы максимизировать функцию вознаграждения. Ученику не

говорят, какие действия предпринять, но вместо этого он должен выяснить, какие действия приносят наибольшую награду, пробуя их [1].

Агент обучения с подкреплением взаимодействует со своей средой дискретными временными шагами. В каждый момент времени t он получает от среды текущее состояние S_t и вознаграждение R_t (где $S_t \in S$ и $R_t \in R$ – элементы множеств состояний среды и вознаграждений соответственно). Далее агент выбирает действие A_t из множества доступных действий $A(s)$, которое затем отправляет в среду. Среда переходит в новое состояние, и определяется вознаграждение, связанное с переходом. Целью обучения агента является изучение стратегии, которая максимизирует ожидаемую награду

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \quad \text{где } \gamma \in [0;1] \text{ – коэффициент дисконтирования будущих вознаграждений.}$$

В данной работе используется алгоритм обучения с подкреплением Deep Q-Network (DQN). Суть данного алгоритма заключается в следующем [3]:

Инициализировать буфер примеров D вместимости N

Инициализировать нейронную сеть $Q(s,a)$, отображающую пространство состояний в множество пар «действие-награда», случайными весами

Инициализировать целевую сеть $\hat{Q}(s,a)$ весами $Q(s,a)$

for episode:=1, M **do**

while episode is run **do**

Случайным образом с вероятностью ε выбрать действие a , в остальных случаях $a = \arg \max_a Q(s,a)$

Выполнить в эмуляторе среды действие a ; получить вознаграждение R и следующее состояние s'

Сохранить переход (s, a, R, s') в буфер D

Выбрать случайный набор переходов из D

$$y = \begin{cases} R, & \text{если эпизод закончился в } s' \\ R + \gamma \max_{a'} \hat{Q}(s', a'), & \text{иначе} \end{cases}$$

Рассчитать потери $L = (Q(s,a) - y)^2$

Обновить $Q(s,a)$, используя алгоритм стохастического градиентного спуска с

минимизацией по потерям L относительно параметров модели

Каждые N шагов копировать параметры из Q в $\hat{Q}$.

od

od

В. Федеративное обучение

Федеративное обучение – подраздел машинного обучения, при которой данные не передаются с источника данных в процессе обучения [2].

ФО отличается от обычного распределенного МО переносом процесса обучения на источники данных. Итоговая модель получается путём агрегирования локальных моделей в глобальную. После агрегации единая модель рассылается обратно на устройства и продолжает обучаться еще несколько раундов. Раундом ФО называется цикл из 3 шагов:

- получение клиентом-владельцем данных модели для обновления;
- выполнение процесса обучения полученной модели;
- отправка модели на сервер агрегации и объединения модели в одну глобальную.

Процесс ФО в целом состоит из 3 фаз:

- инициализация первоначальной, необученной модели;
- N раундов ФО;
- оценка полученной модели.

Критерий применимости ФО:
 $a_g \geq \forall a_i \in A_L = (a_1 \dots a_n)$,

где a_g – результирующая точность глобальной модели, а $A_L = (a_1 \dots a_n)$ – точности локальных моделей. Таким образом ФО предполагает, что за счет нескольких раундов, эффективного метода агрегации и обучения на большем объеме данных, точность глобальной модели должна быть не хуже, чем точность любой иной модели, полученной локально на каждом из клиентов с данными.

III. ОПИСАНИЕ РЕШЕНИЯ

На данный момент среди существующих библиотек не было найдено готовых решений по реализации алгоритмов обучения с подкреплением методом ФО. С другой стороны, имеется несколько статей [4-6], описывающих ряд алгоритмов и подходов к решению задачи ФО с подкреплением, но в приведенных работах не используется ни один из существующих библиотек ФО. Однако в качестве основы была взята разрабатываемая в СПбГЭТУ «ЛЭТИ» библиотека ФО на Java – FL4J [7]. В сравнении с остальными решениями FL4J предоставляет несколько важных преимуществ:

- Поддержка вертикального и горизонтального распределения данных за счет представления алгоритмов в виде составных блоков, поддающихся преобразованию для

использования в обработке как вертикальных, так и горизонтальных данных [8].

- Поддержка различных топологий и сценариев работы вычислительных узлов за счет графового представления связей и планировщика, учитывающего структуру федерации – графа из клиентов и серверов ФО.
- Независимость от языка разработки клиента ФО и технологий реализации взаимодействия между узлами.
- Поддержка различных алгоритмов и моделей МО, не только нейронных сетей.

В данной работе рассматривается разработка модуля библиотеки для работы с обучением с подкреплением.

Одной из основных частей такого модуля является симулятор среды для обучаемого агента. В данный момент стандартом для экспериментов является симулятор OpenAI Gym[9].

Ввиду того, что FL4J реализована на JVM, а Gym – библиотека, написанная на языке Python, было разработано приложение-сервер, предоставляющее доступ к gym-средам, а также Java-клиент, предоставляющий доступ к данному серверу. Данное решение было принято исходя из соображений, что при использовании сервера с Gym появляется возможность обслуживания нескольких клиентов одновременно и в случае возникновения критической ошибки на одном из клиентов сервер продолжит свою работу, что приведет к повышению устойчивости системы.

А. Разработка связи между Gym и Java

Для связи между gym-средой и JVM используется клиент-серверная архитектура: сервер на языке Python предоставляет доступ к заранее сконфигурированным gym-средам посредством HTTP-запросов:

- GET: предоставление информации о доступных средах, пространстве действий и состояний, поддерживаемых конкретной средой;
- POST: создание, сброс к начальному состоянию и удаление среды, а также выполнение средой заданного действия.

Со стороны FL4J был разработан класс `GymClient`, предоставляющий интерфейс обращения к серверу посредством методов:

- `getActionSpace` – метод, возвращающий информацию об пространстве действий;
- `getObservationSpace` – метод, возвращающий информацию об пространстве состояний
- `reset` – метод, переводящий среду в начальное состояние
- `step` – метод, отвечающий за приказ среде выполнить определенное действие. Возвращает объект, описывающий новое состояние среды, информацию о полученной награде и информацию, закончился ли эпизод.

В. Разработка связи между GymClient и FL4J

Для завершения разработки модуля связи FL4J и Gym следует реализовать наследник класса `MiningInputStream`, представляющего собой поток входных данных для обучения, отвечающий за взаимодействие с сервером Gym.

Реализация данного класса, предназначенная для коммуникации между библиотекой и симулятором, должна:

- предоставлять получение данных с сервера Gym;
- представлять полученные вектора данных в соответствии с описанным физическим форматом;
- преобразовывать физические данные в логические;
- предоставлять обратную связь с симулятором для отправки ответного действия агента из множества доступных.

Разработанный в соответствии с данными требованиями класс `GymInputStream` добавляет к базовому классу методы для получения информации об пространствах действий и состояний среды, устанавливания текущего действия по умолчанию, получения текущего состояния среды. При этом был переопределён метод получения очередного вектора данных таким образом, что при его вызове в среду отправляется запрос на выполнение определённого действия, и при этом возвращается вектор, состоящий из n элементов, где первые $n-1$ элементов соответствуют состоянию, в которое перешла среда после выполнения действия `action`, а n -й элемент соответствует полученной награде.

С. Реализация алгоритма обучения с подкреплением

В FL4J в качестве интерфейса для алгоритмов используется интерфейс `AlgorithmBlock`, каждый экземпляр которого содержит ссылки на настройки модели и сам алгоритм. Также в данной библиотеке есть возможность разделения объявления блока алгоритма и реализации его шагов на `EnginedBlock` и `LearningEngine` соответственно. Связь между объявлением и реализацией устроена при помощи передачи имени класса во время объявления, а также с использованием загрузчика класса во время исполнения.

Для алгоритма обучения с подкреплением были реализованы следующие `AlgorithmBlock` и `LearningEngine`:

- `DQNInitializationBlock/DQNInitializationEngine` – используются для инициализации нейронной сети;
- `DQNEvaluationBlock/DQNEvaluationEngine` – используются для получения оценки модели посредством использования её на заданном блоке данных
- `DQNLearningBlock/DQNLearningEngine` – используются непосредственно для обучения модели на переданных данных.

`DQNLearningEngine` устроен таким образом, что в рамках метода, отвечающего за обучение модели создаётся и инициализируется специальный объект `DQNInputStreamDataSetIterator`, преобразующий потоковые данные от `GymInputStream` в наборы обучающих данных, понятных нейронной сети библиотеки глубокого обучения `DeepLearning4J`. Далее, данный объект несколько раз (равное количеству обучающих эпизодов, заданному в настройках алгоритма) передаётся в метод `fit` модели для её обучения. В качестве итоговой точности модели устанавливается значение, равное средней награде, полученной за все эпизоды, пройденные в рамках обучения.

Класс `DQNInputStreamDataSetIterator` реализует основную логику работы DQN – при его инициализации создаётся копия обучаемой нейронной сети, а также буфер примеров. Особенностью работы используемой реализации нейронной сети является то, что она в процессе обучения сначала сбрасывает состояние объекта `DataSetIterator` к начальному значению, после чего запрашивает у него обучающие выборки, пока он способен их генерировать. При этом класс `DQNInputStreamDataSetIterator` настроен таким образом, что на каждый запрос данных от обучаемой модели он делает один шаг в среде, сохраняет результат действия в буфер и возвращает в качестве обучаемых данных выборку из этого буфера. Данные выборки объект способен генерировать до тех пор, пока не закончится эпизод в среде.

IV. ЭКСПЕРИМЕНТЫ

Для проведения тестирования в качестве среды была использована модель `CartPole-v0`: маятник, представляющий собой шест, прикреплен к тележке, которая движется по рельсам без трения. Система управляется путем приложения силы $+1$ или -1 к тележке. В начале маятник стоит вертикально, и цель состоит в том, чтобы предотвратить его падение. Награда $+1$ предоставляется за каждый временной шаг, при котором шест остается в вертикальном положении. Эпизод заканчивается, когда шест отклоняется от вертикали более чем на 15 градусов или тележка перемещается более чем на 2.4 единицы от центра. [10].

В качестве обучаемой модели выступает трёхслойная нейронная сеть. Обучение происходит в рамках 3 раундов ФО по 1000 эпизодов каждый. В каждом раунде используется параметр ε , уменьшаемый по мере работы алгоритма от 1 до 0.02 за 450 шагов. Размер буфера переходов: 20 ; на каждом шаге из буфера выбирается 10 элементов. Коэффициент дисконтирования $\gamma = 0.99$.

Эксперименты проводились с использованием сред из одного, двух и трёх агентов. Для оценки полученных моделей используется средняя награда за 10 эпизодов, проигранных с помощью анализируемой модели.

Результаты экспериментов сведены в табл. 1.

ТАБЛИЦА I РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Количество агентов	№ Эксперимента					Среднее	Дисперсия
	1	2	3	4	5		
1	9.1	9.5	8.7	9.5	9.4	9.24	0.0944
2	9.5	9.6	9.3	9.5	9	9.38	0.0456
3	9.2	9.2	9.7	9.6	9.7	9.48	0.0536

Из результатов экспериментов видно, что средняя награда при использовании нескольких агентов возрастает. Таким образом, использование федеративного подхода позволяет распределённым образом обучать модели агентов, при этом, не передавая данные об обучаемой среде, что может быть критически важно для некоторых предметных областей.

V. ЗАКЛЮЧЕНИЕ

В ходе данной работы была произведена адаптация обучения с подкреплением для библиотеки федеративного обучения FL4J. Так была разработана клиент-серверная архитектура взаимодействия симулятора OpenAI Gym с одной стороны и Java-библиотеки ФО с другой. Так же в рамках данной библиотеки был реализован алгоритм Deep Q-Network.

В конце был проведён ряд экспериментов, показавший возможность совместного применения алгоритмов обучения с подкреплением и федеративного обучения, а также потенциальную перспективность дальнейших исследований в данной области.

СПИСОК ЛИТЕРАТУРЫ

- [1] Sutton R.S., Barto A.G. Reinforcement learning: An introduction. MIT press, 2018
- [2] Liu Y. Federated learning and transfer learning for privacy, security and confidentiality. Y. Liu, Q. Yang, T. Chen et al. AAAI 2019 tutorial. 101 p.
- [3] Roderick M., MacGlashan J., Tellex S. Implementing the deep q-network //arXiv preprint arXiv:1711.07478. 2017.
- [4] Liu B., Wang L., Liu M. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems //IEEE Robotics and Automation Letters. 2019. Т. 4. №. 4. С. 4555-4562.
- [5] Zhuo H.H. et al. Federated deep reinforcement learning //arXiv preprint arXiv:1901.08277. 2019.
- [6] Lim H.K. et al. Federated reinforcement learning for training control policies on multiple IoT devices //Sensors. – 2020. – Т. 20. – №. 5. – С. 1359.
- [7] Efremov Mikhail A., Kholod Ivan I., Kolpaschikov Maxim A. Java Federated Learning Framework Architecture // 2021 IEEEConference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). 2021. Pp. 306–309.
- [8] Kholod I., Shorov A., Gorlatch S. Efficient distribution and processing of data for parallelizing data mining in mobile clouds // J. Wirel. Mob. Networks Ubiquitous. jowua-v11n1-2.pdf
- [9] Gym [Электронный ресурс]. URL: <https://gym.openai.com/> (дата обращения: 13.09.2021)
- [10] CartPole-v0 [Электронный ресурс]. URL: <https://gym.openai.com/envs/CartPole-v0/> (дата обращения: 26.12.2021)

Исследование локальных и глобальных методов объяснимого искусственного интеллекта. Метрики для оценки методов объяснения

Наталья В. Шевская

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*
natalia.shevskaya@etu.ai

Иван В. Шишкин

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*
rayman.4@list.ru

Владислав Д. Степанов

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*
vlstep2200@gmail.com

Александр Д. Облизов

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*
kivant31580@gmail.com

Аннотация. Системы искусственного интеллекта в популярном понимании последнего (машинное обучение) сталкиваются не только с техническими (сбор достаточного количества данных, эффективная аппаратная реализация), и с социальными трудностями при внедрении, причина которых заключается в недоверии из-за непонимания конечными пользователями, как такие системы работают, и почему принимают те или иные решения. Поэтому в данной работе обсуждаются методы объяснимого искусственного интеллекта, нацеленные на решение проблемы непонимания, также рассматриваются существующие подходы к оценке данных методов и определены требования к системе оценки и метрикам, которые в ней могут быть использованы. Также будет дан ответ на вопрос, пронизывающий данную работу, – какова роль гибридного интеллекта в задачах объяснения моделей?

Ключевые слова: искусственный интеллект, машинное обучение, объяснимый искусственный интеллект, ХАИ, LIME, SHAP, качество, метрики

I. ВВЕДЕНИЕ

Одно из направлений развития технологий AI (англ. Artificial Intelligence) – разработка методов объяснимого искусственного интеллекта (далее – ХАИ), позволяющих пользователям понять, почему именно алгоритмы машинного обучения пришли к тем или иным результатам и выводам. Данные методы главным образом нацелены на то, чтобы повысить доверие пользователей к технологиям AI, но их несовершенство подрывает это доверие [1].

Существует множество подходов к классификации методов объяснимого ИИ [2], но наиболее общей является классификация по глобальные (объясняют модель целиком) и локальные методы (объясняют конкретные предсказания).

При оценке методов ХАИ используются разные проблемно ориентированные подходы [3, 4, 5], часто не обладающие свойством переносимости [6]. Для решения данных проблем предлагается определить комплекс метрик, учитывающих широкий спектр технических

характеристик метода ХАИ (например, вычислительная сложность, точность [7]).

II. СУЩЕСТВУЮЩИЕ ПОДХОДЫ К ОЦЕНКЕ МЕТОДОВ ОБЪЯСНЕНИЯ

Рассмотрены 5 основополагающих аналогов, предлагающих уникальные методики оценки методов ХАИ. Часть из них [7, 8, 9] основываются на оценке технических аспектов метода, таких как вычислительная сложность, точность, стабильность. Однако лишь два исследования предлагают полноценное математическое описание процедуры вычисления предложенных метрик. В [10] предлагается проводить оценку на основе социологических исследований, которые позволят определить, насколько методы ХАИ помогают раскрыть модель черного ящика для конечного пользователя. Такой подход, однако, упускает из виду какие-либо технические аспекты, а также предлагает сильно зависящие от объема выборки метрики. В [11] рассматриваются и социологические, и технические метрики, а также возможность их совместного использования в единой системе оценки, однако не предлагает методических или математических описаний того, как вычислять и осуществлять взаимный учет метрик. Также в исследовании приведено разделение метрик на типы и подтипы по способу вычисления и объективности.

III. ТРЕБОВАНИЯ К РАЗРАБОТКЕ МЕТРИКИ ОЦЕНИВАНИЯ МЕТОДА ОБЪЯСНЕНИЯ

На основе анализа существующих подходов к оценке были составлены основные требования к решению:

- учет технических характеристик метода ХАИ (не менее 4 метрик);
- учет социологических и когнитивных характеристик (не менее 2 метрик);
- наличие методического или математического описания каждой метрики;
- возможность учета приоритетов пользователя.

Социологические и когнитивные метрики должны учитывать удовлетворенность, доверие пользователей, понимание результата, выдаваемого методом. Технические метрики метода должны учитывать его точность, устойчивость, асимптотическую сложность и скорость, уязвимость к атакам.

Нормализация метрик и возможность применения к ним весовой функции позволят проводить сравнения методов ХАИ с учетом приоритетов пользователя. Например, если пользователь больше заинтересован в высокой скорости работы метода, чем в точности объяснения, соответствующие метрики в сравнении будут иметь различный вес.

IV. МЕТОДЫ ОБЪЯСНИМОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Исследование методов объяснения, проведенное в [2], позволяет сузить круг исследования локальных и глобальных методов до ряда конкретных методов, описание которых представлено далее.

A. Локальные методы

a) Lime explainer

Lime [12] – это локальный алгоритм объяснения поведения моделей (классификаторов) машинного обучения. Lime способен объяснить любой классификатор типа black box с двумя или более классами. Все, что требуется, – это чтобы классификатор реализовал функцию, которая принимает необработанный текст или массив numpy и выводит вероятность для каждого класса.

b) Anchor explainer

Anchor Explainer [13] основан на статье Рибейро и др. «High-Precision Model-Agnostic Explanations» и основывается на открытом исходном коде первого автора статьи. Идея, лежащая в основе якорей, состоит в том, чтобы объяснить поведение сложных моделей с помощью высокоточных правил, называемых якорями. Эти привязки являются локально достаточными условиями для обеспечения определенного прогноза с высокой степенью достоверности.

c) Integrate Gradient explainer

Integrate Gradient explainer [14] – это метод, первоначально предложенный в статье «Axiomatic Attribution for Deep Networks», целью которого является присвоение значения важности каждой входной функции модели машинного обучения на основе градиентов выходных данных модели относительно входных данных. В частности, интегрированные градиенты определяют значение атрибуции для каждого объекта, рассматривая интеграл градиентов, взятых по прямому пути от базового экземпляра x' к входному экземпляру x . Сравнительная таблица методов представлена в табл. I.

ТАБЛИЦА I ТЕОРЕТИЧЕСКОЕ СРАВНЕНИЕ ЛОКАЛЬНЫХ МЕТОДОВ

Критерии	Методы		
	LIME	Anchor Explainer	Integrate Gradient
Табличные данные	+	+	+
Текст	+	+	+
Изображения	+	+	+
Классификация	+	+	+
Регрессия	+	-	+
Модель искусственного интеллекта	Black box	Black box	TF/Keras

B. Глобальные методы

Были отобраны 3 глобальных метода: Kernel SHAP, Tree SHAP, Accumulated Local Effects. Сравнение возможностей этих методов представлено в табл. II.

Shap [15] используется для определения того, как сильно каждый параметр влияет на результат предсказания, используя значения Шэпли. Основная идея при вычислении значений Шэпли заключается в том, что они вычисляют сначала без определенного параметра, а затем с ним, таким образом, получая влияние этого параметра.

ALE [16] же является более быстрой альтернативой Partial Dependence Plots (PDP). Они сводят сложную функцию прогнозирования к функции, которая зависит только от одного (или двух) параметров. ALE усредняет изменения в прогнозах и накапливает их по сетке.

ТАБЛИЦА II СРАВНЕНИЕ ГЛОБАЛЬНЫХ МЕТОДОВ ОБЪЯСНЕНИЯ

Критерий	Глобальный метод объяснения		
	Kernel SHAP	Tree SHAP	ALE
Модель на входе	Любая (black-box)	На основе дерева решения	Любая (black-box)
Основа объяснения	Используя значения Шэпли	Используя значения Шэпли	Интервальное условное распределение
Необходимость тренировочных данных	Да	Нет	Да
Объяснение единичных случаев, помимо всех данных	Да	Да	Нет

V. ВЫБРАННЫЕ МОДЕЛИ И НАБОРЫ ДАННЫХ

A. Наборы данных и модели для локальных методов объяснения

a) Alzheimer MRI Model

Модель [17] разработана при помощи библиотеки для машинного обучения TensorFlow и представляет собой сверточную нейронную сеть (CNN – convolutional neural network). Данные, на которых обучена нейронная сеть – это снимки магнитно-резонансной томографии, разделенные на два класса (Изображения мозга человека больного болезнью Альцгеймера и здорового человека). Точность обученной модели составляет около 93% с ошибкой равной 0.05

b) Dogs vs. Cats Model

Модель [18] основана на архитектуре сверточной сети MobileNet, построенной при помощи TensorFlow. Датасет, на котором обучена модель, и представляет собой фотографии кошек и собак разных пород, цветов и снятые с разных ракурсов. Точность модели составляет 93 %, а ошибка 0.16.

Результаты обучения выше представленных моделей сведены в табл. III.

ТАБЛИЦА III РЕЗУЛЬТАТЫ ОБУЧЕНИЯ МОДЕЛЕЙ

Критерий	Набор	
	Alzheimer MRI Model (CNN)	Dogs vs. Cats Model (MobileNet)
Точность	93%	93%
Ошибка	0.05	0.16

B. Наборы данных для глобальных методов объяснения

a) HandPD

Набор данных [19] состоит из рукописных спиралей. Две группы людей: здоровые и страдающие болезнью Паркинсона обводят спираль, распечатанную на листе бумаги.

b) Parkinson Disease Detection

При болезни Паркинсона обычно наблюдается заметное влияние на речь. В датасете [20] представлены данные о голосе здоровых и больных людей.

C. Модели для глобальных методов объяснения

Были использованы шесть популярных моделей.

a) Support Vector Machines (SVM)

Набор схожих алгоритмов обучения с учителем.

b) Random Forest Classifier (RFC)

Ансамблевый алгоритм машинного обучения.

c) K-nearest neighbors (KNN)

Метрический алгоритм для автоматической классификации объектов или регрессии.

d) Multi-layer Perceptron classifier (NN)

Многослойный перцептрон – класс искусственных нейронных сетей.

e) Bagging Classifier (BAG)

Bagging – это ансамблевый алгоритм машинного обучения, который объединяет прогнозы из множества деревьев решений.

f) Extra Trees Classifier (ETC)

То же самое, что и предыдущее, только используются чрезвычайно рандомизированные деревья.

Модели сравнивались по следующим метрикам: точность и время, затраченное на обучение модели. Сравнение представлено в табл. IV и V.

ТАБЛИЦА IV Точность обученных моделей

Набор данных	Модели					
	SVM	RFC	KNN	NN	BAG	ETC
HandPD	0.90	0.95	0.82	0.90	0.96	0.92
P. voice	0.84	0.90	0.86	0.87	0.87	0.92

ТАБЛИЦА V ВРЕМЯ РАБОТЫ МОДЕЛЕЙ (СЕК.)

Набор данных	Модели					
	SVM	RFC	KNN	NN	BAG	ETC
HandPD	0.430	0.140	0.004	0.470	0.020	0.120
P. voice	0.460	0.140	0.002	0.470	0.030	0.100

Как видно, по точности лучшими моделями оказались BAG и ETC, а по времени – KNN.

VI. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

A. Исследование локальных методов

Для работы Lime и Anchor Explainer необходимо разделить картинку на суперпиксели. Важно, чтобы суперпиксели отображали логичные части картинки. Например, для картинки важно выделить отдельный части тела (лапы, хвост, морда и другие). Наиболее популярными алгоритмами суперпиксельной сегментации являются SLIC, Quickshift, Compact watershed и Felzenszwalbs's method. Пример разделения каждым алгоритмом изображен на рис. 1.

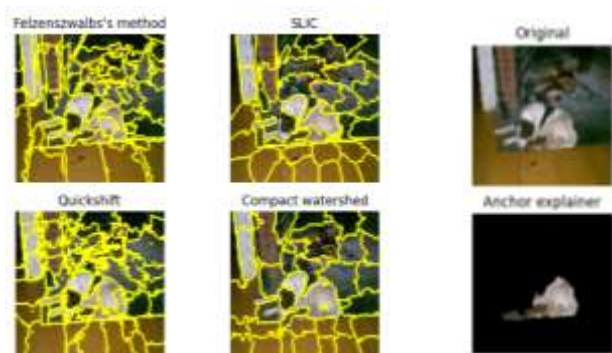


Рис. 1. Разбиение картинки на суперпиксели

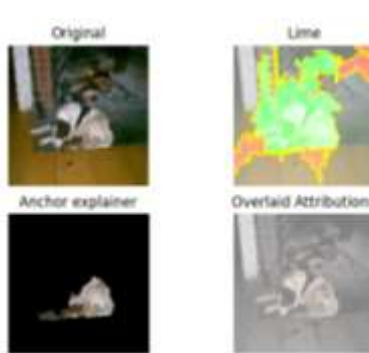


Рис. 2. Результаты объяснения предсказания

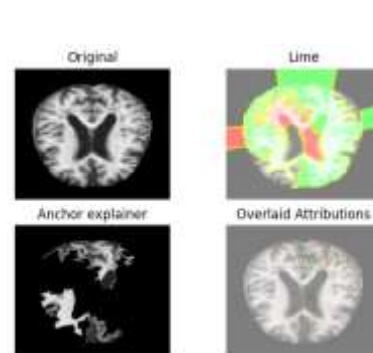


Рис. 3. Результаты объяснения предсказания

Далее будем использовать алгоритм SLIC для разбиения картинки на суперпиксели.

Рассмотрим объяснение предсказания разными методами (рис. 2).

Модель предсказала, что на картинке изображена собака с уверенностью в 98 %. Видно, что каждый из алгоритмов выделил суперпиксели принадлежащие собаке на картинке, что может говорить о том, что модель обучилась правильно и сделала свое предсказание основываясь на части картинки, где изображена собака.

Попробуем запустить методы для модели классификации изображений мозга на наличие или отсутствие болезни Альцгеймера. Результат представлен на рис. 3.

В отличие от изображений кошек и собак, распознать болезнь Альцгеймера по снимку мозга способен на каждый человек. Но предполагая, что модель обучена верно, видно, что каждый из методов выделил верхнюю правую и нижнюю центральную части мозга, можно сказать, что больше всего повлияли на решение модели именно эти части картинки. Корректность принятого решения нельзя комментировать, так как требуется экспертиза профессионалов, работающих в этой области.

Каждый из алгоритмов помимо верных частей картинки выделяет еще другие пиксели, которые трудно интерпретировать как что-то разумное. Это можно объяснить тем, что используемые модели не абсолютно точны. Поэтому для увеличения точности объяснения лучше всего использовать несколько алгоритмов объяснения.

В. Исследование глобальных методов

На двух датасетах и модели ЕТС, как лучшей по соотношению точности и времени (а также может быть

использована во всех трех методах), были произведены эксперименты. На рис. 4 представлены результаты сравнений трех методов по локальной точности и ошибке объяснения. Помимо этого, вычислялось время, затраченного на объяснение. Результат в табл. VI.

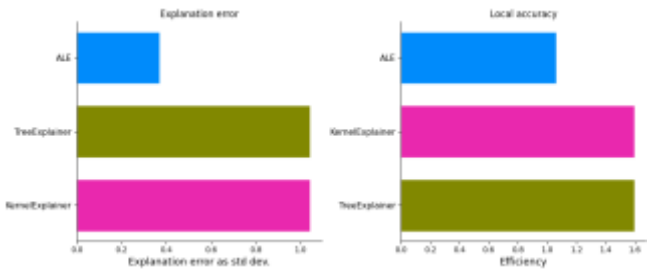


Рис. 4. Результаты сравнения методов

ТАБЛИЦА VI ВРЕМЯ ОБЪЯСНЕНИЯ ГЛОБАЛЬНЫМИ МЕТОДАМИ (СЕК.)

Критерий	Глобальные методы объяснения		
	ALE	Tree SHAP	Kernel SHAP
Время	0.184	2555.1	0.424

Как видно, лучшим по времени оказался ALE, при этом локальная точность больше у Tree SHAP и Kernel SHAP.

На рис. 5–7 представлены графики влияния каждого параметра на примере датасета HandPD.

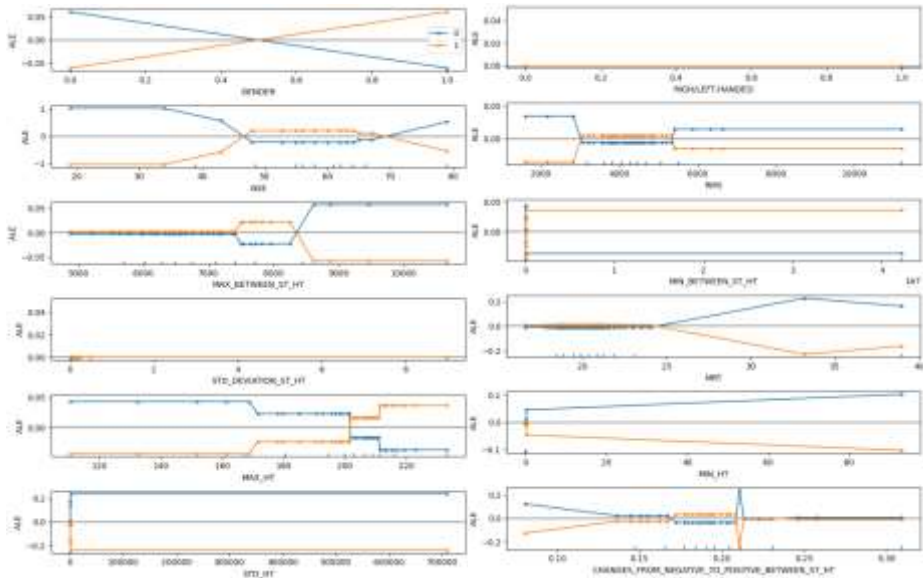


Рис. 5. Оценка параметров датасета HandPD методом ALE

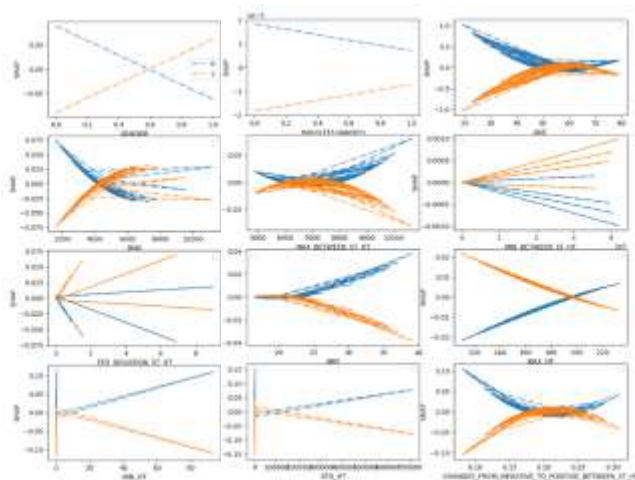


Рис. 6. Оценка параметров датасета HandPD методом Kernel SHAP

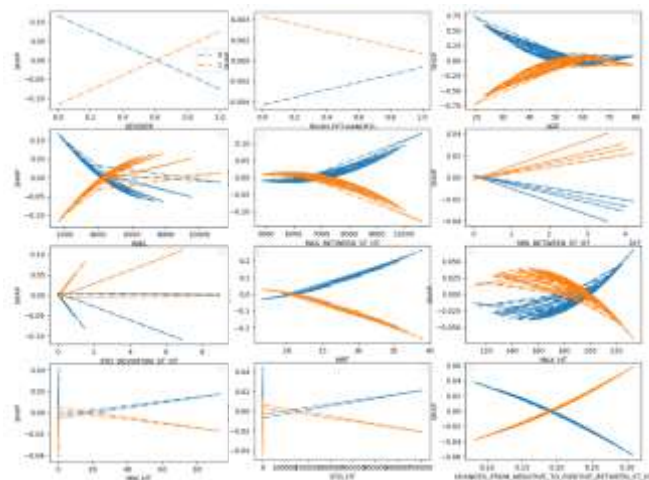


Рис. 7. Оценка параметров датасета HandPD методом Tree SHAP

Как видно, результаты трех методов похожи: самый влиятельный для принятия решения параметр – возраст.

VII. ЗАКЛЮЧЕНИЕ

В результате проведенного исследования были определены требования, которым должна удовлетворять система оценки ХАИ: помимо учета характеристик метода различного типа, необходимы четкие математические или методические описания подсчета каждой метрики. Кроме того, метод должен быть ориентированным на пользователя и должен учитывать возможность учета его приоритетов по тем или иным характеристикам.

Исследование глобальных и локальных методов объяснения показало, что, несмотря на высокую точность и скорость работы моделей, при объяснении результатов методами объяснения требуется верифицировать результаты объяснения, привлекая специалистов предметных областей.

Направления будущих исследований в области оценки качества объяснения также можно направить на математическое описание метрик, их программную реализацию, на способы интегрирования социологических и технических метрик в единую систему оценки.

Одна из важнейших задач, которую можно исследовать в будущем: как сделать так, чтобы верификация результатов объяснения не превратилась в рекурсивную, или какой должна быть метрика оценки качества объяснения – совершенно новой или модификацией существующей. Авторы отмечают, что с большей вероятностью поставленную проблему решит гибридный интеллект, а именно взаимодействие искусственного (модели машинного обучения) и естественного (специалисты предметных областей) интеллектов.

СПИСОК ЛИТЕРАТУРЫ

- [1] Poursabzi-Sangdeh, F., Goldstein, D. G., Hofman, J. M., Wortman Vaughan, J., and Wallach, H., "Manipulating and Measuring Model Interpretability", arXiv e-prints, 2018.
- [2] Shevskaya N.V. Explainable artificial intelligence and methods for interpreting results. Modeling, Optimization and Information Technology. 2021;9(2). Available from: <https://moitvvt.ru/ru/journal/pdf?id=1005> DOI: 10.26102/2310-6018/2021.33.2.024 (In Russ).
- [3] Utkin L.V., Konstantinov A.V., "Ensembles of Random SHAPs", arXiv e-prints, 2021.

- [4] Utkin L.V., Konstantinov A.V., Vishniakov K.A., "An Imprecise SHAP as a Tool for Explaining the Class Probability Distributions under Limited Training Data", arXiv e-prints, 2021.
- [5] Lundberg S. and Lee S.-I., "A Unified Approach to Interpreting Model Predictions", arXiv e-prints, 2017.
- [6] Doshi-Velez F. and Kim B., "Towards A Rigorous Science of Interpretable Machine Learning", arXiv e-prints, 2017.
- [7] Rosenfeld A. "Better Metrics for Evaluating Explainable Artificial Intelligence." Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, 2021.
- [8] Keane M.T., Kenny E.M., Delaney E., Smyth B., "If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques", arXiv e-prints, 2021.
- [9] Lin Y.-S., Lee W.-C., Berkay Celik Z., "What Do You See? Evaluation of Explainable Artificial Intelligence (XAI) Interpretability through Neural Backdoors", arXiv e-prints, 2020.
- [10] Hui-wen Hsiao, J., Ngai, H. H. T., Qiu, L., Yang, Y., and Cao, C. C., "Roadmap of Designing Cognitive Metrics for Explainable Artificial Intelligence (XAI)", arXiv e-prints, 2021.
- [11] Zhou J., Gandomi A. H., Chen F., & Holzinger A., "Evaluating the quality of machine learning explanations: A survey on methods and metrics", Electronics, 10(5), 593, 2021.
- [12] Sundararajan M., Taly A., & Yan Q. (2017, July). Axiomatic attribution for deep networks. In International conference on machine learning (pp. 3319-3328). PMLR. available at: <https://arxiv.org/abs/1703.01365> (accessed 2022-04-04)
- [13] Ribeiro M.T., Singh S., Guestrin C. (2018, April). Anchors: High-precision model-agnostic explanations. In Proceedings of the AAAI conference on artificial intelligence (Vol.32, No.1). Available at: <https://homes.cs.washington.edu/~marcotcr/aaai18.pdf> (accesses 2022-04-04)
- [14] Ribeiro M.T., Singh S., & Guestrin C. (2016, August). "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144). Available at: <https://arxiv.org/abs/1602.04938> (accessed 2022-04-04)
- [15] Interpretable machine learning. Available at: https://originalstatic.aminer.cn/misc/pdf/Molnar-interpretable-machine-learning_compressed.pdf (accessed 2022-04-04)
- [16] Explainable AI: A Review of Machine Learning Interpretability Methods. Available at: <https://www.mdpi.com/1099-4300/23/1/18/pdf> (accessed 2022-04-04)
- [17] Kaggle. Alzheimer's Dataset (4 class of Images). Available at: <https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images> (accessed 2022-04-04)
- [18] Kaggle. Cats vs. Dogs Dataset. Available at: <https://www.kaggle.com/c/dogs-vs-cats/data> (accessed 2022-04-04)
- [19] Kaggle. HandPD Dataset. Approach to Aid the Diagnosis of Parkinson's Disease. Available at: <https://www.kaggle.com/claytonteybaur/meander-handpd-recogna> (accessed 2022-04-04)
- [20] Kaggle. Parkinson Disease Detection Dataset. Available at: <https://www.kaggle.com/debasidotcom/parkinson-disease-detection> (accessed 2022-04-04)

Accelerating Neural Network Training Process on Multi-Core Machine Using OpenMP

Omar T. Mohammed

Department of Computer Science
and Engineering
Saint Petersburg Electrotechnical
University "LETI"
Saint Petersburg, Russia
omar.taha.mohammed@gmail.com

Alexey A. Paznikov

Department of Computer Science
and Engineering
Saint Petersburg Electrotechnical
University "LETI"
Saint Petersburg, Russia
apaznikov@gmail.com

Sergei Gorlatch

University of Münster
Münster, Germany
gorlatch@uni-muenster.de

Abstract—Modern machine learning algorithms when applied to some real-world data, such as social networks and web graphs, can be very time-consuming. Despite enormous researches were focusing on making their approaches more scalable, however, their proposed approaches are running sequentially which makes the training run time remain noticeably long. Thus, it becomes necessary to split the work among multiple processes. This is where we believe parallel processing can help. In this paper, we develop an OpenMP-based approach for parallelizing neural networks on multi-core CPUs. The novelty of our approach is that it is more general as it mainly covers CPU-based parallel training implementation, we focus on accelerating the training phase of neural networks using OpenMP that divides the work among multiple threads to run in parallel. Our experimental evaluation of a binary classification problem run on the banknote authentication dataset which contains images of banknotes with a machine with 12 cores demonstrates a significant acceleration of the training process compared to related works. In the end, we outline some possible approaches for further research concerning parallel optimization of execution time in neural network processes.

Keywords—parallel computing, machine learning, neural networks, OpenMP, Multithreading, Training for supervised learning

I. INTRODUCTION

Machine learning in general and supervised learning in particular are popular and successful in various application areas [1]. Supervised learning is a task of learning a function that maps an input to an output, based on examples of input-output pairs that are called patterns. Usually, the more complex the problem gets the more examples are needed. Training a neural network for complex and multidimensional problems requires using large numbers of training examples with hundreds of thousands or even millions of patterns. Therefore, training process may take prohibitively long time. Furthermore, finding an optimal neural network configuration often requires a certain amount of cross-validation experiments [2]. Hence, one of major challenges in supervised machine learning is that training and testing machine learning models for large, real-world datasets becomes very time-consuming [3].

In this paper, we develop a parallelization approach to accelerate the the training process in the supervised machine

learning, and we implement it for multi-core processors (CPU) using the OpenMP (Open Multi Processing) standard.

II. RELATED WORK

Recent attempts to accelerate neural network training have shown good results, especially on GPU (Graphics Processing Units) [4]. An alternative approach to dealing with big data for neural network models is based on the MapReduce programming model [5]. However, with the increase of the number of parameters which may be very numerous in the neural network models, it becomes a challenge to train the model [6].

Nickolls et al. [7] and Che et al. [8] employ CPU based parallel approaches such as MPI (Message Passing Interface), PThreads, and OpenMP. Papers [7, 8] use CUDA (Compute Unified Device Architecture) to employ streaming processors connected with external dynamic RAM (Random Access Memory) partitions.

Meng et al. [9] present an OpenMP parallelization and optimization of two new classification algorithms based on graphs and PDE (Partial Differential Equations) techniques, with performance and accuracy advantages over the traditional data classification.

In existing approaches, all training examples are processed one set per training update which is usually called at each iteration. This sequential organization increases the training time of neural networks [10]. A common practical solution is to employ the so-called mini-batch training: at each iteration, the neural network processes a subset of training examples until all examples are processed and then aggregates the training updates of the processed subsets. However, the aggregation cost of the mini-batch training causes an additional latency in the training process in large-scale applications that require large subsets of training examples [11]. J. Duchi et al. [12] introduce a parallel training approach that minimizes this latency.

We present in this paper a parallelization approach that exploits multi-core parallelism of modern CPU. We follow the idea of the mini-batch training, i.e., we process subsets of training examples in parallel, but we exclude some subsets from the aggregation of updates after several iterations depending on the update values.

In the remainder of the paper, Section 2 briefly describes the concept of multi-layer neural networks and presents different approaches to the training in neural networks. Section 3 describes our proposed parallelization approach. In Section 4, we explain our parallel implementation on a multi-core CPU using OpenMP. Section 5 presents our

This research was supported by RSCF (project № 22-21-00686).

experimental evaluation using a real-world dataset: we compare our approach against the existing parallelization approach [12] for mini-batch training. Section 6 summarizes our findings.

III. SUPERVISED LEARNING IN MULTU-LAYER NEURAL NETWORKS

Fig. 1 shows a simple illustrative example of a neural network architecture often used in supervised learning – the Multi-Layer Perceptron (MLP) [13]. It consists of an input layer, at least one or more hidden layers, and an output layer. Each layer contains a certain number of nodes which are called neurons, and all neurons of neighboring layers are interconnected [14]. Our approach and experiments in this paper consider significantly more complex structures of neural networks than shown in Fig. 1 by having more hidden layers and neurons.

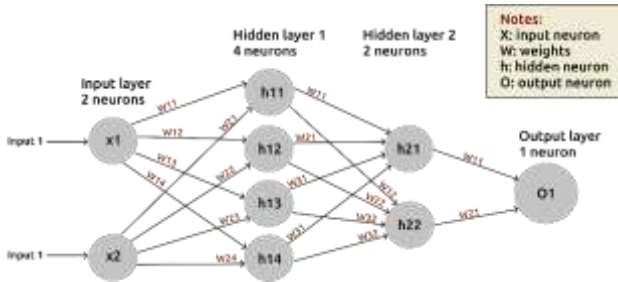


Fig. 1. Example of a 4-layer Multi-Layer Neural Network that has 2 input neurons, two hidden layers with 4 and 2 neurons respectively, and 1 output neuron

We address neural network structures that in terms of complexity are similar to those used, e.g., in Google’s very successful Gboard application: it uses a network with 8 layers for the problem of end-to-end speech recognition on a mobile device. A supervised learning algorithm receives a known set of input data (patterns) and known responses to those data and it trains a model to generate proper predictions about the response to arbitrary new data [15].

In the network like the one in Fig. 1, each connection is associated with a weight value that is usually initialized randomly before the training begins. The job of each neuron is to compute the weighted sum of its inputs or, in other words, compute the sum of all weights in the previous layer and then transmit it into every neuron of the next layer. This transformation is done by an activation function; a widely used activation function for MLP is the sigmoid function: $\text{sigmoid}(x) = 1/(1+e^{-x})$.

Training of an MLP consists in adjusting weights using an optimization algorithm, in order to find a set of weight values that can best map inputs to desired outputs (targets). The training accuracy describes how precise this mapping becomes. The optimization algorithms have two variants: feed-forward and backpropagation. In feed-forward, computations originate from the input layer and proceed to the hidden layers and further to the output layer that finally calculates the output, which is the desired prediction [16].

The difference between the output (prediction) and the target value is called loss. The prediction accuracy of neural networks increases when their loss value decreases. Backpropagation is used to minimize the loss value: it computes the gradient of the loss value with respect to all weight values of a neural network for a single input-output training example to iteratively adjust the weight values. [17]

There are three approaches to backpropagation, as follows. First, in Fig. 2 we show the approach called full batch training. A batch consists of training examples processed before the model is updated. Full-batch training trains the entire batch once per iteration as one set, by computing the feedforward for all training examples to get outputs (predictions) and loss values; then it computes the overall loss as the average of the accumulated loss values. For reducing the loss value, neural network uses backpropagation which updates the weight values per each training iteration. The process is repeated until the average loss value does not decrease anymore; this value is called *local minimum*.

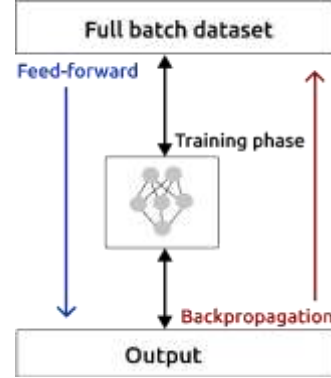


Fig. 2. Full-batch training of a neural network

Second, in Fig. 3 we illustrate the approach called mini-batch training. It divides a batch into subsets of training examples (mini-batches) of size $S > 1$ and then trains each mini-batch separately by applying feedforward per iteration to produce an output (prediction) and a loss value. Afterwards, backpropagation is applied to minimize the loss which is called gradient of loss; the gradient losses are then averaged for all mini-batches. The process is repeated until the average gradient loss does not decrease anymore: it is called *global minimum*.

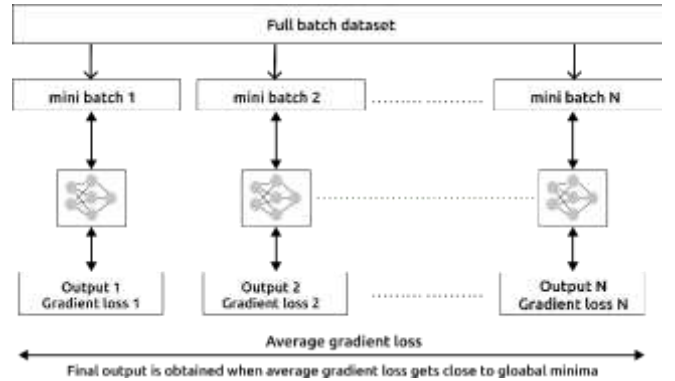


Fig. 3. Mini-batch training of a neural networks

The third approach, called stochastic training, works similarly to mini-batch training, but it performs training on one randomly selected example (one pattern) at a time i.e., the batch size is always equal to one.

IV. OUR PARALLELIZATION APPROACH

Our approach is shown in Fig. 4: we parallelize the mini-batch training by distributing the training of the mini-batches across the cores of a multi-core CPU. Our approach differs from the existing mini-batch training the paper [12]: after a several iterations we exclude some mini-batches from

averaging their gradient loss, based on their gradient loss values as explained in the following.

Fig. 4 shows that our parallelization approach proceeds in two steps as follows. In the first step, we assign each batch to a separate thread: we run threads in parallel to train the assigned mini-batches and produce a gradient loss with an output value per iteration. We repeat the first step for N iterations. Here, N is a hyperparameter to control the training process [18]. In neural networks, hyperparameter values are usually chosen experimentally. Our experiments show that we obtain a fast training process for $N = 100$. We save the gradient loss values of all mini-batches in a container which is called error container. Based on the error container, the second step of our approach selects a percentage (determined by another hyperparameter) of the mini-batches with a minimum gradient loss value for further training and excludes the rest. In our experiments, we obtain good training accuracy when we set the percentage hyperparameter value to 75 %. Finally, we repeat the first step for M iterations and we average gradient loss values until the average gradient loss value does not decrease anymore, i.e., we obtain the global minimum. Hyperparameter M represents the maximum value of training iterations. In our experiments, it was found that a good value for M is 10000.

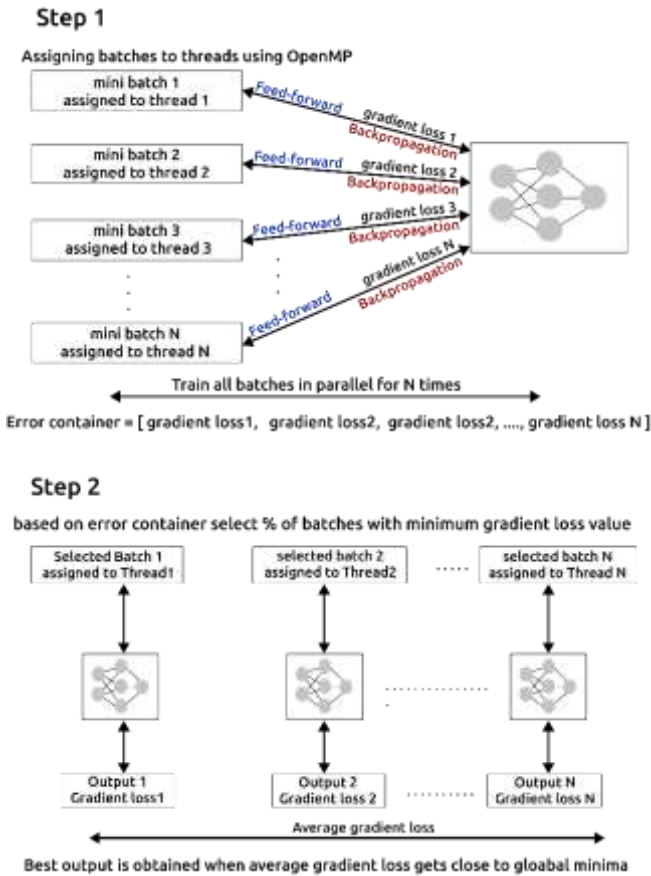


Fig. 4. Our parallelization approach: two-step scheme

V. IMPLEMENTATION IN OPENMP

Fig. 5 shows the implementation of our parallelization scheme described in the previous section: the work is divided between the master thread and several slave threads, according to the general OpenMP paradigm.

In the master thread of our OpenMP C++ program, we first create a new neural network model by initializing the input, the hidden, and the output layers along with their

neurons and connecting them by arrows with the automatically generated random weights ranging between -1 and 1. The master thread reads training examples (dataset) as one batch and splits it into K minibatches, where K is the hyperparameter that influences the batch size. In our experiments, the batch size of 7 provides the fastest training process. The master thread then starts multiple concurrent slave threads and assigns to each of them a separate mini-batch with a copy of the original neural network model. Each slave thread performs the training and updates the weight values based on the current copy of the neural network and the mini-batch assigned to that slave.

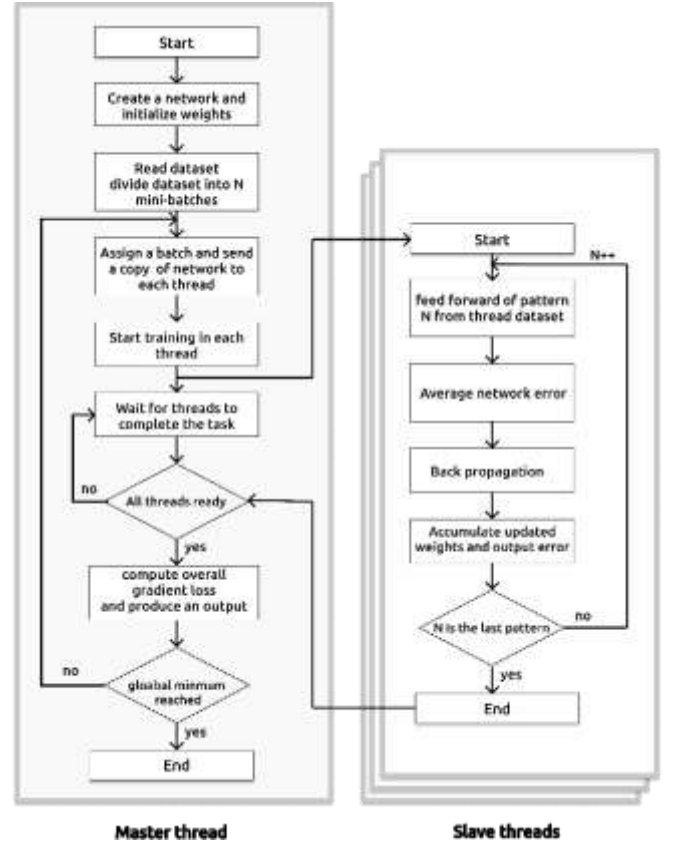


Fig. 5. Our parallel implementation of the backpropagation learning algorithm in OpenMP. Master thread divides the dataset into mini-batches of equal size. Slaves train the neural network using their subsets training examples. Master then computes the overall weights update (gradient loss) of the neural network

As soon as all slaves finish their computations, they send accumulated updated weight values to the master thread; the master thread combines them to update the overall weight values in order to minimize the gradient loss of the neural network. This process is repeated until the gradient loss does not decrease anymore or until M is reached.

VI. EXPERIMENTAL EVALUATION

For experiments, we use the Banknote Authentication Dataset [19] which is a representative real-world use case for a binary classification representation. The dataset contains 1372 patterns which are images of banknotes with 4 predictor variables. The goal is to predict whether a given banknote is authentic or not, based on some measurements taken from a photograph. For all binary classification problems, the output layer of the neural network consists of one neuron that produce one output (prediction) and its value is a probability ranging between 0 and 1. We choose the binary classification representation for evaluating our

approach because of its low computational cost with fewer hyperparameters than other classifiers and also because of its competitive accuracy measures [20].

We experiment with a dataset that is complex, but not very large, because using a complex and very large dataset would require a machine with more parallel cores than the one available in our experiments.

All experiments are conducted on the following multicore processor: Intel Core i7-8750HQ 2.20 GHz, with 12 Cores and 16 GB RAM.

The structures of neural networks in our study are motivated by [21] and chosen after extensive experiments. We show in Table 1 two different neural network structures: the first network consists of 3 hidden layers with 18 neurons in each layer and overall, 5 layers including the input and output ones; the second network has 4 hidden layers, each having 24 neurons, overall, 6 layers. The complexity of the neural networks used in our experiments is similar to the networks currently used in practice. For example, Google’s recent Gboard application for speech recognition uses an on-device neural network with the structure based on paper [22]: it has only 8 layers for solving a very complex problem of streaming end-to-end speech recognition on a mobile device. This application is very successful in practice.

In Table 1, we compare our proposed approach against the existing mini-batch training approach [12]. In our experiments, we start the training process for both approaches using the same initial generated random weight values of the network. To obtain more precise results, we conduct each experiment several times and then average the computed values. The table contains information on the neural network configurations, batch size, number of used threads, and the measured execution time of the training.

Table 1 demonstrates an obvious trend that using more threads than available processor cores cause some synchronization overhead which lowers the efficiency of parallelization. We also notice that our approach has a faster training process but slightly lower accuracy compared to the mini-batch training approach in [12]. A probable reason is that our approach excludes some training examples during the training process.

Fig. 6 compares the speed-up of two OpenMP-based parallelization approaches: the existing mini-batch training [12] and the approach described in this paper.

We observe in the figure that our proposed approach achieves its best speed-up when it uses a medium batch size with as many threads as available cores. Every core is loaded with the training of the assigned patterns, and a relatively short synchronization time is needed for updating the weights of the master thread and copying them back to the slave threads. In Fig. 6 we see that the fastest training is obtained when using 12 threads on a 12-core CPU.

We also observe that the achieved speed-up is quite modest. We explain this by the general restrictions of the supervised learning based on mini-batch approach [23]: 1) to compute the overall gradient loss update, the master thread needs to combine the gradient loss updates received from all slave threads – this reduction step is an extra work compared to sequential execution; 2) for every batch, the parallel method performs two thread synchronizations – after all slaves complete their gradient loss update, and after the reduction step is done;

TABLE I. EXPERIMENTAL RESULTS: OUR APPROACH VS. EXISTING MINI-BATCH PARALLELIZATION IN [12]

Training Algorithms	Threads	Hidden Layers	Neurons	Execution Time	Accuracy
Proposed approach	10	3	18	28 ms	92.4 %
Existing mini-batch approach	10	3	18	35 ms	94.3 %
Proposed approach	12	3	18	18 ms	88.7 %
Existing mini-batch approach	12	3	18	24 ms	93.6 %
Proposed approach	14	3	18	31 ms	72.2 %
Existing mini-batch approach	14	3	18	35 ms	94.3 %
Proposed approach	10	4	24	35 ms	83.3 %
Existing mini-batch approach	10	4	24	42 ms	72.7 %
Proposed approach	12	4	24	29 ms	92.4 %
Existing mini-batch approach	12	4	24	38 ms	97.2 %
Proposed approach	14	4	24	34 ms	87 %
Existing mini-batch approach	14	3	18	48 ms	84.1 %

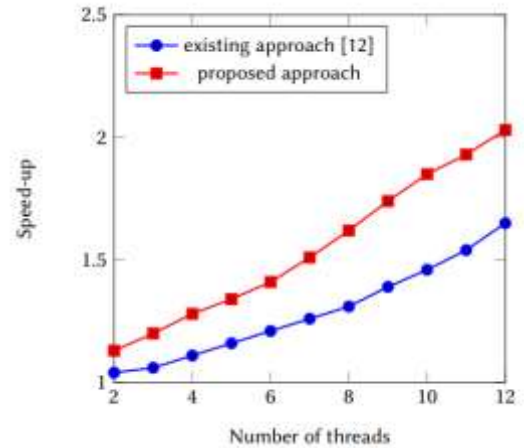


Fig. 6. Speed-up of our parallelization vs. existing minibatch parallelization [12]

and 3) since the overall update of the model is not applied until the end of the entire batch (dataset), updates within the batch become increasingly obsolete, as being based on a model that is out of date. We plan to address these restrictions of our approach in future work.

CONCLUSION

In this paper, we propose a parallelization approach to supervised learning of neural networks. Our approach is implemented using OpenMP. Our experimental evaluation runs on the processor with 12 parallel cores for the banknote authentication dataset that contains images of banknotes for deciding which of them are genuine and which are not. We demonstrated the improvements made by our approach

regarding the speed-up of the training process as compared to the existing parallelized mini-batch training.

REFERENCES

- [1] J. Li, Regression and classification in supervised learning, in: Proceedings of the 2nd International Conference on Computing and Big Data, Association for Computing Machinery, New York, NY, USA, 2019, p. 99–104. doi:10.1145/3366650.3366675)
- [2] S.B. Kotsiantis, Supervised machine learning: A review of classification techniques, in: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies, IOS Press, NLD, 2007, p. 3–24.
- [3] O. Alonso, Challenges with label quality for supervised learning, *J. Data and Information Quality* 6 (2015). URL: <https://doi.org/10.1145/2724721>.
- [4] O. Yadan, K. Adams, Y. Taigman, M. Ranzato, Multi-GPU training of convnets, 2013. arXiv:1312.5853.
- [5] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, K. Olukotun, Map-Reduce for machine learning on multicore, in: Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS'06, MIT Press, Cambridge, MA, USA, 2006, p. 281–288.
- [6] L. Mai, A. Kolioussis, G. Li, A.-O. Brabete, P. Pietzuch, Taming hyper-parameters in deep learning systems, *SIGOPS Oper. Syst. Rev.* 53 (2019) 52–58. URL: <https://doi.org/10.1145/3352020.3352029>.
- [7] J. Nickolls, I. Buck, M. Garland, K. Skadron, Scalable parallel programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for?, *Queue* 6 (2008) 40–53. doi:10.1145/1365490.1365500.
- [8] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, K. Skadron, A performance study of generalpurpose applications on graphics processors using CUDA, *Journal of Parallel and Distributed Computing* 68 (2008) 1370–1380. doi:<https://doi.org/10.1016/j.jpdc.2008.05.014>.
- [9] Z. Meng, A. Koniges, Y. H. He, S. Williams, T. Kurth, B. Cook, J. Deslippe, A. L. Bertozzi, Openmp parallelization and optimization of graph-based machine learning algorithms, Lawrence Berkeley National Laboratory 9903 (2016).
- [10] V. Md, S. Misra, G. Ma, R. Mohanty, E. Georganas, A. Heinecke, D. Kalamkar, N. K. Ahmed, S. Avancha, Distgmn: Scalable distributed training for large-scale graph neural networks, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3458817.3480856
- [11] L. Ziyin, K. Liu, T. Mori, M. Ueda, On minibatch noise: Discrete-time sgd, overparametrization, and bayes, *ArXiv abs/2102.05375* (2021).
- [12] S. Chaturapruek, J.C. Duchi, C. Ré, Asynchronous stochastic convex optimization: the noise is in the noise and sgd don't care, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 28, Curran Associates, Inc., 2015.
- [13] J. Singh, R. Banerjee, A study on single and multilayer perceptron neural network, in: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 35–40. doi:10.1109/ICCMC.2019.8819775.
- [14] O.T. Mohammed, M.S. Heidari, A.A. Paznikov, Mathematical computations based on a pretrained AI model and graph traversal, in: 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020, pp. 1–4. doi:10.1109/MECO49872.2020.9134081.
- [15] J. Alcaraz, S. Sleder, A. TehraniJamsaz, A. Sikora, A. Jannesari, J. Sorribes, E. Cesar, Building representative and balanced datasets of openmp parallel regions, in: 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2021, pp. 67–74. doi:10.1109/PDP52278.2021.00019.
- [16] I. Kholod, A. Rukavitsyn, A. Paznikov, S. Gorlatch, Parallelization of the self-organized maps algorithm for federated learning on distributed sources, *The Journal of Supercomputing* (2020). doi:10.1007/s11227-020-03509-2.
- [17] H. B. Demuth, M. H. Beale, O. De Jess, M. T. Hagan, *Neural Network Design*, 2nd ed., Martin Hagan, Stillwater, OK, USA, 2014.
- [18] R. Liu, S. Krishnan, A. J. Elmore, M. J. Franklin, Understanding and optimizing packed neural network training for hyper-parameter tuning, *DEEM '21*, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3462462.3468880.
- [19] V. Lohweg, Banknote-authentication dataset, *Dataset* (2012). URL: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>.
- [20] K.-A. Toh, Z. Lin, L. Sun, Z. Li, Stretchy binary classification, *Neural Netw.* 97 (2018) 74–91. doi:10.1016/j.neunet.2017.09.015.
- [21] M. Madhwarasan, S. N. Deepa, A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting, *Applied Intelligence* 44 (2016) 878–893. doi:10.1007/s10489-015-0737-z.
- [22] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, et al., Streaming end-to-end speech recognition for mobile devices, in: *ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6381–6385.
- [23] S. Sallinen, N. Satish, M. Smelyanskiy, S. S. Sury, C. Ré, High performance parallel stochastic gradient descent in shared memory, in: 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2016, pp. 873–882. doi:10.1109/IPDPS.2016.107.

Подходы к классификации этических проблем нейросетей

Н. Н. Покровская

Санкт-Петербургский государственный электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

nnp@spbstu.ru

Аннотация. В статье анализируется ряд кейсов применения нейронных сетей как технологий решения социальных задач в разнообразных сферах деятельности. Выявлены типовые проблемы извлечения выводов и совершения выбора нейронными сетями в случаях, когда решения опираются на личностные, социально-демографические, политико-экономические и административно-географические критерии. Приведены основные особенности применения социально-этического, социокультурного и экономико-управленческого подходов к проблемам этики в функционировании нейросетей. Сформулированы некоторые соображения по разрешению этих проблем в каждом из подходов.

Ключевые слова: этика, нейросеть, дискриминация, стагнация, социокультурная регуляция, культурная инерция, управленческая эффективность

I. ВВЕДЕНИЕ

Внедрение нейросетевых приложений потребовали не только их инженерно-технического совершенствования, но и соответствия общественным правилам взаимодействия, от этикета и вежливости до этики и недискриминации.

Социальная регуляция в традиционном общественном укладе основана на статистической частоте поведенческих паттернов. Оценка высокой вероятности события переводится в предрассудки, суеверия, предубеждения и стереотипы, которые наследуются в рамках механизмов культурной инерции. Стереотипы помогали людям делать быстрый выбор, который мог предотвратить опасные для общества ошибки, чем выше угроза и быстрее требуется реакция, тем эффективнее механизм предвзятых суждений.

Стереотипное поведение может иметь травмирующие последствия, социологи определяют стигматизацию как процесс отнесения черт личности индивида к свойствам социальной категории, как правило, отрицательных качеств, но также и нейтральных, или положительных (например, «все бабушки добрые и ласковые», «эта молодёжь только болтает в блогах и ничего не умеет»).

Корпоративные управленческие практики направлены на снижение рисков, например, подбор персонала до конца II тысячелетия включал в себя логику «мы не нанимаем молодых женщин, они выйдут замуж, нарожают детей и осядут дома», или «мы увольняем старых людей, потому что они с трудом осваивают новые технологии». Сегодня корпоративное управление не допускает таких практик в рамках политики недискриминации и этических кодексов.

С последней трети XX века, возросла оценка вклада талантливой личности в благосостояние общества, человеческий капитал стал ключевым фактором в

цепочке создания стоимости [1]. В экономике знаний компетенции, личностные качества, харизма и талант играют решающую роль в успешном развитии страны, региона, частной компании и общественного учреждения, политической группы, гражданской инициативы [2]. Со статистической точки зрения, как социальная, так и управленческая практики были экономически оправданы, расчет рисков включает вероятность: значительных угроз следует избегать даже с более высокими затратами.

Машинное обучение идет по тому же пути в научении социальной жизни. Эта траектория включает типовые тенденции: нейронные сети, обученные на предыдущих фактах дискриминации, склонны фиксировать и усиливать её – например, расистские, сексистские и эйджистские мотивы определяют решения искусственного интеллекта (ИИ) по отбору и найму персонала (оценивая структуру штата компании по критериям, выявленным самим ИИ). Алгоритмы фиксируют и углубляют поляриность оценок.

Этические проблемы выявляются не во всех группах стран (например, в Японии Минздрав одобрил требование компании к женщинам носить обувь на каблуках, потому что женские ноги выглядят красивее на каблуках). Разграничение этических вопросов имеет культурный характер (этнический, религиозный, исторический). В то же время ряд ценностей кажутся всеобщими, например, жизнь и здоровье человека, справедливость и правда, разумность и рациональность. Иерархия ценностей определяет три основных аспекта этической проблематики в связи с внедрением нейронных сетей – культурное, этическое и экономическое, рассмотренные в статье.

II. АНАЛИЗ И КЛАССИФИКАЦИЯ ЭТИЧЕСКИХ ПРОБЛЕМ НЕЙРОТЕХНОЛОГИЙ

A. Участники проблемных ситуаций и субъекты обнаружения проблем

Первоначальный интерес к ошибкам, допущенным ИИ при категоризации людей, был основан на реакции трех групп населения: затронутых неэтичными решениями ИИ людей, журналистов, исследователей и специалистов.

- Пострадавшие люди реагируют в социальных сетях и рассказывают близким, друзьям и коллегам о замеченной ошибке, недоразумении или неверной классификации (так, изображение чернокожей женщины было определено в каталог горилл [3]). Обычно, такие ошибки обнаруживаются случайно или по уведомлению системы (гражданину Новой Зеландии при попытке получить паспорт десятки раз вернулось

уведомление, что на фото у него закрыты глаза, и он должен сфотографироваться с открытыми глазами) [4, 5]. Эти примеры известны благодаря сообщениям людей в социальных сетях.

- Журналисты публикуют такие истории в СМИ, как для сенсации, так и в связи с серьёзной заботой о будущем человечества с машинными алгоритмами, регулирующими жизнь людей.
- Исследователи, IT-профессионалы и специалисты по этике исследуют кейсы и логику алгоритмов, чтобы понять, почему массовое человеческое поведение, приемлемое для людей, не приемлемо во взаимодействии человека и машины (например, если резюме кандидата на должность медсестры отвергается отделом кадров и HR-менеджером, их меньше винят в дискриминации, чем машинный алгоритм – мы пытаемся оправдать и объяснить поведенческий выбор человека, но мы куда менее милосердны к машинным кодам).

В. Группы нейротехнологий, «обвиняемые» в нарушении этических требований

Нейротехнологии продемонстрировали значительные положительные результаты в таких сферах деятельности, как медицинская диагностика, правовая предиктивная аналитика, прогнозирование увольнений сотрудников (по их активности в корпоративных соцсетях и чат-ботах). В то же время есть области и группы интеллектуальных технологий, которые вызывают менее позитивное впечатление, что вредит имиджу нейросетей в целом как эффективных и забавных помощников человека в быстром и эффективном принятии решений.

- Преимущества рекомендательных алгоритмов привели к их распространению благодаря удобству пользователя, уже после одного дня использования социальной сети пользователю не нужно подбирать контент (Tiktok известен успешным мониторингом предпочтений пользователей), потому что система рекомендует пользователю содержание, похожее на ранее им выбранное. Такая система образует «информационный пузырь», через который трудно проникнуть в другие области знаний или мнений.
- Технология распознавания лиц является полезным инструментом для полиции и силовых структур, но в то же время, если компания или город внедряет «эмпатические» технологии для изучения эмоций и настроения сотрудников или жителей, мониторинг провоцирует два типа опасений: во-первых, контроль эмоций превышает «нормальное» право руководства контролировать подчинённых; во-вторых, ошибочное опознание человека приводит к его появлению в базе данных полиции или службы безопасности под ярлыком «неблагонадёжных».
- Эмпатия представляет собой нормальную практику между людьми, но машинное обучение может дать сбой, когда оно используется для обнаружения внутреннего мира людей. Так, машина путает намерения людей (задача

полицейской системы распознавания эмоций состояла в предотвращении преступных действий) и их привычную мимику (полицейская система классифицировала грустных и угрюмых людей как преступников, а улыбчивых и счастливых – как добросовестных граждан).

- Технологии дипфейка привели к ряду улучшений в банковской безопасности, после анализа фактов снятия средств с помощью фальсифицированных биометрических данных, голоса, изображения и т.д. Эти нейротехнологии используются как инструмент конкуренции в информационных войнах между корпорациями, странами и даже обычными гражданами, появились публикации фейкового видео с изображением соседей или коллег в неприглядных обстоятельствах.

Перечень не исчерпывает этические проблемные зоны при применении нейротехнологий, он лишь отражает группы накопившихся кейсов.

С. Типы этических проблем

Этика – сложная область, в которой люди полагаются на интуитивное оценочное суждение. По критериям оценки можно выделить несколько проблемных областей.

- Процедура отбора обычно учитывает социально-демографические факторы. Раса, пол и возраст являются известными критериями дискриминации и вызывают этические и экономические вопросы – с этических позиций отбор по демографическим параметрам слишком ограничен и препятствует личностному росту, образованию и развитию; в экономике оптимальное включение человеческого капитала требует выбора на основе потенциального результата, который можно получить от человека. Это рассуждение отражает процесс расстановки приоритетов – демографические параметры менее важны, чем личные предпочтения и способности. Индивидуальная личная свобода самореализации приоритетна в постсовременном обществе. Нейро-технологии исследуют параметры массива людей (например, все профили в социальной сети) и строят алгоритмы выбора на распространенных, но не оптимальных критериях.
- Личный выбор (религия, интересы, предпочтения, гендерное поведение) определяет включение (или исключение) человека в сообщество: социально-профессиональное, по интересам, локальное. Выбор в символическом потреблении влияет на успех, например, топ-менеджеры в переговорах демонстрируют предметы роскоши как символ успеха. Религиозные убеждения или экологическое поведение понимаются как частная жизнь, но они широко представлены в Интернете (например, участие в беспорядках, гражданская инициатива и т. д.), нейронные сети накапливают эти данные и оценивают статистические корреляции и вероятное поведение индивида. Прогностические технологии нередко приводят к неправильной

атрибуции намерения или свойства к конкретной личности.

- Политико- и социально-экономические показатели часто имеют значительное влияние, например, на образование человека, его манеры и модели восприятия и поведения. Но степень социальной интеграции в общественную систему (например, уровень доходов и участие в выборах) дает ограниченное понимание потенциального вклада человека в благополучие региона или компании. Нейронные сети не могут обнаружить, например, намерение человека сменить профессию, сферу деятельности, экономическую позицию, например, планы дауншифтинга или решение работника компании уволиться и начать свой малый бизнес.
- Административно-территориальное расположение и «принадлежность» легко обнаруживаются с помощью геолокации. Рождение и проживание в каком-либо месте существенно для формирования личности, социокультурного опыта. Это богатство разнообразия используют в многонациональных мультикультурных командах, но нейротехнологии иногда ограничивают потенциальное планирование карьеры людей с их географическим положением.

III. РЕАЛИЗАЦИЯ СОЦИАЛЬНО-ЭТИЧЕСКИХ, СОЦИОКУЛЬТУРНЫХ И ЭКОНОМИЧЕСКО-УПРАВЛЕНЧЕСКИХ ПОДХОДОВ К ЭТИКЕ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Теоретический анализ этических вопросов регуляции основан на 3 компонентах: фундаментальная абсолютная универсальная этика как основание для любых оценочных суждений; социальные условности, принятые сообществом (культура); инструментально-технические аспекты выбора наиболее эффективного средства достижения цели влияют на способ получения совокупности результатов и затрат, реализуя рациональность в качестве основного принципа.

Компоненты универсальной ценности, культурных условностей и экономической рациональности образуют три группы подходов, которые могут быть выявлены при анализе проблем этики при внедрении нейротехнологий. Изучение нескольких примеров подтверждает выделенные группы подходов в следующей классификации.

A. Социально-этический подход

Самый широкий подход основан на том, что все виды этики возникают в социальных отношениях людей. Шкала фундаментальных ценностей включает приоритеты жизни над смертью, удовлетворения над потребностью, мирных переговоров над насилием и т. д. Шкала предполагает, что люди автономны, здоровы и разумны, обладают всей информацией, необходимой, чтобы сделать выбор. Эти принципы не рассчитывают статистически, их надо задавать нейронным сетям на начальном этапе машинного обучения, они не рациональны с «экономической» оценки сопоставления результатов и затраченных ресурсов. Психологические эксперименты показали, что ценности справедливости и честности присущи детям с возраста 6–8 месяцев,

млекопитающие распознают несправедливость и настроены на протест против нечестного дележа. Машина не обладает от «природы» врожденными ценностями.

B. Социокультурный подход

Каждое общество на протяжении длительной истории создает образцы реакций на многообразие природы и вызовов окружающей среды. Культура представляет собой основные модели трех инструментов:

- Схема интерпретации действительности помогает членам сообщества иметь одинаковое толкование фактов. Общая трактовка основана на предыдущем опыте и служит долгосрочному выживанию и развитию сообщества. Единое видение мира помогает достичь целей и одновременно выделяет зоны неопределенности для новых решений.
- Шкала ценностей конкретного общества отражает интенциональное отношение к миру: поведение отражает намерение достигать привлекательных позиций («положительные ценности») и состояний и избегать ситуаций и обстоятельств, оцениваемых как неприятные. Ценности составляют сердцевину религий, определяя правильное поведение людей и направляя к лучшим для общества условиям.
- Способы реализации ценностей образуют нормы и регуляцию поведения человека через принятые каналы удовлетворения потребностей. Обычно их легко формализовать, поскольку они проявляются видимым образом, артикулировать и представить в вербальной форме (законы, правила, наставления, сказки, песни, легенды, мифы и т. д.). Зависимость от сделанного выбора отражается в концепции «path dependency», краткосрочного ситуационного предпочтения перед долгосрочным оптимальным решением. Уже вложенные ресурсы в освоение неоптимального решения компенсируются каждым членом общества. Переход с одного «пути» на другой, более эффективный, требует образования и компетентности, а иногда и нового оборудования [6, 7]. Нормативная регуляция легко отражается в цифровых инструментах, так, смарт-контракты автоматизируют уже существующие процедуры.

Социокультурный подход учитывает различия между сообществами и их предыдущий выбор. Разнообразие культур связано с эволюцией и выживанием человечества.

C. Экономическо-управленческие рассуждения

Экономическое обоснование близко к технологическим решениям в кибернетике, вычислительной технике и науке о данных. Экономика рассматривает данные как абстрактные измерения любых величин и интерпретирует их в четком математическом моделировании. Цифровая трансформация определяет широкое внедрение нейронных технологий в социально-экономическую жизнь. Корпоративное управление опирается на свод правил для решения экономических задач. Задачи

оптимизации переносятся в ИТ-решения для максимизации результата.

Противоречие возникает между кратко-, средне- и долгосрочными экономическими целями – на кратком горизонте планирования математические решения и принятие решений, основанных на эффективности, дают оптимальный результат. Долгосрочное прогнозирование дает иногда противоположную оценку – если нейросеть экономит затраты на подбор кадров (с алгоритмами отбора по половозрастным стереотипам и по заранее выстроенной кадровой структуре), то в долгосрочной перспективе компания утратит эффективность с отказом от более эффективных человеческих ресурсов (при оценке человеческого капитала по статистическим, а не содержательным критериям) и может получить конфликт с социальными группами, обвинения в дискриминации и недостаточной социальной ответственности. [8]

Д. Возможные решения для групп

Упомянутые выше подходы дают ряд инструментов для направлений деятельности, помогающих хотя бы частично решить этические проблемы. Сгруппированные видения таких решений представлены в таблице.

ТАБЛИЦА 1 Подходы к ЭТИЧЕСКИМ ПРОБЛЕМАМ ПРИ РЕАЛИЗАЦИИ НЕЙРОННЫХ СЕТЕЙ

Подход	Сущность проблемы и направление решения		
	Характер основных этических проблем	Принцип решения	Пример
Социально-этический	Все живые существа имеют универсальные ценностные приоритеты, в том числе люди, которые могут распознавать и проводить ценностное состояние по оси критерия	Машинное обучение включает в себя шкалу начальных значений	Нейронные сети начинают с усвоения базовых правил
Социокультурный	Иерархии ценностей условны и могут анализироваться нейротехнологиями, но нейронные сети не способны различать уровни приоритета в каждой культуре. Человеческий опыт является в процессе обучения с первых дней в культуре, нейронные сети должны приобрести отличие этих глубоко укоренившихся убеждений	Слои базовых правил могут включать культуру как критерий высшей степени	Экспертное машинное обучение помогает сети выявить свойства культурной среды
Экономико-управленческий	Экономическо-административная функция в институционализированных единицах помогает организовать процессы для достижения целей. Но менеджмент, администрация или управление ориентированы на разные горизонты планирования.	Алгоритмы оценки машинного обучения вводят обязательное долгосрочное планирование	Сетка горизонтов моделей для организации процесса измерения и прогнозирования

^a. Составлено автором

Эти потенциальные типы решений требуют сложной системы предустановленных правил и процедур в процессе выбора в работе многослойных нейронных сетей. Цифровизация отвечает на вопрос «как» сконструировать нейронную сеть, более быструю и адекватную задачам, а социально-этические и культурологические подходы сосредоточены на вопросах причины и цели, «зачем» [9]. Вертикальные и горизонтальные информационные потоки и каналы

коммуникации в матрицах фундаментального горизонта планирования [10], включение ценностных и культурных измерений помогут предотвратить конфликты и потери, связанные с итеративной дискриминацией при дальнейшем внедрении нейронных сетей. [11]

IV. ЗАКЛЮЧЕНИЕ

Предложенная классификация этических проблем помогает выделить несколько групп решений в построении машинного обучения и реализации нейротехнологий в социальных отношениях. Эти три категории подходов к выработке этических решений не исчерпывают всех аспектов, но они могут улучшить дизайн нейронных сетей, а также вопросы обучения людей и кибербезопасности. Этические вопросы изучают философия, социология, психология, менеджмент, социальные науки, но цифровые технологии традиционно воспринимаются как дело инженеров, человеческий гений повторяет нейронный процесс человеческого мозга и развивает возможности существа справиться с человеческими проблемами, с разработкой новых решений для несовершенных нейронных сетей, чтобы сделать их все более и более похожими на человеческое познание.

СПИСОК ЛИТЕРАТУРЫ

- [1] Roundy P.T., Burke-Smalley L. Leveraging entrepreneurial ecosystems as human resource systems: A theory of meta-organizational human resource management // Human Resource Management Review. 2021. P. 100863. doi: 10.1016/J.HRMR.2021.100863.
- [2] Pokrovskaja N.N., Petrov M.A., Gridneva M.A. Diagnostics of Professional Competencies and Motivation of the Engineer in the Knowledge Economy // Proceedings of 3rd Int. Conference on Human Factors in Complex Technical Systems and Environments (ERGO). St Petersburg: LETI, 2018. P. 28-31. doi: 10.1109/ERGO.2018.8443851.
- [3] Zhang M. Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software // Forbes. 1 July 2015.
- [4] Regan J. New Zealand passport robot tells applicant of Asian descent to open eyes // Reuters. 7 December 2016.
- [5] Mac R. Facebook Apologizes After A.I. Puts 'Primates' Label on Video of Black Men // The New York Times. 4 September 2021.
- [6] Arrow K.J. A difficulty in the Concept of Social Welfare // Journal of Political Economy. 1950. Vol. 58. Pp. 328-346.
- [7] Arrow K.J. Social Choice and Individual Values. New Haven and London: Yale University Press, 1951.
- [8] Покровская Н.Н., Шангутов А.О., Шангутов О.А., Гарин А.К. Цифровые технологии государственных функций и сетевые модели коммуникации социального предпринимательства // Телескоп: журнал социологических и маркетинговых исследований. 2022. № 1 (5). С. 32-41.
- [9] Слободской А.Л., Теребкова Т.А., Гарин А.К. Цифровое образование и новое технологическое поколение: спрос на новый контент в обучении // Наука о данных. Материалы международной научно-практической конференции. СПб.: СПбГЭУ, 2020. С. 287-289.
- [10] Гарин А.К., Покровская Н.Н. Партиципативное конструирование коммуникационного пространства: инди-игра Minecraft как реализация концепции распределенных сетей // Технологии PR и рекламы в современном обществе. Материалы XIII Всероссийской научно-практической конференции. Под ред. И.П. Тростинской. 2018. С. 104-108.
- [11] Waelen R.A. The struggle for recognition in the age of facial recognition technology // AI Ethics. 8.03.2022. doi:10.1007/s43681-022-00146-8.

Нейросетевое моделирование динамики показателей научно-технической безопасности регионов России

Е. С. Митяков¹ С. В. Артемова²,
А. И. Ладынин³, А. Г. Шмелева⁴

МИРЭА – Российский технологический университет

¹iyao@mail.ru, ²sartemova@mail.ru,

³andrey.ladynin@hotmail.com, ⁴ann_shmeleva@mail.ru,

М. А. Каменская

Тамбовский государственный технический
университет

art_mari@bk.ru

С. Н. Митяков

Нижегородский государственный технический
университет им. Р.Е. Алексеева

snmit@mail.ru

Аннотация. В статье проведено ранжирование показателей научно-технической безопасности регионов России. Решение сложно формализуемой задачи анализа региональной научно-технической безопасности выполнено с использованием математического аппарата искусственных нейронных сетей на базе данных Федеральной службы государственной статистики РФ. Для проведения расчетов задействованы самоорганизующейся нейронной сети Кохонена, что позволило провести кластеризацию данных, результатом которой является ранжирование субъектов Федерации. Проведенный анализ позволил выявить значительную дифференциацию регионов по рассматриваемым показателям научно-технической безопасности.

Ключевые слова: нейросетевое моделирование, самоорганизующиеся карты Кохонена, кластерный анализ, научно-техническая безопасность, регионы Российской Федерации

I. ВВЕДЕНИЕ

Современные реалии диктуют использование актуальных технологических возможностей для оценки параметров социально-экономического развития систем различных иерархических уровней. К важнейшей экономической системе сегодня следует отнести систему научно-технической безопасности (НТБ), которая выступает одной из важнейших характеристик национальной безопасности государства и тесно связана с экономической безопасностью. Данный факт требует детального анализа параметров НТБ на различных иерархических уровнях народного хозяйства с целью выработки научно-обоснованных решений по организации эффективного управления в данной сфере.

Содержание научно-технологической безопасности можно охарактеризовать существенной сложностью, многоуровневостью и многоаспектностью [1]. Сущность НТБ определяется воздействием субъективных и объективных факторов, а также условиями и сферами проявления негативных экзогенных и эндогенных тенденций. Кроме этого, не прекращающийся процесс институциональных преобразований в России, корректировка центров принятия решения в сфере НТБ,

регионализация экономики диктует целесообразность проведения комплексных исследований не только на макро-, но и на мезоуровне [2]. Важность исследований параметров научно-технологической безопасности в региональном разрезе определяется тем, что развитие страны в целом возможно лишь при устойчивом и гармоничном развитии всех ее регионов.

В данном исследовании приведена одна из возможных методик анализа региональной научно-технической безопасности с использованием математического аппарата искусственных нейронных сетей. Для проведения расчетов задействованы самоорганизующейся нейронной сети Кохонена.

II. МЕТОДЫ ИССЛЕДОВАНИЯ

В настоящее время доступность значительного количества данных, наличие достаточного количества статистических наблюдений функционирования разнообразных экономических систем открывает значительный спектр инструментария для эмпирического анализа. Наряду с классическими экономико-статистическими методами, в последнее время при исследовании социально-экономических процессов особое внимание уделено современным технологиям нейросетевого моделирования.

К классической литературе по машинному обучению выделяют два принципиально разных подхода к организации нейронных сетей: сети, обучаемые с учителем и без учителя [3]. Первые – представляют средство для поиска из заданного набора данных информации о взаимосвязях между выходами и входами нейронной сети (входной информацией и результатами), которые могут быть формализованы для принятия решений. Вторые – предназначены для решения задачи классификации, организации и визуального представления больших объемов данных [4].

Одним из примеров класса нейронных сетей, обучаемых без учителя, выступают самоорганизующиеся карты Кохонена (СОК). Они предназначены для автоматизации методов визуализации и поиска характерных закономерностей и зависимостей в больших

объемах данных [5]. При использовании СОК основной акцент направлен на извлечения новых знаний из имеющейся информации и выявление новых паттернов и структур в исследуемых массивах. Ключевой особенностью СОК является отсутствие необходимости в априорных предположениях об исходном распределении данных. Предложенный инструментарий является эффективным средством для анализа в условиях динамично меняющихся данных.

Самоорганизующиеся карты содержат узлы (нейроны), количество которых задается исследователем перед началом анализа. При этом каждый нейрон характеризуется парой векторов: вектором весов и вектором координат узла на карте. Визуально карты Кохонена представляют из себя множество ячеек шестиугольной или прямоугольной формы.

Перед началом анализа задана размерность входных параметров, по которой строится начальная карта. Далее, в процессе обучения нейронной сети, векторы веса узлов приближаются к входным данным. Для каждой анализируемой точки подбирается максимально схожий по значению вектора веса узел, и значение его вектора веса приближается к наблюдению. Кроме этого, к наблюдению также приближаются векторы веса нескольких точек наблюдения. Следовательно, если во множестве исходной информации две похожих точки наблюдения, на карте Кохонена им будут соответствовать узлы, располагающиеся по соседству. Итеративный процесс обучения сети, который перебирает исходные входные данные, завершается по достижению надлежащей погрешности, которая задается перед началом анализа, или после завершения заданного числа итераций цикла обучения. В результате работы обучающего алгоритма, на карте получаются изображения классифицированных данных, с близкими признаками в соседних ячейках, раскрашенные в зависимости от анализируемых параметров нейронов.

Таким образом, использование карт Кохонена позволяет решать задачи кластеризации входных данных и визуального изображения многомерных информационных массивов в двумерной плоскости [6]. Блок схема алгоритма обучения самоорганизующихся карт Кохонена представлена на рис. 1.



Рис. 1. Блок-схема алгоритма Кохонена

Оценка результатов кластеризации, полученных с использованием карт Кохонена, как правило, проводится по трем ключевым критериям: количество кластеров, качество кластеризации и устойчивость результатов кластеризации [4]. Оптимальное число кластеров зависит от исходной информации и целей проводимого исследования.

III. СИСТЕМА ПОКАЗАТЕЛЕЙ НАУЧНО-ТЕХНИЧЕСКОЙ БЕЗОПАСНОСТИ РЕГИОНОВ

В рамках данной работы предлагается задействовать индикаторы, состав которых обоснован в работе по исследованию экономической безопасности регионов [7]. В состав исследуемых показателей предлагается включить следующие:

- число лиц, занятых НИР, на 10000 занятого населения (K_1);
- внутренние затраты на научные исследования и разработки, % к ВРП (K_2);
- интенсивность затрат на технологические инновации, % (K_3);
- доля отгруженной инновационной продукции во всей отгруженной продукции промышленности, % (K_4).

Показатель K_1 характеризует кадровый потенциал начальной стадии научно-технической деятельности в регионе. Данный индикатор особенно актуален, поскольку в последнее время наблюдается значительное сокращение научных кадров в стране. Кроме того, можно отметить усиление разрыва между различными поколениями исследователей. По показателю K_1 Россия отстает от множества развитых стран.

В последние годы происходит сокращение воспроизводства научных кадров, нарастает разрыв между разными поколениями исследователей. В настоящее время численность исследователей составляет чуть более 45 % от уровня 1991 г. По этому показателю Россия уступает большинству развитых стран, включая США, Японию, Францию, Финляндию и др. Возникла реальная опасность утраты лидирующего положения российских научных школ.

Вторым индикатором в проекции выступает показатель внутренних затрат на научные исследования и разработки. Данный индикатор отражает уровень обеспечения финансами начальной стадии инновационного процесса в регионах. Основным источником финансирования затрат на научные исследования и разработки выступает государство (финансирование из федерального и региональных бюджетов). Среди других финансовых источников – средства предпринимательского сектора, иностранных инвесторов и др. Для сравнительного анализа регионов используется индикатор «Внутренние затраты на научные исследования и разработки, % к ВРП»

Следующий показатель проекции – индикатор интенсивности затрат на технологические инновации. Подобного рода затраты представляют собой выраженные в денежной форме фактические расходы, которые связаны с осуществлением различного рода инновационной деятельности, выполняемой в масштабе объекта народного хозяйства. В их состав включены затраты на научные исследования и разработки, которые напрямую связаны с выведением на рынок новой продукции, методов и услуг передачи и производства; затраты на покупку программных продуктов, нацеленных на внедрение и реализацию инноваций; затраты на проектирование производства; затраты на приобретение оборудования и машин, которые связаны с технологическими инновациями, и др. В большинстве ведущих стран Европы расходы на исследования и разработки существенно превышают расходы на приобретение машин и оборудования, что обеспечивает высокий уровень конкурентоспособности инновационной продукции. В России, к сожалению, пока наблюдается обратная ситуация. При этом основным источником финансирования таких затрат выступают собственные средства организаций.

Для сравнительного анализа экономических систем используется индикатор «Интенсивность затрат на технологические инновации, %», равный отношению затрат на технологические инновации к объему выпущенной продукции. В качестве порогового значения этого индикатора принято значение 3,2 %, которое примерно соответствует среднему значению индикатора в развитых странах. Замыкает набор показателей индикатор доли отгруженной инновационной продукции во всей отгруженной продукции промышленности, %. Он является базовым индикатором, характеризующим результативность инновационной деятельности региона, и входит в состав индикаторов экономической безопасности страны. В настоящее время данный показатель в среднем по России не превышает 10 %. Это весьма низкая цифра, свидетельствующая о недостаточном уровне инновационного развития страны.

IV. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Разбивка объектов на кластеры должна удовлетворять трем требованиям: – каждый объект входит только в одну из групп; – объекты внутри одного кластера близки друг другу по своим характеристикам; – объекты из разных групп заметно различаются. При проведении кластеризации субъектов Федерации, разбивка объектов на кластеры должна удовлетворять следующим требованиям [8,9]: каждый регион входит только в один кластер из групп; субъекты Федерации внутри одного кластера близки по показателям НТБ; регионы из разных групп заметно различаются по показателям НТБ.

Отметим, что поскольку априорное количество кластеров заранее неизвестно, объективная оценка качества полученного решения может быть осуществлена с использованием индекса силуэта [10]. На рис. 2, 3 показаны результаты кластерного анализа методом самоорганизующихся карт Кохонена для регионов России по индикаторам их научно-технической безопасности за 2017, 2019 гг.

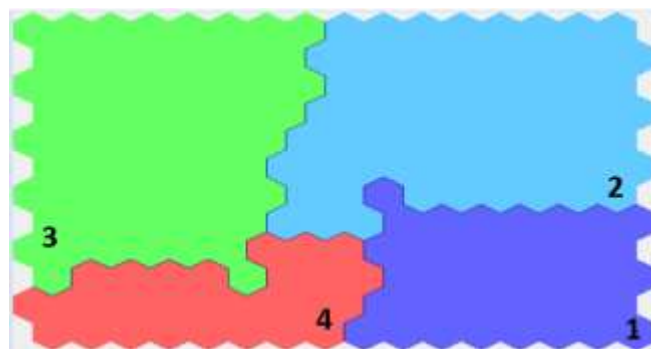


Рис. 2. Результаты кластерного анализа в 2017 г.

На рисунках отсутствуют перекрытия кластеров, что может служить подтверждением обоснованность полученных результатов. В 2017 году расчеты показывают наличие четырех кластеров, а в 2019 – уже семи.

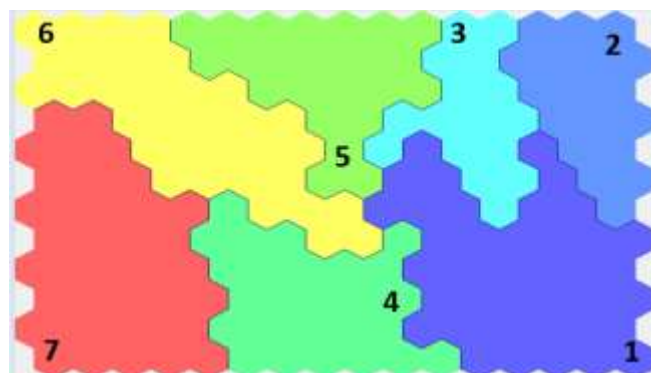


Рис. 3. Результаты кластерного анализа в 2019 г.

В табл. 1, 2 приведено распределение регионов по кластерам, характерные 2017 и 2019 годам (наименьшее и наибольшее количество кластеров соответственно).

ТАБЛИЦА I СОСТАВ КЛАСТЕРОВ ЗА 2017 Г.

№	Состав кластеров
1	Воронежская область, Калужская область, Московская область, г. Москва, г. Санкт-Петербург, Пермский край, Нижегородская область, Пензенская область, Ульяновская область, Свердловская область, Новосибирская область, Томская область
2	Белгородская область, Липецкая область, Рязанская область, Тамбовская область, Тверская область, Тульская область, Ленинградская область, Краснодарский край, Ростовская область, Ставропольский край, Республика Башкортостан, Республика Мордовия, Республика Татарстан, Чувашская Республика, Кировская область, Оренбургская область, Самарская область, Саратовская область, Тюменская область, Красноярский край, Иркутская область, Омская область, Хабаровский край, Амурская область, Сахалинская область
3	Брянская область, Ивановская область, Костромская область, Орловская область, Смоленская область, Республика Карелия, Республика Коми, Архангельская область, Вологодская область, Калининградская область, Псковская область, Республика Адыгея, Республика Калмыкия, Республика Крым, Астраханская область, Волгоградская область, Республика Дагестан, Республика Ингушетия, Кабардино-Балкарская Республика, Карачаево-Черкесская Республика, Республика Северная Осетия-Алания, Чеченская республика, Республика Марий Эл, Удмуртская Республика, Курганская область, Республика Алтай, Республика Бурятия, Республика Тыва, Республика Хакасия, Алтайский край, Забайкальский край, Кемеровская область, Республика Саха (Якутия), Еврейская автономная область, Чукотский автономный округ
4	Владимирская область, Курская область, Ярославская область, Мурманская область, Новгородская область, г. Севастополь, Челябинская область, Камчатский край, Приморский край, Магаданская область

ТАБЛИЦА II СОСТАВ КЛАСТЕРОВ ЗА 2019 Г

№	Состав кластеров
1	Владимирская область, Воронежская область, Калужская область, Ярославская область, Ленинградская область, Пермский край, Пензенская область, Свердловская область, Челябинская область, Иркутская область, Новосибирская область
2	Московская область, г. Москва, г. Санкт-Петербург, Нижегородская область, Ульяновская область, Алтайский край
3	Тульская область, Ростовская область, Республика Татарстан, Самарская область, Забайкальский край
4	Курская область, Рязанская область, Тверская область, Мурманская область, Новгородская область, г. Севастополь, Саратовская область, Кемеровская область, Томская область, Камчатский край, Приморский край, Магаданская область
5	Белгородская область, Липецкая область, Тамбовская область, Республика Мордовия, Чувашская Республика, Республика Хакасия, Хабаровский край, Сахалинская область
6	Смоленская область, Республика Карелия, Республика Коми, Краснодарский край, Республика Башкортостан, Кировская область, Оренбургская область, Республика Алтай, Республика Бурятия, Амурская область, Еврейская автономная область
7	Брянская область, Ивановская область, Костромская область, Орловская область, Архангельская область, Вологодская область, Калининградская область, Псковская область, Республика Адыгея, Республика Калмыкия, Республика Крым, Астраханская область, Волгоградская область, Республика Дагестан, Республика Ингушетия, Кабардино-Балкарская Республика, Карачаево-Черкесская Республика, Республика Северная Осетия-Алания, Чеченская республика, Ставропольский край, Республика Марий Эл, Удмуртская Республика, Курганская область, Тюменская область, Республика Тыва, Красноярский край, Омская область, Республика Саха (Якутия), Чукотский автономный округ

Анализ рисунков и таблиц показал, что со временем перманентно меняется не только количество, но и состав кластеров. Это свидетельствует о неоднородной структуре регионов России по показателям НТБ и значительной дифференциации субъектов Федерации по совокупности рассматриваемых индикаторов.

V. Выводы

Специфика имеющихся данных об индикаторах НТБ регионов РФ обуславливает необходимость разработки новаторских методов выявления новых структур в массивах данных и усовершенствования способов обобщения и визуального представления больших объемов информации. При этом интеграция нового инструментария должна носить более фундаментальный характер и создавать базис для развития методологии экономической науки.

Таким образом, предложенный в работе метод кластерного анализа многомерных статистических данных на базе нейросетевого моделирования с использованием самоорганизующихся карт Кохонена является одним из эффективных методов исследования, свободных от модельных ограничений и позволяющих решать задачу кластеризации без априорных требований к количеству кластеров в начале исследования.

Проведенный анализ показал, что со временем перманентно меняется не только количество, но и состав кластеров, что свидетельствует о крайне неоднородной структуре регионов России по показателям НТБ и значительной дифференциации субъектов Федерации по совокупности рассматриваемых индикаторов.

СПИСОК ЛИТЕРАТУРЫ

- [1] Морозова Т.В. Совершенствование методики оценки и мониторинга научно-технологической безопасности на региональном уровне: дис. ... канд. экон. наук.: 08.00.05/ Морозова Татьяна Валерьевна. М., 2006. 242 с.
- [2] Мандыч И.А., Быкова А.В. Трудности и перспективы развития высокотехнологичных проектов в эпоху цифровой трансформации экономики. *Russian Technological Journal*. 2021;9(2):88-95. <https://doi.org/10.32362/2500-316X-2021-9-2-88-95>
- [3] Галушкин А.И., Цыпкин Я.З. Нейронные сети: история развития теории: Учебное пособие для вузов. М.: Альянс, 2015. 840 с.
- [4] Лочмеле Р.Р. Современные количественные методы экономического анализа: самоорганизующиеся карты Кохонена (СОК). Государственное управление. Электронный вестник. 2003. №1. URL: <https://cyberleninka.ru/article/n/sovremennye-kolichestvennye-metody-ekonomicheskogo-analiza-samoorganizuyuschiesya-karty-kohonena-sok> (дата обращения: 26.02.2022)
- [5] Kohonen T. Self-Organizing Maps (Third Extended Edition), New York, 2001, 501 pages. ISBN 3-540-67921-9
- [6] Дебок Г., Кохонен Т. Анализ финансовых данных с помощью самоорганизующихся карт / Пер. с англ. М. 2001.
- [7] Митяков С.Н., Лапаев Д.Н., Митяков Е.С. [и др.] Экономическая безопасность регионов России: монография / 3-е изд., перераб. и дополн. Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2019. 299 с.
- [8] Перова В.И., Ласточкина Е.И. Нейросетевое моделирование динамики инновационной деятельности в регионах Российской Федерации // Вестник Нижегородского университета им. Н.И. Лобачевского. Серия: Социальные науки. 2015. № 3(39). С. 49-58.
- [9] Евстифеев А.А. Системный анализ поддержки принятия решений при развитии сети газовых заправочных станций // Вестник Тамбовского государственного технического университета. 2021. Т. 27. № 2. С. 203-211. DOI 10.17277/vestnik.2021.02.pp.203-211.
- [10] Kraufman, L., Rousseeuw, P. (2005). Finding groups in Data. An Introduction to Cluster Analysis. Wiley. DOI: <http://dx.doi.org/10.1002/9780470316801.ch1>

Применение нейронных сетей в датчиках оценки влажности материала

С. В. Артемова¹, А. И. Ладынин²,
Е. С. Митяков³, Ву Чи Чиен⁴

МИРЭА – Российский технологический университет

¹sartemova@mail.ru, ²andrey.ladynin@hotmail.com,

³ann_shmeleva@mail.ru, ⁴vturichien00@gmail.com

М. А. Каменская

Тамбовский государственный университет

art_mari@bk.ru

Д. Н. Лапаев

Нижегородский государственный технический университет им. Р.Е. Алексеева

dnlapaev@mail.ru

Аннотация. В статье рассматривается методика создания датчиков оценки влажности материала, пригодных для использования в информационно-управляющих системах сушки как сыпучих, так и пастообразных материалов. Приводятся примеры разработанных датчиков для сушильных установок вальце-ленточного и барабанного типа. Обучение нейронных сетей таких датчиков организуют по образцовым мерам технологических параметров в адаптивном диапазоне в сушильных установках с фиксированной точностью. На входы обученной нейронной сети подаются нормализованные измеренные значения технологических параметров, оказывающих влияние на процесс сушки. С выхода нейронной сети получают нормализованную оценку влажности материала. На основании полученной информации от датчика о текущей влажности внутри сушильной установки информационно-управляющей системой принимается решение об управлении процессом сушки с целью минимизации заданных критериев. Применение подобных систем в промышленности позволяет увеличить выпуск качественного материала без снижения производительности процесса сушки.

Ключевые слова: оперативная оценка влажности материала, состояния функционирования, нейронные сети, фреймовая модель, адаптивный датчик влажности

I. ВВЕДЕНИЕ

Сушка широко распространенный процесс в различных отраслях промышленности и сельского хозяйства. Управление процессом сушки с использованием информационно-управляющей системы способной вырабатывать управляющие воздействия минимизирующие потери качества выпускаемого материала и производительности его производства является важной задачей. Современные исследования в области управления высокотехнологичным производством требуют реализации цифровых методов обработки и анализа информации [1, 2]. Для оперативного мониторинга параметров процесса сушки, оказывающих существенное влияние на синтез управляющего воздействия, предлагается использовать датчики влажности, позволяющие в реальном режиме времени оценивать влажность материала в процессе его сушки внутри сушильных установок.

В процессе сушки в большинстве случаев существенное влияние на качество выпускаемой продукции оказывает информация о текущей влажности материала. Оперативный мониторинг параметров сушки в реальном режиме времени позволяет информационно-управляющей системе вырабатывать управляющее воздействие, минимизирующее потери качества и производительности при выпуске высушиваемого материала.

На современном рынке присутствует множество датчиков влажности материала, позволяющих с минимальной погрешностью определять влажность материала в процессе сушки, однако они в силу ряда причин не могут быть использованы для решения поставленной задачи. Так, например, применение гравитационных датчиков затрудняет осуществлять управление процессом сушки в реальном режиме времени. Применение СВЧ датчиков становится невозможным из-за конструктивных особенностей в частности наличие металлических частей сушильных установок. Применение инфракрасных влагомеров затруднено в связи с запыленной средой внутри зон сушки. Анализаторы влажности радарного типа применяются для измерения влажности сыпучих материалов, а не пастообразных [3]. Поэтому создание универсальных датчиков, в основе работы которых лежат нейронные сети, позволяющие производить оперативную оценку влажности материала с приемлемой точностью для пастообразных и сыпучих материалов в процессе их сушки, является актуальной задачей.

II. ПОСТАНОВКА ЗАДАЧИ

Рассмотрим методику создания датчиков оценки влажности материала пригодных для использования в информационно-управляющих системах сушки как сыпучих, так и пастообразных материалов. Косвенное измерение влажности осуществляется по аналитической модели, представляющей собой нейронную сеть. Методика содержит следующие этапы.

1. Выявление технологических параметров, оказывающих существенное влияние на процесс сушки в исследуемом типе сушильных установок.

2. Определение по образцовым мерам выявленных технологических параметров в адаптивном диапазоне в сушильных установках с фиксированной точностью.

3. Создание статистической выборки выделенных технологических параметров на основании их мониторинга в процессе сушки. Объем выборки должен быть достаточным для обучения нейронной сети.

3. Анализ и обработка полученной статистической выборки.

4. Выбор способа нормировки исходных данных и нормировка данных.

5. Выделение их множества данных следующих выборок обучающей, валидации и тестовой. При этом помещать в обучающую выборку примерно равного количества самые разнообразные данные.

6. Определение:

- входов и выходов нейронной сети;
- метода обучения нейронной сети с учителем или без учителя;
- типа и архитектуры нейронной сети;
- функций активации нейронов и критериев качества нейронной сети;
- методов оптимизации определения весов нейронов;
- критериев останова обучения.

7. Проведение обучения нейронной сети.

8. Выбор микроконтроллера, на базе которого реализуется датчик оценки влажности.

9. Построение информационно-измерительной системы оперативной оценки влажности лежащей в основе функционирования датчика.

Применим описанную методику для разработки датчиков оценки двух сушильных установок вальце-ленточного и барабанного типа.

Сушка влажных материалов представляет собой сложный процесс переноса тепла и массы. Этот процесс сопровождается структурно-механическими, а иногда и химическими изменениями материала. Он может осуществляться различными методами в конструктивно отличающихся сушильных установках. Методы сушки различаются по способу подвода тепла к материалу и по гидродинамическим условиям в сушильной камере, а также по способам обеспечения требуемой формы конечного продукта (порошок, гранулы, твердое вещество). Поэтому выявление технологических параметров, оказывающих существенное влияние на процесс сушки в исследуемом типе сушильных установок, является начальным этапом предлагаемой методики создания рассматриваемых датчиков [4].

III. МОДЕЛЬ НЕЙРОННОЙ СЕТИ

Входами сети являются следующие значения:

- x_1 – начальная влажность барды на входе сушильной установки, % от сухого вещества;
- x_2 – температура барды на выходе сушилки;
- x_3 – давление теплоносителя, атм.;
- x_4 – температура воздуха в сушильной установке;

- x_5 – мощность вытяжных вентиляторов;
- x_6 – содержание кислорода в воздухе сушильной установки, %;
- x_7 – температура выпара;
- x_8 – нагрузка электродвигателя сушильной установки, %.

В вальце-ленточной сушильной установке основными технологическими параметрами оказывающими влияние на процесс сушки являются:

- x_1 – начальная влажность материала;
- x_2, x_3 – температура и влажность окружающей среды;
- x_4 – скорость движения ленты продуктового транспортера;
- x_5 – наличие или отсутствие ворошителя;
- x_6 – процент времени работы вытяжного вентилятора;
- x_7, x_8 – температура и влажность сушильного агента;
- x_9, x_{10} – процент открытия воздухозаборного окна и сбросногошибера.

Статистическая выборка формировалась на основании данных измеряемых несколько лет непрерывной работы вальце-ленточной сушильной установки.

При проектировании датчика влажности большое внимание следует уделять минимизации погрешностей производимого им косвенного измерения, в том числе снижению погрешности используемой аналитической модели [5, 6]. С этой целью при обучении нейронной сети, которая и является аналитической моделью датчика, ее входные данные требуется определять по образцовым мерам измеренных технологических параметров в адаптивном диапазоне с фиксированной точностью.

Важным при создании аналитической модели является этап анализ и обработки измеренных статистических данных для получения обучающей выборки. Если в процессе реальной эксплуатации будут встречаться измеренные значения, сильно отличающиеся от обучающей выборки, то погрешность аналитической модели будет увеличиваться. С целью минимизации погрешности в процессе функционирования датчика обучающая выборка должна быть максимального объема и содержать разнообразные данные, встречающиеся в процессе сушки с равной частотой. Однако различные выбросы данных, встречающиеся в статистических данных, следует определить и не использовать в обучающей выборке.

После анализа и обработки данных их нормализация осуществлялась по формуле:

$$\overline{x_i} = x_i \cdot K_{\text{амп}}^n + K_{\text{оф}}^n, \quad i = \overline{1, V_n}, \quad (1)$$

где i – номер параметра технологического процесса сушки; $\overline{x_i}, x_i$ – нормированное и ненормированное значения i -го параметра соответственно; $K_{\text{амп}}^n, K_{\text{оф}}^n$ –

амплитудный и офсетный коэффициенты нормализации;
 V_n – число параметров для расчета влажности φ_n .

Аналитическая модель оценки влажности материала в n камере вальце-ленточной сушильной установки имеет архитектуру многослойного персептрона. Она обучена по алгоритму обратного распространения ошибки и имеет следующий вид:

$$\varphi_n = \left[1 + \exp \left(-\beta_{\text{вых}}^{(n)} \cdot \left[\sum_{j=1}^{N_n} w_j^{(n \text{ вых})} \cdot \left(1 + \exp \left(-\beta_j^{(n)} \cdot \left[w_{1,j}^{(n)} \cdot \bar{x}_1 + w_{2,j}^{(n)} \cdot \bar{x}_2 + w_{3,j}^{(n)} \cdot \bar{x}_3 + w_{4,j}^{(n)} \cdot \bar{x}_4 + w_{5,j}^{(n)} \cdot \bar{x}_5 + w_{6,j}^{(n)} \cdot \bar{x}_6 + w_{7,j}^{(n)} \cdot \bar{x}_7 + w_{8,j}^{(n)} \cdot \bar{x}_8 + w_{9,j}^{(n)} \cdot \bar{x}_9 + w_{10,j}^{(n)} \cdot \bar{x}_{10} \right] - \theta_j^{(n)} \right) \right]^{-1} - \theta_{\text{вых}}^{(n)} \right] \right]^{-1} \right], \quad (2)$$

где n – номер камеры, $\beta_j^{(n)}, \beta_{\text{вых}}^{(n)}, \theta_j^{(n)}, \theta_{\text{вых}}^{(n)}$ – углы наклона и сдвиги сигмоидальных активационных функций нейронов скрытого и выходного слоев нейронной сети; $w_{i,j}^{(n)}, w_j^{(n \text{ вых})}$ – весовые коэффициенты связей нейронов скрытого и выходного слоев нейронной сети; N_n – число нейронов в скрытом слое сети для расчета влажности материала в n камере сушильной установки; нормализованные значения: φ_n – нормализованная влажность материала.

Денормализация выходного значения нейронной сети позволяет получить оценку значения влажности материала φ_n

$$\varphi_n = \frac{\bar{\varphi}_2 - K_{\text{оф}}^{\text{дн}}}{K_{\text{амп}}^{\text{дн}}}, \quad (3)$$

где $K_{\text{амп}}^{\text{дн}}, K_{\text{оф}}^{\text{дн}}$ – амплитудный и офсетный коэффициенты денормализации.

Аналитическая модель (2) оценивает значение влажности материала в реальном времени в процессе его сушки в вальце-ленточной сушильной установке с относительной погрешностью, не превышающей 2 % [2]. Результаты расчета по модели приведены на рис. 1.

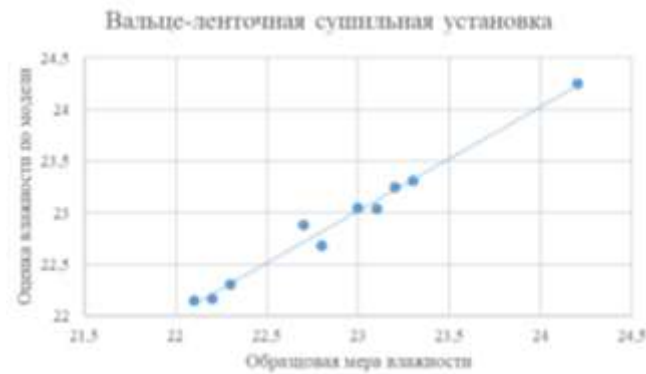


Рис. 1. Результаты вычислений для вальце-ленточной сушильной установки

В барабанной сушильной установке основными технологическими параметрами, оказывающими влияние на процесс сушки являются: x_1 – начальная влажность

барды на входе сушильной установки, % от сухого вещества; x_2 – температура барды на выходе сушилки, °C; x_3 – давление теплоносителя, атм.; x_4 – температура воздуха в сушильной установке, °C; x_5 – мощность вытяжных вентиляторов, %; x_6 – содержание кислорода в воздухе сушильной установки, %; x_7 – температура выпара, °C; x_8 – нагрузка электродвигателя сушильной установки, %.

Статистическая выборка также, как и вальце-ленточной сушильной установке формировалась на основании данных измеряемых несколько лет непрерывной работы барабанной сушильной установки.

Здесь нормализация параметров технологического процесса проводилась согласно следующему выражению:

$$x_{iN} = \frac{x_i - \mu_i}{\sigma_i} \quad (4)$$

где x_{iN} – нормированное значение i -ого технологического параметра, x_i – значение i -го технологического параметра, μ_i – математическое ожидание i -ого технологического параметра, σ_i – дисперсия i -ого технологического параметра.

Обучение нейронной сети организовано по образцовым мерам технологических параметров в адаптивном диапазоне сушильной установки с фиксированной точностью.

Аналитическая модель оценки влажности материала в барабанной сушильной установке представляет собой нейронную сеть, которая имеет архитектуру многослойного персептрона, обученного с помощью алгоритма обратного распространения ошибок и имеет следующий вид:

$$\varphi = 100 - f_4 \left(\sum_{h=1}^{m_3} f_3 \left(\sum_{k=1}^{m_2} f_2 \left(\sum_{j=1}^{m_1} f_1 \left(\sum_{i=1}^n x_{iN} \mathbf{W}_{i,j}^{(1)} + b_j \right) \mathbf{W}_{j,k}^{(2)} + b_k \right) \mathbf{W}_{k,h}^{(3)} + b_h \right) \mathbf{W}_h + b^{\text{вых}} \right), \quad (5)$$

где φ – оценка влажности материала; x_{iN} – нормированное значение i -той входной переменной; f_1, f_2, f_3 – активационные функции ReLu – функция.

$$f_1 = \max(0, \sum_{i=1}^n x_i \mathbf{W}_{i,j}^{(1)} + b_j), \quad (6)$$

$$f_2 = \max(0, \sum_{j=1}^{m_1} \max(0, \sum_{i=1}^n x_i \mathbf{W}_{i,j}^{(1)} + b_j) \mathbf{W}_{j,k}^{(2)} + b_k), \quad (7)$$

$$f_3 = \max(0, \sum_{k=1}^{m_2} \max(0, \sum_{j=1}^{m_1} \max(0, \sum_{i=1}^n x_i \mathbf{W}_{i,j}^{(1)} + b_j) \mathbf{W}_{j,k}^{(2)} + b_k) \mathbf{W}_{k,h}^{(3)} + b_h) \quad (8)$$

$$f_4 = \sum_{h=1}^{m_3} \max(0, \sum_{k=1}^{m_2} \max(0, \sum_{j=1}^{m_1} \max(0, \sum_{i=1}^n x_i \mathbf{W}_{i,j}^{(1)} + b_j) \mathbf{W}_{j,k}^{(2)} + b_k) \mathbf{W}_{k,h}^{(3)} + b_h) \mathbf{W}_h + b^{\text{вых}} \quad (9)$$

$$f_4 = \sum_{h=1}^{m_3} \max(0, \sum_{k=1}^{m_2} \max(0, \sum_{j=1}^{m_1} \max(0, \sum_{i=1}^n x_i \mathbf{W}_{i,j}^{(1)} + b_j) \mathbf{W}_{j,k}^{(2)} + b_k) \mathbf{W}_{k,h}^{(3)} + b_h) \mathbf{W}_h + b^{\text{вых}}, \quad (10)$$

n – количество входных переменных (в нашем случае $n = 8$), $(i=1 \dots n)$.

m_1 – количество нейронов в 1-ом скрытом слое (в нашем случае $m_1 = 36$) ($j=1 \dots m_1=36$).

m_2 – количество нейронов в 2-ом скрытом слое (в нашем случае $m_2 = 36$) ($k=1 \dots m_2 = 36$).

m_3 – количество нейронов в 3-ем скрытом слое (в нашем случае $m_3 = 36$) ($h=1 \dots m_3 = 36$).

x_i – значение i -той входной переменной.

$W^{(1)}_{i,j}$, $W^{(2)}_{j,k}$, $W^{(3)}_{k,h}$, $W^{(b)}_k$ – скрытые и выходные весовые коэффициенты, $b^{(b)}_j$, b_k , b_h , $b^{(b)}$ – входные и выходные смещения.

Аналитическая модель (5) дает возможность оценивать влажность материала в реальном масштабе времени в сушильной установке барабанного типа в процессе его сушки с относительной погрешностью, менее 2 % [3]. Результаты расчета по модели приведены на рис. 2.

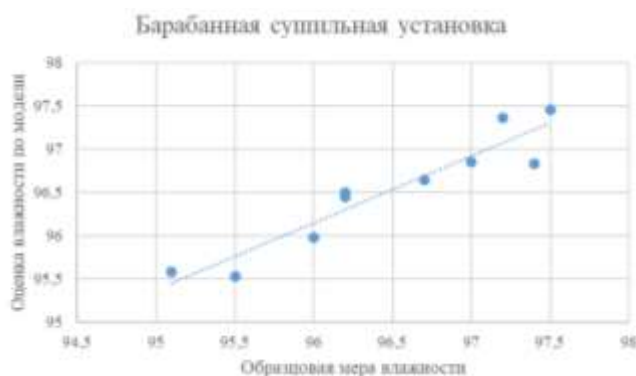


Рис. 2. Результаты вычислений для барабанной сушильной установке

Для аппаратной реализации датчика влажности подойдет большинство недорогих современных микроконтроллеров, имеющих в наличии требуемое количество портов ввода/вывода данных и возможность расширения памяти для организации баз данных и знаний в которых хранится информация, обеспечивающая расчеты по аналитической модели оценки влажности материала.

Функциональная схема информационно-измерительной системы оценки влажности материала представлена на рис. 3. Она включает в свой состав два основных программных модуля: нейросетевой блок, позволяющий оценить влажность на выходе сушильной установки по аналитическим моделям (2) и (5); базу данных, которая содержит исчерпывающий объем априорной и апостериорной информации, необходимой для реализации процедур анализа. Сигналы, получающие от датчиков, нормализуются и поступают на входы обученной нейросетевой модели, с выхода которой в реальном масштабе времени производится оценка текущего значения влажности материала в сушильной установке. Эта информация передается для отображения и дальнейшего использования в информационно-управляющей системе.

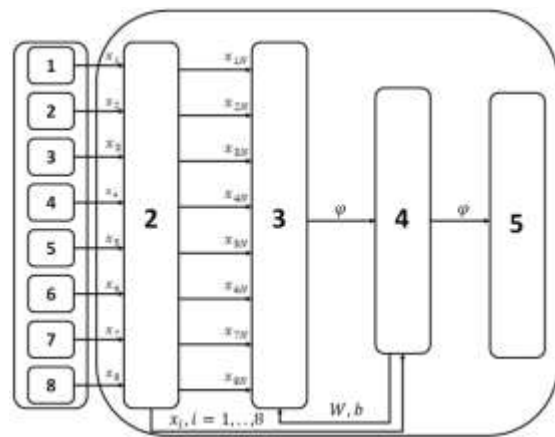


Рис. 3. Модель информационной системы оценки влажности

1. Сушильная установка с технологическими датчиками.
2. Модули опроса датчиков, анализа, расчета и нормализации входных значений нейронной сети.
3. Нейросетевой модуль расчета влажности.
4. База данных.
5. Средства отображения графической и текстовой информации.
6. Система информационно-управляющая система производственным процессом.

Рассмотрим алгоритм работы информационно-измерительной системы, который включает следующие шаги:

Шаг 1. Опрос технологических датчиков сушильной установки.

Шаг 2. Анализ исходных данных включающий проверку условия вхождения измеренных значений в адаптивные диапазоны. Если условие не выполняется, то пользователю отправляется сообщение о нарушении процесса сушки.

Шаг 3. Нормализация значений входов нейронной сети.

Шаг 4. Оценка влажности по аналитическим моделям в нейросетевом модуле.

Шаг 5. Отображение измеренной и рассчитанной информации.

Шаг 5. Сохранение информации об измеренных и рассчитанных значениях в базе данных.

Шаг 6. Передача полученной оценки влажности в информационно-управляющую систему.

IV. ЗАКЛЮЧЕНИЕ

С применением приведенной в статье методики были созданы два датчика в основе процедуры косвенного измерения влажности материала которых лежат аналитические модели, представляющие собой нейронные сети. Относительная погрешность оценки влажности в обоих случаях не превышает 2 %. Первый датчик позволяет оценивать влажность пастообразного материала в процессе его сушки в вальце-ленточной сушильной установке, а второй оценивает влажность

гранулированного материала в барабанной сушильной установке. По созданной методике можно проектировать датчики оперативной оценки влажности независимо от конечной формы материала (твердой, пастообразной, порошковой и т. д.).

Разработанные датчики оценки влажности, представляют собой информационно-измерительную систему, передающую информацию в информационно-управляющую систему, синтезирующую в реальном режиме времени управляющее воздействие, минимизирующее потери качества и производительность. Внедрение информационно-управляющей системы на предприятиях ОАО «Пигмент» г. Тамбов и АО «Талвис» Новая Ляда позволило повысить выпуск качественного продукта без снижения производительности процесса сушки.

СПИСОК ЛИТЕРАТУРЫ

- [1] Shmeleva A.G., Ladynin A.I. Industrial Management Decision Support System: from Design to Software. Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow, 2019, pp. 1474-1477-369. Doi: 10.1109/EIConRus.2019.8657313
- [2] Мандыч И.А., Быкова А.В. Трудности и перспективы развития высокотехнологичных проектов в эпоху цифровой трансформации экономики. *Russian Technological Journal*. 2021;9(2):88-95. <https://doi.org/10.32362/2500-316X-2021-9-2-88-95>
- [3] Y.I. Eremenko, A.I. Glushchenko, A.V. Fomin and V.A. Petrov, "On comparison of effectiveness of neural tuner based adaptive control system and observer based controller to solve heating plant control problem," *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017, pp. 135-138, doi: 10.1109/ICICI.2017.8365323.
- [4] O A Rokachevskii, V V Sleptsov, A E Ablaeva, S A Kanaev and O V Moskalenko. Metrological assurance of the characteristics of electric drives at the design stage. *Journal of Physics: Conference Series*, 1889 (2021) 042057, ICMSIT-II 2021, p.3-p6. IOP Publishing, doi:10.1088/1742-6596/1889/4/042057
- [5] Kholopov V.A., Kashirskaya E.N., Shmeleva A.G., Kurnasov E.V. An intelligent monitoring system for execution of machine engineering processes. *Journal of Machinery Manufacture and Reliability*. 2019. T. 48. no. 5. P. 464-475.
- [6] Разработка программного комплекса системы управления сложными производственными процессами / В.Г. Матвейкин, Б.С. Дмитриевский, А.А. Терехова [и др.] // Вестник Тамбовского государственного технического университета. 2021. Т. 27. № 2. С. 168-184. DOI 10.17277/vestnik.2021.02.pp.168-184.

COVID скрининг, основанный на применении нейросетевой классификации спектров выдоха

А. В. Кугаевских

Новосибирский государственный университет

a-kugaevskikh@yandex.ru

Аннотация. В работе представлено описание разработанных нейронных сетей для классификации спектров газо-воздушной смеси, выдыхаемой человеком. Биомаркеры COVID до настоящего времени до конца не определены, поэтому проводился анализа всего спектра выдоха. Разработанные подходы позволили с высокой степенью точности определять COVID. Проведенная серия экспериментов подтвердила возможность диагностики COVID-19 с помощью нейронных сетей.

Ключевые слова: COVID, скрининг, нейронная сеть, классификация спектров, выдох

I. ВВЕДЕНИЕ

Идея классификации болезней по спектру выдоха человека не нова. Наибольшей популярностью обладают два подхода получения спектров для этой задачи: спектроскопия и использование электронного носа. Для классификации спектров, полученных с помощью этих технологий, используют нейронные сети, что дает возможность сочетать локальное выделение признаков и их глобальное пространственное расположение на всем спектре.

Электронный нос представляет собой массив сенсоров, чаще всего на базе металло-оксидных полупроводников или электропроводящих полимеров. В случае с диагностикой заболеваний электронный нос отвечает на специфические признаки летучих органических веществ (volatile organic compounds, VOC), которых содержится в выдохе человека около 3000. В работе Dragonieri [1] приведен хороший обзор применения машинного обучения для диагностики с помощью технологии электронного носа заболеваний, среди которых астма, рак легких, респираторные инфекции, хроническая обструктивная болезнь легких и др. Также машинное обучение к спектрам летучих органических веществ применяют для диагностики колоректального рака [2], рака молочной железы [3], рак простаты [4]. Рак легких диагностируется с точностью выше 80 % [1, 5]. Помимо заболеваний, профили электронного носа используют для распознавания взрывоопасных газов [6].

Спектроскопия объединяет большое количество технологий, спектры которых обрабатывают с помощью нейросетей, а именно спектры лазерно-искровой эмиссионной спектроскопии (LIBS) [7–15], атомно-эмиссионной спектроскопии с индуктивно связанной плазмой [17], светоразрядной оптической эмиссионной спектроскопии [16, 18], инфракрасной спектроскопии [19, 20]. В основном спектроскопия используется для классификации твердых образцов и гораздо реже газов.

Среди нейросетевых моделей особой популярностью пользуется многослойный персептрон [2–12, 16–18].

Отдельно стоит отметить применение сверточных нейронных сетей [13–15, 19, 20], что позволяет выделять признаковые окна классов.

Но далеко не всегда эти попытки заканчиваются успешно. Причин может быть несколько: 1) высокая зашумленность сигнала, 2) большая флуктуация внутри каждого класса из-за неправильной эксплуатации спектрометра и индивидуальных особенностей человека, 3) специфические признаки могут составлять очень малую часть спектра, которая на фоне остальной малозначимой информации не будет ярко выделена в силу глобального характера построения разделяющей гиперплоскости, 4) небольшая и несбалансированная выборка.

II. МАТЕРИАЛЫ И МЕТОДЫ

Нами рассматривается задача скрининга COVID-19 по спектрам, полученным в результате использования светоразрядной оптической эмиссионной спектроскопии. На рис. 1 представлена схема устройства [21], основанного на этой технологии и предназначенного для измерения концентрации ацетона в выдохе человека. Для скрининга мы использовали эмиссионный спектр выдоха человека максимального насыщения, получаемый с усовершенствованной версии этого устройства.

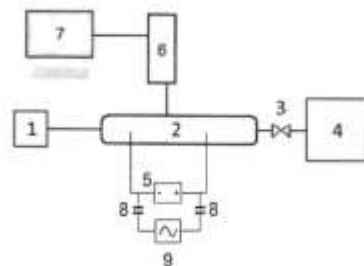


Рис. 1. Схема газоанализатора (1 – насос прокачки, 2 – разрядная ячейка, 3 – регулируемый клапан, 4 – линия отбора пробы, 5 – источник питания постоянного тока, 6 – спектрометр, 7 – блок анализа и обработки, 8 – конденсатор, 9 – источник питания переменного тока)

Было собрано несколько отличающихся наборов данных, собранных на разных аппаратах. Характеристики наборов данных представлены в табл. 1.

ТАБЛИЦА I НАБОРЫ ДАННЫХ

Метка	Кол-во «здоровых» спектров	Кол-во КОВИД спектров	Метод подтверждения	Метка устройства
D1	76	71	СТ, PCR	M1
D2	212	31	PCR	M1
D3	252	115	PCR	M2
D4	322	134	PCR	M2, M3, M4

Визуализация набора данных D4 методом t-sne, рис. 2, показывает нелинейную разделимость классов, что в совокупности с неизвестными биомаркерами КОВИД ограничивает применимые методы машинного обучения только нейронными сетями.

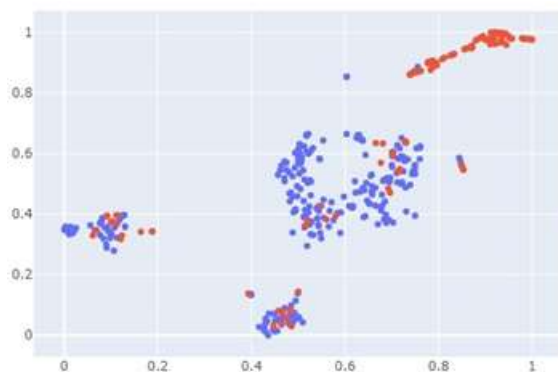


Рис. 2. Визуализация набора данных

Аппарат нейронных сетей сейчас очень разнообразен. Классификация спектров трехслойным персептроном показала низкий результат (ассигасу 52 %), при этом, в процессе обучения растет разброс между ассигасу на обучающей и тестовой выборке, что говорит о переобучении нейронной сети. Поэтому в дальнейших экспериментах использовалось представление спектра не в виде вектора, а в виде изображения 2048*1929 пикселей.

Кроме того, переход от спектров к их изображениям обусловлен рядом более существенных причин:

- 1) значение концентрации, измеренное в каждом пикселе спектрометра, варьируется на ± 500 единиц в разных образцах, при этом общая картина спектра сохраняется.
- 2) для диагностики заболевания нас интересует не точное значение концентрации, а ее превышение некоторого предела, еще не описанного в литературе. В этом случае окно свертки на графике спектра может помочь больше, чем взвешенная сумма спектрального сигнала.

Соответственно, применяемые нейронные сети были ограничены нейронными сетями, классифицирующими изображения. Учитывая малое отличие изображений друг от друга и сложность вычленения характерных для классов признаков, была выбрана архитектура ResNet-34, которая представляет собой последовательность сверток разного размера. Ее отличительной особенностью является использование соединений быстрого доступа, позволяющее избежать проблемы затухающих градиентов. Применительно к нашей задаче, нейросеть обучаясь на имеющейся статистике, выделяет зависимости между классом (заболеванием) и окнами на изображении (биомаркерами).

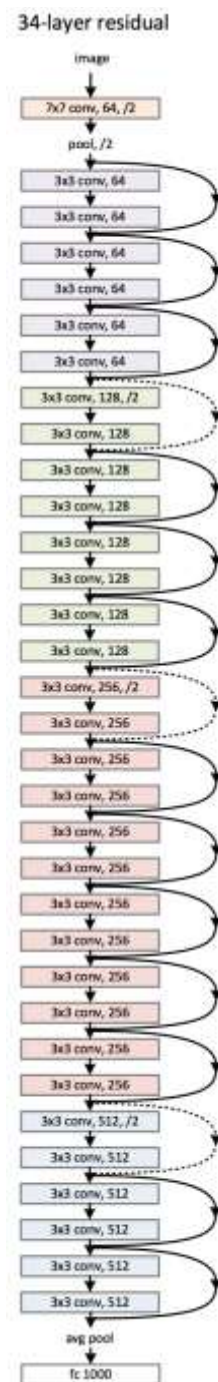


Рис. 3. Архитектура ResNet

III. РЕЗУЛЬТАТЫ

Нами было проведено обучение архитектуры ResNet со следующими параметрами:

1. Функция потерь – перекрестная энтропия
2. Оптимизатор – Adam
3. Скорость обучения – 0,00001
4. Размер пакета – 32

Обучающая выборка составляла 70 % спектров набора данных в каждом классе, остальные спектры составили тестовую выборку. Процесс обучения был ограничен 45 эпохами. Рис. 3 показывает изменение значения функции потерь с каждой эпохой.

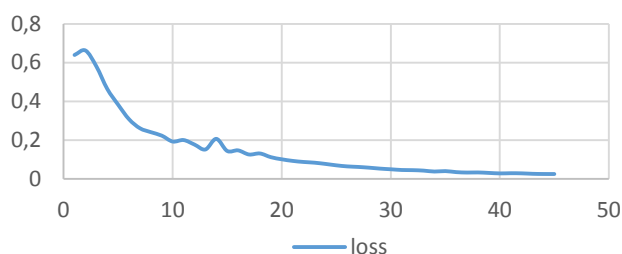


Рис. 4. Изменение значения функции стоимости при обучении

Табл. 2 показывает результаты проверки на сформированных наборах данных.

ТАБЛИЦА 2 РЕЗУЛЬТАТЫ ПРОВЕРКИ

Метка	Accuracy	Sensitivity	Specificity
D1	85,37%	66,67%	100%
D3	87,14%	74,29%	100%
D4	87,5%	82,5%	91,07%

Как было указано Dragonieri [1] пол и возраст хоть и модифицируют отдельные VOC, но, по-видимому, не влияют на весь профиль спектра. Мы также не заметили значительного влияния пола и возраста на спектр выдоха. При этом беременность, сопутствующие заболевания (сахарный диабет, гастроэзофагеальный рефлюкс, почечная недостаточность, болезни сердца и т. п.), курение могут влиять как на точность классификации профиля выдоха, полученного с помощью технологии электронного носа, что также отметил Dragonieri [1], так и на спектр выдоха, полученный путем применения устройства [21].

IV. ЗАКЛЮЧЕНИЕ

Предложенная нами гипотеза использования спектров для классификации доказала, как минимум, свою состоятельность. Также можно сказать и про использование светоразрядной оптической эмиссионной спектроскопии, не смотря на определенные сложности с получением стабильных спектров. Основная трудность в развертывании данной технологии заключается в малой и несбалансированной выборке данных. Влияние целого ряда факторов, упомянутых выше, еще предстоит изучить. Тем не менее есть все основания утверждать, что использование нейронных сетей для диагностики некоторых заболеваний по спектру выдоха человека вполне возможно.

Даже в идеальных условиях устройства не могут быть собраны одинаково, всегда будет небольшая разница в смещении пиков спектра вдоль оси абсцисс. Предложенная гипотеза, как показал эксперимент с набором данных D4, позволяет нивелировать эту разницу.

СПИСОК ЛИТЕРАТУРЫ

- [1] S. Dragonieri, G. Pennazza, P. Carratu, and O. Resta, 'Electronic Nose Technology in Respiratory Diseases', *Lung*, vol. 195, no. 2, pp. 157–165, Apr. 2017, doi: 10.1007/s00408-017-9987-3.
- [2] E. G. M. Steenhuis et al., 'Feasibility of volatile organic compound in breath analysis in the follow-up of colorectal cancer: A pilot study', *European Journal of Surgical Oncology*, p. S0748798320306557, Jul. 2020, doi: 10.1016/j.ejso.2020.07.028.
- [3] O. Herman-Saffar, Z. Boger, S. Libson, D. Lieberman, R. Gonen, and Y. Zeiri, 'Early non-invasive detection of breast cancer using exhaled breath and urine analysis', *Computers in Biology and Medicine*, vol. 96, pp. 227–232, May 2018, doi: 10.1016/j.combiomed.2018.04.002.
- [4] C. G. Waltman, T. A. T. Marcelissen, and J. G. H. van Roermund, 'Exhaled-breath Testing for Prostate Cancer Based on Volatile

- Organic Compound Profiling Using an Electronic Nose Device (Aeonose™): A Preliminary Report', *European Urology Focus*, p. S2405456918303584, Nov. 2018, doi: 10.1016/j.euf.2018.11.006.
- [5] D. Marzorati, L. Mainardi, G. Sedda, R. Gasparri, L. Spaggiari, and P. Cerveri, 'A Metal Oxide Gas Sensors Array for Lung Cancer Diagnosis Through Exhaled Breath Analysis', in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, Jul. 2019, pp. 1584–1587, doi: 10.1109/EMBC.2019.8856750.
- [6] D.-S. Lee et al., 'Explosive gas recognition system using thick film sensor array and neural network', *Sensors and Actuators B: Chemical*, vol. 71, no. 1–2, pp. 90–98, Nov. 2000, doi: 10.1016/S0925-4005(00)00614-6.
- [7] E. Ewusi-Annan, D. M. Delapp, R. C. Wiens, and N. Melikechi, 'Automatic preprocessing of laser-induced breakdown spectra using partial least squares regression and feed-forward artificial neural network: Applications to Earth and Mars data', *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 171, p. 105930, Sep. 2020, doi: 10.1016/j.sab.2020.105930.
- [8] E. D'Andrea et al., 'An artificial neural network approach to laser-induced breakdown spectroscopy quantitative analysis', *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 99, pp. 52–58, Sep. 2014, doi: 10.1016/j.sab.2014.06.012.
- [9] J.O. Caceres, S. Moncayo, J.D. Rosales, F.J. M. de Villena, F.C. Alvira, and G.M. Bilmes, 'Application of Laser-Induced Breakdown Spectroscopy (LIBS) and Neural Networks to Olive Oils Analysis', *Appl Spectrosc*, vol. 67, no. 9, pp. 1064–1072, Sep. 2013, doi: 10.1366/12-06916.
- [10] X. Cui, Q. Wang, Y. Zhao, X. Qiao, and G. Teng, 'Laser-induced breakdown spectroscopy (LIBS) for classification of wood species integrated with artificial neural network (ANN)', *Appl. Phys. B*, vol. 125, no. 4, p. 56, Apr. 2019, doi: 10.1007/s00340-019-7166-3.
- [11] R. Junjuri, A. Prakash Gummadi, and M. Kumar Gundawar, 'Single-shot compact spectrometer based standoff LIBS configuration for explosive detection using artificial neural networks', *Optik*, vol. 204, p. 163946, Feb. 2020, doi: 10.1016/j.ijleo.2019.163946.
- [12] D. Pokrajac et al., 'Automatic Classification of Laser-Induced Breakdown Spectroscopy (LIBS) Data of Protein Biomarker Solutions', *Appl Spectrosc*, vol. 68, no. 9, pp. 1067–1075, Sep. 2014, doi: 10.1366/14-07488.
- [13] J. Chen, J. Pisonero, S. Chen, X. Wang, Q. Fan, and Y. Duan, 'Convolutional neural network as a novel classification approach for laser-induced breakdown spectroscopy applications in lithological recognition', *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 166, p. 105801, Apr. 2020, doi: 10.1016/j.sab.2020.105801.
- [14] L.-N. Li, X.-F. Liu, W.-M. Xu, J.-Y. Wang, and R. Shu, 'A laser-induced breakdown spectroscopy multi-component quantitative analytical method based on a deep convolutional neural network', *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 169, p. 105850, Jul. 2020, doi: 10.1016/j.sab.2020.105850.
- [15] C. Lu, B. Wang, X. Jiang, J. Zhang, K. Niu, and Y. Yuan, 'Detection of K in soil using time-resolved laser-induced breakdown spectroscopy based on convolutional neural networks', *Plasma Sci. Technol.*, vol. 21, no. 3, p. 034014, Mar. 2019, doi: 10.1088/2058-6272/aaef6e.
- [16] S. Das, D. P. Das, C. K. Sarangi, and B. Bhoi, 'Estimation of hydrogen flow rate in atmospheric Ar:H₂ plasma by using artificial neural network', *Neural Comput & Applic*, vol. 32, no. 5, pp. 1357–1365, Mar. 2020, doi: 10.1007/s00521-018-3674-z.
- [17] Z. Li, S. Huang, and V. Karanassios, 'Further development of artificial neural networks for spectral interference correction in optical emission spectrometry', *Baltimore, Maryland*, May 2012, pp. 84010Y-84010Y-7, doi: 10.1117/12.919570.
- [18] C.-Y. Wang and C.-C. Hsu, 'Development and testing of an efficient data acquisition platform for machine learning of optical emission spectroscopy of plasmas in aqueous solution', *Plasma Sources Sci. Technol.*, vol. 28, no. 10, p. 105013, Oct. 2019, doi: 10.1088/1361-6595/ab45e5.
- [19] C. Yuanyuan and W. Zhibin, 'Quantitative analysis modeling of infrared spectroscopy based on ensemble convolutional neural networks', *Chemometrics and Intelligent Laboratory Systems*, vol. 181, pp. 1–10, Oct. 2018, doi: 10.1016/j.chemolab.2018.08.001.
- [20] S. Zhou et al., 'An adaptive Kalman filtering algorithm based on back-propagation (BP) neural network applied for simultaneously detection of exhaled CO and N₂O', *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 223, p. 117332, Dec. 2019, doi: 10.1016/j.saa.2019.117332.
- [21] M.A. Dvedenidov and S.N. Atutov, 'Method of controlling concentration of acetone in air exhaled by a human, and a device for realizing it', RU 2697809 C1, Aug. 20, 2019.

Подход к решению оптимизационной задачи в системе управления балансом газа региональной газотранспортной сети

А. К. Петрова

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
akpetrova@etu.ru

Аннотация. Сведение к минимуму небаланса газа в региональных газотранспортных сетях является актуальной задачей. Переменными в данной оптимизационной задаче являются объемы поставленного и потребленного газа, запаса газа в магистральном трубопроводе, потребление на собственные нужды. Ограничениями являются расходы на контроль и поддержание объектов ГТС в состоянии, обеспечивающем ее стабильную работу. Поиску решений данной задачи при помощи интеллектуальных методов посвящена эта статья.

Ключевые слова: оптимизационная задача, баланс и небаланс газа в газотранспортной сети, нелинейные переменные, условия/ограничения, интеллектуальные методы

I. ПОСТАНОВКА ЗАДАЧИ

Баланс газа – количественное итоговое соотношение поступлений газа и отбора, в том числе на собственные нужды, и (или) отпуска газа, с учетом остатков [1] – служит важным критерием эффективности газотранспортной системы. Система управления балансом газа – это совокупность носящих систематический характер мероприятий, методов, алгоритмов и процедур оценивания баланса газа в ГТС, целенаправленная на выявление небаланса газа, а также его количества. Цель: достижение баланса газа входящих и выходящих потоков в ГТС. Формирование баланса газа за отчетный период происходит в результате выполнения операций учета расхода газа (сутки, месяц, год). В идеале должно быть обеспечено равенство количества поставленного и потребленного газа. Если равенства не происходит, значит, в системе имеет место небаланс. Согласно [1]

$$Q_{\text{неб}} = Q_{\text{потр}} - Q_{\text{пост}} + Q_{\text{запас}} \quad (1)$$

где $Q_{\text{потр}}$ – объемы потребленного газа:

$$Q_{\text{потр}} = Q_{\text{грс}} + Q_{\text{гис}} + Q_{\text{сн}} \quad (2)$$

В данной формуле $Q_{\text{грс}}$ – расход газа на газораспределительных станциях ГРС (поставка газа потребителям); $Q_{\text{гис}}_{\text{вых}}$ – расход газа на ГИС (газоизмерительных станциях, передача в другой регион); $Q_{\text{сн}}$ – расход газа на собственные нужды; $Q_{\text{пост}} = Q_{\text{гис}}_{\text{вх}}$ – объемы поставленного газа, определяется на выходе газоизмерительной станции на входе региона.

$Q_{\text{запас}}$ – запас газа на рассматриваемом участке.

$$Q_{\text{запас}} = Q_{\text{пхг}} + Q_{\text{мг}} \quad (3)$$

$Q_{\text{пхг}}$ – запас газа в подземном хранилище газа (ПХГ);

$Q_{\text{мг}}$ – запас газа в магистральном газопроводе (МГ).

Тогда из (1) можно выразить величину небаланса газа в ГТС:

$$Q_{\text{неб}} = Q_{\text{грс}} + Q_{\text{гис}} + Q_{\text{сн}} + Q_{\text{пхг}} + Q_{\text{мг}} - Q_{\text{пост}} \quad (4)$$

Задачу контроля баланса газа в ГТС можно свести к оптимизационной задаче:

$$Q_{\text{неб}} = (Q_{\text{пост}} - (Q_{\text{грс}} + Q_{\text{гис}} + Q_{\text{сн}} + Q_{\text{пхг}} + Q_{\text{мг}})) \rightarrow \min \quad (5)$$

где $Q_{\text{неб}}$ – целевая функция.

II. ПЕРЕМЕННЫЕ И ОГРАНИЧЕНИЯ

А. Объемы потребленного газа

В технологическом процессе учета расхода газа, для определения величин $Q_{\text{пост}}$, $Q_{\text{грс}}$, $Q_{\text{гис}}$, $Q_{\text{сн}}$, при использовании наиболее распространенного метода – переменного перепада давлений, формула определения расхода газа с помощью диафрагмы имеет следующий вид [2]:

$$q = (\pi \cdot d_{20}^2 / 4) \cdot K_{\text{св}}^2 \cdot K_{\text{ш}} \cdot K_{\text{п}} \cdot C \cdot E \cdot (2 \Delta p / \rho_v)^{0.5} \quad (6)$$

где q – расход газа, м³; d_{20} – диаметр отверстия диафрагмы при температуре 20°C; $K_{\text{св}}$ – коэффициент, учитывающий изменение диаметра отверстия диафрагмы, вызванное отклонением температуры природного газа от 20 °C; $K_{\text{ш}}$ и $K_{\text{п}}$ – коэффициенты шероховатости внутренней поверхности стенки ИТ и притупления входной кромки диафрагмы (при использовании других сужающих устройств $K_{\text{п}} = 1$); C – коэффициент истечения; Δp – перепад давления на сужающем устройстве, Па; ρ_v – плотность среды при рабочих условиях; $E = 1/(1 - \beta^4)^{0.5}$ – коэффициент скорости входа; $\beta = d/D$ – относительный диаметр отверстия сужающего устройства; d – диаметр отверстия диафрагмы при рабочей температуре; D – диаметр трубы.

Для $Q_{\text{пост}}$, $Q_{\text{грс}}$, $Q_{\text{гис}}$, $Q_{\text{сн}}$ будем меняться величина q , соответственно.

В. Запас газа

- Запас газа в магистральных трубопроводах

Согласно [3] для оценки запаса газа в магистральных трубопроводах в пределах газотранспортного предприятия используется формула:

$$Q = \sum_{j=1}^J Q_i \quad (7)$$

где J – число расчетных участков.

Запас газа на участке многониточной системы магистрального газопровода при стандартных условиях определяется по формуле:

$$Q_j = \sum_{i=1}^n \frac{V_i \cdot P_{icp} \cdot 293.15}{1.033 \cdot z_{icp} \cdot 10^6 (T_{icp} + 273.15)} \quad (8)$$

где V_i – геометрический объем j -й нитки газопровода, м³; P_{icp} – среднее давление j -ой нитки газопровода, кгс/см²; n – число ниток магистрального газопровода; z_{icp} – средний коэффициент сжимаемости j -ой нитки; T_{icp} – средняя температура газа j -ой нитки, °C.

- Запас газа в подземных хранилищах газа

Остаточный запас газа в ПХГ, может быть определен объемным методом [4].

Формула для подсчета запасов газа в ПХГ имеет следующий вид:

$$V = F \cdot h \cdot k \cdot K \cdot P / O_r \quad (9)$$

где V – количество газа, находящегося в ПХГ на дату расчета, м³; F – площадь в пределах продуктивного контура, м; h – мощность пористой части пласта, м; k – коэффициент пористости; $K = (T + t_{cm}) / (T + t_m)$ – поправка на температуру для приведения объема газа к стандартной температуре; $t_{ct} = 20^\circ\text{C}$; t_m – пластовая температура; T – абсолютная температура, К; P – среднее абсолютное давление в залежи на дату расчета, кгс/см²; O_r – коэффициент газонасыщенности.

С. Определение условий и ограничений

В результате опроса экспертов на данном этапе были определены следующие ограничения:

- затраты на поддержание баланса газа в ГТС (на профилактическое обслуживание, устранение утечек и других инцидентов, и пр.); $K_z \leq K_{max}$;
- затраты на транспортировку газа $K_{tr} = \min$
- загрузка ГРС: $K_{ГРС} = \max$;
- величина запаса газа в магистральном газопроводе и в подземном хранилище не может быть отрицательной $Q_{mg} \geq 0$, $Q_{пхг} \geq 0$;
- величина поставленного и израсходованного газа на ГИС, ГРС, и на собственные нужды должна быть положительной: $Q_{пост} > 0$, $Q_{грс} > 0$, $Q_{гис} > 0$, $Q_{сн} > 0$
- утечки газа в ГТС: $K_{ут} = \min$;
- изменения в режиме поставок должны быть минимальны $K_{изм} = \min$.

Общий вид оптимизационной задачи:

Целевая функция: $Q_{неб} = \min$

Переменные: $Q_{пост}$, $Q_{гис}$, $Q_{сн}$, $Q_{пхг}$, Q_{mg}

Ограничения: K_z , $K_{ГРС}$, K_{tr} , $K_{ут}$, $K_{изм}$, $Q_{пост}$, $Q_{гис}$, $Q_{сн}$, $Q_{пхг}$, Q_{mg} .

Проанализировав формулы, описывающие переменные, можно увидеть, что и целевая функция, и переменные нелинейно зависят от совокупности факторов, что приводит к задаче нелинейной оптимизации. С другой стороны, анализ критериев показал, что задача является многокритериальной, так как для различных режимов функционирования решаются разные задачи. Например, в штатном режиме решается частная задача минимизации затрат на транспортировку при соблюдении плана поставок газа, а в аварийном режиме решается частная задача минимизации потерь при условии ограничения на изменения в режиме поставок. Таким образом, необходимо решить задачу многокритериальной нелинейной оптимизации [5–6].

III. ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ НА ОСНОВЕ ИНТЕЛЛЕКТУАЛЬНЫХ МЕТОДОВ

Существует множество методов оптимизации: методы случайного поиска, метод ветвей и границ [5–7], эволюционные вычисления [8–9] и др. Решение задачи оптимизации также может быть получено путем создания экспертных систем [10, 11], систем поддержки принятия решений [12]. Для решения таких задач в настоящее время исследуется применение современных интеллектуальных методов, таких, как нейронные сети, поскольку они обладают возможностью обрабатывать большие объемы нелинейных данных, инварианты к шумам в данных [13, 14]. В частности, в работе [15] рассматривается новый метаэвристический алгоритм оптимизации. Предлагаемый метод, названный алгоритмом нейронной сети (НСА), разработан на основе уникальной структуры ИНС. NNA выигрывает от сложной структуры ANN и ее операторов для создания новых решений. В работе [16] предложено использовать нейронные сети Хопфилда. «Нейрон» в этой работе представляется вычислительным процессом, обладающим состоянием. Искусственная сеть «обучается» решению некоторой задачи, что, по сути, сводится к вычислениям весовых коэффициентов матрицы. В работе [17] исследуется нейросетевой метод решения нелинейной задачи оптимального распределения неоднородного ресурса. Авторы предлагают принимать энергетическую функцию нейронной сети в качестве оптимизируемого параметра, и подбирать веса в ходе обучения таким образом, чтобы достичь минимума.

В данной статье рассмотрен подход к решению оптимизационной задачи на основе применения нейросетевых алгоритмов для прогнозирования некоторых входных переменных. Результат сравнивается с решением, полученным на основе применения выражений (6)–(9) для определения этих переменных. Для расчетов использовался инструментальный Matlab, пакеты Optimization Toolbox, Neural Network Toolbox, Deep Learning Toolbox. Расчеты строились на основе статистических данных региональной ГТС, данных были предварительно нормированы. Для упрощения расчетов, на данном этапе некоторые переменные, (такие как расходы на собственные нужды, объемы поставленного газа и запаса в ПХГ), а также ограничения, были зафиксированы: выбран штатный режим функционирования, затраты на транспортировку газа в ГТС постоянны, изменения в режиме поставок $K_{изм} = 0$, загрузка $K_{ГРС} = 100\%$.

А. Нейронная сеть прямого распространения для прогнозирования расхода газа на ГИС

В результате сравнения различных архитектур, синтезирована и обучена трехслойная нейронная сеть для прогнозирования объемом расхода газа на ГИС. Архитектура сети представлена на рис. 1.

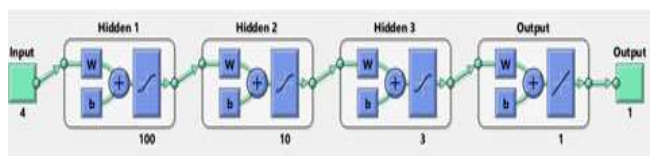


Рис. 1. Архитектура трехслойной сети прямого распространения для прогнозирования расхода газа на ГИС

Входные факторы: расход в стандартных условиях $Q, \text{м}^3$, избыточное давление в трубе $P_{\text{н}}, \text{кгс/см}^2$, температура измерительной нитки на выходе трубы $t_{\text{вых}}, ^\circ\text{C}$, теплотворная способность газа $H, \text{МДж}$. На вход подаются вектора входных параметров, второй и третий – полносвязный слой прямого распространения, далее следует слой Softmax, который нормализует входные данные слоя таким образом, что сумма его элементов равна 1, на выходе – величина объемного расхода газа на ГИС, полученная в результате работы алгоритма. Рекуррентная LSTM сеть для прогнозирования расхода газа на ГПС

Сеть LSTM – тип рекуррентной нейронной сети, которая содержит LSTM-модули, способные «забывать» или «запоминать», поступившие на них данные как на короткие, так и на длинные промежутки времени. Поскольку статистические данные ГТС представляют из временные ряды, то такая сеть может показать хорошие результаты [18]. Архитектура LSTM сети представлена на рисунке 2.

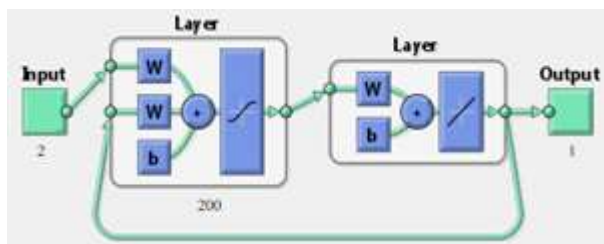


Рис. 2. Архитектура LSTM сети для прогнозирования расхода газа на ГПС

В качестве входных факторов первого слоя используются: дата, расход в стандартных условиях $Q, \text{м}^3$, второй слой - LSTM, далее следует полносвязный слой прямого распространения. На выходе – величина объемного расхода газа на ГПС, полученная в результате работы алгоритма. Результатом обучения сети LSTM является такой набор весов, который, на основе имеющихся статистических данных, обеспечит правильное определение данных, которые необходимо «забыть» или «запомнить».

В. Нейронная сеть прямого распространения для прогнозирования запаса газа

Построена модель регрессионной зависимости запаса газа на участке газораспределительного трубопровода со следующими параметрами: давление в трубопроводе $P_{\text{ср}}(\text{кгс/см}^2)$; температура газа в трубопроводе $T_{\text{ср}}, ^\circ\text{C}$;

коэффициент сжимаемости газа в трубопроводе $Z_{\text{ср}}$; температура грунта $T_{\text{гр.реал}}, ^\circ\text{C}$, изменение запаса газа в трубопроводе, $Q_{\text{запас.реал}}, \text{тыс. м}^3$. На выходе – величина запаса газа в МГ, полученная в результате работы алгоритма. Архитектура сети представлена на рис. 3.

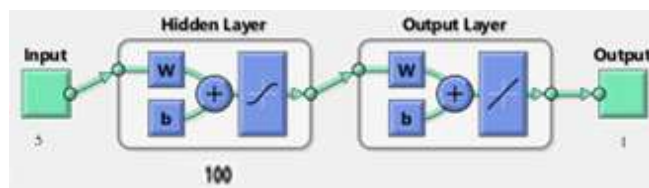


Рис. 3. Архитектура однослойной сети прямого распространения для прогнозирования запаса газа в МГ

С. Сравнение результатов

Поскольку матрицы спутанности показали корректность работы указанных выше алгоритмов, построенные на их основе функции были применены в качестве входных переменных для оптимизационной задачи, поставленной ранее. Для сравнения, такие же расчеты осуществлены с применением в качестве входных переменных расчетных формул (6)–(9). В таблице приведены результаты сравнения применённых подходов.

ТАБЛИЦА 1 СРАВНЕНИЕ ПОДХОДОВ К РЕШЕНИЮ ОПТИМИЗАЦИОННОЙ ЗАДАЧИ В ГТС

Характеристики подходов	Критерии сравнения	
	Длительность работы, сек	Решение, м^3
определены с помощью формул (6)–(9)	5,6	1.21
Входные переменные определены с помощью нейросетевых алгоритмов	11,78	0.87

Результаты показали эффективность интеллектуальных методов, однако длительность их работы больше, что может сказаться негативно на эффективности принимаемых решений.

IV. ЗАКЛЮЧЕНИЕ

В рамках работы проведено сравнение подходов к решению задачи оптимизации баланса газа в газотранспортной сети. Интеллектуальные методы показали эффективность с точки зрения найденного минимального решения, но большую длительность вычислений. В дальнейшем предполагается изучить возможность задавать фиксировано веса для некоторых входных переменных и ограничений в соответствии с ситуацией. Например, в штатном режиме задается максимальный вес для коэффициентов затрат на поддержание баланса и изменений в режимах поставок $K_z, K_{\text{изм}}$, а в случае обнаружения утечки, задается максимальный весовой коэффициент для коэффициента утечек $K_{\text{ут}}$. Помимо этого, возможно исследование влияния на результат различных архитектур и гиперпараметров используемых алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

- [1] Организация, планирование и управление на предприятиях транспорта и хранения нефти и газа: учебник для вузов / А.Д. Бренц [и др.]. М.: Недра, 1980. 360 с.

- [2] ГОСТ 8.586.1-2005 - ГОСТ 8.586.5.-2005. Измерение расхода и количества жидкостей и газов с помощью стандартных сузжающих устройств.
- [3] СТО ГАЗПРОМ 2-3.5-051-2006.
- [4] Методические указания по определению технологически необходимых безвозвратных потерь газа при создании и эксплуатации газохранилищ в пористых пластах. Москва, 1996, ПАО "Газпром". "ВНИИГАЗ".
- [5] Акулич И. Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов. М.: Высшая школа, 1986.
- [6] Гилл Ф., Мюррей У., Райт М. Практическая оптимизация / Пер. с англ. М.: Мир, 1985.
- [7] Жиглявский А.А., Жилинскас А.Г. Методы поиска глобального экстремума. М.: Наука, 1991. 248 с.
- [8] Букатова И.Л., Михасев Ю.И., Шаров А.М. Эвоинформатика: Теория и практика эволюционного моделирования М.: Наука, 1991. 206 с.
- [9] Методы робастного, нейро-нечеткого и адаптивного управления / К.А. Пупков [и др.]; под ред. Н.Д. Егупова. М.: МГТУ им. Н. Э. Баумана, 2001. 744 с.
- [10] Джарратано Д., Райли Г. Экспертные системы: принципы разработки и программирование. М.: Вильямс, 2007. 1152 с.
- [11] Джексон П. Введение в экспертные системы. М.: Вильямс, 2001. 624 с.
- [12] Терелянский П.В. Системы поддержки принятия решений. Опыт проектирования : монография. Волгоград: Изд-во ВолгГТУ, 2009. 127 с.
- [13] Назаров А.В., Лоскутов А.А. Нейросетевые алгоритмы прогнозирования и оптимизации систем: Наука и Техника, СПб., 2003, 384 с.
- [14] Усков А.А., Кузьмин А.В. Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика. М.: Горячая линия - Телеком, 2004. 143 с.
- [15] Али Садолла (2022). Алгоритм нейронной сети (NNA) для оптимизации с ограничениями (<https://www.mathworks.com/matlabcentral/fileexchange/68473-neural-network-algorithm-nna-for-constrained-optimization>), MATLAB Central File Exchange.
- [16] Хассанин Хатем Мохамед Абдель Максуд Решение задач оптимизации с помощью нейронной сети Хопфилда. Фундаментальные исследования № 2, 2015.
- [17] Жук А.А., Булойчик В.М. Нейросетевой метод решения нелинейной задачи оптимального распределения неоднородного ресурса. «Системный анализ и прикладная информатика». 2021;(1):45-52.
- [18] S. Hochreiter, J.Schmidhuber Long short-term memory, Neural Computation 9(8):1735{1780, 1997.

Prediction of Next App in OS

Ruimin Ma

Faculty of Applied Mathematics
and Control Processes
Saint-Petersburg State University
Saint Petersburg, Russia
mrm19990206@gmail.com

Yuyi Zhang

Faculty of Applied Mathematics
and Control Processes
Saint-Petersburg State University
Saint Petersburg, Russia
lesliezhang0825@gmail.com

Jing Liu

Faculty of Applied Mathematics
and Control Processes
Saint-Petersburg State University
Saint Petersburg, Russia
st082130@student.spbu.ru

Ovanes Petrosian^{1,2}

Faculty of Applied Mathematics
and Control Processes

¹Saint-Petersburg State University;
Faculty of Computer Science and
Technology

²Saint Petersburg Electrotechnical
University "LETI"
Saint Petersburg, Russia
petrosian.ovanes@yandex.ru

Kirill Krinkin

Faculty of Computer Science and
Technology

Saint Petersburg Electrotechnical
University "LETI"
Saint Petersburg, Russia
kirill@krinkin.com

Abstract—With the popularity of smart devices, application responsiveness is one of the most important indicators of user experience, and forecast the next application will be used by users is crucial to the management planning for related companies. In this paper, more sophisticated machine learning, and even deep learning, which can achieve better forecast performance, is widely used in this field. However, the opacity of these black-box models greatly limits user trust and how well developers can optimize the model. To address these issues, this paper first tests six of the most popular forecasting algorithms, including ensemble models and neural networks, to select the optimal model. As an innovation, this paper also uses XAI techniques to explain black-box models to increase user trust in the results generated by forecast models and to help developers in their work. After completing the above work, on the one hand, we found that the ensemble model performed better in the time series datasets with user application usage information, especially with LightGBM, on the other hand, we found that the prediction model using the SHAP method showed that the target variable categorical features and lags Features are important features to forecast the user's next application.

Keywords—Application, Time-series forecasting, Ensemble model, Neural network, Explainable AI

I. INTRODUCTION

With the rapid development of society and the increasing improvement of people's living standards, users have installed a large number of mobile applications (referred to as Apps) on their mobile phones, and one of the important indicators of user experience degree is response speed. Trinh et al [1] collected data on users' use of Apps and concluded that there is a pattern in users' behaviour in using Apps on their mobile phones, so forecast in advance which one users will use next. Further, by improving the user experience, the user experience of digital services [2] can be fundamentally improved.

Much research work has been conducted on predicting which app user will use [3]. According to a survey by [4], Liao et al [5] forecasted the next app based on the temporal profile of each app mined, Huang et al [6] noted that the latest used app and time is more effective than location.

Currently, models based on different algorithms are widely used in time series forecasting competitions, including boosting algorithms [7], bagging algorithms [8] and neural network algorithms [9]. The performance of these algorithms [10] is not consistent across competitions, which means that they are difficult to compare and measure. Therefore, the selection of a better forecast model by comparing different forecast models is crucial to solving practical forecast problems.

In this paper, on the one hand, we use popular forecast models to predict in advance which app a user will use next, and compare the forecast models in a comprehensive way to identify a better forecast model, these forecast models include XGBoost [11], LightGBM [12], Random Forest [13], Bi-RNN [14], Bi-LSTM [15] and Bi-GRU [16]. On the other hand, we need to explain the identified better forecast models in order to gain more insight into the feature information learned by the forecast models and the explanation method chosen is SHAP [17].

To achieve better define the prediction model, Section 2 details the relevant datasets information, Section 3 presents the forecast model and the XAI algorithms, Section 4 shows the forecast results and comparison results using classical measures, as well as the explanation results using the XAI algorithms, and Section 5 shows the conclusions and an introduction to future work.

II. DATASETS DESCRIPTION

A. LiveLab – Measuring wireless networks and smartphone users in the field

It contains the name of the application, time of use, date of use, duration, type, price and other information [18].

B. LSApp – Large dataset of Sequential mobile App usage

It contains the unique user identifier, the unique session identifier, the application name, timestamp, type of the event recorded and other information [19].

C. Tsinghua App Usage Dataset

It contains User ID, Timestamp, Location (base station ID), Traffic, Category ID, application name, Number of PoIs (Points of Interests) for each category and other information [20].

III. SOLUTION APPROACH

A. Forecast Algorithms

Both ensemble models and neural networks are considered to be the best forecast algorithms in relation to defining the direction of application usage. Among them, the ensemble model is divided into boosting algorithm and bagging algorithm. In this work, we choose XGBoost [11] and LightGBM [12] algorithms as representatives of boosting algorithms, and Random Forest [13] as representatives of bagging algorithms. Bi-RNN [14], Bi-LSTM [15] and Bi-GRU [16] are used as representations of neural network models.

XGBoost extends and improves the GBDT algorithm, and its base learner can be either a CART or a linear classifier. XGBoost uses column block to optimize memory to achieve parallel processing to improve model training speed [11].

LightGBM is a gradient boosting framework that uses a tree-based learning algorithm [9]. It uses a histogram algorithm and only keeps discrete implementations of features to reduce memory consumption and increase the speed of the algorithm. It uses the GOSS algorithm, keeping instances with larger gradients and randomly selecting instances with smaller gradients to balance the speed and performance of LightGBM.

Random forest algorithm [10] is based on independent decision trees. The results of random sampling methods have broad diversity and can reduce the correlation between multiple independent decision trees. The “voting” method selects the best result.

The idea of a bidirectional neural network is to divide the neurons of the traditional neural network model into two parts, one of which is responsible for the forward state and the other for the backward state [11]. Compared with traditional RNN, LSTM and GRU, Bi-RNN, Bi-LSTM and Bi-GRU add a backward layer.

B. Explainable AI Algorithm

Explainable AI improves the transparency of black box models by making them understandable to humans, and calculating the contribution of features is now a popular and commonly used method. [10, 21–25]. In this paper, the Shapley Additive explanation, a game theoretically optimal Shapley value-based explanation of individual forecasts, proposed by Lundberg and Lee [17], is chosen to explain the existing forecasting models.

SHAP (SHapley Additive exPlanation) is an explanation method developed based on Shapley values [25, 26]. It uses cooperative games as the theoretical basis, treats the black box model as a “game”, and treats each feature as a “player”. Through the calculation of the Shapley value, we can determine the contribution degree of each player (feature) in the game process (black box model operation), and then know the contribution degree of each feature. The higher contribution, the greater influence of the feature on the forecast result.

IV. SIMULATION RESULTS

For calculations we used a computer with CPU—Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz; RAM—16.0GB; OS—Windows 10; for the implementation of the forecast models Python—Jupyter 3.8.3 was used.

A. Forecast Results

In this section, for reasons of space, only the results of the LiveLab dataset for the loss values (Fig. 1) is used as examples.

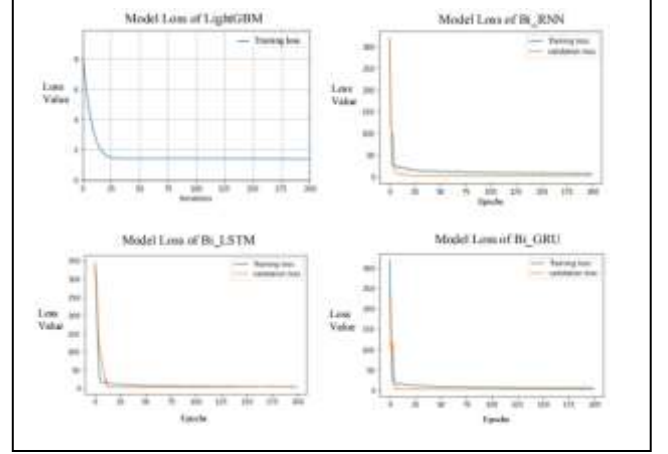


Fig. 1. LiveLab: Learning curves of forecast models. Note: The ensemble algorithm learning curve within the LightGBM model is RMSE, and the neural network is MSE

The learning curve for all models shows a decreasing trend. And given the need for a comprehensive comparison, a detailed comparison of quantitative metrics is required. Tables I, II and III show the results of all forecast models of the LiveLab, LSApp and Tsinghua datasets respectively. In this paper, performance is measured mainly from the perspective of variance, while considering the algorithm time. Training time refers to the training time of the algorithm, and inference time refers to the inference time of the algorithm.

Analysing Table I, it is easy to see that all the forecast models produce good performance on LiveLab, with Bi-RNN producing the best performance and LightGBM the shortest time.

TABLE I. FORECAST QUALITY OF LIVELAB

<i>LiveLab</i>	<i>R²</i>	<i>MSE</i>	<i>Training Time</i>	<i>Inference Time</i>
XGBoost	0.9693	1.6315	0.6951s	0.0100s
LightGBM	0.9740	1.5030	0.2044s	0.0050s
Random Forest	0.9753	1.4650	12.5185s	0.0322s
Bi-RNN	0.9819	1.2552	72.9370s	0.4707s
Bi-LSTM	0.9600	1.8628	80.3573s	1.4071s
Bi-GRU	0.9606	1.8499	112.5934s	1.1597s

Analyzing Table II, the LightGBM and Random Forest forecast models produced good performance on LASpp, and we can obtain that the neural network forecast models did not produce good performance relative to the ensemble models, with the ensemble model of Random Forest producing the best performance and the LightGBM having the shortest time.

TABLE II. FORECAST QUALITY OF LSAPP

<i>LSApp</i>	<i>R²</i>	<i>MSE</i>	<i>Training Time</i>	<i>Inference Time</i>
XGBoost	0.9704	0.6705	1.1090s	0.0080s
LightGBM	0.9606	0.7738	0.1745s	0.0050s
Random Forest	0.9730	0.6400	1.5499s	0.0499s
Bi-RNN	0.3804	3.1356	31.7785s	0.5216s
Bi-LSTM	0.4576	2.9339	53.7056s	1.3167s
Bi-GRU	0.5267	2.7408	33.0680s	1.1437s

Analyzing Table III, all the forecast models produced good performance on Tsinghua, with LightGBM producing the best performance and the shortest time.

TABLE III. FORECAST QUALITY OF TSINGHUA

<i>Tsinghua</i>	<i>R²</i>	<i>MSE</i>	<i>Training Time</i>	<i>Inference Time</i>
XGBoost	0.7719	62.8898	1.5765s	0.0100s
LightGBM	0.8155	56.5592	0.2692s	0.0050s
Random Forest	0.7519	65.5902	10.3956s	0.0469s
Bi-RNN	0.6736	75.2260	66.2006s	0.6293s
Bi-LSTM	0.6291	80.1927	58.4390s	1.8558s
Bi-GRU	0.6257	80.5618	98.6826s	1.0921s

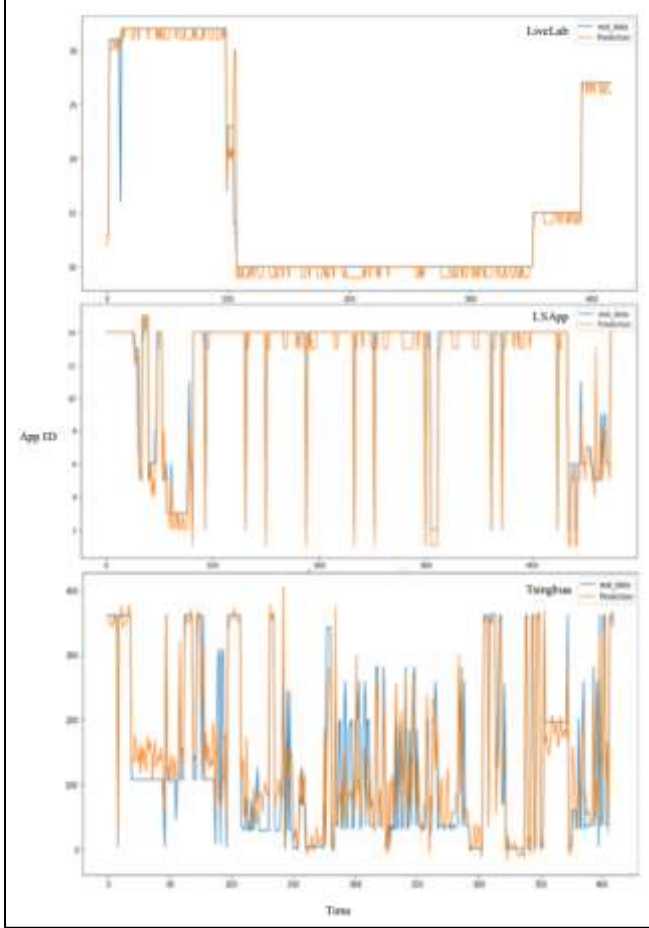


Fig. 2. Comparison of real and predicted values

Fig. 2 shows the comparison of true and predicted values for the three datasets (Fig. 2).

B. Explanation Results

Based on a combination of both variance and algorithm time, we chose to explain the results of the three datasets forecasted by the LightGBM-based algorithm using the SHAP method.

In Fig. 3 and Fig. 4, the transition from blue to red indicates an increase in the value of the feature, and the transition from top to bottom indicates a decrease in the impact of the feature on the model's forecast results.

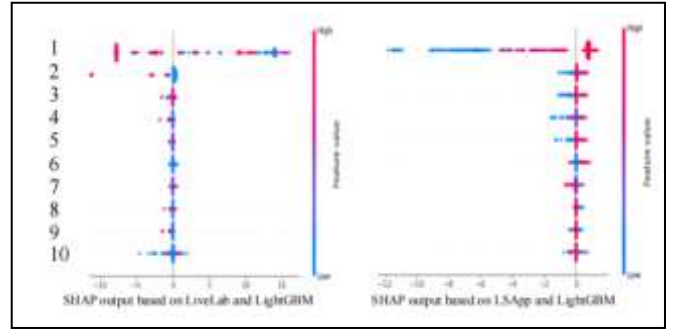


Fig. 3. SHAP output based on LiveLab, LSApp and LightGBM

In the left half of Fig. 3 and in Fig. 4, we conclude that “genre” and “Category ID” have the greatest impact on App forecast results, and in the right half of Fig. 3, we conclude that “lag App ID1” has the greatest impact on the App forecast results.

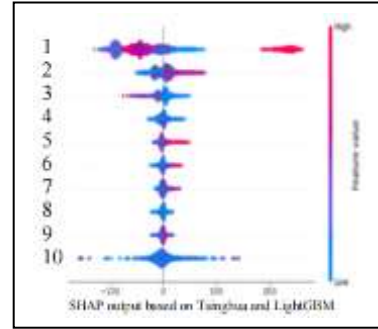


Fig. 4. SHAP output based on Tsinghua and LightGBM

V. CONCLUSION

In this paper, we use ensemble models and neural network models to forecast a time-series type of application usages based on operating system datasets. The prediction target is the app that the user will use at the next time point. For the prediction target, we selected six models (XGBoost, LightGBM, Random Forest, Bi-RNN, Bi-LSTM and Bi-GRU) in three different datasets (LiveLab, LSApp and Tsinghua) to make predictions. Through the analysis and comparison of forecast results, we consider LightGBM to be a better forecast model in the time series type of OS-based application usages datasets. We also gain insight into the chosen optimal forecast model at an explainable level, increasing the transparency of the black-box forecast models and helping users determine whether the features learned from the model are valid. Taking the explained results of this work as an example, we conclude that the category features and lag features of applications are important in predicting the next application user uses.

In the future, we will increase the part of traditional time series forecasting methods, compare and explain traditional time series forecasting methods, ensemble models, and neural network models, hoping to find a forecasting model that is more suitable for the forecasting problem of the user's next application, and increase the trust of the user's results.

REFERENCES

- [1] Nayar S. K., Narasimhan S. G. Vision in bad weather. Proceedings of the Seventh IEEE International Conference on Computer Vision IEEE. 1999, vol. 2, pp. 820–827.
- [2] Revels J., Tojib D., Tsarenko Y. Understanding consumer intention to use mobile services. Australasian Marketing Journal. 2010, vol. 18, no 2, pp. 74–80.

- [3] Tian Y., Zhou K., Pelleg D. What and How long: Prediction of Mobile App Engagement. *ACM Transactions on Information Systems*. 2022, vol. 40, no 8, pp. 1-38.
- [4] Cao H., Lin M. Mining smartphone data for app usage prediction and recommendations: A survey. *Pervasive and Mobile Computing*. 2017, vol. 37, pp. 1–22.
- [5] Liao Z. X., Lei P., Shen T., Li S., Peng W. Mining Temporal Profiles of Mobile Applications for Usage Prediction. *IEEE International Conference on Data Mining Workshops IEEE*. 2012, pp. 890–893.
- [6] Huang K., Zhang C. H., Ma X. X., Chen G. L. Predicting mobile application usage using contextual information. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. 2012, pp.1059–1065.
- [7] Nikolay Robinsonov, Gerhard Tutz, and Torsten Hothorn. Boosting techniques for nonlinear time series models. *AStA Advances in Statistical Analysis*, 96(1):99–122, 2012.
- [8] Atsushi Inoue and Lutz Kilian. How useful is bagging in forecasting economic time series? a case study of us consumer price inflation. *Journal of the American Statistical Association*, 103(482):511–522, 2008.
- [9] Georg Dorffner. *Neural networks for time series processing*. In *Neural network world*. Citeseer, 1996.
- [10] Zhang Y., Xu F., Zou J., Petrosian O., Krinkin K. XAI Evaluation: Evaluating Black-Box Model Explanations for Prediction. *Proceedings of 2021 2nd International Conference on Neural Networks and Neurotechnologies*. 2021, pp. 13–16.
- [11] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. *ACM*, 2016.
- [12] Ke G. L., Qi M., Thomas F., Wang T. F., Chen W., Ma W. D., Ye Q. W., Liu T. Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *NIPS*. 2017, vol. 30, pp. 74–80.
- [13] Tin K. H. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998, vol. 20, no 8, pp. 832–844.
- [14] Schuster M., Paliwal K.K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*. 1997, vol. 45, no 11, pp. 2673–2681.
- [15] Hochreiter S., Schmidhuber J. Long Short-Term Memory, *Neural Computation*. 1997, vol. 9, no 8, pp. 1735–1780.
- [16] Cho K., Merrienboer B. V., Gulcehre C., Bahdanau D., Bougares F., Schwenk H. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *Computer Science*. 2014, pp. 1724–1734.
- [17] Lundberg S. M., Lee S. I. A Unified Approach to Interpreting Model Predictions, *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 4768–4777.
- [18] Available at: <http://yecl.org/livelab/traces.html> (accessed 22 December 2021).
- [19] Available at: <https://github.com/aliannejadi/LSApp> (accessed 21 January 2022).
- [20] Available at: <http://fi.ee.tsinghua.edu.cn/appusage> (accessed 17 February 2022).
- [21] Das A., Rad P. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *CoRR*. 2020, vol. 2006.11371.
- [22] Zhang Y., Ma R., Liu J., Liu X., Petrosian O., Krinkin K. Comparison and explanation of forecasting algorithms for energy time series. *Mathematics*. 2021, vol. 9. no 21.
- [23] Arrieta A. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. *Information Fusion*. 2020, vol. 58, pp. 82-115.
- [24] Zou J., Xu F., Zhang Y., Petrosian O., Krinkin K. High-Dimensional Explainable AI for Cancer Detection, *International Journal of Artificial Intelligence*. 2021, vol. 19, no 2.
- [25] Zakharov V., Balykina Y., Petrosian O., Gao H. CBRR Model for Predicting the Dynamics of the COVID-19 Epidemic in Real Time. *Mathematics*. 2020, vol. 8, no 10.
- [26] Zou J., Petrosian O. Explainable AI: Using Shapley Value to Explain Complex Anomaly Detection ML-based Systems. *Machine Learning and Artificial Intelligence: Proceedings of MLIS 2020*. 2020, pp. 152–164.

Применение нейросетевого регулятора для многоканальной системы с нестационарными непараметрическими неопределенностями

А. А. Воевода

Новосибирский государственный технический
университет
voevoda@corp.nstu.ru

В. И. Шипагин

Новосибирский государственный технический
университет
shipagin@mail.ru

Аннотация. Нейросетевой подход к решению задачи синтеза системы автоматического управления показал свою эффективность для объектов, содержащих неопределенности различной природы. Однако использование такого подхода бывает затруднительно в следствии отсутствия формализованной методики по выбору: достаточной, но не избыточной архитектуры нейронной сети и начальных параметров весовых коэффициентов.

В данной работе приводится пример применения формализованной процедуры синтеза нейросетевого регулятора для системы автоматического управления, содержащую нестационарную непараметрическую неопределенность. Синтез нейросетевого регулятора происходит на основе информации о регуляторе, рассчитанном для линеаризованной модели объекта модальным методом. Эта процедура позволяет получить не избыточную архитектуру нейронной сети. Кроме этого, решается вопрос инициализации весовых коэффициентов нейронной сети.

В качестве примера работы алгоритма приведен случай решения задачи управления моделью перевернутого маятника на тележке с неопределенностью в канале управления. Данная задача является классическим тестовым примером многоканального неустойчивого объекта с неквадратной матричной передаточной функцией.

Кроме этого, приведен пример использования структурной схемы регулятора в матричном виде, позволяющем рассматривать многоканальные объекты в компактном виде. Предлагаются некоторые структурные преобразования нейросетевого регулятора, позволяющие упростить процедуру его тренировки. Полученный нейросетевой регулятор позволяет расширить возможности по управлению объектом, содержащем неопределенности различной природы.

Ключевые слова: нейросетевой регулятор, архитектура нейронной сети, инициализация весовых коэффициентов, перевернутый маятник на тележке, многоканальный объект, непараметрические неопределенности

I. ВВЕДЕНИЕ

Нейросетевой подход нашел широкое распространение для управления современными техническими системами [1–3]. Этот подход используется и для систем, содержащих неопределенности различной природы [4, 5]. Однако, согласно работам [6–8] данный подход не лишен некоторых трудностей. Актуальным остается вопрос поиска архитектуры нейронной сети. С одной стороны,

она должна быть достаточной сложности для решения поставленной задачи. А с другой стороны по возможности не излишне сложной, так как с новыми связями в архитектуре появляются дополнительные весовые коэффициенты, что усложняет задачу поиска их значений. Именно поэтому выработка формализованного подхода к задаче синтеза нейрорегулятора для многоканальных объектов (в том числе с неквадратной матричной передаточной функцией) является актуальной на сегодняшний день.

Существуют множество задач управления многоканальными неустойчивыми объектами. Особое место в них занимает задачи с неквадратной матричной передаточной функцией. К таким объектам можно отнести [9–11]. Не смотря на обилие таких объектов среди реальных технических систем задача управления «перевернутым маятником на тележке» представляется наиболее удобной для демонстрации работы алгоритма синтеза нейрорегулятора по нескольким причинам. Во-первых, эта эталонная задача управления многоканальными объектами в ТАУ с неквадратной матричной передаточной функцией. Во-вторых, из-за относительной простоты расчетов она позволяет, в отличии от других примеров, продемонстрировать вычислительный процесс работы алгоритма в рамках этой статьи. И наконец, в-третьих, существуют наработки по синтезу регуляторов для данного примера (авторы данной статьи принимали участие в расчете регулятора для системы «перевернутый маятник на тележке» методом полиномиально-матричного разложения системы PMFD – [12]), позволяющие не загромождать данную статью излишними расчетами, не относящимися к работе алгоритма.

В работе [13] приводился пример решения задачи управления нелинейной моделью перевернутого маятника. В данной же работе приводится пример управления линеаризованной моделью объекта, содержащей нестационарные непараметрические неопределенности в канале управления. Нейросетевой регулятор получен из обобщенного регулятора, взятого из статьи [12]. Для увеличения диапазона возможного управления системы с нестационарной непараметрической неопределенностью в своем составе необходимо усложнить структуру и настроить нейросетевой регулятор.

Работа состоит из четырех разделов. Второй раздел показывается пример синтеза нейросетевого регулятора для объекта перевернутый маятник на тележке по

указанному выше алгоритму. Третий раздел посвящен демонстрации полученных результатов.

II. ПРИМЕР РАБОТЫ АЛГОРИТМА СИНТЕЗА НЕЙРОСЕТЕВОГО РЕГУЛЯТОРА

В данной статье приведен алгоритм синтеза нейросетевого регулятора для системы, содержащей нестационарные непараметрические неопределенности. Синтез нейрорегулятора возможен при условии доступной информации о передаточной функции регулятора. В данной статье передаточная функция регулятора для линеаризованной модели объекта получена модальным методом, использующим полиномиальное матричное разложение системы *PMFD*. Более подробно о синтезе регуляторов с использованием этого алгоритма можно прочитать в работах [14–16].

A. Постановка задачи

В качестве примера работы алгоритма синтеза нейросетевого регулятора рассмотрим линеаризованную систему «перевернутого маятника на тележке». Математическая модель этой системы описана в статье [13]. Она описывается уравнениями:

$$(1 - \frac{ml}{M_t L})\ddot{s} + \frac{ml}{M_t L} g \theta = \frac{1}{M_t} u, \quad (1)$$

$$(1 - \frac{ml}{M_t L})\ddot{\theta} - \frac{g}{L} \theta = -\frac{1}{M_t L} u. \quad (2)$$

где l – длина стержня маятника; s – координата центра тяжести тележки (по горизонтальной оси – расстояние от центра тяжести); θ – отклонение маятника от вертикали; u – управляющее воздействие, $M_t = M + m$, M – масса тележки, m – масса маятника; $L = (I + ml^2)/ml$, I – момент инерции маятника относительно центра тяжести, $\alpha = \frac{M + m}{M}$, $a = \alpha ml g / M_t L$, $b = \alpha g / L$.

В дальнейшем будут использоваться следующие параметры объекта: $I = 0$, $m = 70 \text{ кг}$, $M = 30 \text{ кг}$, $l = 1 \text{ м}$, $g = 10 \text{ м/сек}^2$. Тогда $\alpha = 3.33$, $a = 23.3 \frac{\text{м}}{\text{сек}^2}$, $b = 33.3 \frac{1}{\text{сек}^2}$. Структурная схема линеаризованной модели объекта будет выглядеть следующим образом:

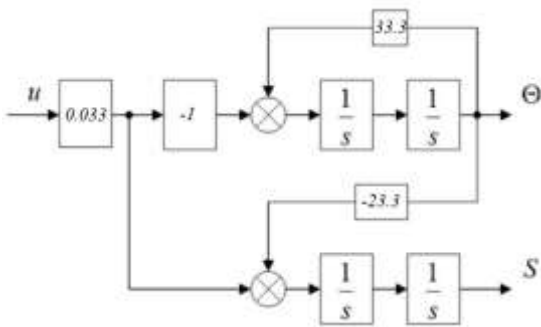


Рис. 1. Линеаризованная структурная схема перевернутого маятника на тележке

Как видно из структурной схемы на рис. 1 представлен многоканальный неустойчивый объект с неквадратной матричной передаточной функцией. Необходимо добиться астатизма по заданным выходным каналам. При этом по первому каналу (угол перевернутого маятника θ) будем решать задачу стабилизации системы $\theta = 0^\circ$.

B. Процедура синтеза регулятора

Регулятор расположим в прямой связи с объектом так, как показано на рис. 2.

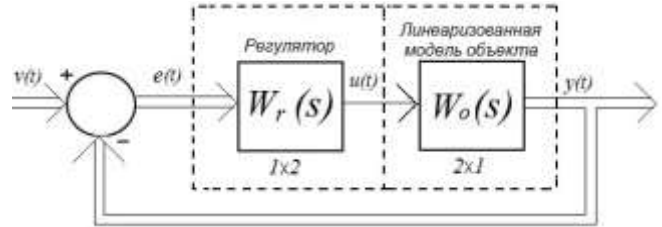


Рис. 2. Структурная схема САУ

где $u(t) \in \mathbf{R}^{1 \times 1}$ – управляющее воздействие, $e(t) \in \mathbf{R}^{2 \times 1}$, $e(t) = v(t) - y(t)$ – вектор «неувязок» между вектором заданий $v(t) \in \mathbf{R}^{2 \times 1}$, $v(t) = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ и выходом модели объекта $y(t) \in \mathbf{R}^{2 \times 1}$, $y(t) = \begin{pmatrix} \Theta(t) \\ S(t) \end{pmatrix}$. При этом запись $e(t) \in \mathbf{R}^{2 \times 1}$ обозначает, что вектор «неувязок» представляют из себя вектор вещественных чисел размерностью 2×2 .

В качестве регулятора, рассчитанного для линеаризованной модели перевернутого маятника на тележке, возьмем регулятор, полученный методом *PMFD* в статье [12]. Если в качестве желаемого характеристического матрицы системы зададим¹:

$$C(s) = \begin{pmatrix} (s+1)^5 & 0 \\ 0 & (s+1)^5 \end{pmatrix}, \quad (3)$$

то получим передаточную функцию регулятора:

$$W_{reg}(s) = \begin{pmatrix} \frac{-1327.3s - 5351.5}{s+5} & \frac{-15.2s - 3}{s+5} \end{pmatrix}. \quad (4)$$

Передаточная функция регулятора также неквадратной структуры. При этом у нее число входных каналов больше числа выходных каналов. Теперь можно приступить к алгоритму синтеза нейрорегулятора.

C. Синтез нейрорегулятора

На первом шаге переведем передаточную функцию регулятора (6) из непрерывного вида в дискретный. В качестве метода дискретизации использовался оператор “c2d” из Matlab. В качестве настроек выбирались метод

¹ В данной работе выбор полюса передаточной функции системы был произвольным с учетом расположения в левой полуплоскости комплексной плоскости. Вообще выбор полюсов передаточной функции системы достаточно обширная задача и ей посвящено достаточно большое число публикаций. Однако, в рамках данной работы не ставилось целью поднять вопрос выбора полюсов передаточной функции системы.

дискретизации “zero-order hold”. Приемлемый результат управления достигается при шаге дискретизации возьмем равном 0.01 сек.

$$W_{reg}(z) = \begin{pmatrix} \frac{-1327z + 1275}{z - 0.9512} & \frac{-15.15z + 15.12}{z - 0.9512} \end{pmatrix} \quad (5)$$

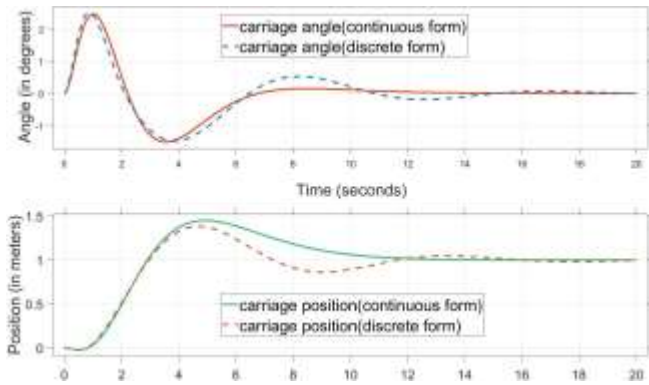


Рис. 3. Переходные процессы для системы с регулятором в непрерывном и дискретном виде

Затем проверяется адекватность полученного дискретного регулятора на линеаризованной модели объекта (1), (2). Как видно из рис. 3 при переходе к дискретной модели произошло понижение качества управления объектом, но в целом система остается устойчивой.

На шаге 2 необходимо представить передаточную функцию регулятора в структурном виде. Для этого необходимо провести некоторые преобразования уравнения (5). Подробнее об этом можно прочитать в работе [13]. Итак, представим передаточную функцию регулятора как отношение его выхода ко входу:

$$u(z) = \begin{pmatrix} \frac{-1327z + 1275}{z - 0.9512} & \frac{-15.15z + 15.12}{z - 0.9512} \end{pmatrix} e(z)$$

После определенных преобразований получим:

$$\begin{aligned} (u(z)(z - 0.9512) &= e(z)(-1327z + 1275) \\ u(z)(z - 0.9512) &= e(z)(-15.15z + 15.12)) \end{aligned}$$

Сократим это выражение на z . В левой части оставим только входное воздействие $u(z)$:

$$\begin{aligned} \left(u(z) &= e(z)(-1327 + 1275 \frac{1}{z}) + 0.9512u(z) \frac{1}{z} \right. \\ u(z) &= e(z)(-15.15 + 15.12 \frac{1}{z}) + 0.9512u(z) \frac{1}{z} \end{aligned}$$

Сформируем уравнения, содержащее наименьшее число звеньев запаздывания ($1/z$):

$$\begin{aligned} \left(u(z) &= -1327e(z) + (1275e(z) + 0.9512u(z)) \frac{1}{z} \right. \\ u(z) &= -15.15e(z) + (15.12e(z) + 0.9512u(z)) \frac{1}{z} \end{aligned}$$

На основании этих уравнений может быть представлена матричная форма структурной схемы.

Подобная форма записи использовалась в работах [5, 15] и позволяет представить многоканальные системы в компактном виде.

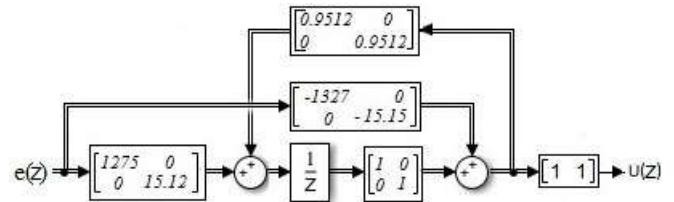


Рис. 4. Структурная схема регулятора в матричном виде

Данная структурная реализация должна содержать, по возможности, минимальное количество звеньев запаздывания в своей структуре. Это необходимо, чтобы избежать излишней сложности при настройке параметров нейрорегулятора в дальнейшем. Об этом сказано в работе [8].

На шаге 3 переходим к нейросетевому представлению структурной схемы регулятора. Более подробно о таком переходе можно прочитать в работе [13].

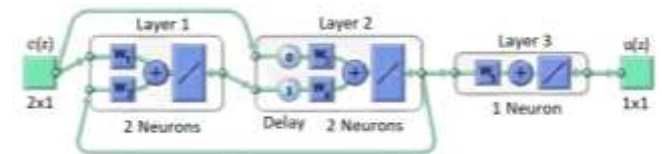


Рис. 5. Структурная схема нейрорегулятора

На рис. 5 представлена структурная схема нейрорегулятора. Она получена путем замены элементов структурной схемы регулятора (рис. 4) на элементы нейронной сети согласно таблице соответствия (таблица):

ТАБЛИЦА 1 ТАБЛИЦА СООТВЕТСТВИЯ ЭЛЕМЕНТОВ СТРУКТУРНОЙ СХЕМЫ РЕГУЛЯТОРА

Элемент структурной схемы	Элемент нейронной сети
Сумматор	Нейрон с линейной функцией активации
Коэффициенты усиления (элементы числовых матриц)	Весовые коэффициенты нейронной сети: $W_i \in \mathbf{R}^{2 \times 2}, i = \overline{1, 4}$ $W_5 \in \mathbf{R}^{1 \times 2}$
Элемент $1/z$	Звено запаздывания в канале нейронной сети

При этом разница в размерностях весовых коэффициентов связана с тем, что на выходе регулятора определен единственный скалярный выход.

Из рис. 5 видно, что нейронная сеть регулятора состоит из трех слоев. При переходе от первого ко второму слою имеется задержка во времени равная одному такту (один такт соответствует шагу дискретизации и равен 0.01 сек.). Третий слой нейронов был необходим для перехода от векторной величины к скалярной (рис. 4). На вход нейрорегулятора поступает информация о «неувязке» $e(t)$ между заданием системы $v(t)$ и выходом системы $y(t)$. На выходе формируется управляющее воздействие – $u(t)$. Оно представляет из

себя скалярную величину. Всего в нейрорегуляторе задействовано 5 нейронов.

Затем произведем некоторые структурные преобразования полученного нейрорегулятора. Это необходимо во избежание проблем, связанных с настройкой нейросетевого регулятора [8].

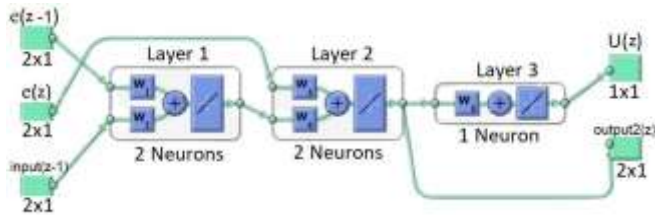


Рис. 6. Модифицированная структурная схема нейрорегулятора

Как видно из рис. 6 теперь структурная схема не содержит обратных связей. Кроме этого, звенья запаздывания также вынесены за пределы регулятора. Важно отметить, что структурные преобразования не изменяют характер работы регулятора. Система с преобразованным регулятором будет выглядеть следующим образом:

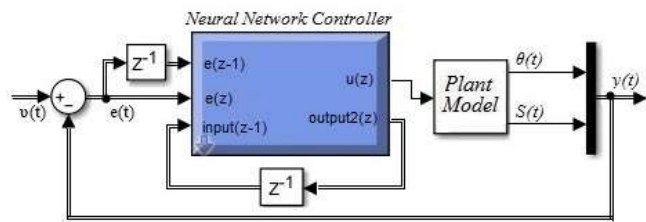


Рис. 7. Структурная схема системы с модифицированным нейрорегулятором

На шаге 4 усложним архитектуру нейросетевого регулятора. Это необходимо для введения в нейронную структуру нелинейностей и управления таким образом нелинейным объектом. Добавим слой (Layer 3) между вторым и третьим слоями. Этот слой будет состоять из 3 нейронов с функцией активации «сигмоида гиперболический тангенс – tansig». Изменим функции активации первого слоя с линейной также на «сигмоида гиперболический тангенс – tansig». Эта функция обладает всеми свойствами, необходимыми для нелинейности в нейронной сети: ограниченность ($y \rightarrow -1$ при $x \rightarrow -\infty$, и $y \rightarrow 1$ при $x \rightarrow \infty$), дифференцируемость на всём диапазоне определения, малые вычислительные затраты на определение производной [17]. Преобразования архитектуры нейросетевого регулятора можно увидеть на рис. 8.

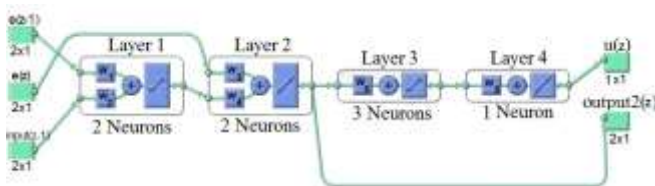


Рис. 8. Усложненная структурная схема нейрорегулятора

Затем подбираем весовые коэффициенты для вновь вводимых нейронов таким образом, чтобы его работа не влияла бы на устойчивость системы в целом. Этого можно добиться, если с помощью весовых коэффициентов добиться того, что аргумент нелинейной

функции «tansig» будет принимать значения в диапазоне $[-0.5, 0.5]$. Для этой задачи именно в этом диапазоне график функции «tansig» на «приемлемом уровне» аппроксимирует линейную функцию активации, которая использовалась до этого. Для каждой задачи диапазон возможного изменения аргумента нелинейной функции подбирается индивидуально. Приемлемость в данной задаче оценивалась по возможности регулятора «удерживать» систему в стабильном состоянии при небольших отклонения в задании на положение каретки перевернутого маятника.

На шаге 5 необходимо обучить полученный нейросетевой регулятор. Обучение будем проводить по эталону. В качестве эталона будет выступать работа дискретного регулятора (5). Для этой процедуры необходимо получить учебную выборку, состоящую из входящих и исходящих воздействий регулятора. Выборка собиралась с шагом дискретизации равной шагу дискретизации регулятора (5) и равна 0.01 секунде.

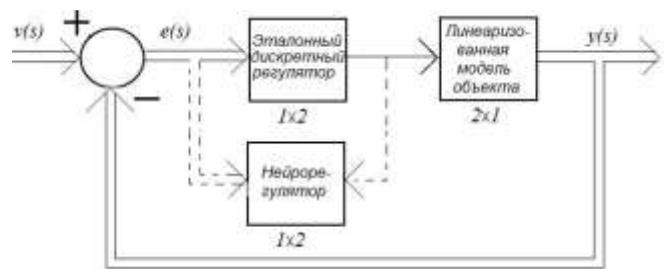


Рис. 9. Структурная схема САУ с эталоном и нейросетевым регулятором

Обучение в данной задаче проводилось алгоритмом Левенберга–Марквардта. Эффективность обучения нейронной сети вычислялась по среднеквадратичной нормализованной ошибке (MSE). Обучение проводилось с постепенным расширением задания по окончательному положению тележки v_2 . В общей сложности было проведено около 70000 эпох обучения нейронной сети. Однако предел возможностей по обучению не достигнут.

На шаге 6 в структурную схему системы, представленной на рис. 2, в канал управления добавляем нестационарную непараметрическую неопределенность с помощью мультипликативной помехи способом, указанным в [18].



Рис. 10. Структурная схема САУ с нестационарной непараметрической неопределенностью

Полученный нейросетевой регулятор применим для полученной системы. Результаты переходных процессов продемонстрированы в разделе 3 настоящей статьи.

III. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Моделирование работы системы проводилось в программном средстве Simulink Matlab. Выясним, какие преимущества дает использование нейросетевого регулятора. Для этого смоделируем работы систем, содержащих регулятор, полученный методом PMFD (4) – система 1 и нейросетевой регулятор – система 2. При этом в обеих системах будет использоваться система автоматического управления, содержащая нестационарные непараметрические неопределенности. Она была смоделирована с помощью блока «Uniform random number» в пакете Simulink Matlab (блок генерации случайных значений, распределенных по равномерному закону). Случайное значение генерировалось каждые 0.1 секунды.

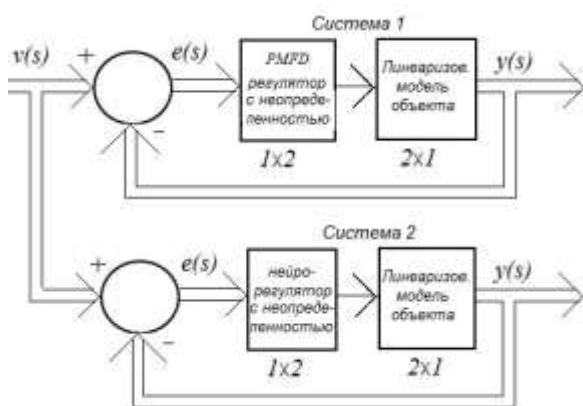


Рис. 11. Структурная схема систем с регулятором PMFD – система 1 и нейросетевым регулятором – система 2 для системы автоматического управления с неопределенностями в канале управления

Было выяснено, что в случае нахождения в канале управления нестационарной непараметрической неопределенности нейросетевой регулятор лучше справляется с задачей управления линейризованной моделью объекта. Так, например, если в качестве задания на перемещение тележки будет указано перемещение на 5 метров, тогда в случае нахождения неопределенности в канале управления пределах 40 % от номинального значения будем наблюдать следующие результаты:

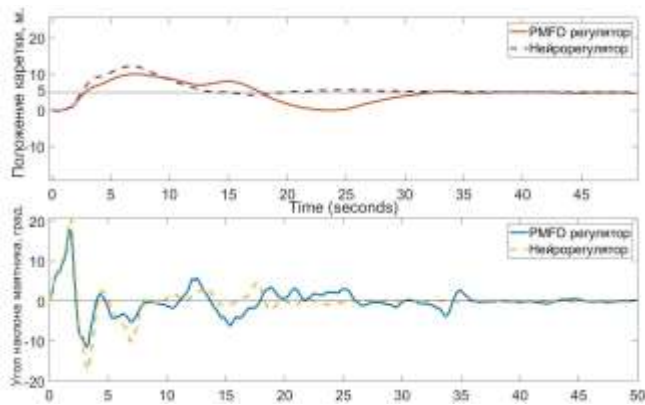


Рис. 12. Переходные процессы системы автоматического управления с неопределенностью в канале управления в пределах 40 %

Как видно из рис. 12 нейросетевой и PMFD регуляторы справляются с управлением системой, содержащей неопределенности в канале управления. Однако, в случае использования нейросетевого регулятора стабилизация системы происходит уже на

23 секунде, в отличие от PMFD регулятора (на 37 секунде).

Если повысить диапазон нестационарной непараметрической неопределенности и установить ее в пределах 45 % от номинального значения тогда получим следующие значения:

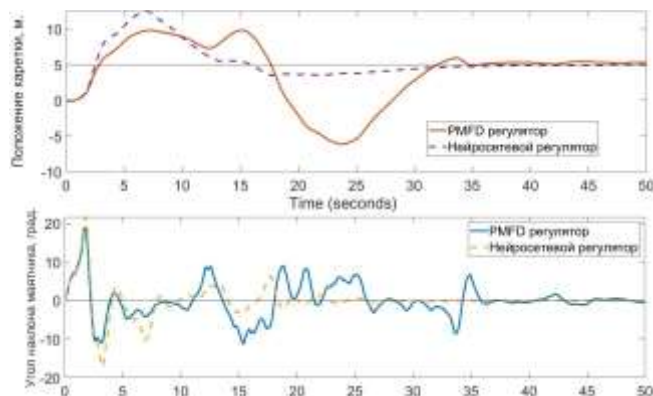


Рис. 13. Переходные процессы системы автоматического управления с неопределенностью в канале управления в пределах 45 %

Из рис. 13 видно, что в случае использования нейросетевого регулятора стабилизация системы происходит уже на 25 секунде, в отличие от PMFD регулятора (на 46 секунде).

То есть использование в системе автоматического управления нейросетевых регуляторов позволило повысить качество управления такой системой (в частности повысить ее быстродействие).

IV. ЗАКЛЮЧЕНИЕ

Для модальных методов синтеза регуляторов (к которым относится в частности и PMFD) учет неопределенностей различной природы в системе трудновыполним. Нейросетевой подход к синтезу регуляторов показал свою актуальность в случае управления системой, содержащей неопределенности различной природы. Однако и такой подход не лишен некоторых недостатков. К ним можно отнести отсутствие формализованного подхода: к выбору архитектуры нейронной сети и к инициализации начальных весовых коэффициентов. При этом неправильный выбор этих параметров может повлиять на качество найденного решения.

В данной работе предлагается алгоритм, на основании которого решается вопрос выбора архитектуры и начальных весовых коэффициентов для нейронной сети нейросетевых регуляторов. Работа предложенного алгоритма демонстрируется на примере эталонной задачи многоканального неустойчивого объекта «перевернутый маятник на тележке». В результате работы алгоритма удается получить нейросетевых регуляторов. Его возможности по управлению системой, содержащей неопределенности в канале управления возрастают по сравнению с регулятором, полученным методом PMFD для линейризованной модели объекта.

СПИСОК ЛИТЕРАТУРЫ

- [1] Епихин А.И. Принципы нейроуправления и варианты архитектуры нейронных сетей, применительно к сложной динамической системе сзу-судно /Епихин А.И., Хекерт Е.В.,

- Модина М.А. // Морские интеллектуальные технологии. 2020. № 4-4 (50). С. 18-22.
- [2] Дмитриенко В.Д., Заковоротный А.Ю. Решение проблемы дообучения классических нейронных сетей // Автоматизир. технологии и производства. 2015. № 4 (10). С. 32–40.
- [3] Белов М.П., Носиров И.С., Фьюнг Х.Ч. Исследование системы управления электроприводом подачи токарного станка с применением метода обобщенного нейроуправления с прогнозированием // Изв. СПбГЭТУ «ЛЭТИ». 2018. № 4. С. 73–80.
- [4] Пшихопов В.Х., Шанин Д.А., Медведев М.Ю., Чикин В.В. Применение нейросетевого адаптивного регулятора для управления объектом с неопределенностью // Известия ЮФУ. Технические науки. 2008. №7
- [5] Мухутдинов Аглям Рашидович, Марченко Герман Николаевич Повышение стабильности процесса горения твердого топлива на тепловых электрических станциях // Известия вузов. Проблемы энергетики. 2009. №7-8.
- [6] Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. ICML (3), 28, 1310-1318.
- [7] Бурнаев Е.В., Ерофеев П.Д. Влияние инициализации параметров на время обучения и точность нелинейной регрессионной модели // Информационные процессы, Том 15, № 3, 2015, стр. 279–297.
- [8] Воевода А.А., Шипагин В.И. Структурные преобразования нейросетевого регулятора с рекуррентным типом сети // Сборник научных трудов НГТУ. 2020. № 3 (98). С. 7–16. DOI: 10.17212/2307-6879-2020-3-7-16. 09.2021
- [9] Bian, Z. Kinematic analysis and simulation of 6-DOF industrial robot capable of picking up die-casting products [Text] // Z. Bian, Z. Ye and W. Mu/ 2016 IEEE International. Conference on Aircraft Utility Systems (AUS), 2016, 41-44 pp.
- [10] Chen Zh. Balancing control of a cubical robot balancing on its corner [Text] / Zhigang Chen, Xiaogang Ruan, Yuan Li // 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC), 2018, 631-636 pp.
- [11] Dai H. Planning robust walking motion on uneven terrain via convex optimization [Text] // H. Dai, R. Tedrake / 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), 2016, 579-586 pp.
- [12] Воевода А.А., Филюшов В.Ю., Шипагин В.И. Полиномиальный метод синтеза регуляторов для частного случая многоканальных объектов с одной входной переменной и несколькими выходными // Безопасность цифровых технологий. 2021. № 3 (102). С. 21–42. DOI: 10.17212/2782-2230-2021-3-21-42
- [13] Воевода А.А., Шипагин В.И. Синтез нейросетевого регулятора управления нелинейной моделью перевернутого маятника на тележке // Научный вестник НГТУ. 2020. № 2–3 (79). С. 25–36. DOI: 10.17212/1814-1196-2020-2-3-25-36.
- [14] Antsaklis, P. J. Linear systems [Text] / P. J. Antsaklis, A. N. Michel. Switzerland: Birkhauser, 1997. 669 p.
- [15] Бобобеков К.М. Полиномиальный метод синтеза многоканальных регуляторов с использованием матрицы Сильвестра [Текст]: дисс. ... канд. техн. наук: 05.13.01 С-Петербург, 2019. 168 стр.
- [16] Chen C.T. Linear System Theory and Design [Text] / C.T. Chen. 2ed. New York :Oxford, 1999. 334 p.
- [17] Кобзев А.А., Лекарева А.В., Сидорова О.С. Анализ алгоритмов обучения нейронной сети // Современные наукоемкие технологии. 2021. № 6-1. С. 23-28
- [18] Целигоров Н.А. Математические модели неопределённостей систем управления и методы, используемые для их исследования / Целигоров, Н.А., Целигорова Е.Н., Мафуря Г.М. // ИВД. 2012. № 4-2.

Система детектирования и анализа объектов на КТ-снимках в урологии

М. А. Руденко¹, А. В. Руденко², М. А. Крапивина³, В. С. Лисовский⁴

Крымский федеральный университет им. В.И. Вернадского

¹rudenko.ma@cfuv.ru, ²rudenkoandre@mail.ru, ³miladakrapivina@gmail.com, ⁴vllisovsky@mail.ru

Аннотация. В статье приводится анализ возможностей методов искусственного интеллекта для решения задачи нахождения и оценки объектов на снимках компьютерной томографии и рентгенографии с помощью методов глубокого машинного обучения.

Ключевые слова: компьютерное зрение, искусственная нейронная сеть, глубокое обучение, распознавание объектов, сверточная нейронная сеть, детектирование объектов

I. ВВЕДЕНИЕ

Одним из самых перспективных направлений использования искусственного интеллекта (ИИ) является применение в медицине, причем технологии ИИ приносят не только теоретическую пользу в ряде различных научных исследований, но и практическую. Применение ИИ дает возможность разработки новых, более эффективных подходов к анализу многомерной информации, например, при распознавании медицинских изображений. Это имеет большое значение при заболеваниях, диагностика которых основывается на исследовании медицинских изображений. К данному типу заболеваний относится мочекаменная болезнь, при которой диагноз чаще всего ставится по результатам проведения исследований методами рентгенографии и компьютерной томографии.

В данной статье будет рассмотрено применение компьютерного зрения (КЗ) в урологии, а именно детектирование и распознавание объектов свёрточными нейронными сетями (СНС) на снимках компьютерной томографии для проведения диагностики и лечения мочекаменной болезни.

Просмотр изображений, поиск, локализация почечных камней и ручная разметка снимков КТ сами по себе являются довольно сложными и трудоемкими процедурами, требуют значительных временных затрат специалистов и часто осложняются неполнотой и неточностью исходных данных (наличие различных видов искажений в снимках). Использование методов КЗ для решения этих задач имеет ряд преимуществ и, главным образом, позволит ускорить процесс анализа медицинских снимков, снизить вероятность и риск ошибочных выводов и оптимизировать нагрузку на медицинских сотрудников.

Целью исследования является создание автоматизированной системы поддержки принятия врачебных решений (СПВР) в урологии на основе

применения технологий компьютерного зрения для анализа и оценки результатов КТ.

Процесс поддержки принятия решений предполагает разработку нейросетевых моделей для обнаружения заданного класса объектов, а также системы нечеткого вывода об уровне принадлежности объекта классу. При планировании хирургического вмешательства для обнаруженных конкрементов необходимо вычислить характеристики, такие как размер, объем, масса, распределение плотности. СПВР должна сформировать заключения в понятном для врача виде.

Главной задачей, которая определяет эффективность разрабатываемой СПВР, является создание нейронной сети распознавания заданных классов объектов. Точность и адекватность нейросетевой модели зависит, в первую очередь, от выбранной технологии и качества обучения.

Современное развитие технологий компьютерного зрения предоставляет разработчику целый ряд библиотек и решений для различных платформ и на разных языках программирования. Выбор конкретного решения зависит от вида и характеристик изображений и объектов, которые необходимо обнаружить. Сложным этапом является обучение нейронной сети, которое требует привлечения специалистов-медиков для разметки изображений для обучения нейронной сети, а также формирование вывода и заключения относительно объекта, диагноза и пути лечения.

II. ОПИСАНИЕ МЕТОДИКИ

Компьютерная томография – медицинское рентгенологическое исследование, основанное на послойном исследовании структуры внутренних органов и систем, позволяющее получить рентгеновское изображение внутренних органов [1, 4]. Рентгеновские лучи проходят сквозь тело человека, а компьютерная система формирует изображение как поперечных слоев (срезов), так и продольных. Результаты этой процедуры сохраняются в виде DICOM файлов. DICOM (Digital Imaging and Communications in Medicine) – объектный файл с теговой организацией для представления кадра изображения (или серии кадров) и сопровождающей информации (в виде DICOM тегов), который имеет 4-х ступенчатую структуру: пациент (patient) → исследование (study) → серия (series) → изображение (кадр или серия кадров, image) [4].

Таким образом, в результате проведения исследования врач-специалист получает большой набор изображений срезов внутренних органов по одному пациенту. Далее, с помощью специализированного программного обеспечения он просматривает

Финансирование исследования проводится ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере», программа «Старт», конкурс «Старт-Искусственный интеллект-1» (I очередь), заявка С1ИИ-112266, договор № 27ГС1ИИИС12-D7/71365

полученные срезы для установки точной локализации, определения формы и размеров конкрементов, производит разметку среза. Размеченный срез представляет из себя собственно изображение и текстовый документ, содержащий информацию о выделенных объектах на данном изображении: их класс, координаты на срезе и размеры. После разметки срезов специалист определяет параметры объектов – размеры и плотность почек и камней.

Очевидны пути оптимизации этого процесса с помощью нейронной сети. Она должна получать изображения из DICOM файлов, распознавать их, детектировать на изображении объекты заданных классов, оценивать результаты распознавания и рассчитывать параметры (размеры, плотность, координаты) заданных медицинских объектов для дальнейшей 3D-визуализации объектов с картированием по плотности найденных объектов, характеризующих выбранную патологию, с целью поддержки процессов планирования хирургических вмешательств в области урологии.

III. ПРИМЕНЕНИЕ СНС ДЛЯ АНАЛИЗА КТ

Поиск объектов на медицинских изображениях является одной из важнейших задач при дифференциальной диагностике патологий на КТ. Задача поиска объектов на КТ снимках обычно состоит из двух подзадач: первая из них относится к классу задач обнаружения (детектирования) объекта, принадлежащего одному из заданных классов на срезе; вторая – в оценке принадлежности найденного объекта одному из заданных классов.

До недавнего времени задачи детектирования решались с использованием различных алгоритмов обработки изображений, в том числе: алгоритмов адаптивной фильтрации и усиления, алгоритмов, основанных на использовании гистограмм градиентов, алгоритмов каскадных классификаторов на основе метода Виолы-Джонса, алгоритмы, основанные на методах контурного анализа и др. [3].

Решения, основанные на обычных нейронных сетях с предварительным выделением признаков классификации объектов, а также стандартных глубоких сверточных нейронных сетей, применяемых в режиме просмотра последовательно перебираемых в скользящем режиме окон, в которых проводится анализ [2]. Следует отметить, что такой подход является весьма затратным в вычислительном отношении, несмотря на все преимущества, которые изначально предоставляют нейронные сети с точки зрения точности классификации.

В последнее время для решения данной задачи широкое распространение получили алгоритмы, основанные на применении региональных глубоких сверточных нейронных сетей (Regional Convolutional Neural Networks, R-CNN) и сетей детектирования YOLO (You Only Look Once), которые принципиально ориентированы на решение задачи поиска объектов с одновременной их классификацией. По сравнению с методами, представленными выше, эти алгоритмы принципиально адаптированы для решения задачи поиска объектов на изображениях. Исходная реализация R-CNN базируется на использовании специальных алгоритмов предобработки – алгоритмов region-proposal-

function, обеспечивающих предложение так называемых областей внимания, в которых потенциально могут находиться интересующие нас объекты. Такой «специализированный» подход предлагает сократить вычислительные затраты, а также позволяют добиться минимального времени определения местоположения объекта и высокой точности его классификация. К настоящему моменту имеется большое количество вариантов реализации подобных алгоритмов, которые достигли хороших показателей по данным критериям [3, 6, 8].

Многие алгоритмы решения задачи детекции объектов на изображениях с помощью СНС заключается в разбиении изображения на квадратные области, затем классификация этих областей на наличие объекта и классификация самого объекта. Таким образом, изображение просматривается дважды (один раз для определения областей, где есть объект, второй — для классификации этого объекта.) Этот способ работает долго и требует больших затрат вычислительных мощностей.

YOLO же использует другой принцип. Исходное изображение сжимается таким образом, чтобы получить квадратную матрицу размером 13 на 13, в каждой клетке которой записана информация о наличии объекта и классе этого объекта на соответствующей части картинки. Таким образом, YOLO просматривает картинку за один проход, что существенно увеличивает скорость обработки изображения. Данные характеристики позволяют использовать YOLO для задач распознавания объектов в реальном времени и на портативных устройствах. Отличительной особенностью YOLO является подход к решению задачи детекции, который предполагает однократный проход CNN для прогнозирования положения объекта и классификации нескольких кандидатов. Таким образом, достигается сквозное обнаружение всех классов объектов. По результатам исследований [5, 8] сети YOLO являются наиболее эффективными по скорости и точности из всех нейронных сетей для детектирования объектов.

Для решения задачи диагностики в СПБР была выбрана архитектура YOLO, которая дает лучшие результаты в области детектирования объектов на изображениях.

В настоящее время последней версией СНС архитектуры YOLO является YOLOv5, реализованная на фреймворке PyTorch, скорость распознавания которой составляет 140 кадров в секунду. YOLOv5 включает в себя несколько моделей, которые отличаются по количеству слоев и размеру, скоростью и точностью детектирования: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x [2, 8].

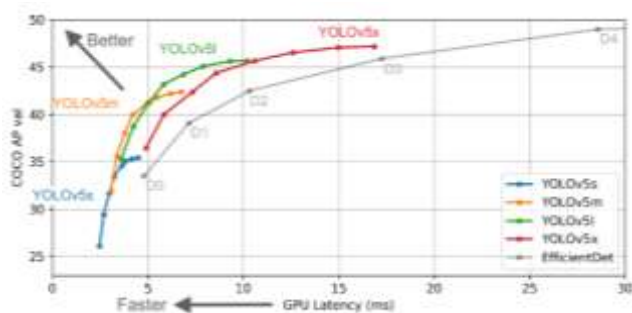


Рис. 1. Сравнение версий моделей YOLOv5

Использование сетей YOLO предполагает обучение на размеченных изображениях, которые представляют выборку заданных классов. Разметка изображений требует знаний специалиста в области хирургии и урологии. Это классифицируется как обучение с учителем, так как предполагается использование полного набора аннотированных данных (размеченного обучающего датасета) для тренировки системы ИИ на всех стадиях ее создания. Разработчиком YOLOv5 для получения хороших результатов обучения рекомендовано более 1500 изображений объектов одного класса, более 10000 размеченных экземпляров объектов для каждого класса [7].

В исследуемой области получить необходимое количество изображений для создания датасета нужного размера в медицине бывает проблематичным. Несмотря на то, что КТ в перспективе может давать множество материала для обучения (к примеру, толщина среза у современных 16-х срезовых аппаратов может достигать 0,75–0,6 мм, обучение проходит с использованием изображений, размеченных специалистами в предметной области, а это очень трудоемкий и долгий процесс.

Для дообучения нейросети и повышения точности детектирования можно использовать датасет, созданный на основе данных, которые были получены при первичном детектировании самой нейросетью и которые были проверены и откорректированы специалистом. Таким образом происходит оптимизация процесса дополнения обучающей выборки данными исследований новых пациентов.

На начальных этапах работы было использовано увеличение размера обучающего датасета путем клонирования исходных изображений с использованием фильтрации изображений – усилением резкости изображения (sharping), а также путем дублирования файлов меток, содержащих разметки для изображений, на ближайшие соседние к ним.

Применение этих методов тиражирования изображений, хотя и ведут к снижению уникальности данных в датасете, оправданы на этапе создания обучающего датасета. В клинической практике с КТ сопутствуют различные искажения, такие как засвеченное изображение, наличие посторонних предметов, кроме того, вероятны проведения процедуры без контрастного вещества и предварительной подготовки пациента, что приводит к ухудшению четкости изображений. Что касается дублирования меток на соседние срезы – расстояние между ними, как правило, достаточно мало, чтобы это привело к возникновению существенной ошибки.

IV. МОДЕЛИ

Применяемая классификация объектов, которые могут быть детектированы нейросетью на изображениях КТ, содержит следующие классы:

- stone,
- left_kidney,
- right_kidney,
- left_kidney_pieloectasy,
- right_kidney_pieloectasy,
- staghorn_stones.

Для детектирования объектов была выбрана модель YOLOv5s, поскольку она содержит небольшое количество слоев, имеет достаточную скорость распознавания, файл весов имеет размер 14 МБ.

Первая версия модели YOLOv5s была обучена на датасете, содержащем 1586 изображений, в том числе 793 оригинальных изображений КТ, полученных из КТ у 21 пациента, 793 – обработанных изображений. Обучение проводилось 1200 эпох. Распределение изображений по классам неравномерно: классы left_kidney, stone, right_kidney значительно популярнее (рис. 2).

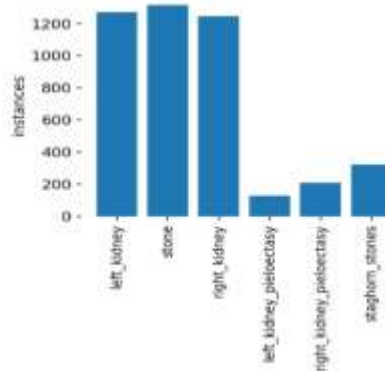


Рис. 2. Распределение классов первой версии обученной модели нейросети

Наблюдается прямая зависимость между количеством встреч объектов одного класса и точностью их распознавания. Тем не менее, достаточно высокая точность распознавания наблюдаются для всех классов объектов.

Наибольшую точность модель показала при детектировании почек (рис. 3). Интересно также то, что самый распространенный класс – «stones» имеет одну из самых низких точностей, что объясняется асимметрией обучающего датасета.

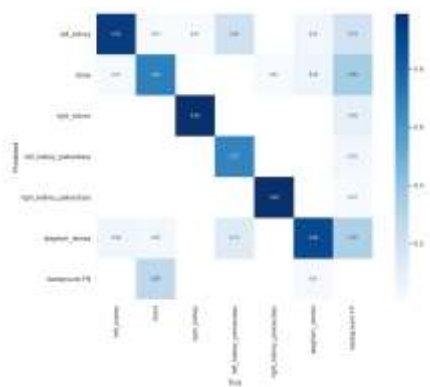


Рис. 3. Результаты обучения первой модели

Исходя из результатов первой модели, можно сделать вывод, что для повышения точности распознавания камней необходимо учитывать их локализацию. Это можно сделать путем добавления дополнительных алгоритмов оценки результатов распознавания объектов. Авторами был разработан и внедрен в процесс детектирования нейросетью алгоритм оценки правдоподобия распознавания, который «отсекает» детектированные объекты «неправильной» локализации. В результате удалось добиться того, что объекты классов камней «stone» и «staghorn_stones» детектируются только в пределах объектов классов, принадлежащих почкам: правой и левой.

Вторая версия модели была обучена на датасете, содержащем 2075 изображений, в том числе 1582 оригинальных изображений, полученных из КТ 24 пациентов, остальные изображения были получены путем изменения оригиналов добавлением резкости. Обучение проводилось на 600 эпохах. Несмотря на увеличение количества снимков по сравнению с предыдущей моделью, соотношение классов не изменилось (рис. 4).

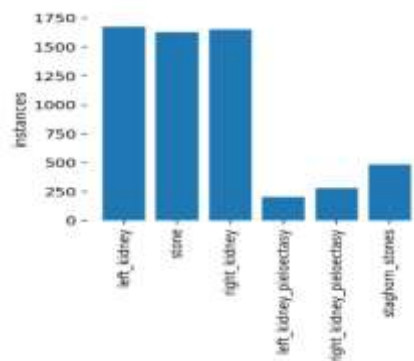


Рис. 4. Распределение классов второй версии обученной модели нейросети

Тем не менее, очевиден прогресс и значительное увеличение точности работы нейросети (рис. 5).

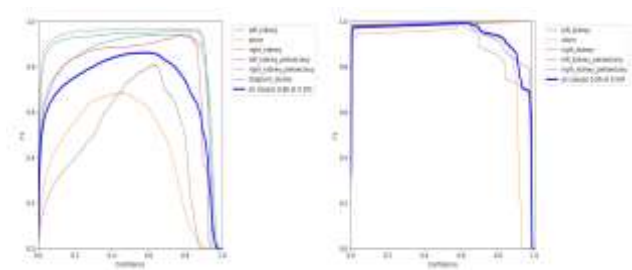


Рис. 5. Сравнение точностей первой и второй моделей

Вторая модель с высокой точностью определяет все заданные классы объектов (рис. 6).

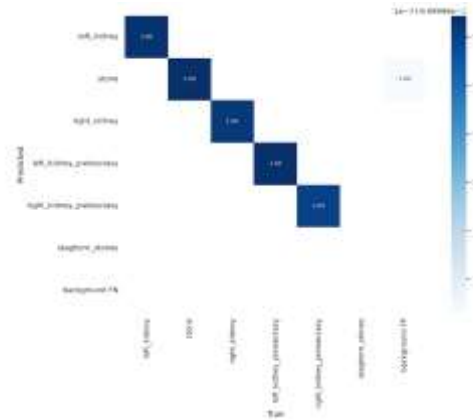


Рис. 6. Результаты обучения второй модели

Полученная модель нейронной сети для детектирования объектов на снимках КТ для проведения диагностик мочекаменной болезни была апробирована на базе Клинического медицинского многопрофильного центра имени Святителя Луки г. Симферополь. На рис. 7 приведен результат детектирования с помощью нейросети.



Рис. 7. Результат детектирования объектов на изображении КТ

Разработанная система детектирования и анализа объектов на КТ-снимках в урологии показала свою эффективность на объектах всех классов в процессе диагностирования и планировании хирургических вмешательств.

V. ЗАКЛЮЧЕНИЕ

Результаты исследования показали высокую эффективность применения искусственных нейронных сетей архитектуры YOLOv5 для детектирования заданных объектов на медицинских изображениях. Решение данных задач невозможно без широкого использования сверточных нейронных сетей. Однако наибольшую ценность представляют выводы, построенные на комплексной оценке результатов распознавания как всего изображения, так и отдельных объектов с помощью детектирования с использованием алгоритма оценки правдоподобия распознавания.

СПИСОК ЛИТЕРАТУРЫ

- [1] Борисов Д.Н., Кульнев С.В., Лемешкин Р.Н. Использование искусственного интеллекта при анализе цифровых диагностических изображений // Состояние и перспективы развития современной науки по направлению "ТЕХНИЧЕСКОЕ ЗРЕНИЕ И РАСПОЗНАВАНИЕ ОБРАЗОВ". 2019. С. 163-169 URL: <https://www.elibrary.ru/item.asp?id=41824272> (дата обращения: 28.04.2022).
- [2] Руденко М.А., Руденко А.В. Нечеткая модель классификации медицинских изображений на основе нейронных сетей // XXIV Международная конференция по мягким вычислениям и измерениям (SCM-2021). Сборник докладов. Санкт-Петербург. 26 – 28 мая 2021 г. СПб.: СПбГЭТУ «ЛЭТИ». 351 с. ISBN 978-5-7629-2864-92
- [3] Сирота А.А., Митрофанова Е.Ю., & Милованова А.И. (2019). Анализ алгоритмов поиска объектов на изображениях с использованием различных модификаций сверточных нейронных сетей. // Вестник ВГУ. Серия: Системный анализ и информационные технологии, (3), 123-137. <https://doi.org/10.17308/sait.2019.3/1313>
- [4] Хамад Ю.А., Симонов К.В. Программа для обработки и анализа изображений компьютерной томографии и рентгенографии грудной клетки / Федеральное государственное автономное образовательное учреждение высшего образования «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» (СФУ) 2020. URL: <https://www.elibrary.ru/item.asp?id=43965796> (дата обращения: 28.04.2022).
- [5] Priya, Dwivedi YOLOv5 compared to Faster RCNN. Who wins? / Dwivedi Priya. — Текст : электронный // towardsdatascience.com : [сайт]. — URL: <https://towardsdatascience.com/yolov5-compared-to-faster-rcnn-who-wins-a771cd6c9fb4> (дата обращения: 28.04.2022).
- [6] Srivastava S., Divekar A.V., Anilkumar C. et al. Comparative analysis of deep learning image detection algorithms. J Big Data 8, 66 (2021). <https://doi.org/10.1186/s40537-021-00434-w>
- [7] Tips for Best Training Results. — Текст : электронный // docs.ultralytics.com: [сайт]. — URL: <https://docs.ultralytics.com/tutorials/training-tips-best-results/> (дата обращения: 28.04.2022).
- [8] YOLOv5: state-of-the-art модель для распознавания объектов. — Текст: электронный // neurohive.io: [сайт]. — URL: <https://neurohive.io/ru/papers/yolov5-state-of-the-art-model-dlya-raspoznavaniya-obektov/> (дата обращения: 28.04.2022).

Анализ архитектур умножителей для аппаратной реализации нейронных сетей

С. Э. Миронов¹, О. И. Буренева², А. Д. Милакин³

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

¹semironov@etu.ru, ²oibureneva@etu.ru, ³milakins.s98@gmail.com

Аннотация. Нейронные сети обеспечивают высокую производительность при решении разного типа задач, при этом их сложность растет с увеличением сложности приложения и требует высокой вычислительной мощности. Решение проблемы производительности лежит в области создания аппаратных средств для реализации нейронных сетей в виде заказных интегральных микросхем. Основными элементами нейронной сети являются блоки хранения данных, сумматоры и умножители. В статье рассматриваются варианты ускорения работы умножителей с использованием методов умножения на группу разрядов, предложены варианты топологических реализаций рассмотренных решений, позволяющие достичь компромисса между быстродействием и площадью кристалла.

Ключевые слова: элементы нейронных сетей, аппаратная реализация нейронных сетей, умножители, умножение на группу разрядов, алгоритм Бута

I. ВВЕДЕНИЕ

Аппаратные решения играют все более важную роль в области искусственного интеллекта, поскольку позволяют существенно увеличить быстродействие нейронных сетей по сравнению с их программными аналогами [1]. При построении высоко параллельных архитектур нейронных сетей (НС) и алгоритмов машинного обучения применяются программируемые логические интегральные схемы (ПЛИС). При этом разрабатываются как подходы к реализации на базе ПЛИС традиционных архитектур нейронных сетей [2, 3], так и методы адаптации элементов нейронных сетей к структурам типовых ячеек ПЛИС для получения максимально эффективного результата [4]. Несмотря на широкие возможности ПЛИС при их использовании остается ряд ограничений, связанных с требованиями по соотношению ресурсов: для реализации нейронных сетей требуются FPGA с большим количеством встроенных умножителей и блочной памяти. При этом модификация специализированных ресурсов ПЛИС невозможна, поскольку арифметические блоки имеют фиксированную архитектуру с минимальными возможностями настройки. Очевидно, что аппаратные решения в области интегральных схем для нейронных сетей связаны с проектированием специализированных интегральных схем.

Основными элементами нейронной сети являются блоки хранения данных, сумматоры, умножители и вычислители функции активации, на основе которых строятся нейроны, объединяющиеся в слои, которые в дальнейшем собираются в сеть. Для построения быстродей-

ствующих НС требуется разработка быстродействующих элементов, что может быть достигнуто путем разработки аппаратных ускорителей [5, 6] или применения нетрадиционных алгоритмов выполнения соответствующих операций [7, 8].

Для ускорения операции умножения существует ряд оригинальных алгоритмов, которые могут быть реализованы аппаратно. В статье мы показываем варианты топологической реализации алгоритмов и методов быстрого умножения в виде макроблоков интегральных схем.

II. МЕТОДЫ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ УМНОЖИТЕЛЕЙ

Умножение обычно выполняют путем сложения частичных произведений, являющихся простыми кратными множимому. Время выполнения умножения пропорционально количеству сложений. В связи с этим для повышения быстродействия умножителей очень важной задачей является уменьшение числа операций сложения. Существует два подхода к ее решению: уменьшение количества частичных произведений и снижение числа этапов суммирования.

A. Уменьшение количества частичных произведений

Уменьшения количества частичных произведений добиваются, объединяя в группы разряды множителя Y и умножая множимое X уже не на один разряд, а на группу разрядов числа Y одновременно. Это требует использования чисел, кратных операнду X . Из таких чисел на каждой стадии умножения выбирается одно: то, отношение которого к операнду X равно сумме разрядов рассматриваемой группы. Схемы, выполняющие выбор требуемого числа, представляют собой мультиплексоры. С помощью различных вариантов кодирования групп разрядов, стараются уменьшать количество кратных множимому X чисел. Этого можно достичь, используя числа, равные « $X \times 2^q$ », (X - множимое, q - некоторое положительное целое число), так как получить такие числа можно не путем вычислений, а простым сдвигом на q разрядов влево множимого X (или его инверсии).

Одним из подобных способов является алгоритм Бута умножения на два разряда. В соответствии с этим алгоритмом сумме пары разрядов множителя B и переноса из соседней младшей пары разрядов ставится в соответствие некоторое число со знаком («-2», «-1», «0», «+1», «+2») и перенос в соседнюю старшую пару разрядов.

На рис. 1 приведена топология матричного умножителя Бута разрядностью 16×16 , полученная с помощью параметризованных по разрядности средств кремниевой компиляции на основе системы иерархического сжатия

Данное исследование было профинансировано «Программой развития ЭТУ «ЛЭТИ» в рамках программы стратегического академического лидерства» Приоритет-2030 № 075-15-2021-1318 29 сентября 2021 года.

топологии «*Matching of cell*», разработанной на кафедре вычислительной техники СПб ГЭТУ «ЛЭТИ» [9, 10].

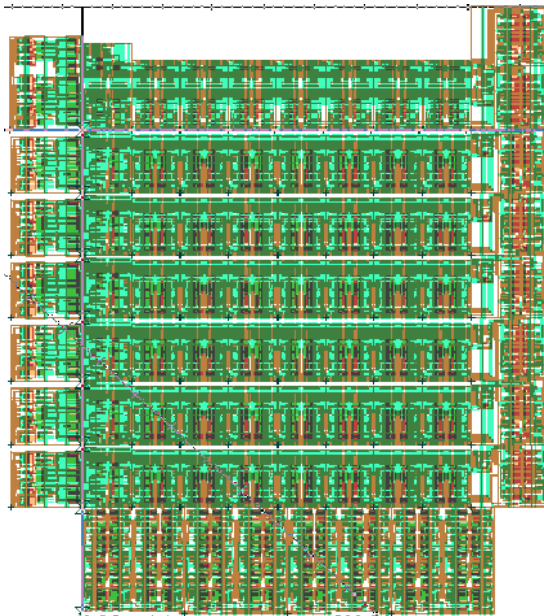


Рис. 1. Топология матричного умножителя Бута разрядностью 16х16

В. Уменьшение числа этапов суммирования

Все методы уменьшения количества этапов суммирования частичных произведений состоят в:

- группировке полученных частичных произведений;
- параллельном их сложении в каждой из групп;
- сложении полученных в группах результатов.

Матричные умножители (МУ), которые используют такой способ увеличения быстродействия, называют многослойными матричными умножителями (ММУ). В них для уменьшения времени задержки матрицы ее разбивают на части, которые начинают работать одновременно. Получаемые на выходах подматриц результаты суммируются.

Есть несколько типов ММУ, которые отличаются друг от друга способом организации топологии. В одних ММУ части матрицы не отделяются друг от друга. Наоборот, их топологически объединяют таким образом, чтобы их строки чередовались друг с другом, как слои в слоеном пироге. В другом типе ММУ подматрицы умножителя топологически разделяют. Каждая из подматриц за исключением первой содержит по две дополнительные строки. Они преобразуют 4-рядный код, получаемый на выходах основной части подматриц, в код 2-рядный. С целью обеспечения параллельной работы строк в подматрицах глубину каждой из них (число последовательно работающих строк сумматоров) увеличивают по отношению к глубине предыдущей подматрицы на «2».

На рис. 2 приведены фрагменты матриц умножителей, состоящих из двух параллельно работающих подматриц, топологически совмещенных (рис. 2, а) и топологически разделенных (рис. 2, б). Квадратами на схеме изображены сумматоры, а кружками – элементы умножения (элементы «2И» или мультиплексоры).

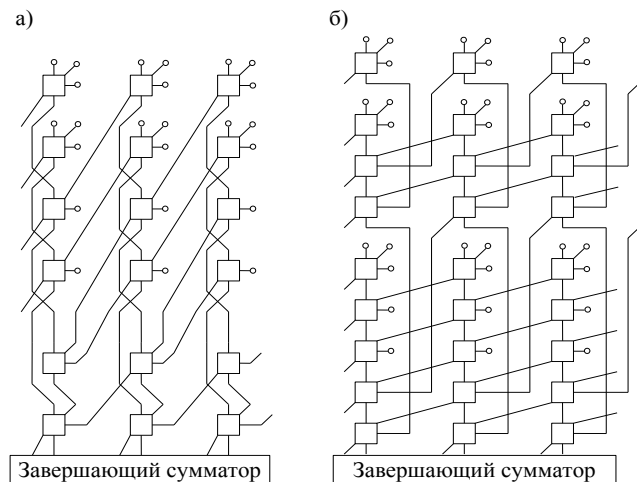


Рис. 2. Фрагменты матриц умножителей, с топологически совмещенными (а) и с топологически разделенными подматрицами (б)

Матрица может быть разделена и на большее число частей, что уменьшит глубину схемы. Глубина схемы — это максимальное количество строк сумматоров, через которые сигналы проходят от входа в матрицу до ее выхода (до формирования двухрядного кода). Однако, при увеличении количества подматриц увеличивается и число связей между ними. Это приводит к дополнительным аппаратным затратам.

III. ПОВЫШЕНИЕ РАБОЧЕЙ ЧАСТОТЫ МУ. КОНВЕЙЕРНЫЕ УМНОЖИТЕЛИ

Есть еще один метод, позволяющий распараллелить процесс вычисления в матричных умножителях. Однако направлен он не на увеличение их быстродействия (наоборот, время умножения увеличивается), а на повышение рабочей частоты МУ. Это конвейеризация работы схем. Оно позволяет устранить «простои» в устройствах. Для этого в схему вводятся конвейерные регистры. В них запоминаются промежуточные результаты, а сформировавшие их части схемы приступают к обработке следующей порции информации.

Простейшим вариантом конвейерного МУ (КМУ) является двухтактный МУ. Его структура предполагает отделение от основной матрицы, которая генерирует и суммирует частичные произведения, сумматоров младших разрядов и старших разрядов произведения. Производительность КМУ можно и еще более увеличить. Для этого помимо отделения от основной матрицы схем сумматоров младших и старших разрядов произведения (как это было в рассмотренном выше варианте) можно разбить на составляющие множитель. Так основная матрица умножителя разбивается на соответствующее количество частей.

Этот подход иллюстрируется топологией КМУ на основе алгоритма Бута с разбиением матрицы на три части, приведенной на рис. 3. Топология сгенерирована параметризованным по разрядности кремниевым компилятором, также разработанным в СПб ГЭТУ «ЛЭТИ».

Своеобразной «вершиной» метода конвейеризации является так называемая «поразрядная конвейеризация» [11–13]. Она обеспечивает наибольшую производительность работы устройства умножения по сравнению с рассмотренными ранее вариантами. К этому классу от-

носятся МУ, у которых в каждом такте вычисляется один разряд произведения, начиная с младшего. 8-разрядный КМУ, созданный фирмой AT&T Bell Laboratories по этой технологии, показан на рис. 4.

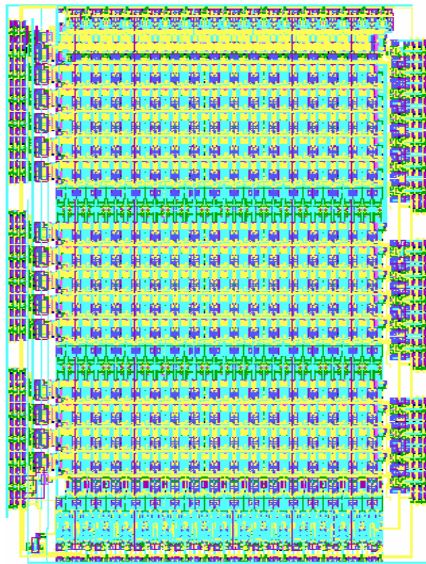
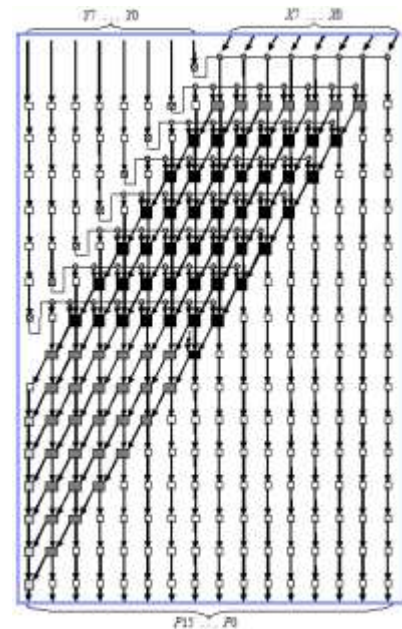


Рис. 3. Топология конвейерного матричного умножителя на основе алгоритма Бута

Как видно из рис. 4 в схеме достаточно много разряженных зон, которые вносят существенный вклад в аппаратные затраты.

В качестве примера на рис. 5 представлены результаты структурно-топологического преобразования конвейерного матричного умножителя. Преобразования, выполненные в соответствии с изложенной в [14] методикой оптимизации регулярных макроблоков СБИС, позволяют снизить площадь схем на кристалле примерно на 35 %.



- Сумматор и три триггера
 - Полусумматор и два триггера
 - ▣ Триггер и буфер
 - Триггер
 - Элемент «EXOR» и триггер
 - Элемент «2И» и триггер
- X – множимое, Y – множитель,
P – произведение

Рис. 4. Структурно-топологическая схема 8-разрядного матричного умножителя с поразрядной конвейеризацией, фирмы AT&T Bell Laboratories

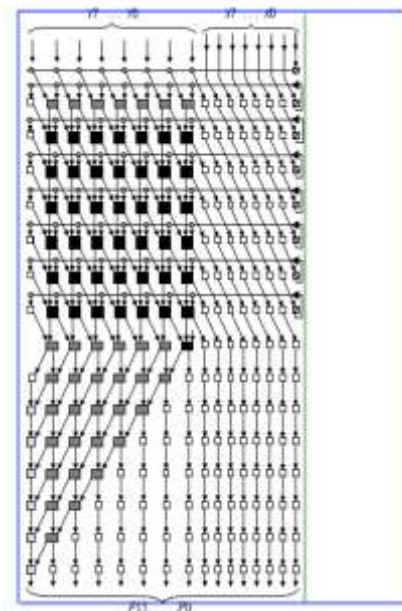


Рис. 5. Результаты структурно-топологического преобразования матричного умножителя с поразрядной конвейеризацией, фирмы AT&T Bell Laboratories

На рис. 6 представлена топология КМУ с поразрядной конвейеризацией. Она также получена в системе иерархического сжатия топологии «Matching of cell».

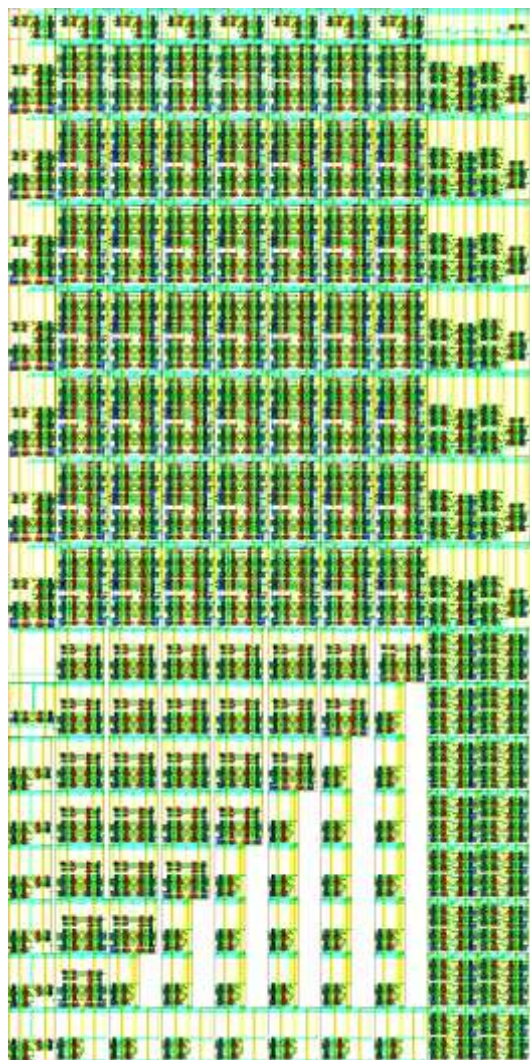


Рис. 6. Топология конвейерного матричного умножителя с поразрядной конвейеризацией, полученная методом технологически инвариантной программной генерации

IV. ЗАКЛЮЧЕНИЕ

В работе рассмотрены варианты оптимизации умножителя – одного из основных элементов нейронной сети, определяющего ее быстродействие и производительность. Применяющиеся приемы иллюстрируются примерами топологии быстродействующих матричных умножителей, реализованных авторами методом программной генерации с помощью оригинальной САПР. Среди возможных решений, позволяющих достичь компромисса между быстродействием и площадью кристалла, особый интерес для разработчиков могут представлять матричные умножители на основе комбинации рассмотренных приемов оптимизации. Высокие характеристики обеспечиваются сочетанием метода умножения на группу разрядов с методами распараллеливания процесса сложения или методами конвейеризации вычислений.

СПИСОК ЛИТЕРАТУРЫ

- [1] E. Nurvitadhi, Jaewoong Sim, D. Sheffield, A. Mishra, S. Krishnan and D. Marr, "Accelerating recurrent neural networks in analytics servers: Comparison of FPGA, CPU, GPU, and ASIC," 2016 26th International Conference on Field Programmable Logic and Applications (FPL), 2016, pp. 1-4, doi: 10.1109/FPL.2016.7577314.
- [2] P.D. Deotale and L. Dole, "Design of FPGA based general purpose neural network," *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 2014, pp. 1-5, doi: 10.1109/ICICES.2014.7033843.
- [3] Sahin S., Becerikli Y., Yazici S. (2006). Neural Network Implementation in Hardware Using FPGAs. In: King, I., Wang, J., Chan, L.W., Wang, D. (eds) *Neural Information Processing. ICONIP 2006. Lecture Notes in Computer Science*, vol 4234. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11893295_122
- [4] J. Ravichandran and R.K. Radhakrishnan Nair, "An Optimization Technique for General Neural Network Hardware Architecture," *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, 2020, pp. 252-255, doi: 10.1109/ICDCS48716.2020.243591.
- [5] M. Kim and J. Seo, "Deep Convolutional Neural Network Accelerator Featuring Conditional Computing and Low External Memory Access," *2020 IEEE Custom Integrated Circuits Conference (CICC)*, 2020, pp. 1-4, doi: 10.1109/CICC48029.2020.9075931.
- [6] Raju Machupalli, Masum Hossain, and Mrinal Mandal, "Review of ASIC Accelerators for Deep Neural Network," *Microprocessors and Microsystems* 89(16):104441 DOI: 10.1016/j.micpro.2022.104441
- [7] R. Balakumaran and E. Prabhu, "Design of high speed multiplier using modified Booth algorithm with hybrid carry look-ahead adder," *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2016, pp. 1-7, doi: 10.1109/ICCPCT.2016.7530164.
- [8] Akella Srinivasa Krishna Vamsi, S. R. Ramesh, "An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics", *Communication and Signal Processing (ICCSPP) 2019 International Conference on*, pp. 0319-0322, 2019.
- [9] S.E. Mironov and A.Y. Vasiliev, "Controlling the process of coordination of complex layout objects of microelectronic systems in conditions of uncertainty of design rules," *2017 IEEE II International Conference on Control in Technical Systems (CTS)*, 2017, pp. 192-195, doi: 10.1109/CTS.2017.8109523.
- [10] S.E. Mironov and K.M. Zibarev, "Management of Layout Matching of Objects of Complex Microelectronic Systems with Uncertainty of Design Rules," *2019 III International Conference on Control in Technical Systems (CTS)*, 2019, pp. 69-73, doi: 10.1109/CTS48763.2019.8973275.
- [11] M. Hatamian and G.L. Cash, "A 70-MHz 8-bit x 8 bit Parallel Pipelined Multiplier in 2.5-um CMOS" in *IEEE Journal of Solid-State Circuits*, August 1986. vol. SC-21, no. 4. pp. 505-513.
- [12] M. Hatamian and G.L. Cash, "Parallel bit-level pipelined VLSI designs for High-speed signal processing". *Processings of the IEEE*. 1987. vol. 75, no. 9, P. 1192-1202, Sep.
- [13] Glenn L. Cash, Mehdi Hatamian, Adrianus Ligtenberg, "Pipeline Arithmetic Adder and Multiplier". American Telephone and Telegraph Company, AT&T Bell Laboratories. Patent Number: 4,887,233. Date: 12 Dec., 1989.
- [14] Миронов С.Э., Сафьянников Н.М., Фролкин А.К. Методика структурно-топологической оптимизации регулярных макроблоков СБИС. // *Известия СПбГЭТУ «ЛЭТИ»*, 2015, стр. 17-23.

Нейросетевой подход к прогнозированию рынка электроэнергии с использованием разночастотных данных среди объясняющих переменных

В. С. Косарев

РАНХиГС при Президенте РФ

kosarev-vs@ranepa.ru

Аннотация. Для генераторов оптового рынка электроэнергии остро стоит проблема краткосрочного прогнозирования, поскольку запуск и остановка генерирующего оборудования является трудоемким и дорогостоящим процессом. Мы хотим предложить гибридную модель машинного обучения для задачи условного прогнозирования почасового спроса на электроэнергию на рынке на сутки вперед, позволяющую использовать разночастотные данные среди объясняющих переменных. Необходимость такой модели объясняется тем, что многие причинные факторы имеют относительно низкую частоту измерения. Мы предлагаем модель, позволяющую использовать разночастотные данные без предварительного приведения к одной размерности, которая представляет собой комбинацию сверточной и рекуррентной нейронных сетей. Идея состоит в том, что данные с разной частотой измерения подаются в сеть в виде двух отдельных матриц. К каждой матрице применяются аффинные преобразования, сводящие данные к одной размерности внутри сети посредством обучаемых параметров. Приведенные к одной размерности данные используются совместно для выделения значимых паттернов рекуррентной нейронной сетью. По результатам серии экспериментов точность прогнозирования предложенной модели превосходит ряд моделей бенчмарков.

Ключевые слова: нейронные сети, прогнозирование, временные ряды, оптовый рынок электроэнергии, разночастотные данные

I. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Во многих работах [1, 2, 3] по прогнозированию спроса на электроэнергию используются модели с одинаковой частотой наблюдений объясняющих переменных. Однако показатели энергетического рынка, в том числе цены и объемы произведенной электроэнергии, могут иметь высокую частоту измерения – часы и минуты. И, соответственно, прогноз, имеющий практическую ценность для генераторов, должен быть построен с таким же уровнем детализации. При этом большая часть причинных экономических факторов имеет меньшую частоту измерения. К примеру, индексы промышленного производства – месячную. При этом включение таких факторов в модель позволяет лучше прогнозировать трендовую компоненту спроса [4]. Таким образом, возникает проблема совмещения данных разной частотности.

В практике обработки временных рядов, как правило, данные с более высокой частотой агрегируются до самой

низкой частоты путем усреднения, суммирования или взятия репрезентативного значения, либо, наоборот, данные с более низкой частотой интерполируются до самой высокой частоты. Однако, при агрегации теряется информация о структуре временного ряда, что приводит к ухудшению качества модели или же такое невозможно сделать из практических соображений, как в случае с прогнозированием электроэнергии, когда важно прогнозировать непосредственно высокочастотный показатель. Использование же линейных методов интерполяции может приводить к зашумлению и снижению объясняющей способности фактора.

На настоящий момент в научной литературе, при возникновении проблемы совмещения разночастотных данных при прогнозировании, исследователи наиболее часто прибегают к использованию MIDAS моделей [5, 6], фильтра Калмана для восстановления пропущенных данных и их различных сочетаний [7]. Недостатком подхода моделей семейства MIDAS является экзогенная природа весовых коэффициентов матрицы преобразования размерности. В результате конечный результат может быть чувствительным к выбору данной матрицы. Модели на основе фильтра Калмана приводят данные к одной частотности эндогенно выбранной модели, используя метод максимального правдоподобия. Собственное исследование предполагает создание альтернативной модели, основанной на технологии сверточных нейронных сетей, которая идейно сводится к фильтру Калмана. Предполагается, что нейросетевая модель, за счет гибкости функциональной формы, сможет превзойти альтернативные модели, с точки зрения сохранения максимального объема информативности временного ряда, при решении прикладной задачи – прогнозировании показателей электроэнергетического рынка.

II. МОДЕЛЬ И РЕЗУЛЬТАТЫ

Предлагаемая в данной работе архитектура нейронной сети для прогнозирования показателей энергетического рынка представляет собой комбинацию сверточной (CNN) и рекуррентной сети (LSTM) с механизмом внимания. В качестве прогнозируемой переменной выступает равновесная цена продажи электроэнергии. Среди объясняемых переменных имеются данные с месячной и почасовой частотой измерения. К месячным данным относятся четыре индекса промышленного производства. К почасовым данным: лаг цены, лаг равновесных объемов, среднесуточная температура и четыре специально

сконструированных фактора для учета сезонных компонент объясняемой переменной. Прогноз строится на один час вперед. Одна прогнозная точка зависит от 72 лаговых значений почасовых данных и трех лаговых значений месячных данных. Таким образом, три дня почасовых данных объясняют цену в следующий час. Одновременно с этим 3 точки, т. е. три предшествующих месяца объясняют каждое прогнозируемое значение следующего месяца. С переходом в эндогенной переменной к следующему месяцу окно сдвигается на месяц вперед. Таким образом, в случае с ИПП – динамика промышленного производства за 3 последних месяца обуславливает прогноз цены в следующем месяце в каждый час.

Возможность использования разночастотных данных в такой сети обусловлена специальной конструкцией сверточного блока. Идея состоит в том, что данные с разной частотой измерения подаются в сеть независимо в виде двух отдельных матриц. К каждой матрице затем применяются операции, сводящие данные к одной размерности при помощи аффинных преобразований с обучаемыми параметрами. Первая матрица имеет размерность $p_1 \times m_1$, где p_1 – это гиперпараметр, отвечающий за продолжительность временного окна высокочастотных данных, а m_1 – число факторов. Протяженность данного окна определяет протяженность рецептивного поля, а количество факторов – его ширина для применяемого в последующем сверточного слоя, что и есть аффинные преобразования для матрицы с высокочастотными данными. В результате размерность высокочастотных данных внутри сети сокращается. Масштаб сокращения размерности определяется внутренними параметрами сверточного слоя. Вторая матрица имеет размерность $p_2 \times m_2$, где p_2 – это продолжительность временного окна для низкочастотных данных, а m_2 – число факторов. С матрицей низкочастотных данных совершается обратная операция – увеличение размерности от p_2 до p_1 по оси времени.

Полносвязная нейронная сеть в дальнейшем используется в качестве декодировщика, приводящего размерность вектора с низкочастотными данными к редуцированной размерности вектора высокочастотных данных. Предполагается, что высокочастотные данные в нейросетевой модели служат в первую очередь источником информации относительно поведения сезонной компоненты и, в том числе, тренда, а экономические факторы – источник информации только для определения положения тренда. Веса полносвязной сети определяют какие лаговые значения месячных данных и в какой степени влияют на формирование целевого вектора, который в дальнейшем конкатенируется с редуцированной матрицей часовых данных, а полученный результат используется последующими блоками непосредственно для построения прогноза. На рис. 1 приведена архитектура описанной нейронной сети.

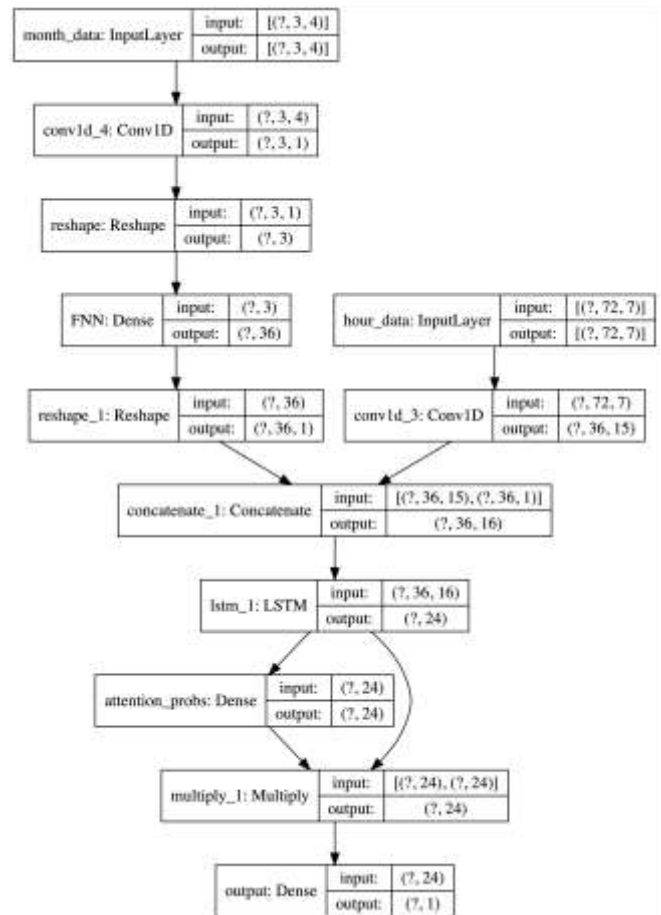


Рис. 1. Архитектура нейронной сети для прогнозирования почасовых объемов электроэнергии оптового рынка

Приведенные к одной размерности месячные и часовые данные используются совместно для выделения значимых паттернов при прогнозировании непосредственно рекуррентной нейронной сетью. В процессе обучения в данном блоке главным образом определяются временные закономерности, а также строится отображение входного вектора в отдельные пространства с уникальными паттернами. После рекуррентного блока полученный вектор передается в сеть внимания. Сеть внимания определяет какой из выделенных паттернов окажет большее влияние на построение прогноза в определенный момент времени. Затем данные попадают в последний выходной слой нейросети, который представляет сумматор с линейной функцией активации. Выход нейронной сети – это объем произведенной электроэнергии в определенный час.

Для проверки оптимальности конструкции предложенной модели также используются две нейронных сети с базовой исходной архитектурой, но исключенными и измененными некоторыми блоками. Первая такая модель предполагает замену блока преобразования месячных данных с использованием инкорпорированной полносвязной нейронной сети на механизм «UpSampling1D», который также приводит к изменению размерности месячных данных до редуцированной размерности часовых данных, но делает это более линейно и задействует на порядок меньше параметров. Вторая модель предполагает исключение блока с месячными данными. Таким образом, в рамках упрощенной модели предполагается, что одно прогнозное значение – это функция от трехдневного

временного окна почасовых данных, которые включают в себя лаговые значения целевой переменной и синтетические переменные, отражающие сезонную структуру данных. Кроме того, в качестве бенчмарков используются: рекуррентная нейронная сеть и сезонная ARIMA с экзогенными факторами.

Расчет ошибки основной и альтернативных моделей осуществляется с учетом валидации, которая предполагает разделение выборки на обучающую, валидационную и тестовую несколькими способами. Каждая модель обучается четыре раза, используя каждый раз сдвинутые на квартал вперед данные. Схема на рис. 2 иллюстрирует принцип работы с данными.



Рис. 2. Валидация выборки в процессе обучения моделей

На блоках с голубым фоном обозначены тренировочные множества модели. Эти данные используются непосредственно для обучения основных параметров нейронной сети. Желтым фоном обозначены валидационные множества. По этим данным определяется момент остановки процесса обучения и точка экстремума функции ошибки с предполагаемой максимальной обобщающей способностью. Зеленым фоном обозначены тестовые множества. Эти данные скрыты на всех этапах обучения и используются только для расчета ошибки. На белом фоне обозначены отложенные данные для следующих итераций расчетов. После каждого этапа обучения рассчитывается три вида ошибки: MAE, MSE и MSLE.

В таблице приведены усредненные по четырем кварталам ошибки основной и альтернативных моделей.

ТАБЛИЦА I СРАВНЕНИЕ ТОЧНОСТИ ПРОГНОЗИРОВАНИЯ РАССМАТРИВАЕМЫХ МОДЕЛЕЙ

Модель	MAE	MSE	MSLE
CNN-RNN	0,049	0,005	0,002
CNN-RNN UpSampling1D	0,049	0,005	0,002
CNN-RNN без месячных данных	0,054	0,006	0,003
LSTM	0,051	0,005	0,002
SARIMA	0,078	0,025	0,023

Наименьшую среднеквадратическую ошибку прогноза ($MSE = 0.049$) и наиболее высокий процент объясненной дисперсии имеют CNN-RNN модели с двумя различными механизмами обработки

низкочастотных данных. Незначительно уступает в точности прогнозирования двухслойная рекуррентная нейронная сеть ($MSE = 0.051$). Несколько в большей степени уступает в точности прогнозирования нейронная сеть CNN-RNN без месячных данных ($MSE = 0.054$). Это говорит о том, что использование месячных данных позволяет нейронной сети задействовать дополнительную информацию для результирующего вывода, которую не содержат высокочастотные объясняемые переменные. Возможно, это связано с тем, что низкочастотные данные, в данном случае – индексы промышленного производства наиболее энергоемких отраслей, могут нести информацию о конъюнктуре рынка, которая влияет на прогнозируемое значение уровня тренда. Наибольшую среднюю ошибку прогноза (0.078) и одновременно низкий процент объясненной дисперсии (62 %) имеет сезонная ARIMA.

III. ЗАКЛЮЧЕНИЕ

В работы была предложена модель машинного обучения для краткосрочного прогнозирования спроса на электроэнергию, которая позволяет использовать низкочастотные данные среди объясняющих переменных без интерполяции. По результатам проведенных расчетов точность прогноза предложенной конфигурации нейросети, измеряемая как средняя ошибка на тестовых выборках валидационных множеств, превосходит сезонную ARIMA и двухслойную LSTM модель. Кроме того, вариация архитектуры модели с отсутствующим блоком месячных макроэкономических факторов в числе объясняющих переменных продемонстрировала более низкие показатели точности, что подтверждает важность данных факторов в формировании трендовой компоненты целевой переменной.

СПИСОК ЛИТЕРАТУРЫ

- [1] Kim T. Y., Cho S. B. Predicting residential energy consumption using CNN-LSTM neural networks // Energy. 2019. V. 182. pp. 72-81.
- [2] Kuo P.H., Huang C.J. A high precision artificial neural networks model for short-term energy load forecasting // Energies. 2018. V. 11. № 1. p. 213.
- [3] Chen Y. et al. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings // Applied Energy. 2017. T. 195. pp. 659-670.
- [4] Do L. P. C., Lin K. H., Molnár P. Electricity consumption modelling: A case of Germany // Economic Modelling. 2016. V. 55. pp. 92-101.
- [5] Kuzin V., Marcellino M., Schumacher C. MIDAS vs. mixed-frequency VAR: Nowcasting GDP in the euro area // International Journal of Forecasting. 2011. V. 27. № 2. pp. 529-542.
- [6] Pan Z. et al. Oil price volatility and macroeconomic fundamentals: A regime switching GARCH-MIDAS model // Journal of Empirical Finance. 2017. T. 43. pp. 130-142.
- [7] Andreou E., Ghysels E., Kourtellis A. Forecasting with mixed-frequency data // The Oxford handbook of economic forecasting. 2011.

Корреляционный дискриминатор сигналов в классе быстрых нейронных сетей

А. Ю. Дорогов

Санкт-Петербургский государственный электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

vaksa2006@yandex.ru

Аннотация. В работе представлены методы построения топологии и параметрического обучения быстрых нейронных сетей (БНС) для задач корреляционных измерений сигналов. Описана математическая модель БНС. Отмечена связь быстрых алгоритмов с самоподобными структурами. Показан способ мультипликативного представления произвольных дискретных функций. Предложены алгоритмы построения топологии БНС и обучения для реализации корреляционного дискриминатора сигналов.

Ключевые слова: линейный дискриминант; быстрая нейронная сеть; фрактальные произведения; самоподобные структуры

I. ВВЕДЕНИЕ

Одним из основных способов распознавания сигналов состоит в сопоставлении сигнала с эталоном. Если сходство между неизвестным сигналом и эталоном велико, то сигнал помечается как соответствующий эталонному. Простейшей мерой сходства является линейный дискриминант Фишера [1]. В частном случае этот дискриминант представляет собой меру взаимной энергии сигналов и выражается через нормированное скалярное произведение

$$d = (x, e) / \sqrt{(x, x)(e, e)} > \alpha,$$

где x – неизвестный сигнал представленный вектором, e – вектор эталонного сигнала, скобки $(\cdot, \cdot)$ – обозначают скалярное произведение векторов, скаляр α определяет пороговый уровень различения сигналов.

Разработку алгоритма быстрого преобразования Фурье (БПФ) безусловно, можно считать одним из выдающихся достижений второй половины 20 века. Дальнейшим развитием этого направления является класс перестраиваемых быстрых алгоритмов. Сохраняя структуру графа и изменяя только параметры базовых операций в перестраиваемых алгоритмах можно переходить от одного быстрого ортогонального преобразования к другому. Первые предложения по построению обобщенного ортогонального преобразования были высказаны Эндрюсом и Каспари [2] в 70-х годах прошлого века. А первые подходы к обучению подобных преобразований были развиты в работах А.И. Солодовникова и его научной группы [3]. В то время подобный класс преобразований называли приспособленными быстрыми преобразованиями.

Возможность перестройки значений весовых коэффициентов и многослойная структура алгоритма роднит быстрые перестраиваемые преобразования с

многослойными нейронными сетями прямого распространения. Иногда используют термин – ортогональные нейронные сети. В рамках данной парадигмы быстрые линейные перестраиваемые преобразования являются частным случаем многослойных нейронных сетей и отличаются от последних линейными функциями активации и нулевыми смещениями в нейронах. Для обозначения нового класса сетей используется термин быстрые нейронные сети (БНС) [4].

Благодаря своей структуре БНС обладают специфичными алгоритмами обучения принципиально отличающимся от классического ErrorBackPropagation отсутствием обратного распространения ошибки, что обеспечивает высокую скорость обучения и абсолютную сходимость. В основе алгоритмов обучения БНС лежит доказанное свойство структурной фрактальности, которое можно выразить системным инвариантом морфологического уровня [5]. Идея метода обучения БНС к одной или нескольким функциям основана на представлении каждой функции заданного набора в виде предфрактального произведения, отвечающего мультипликативной форме представления элементов матрицы быстрого преобразования.

В данной работе показано использование БНС для построения дискриминатора сигналов. Дано описание математической модели БНС, представлены методы построения топологии и обучения БНС для задачи корреляционных измерений.

II. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ БНС

В работе Гуда [6] впервые было показано, что элементы матрицы быстрого преобразования Фурье можно представить в виде произведения элементов матриц базовых операций. На рис. 1 представлен граф быстрого преобразования в топологии Гуда с явным выделением базовых операций. В терминах нейронной сети базовая операция графа – это однослойная нейронная сеть с размерностью рецепторного и аксонового полей равной 2.

Базовую операцию определим матрицей $W_{i_m}^m$, где m – номер слоя, и i_m – номер базовой операции в пределах слоя. Элементы матриц базовых операций обозначим через $w_{i_m}^m(u_m, v_m)$, где u_m, v_m – определяют номер строки и номер столбца матрицы. В общем случае размерности полей базовой операции могут быть отличны от 2.

В алгоритмах быстрых преобразований удобно использовать нумерацию индексов, начиная с нулевого значения. Гуд показал, что элементы матрицы быстрого

преобразования размерности $N = 2^n$ могут быть представлены в виде:

$$h(U, V) = w_{p_0}^0(u_0, v_0) w_{p_1}^1(u_1, v_1) \dots w_{p_{n-2}}^{n-2}(u_{n-2}, v_{n-2}) w_{p_{n-1}}^{n-1}(u_{n-1}, v_{n-1}), \quad (1)$$

где топология преобразования определяется кортежами:

$$\begin{aligned} U^m &= \langle v_{m-1} v_{m-2} \dots v_0 u_{n-1} u_{n-2} \dots u_m \rangle, \\ V^m &= \langle v_m v_{m-1} \dots v_0 u_{n-1} u_{n-2} \dots u_{m+1} \rangle, \\ i^m &= \langle v_{m-1} \dots v_0 u_{n-1} u_{n-2} \dots u_{m+1} \rangle. \end{aligned} \quad (2)$$

Здесь кортежи используются для поразрядного представления чисел в позиционной системе счисления. Например, для системы счисления с основанием 2 имеем:

$$u = \langle u_{n-1} u_{n-2} \dots u_0 u_1 \rangle = u_{n-1} 2^{n-1} + u_{n-2} 2^{n-2} + \dots + u_1 2 + u_0.$$

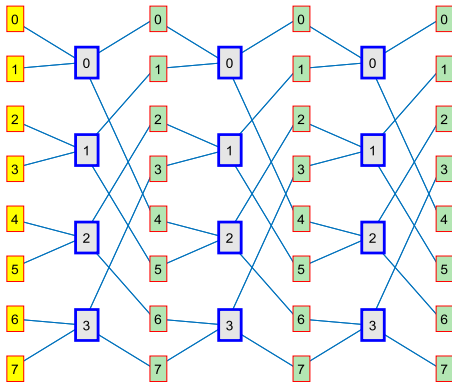


Рис. 1. Топологическая схема Гуда для размерности быстрого преобразования 8 с явным выделением базовых операций

В кортеже поразрядного представления крайний левый разряд считается старшим. Значение U^m определяет номер рецептора, а значение V^m — номер нейрона в слое m . Нумерация слоёв сети начинается с нуля, полное число слоёв равно n . Обобщённая теорема мультипликативной факторизации элементов матриц быстрых преобразований представлена в работе [7].

III. МУЛЬТИПЛИКАТИВНОЕ ПРЕДСТАВЛЕНИЕ ДИСКРЕТНЫХ ФУНКЦИЙ

Рассмотрим сигнал, заданный функцией $f(u)$ на дискретном интервале длиной $N = p_0 p_1 \dots p_{n-1}$, где p_m произвольные целые числа. Представим аргумент функции в позиционной многоосновной системе счисления с основаниями $p_0, p_1, \dots, p_{n-1}$. Формула перехода, как известно, имеет вид:

$$u = \langle u_{n-1} u_{n-2} \dots u_0 \rangle = u_{n-1} p_{n-2} p_{n-3} \dots p_0 + \dots + u_{n-2} p_{n-3} p_{n-4} \dots p_0 + \dots + u_1 p_0 + u_0,$$

где $u_i \in [0, p_i - 1]$ — разрядные переменные. В результате данного преобразования сигнал представляется как многомерная функция $f \langle u_{n-1} u_{n-2} \dots u_0 \rangle$. Каждый аргумент функции определяет

некоторый масштабный срез сигнала. Зафиксируем все аргументы функции кроме u_m . Варьируя свободный аргумент u_m , получим выборку S_m (с числом элементов p_m). Фрактальным фильтром [7] частотной локализации m называется произвольный функционал $F(S_m)$, определённый на выборке S_m . Операцию фрактальной фильтрации можно записать в виде:

$$f_{out} \langle u_{n-1} u_{n-2} \dots u_{m+1} u_{m-1} \dots u_0 \rangle = F_{u_m} \left(f_{inp} \langle u_{n-1} u_{n-2} \dots u_0 \rangle \right).$$

В простейшем варианте фрактальный фильтр выполняет суммирование значений функции по аргументу u_m . Если $m=0$, то такой фильтр генерализует сигнал, сглаживая мелкие детали. Фрактальная фильтрация по аргументу u_m приводит к сокращению интервала определения сигнала в p_m раз.

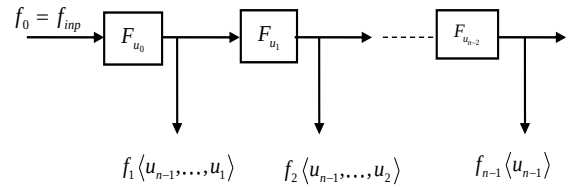


Рис. 2. Цепочка фрактальных фильтров

Рассмотрим цепочку фрактальных фильтров, показанную на рис. 2. Выходные сигналы для фильтров цепочки определяются рекуррентным соотношением:

$$f_m \langle u_{n-1} \dots u_m \rangle = F_{u_{m-1}} \left(f_{m-1} \langle u_{n-1} \dots u_{m-1} \rangle \right).$$

Введем функции:

$$\varphi_i^m(u_m) = \frac{f_m \langle u_{n-1} u_{n-2} \dots u_m \rangle}{f_{m+1} \langle u_{n-1} u_{n-2} \dots u_{m+1} \rangle}, \quad m = 0, 1, \dots, n-2,$$

где $i^m = \langle u_{n-1} u_{n-2} \dots u_{m+1} \rangle$.

Используя определения функций, можно записать

$$f_m \langle u_{n-1} u_{n-2} \dots u_m \rangle = \varphi_i^m(u_m) f_{m+1} \langle u_{n-1} u_{n-2} \dots u_{m+1} \rangle, \quad m = 0, 1, \dots, n-2. \quad (3)$$

Из рекуррентных соотношений (3) непосредственно следует:

$$f(u) = f_0 \langle u_{n-1} u_{n-2} \dots u_0 \rangle = \varphi_{i^0}(u_0) \varphi_{i^1}(u_1) \dots \varphi_{i^{n-2}}(u_{n-2}) \varphi_{i^{n-1}}(u_{n-1}). \quad (4)$$

IV. ПРОЕКТИРОВАНИЕ ТОПОЛОГИИ И ОБУЧЕНИЕ КОРРЕЛЯЦИОННОЙ НЕЙРОННОЙ СЕТИ

Будем полагать, что входной и эталонные сигналы нормированы к энергии, так что $(x, x) = 1$ и $(e, e) = 1$. В этом случае дискриминант определяется скалярным произведением

$$d = (x, e).$$

Скалярное произведение можно рассматривать как произведение вектора-строки x на одно-столбцовую матрицу $H = e'$. Или в покомпонентном выражении

$$d = \sum_U x(U)h(U, V).$$

Будем полагать, что матрица H является матрицей быстрого преобразования с топологией Гуда. Поскольку сеть имеет только один выход, то

$$V = V^{n-1} = \langle 0_{n-1} 0_{n-2} \dots 0_1 0_0 \rangle.$$

Таким образом, все разрядные переменные $v_m = 0$ для $m = 0, 1, \dots, n-1$. Т.е. каждая базовая операция также будет иметь только один выход с локальным номером 0. Выражения (2) для топологии сети в этом случае примут вид:

$$\begin{aligned} U^m &= \langle 0_{m-1} 0_{m-2} \dots 0_0 u_{n-1} u_{n-2} \dots u_m \rangle, \\ V^m &= \langle 0_m 0_{m-1} \dots 0_0 u_{n-1} u_{n-2} \dots u_{m+1} \rangle, \\ i^m &= \langle 0_{m-1} \dots 0_0 u_{n-1} u_{n-2} \dots u_{m+1} \rangle. \end{aligned} \quad (5)$$

В частности для входного слоя сети $m=0$, в этом случае имеем $i^0 = \langle u_{n-1} u_{n-2} \dots u_1 \rangle$. Полагая, что разрядные переменные принимают значения $u_i \in [0, 1, \dots, p_i - 1]$, можно сделать вывод, что слой будет содержать $Z^0 = p_{n-1} p_{n-2} \dots p_1$ базовых операций. Следующий слой, $m=1$, будет содержать $Z^1 = p_{n-1} p_{n-2} \dots p_2$ базовых операций. Для последнего слоя имеем $i^{n-1} = \langle 0_{n-1} 0_{n-2} \dots 0_0 \rangle = 0$, и поэтому этот слой будет содержать только одну базовую операцию. Топология межслойных переходов определяется правилом $V^{m-1} = U^m$.

На рис. 3 представлен топологический граф коррелирующей сети для $N = 2^3$. Каждая базовая операция данной нейронной сети представляет собой один нейрон, имеющий два входа. Граф сети строится непосредственно по топологическим выражениям. Детали построения топологического графа изложены в работе [7].

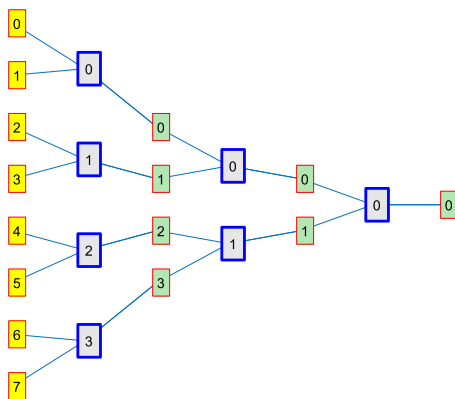


Рис. 3. Топология коррелирующей сети

Сравним между собой выражения (1) и (4). Нетрудно заметить, что при топологии (5) они совпадают. Отсюда сразу следует правило обучения нейронной сети:

$$w_{i^m}^m(u_m, 0) = \phi_{i^m}(u_m).$$

V. ЗАКЛЮЧЕНИЕ

Быстрые нейронные сети относятся к широкому классу самоподобных многослойных нейронных сетей [8]. Самоподобные многослойные сети с регулярной структурой обладают уникальными свойствами аналитического представления графа топологической модели, что позволяет разработать быстрые абсолютно сходящиеся алгоритмы обучения с конечным числом вычислительных операций. В данной статье алгоритм быстрого преобразования с топологией Гуда использован как пример, подобный алгоритм может быть реализован для топологии Кули-Тьюки или любой иной топологии быстрых преобразований.

Представленные в работе результаты показывают, что сети могут быть эффективно использованы для корреляционных измерений в одномерном сигнальном пространстве. Подобным образом могут быть построены дискриминаторы для локализации изображений и объектов в многомерных пространствах. Более того, регулярный лес корреляционных сетей позволяет реализовать быстрые нейронные сети глубокого обучения [9].

СПИСОК ЛИТЕРАТУРЫ

- [1] Фишер Р.А. Статистические методы для исследователей. Москва, Госиздат, 1958. 267 с.
- [2] Andrews H.C., Caspari K.L. A General Techniques for Spectral Analysis // IEEE. Tr. Computer. 1970. Vol C-19. Jan, No 1. P.16-25.
- [3] Солодовников А.И., Спиваковский А.М. Основы теории и методы спектральной обработки информации. Л., 1986. 272с.
- [4] Дорогов А.Ю. Быстрые нейронные сети: Проектирование, настройка, приложения. // Лекции по нейроинформатике Ч.1. В тр. школы-семинара «Современные проблемы нейроинформатики», науч.-техн. конф. «Нейроинформатика-2004» Изд. Москва: МИФИ, 2004. С. 69-135.
- [5] Дорогов А.Ю., Шестопалов М.Ю. Нейросетевое моделирование регулярных фракталов. // Нейрокомпьютеры: разработка и применение. №6, 2007. С. 3-15.
- [6] Good I.J. The Interaction Algorithm and Practical Fourier Analysis // Journal of Royal Statistical Soseity. Ser.B. 1958. Vol.20. No.2. P. 361-372.
- [7] Дорогов А.Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб.: «Политехника», 2014. 328 с.
- [8] Дорогов А.Ю. Самоподобные структуры многослойных нейронных сетей // XXII Международная научно-техническая конференция «Нейроинформатика-2020»: Сборник научных трудов. М. НИЯУ МИФИ, 2020. С. 214-224.
- [9] Дорогов А.Ю. Быстрые нейронные сети глубокого обучения // Сборник докладов III Международной научной конференции по проблемам управления в технических системах (CTS'2019). СПб.: СПбГЭТУ «ЛЭТИ». С. 275-280.

Модель распознавания информационных угроз с использованием рекуррентной нейронной сети

Ю. А. Шичкина, Р. Р. Фаткиева

Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
strange.y@mail.ru

И. А. Пузако

Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
Ivanpuzako@gmail.com

Аннотация. Предложен метод непрерывной обработки сетевых данных, который повышает скорость и эффективность реагирования на атаки инфраструктуры сети за счет представления сетевого трафика в виде многомерного временного ряда. В работе реализован метод сбора данных, определены основные характеристики, позволяющие сформировать пространство атак. Предложена архитектура нейронной сети, состоящая из двух рекуррентных слоев и одного полносвязного слоя, сформирован процесс обучения и анализа эффективности распознавания сетевых угроз на отложенной выборке данных. Результатом работы является программный модуль, способный классифицировать текущее состояние безопасности сетевого трафика и сообщать пользователю или системе защиты об обнаружении угрозы.

Ключевые слова: сетевые атаки; рекуррентная нейронная сеть, сетевой трафик

I. ВВЕДЕНИЕ

Эксперты в сфере информационной безопасности сходятся во мнении, что реально оценить ущерб от кибератак практически невозможно. Во-первых, не все организации точно знают о своих потерях, в связи с тем, что не занимаются информационной безопасностью в своих компаниях. Во-вторых, многие организации, столкнувшиеся с кибератаками, не спешат обнародовать свои убытки, чтобы избежать санкций со стороны регуляторов. Было посчитано, что в 2018 году убытки российских компаний от сетевых атак превысили 116 млн руб. Если говорить о мировых масштабах ущерба для мировой экономики, компания Allianz Global Corporate & Specialty оценила его в более чем 575 млрд долларов – порядка 1 % мирового ВВП [1]. В октябре 2021 года «Лаборатория Касперского» сообщила о самых дорогих видах кибератак. Дороже всего компаниям обходятся так называемые таргетированные или целенаправленные нападения – один такой инцидент приводит к убыткам крупных компаний в среднем в 695 тысяч долларов, а малый и средний бизнес теряет около 32 тысяч долларов [2]. В течение 2021 года с такими киберинцидентами столкнулись около 35 % организаций в РФ. В связи с этим повышаются и требования к безопасности информационных систем. Компьютерные сети, обеспечивающие взаимосвязь между различными устройствами, подвержены постоянному риску информационных атак, которые могут стать причиной кражи данных, нестабильности работы или полного отказа системы. В связи с этим появляется необходимость разработки более эффективных систем обнаружения вторжений.

Существующие во многих системах методы основаны на распознавании сигнатур и эвристическом анализе, в результате чего имеют проблемы с адаптивностью к новым типам атак и зашумленным данным [3–5].

Нейросетевые методы позволяют удовлетворить высокие требования к производительности за счет автоматического нахождения сложных взаимосвязей в необработанных данных, это повышает точность распознавания в сравнении с сигнатурными методами за счет устранения проблемы постоянного обновления актуальных баз данных сетевых угроз [6].

Задачей работы является повышение эффективности распознавания сетевых угроз для систем анализа сетевого трафика за счет использования нейронной сети.

II. АРХИТЕКТУРА НЕЙРОННОЙ СЕТИ

Модель предложенной нейронной сети основана на рекуррентной архитектуре GRU (Gated Recurrent Unit) [7], которая является одной из модификаций архитектуры LSTM (long short-term memory) [8]. Модель позволяет уйти от проблем долгосрочных зависимостей и затухающих градиентов, присутствующих у стандартных рекуррентных блоков, за счет уменьшения количества матриц обучаемых весов, состоящих из обучаемых параметров. Это ускоряет обучение сети и позволяет использовать больше последовательно соединённых рекуррентных слоев. В работе [9] показано, что наиболее важными частями LSTM являются выходной и забывающий гейты, которые совмещены в модели GRU, а два передаваемых скрытых состояний объединены в одно – h_t . Формально один блок GRU можно описать следующим образом:

$$u_t = \sigma(W_{xu} \cdot x_t + W_{hu} \cdot h_{t-1} + b_u) - \text{update gate},$$

$$r = \sigma(W_{xr} \cdot x_t + W_{hr} \cdot h_{t-1} + b_r) - \text{reset gate},$$

$$h'_t = \tanh(W_{xh'} \cdot x_t + W_{hh'} \cdot (r_t \odot h_{t-1})),$$

$$h_t = (1 - u_t) \odot h'_t + u_t \odot h_{t-1},$$

где x_t – входной вектор на шаге t , h_t – передаваемое скрытое состояние блока, которое также служит выходным вектором ячейки и считывается следующими слоями в качестве входного вектора. В данной архитектуре u_t определяет, какую часть информации нужно перенести без изменений с предыдущего скрытого состояния, а r_t так же описывает, какую часть информации нужно перенести из предыдущего шага и каким образом объединить новый входной вектор с имеющимся скрытым состоянием на данном временном

шаге. Конечная схема обработки данных в нейронной сети представлена на рис. 1.

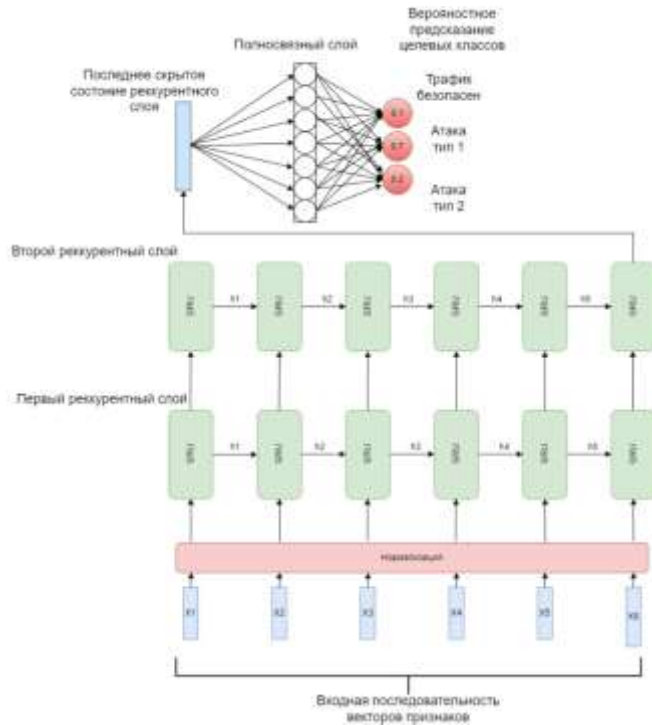


Рис. 1. Схема обработки данных в нейронной сети

1) На вход модели подается последовательность векторов фиксированной длины с характеристиками сетевого трафика. Каждый элемент последовательности представлен в виде вектора со значениями после нормализации данных.

2) Полученная последовательность векторов характеристик описания сетевого трафика обрабатывается с использованием двух последовательных рекуррентных блоков GRU. Скрытые состояния h_t на каждом временном шаге первого рекуррентного блока являются входными векторами второго блока GRU. В результате этого этапа не происходит изменений длины последовательности, обрабатывается только содержимое признаков и извлекается информация из последовательности.

3) Последнее скрытое состояние второго слоя GRU описывает в латентном представлении всю извлеченную информацию из последовательности, которую необходимо преобразовать в метку целевого класса, идентифицирующего атаку. Для этого последнее скрытое состояние пропускается через полносвязный слой, размерность выхода которого совпадает с количеством целевых классов, с функцией активации LogSoftmax:

$$\text{LogSoftmax}(x_i) = \log_2 \left(\frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \right),$$

где x_i – выходные значения полносвязного слоя до функции активации для одного из n классов. Значения этой функции активации возможно интерпретировать как двоичный логарифм вероятности меток классов, поскольку значения, стоящие под знаком логарифма принадлежат промежутку от 0 до 1, а сумма значений для всех классов равна 1. Это позволяет выбрать

наиболее вероятный класс и принять решение о безопасности сетевого трафика.

4) На этапе обучения после каждого обработанного окна вычисляется значение функции ошибки модели, которое должно быть минимизировано следующим образом:

$$L(\theta, x, y) = - \sum_{k=1}^K y_k \cdot f(\theta, x)_k \rightarrow \min_{\theta}$$

где f – обучаемая нейронная сеть, K – количество целевых классов, y_k – истинная бинарная метка 0 или 1 для определенного целевого класса, θ – обучаемые параметры модели, x – входные данные $f(\theta, x)_k$ – предсказанное нейронной сетью значение двоичного логарифма вероятности определенного целевого класса. В результате, имеется возможность использовать модель в двух режимах:

- **многоклассовый** – используя выходное распределение вероятностей классов для выделения конкретного типа сетевой атаки;
- **бинарный** – используя предсказанную вероятность для безопасного трафика. Таким образом появляется возможность простого выделения активности, отличной от нормального состояния.

III. МОДЕЛЬ РАСПОЗНАВАНИЯ ИНФОРМАЦИОННЫХ УГРОЗ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОЙ СЕТИ

A. Формирование наборов данных, описывающих период сетевой активности для организации обучения нейронной сети и валидации результатов обучения

Если входящие и исходящие информационные потоки представить в виде многомерного временного ряда, то отклонения от статистических закономерностей, характеризующих ряд, позволяет предположить сетевую атаку. На каждом шаге временного ряда необходимо выставить метку, сигнализирующую об опасности или безопасности сетевого трафика, с распознаванием соответствующего типа угроз в случае наступления инцидента. Формально, модель представления данных можно представить себе следующим образом:

$$\begin{array}{ccccc} y_1 & \dots & y_n \\ \uparrow & \uparrow & \uparrow \\ x_{1,1} & \dots & x_{n,1} \\ \dots & \dots & \dots \\ x_{1,k} & \dots & x_{n,k} \end{array}$$

где n – длина исследуемого временного ряда, а k – количество признаков, y_i – числовые метки, характеризующие тип атаки, $i \in \overline{1, n}$ а x_{ij} – значение j признака сетевого трафика, $j \in \overline{1, k}$ на временном шаге i .

Тогда, для распознавания типа сетевых атак в модели необходимо для каждого i -ого шага предусмотреть метки целевого класса, соответствующего определённому типу угроз, а также класс, сигнализирующий о том, что трафик безопасный. Однако для наиболее полного анализа состояния сети на i -ом шаге необходимо учитывать не только признаковое описание текущего

шага, но и предыдущих. Наиболее подходящим решением в подобной ситуации будет учет не всей предыдущей последовательности, а только промежутка (окна) определённой длины, что позволяет учитывать временной контекст, с ограничением потребляемых ресурсов. В этом случае, на вход нейронной сети подается k последних шагов временного ряда, а предсказание метки целевого класса осуществляется на последнем шаге.

В. Подготовка данных для обучения нейронной сети

Предобработка данных увеличивает точность распознавания и предотвращает сбои в формате входных данных. Разделение данных на тренировочную (для построения системы распознавания угроз) и тестовую (для оценки и получения численных характеристик качества обученной модели) части (рис. 2).

- предобработка исходного набора данных, для получения информации о признаковых описаниях сетевого трафика на каждом временном шаге и отбор конечных признаков;
- избавление от пропусков и возможных неожиданных значений признаков, как например отрицательные, если признак характеризует количество каких-либо единиц;
- нормализация данных. Эта процедура необходима как во время обучения нейронной сети, так и во время работы модели в режиме реального времени для того, чтобы распределения данных, на которых модель обучалась, были близки к тем, которые приходится обрабатывать.

С. Обучение нейронной сети

На данном этапе происходит последовательная подача на нейронную сеть элементов обучающей выборки, считается значение выбранной функции ошибки и делаются шаги обратного распространения для подстройки обучаемых параметров для её минимизации.

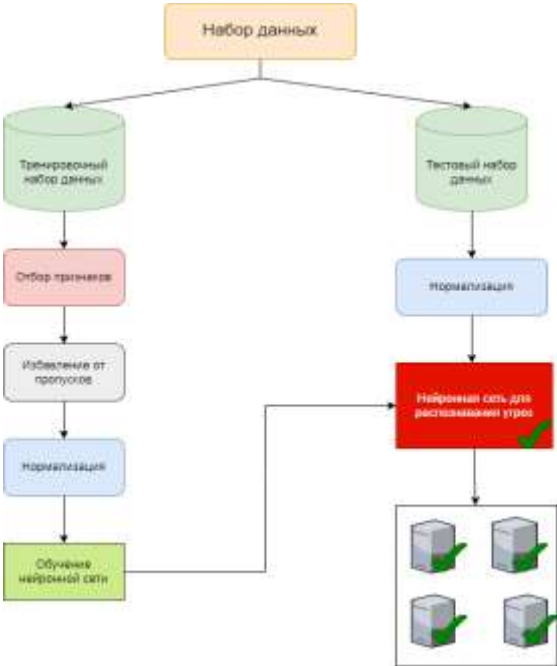


Рис. 2. Схема обработки данных

Д. Оценка точности распознавания сетевых атак обученной моделью и интерпретация результата.

Поскольку в работе стоит задача многоклассовой классификации, то для оценки качества используются такие метрики как: точность, полнота (Precision, Recall), F-мера и матрица ошибок [10]. Все перечисленные метрики, кроме матрицы ошибок, считаются отдельно для каждого целевого класса. В матрице ошибок по строкам описываются истинные значение классов, а по столбцам – предсказанные, так на пересечении какой-либо строки и столбца будет количество элементов, для которых было предсказан класс, указанный в столбце, а верным будет класс, указанный в строке. В случае, если модель верно распознаёт тип атаки, матрица ошибок будет диагональной.

IV. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ РАСПОЗНАВАНИЯ ИНФОРМАЦИОННЫХ УГРОЗ

Для разработки программного модуля распознавания угроз выбран язык Python 3.8, библиотека Pandas, matplotlib, фреймворк глубокого обучения PyTorch.

Шаг 1. Для исследования были взяты данные, представленные в открытый доступ Канадским университетом Нью-Брунсуика CIC-DDoS-2019 [11], которые позволяют извлечь около 80 описательных признаков из текущего сетевого трафика за короткий промежуток времени (длина шага, количество пакетов, количество входящих и исходящих байт). Для распознавания были взяты следующие типы сетевых атак: DdoS MSSQL; DdoS LDAP; SYN flood; UDP flood; DdoS NetBIOS. Реализуемая модель будет обучаться на распознавание 6 классов, 5 из которых – это различные DdoS атаки, и специальный класс для трафика с нормальной активностью.

Шаг 2. В сформированных тренировочных и тестовых данных отобраны только те признаки, которые могут идентифицировать атаку (табл. 1). Отброшены неинформативные признаки, представляющие собой константное значение, а также признаки, в которых содержится большое количество пропусков. Также были удалены признаки, характеризующие конкретные соединения, такие как номер порта и IP-адрес.

ТАБЛИЦА I ЗНАЧЕНИЯ ОПИСЫВАЮЩИХ ПРИЗНАКОВ СЕТЕВОГО ТРАФИКА

Имя вещественного признака	Тип данных признака	Описание
ACK Flag Count	Бинарный	Присутствие ACK сообщений
Average Packet Size	Вещественный	Средний размер пакета в входящем и исходящем трафике
Avg Bwd Segment Size	Вещественный	Средний размер сегмента, на который разделены входящие пакеты для передачи по сети
Avg Fwd Segment Size	Вещественный	Средний размер сегмента, на который разделены исходящие пакеты для передачи по сети
Bwd IAT Min	Целочисленный	Минимальное время, которое приходит после получения пакета до прибытия следующего, мс
Down/Up Ratio	Вещественный	Отношение входящего трафика к исходящему

Шаг 3. Подготовка данных для передачи на нейронную сеть. Для каждого из 6 распознаваемых классов (5 для различных атак и 1 для безопасного трафика) было взято по 150 000 последовательностей векторов, каждая из которых имеет длину в 30 элементов, в которых необходимо будет предсказать класс последнего элемента. Такой размер был выбран ввиду достаточной скорости обучения и обработки таких данных в реальных условиях: время обработки одной такой подпоследовательности составляет 0.007 секунд на процессоре Intel Core i5-10310U с тактовой частотой 1.7 ГГц, что не превышает время между двумя временными шагами, получаемыми с помощью CICFlowMeter. Равное количество элементов обучения для каждого класса взяты для того, чтобы избежать проблемы несбалансированности классов, которая может привести к тому, что нейронная сеть во время обучения хуже подстроится под закономерности, характерные для малочисленных классов, что приведёт к уменьшению точности распознавания.

Для валидационной и тестовой выборки аналогичным образом было взято по 5 000 и 10 000 элементов. Таким образом, размер выборок после разбиения представлен в табл. 2.

ТАБЛИЦА II Количественное описание выборок для РАЗДЕЛЕНИЯ ДАННЫХ

Название выборки	Количество элементов
Тренировочная	800 000
Валидационная	30 000
Тестовая	60 000

Для снижения некорректной оценки финальных результатов распознавания, элементы тестовой выборки взяты не из тех же файлов, которые использовались для тренировки, а из файлов, записанных в другие дни.

Шаг 3.1. Обработка пропущенных значений, которые оставили около 5 % от всех данных и удалены из обучающей выборки ввиду предположения, что пропущенные временные шаги не приведут к значительному падению точности модели.

Шаг 3.2. Нормализация количественных признаков проводилась на тестовых данных и во время работы нейросети в реальных условиях. Для вычисления нормализованных значений использовались медиана и межквартильных размах, которые вычислялись на тренировочной выборке.

Шаг 4. Реализация модели и процесс обучения проводилась с использованием класса torch.nn.Module, который является базовым для создания собственных нейронных сетей. Для описания архитектуры модели также использовался модуль torch.nn, предоставляющий возможность переиспользования многих реализованных слоёв нейронной сети. Гиперпараметры для обучения представлены в табл. 3.

ТАБЛИЦА III ПАРАМЕТРЫ ОБУЧЕНИЯ МОДЕЛИ.

Наименование параметра	Значение параметра
Размер мини-батча	16
Размер окна последовательности	30
Алгоритм стохастического градиентного спуска	Adam
Скорость обучения	0.001

Наименование параметра	Значение параметра
Размерность скрытого состояния первого рекуррентного слоя GRU	64
Размерность скрытого состояния второго рекуррентного слоя GRU	64
Размерность входного вектора	65

Процесс обучения нейронной сети продолжался в течение 7 эпох, после каждой эпохи было посчитано среднее значение функции ошибки на валидационной выборке, данные представлены на рис. 3:

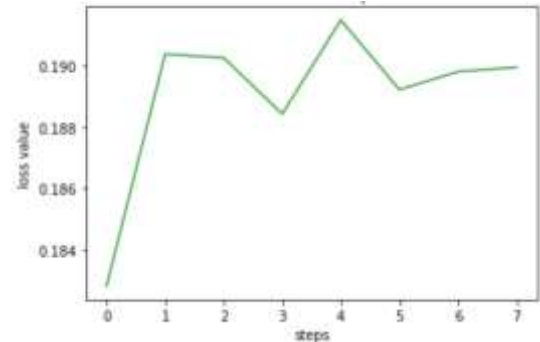


Рис. 3. Среднее значение функции ошибки на валидационной выборке

Из рисунка видно, что наименьшее значение функции ошибки на валидационных данных приходится на эпоху с индексом 3, после чего происходит увеличение данного значения в связи с эффектом переобучения, описанным выше. В процессе обучения были сохранены веса модели.

В результате запуска модели на тестовых данных получены следующие метрик Precision, Recall, F1-мера.

ТАБЛИЦА IV ЗНАЧЕНИЯ ЧИСЛЕННЫХ МЕТРИК НА ТЕСТОВОЙ ВЫБОРКЕ

Тип атаки	Значение метрики		
	Precision	Recall	F1-score
Безопасный трафик	0.76	1.0	0.86
DDoS_NetBIOS	1.0	0.77	0.87
DDoS_UDP	1.0	0.99	1.0
DDoS_Syn	0.98	1.0	0.99
DDoS_LDAP	1.0	0.98	0.99
DDoS_MSSQL	1.0	0.91	0.95

Из таблицы видно, что наибольшее значение ошибок приходится на распознавание атаки DDoS NetBIOS, остальные атаки распознаются верно в более чем 90 % случаев. Матрицы ошибок модели представлены на рис. 4.

Наиболее частая ошибка модели, как видно из матриц, что в 22,6 % атака DDoS NetBIOS и в 8.9 % атака DDoS MSSQL принимаются за безопасный трафик. Однако при продолжительной DDoS атаке данных типов их удастся распознать, так как значительная часть трафика этих классов распознаётся без ошибок. Также только на 0.36 % безопасного трафика была обнаружена угроза, что говорит о том, что разрабатываемый модуль не будет подвержен проблеме частых ложных срабатываний.

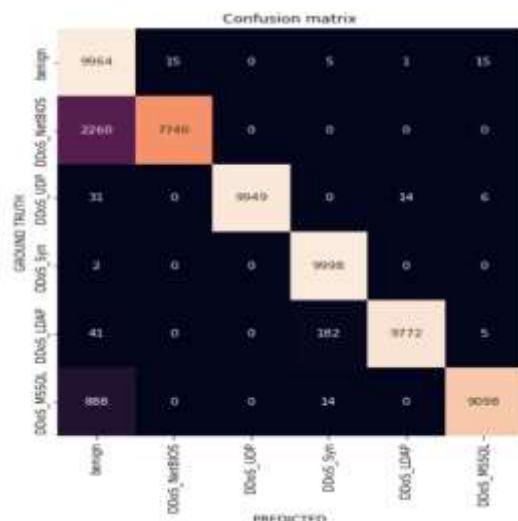


Рис. 4. Матрица ошибок нейронной сети

V. ЗАКЛЮЧЕНИЕ

Предполагается, что представленный метод может быть распространён на многие другие сетевые атаки, которые не попали в рассмотрение в ходе этой работы. Был также разработан программный модуль, способный в режиме реального времени детектировать аномальное поведение и реагировать на него, отдавая оповещение в систему обнаружения вторжений.

СПИСОК ЛИТЕРАТУРЫ

- [1] <https://www.agcs.allianz.com/news-and-insights/news/allianz-risk-barometer-2016.html#>
- [2] <https://www.rbc.ru/rbcfreenews/616473b29a79475aacc1fd76>
- [3] Bednár, Peter & Sarnovsky, Martin & Halas, Pavol. (2020). Semantic Models for Network Intrusion Detection.
- [4] Osama Faker and Erdogan Dogdu. Intrusion Detection Using Big Data and Deep Learning Techniques. In Proceedings of the 2019 ACM Southeast Conference (ACM SE '19). Association for Computing Machinery, 2019, New York, NY, USA, 86–93. DOI:<https://doi.org/10.1145/3299815.3314439>
- [5] Kaja, N., Shaout, A. & Ma, D. An intelligent intrusion detection system. Appl Intell 49, 3235–3247 (2019). <https://doi.org/10.1007/s10489-019-01436-1>
- [6] Shichkina Y.A., Fatkueva R.R. Detection of network attacks using of growing pyramid networks // 2021 10th Mediterranean Conference on Embedded Computing, MECO 2021: 10, Budva, 07–10 june 2021. Budva, 2021. P. 9460188. – DOI 10.1109/MECO52532.2021.9460188
- [7] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, arXiv preprint arXiv:1406.1078.
- [8] Sepp Hochreiter, Jürgen Schmidhuber, “Long short-term memory”, Neural Computation 9(8), pp. 1735-1780, 1997.
- [9] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, Jürgen Schmidhuber, “LSTM: a search space odyssey”, IEEE Transactions on Neural Networks and Learning Systems vol. 28, 2017, pp. 2222 – 2232.
- [10] Margherita Grandini, Enrico Bagli, Giorgio Visani, “Metrics for multi-class classification: an overview”, arXiv preprint arXiv:2008.05756.
- [11] <https://www.unb.ca/cic/datasets/ddos-2019.html>

Применение рекуррентных нейросетей с управляемыми элементами для повышения точности определения звуковых событий в туманной вычислительной среде

А. Н. Субботин¹, Н. А. Жукова², Ф. А. Анаам³

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

¹alesu1543@gmail.com, ²nazhukova@mail.ru, ³fahemye@gmail.com

Аннотация. В статье рассмотрены вопросы применения рекуррентных нейронных сетей для повышения точности определения звуковых событий в системах интеллектуального видеонаблюдения. Рассмотрены новые возможности по определению звуковых событий с применением рекуррентных нейронных сетей с управляемыми элементами. Предложен новый способ определения звуковых событий с применением нейросетей в туманной вычислительной среде. Разработаны приложения для ПК и смартфона. Доказана эффективность использования предложенного способа. Приведены конкретные примеры.

Ключевые слова: рекуррентные нейросети; туманные и облачные среды; обработка звука; машинное обучение; Интернет Вещей

I. ВВЕДЕНИЕ

Нейросети с каждым годом становятся сложнее. В настоящее время это уже не простые свёрточные нейросети для глубокого обучения или рекуррентные сети, напоминающие программы [1], а комплексы, включающие в себя архитектуру самой сети и систему высокопроизводительных вычислений. Рекуррентные нейросети способны показывать значительно более высокие результаты при их применении в туманных и облачных средах [2].

Современные архитектуры рекуррентных нейронных сетей показывают высокую эффективность в различных прикладных областях: распознавание звуков с целью подавления сигналов [3]; повышение точности определения объектов на серии кадров (самолёт, находящийся в движении или космический объект); формирование текстового (семантического) описания на основе картинок, заданных пользователем. Точность определения событий определяется архитектурой нейронной сети, а также проведенным обучением и возможностями по дообучению сети. За счет обучения нейросети адаптируются к решаемой задаче и к типам обрабатываемой информации.

Для решения задач обработки звуковых данных значительный интерес представляет рекуррентная нейросеть с управляемыми элементами [4], которая позволяет восстанавливать искаженные кадры, что

обеспечивает повышение точности определения объектов, находящиеся в постоянном движении.

В целом, нейронные сети можно представить как ряд подпрограмм для определения событий, которые характеризуются:

- типом принимаемой информации;
- сложностью вычислительной и коммутационной архитектуры (серверы, кластеры, роутеры, сетевое оборудование, облака, туманные среды и пр.).

Выходная информация, как правило, является текстовой (в виде сообщения) с фрагментами исходной информации (затемнение найденного объекта, отрывок звука, светового или ультразвукового сигнала и пр.). На входные нейроны рекуррентной нейросети поступает предобработанная информация (умные данные – картинки или звук очень малого размера, где есть только необходимая информация для нейросети). Нейронам промежуточных слоёв в рекуррентной нейросети веса назначаются в режиме обучения. Точность определения объектов и событий определяется процентом, пороговое значение которого задается в соответствии с задачами, решаемыми системой мониторинга. В значительном числе случаев оно устанавливается равным 90%, при превышении порога, информация об обнаружении объекта или о событии передается оператору системы интеллектуального видеонаблюдения. В случае ошибочной работы нейросети, оператор фиксирует ошибку в работе системы, которая затем анализируется специалистами по обработке данных с привлечением экспертов предметной области. Для устранения ошибок, как правило, выполняется дообучение нейронной сети [10], [11].

II. МОТИВАЦИЯ И ОПИСАНИЕ ПРОБЛЕМЫ

Точность определения объектов с применением нейросетей достаточно велика, что подтверждается результатами их применения в различных предметных областях (наблюдение за покупателями на объектах розничной торговли, выявление террористических угроз, предупреждение возникновения опасных ситуаций и пр.). Однако, значительная часть моделей, находящихся в открытом доступе, ориентированы на работу со статическими данными. Применение таких моделей не

Работа выполнена при финансовой поддержке РФФИ, проект № FFZF-2022-0006

позволяет обеспечить требуемую точность определения событий на основе обработки звука.

При работе с динамическими данными хорошо себя зарекомендовали рекуррентные нейросети с управляемыми элементами, рассмотренные в [4]. Визуально такую нейросеть можно представить в виде многослойной спиральной рекуррентной нейросети (рис. 1), где первый слой (1) включен в последующие слои (4), а второй включен в третий (3), имеющий связи с каждым слоем (2) и т. д. по принципу спирали.

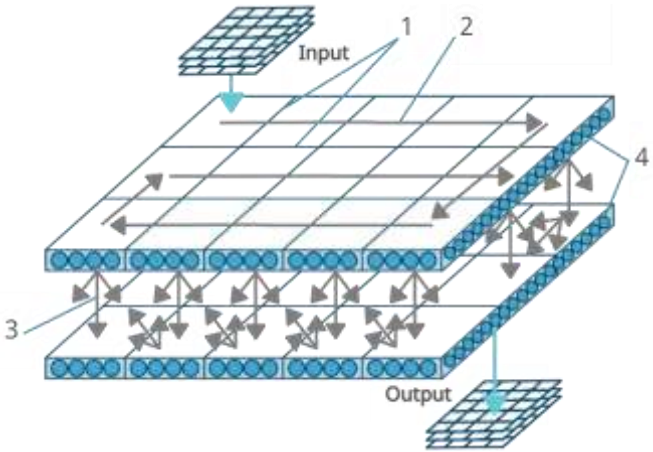


Рис. 1. Рекуррентная нейросеть с управляемыми элементами

Рекуррентная нейросеть с управляемыми элементами успешно применяется при распознавании подвижных объектов на видеоизображениях (летающий самолёт и пр.), которые представляет собой серию кадров, передающихся на входные нейроны нейросети.

III. СПОСОБ РЕШЕНИЯ

Предлагается применить рекуррентную нейронную сеть с управляемыми элементами для обработки звуковых сигналов. Для визуализации выявляемых событий разработаны две программы: 1) приложение рабочего стола под ОС Windows; 2) приложение для смартфона под ОС Android.

Первая программа для визуализации событий разработана в RAD Studio C++ Builder 10.4, выбор языка C++ обусловлен его широким распространением, а использование RAD Studio обеспечило поддержку современных стандартов. Компиляция разработанной программы возможна под разные операционные системы: Windows и MacOS, при этом элементы графических форм (списки, поля, кнопки), функции для обработки изображений и работы с базами данных, как через драйвер ODBC, так и через дополнительные компоненты: DAC for MySQL и пр. сохраняются. Статистическая информация размещается в базе данных MySQL, поддерживающей возможности агрегации статистики и подсчета времени обработки информации [5].

Вторая программа для мониторинга событий так же разработана в RAD Studio, но под мобильные операционные системы: iOS и Android [6]. Компиляция под iOS не предполагается, поскольку эта операционная система широкого распространения не имеет. Обе программы разрабатывались в одной среде (RAD Studio C++ Builder 10.4), что позволило сократить время

реализации за счет частичного пере использования программного кода, а также обеспечило сходство функциональных возможностей разработанных программ.

Разработка нейронной сети проводилась в среде разработки Visual Studio 2022 с использованием предоставляемых средой возможностей по созданию моделей машинного обучения и созданию нейросетей. Существующие библиотеки Keras, TensorFlow [7], [8], [9] и другие при разработке не использовались, поскольку не содержат реализации нейросетей с требуемой архитектурой. В результате была создана консольная программа с возможностью передачи на вход звукового потока и параметров (формат, разрядность, частота дискретизации, число каналов, битрейт, степень сжатия). Для обработки звукового потока, он разделяется на фрагменты и подается в предобученную нейронную сеть на входные нейроны. В случае ошибочного выявления события или пропуска события, оператор вносит данные об ошибке работы сети в экспертную систему (сайт на РНР и база данных MySQL). После анализа ошибок, выполняется дообучение сети, что позволяет ей в дальнейшем выдавать более точные результаты.

Для оценки эффективности предложенного способа определения звуковых событий совместно со специалистами предметной области были определены показатели и критерии эффективности (табл. I). Повышение точности обеспечивается за счет использования рекуррентной нейронной сети с управляемыми элементами, повышение скорости обработки информации и снижение энергопотребления – за счет распределенных вычислений.

ТАБЛИЦА I ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ

No.	Показатели и критерии эффективности	
	Показатель эффективности	Критерий эффективности
1.	Точность определения звуков ^а	увеличение на 10%
2.	Скорость обработки потока	увеличение в 3 раза
3.	Снижение энергопотребления	снижение в 2 раза

^а. Главный показатель.

IV. ПРИМЕР ИСПОЛЬЗОВАНИЯ

Предложенный способ был реализован при разработке приложения для рабочего стола MacOS и Windows, которая выполнялась в среде RAD Studio C++ Builder 10.4. Созданное под Windows 11 приложение [12] (рис. 2) предназначено, прежде всего, для визуализации событий, данные о которых хранятся в базе данных MySQL, подключаемой через драйвер DAC для MySQL v. 3.3.2. Через приложение доступен просмотр в табличном виде перечня событий, времени их возникновения и статуса, обеспечиваются возможности сортировки данных с использованием кнопок, размещенных справа (Down и Up) внизу таблицы, очистки лога по нажатию кнопки слева (Clear), поиска записей (Search) и контроля выполнения процесса обработки звука (Restart, Stop, Start).

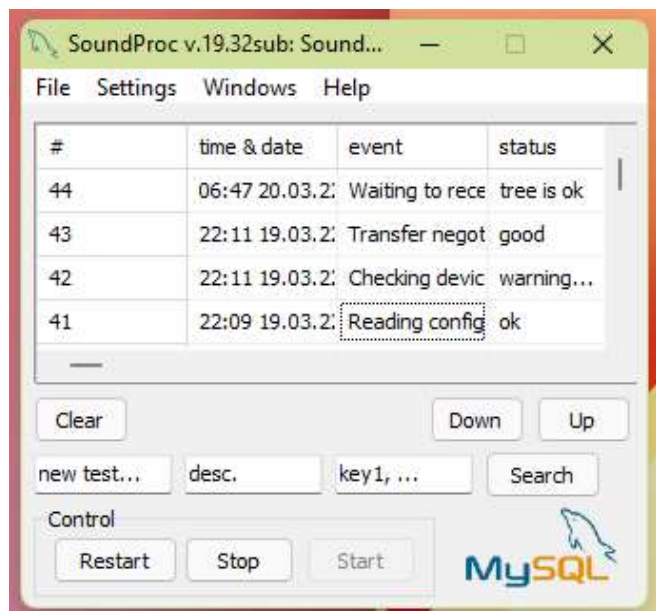


Рис. 2. Внешний вид программы для мониторинга звуковых событий

Вторая программа (рис. 3) предназначена для контроля событий, который осуществляется за счет использования экспертной системы, и последующего привлечения специалиста [13]. Программа также разработана в среде RAD Studio C++ Builder 10.4, скомпилирована под Android. Выбрана версия среды разработки Community, поскольку ее использование разрешено в обучающих и научных целях.



Рис. 3. Внешний вид программы для контроля возникновения событий

После разработки программного обеспечения, реализующего нейросеть на сервере, и программ для смартфона и ПК, была проведена оценка эффективности предложенного способа выявления событий по звуковым данным (табл. II). Выполнялась обработка звуков, источником которых являлся эскалатор в метрополитене, а также находящиеся на эскалаторе пассажиры. На основе скрежета и стука определялись поломки эскалатора, также на основе анализа звуков выявлялись внештатные ситуации, участниками которых являлись пассажиры. В среднем, точность определения событий по звуковым данным на основе нейросети с управляемыми элементами повысилась почти на 13 % относительно базовой модели, созданной в TensorFlow.

ТАБЛИЦА II

Оценка точности определения звуков

№.	Звука	Определение звуков		
		Без использования предлагаемого способа	С использованием предлагаемого способа	Результат
1.	Скрежет ^b	82,7%	94,8%	12,1%
2.	Стук	79,51%	96,12%	16,61%
3.	Крик	84,9%	98,27%	13,37%
4.	Плач	81,04%	94,05%	13,01%
5.	Гул	85,2%	95,02%	9,82%
Среднее:				12,98%

^b. Главный показатель неисправности эскалатора.

На скорость обработки потока звуковых данных влияют запущенные на компьютере другие приложения, установленный приоритет выполнения команд для процессора. При распределенной обработке влияние оказывают зашумленность канала и внешние запросы к разнообразным ресурсам по TCP/IP, которые могут увеличить время выполнения обработки звуковых данных. В связи с этим обработка выполнялась в различные временные интервалы (табл. III). Благодаря применению туманных вычислений, удалось повысить скорость обработки звука в 3,5 раза.

ТАБЛИЦА III

Оценка скорости обработки звукового потока

№.	Период обработки	Длительность	Скорость обработки		Результат
			Без использования предлагаемого способа	С использованием предлагаемого способа	
1.	21:00-23:59, 19 марта 2022 г.	3 ч.	691 мс.	178 мс.	3,88
2.	00:00-07:00, 20 марта 2022 г.	7 ч.	519 мс.	153 мс.	3,39
3.	07:00-09:00, 20 марта 2022 г.	2 ч.	562 мс.	161 мс.	3,49
Среднее:					3,587

Благодаря переносу вычислений в туман, сократилось потребление электроэнергии. По расчетам экспертов-энергетиков на основе данных счетчиков ST 2000-12 (<https://www.sicon.ru/>), потребление электроэнергии сократилось более чем в 3 раза (табл. IV), таким образом, разница составила почти 1,5 раза по сравнению с ожиданиями экспертов.

ТАБЛИЦА IV

Оценка сокращения энергопотребления

№.	Период обработки	Длительность	Энергопотребление (Ватт/ч)		Результат
			Без использования предлагаемого способа	С использованием предлагаемого способа	
1.	Второй период с 1 ч. до 5 ч.	4 ч.	1031	309	3,33
2.	Первый период с 9 ч. до 18 ч.	9 ч.	1787	528	3,38
3.	Третий период с 20 ч. до 1 ч.	5 ч.	1379	371	3,71
Среднее:					3,473

^c. Наиболее показательный период. (Без сильного влияния)

Для воспроизведения результатов экспериментальных исследований, представленных в статье, необходимо разработать программы агрегирования статистики, либо открыть логи программ в Microsoft Excel 2019, применить фильтрацию, функции статистики: суммирование, среднее значение, округление до десятка, и заполнить результирующие таблицы.

Для сбора статистики применялся ноутбук ASUS Laptop R522MA-BR021T с экраном 15.6" и техническими характеристиками: 1366x768, TN+film, Intel Celeron N4020, 2 ядра x 1.1 ГГц, RAM 4 ГБ, HDD 0 ГБ, SSD 0 ГБ, eMMC 128 ГБ, Intel UHD Graphics 600, Windows 11. Для контроля событий был использован смартфон Huawei P50 Pro JAD-LX9 с постоянным доступом в Интернет к серверу Microsoft Azure (<https://azure.microsoft.com>).

V. ЗАКЛЮЧЕНИЕ

В ходе исследования были оценены возможности применения рекуррентных нейронных сетей с управляемыми элементами для повышения точности определения событий в системах видеонаблюдения. Обозначена проблема определения звуковых событий. Предложен новый способ их определения, основанный на применении нейронных сетей в туманной вычислительной среде. Разработаны два приложения для визуализации и контроля возникновения событий. Новый способ применялся для определения событий по звуковым данным на эскалаторе в метрополитене. Полученные результаты показали, что точность определения событий повысилась почти на 13 %, скорость обработки звуковой информации увеличилась в 3,5 раза, расход электроэнергии сократился в 3,5 раза.

Представленный способ может быть применен на различных объектах: в трамваях, поездах и другом общественном транспорте, однако, могут потребоваться дополнительные исследования.

БЛАГОДАРНОСТЬ

Авторы выражают благодарность компаниям: Embarcadero, Microsoft, Microolap за предоставленные среды, средства и компоненты для разработки; организациям: ФНЦ СПИИ РАН (<http://www.spiiras.nw.ru>), ГУП «Петербургский

метрополитен» (<http://metro.spb.ru>) за научную поддержку, возможность проведения исследования, создания статьи и представления доклада на конференции NeuroNT-22 в Санкт-Петербурге, Российская Федерация.

СПИСОК ЛИТЕРАТУРЫ

- [1] Osipov V. & Osipova M. Space-time signal binding in recurrent neural networks with controlled elements. *Neurocomputing* 308, 194-204 (2018).
- [2] Matrouk K. & Alatoun K. Scheduling algorithms in fog computing: a survey. *Int. J. Networked Distrib. Comput.* 9, 59-74 (2021).
- [3] Osipov V., Nikiforov V., Zhukova N., Miloserdov D. Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers // *Neural Computing and Applications*, 2020. doi:10.1007/s00521-020-04843-5.
- [4] Osipov V. & Nikiforov V. Recurrent neural networks with controlled elements in restoring frame flows. *Inf. Control Syst.*, 10-17 (2019). doi: 10.31799/1684-8853-2019-5-10-17.
- [5] Ahmed K.D. & Zeebaree S.R.M. Resource allocation in computing: a review. *Int. J. Sci. Bus.* 5, 54-63 (2021).
- [6] Antunes M. et al. Building an IoT platform based on service containerisation. *Sensors (Basel, Switzerland)* 21, 6688 (2021).
- [7] Kulkarni A., Terpenney J. & Prabhu V. Sensor selection framework for designing fault diagnostics system. *Sensors (Basel, Switzerland)* 21, 6470 (2021).
- [8] Subbotin A.N. and Zhdanov V.S. "Application of Machine Learning Methods to Control the Process of Defectoscopy of Railway Tracks," 2021 IV International Conference on Control in Technical Systems (CTS), 2021, pp. 64-67, doi: 10.1109/CTS53513.2021.9562911.
- [9] Mahmood Z., Guide to Ambient Intelligence in the IoT Environment Principles, Technologies and Application // Springer International Publishing AG, Cham, Switzerland, 2019 – 289 p.
- [10] Subbotin A.N. "Data Processing in Foggy Computing Environments for Machine Learning," 2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT), 2021, Pages 51-53. doi: 10.1109/NeuroNT53022.2021.9472203.
- [11] Korzun D., Balandina E., Kashevnik A., Balandin S., Viola F. Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities; IGI-Global, 2019, 199 p.; doi:10.4018/978-1-52258973-0.
- [12] Subbotin A., Zhukova N. and Man T. "Architecture of the intelligent video surveillance systems for fog environments based on embedded computers," 2021 10th Mediterranean Conference on Embedded Computing (MECO), 2021, pp. 1-8. doi: 10.1109/MECO52532.2021.9460270.
- [13] Khaled Matrouk, Kholoud Alatoun Scheduling. Algorithms in Computing: A Survey. *International Journal of Networked and Distributed Computing*. Volume 9, Issue 1, January 2021, Pages 59-74. doi:10.2991/ijndc.k.210111.001.

Машинные методы анализа треморограмм для оценки функционального состояния человека

Н. М. Сафьянников¹, О. И. Буренева²

Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

¹nmsafyannikov@etu.ru, ²oibureneva@etu.ru

З. А. Алексанян

Институт Мозга Человека им. Н.П. Бехтеревой РАН
aro@ihb.spb.ru

Аннотация. Для определения функционального состояния человека используются методы, основанные на оценке двигательной активности по параметрам функционального или патологического тремора. Результат измерения тремора представляется в виде треморограмм – временных рядов, полученных на основе амплитуд тремора. Для классификации треморограмм можно применить известные методы анализа нестационарных рядов, выполнив их адаптацию в соответствии с особенностями исходных данных. В статье предложен алгоритм анализа треморограмм, основанный на их спектральных характеристиках и предполагающий построение спектра, его сглаживание, сравнение спектра с шаблоном на основе DTW метода. В качестве примера применения предлагаемого алгоритма показан вариант диагностики болезни Паркинсона.

Ключевые слова: нестационарные временные ряды; анализ временных рядов; быстрое преобразование Фурье; сглаживание временных рядов; DTW метод; параметры тремора

I. ВВЕДЕНИЕ

Для определения функционального состояния человека разработаны различные методы, основанные на оценке двигательной активности. Двигательная активность проявляется в форме произвольных и произвольных движений, одним из видов произвольных движений является тремор [1], представляющий собой ритмичные, колебательные движения частей тела. Причины возникновения тремора связаны с процессами, происходящими в отделах мозга, отвечающих за моторные функции. Эти процессы могут быть вызваны нарушениями центральной нервной системы (ЦНС), неврологическими патологиями, употреблением алкоголя и наркотических веществ, а также стрессами. Тремор возникает и у здоровых людей на фоне выполнения определенных действий, например при формировании изометрического усилия, когда происходит сильное напряжение мышц без движения [2]. Изометрическое усилие формируется, например, при сжатии кисти в кулак или удержании тяжелого предмета [3].

Работа выполнена при поддержке Минобрнауки России в рамках соглашения № 075-15-2020-933 от 13.11.2020 г. о предоставлении гранта в форме субсидий из федерального бюджета на осуществление государственной поддержки создания и развития научного центра мирового уровня «Павловский центр «Интегративная физиология – медицине, высокотехнологичному здравоохранению и технологиям стрессоустойчивости»

В [4] мы предложили оригинальный метод анализа произвольных движений на фоне произвольного изометрического усилия с контролем действий обследуемого человека с помощью биологической обратной связи и разработали анализатор, реализующий этот метод. Также мы создали различные варианты проведения тестирования, расширяющие диагностические возможности анализатора [5]. Результаты измерений представлены в виде треморограмм – временных рядов, состоящих из оцифрованных значений амплитуд тремора, возникающего при воздействии рук на чувствительные элементы прибора.

Для исследования основных структурных компонент нестационарных временных рядов, полученных при измерении биологических сигналов, разработаны различные методы, основанные на преобразовании Фурье [6], сингулярном спектральном разложении [7], вейвлет-преобразовании [8] и фрактальных методах [9]. Каждый из методов дает дополнительную диагностически ценную информацию, поэтому в исследованиях часто используют комбинацию методов [10].

Наиболее часто вне зависимости от природы сигнала используется спектральный анализ, основанный на быстром преобразовании Фурье (БПФ). БПФ позволяет получить информацию о частотных составляющих исходного сигнала. Поскольку для различных видов тремора характерен различный частотный состав, то, следовательно, по спектральной характеристике произвольного компонента движения можно определить тип тремора и получить информацию о состоянии центральной нервной системы человека или выявить нейродегенеративные заболевания.

Повысить точность и сократить время оценки функционального состояния человека можно применением машинных методов анализа данных. Подобные методы применяются в медицинской практике для диагностики неврологических нарушений [11], однако обрабатываемые данные имеют иную природу: голосовые записи, данные о движении и рукописных узорах, результаты магнитно-резонансной и позитронно-эмиссионной томографии, результаты лабораторных анализов биологических жидкостей. В статье мы показываем подход к организации машинного анализа спектра треморограмм для классификации тремора и определении патологий ЦНС.

II. МЕТОДЫ АНАЛИЗА ТРЕМОРОГРАММ

Известный спектральный состав тремора различной этиологии [2, 12] позволяет определить набор шаблонов, по соответствию с шаблонами можно классифицировать тремор. Таким образом первым этапом анализа является применение быстрого преобразования Фурье к исследуемому временному ряду. Для многих заболеваний имеется качественное описание изменений в частотном составе треморографического сигнала. Поэтому необходимо оценивать «похожесть» спектра, сравнивая его основные тенденции. Чтобы эти тенденции проявились более явно, и допускали автоматизацию анализа необходимо выполнить сглаживание анализируемого спектра, который также представляет собой временной ряд.

Существуют различные методы сглаживания рядов: центрированное, прямое или обратное скользящее среднее; экспоненциальное сглаживание; адаптивный метод, основанный на локальной линейной регрессии; методы «аналитического» выравнивания.

Самый простой метод сглаживания рядов – скользящее среднее [13]. Метод предполагает использование окна нечетной длины, которое последовательно накладывается на точки ряда X , и центральная точка окна x_i заменяется на среднее арифметическое остальных точек s_i :

$$s_i = \frac{1}{2k+1} \sum_{j=-k}^k x_{i+j}$$

Метод скользящего среднего характеризуется низкой эффективностью и наличием краевых искажений, что важно, когда интерес представляют именно крайние точки.

Более эффективный метод сглаживания – экспоненциальное сглаживание [14], представленный в различных вариантах, каждый из которых учитывает разные составляющие ряда: линейную составляющую (метод Хольта), мультипликативные экспоненциальный тренд и периодическую составляющую (метод Хольта-Уинтерса), аддитивные линейную и периодическую составляющую (метод Тейла-Вейджа).

Метод одинарного экспоненциального сглаживания применяется, когда у анализируемого ряда отсутствует и тренд, и периодическая компонента. Базовое уравнение для расчета значения сглаженного s_t ряда в момент времени t имеет следующий вид:

$$s_t = \alpha x_t + (1 - \alpha)s_{t-1}, \quad 0 < \alpha < 1$$

где s_{t-1} – сглаженное значение ряда в предыдущий момент времени и α – параметр экспоненциального сглаживания. Параметр α определяет соотношение между не сглаженным значением на текущем шаге и сглаженным значением с предыдущего шага. Чем меньше значение α , тем больше степень сглаживания. Если последовательно использовать рекуррентное это соотношение, то экспоненциальную среднюю можно описать следующим образом:

$$s_t = \sum_{i=0}^{t-1} \alpha(1-\alpha)^i x_{t-i}.$$

Анализ выражения показывает, что все предыдущие значения ряда учитываются при вычислении текущего сглаженного значения, однако их вклад угасает экспоненциально за счёт роста степени параметра α . Метод достаточно точен и пригоден для автоматического сравнения. Остальные варианты экспоненциального сглаживания не улучшают результат, так как в спектрах треморограмм нельзя выделить составляющие, на учет которых они ориентированы.

Для сглаженных рядов можно выполнить оценку похожести. При исследовании спектров треморограмм анализируемый временной ряд имеет ряд фиксированную длину. Понятие похожести определяем следующим образом: спектр $X = (x_1, x_2, x_3, \dots, x_n)$ является похожим на шаблон $S = (s_1, s_2, s_3, \dots, s_n)$, если существует такое преобразование T , что $T(X) \approx S$. Примерная похожесть временных рядов оценивается на основе оценки несоответствия с учетом выбираемой точности сравнения e . Для оценки данного несоответствия будем использовать значение D , на основании которого происходит расчет разницы путем вычисления расстояния между рядами. Спектр X и шаблон S являются примерно похожими, если расстояние между ними удовлетворяет выражению

$$D(T(X), S) \leq e.$$

В качестве меры, определяющей точность между заданным шаблоном и анализируемым графиком используются различные меры сходства.

Наиболее простая и часто используемая метрика основана на расчете евклидова расстояния, являющегося векторной нормой между двумя временными рядами, и вычисляется по формуле:

$$d_E = \sqrt{\sum_{i=0}^k |x_i - s_i|^2}$$

Основное требование при расчете – одинаковая длина анализируемых рядов. При сравнении спектров треморограмм это требование легко удовлетворяется, поскольку наибольшей амплитудой обладает ограниченное количество гармоник Фурье, то можно рассчитать евклидово расстояние на основе только n -первых значений ДПФ. В решаемой задаче сравнения тенденций в рядах спектров треморограмм евклидово расстояние не может служить адекватной мерой, поскольку оно не позволяет оценивать форму сравниваемых рядов.

Более высокая точность оценки похожести рядов может быть получена методом динамической трансформации времени (Dynamic Time Warping, DTW) [15], являющейся на сегодня наиболее часто используемой мерой во многих приложениях интеллектуального анализа временных рядов. По сравнению с евклидовым расстоянием алгоритмы DTW вычислительно более сложны. Также эта мера дает неточные результаты для схожих рядов, имеющих большую разницу максимальных и минимальных значений [16], однако этот недостаток можно компенсировать масштабированием, что позволит изменить абсолютные значения, но не нарушит общие тенденции изменения спектральной кривой.

Расстояние между двумя временными рядами X и S на основе DTW метода определяется следующим образом

$$D(X, S) = d(n, n).$$

На основе полученной матрицы вычисляется матрица деформаций:

$$d(i, j) = |x_i - y_j| + \min \begin{cases} d(i-1, j) \\ d(i, j-1) \\ d(i-1, j-1) \end{cases}$$

$$d(0, 0) = 0; i = 1, 2, \dots, n; j = 1, 2, \dots, n$$

Основной недостаток метода – сложность расчетов, однако, существуют разные подходы к упрощению реализации DTW алгоритмов, что делает их применение эффективным.

III. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ ТРЕМОРОГРАММ

Для проведения исследований мы проанализировали более 100 треморограмм, из них 60 были получены с помощью устройства [5], остальные были синтезированы с использованием программных методов [17].

A. Получение реальных данных

В исследовании принимали участие практически здоровые люди в возрасте 25–70 лет, а также пациенты с поражением ЦНС. Все участники удерживали разные усилия от 0,5 кг до 5 кг в течение нескольких минут с контролем усилий посредством визуальной обратной связи. Тремор регистрировали непрерывно в течении 10 секунд. Частота квантования аналогового сигнала составляла 320 Гц. Такие настройки измерения позволяют получить достаточное количество точек измерений для статистически достоверной оценки параметров временных рядов. Для выявления признаков заболевания по спектральным характеристикам сигнала в качестве тестовых выбрана болезнь Паркинсона, так как именно для нее наиболее подробно описан спектральный состав тремора. Известно при болезни Паркинсона движения локализуются в диапазоне частот от 3 Гц до 8 Гц. Следовательно, патологические компоненты могут быть извлечены путем фильтрации тремора с использованием метода быстрого преобразования Фурье. На рис. 1 представлены 2-секундные фрагменты треморограмм двух испытуемых: человека без выявленной патологии, и больного с диагнозом болезнь Паркинсона.



Рис. 1. Треморограммы здорового человека и пациента с диагнозом паркинсонизм

На графиках представлена сложная структура колебаний изометрической силы, которую можно разложить на составляющие, соотносимые с разным уровнем управления двигательной системой.

Обработка данных реализована на языке Python с использованием соответствующих библиотек: numpy,

matplotlib, csv. На рис. 2 показаны результаты БПФ треморограммы здорового человека и пациента с диагнозом паркинсонизм.

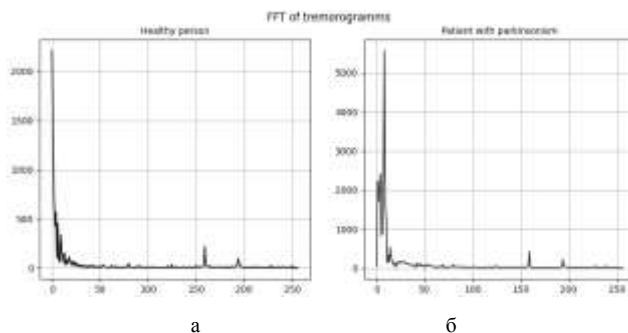


Рис. 2. Спектры треморограмм здорового человека (а) и пациента с диагнозом паркинсонизм (б)

Полученные в результате БПФ треморограмм удобны для визуального анализа, но не могут применяться в интеллектуальном анализе без предварительной обработки.

B. Подготовка анализируемых спектров

Исследования показали, что наилучшие результаты сглаживания получаются при использовании метода экспоненциального одинарного сглаживания. На рис. 3 показаны варианты сглаживания спектров исходных треморограмм для здорового человека и пациента с

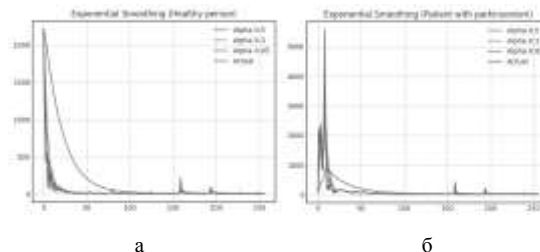


Рис. 3. Пример сглаживания спектров треморограмм для здорового человека (а) и пациента с диагнозом паркинсонизм (б)

Варианты сглаживания для разных значений α приведены на рис. 4.

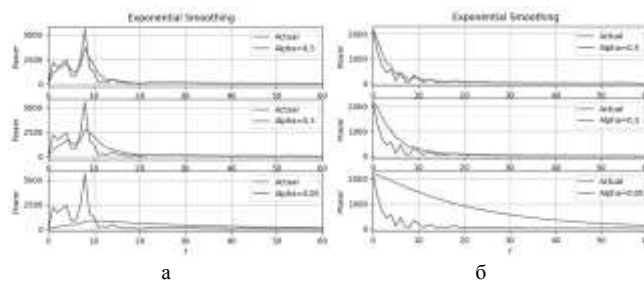


Рис. 4. Сглаживание спектров треморограмм здорового человека (а) и пациента с диагнозом паркинсонизм (б) с использованием экспоненциального одинарного сглаживания со значениями $\alpha = \{0.5, 0.3, 0.05\}$

При анализе спектров треморограмм оптимальным является использование одинарного экспоненциального сглаживания. Значения сглаженного ряда вычисляются путём обновления значений, рассчитанных на предыдущем шаге, используя информацию текущего шага. Информация с предыдущего и текущего шагов берётся с разными весами, которыми можно корректировать качество сглаживания. Для дальнейшего

анализа оптимальным является использование $\alpha = 0,3$, так как с одной стороны сохраняются характерные тенденции изменения графика спектра, с другой – устраняются его резкие изменения.

Сглаженные значения рядов спектров сравнивались с шаблоном, приведенным в [4] с использованием алгоритма DTW. Пример работы алгоритма показан на рис. 5, где на графике (а) представлен сглаженный спектр треморограммы пациента с паркинсонизмом, а на рис. (б) шаблон, характерный для болезни Паркинсона. Рассчитанное значение дистанции $D=0,05$.

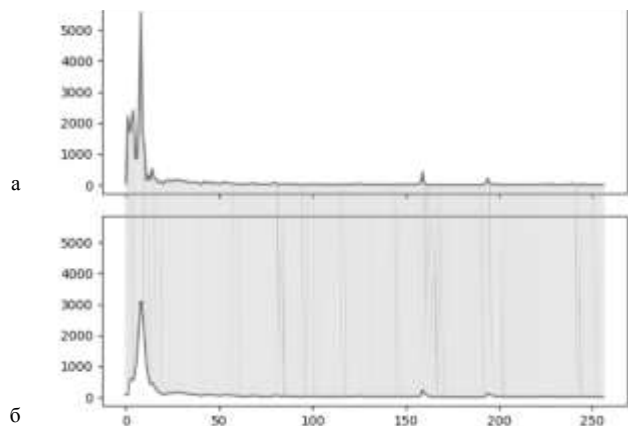


Рис. 5. Работа DTW алгоритма при сравнении спектра треморограммы пациента с паркинсонизмом (а) и шаблона спектра, характерного для болезни Паркинсона (б)

Анализ набора треморограмм показал, что при расстоянии $D \leq 0,1$ можно считать анализируемые графики похожими и, соответственно относить анализируемую треморограмму к группе, характеризующейся этим шаблоном.

IV. ЗАКЛЮЧЕНИЕ

Целью исследования являлся выбор математических методов, на основании которых можно построить обобщенную последовательность действий для машинного анализа треморограмм. Сформированная последовательность опирается на математические методы, адаптированные с учетом специфики исходных биологических сигналов и предполагает выполнение следующих этапов: построение спектра временного ряда треморограммы; сглаживание спектра для приведения его к виду, допускающему сравнение с шаблоном (устранение шумов, сглаживание резких скачков); оценку схожести спектра треморограммы и шаблона, позволяющего отнести результирующую треморограмму к определенной патологической или функциональной группе.

Эксперименты, проведенные на синтетических и реальных данных, показали эффективность разработанного подхода. Исследования, планируемые в дальнейшем, связаны с применением масштабирования

отсчетов временного ряда по оси значений, что позволит повысить точность работы DTW алгоритма и, соответственно, точность оценки функционального состояния человека.

СПИСОК ЛИТЕРАТУРЫ

- [1] Kamble N., Pal P.K. Tremor syndromes: A review // *Neurology India*. 2018. V. 66. Pp 36-47.
- [2] Novak N., Newell K. Physiological Tremor (8–12Hz component) in Isometric Force // *Control Neuroscience Letters*. 2017. V. 641. Pp. 87-93.
- [3] Schaefer L.V., Bittmann F.N. Parkinson patients without tremor show changed patterns of mechanical muscle oscillations during a specific bilateral motor task compared to controls // *Scientific Reports*. 2020. V. 10(1). Pp. 1168.
- [4] Bureneva O., Aleksanyan Z., Safyannikov N. Tensometric tremorography in high-precision medical diagnostic systems // *Medical Devices*. 2018. V. 11. Pp. 321-330.
- [5] Сафьянников Н.М., Буренева О.И., Чепасов А.Е., Алексанян З.А. Изометрический векторный нейроанализатор. // II международная конференция по нейронным сетям и нейротехнологиям (NeuroNT): сборник докладов, 2021. С. 63-66.
- [6] Hari Krishna E., Reddy K. A. Use of Transforms in Biomedical Signal Processing and Analysis in Real Perspective of Fourier Transforms and Current Developments in Superconductivity. London, United Kingdom: IntechOpen, 2021 [электронный ресурс]. Режим доступа: <https://www.intechopen.com/chapters/76938>.
- [7] Golyandina N., Zhigljavsky A. Singular Spectrum Analysis for Time Series. Springer: Berlin, Germany, 2020.
- [8] Acharya U.R., Sree S.V., Alvin A.P., Yanti R., Suri J.S. Application of non-linear and wavelet based features for the automated identification of epileptic EEG signals // *Intern. journal of neural systems*. 2012. V. 22(2):1250002.
- [9] Measuring complexity in different muscles during sustained contraction using fractal properties of SEMG signal / Arjunan S.P., Kumar D.K. // 40th Annual Intern. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC). 2018. Pp. 5656-5659.
- [10] Yuxin Lin, Bingo Wing-Kuen Ling, Nuo Xu, Ringo Wai-Kit Lam, Charlotte Yuk-Fan Ho. Effectiveness analysis of bio-electronic stimulation therapy to Parkinson's diseases via joint singular spectrum analysis and discrete fourier transform approach // *Biomedical Signal Processing and Control*. V. 62. 2020.
- [11] Machine Learning for the Diagnosis of Parkinson's Disease: A Review of Literature // *Frontiers in Aging Neuroscience*. 2021. V. 13:633752.
- [12] Mansur P.H., Cury L.K., Andrade A.O., Pereira A.A., Miotto G.A., Soares A.B., Naves E.L. A review on techniques for tremor recording and quantification // *Critical Reviews in Biomedical Engineering*. 2007. V. 35(5). Pp. 343-362.
- [13] Moving averages, July 2020, DOI: 10.1007/978-1-0716-0737-4_15, In book: Heavy-Tailed Time Series, Rafal Kulik, Philippe Soulier.
- [14] Fried R., Cathrice George A. International Encyclopedia of Statistical Science. Exponential and Holt-Winters Smoothing. Springer: Berlin, Heidelberg, 2014.
- [15] Path clustering using Dynamic Time Warping technique / Siang K.L.Y., Siak Wang Khor. // 8th International Conference on Computing Technology and Information Management (NCM and ICNIT). 2012. Pp. 449-452.
- [16] Derivative Dynamic Time Warping / Keogh E.J., Pazzani M.J. // First SIAM Intern. Conf. on Data Mining (SDM'2001). 2001. V. 1. Pp. 5-7.
- [17] Combinatorial Test Generation for Software Product Lines Using Minimum Invalid Tuples / Yu L., Duan, F., Lei, Y., Kacker R.N., Kuhn, D.R. // IEEE 15th Intern. Symp. on High-Assurance Systems Engineering. 2014. Pp. 65-72.

Концептуальная модель информационной системы сбора, анализа и оценки физического, функционального и психологического состояния человека

Н. О. Сергеев^{1,2}

¹Санкт-Петербургский
государственный
электротехнический
университет «ЛЭТИ»

им. В.И. Ульянова (Ленина);
²Севастопольский
государственный университет
nosergeev@sevsu.ru

Ю. А. Шичкина

Санкт-Петербургский
государственный
электротехнический
университет

«ЛЭТИ» им. В.И. Ульянова
(Ленина)
strange.y@mail.ru

В. И. Шевченко

Севастопольский
государственный университет
VIShevchenko@sevsu.ru

Аннотация. В данной статье рассматривается проектирование концептуальной модели информационной системы сбора, анализа и оценки физического, функционального и психологического состояния человека, позволяющей оценить динамику данных состояний и спрогнозировать их изменения с целью поддержания здоровья персонала на достаточно высоком уровне. Рассматриваются цели и задачи проектирования, методика прогнозирования состояний, предложена структура системы и хранения данных.

Ключевые слова: концептуальная модель, нейронные сети, информационная система, минимизация, прогноз параметров, аппроксимация

I. ВВЕДЕНИЕ

В настоящее время, благодаря общедоступности и развитию персональных вычислительных устройств, таких как смартфоны, планшетные компьютеры, ноутбуки, стало возможным осуществлять мониторинг различных сфер деятельности и состояния человека. Такие устройства хорошо подходят для реализации этих систем, так как обладают необходимой вычислительной мощностью, большим дисплеем и встроенными интерфейсами для обмена данными по радиоканалу для подключения различных внешних устройств и обеспечения доступа в Интернет. Необходимое программное обеспечение позволит анализировать информацию, полученную в результате мониторинга параметров человека.

Тема является актуальной в связи с текущей ситуацией в мире: переход многих сфер деятельности в «удаленный» режим работы способствует снижению уровня физической активности и, как следствие, ухудшению самочувствия и состояния здоровья персонала.

Отслеживание и поддержание здоровья персонала на достаточно высоком уровне, а также прогнозирование возможного изменения состояний персонала в зависимости от различных событий, происходящих в ходе деятельности, является одним из направлений

деятельности предприятий, поскольку проблемы со здоровьем выражаются в снижении производительности, более частому отсутствию на рабочих местах и, как следствие, убыткам, несоблюдениям сроков договоренностей и другим последствиям.

К основным состояниям человека относят физическое, функциональное и психологическое состояния организма:

- Физическое состояние – это состояние организма человека, характеризующееся возможностями адаптироваться к различным факторам среды обитания, уровнем физического развития, физической и функциональной подготовленностью организма к выполнению физических нагрузок.
- Функциональное состояние – это интегральный комплекс характеристик тех функций и качеств человека, которые прямо или косвенно обуславливают выполнение любой деятельности.
- Психологическое состояние – целостная характеристика психической деятельности за определенный период времени, показывающая своеобразие протекания психических процессов в зависимости от отражаемых предметов и явлений действительности, предшествующего состояния и свойств личности.

II. ИНФОРМАЦИОННАЯ СИСТЕМА СБОРА, АНАЛИЗА И ОЦЕНКИ ФИЗИЧЕСКОГО, ФУНКЦИОНАЛЬНОГО И ПСИХОЛОГИЧЕСКОГО СОСТОЯНИЯ ЧЕЛОВЕКА

A. Цели и задачи при разработке системы

Информационная система (ИС) является неотъемлемой частью функционирования любой сферы деятельности человека. Системность подходов к проектированию ИС является обсуждаемым вопросом [1, 2]. ИС, предназначенная для сбора, анализа и оценки физического, функционального и психологического состояния человека должна быть модульной, поддерживать работу с данными и

знаниями напрямую с конечным пользователем и решать следующие основные задачи:

- иерархическое хранение данных о проведенных тестированиях;
- автоматическая оценка результатов тестирования по соответствующим нормативам, с учетом возраста, пола. Автоматический расчет интегральной оценки физического состояния в баллах (по 5-балльной шкале) с использованием результатов тех тестов, которые оцениваются в баллах;
- автоматический расчет обобщенных оценок физического, функционального и психологического состояния по группам исследуемых;
- автоматическое формирование рейтингов групп исследуемых;

- внесение в библиотеку, если это необходимо, специализированных тестов, при условии их предоставления специалистами;
- распределенный, защищенный доступ к результатам тестирования и их оценки;
- автоматическое построение графиков и таблиц, позволяющих анализировать динамику изменения показателей тестирования, с возможностью сравнения оценок показателей между собой;
- адаптация механизма загрузки результатов тестирования.

В. Концептуальная модель информационной системы

Исходя из поставленных целей, требований и задач была сформирована структура будущей информационной системы (рис. 1).

На основе разработанной структуры была предложена схема хранения данных информационной системы (рис. 2).

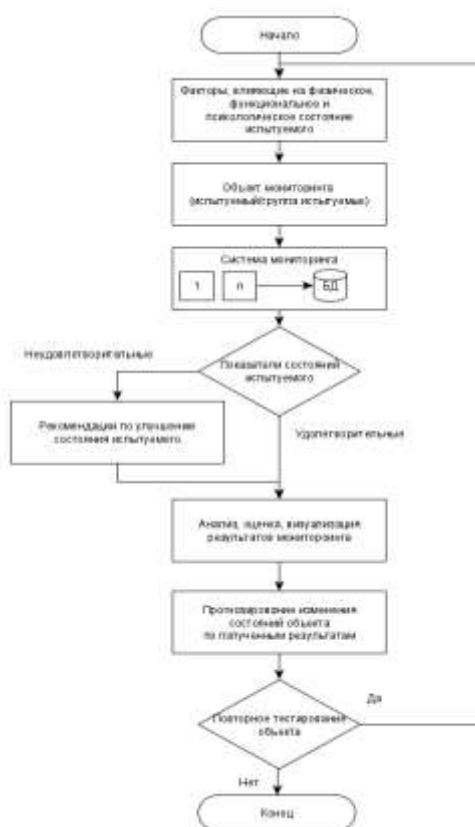


Рис. 1. Структура информационной системы сбора, анализа и оценки физического, функционального и психологического состояния человека, где n – результаты тестирования объекта исследования, БД – база данных значений результатов

Для прогнозирования и оценки динамики изменения состояний группы исследуемых на протяжении длительного времени наблюдения необходимо разработать модели и методы сбора и анализа состояний человека.

Универсальной теорией, которая занимается вопросами аппроксимации и прогнозирования значений функций от нескольких переменных, является теория

искусственного интеллекта – раздел «нейронные сети» [3].

Моделью прогнозирования в области искусственного интеллекта является нейронная сеть прогнозирования [4]. Нейронная сеть представляет собой систему уравнений, включающую в себя весовые коэффициенты, которые преобразуют функцию таким образом, чтобы ее значение от рассматриваемых переменных принимало аргумент выходного параметра. Процесс моделирования

нейронной сети (прогнозирование выходного параметра) осуществляется путем выбора и решения уравнения системы в зависимости от текущих значений входных переменных (1):

$$y_{inj} = \frac{1}{1 + \exp(-\beta \cdot ((w_{s11} \cdot x_{11}) + (w_{s12} \cdot x_{12}) + \dots + (w_{s1n} \cdot x_{1n})))} \cdot \frac{1}{1 + \exp(-\beta \cdot ((w_{s21} \cdot x_{21}) + (w_{s22} \cdot x_{22}) + \dots + (w_{s2n} \cdot x_{2n})))} \cdot \dots \cdot \frac{1}{1 + \exp(-\beta \cdot ((w_{sm1} \cdot x_{m1}) + (w_{sm2} \cdot x_{m2}) + \dots + (w_{smn} \cdot x_{mn})))}, \quad x_i = x, \quad (1)$$

где i – номер входного (выходного) параметра слоя нейронной сети;

n – номер персептрона (нейронной связи);

j – номер итерации (эпохи) обучения;

w – значение весового коэффициента соответствующего входного параметра x_i , изменение значения которого осуществляется на каждой итерации (эпохи) обучения j с целью минимизации отклонения между ответом нейронной сети по значению выходного параметра y_{inj} и фактическим значением выходного параметра обучающей выборки a_{in} ;

x – значение переменной (входного параметра), физически влияющей на выходной параметр y_{inj} ;

m – количество персептронов (нейронных связей) слоя нейронной сети.

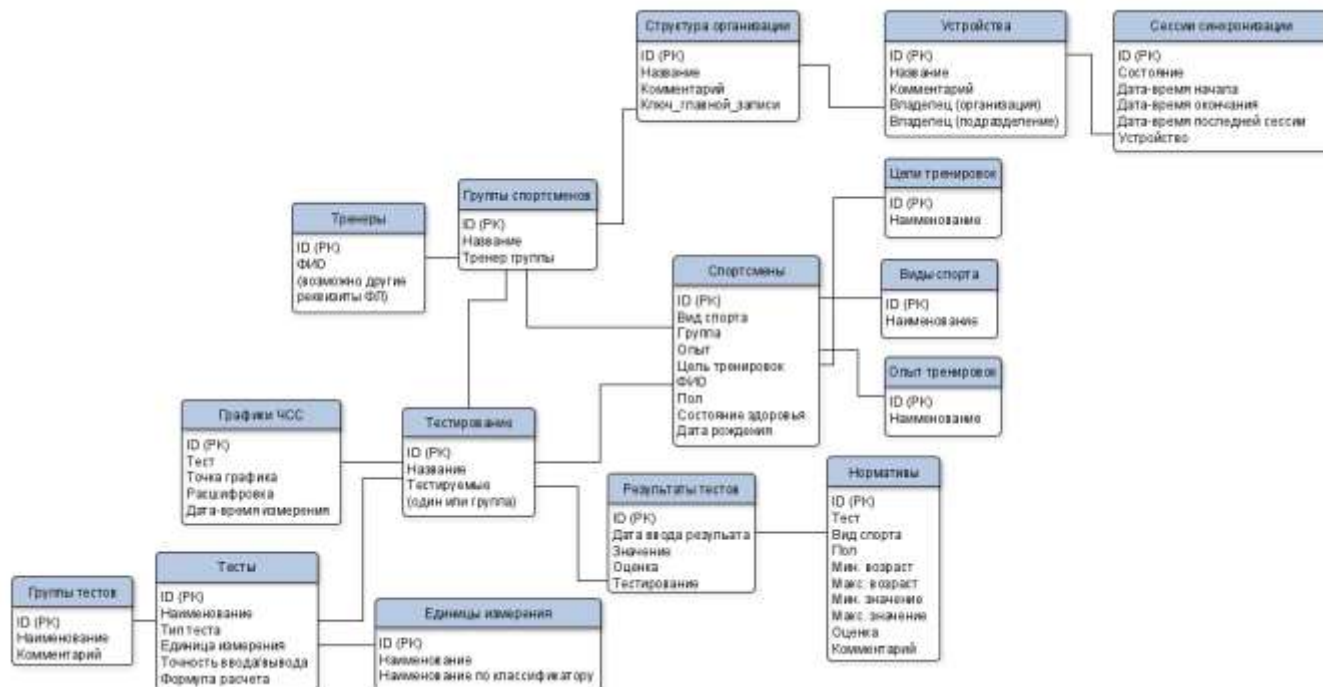


Рис. 2. Схема хранения данных информационной системы

Процесс обучения нейронной сети сводится к задаче минимизации функции. Наиболее эффективным методом обучения нейронной сети, предназначенной для прогноза выходных параметров, является алгоритм минимизации Левенберга–Марквардта. Данный алгоритм основан на синтезе двух методов минимизации функции – на алгоритме наискорейшего спуска и методе Гаусса–Ньютона, которые в отдельности имеют свои положительные стороны и недостатки [5, 6].

По окончании процесса минимизации функции вследствие снижения эффективности обучения нейронной сети или достижения допустимого предела минимизации функции проводится регрессионный линейный анализ слоя, при этом в выходном слое используют линейную функцию активации с целью дальнейшего определения коэффициента корреляции и детерминации нейронной сети [7]. При достижении высоких значений коэффициента детерминации обученная нейронная сеть позволяет спрогнозировать значения диагностического параметра в зависимости от входных данных в рамках аппроксимированной выборки, а значит, рассматриваемая модель способна к комплексному контролю технического состояния

различных узлов технических систем, что доказывает универсальность подхода контроля и диагностики при использовании рассматриваемой модели.

III. ЗАКЛЮЧЕНИЕ

В данном исследовании описана концептуальная модель информационной системы, позволяющей выполнить сбор, анализ и оценку показателей физического, функционального и психологического состояния человека или группы исследуемых. Был описан подход, позволяющий оценить динамику изменения этих состояний, а также спрогнозировать их дальнейшие показатели. Результаты исследований и выявленные методы будут применяться при дальнейшем проектировании базы хранения и обработки данных, визуализации результатов тестирования и сравнения показателей между собой, автоматической оценки с использованием разработанных нормативов оценивания, а также построения рейтинга групп исследуемых.

СПИСОК ЛИТЕРАТУРЫ

- [1] Охтилев П.А., Бахмут А.Д., Крылов А.В., Охтилев М.Ю. Соколов Б.В. Подход к оцениванию структурных состояний сложных организационно-технических объектов на основе

обобщенных вычислительных моделей // Научные технологии в космических исследованиях Земли. 2017. Т. 9. № 5. С. 73–82.

- [2] Егорова А.А., Козлов С.А. Информационные системы: Методы и средства проектирования // Научный вестник МГТУ ГА, 2006. Вып. 105. С. 84-92
- [3] Андрейченко Д.К., Жадаев Ф.М. Обучение нейросетевых регуляторов для стабилизации комбинированных динамических систем [Текст] // Известия Саратовского университета. Новая серия. Сер.: Математика. Механика. Информатика / Саратовский нац. исследоват. гос. ун-т им. Н.Г. Чернышевского. Саратов. 2018. № 3. С. 354–360.
- [4] Айзикович А.А., Усынин А.Ю. Применение персептрона в цифровой фильтрации сигналов [Текст] // Известия Института математики и информатики Удмуртского государственного университета / Удмуртский гос. ун-т. Ижевск. 2006. № 2. С. 109–112.
- [5] Diene O., Bhaya A. Conjugate gradient and steepest descent constant modulus algorithms applied to a blind adaptive array, Signal Processing, Elsevier BV, Amsterdam, 2010, no. 90, pp. 2835–2841.
- [6] Chen J., Li W. Convergence of Gauss-Newton's method and uniqueness of the solution, Applied Mathematics and Computation, Elsevier BV, Amsterdam, 2005, no. 170, pp. 686–705.
- [7] Maric I. Optimization of self-organizing polynomial neural networks, Experts systems with applications, Elsevier BV, Amsterdam, 2013, no. 40, pp. 4528–4538.

СОДЕРЖАНИЕ

Секция 1. Концепции построения гибридного интеллекта

<i>Адаптация обучения с подкреплением для библиотеки федеративного обучения FL4J</i> Азаревич А.Д., Ефремов М.А.	3
<i>Исследование локальных и глобальных методов объяснимого искусственного интеллекта.</i> <i>Метрики для оценки методов объяснения</i> Шевская Н.В., Шишкин И.В., Степанов В.Д., Облизов А.Д.	7
<i>Accelerating Neural Network Training Process on Multi-Core Machine Using OpenMP</i> Mohammed Omar T., Paznikov A.A., Gorlatch Sergei.....	12
<i>Подходы к классификации этических проблем нейросетей</i> Покровская Н.Н.	17

Секция 2. Технологии искусственного интеллекта и их приложения

<i>Нейросетевое моделирование динамики показателей научно-технической безопасности регионов России</i> Митяков Е.С. Артемова С.В., Ладынин А.И., Шмелева А.Г., Каменская М.А., Митяков С.Н.	21
<i>Применение нейронных сетей в датчиках оценки влажности материала</i> Артемова С.В., Ладынин А.И., Митяков Е.С., Чиев Ву Чи, Каменская М.А., Лапаев Д.Н.	25
<i>COVID скрининг, основанный на применении нейросетевой классификации спектров выдоха</i> Кугаевских А.В.	30
<i>Подход к решению оптимизационной задачи в системе управления балансом газа региональной газотранспортной сети</i> Петрова А.К.	33
<i>Prediction of Next App in OS</i> Ma Ruimin, Zhang Yuyi, Liu Jing, Petrosian Ovanes, Krinkin Kirill	37
<i>Применение нейросетевого регулятора для многоканальной системы с нестационарными непараметрическими неопределенностями</i> Воевода А.А., Шипагин В.И.	41
<i>Система детектирования и анализа объектов на КТ-снимках в урологии</i> Руденко М.А., Руденко А.В., Крапивина М.А., Лисовский В.С.	47
<i>Анализ архитектур умножителей для аппаратной реализации нейронных сетей</i> Миронов С.Э., Буренева О.И., Милакин А.Д.	52

Секция 3. Управление данными и организация вычислений в интеллектуальных системах

<i>Нейросетевой подход к прогнозированию рынка электроэнергии с использованием разноточных данных среди объясняющих переменных</i> Косарев В.С.	56
<i>Корреляционный дискриминатор сигналов в классе быстрых нейронных сетей</i> Дорогов А.Ю.	59
<i>Модель распознавания информационных угроз с использованием рекуррентной нейронной сети</i> Шичкина Ю.А., Фаткиева Р.Р., Пузако И.А.	62
<i>Применение рекуррентных нейросетей с управляемыми элементами для повышения точности определения звуковых событий в туманной вычислительной среде</i> Субботин А.Н., Жукова Н.А., Анаам Ф.А.	67
<i>Машинные методы анализа треморограмм для оценки функционального состояния человека</i> Сафьянников Н.М., Буренева О.И., Алексанян З.А.	71
<i>Концептуальная модель информационной системы сбора, анализа и оценки физического, функционального и психологического состояния человека</i> Сергеев Н.О., Шичкина Ю.А., Шевченко В.И.	75

NEURONT.ETU.RU

197022, Россия,
Санкт-Петербург,
ул. Профессора Попова,
д.5, лит. Ф,
СПбГЭТУ «ЛЭТИ»
Тел.: +7 (812) 346-46-37
E-mail: IRVC.eltech@mail.ru

