

## ТЕХНИЧЕСКИЕ НАУКИ



УДК 004.032.26

### Сравнение сверточной и рекуррентной архитектур нейронных сетей при решении задачи анализа тональности текста

**К.В. Трифонов**

Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

#### Аннотация

В работе проведено сравнение рекуррентной архитектуры нейронной сети BiRNN и сверточной архитектуры нейронной сети textCNN. Исследуется одна из фундаментальных задач обработки естественного языка, а именно задача анализа тональности текста. Для решения этой задачи разработано программное средство, с помощью которого были получены результаты, позволяющие сказать, что сверточная нейросеть архитектуры textCNN является более эффективной. Результатом работы авторов является обоснование выбора наиболее эффективной архитектуры нейронной сети, которая позволяет решать практические задачи с большей точностью.

**Ключевые слова:** задача анализа тональности текста, нейронная сеть, textCNN, BiRNN

**Для цитирования.** Трифонов К.В. Сравнение сверточной и рекуррентной архитектур нейронных сетей при решении задачи анализа тональности текста. Молодой исследователь Дона. 2024;9(2):41–44.

### Comparison of Convolutional and Recurrent Neural Network Architectures for Solving the Problem of Text Sentiment Analysis

**Kirill V. Trifonov**

Don State Technical University, Rostov-on-Don, Russian Federation

#### Abstract

The paper compares the recurrent architecture of the BiRNN neural network and the convolutional architecture of the textCNN neural network. One of the fundamental tasks of natural language processing is investigated, namely the task of analyzing the tonality of the text. To solve this problem, a software tool has been developed, with the help of which the results were obtained. This allows us to say that the convolutional neural network of the textCNN architecture is more efficient. The result of the authors' work is the justification for choosing the most effective neural network architecture, which allows solving practical problems with greater accuracy.

**Keywords:** text sentiment analysis task, neural network, textCNN, BiRNN

**For citation.** Trifonov KV. Comparison of Convolutional and Recurrent Neural Network Architectures for Solving the Problem of Text Sentiment Analysis. *Young Researcher of Don*. 2024;9(2):41–44.

**Введение.** Анализ настроений (интеллектуальный анализ мнений или искусственный интеллект эмоций) — это использование обработки естественного языка, анализа текста, компьютерной лингвистики и биометрии для систематической идентификации, извлечения, количественной оценки и изучения аффективных состояний и субъективной информации. Анализ настроений широко применяется для анализа материалов клиентов (обзоры и ответы на опросы онлайн и в социальных сетях), а также сведений в системе здравоохранения для различных приложений. Такие приложения используются в настоящее время достаточно широко, начиная от маркетинга и обслуживания клиентов, заканчивая использованием в клинической медицине. Решением задачи является определение того, какую эмоцию передаёт текст, написанный человеком. В настоящее время нет однозначного ответа на вопрос о том, какая архитектура лучше подходит для решения поставленной задачи. Поэтому одной из известных задач компьютерной лингвистики в настоящее время является задача анализа тональности текста [1]. Целью данной работы является сравнение рекуррентной архитектуры нейронной сети BiRNN [2] и сверточной архитектуры нейронной сети textCNN [3] при решении задачи анализа тональности текста.

**Основная часть.** Одним из основных показателей эффективности нейронной сети в задаче интеллектуального анализа текста является её точность [4]. Точность — это доля правильных прогнозов, сделанных моделью. Для бинарной классификации у нейросети существует четыре варианта классификации результата, а именно ИП — истинный положительный результат, ИЛ — истинный ложный результат, ЛИ — ложный истинный результат и ЛО — ложный отрицательный результат. Таким образом, точность может быть рассчитана по следующей формуле:  $\text{точность} = (\text{ИП} + \text{ИЛ}) / (\text{ИП} + \text{ИЛ} + \text{ЛИ} + \text{ЛО})$ . Считается, что нейросеть обладает достаточной точностью, если её точность находится в диапазоне от 70 до 90 процентов.

**Постановка задачи.** Имеется множество категорий (классов, меток)  $C = \{c_1, \dots, c_{|C|}\}$  и множество документов  $D = \{d_1, \dots, d_{|D|}\}$ . Неизвестна целевая функция  $\Phi: C \times D \rightarrow \{0, 1\}$ . Необходимо построить классификатор  $\Phi'$ , максимально близкий к  $\Phi$ . Имеется некоторая начальная коллекция размеченных документов  $R \subset C \times D$ , для которых известны значения  $\Phi$ . Обычно её делят на «обучающую» и «проверочную» части. Первая используется для обучения классификатора, вторая — для независимой проверки качества его работы. Классификатор может выдавать точный ответ  $\Phi': C \times D \rightarrow \{0, 1\}$  или степень подобия  $\Phi': C \times D \rightarrow [0, 1]$  [5].

**Методы решения.** Для решения поставленной задачи используется множество различных архитектур и моделей, но в этой работе используется сверточная модель textCNN и рекуррентная модель BiRNN. Хотя CNN изначально были разработаны для компьютерного зрения, они также широко используются для обработки естественного языка. Для обработки текста при помощи CNN можно представить текстовую последовательность как одномерное изображение. Таким образом, одномерные CNN могут обрабатывать локальные функции, такие как n-граммы в тексте [6]. Принцип работы одномерной свёртки представлен на рис. 1.

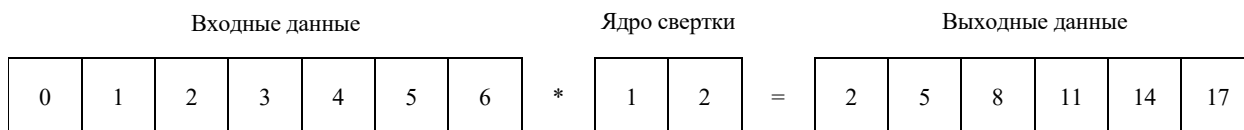


Рис. 1. Одномерная свертка

В одномерном случае окно свертки скользит слева направо по входному тензору [7]. Во время скольжения входной субтензор (0 и 1 на рис. 1), содержащийся в окне свертки в определенной позиции, и тензор ядра (1 и 2 на рис. 1) перемножаются поэлементно. Сумма этих умножений дает единственное скалярное значение ( $0 \times 1 + 1 \times 2 = 2$  на рис. 1) в соответствующей позиции выходного тензора.

Для любого одномерного ввода с несколькими каналами ядро свертки должно иметь одинаковое количество входных каналов. Затем для каждого канала выполняется операция кросс-корреляции над одномерным тензором входа и одномерным тензором ядра свертки, суммируя результаты по всем каналам для получения одномерного выходного тензора. На рис. 2 показана одномерная операция взаимной корреляции с тремя входными каналами.



Рис. 2. Одномерная операция взаимной корреляции с тремя входными каналами

После начального слоя сигнал проходит серию сверточных слоёв, в которых чередуется свёртка и субдискретизация (пулинг). Чередование слоёв позволяет составлять «карты признаков». На каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов. На практике это означает способность распознавания сложных иерархий признаков [7].

Используя одномерную свертку и максимальный пулинг, модель textCNN принимает отдельные предварительно обученные представления токенов в качестве входных данных, затем получает и преобразует представления последовательностей для последующего использования.

Для одной текстовой последовательности с  $n$  токенами, представленными  $d$ -мерными векторами, ширина, высота и количество каналов входного тензора равны  $n$ , 1 и  $d$  соответственно.

Модель textCNN преобразует входные данные в выходные следующим образом. Создаётся несколько одномерных ядер свертки и выполняется операция свертки отдельно для каждого набора входных данных. Ядра свертки разной ширины могут захватывать локальные особенности при разном количестве соседних токенов. Выполняется максимальный пуллинг по времени на всех выходных каналах. Затем все выходные результаты скалярного объединения объединяются в вектор. Объединенный вектор преобразуется в выходные категории, используя полностью связный слой. Dropout [8] можно использовать для уменьшения переобучения.

Стандартная рекуррентная нейронная сеть предполагает, что мы стремимся предсказать следующий токен, учитывая все предыдущие токены в последовательности. В этом сценарии ограничимся только левым контекстом. Поэтому однонаправленное связывание стандартной RNN кажется подходящим. Однако в задаче анализа тональности текста имеет смысл делать прогноз на каждом шаге, учитывая как левый, так и правый контекст. Для создания двунаправленной рекуррентной нейронной сети необходимо реализовать два однонаправленных слоя RNN, соединённых вместе в противоположных направлениях и действующих на один и тот же вход.

Для первого слоя RNN первым входным сигналом является  $x_1$ , а последним —  $x_T$ , но для второго слоя RNN первый входной сигнал это  $x_T$ , а последний —  $x_1$ . Чтобы получить выходные данные этого двунаправленного слоя RNN, просто объединяем соответствующие выходные данные двух базовых однонаправленных слоев RNN. Архитектура двунаправленной RNN представлена на рис. 3.

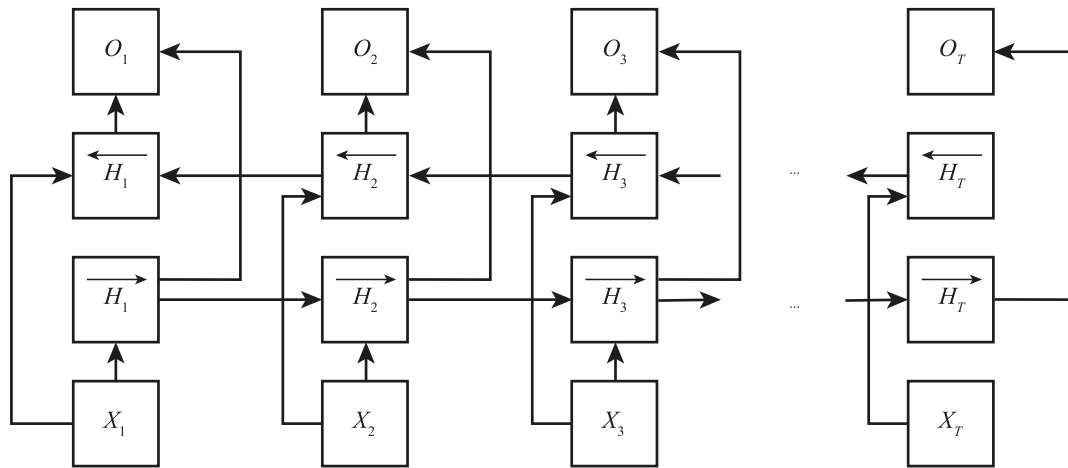


Рис. 3. Архитектура двунаправленной RNN

**Вычислительный эксперимент.** Для определения того, какая архитектура нейронной сети лучше подходит для решения задачи анализа тональности текста, были проведены вычислительные эксперименты, результаты которых позволили определить процент точности реализованных нейронных сетей.

Входные данные:  $b$  — количество элементов, одновременно загружаемых в модель;  $e$  — количество итераций обучения. Для решения задачи анализа тональности текста в виде программы были реализованы сверточная нейросеть textCNN и рекуррентная двунаправленная нейросеть BiRNN. Для получения векторного представления слов использовался 100-мерный GloVe алгоритм. Для обучения нейросетей был использован датасет, содержащий 25 000 отзывов о фильмах. Для тестирования нейросетей был использован датасет, содержащий 25 000 отзывов о фильмах. Для реализации была выбрана среда разработки PyCharm с популярным языком программирования Python. Вычислительные эксперименты проводились на компьютере с использованием операционной системой Microsoft Windows 11, процессором Intel i5-12600k 3,7 Гц и оперативной памятью 32 Гб. Модели имеют следующие входные параметры:  $b$  — 64,  $e$  — 5. В таблице 1 представлены конечные результаты машинных экспериментов.

Таблица 1

Результаты вычислительных экспериментов для сверточной нейронной сети textCNN и рекуррентной нейронной сети BiRNN

Модель textCNN			Модель BiRNN		
Точность при тестировании	Точность при тренировке	Потеря тренировки	Точность при тестировании	Точность при тренировке	Потеря тренировки
0,868	0,979	0,066	0,861	0,884	0,227

**Заключение.** Анализ результатов, полученных в процессе проведенных экспериментов, показал, что сверточная модель textCNN показывает более высокую точность в задаче анализа тональности текста, чем рекуррентная модель BiRNN. Сравнение различных архитектур нейронных сетей в задачах анализа естественного языка позволяет получить данные, необходимые для выбора более оптимальной архитектуры при разработке программного обеспечения, решающего задачу интеллектуального анализа текста.

#### Список литературы

1. *Что такое анализ тональности?* URL: <https://dynamics.microsoft.com/ru-ru/ai/customer-insights/what-is-sentiment-analysis> (дата обращения: 17.12.2023).
2. *Bidirectional recurrent neural network (BiRNN)*. URL: [https://gabormelli.com/RKB/Bidirectional\\_Recurrent\\_Neural\\_Network\\_\(BiRNN\)](https://gabormelli.com/RKB/Bidirectional_Recurrent_Neural_Network_(BiRNN)) (дата обращения: 15.12.2023).
3. *Architecture of text-based convolutional neural network (TextCNN)*. URL: [https://www.researchgate.net/figure/Architecture-of-text-based-convolutional-neural-network-TextCNN\\_fig4\\_364437335](https://www.researchgate.net/figure/Architecture-of-text-based-convolutional-neural-network-TextCNN_fig4_364437335) (дата обращения: 16.12.2023).
4. *Classification: Accuracy*. URL: <https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=> (дата обращения: 18.12.2023).
5. *Классификация текстов и анализ тональности*. URL: [https://neerc.ifmo.ru/wiki/index.php?title=Классификация\\_текстов\\_и\\_анализ\\_тональности](https://neerc.ifmo.ru/wiki/index.php?title=Классификация_текстов_и_анализ_тональности) (дата обращения: 19.12.2023).
6. *Простым языком о языковых моделях и цепи Маркова (Markov Chain)*. URL: <https://habr.com/ru/articles/675218/> (дата обращения: 20.01.2023).
7. Шарипов Р.А. *Быстрое введение в тензорный анализ*. URL: <https://eqworld.ipmnet.ru/ru/library/books/Sharipov2004ru.pdf> (дата обращения: 21.12.2023).
8. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014;15(1):1929–1958. URL: <https://jmlr.org/papers/v15/srivastava14a.html> (дата обращения: 23.12.2023).

*Об авторе:*

**Кирилл Витальевич Трифонов**, магистрант кафедры программного обеспечения вычислительной техники и автоматизированных систем Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [WizarDigeR@gmail.com](mailto:WizarDigeR@gmail.com)

*Конфликт интересов:* автор заявляет об отсутствии конфликта интересов.

*Автор прочитал и одобрил окончательный вариант рукописи.*

*About the Author:*

**Kirill V. Trifonov**, Master's degree student of the Computer Engineering and Automated Systems Software Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), [WizarDigeR@gmail.com](mailto:WizarDigeR@gmail.com)

*Conflict of interest statement:* the author does not have any conflict of interest.

*The author has read and approved the final version of manuscript.*