

РЕШЕНИЕ ЗАДАЧИ АВТОРСКОГО ПРОФИЛИРОВАНИЯ РУССКОЯЗЫЧНЫХ ТЕКСТОВ ПРИ ПОМОЩИ СПАЙКОВЫХ НЕЙРОННЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ НОВОГО МЕТОДА КОДИРОВАНИЯ ТЕКСТОВЫХ ДАННЫХ

© 2021 г. А. Г. Сбоев^{1,2,*}, Р. Б. Рыбка¹, Ю. А. Давыдов¹, Д. С. Власов¹

¹ НИЦ “Курчатовский институт”, Москва, 123182 Россия

² Национальный исследовательский ядерный университет “МИФИ”, Москва, 115409 Россия

*e-mail: sag111@mail.ru

Поступила в редакцию 31.01.2022 г.

После доработки 01.02.2022 г.

Принята к публикации 08.02.2022 г.

В данной работе исследуется возможность применения спайковых нейронных сетей к решению задачи авторского профилирования текстов на русском языке на примере задач определения пола и возраста автора, а также задачи различения текстов, сгенерированных алгоритмически и написанных человеком. Разрабатывается метод преобразования текстов, закодированных последовательностями векторов, полученных при помощи языковой модели FastText, в спайковые последовательности. В рамках поставленной задачи используются два корпуса документов, первый из которых характеризуется большим количеством коротких текстов, второй же — в четыре раза меньшим количеством текстов существенно большей длины. Такой выбор корпусов позволяет сделать выводы об ограничениях и возможностях предложенного способа кодирования. Проведенные эксперименты показывают, что предложенный метод кодирования текстов в сочетании с используемой в задаче спайковой топологией успешно решает поставленные перед ней задачи: полученные точности соответствуют baseline-модели (LinearSVC) на обоих корпусах по метрике f1-score.

Ключевые слова: спайковые нейронные сети, импульсные нейронные сети, обработка естественного языка, классификация текстов, авторское профилирование, локальные правила обучения

DOI: 10.56304/S2304487X21060092

1. ВВЕДЕНИЕ

Спайковые нейронные сети, относимые к третьему поколению алгоритмов машинного обучения и моделирующие процессы, протекающие в биологических нейронах, представляют существенный исследовательский интерес, поскольку характеризуются большей энергоэффективностью в сравнении с формальными нейросетевыми алгоритмами.

Информация внутри спайковых сетей представлена в виде временных последовательностей импульсов (спайков), распространяющихся по синапсам сети. Благодаря этому, спайковые сети обладают внутренним представлением о времени и о порядке событий, протекающих во входной последовательности. Эта особенность делает их применение для анализа последовательностей данных (временных рядов, текстов, последовательностей изображений и т.д.) весьма перспективной задачей.

В настоящей работе исследуется принципиальная возможность решения задачи классификации текстов, закодированных при помощи векторизованных последовательностей слов, при помощи спайковой нейронной сети. В качестве входного представления слов в тексте используются вектора, построенные при помощи языковой модели FastText: такой метод, с одной стороны, не требует сложной предобработки текста и конструирования полезных признаков, а с другой — дает возможность получить представление, сохраняющее синтаксические, лексические и морфологические свойства слов в тексте. Несмотря на это, использование векторов FastText имеет ряд сложностей в контексте последующего анализа при помощи спайковой сети: для перевода векторов в последовательности спайков используются статистические методы, полагающиеся на интерпретацию значений векторов в качестве вероятностных характеристик последовательностей. Наиболее часто используемым является кодирование входных векторов при помощи

пуассоновских последовательностей, в котором элементы входного вектора используются как частоты распределений Пуассона, из которых генерируются спайковые последовательности заданной длины.

В данной работе предлагается метод кодирования объектов, представленных последовательностями векторов признаков, при помощи спайковых последовательностей. Результаты апробации этого метода при решении задачи классификации текстов, в частности авторского профилирования, показывают его эффективность и открывают возможности исследования более широкого класса задач: анализ последовательностей и временных рядов.

2. КОРПУСА ДАННЫХ

Для изучения возможностей, представляемых спайковыми нейронными сетями в контексте решения задачи авторского профилирования, было использовано два корпуса данных.

Первый из них, обозначаемый HRI-C, представляет собой набор команд мобильному роботизированному устройству. Полная версия этого корпуса была описана в статье [2]. В данной работе используется его подмножество, состоящее из 4742 команд и их классов: первый класс представляет собой команды, сгенерированные при помощи компьютерной программы по заданным шаблонам, второй класс — команды, написанные пользователями краудсорсинговой платформы на основе предоставленных им сгенерированных команд. Максимальная длина команды в корпусе равна 21 слову, что делает данный корпус крайне удобным для оценки работоспособности предложенного спайкового алгоритма благодаря большому количеству примеров в сравнении с размерностью входного пространства закодированных слов текстов (признаков).

Второй — RusPersonality (RusPer) — состоит из 1150 сочинений, написанных студентами и в дальнейшем размеченных экспертами по полу автора. Количество примеров первого класса равно количеству примеров второго класса (575 текстов), максимальная длина текста составляет 886 слов. Этот корпус является значительно более сложным с точки зрения анализа при помощи алгоритма на базе спайковой сети: при сравнительно небольшом количестве примеров количество векторов слов в каждом в 40 раз больше, чем в предыдущем корпусе. Таким образом, получение на этом корпусе точности, сравнимой с baseline-алгоритмом, описанным далее, будет означать, что предложенный спайковый алгоритм может применяться для решения задач классификации текстов.

Оба корпуса были полностью очищены от знаков препинания и переведены в нижний регистр, после чего были трансформированы в последовательности векторов (по 300 действительных чисел в каждом) при помощи языковой модели FastText. При этом длины векторизованных текстов были унифицированы при помощи техники увеличения длины более коротких текстов до максимальной при помощи нулевых векторов той же размерности, что и вектора FastText.

Поскольку предложенный метод кодирования при помощи спайковых последовательностей предполагает интерпретацию входных векторов q в качестве вероятностей, вектора текстов были предварительно смещены в область $q \in [0, 1]^N$, где N — длина текста, при помощи следующего линейного преобразования:

$$\hat{q} = q + |\min_{j \in [0, 300)}(q)| \cdot I,$$

$$\underline{q} = \frac{\hat{q}}{\max_{j \in [0, 300)}(\hat{q})}.$$

Здесь индекс j нумерует признаки, входящие в состав вектора слова (300 — длина одного вектора слова FastText), I — единичный вектор длины 300.

Такое преобразование позволяет избавиться от отрицательных значений и перевести все вектора текстов в диапазон от 0 до 1 с сохранением структуры распределения, а значит, позволяет избежать потери информации об устройстве векторного пространства текстов.

3. МЕТОД КОДИРОВАНИЯ ТЕКСТОВЫХ ДАННЫХ ПРИ ПОМОЩИ ПОСЛЕДОВАТЕЛЬНОСТЕЙ СПАЙКОВ

Смещенные последовательности векторов слов, полученные по методу, описанному выше, интерпретировались как меняющиеся во времени вероятности стохастических процессов Бернулли. Чтобы проиллюстрировать разработанный метод, рассмотрим элемент вектора слова, находящийся на позиции $j_0 \in [0, 300)$. Предположим, что для первого слова в тексте этот элемент имеет значение $q_{j_0}^0$, для второго слова в тексте — $q_{j_0}^1$ и т.д. Для работы алгоритма выберем параметр s и будем считать, что вероятность получить спайк в промежуток времени от t_k до t_{k+s} остается постоянной и равной $q_{j_0}^{t_k}$. Выполняя сэмплирование из распределения Бернулли для каждого элемента каждого вектора текста по описанному правилу, мы получим набор спайковых последовательностей $S \in \{0, 1\}_{j,t}^t$, $j \in [0, 300)$, $t \in [0, N \cdot s]$, характеризующих кодируемый текст.

Такой метод кодирования позволяет сохранить последовательность слов анализируемого

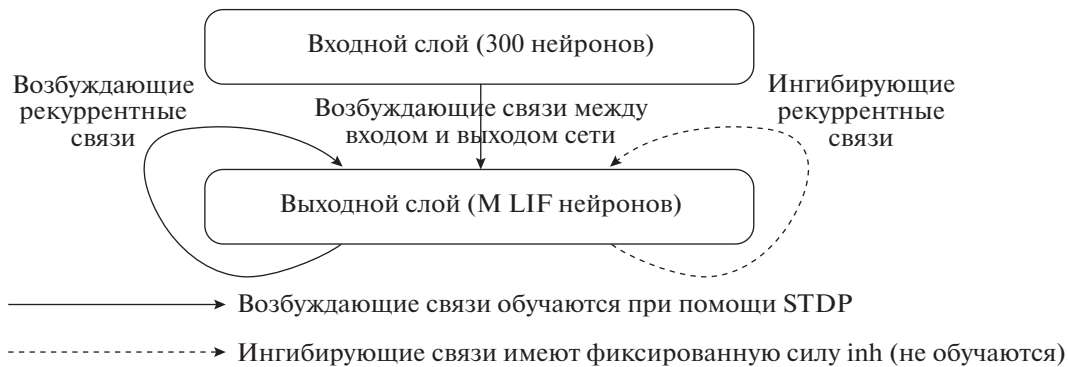


Рис. 1. Топология используемой в работе спайковой сети.

текста, при этом количество входных нейронов спайковой сети сохраняется равным размерности вектора признаков слова (300 значений), полученного с использованием FastText модели. Стоит отметить, что длина каждой спайковой последовательности при этом зависит от выбранного параметра s , а значит, при выборе сравнительно большего его значения (что соответствует более точной аппроксимации соответствующего значения внутри вектора слова) время симуляции (время подачи каждого примера в спайковую сеть), равное $N \cdot s$, может быть слишком продолжительным для эффективного анализа. Таким образом, необходимо выбирать s достаточно большим, чтобы сохранить необходимое количество полезной информации, и при этом не слишком большим для того, чтобы спайковая сеть сохраняла возможность эффективно запоминать и преобразовывать получаемые спайковые паттерны. Это является основным техническим ограничением, накладываемым на предложенный метод.

4. ТОПОЛОГИЯ СЕТИ И АЛГОРИТМЫ

В данной работе используется двухслойная рекуррентная спайковая сеть, топология которой приведена на рис. 1. Первый слой, длина которого совпадает с длиной вектора признаков слова, служит для подачи примеров в спайковую сеть в виде последовательностей спайков, сгенерированных по алгоритму, описанному в предыдущей секции. Второй слой состоит из M leaky integrate-and-fire (LIF) нейронов [3], связанных с входным слоем при помощи синаптических связей, обучаемых при помощи локального правила STDP [4]. LIF-модель спайкового нейрона является одной из наиболее часто используемых в работах, посвященных спайковым нейронным сетям. Эта же характеристика справедлива и для STDP, что делает обе модели хорошим базовым выбором для проверки пригодности предложенного метода кодирования.

Нейроны внутри выходного слоя связаны фиксированными ингибирующими связями с силой inh по принципу “каждый с каждым”, исключая связи нейронов с самими собой. Такие тормозящие связи помогают обеспечить конкуренцию между выходными нейронами и обеспечить более эффективный анализ входного пространства. Подобный механизм, продемонстрированный в работе [5] и многократно использованный в других работах, доказал свою эффективность на различных классах задач. Наконец, выходной слой обладает еще одним набором обучаемых при помощи STDP рекуррентных связей (при этом для этого типа связей связь нейрона с самим собой не запрещается). Эти связи служат для запоминания временных паттернов, характеризующих последовательности слов в тексте.

Спайковая сеть, приведенная на рис. 1, была реализована при помощи открытой библиотеки BindsNET [6]. В качестве модели для сравнения используется машина опорных векторов с линейным ядром из библиотеки scikit-learn [7], для обучения которого использовался вектор текста, полученный усреднением вектор слов. Выбор более сложной модели, к примеру, сверточно-рекуррентной нейронной сети или сети, построенной на топологии Transformer [8], позволит получить существенно более высокую точность на обоих корпусах.

5. ЭКСПЕРИМЕНТЫ

Эксперименты на обоих корпусах проводились по следующей методике: корпус данных разбивался на пять подмножеств (фолдов) с сохранением распределения по классам, после чего один из фолдов выбирался в качестве тестового множества. Оценка точности классификации проводилась по метрике $f1_score$, учитывающей количество ложноположительных и ложноотрицательных предсказаний наравне с классической точностью.

Таблица 1. Результаты экспериментов по решению задачи авторского профилирования

Корпус	Модель	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	AVG
HRI-C	RISNN	90	88	90	92	90	90
	LinearSVC	93	92	95	96	92	94
RusPer	RISNN	66	64	60	65	62	64
	LinearSVC	66	68	63	62	61	64

В экспериментах спайковая сеть использовалась со следующими гиперпараметрами, определенными предварительно путем итеративного поиска:

1. Для корпуса HRI-C:
 - a. $S = 9$ (значимость);
 - b. $M = 1600$ (число выходных нейронов);
 - c. $Inh = -103.06$ (сила ингибирующих связей);
 - d. $Norm_rec = 35.71$ (нормировка рекуррентных весов);
 - e. $Norm_exc = 35.71$ (нормировка весов между слоями);
 - f. $I_tc_trace = 34.49$ мс (характерное время затухания следовой переменной (trace) для входного слоя);
 - g. $Thresh = -51.9$ (порог мембранного потенциала нейронов выходного слоя);
 - h. $Refrac = 9$ мс (время рефрактерности нейронов выходного слоя);
 - i. $Tc_decay = 81.63$ мс;
 - j. $Tc_trace = 34.49$ мс (характерное время затухания следовой переменной (trace) для выходного слоя);
 - k. $Nu_rec = 0.0061$ (скорость обучения (learning rate) рекуррентных связей);
 - l. $Nu_exc = 0.0032$ (скорость обучения связей между входом и выходом).

2. Для корпуса RusPer:

- a. $S = 8$
- b. $M = 1200$
- c. $Inh = 8.77$
- d. $Norm_rec = 57.35$
- e. $Norm_exc = 45.96$
- f. $I_tc_trace = 20$ мс
- g. $Thresh = -37.26$
- h. $Refrac = 8.31$ мс
- i. $Tc_decay = 18.66$ мс
- j. $Tc_trace = 20$ мс
- k. $Nu_rec = 0.0051$
- l. $Nu_exc = 0.019$

3. Общие параметры для обоих корпусов:

- a. $Rest = reset = -65$ (нижний уровень мембранного потенциала)

- b. $Dt = 1$ мс (шаг по времени во время симуляции)

Кроме того, для корпуса RusPers при обучении и тестировании спайковой сети использовались только первые 399 слов в явном виде, при этом 400-й вектор представлял собой усредненный вектор всех оставшихся слов текста. Такой подход обусловлен повышением вычислительной эффективности алгоритма и не вносит существенных искажений, поскольку лишь незначительная часть текстов в корпусе превосходит длину в 400 слов.

Декодирование выхода спайковой сети проводилось при помощи логистической регрессионной модели, обученной на частотах испускания спайков выходных нейронов, записанных в процессе обучения. Сама использованная регрессионная модель была взята из библиотеки `scikit-learn` и применялась с параметрами по умолчанию. Используемый для сравнения алгоритм на основе машины опорных векторов с линейным ядром (LinearSVC) тестировался по тому же принципу с использованием параметров по умолчанию.

В таблице 1 приведены точности спайковой сети (f1-score, %), декодированной при помощи логистической регрессии (обозначенной RISNN), а также baseline-алгоритма.

6. ОБСУЖДЕНИЕ

Из приведенных выше экспериментальных результатов можно заключить, что предложенный метод кодирования текстовых данных, представленных в виде последовательностей векторов действительных значений, в сочетании с предложенной спайковой топологией, позволяет успешно решать задачи авторского профилирования как для коротких текстов с большим количеством примеров, так и для корпусов небольшого объема, содержащих сравнительно длинные последовательности.

Для обоих корпусов предложенный метод позволяет получить точности либо идентичные baseline (64% на RusPer), либо незначительно меньшие (90% против 94% на корпусе HRI-C). Дальнейшее усложнение спайковой топологии, а также использование методов аугментации данных для корпусов небольшого объема, может поз-

волишь существенно улучшить полученный результат.

ВЫВОДЫ

В целом можно заключить, что предложенный в данной работе метод спайкового кодирования текстов, представленных в виде последовательностей векторов FastText, вкупе с применением рекуррентной спайковой сети и частотного декодирования с использованием логистической регрессии позволяет успешно решать задачи авторского профилирования текстов на русском языке. Точности, полученные в данной работе, соответствуют таковым для машины опорных векторов с линейным ядром (90% на корпусе HRI-C и 64% на корпусе RusPers). Дальнейшие исследования будут направлены на исследование возможности использования алгоритмов кодирования последовательной в данных для анализа с использованием спайковых нейронных сетей.

БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РФФИ № 18-29-10084 мк с использованием оборудования центра коллективного пользования “Комплекс моделирования и обработки данных исследовательских установок мега-класса” НИЦ “Курчатовский институт”, <http://ckp.nrcki.ru/>.

СПИСОК ЛИТЕРАТУРЫ

1. *Bojanowski P., Grave E., Joulin A., Mikolov T.* Enriching word vectors with subword information // Transactions of the Association for Computational Linguistics, 2017. V. 5. P. 135–146.
2. *Сбоев А.Г., Давыдов Ю.А., Рыбка Р.Б.* Нейросетевая модель для перевода текстовых команд мобильному роботу на естественном русском языке в семиотический формат RDF. Сб. научных трудов VII международной конференции “Лазерные, плазменные исследования и технологии-ЛАПАЗ-2021”, Москва, 2021. С. 138–139.
3. *Abbott L.F.* Lopicque’s introduction of the integrate-and-fire model neuron (1907) // Brain research bulletin, 1999. V. 50 (5–6). P. 303–304.
4. *Sjöström J., Wulfram G.* Spike-timing dependent plasticity // Scholarpedia, 2010. V. 5 (2). P. 1362.
5. *Diehl P.U., Cook M.* Unsupervised learning of digit recognition using spike-timing-dependent plasticity // Frontiers in computational neuroscience, 2015. V. 9. P. 99.
6. *Hazan H., Saunders D., Khan H., Patel D. et al.* Bindsnet: A machine learning-oriented spiking neural networks library in Python // Frontiers in Neuroinformatics, 2018. V. 12. P. 89.
7. *Pedregosa F., Varoquaux G., Gramfort A. et al.* Scikit-learn: Machine learning in Python // Journal of Machine Learning Research, 2011. V. 12. P. 2825–2830.
8. *Vaswani A., Shazeer N., Parmar N. et al.* Attention is all you need. / 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

Vestnik Nacional'nogo Issledovatel'skogo Yadernogo Universiteta “MIFI”, 2021, vol. 10, no. 6, pp. 523–528

Solution of the Problem of Author’s Profiling of Russian Texts Using Spike Neural Networks and a New Method of Coding Text Data

A. G. Sboev^{a,b,#}, R. B. Rybka^a, Y. A. Davydov^a, and D. S. Vlasov^a

^a National Research Center Kurchatov Institute, Moscow, 123182 Russia

^b National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, 115409 Russia

[#]e-mail: sag111@mail.ru

Received January 31, 2022; revised February 1, 2022; accepted February 8, 2022

Abstract—The possibility of using spike neural networks to solve the problem of author profiling texts in Russian has been examined on the example of the tasks of determining the gender and age of the author, as well as the task of distinguishing texts generated algorithmically and written by a person. A method has been developed to convert texts encoded by sequences of vectors obtained using the FastText language model into spike sequences. Within the framework of the task, two bodies of documents are used, the first of which is characterized by a large number of short texts, the second by four times fewer texts of significantly longer length. Such a choice of enclosures allows us to draw conclusions about the limitations and possibilities of the proposed coding method. The experiments show that the proposed text encoding method in combination with the spike topology used in the problem successfully solves the tasks assigned to it: the accuracy obtained corresponds to the baseline model (LinearSVC) on both cases according to the f1-score metric.

Keywords: spike neural networks, impulse neural networks, natural language processing, text classification, author profiling, local learning rules

DOI: 10.1134/S2304487X21060092

REFERENCES

1. Bojanowski P., Grave E., Joulin A., Mikolov T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 2017, vol. 5. pp. 135–146.
2. Sboev A.G., Davydov Yu.A., Rybka R.B. [A neural network model for translating text commands to a mobile robot in natural Russian language into semiotic RDF format]. *Sb. nauchnyh trudov VII mezhdunarodnoj konferencii "Lazernye, plazmennye issledovaniya i tekhnologii-LAPLAZ-2021"* [Sat. scientific papers of the VII International Conference "Laser, Plasma Research and Technologies-LAPLAZ-2021"]. Moscow, 2021. pp. 138–139 (In Russian).
3. Abbott L.F. Lapique's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 1999, vol. 50 (5–6), pp. 303–304.
4. Sjöström J., Wulfram G. Spike-timing dependent plasticity. *Scholarpedia*, 2010, vol. 5 (2), pp. 1362.
5. Diehl P.U., Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 2015, vol. 9, pp. 99.
6. Hazan H., Saunders D., Khan H., Patel D. et al. Bindsnet: A machine learning-oriented spiking neural networks library in Python. *Frontiers in Neuroinformatics*, 2018, vol. 12, pp. 89.
7. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011, vol. 12, pp. 2825–2830.
8. Vaswani A., Shazeer N., Parmar N. et al. Attention is all you need. *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.