Исследование нейросетевых методов классификации текстовой информации

В. Л. Литвинов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)
vlad.litvinov61@gmail.com

***Аннотация*. Автоматическая категоризация текстов по их содержимому является актуальной задачей в условиях постоянного увеличения объема текстовой информации. При этом оценка субъективности или объективности текста по отношению к субъекту называется задачей анализа тональности текста и является фундаментальной в анализе мнений. В работе проанализированы как теоретические, так и практические аспекты методов классификации текстов, а также предложены новые подходы классификации текстов с использованием рекуррентных нейронных сетей, позволяющие производить анализ тональности рецензий.**

Ключевые слова: нейронные сети; тональность текста; классификация текста; рекуррентная нейронная сеть

Задача классификации текстов – актуальная задача для многих сфер деятельности, в частности для маркетологов, особенно когда речь идет об анализе пользовательских мнений, отзывов или отношения потребителя к какому-то товару или услуге, поэтому исследования на эту тему, предлагающие новые пути решения таких проблем, появляются ежегодно. При этом анализ мнений является задачей классификации скорее предложений, а не обширных текстов – даже в положительном отзыве пользователь зачастую пишет одно-два отрицательно окрашенных предложения, и их также необходимо выделять, чтобы затем анализировать.

Самая большая трудность в классификации предложений заключается в переменной длине входных данных – поскольку предложения в текстах имеют произвольную длину, не всегда понятно, как подать их на вход нейронной сети. Чтобы решить эту проблему был применен подход менее чувствительный к входным данным, который ранее уже успешно применялся в анализе изображений – метод сверточных нейронных сетей (convolutional neural network, CNN) (рис. 1) [1].

В таких задачах важным пунктом является классификация последовательностей – задачи, в которой каждому слову определяется в соответствие одна метка. Например, морфологический разбор, когда словам присваивается в соответствие часть речи.

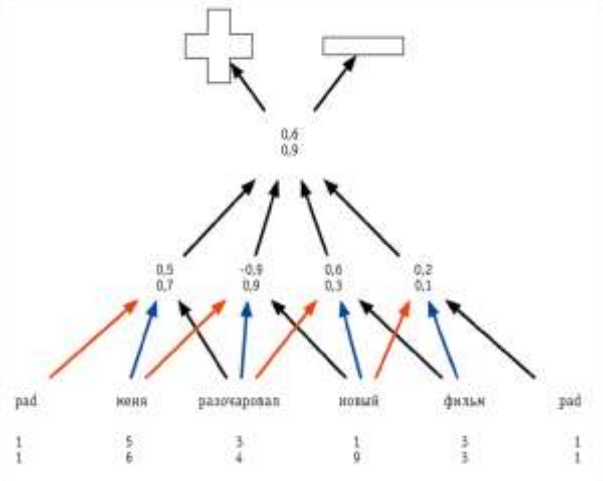

Рис. 1. Пример сверточной нейронной сети для классификации тональности предложения (два класса: положительный и отрицательный)

При классификации последовательностей применяются методы, которые позволяют учитывать контекст слова: если предыдущее слово – часть имени человека, то текущее тоже может быть частью имени, тем самым исключая случаи, когда имя может по ошибке считаться частью названия организации. Решить такую задачу на практике помогают рекуррентные нейронные сети, работающие по другому принципу и расширяющие идею языковых моделей.

Классическая языковая модель в таких системах предсказывает вероятность того, что слово i встретится после слова i–1. Языковые модели можно использовать и для предсказания следующего слова. Для обучения языковых моделей нужны большие текстовые входные данные – корпусы. Чем больше обучающий корпус, тем больше пар слов модель узнает обучаясь, тем самым повышая эффективность своего использования. Рекуррентные нейронные сети позволяют значительно сократить объем хранимых данных.

В зависимости от количества нейронов и числа скрытых слоев, обученная сеть может быть сохранена как определенное количество плотных матриц небольшой размерности. То есть, вместо огромного обучающего

текстового корпуса и всех пар слов в нем можно хранить лишь несколько матриц и небольшой список уникальных слов. Однако такая нейронная языковая модель не позволяет учитывать длинные связи между словами. Эту проблему решают рекуррентные нейронные сети, в которых внутреннее состояние скрытого слоя не только обновляется после того, как на вход приходит новое слово, но и передается на следующий шаг (рис. 2).

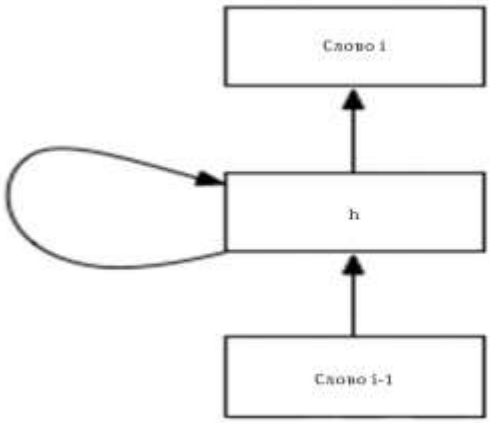

Рис. 2. Модель рекуррентной нейронной сети

Наиболее значимым развитием рекуррентных нейронных сетей для работы с текстом стали архитектуры вида seq2seq [2], применяющиеся в разработках для автоматического перевода и работающие по принципу двух соединенных рекуррентных сетей, где каждая решает свою задачу. Одна из них отвечает за представление и анализ входа (например, вопроса или предложения на одном языке), а вторая – за генерацию выхода (ответа или предложения на другом языке). Эти сети лежат в основе современных систем перевода и чат-ботов.

При этом новые входные данные, которые сеть должна обработать, начинают влиять на память сети: информация в памяти смешивается с новой информацией и через несколько итераций полностью перезаписывается. В задаче анализа тональности текста это особенно важно, поскольку рассматривая большой текстовый корпус, рискуют построить обучение лишь на последнем его отрывке, так как слова, стоящие в начале, не будут вносить изменения в веса нейронов и как-либо учитываться в результатах работы нейронной сети ближе к концу текста. В таком случае ошибка на первых словах текста или предложения будут сильно влиять на общую ошибку нейронной сети. Такое явление называется проблемой исчезающего градиента (vanishing gradient problem), причем универсального решения этой проблемы до сих пор не существует.

Для решения проблемы исчезающего градиента в рекуррентных нейронных сетях была представлена следующая архитектура, которая успела зарекомендовать себя как архитектура «долгая краткосрочная память» (LSTM – long short-term memory) [3]. Эта разновидность рекуррентных нейронных сетей позволяет дольше хранить те веса в нейронах, которые отвечают за первые входные данные, и не позволяет полностью их перезаписать.

Для решения проблемы переобучения в данной работе предлагается воспользоваться техникой Dropout, предложенной в работе [4]. Идея состоит в том, что в процессе обучения выбирается слой, из которого случайным образом выбрасывается определённое количество нейронов, которые выключаются из дальнейших вычислений. Более обученные нейроны получают в сети больший вес.

Таким образом, в работе исследуется модель рекуррентной нейронной сети для анализа тональности текста, состоящая из следующих компонентов:

- входной слой;
- слой LSTM;
- слой Dropout;
- слой усреднения;
- Softmax слой;
- выходной слой.

В работе используется слой Dropout со случайной величиной, распределенной по закону Бернулли с вероятностью наступления события 0.5. Затем данные в нейронной сети попадают в слой усреднения (mean pooling), с архитектурой, которая повторяет слой max pool из сверточных нейронных сетей, но не берет максимальное значение, а усредняет все значения, полученные на выходе предыдущего слоя. После этого данные передаются в Softmax слой, который выполняет функцию классификацию данных.

В Softmax слоях используется обобщение логистической функции для многомерного случая. Функция преобразует вектора z размерности K в вектор σ той же размерности, где каждая координата σ_i полученного вектора представлена вещественным числом в интервале [0,1] а сумма всех координат равна единице.

Координаты σ_i вычисляются следующим образом:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}.$$

На выходе данной модели будет величина $\sigma \in [0,1]$. Поскольку в нашем случае классификация бинарная, значение $\sigma < 0.5$ будет означать, что классифицируемый текст относится к классу «отрицательный отзыв», в противном случае – к классу «положительный отзыв».

Реализация нейронной сети осуществлена с помощью Keras – открытой нейросетевой библиотеки, написанной на языке Python. Она представляет собой надстройку над фреймворками DeepLearning, TensorFlow и Theano Библиотека также содержит готовые слои LSTM и Dropout.

Этап предварительной обработки данных представляет собой следующие операции:

- удаление из текста эмоциональных нетекстовых символов;
- перевод текста в нижний регистр;
- удаление артиклей, предлогов и союзов;
- преобразование отрицательных частиц;
- разбиение текста на слова с учетом знаков препинания.

Этап классификации нового текста включает в себя преобразование текста в представление Мешок слов [5] на основе множества известных слов. Мешок слов (Bag of Words) – это модель текстов на натуральном языке, в которой каждый документ или текст выглядит как неупорядоченный набор слов без сведений о связях между ними. Его можно представить в виде матрицы, каждая строка в которой соответствует отдельному документу или тексту, а каждый столбец – определенному слову. Ячейка на пересечении строки и столбца содержит количество вхождений слова в соответствующий документ.

Для подготовки Мешка слов можно воспользоваться возможностями пакета tm. Этот пакет языка R в качестве объекта работает с так называемым лингвистическим корпусом первого порядка – коллекцией текстов, объединенных общим признаком. Для того чтобы составить корпус сначала нужно преобразовать тексты в вектор, каждый элемент которого представляет отдельный документ, а затем построить на базе корпуса матрицу «документ-термин». Она и станет мешком слов.

В качестве условия останова при обучении сети выбрана модификация метода ранней остановки, который прекращает обучение, если результаты не улучшались на протяжении 5 итераций обучения. Для оптимизации сети используется адаптивный алгоритм градиентного спуска AdaGrad, поскольку стохастический градиентный спуск не всегда дает хорошие результаты в задачах, связанных с анализом последовательностей.

Для обучения сети можно воспользоваться датасетом IMDB, который содержит тексты 50 000 обзоров фильмов из Internet Movie Database. Они разделены на 25 000 обзоров для обучения и 25 000 для проверки модели. Тренировочные и проверочные датасеты сбалансированы, то есть содержат одинаковое количество позитивных и негативных обзоров.

Датасет IMDB доступен в TensorFlow при помощи метода load_data. Он уже подготовлен таким образом, что обзоры (последовательности слов) были конвертированы в последовательность целых чисел, где каждое целое представляет конкретное слово в массиве:

imdb = keras.datasets.imdb

(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=10000).

Величина аргумента num_words=10000 позволяет ограничиться только 10 000 наиболее часто встречающимися словами из тренировочного сета. Все редкие слова исключаются.

Обучение на 7 эпохах позволяет получить приемлемую точность 84 %. Графики точности обучения и проверки представлены на рис. 3.

Увеличение точности за счет увеличения количества нейронов в LSTM слое также возможно, однако это, очевидно, приводит к большим временным затратам на обучение модели. Данный аспект зависит также от используемых аппаратных средств.

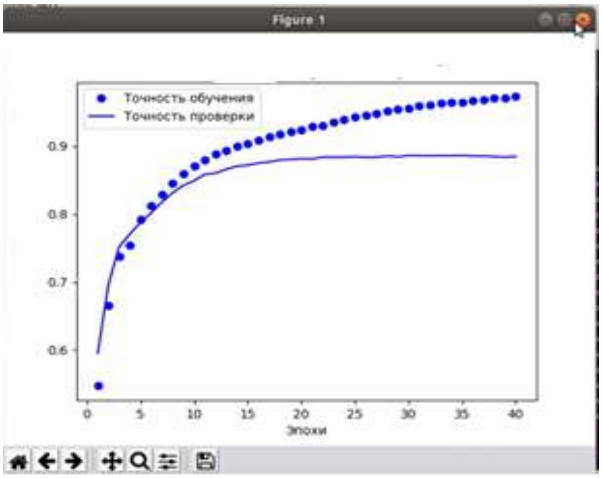

Рис. 3. Графики точности обучения и проверки

Можно обратить внимание, что потери во время обучения уменьшаются, а точность увеличивается с каждой следующей эпохой. Это вполне ожидаемо, поскольку используется градиентный спуск – он минимизирует показатели потерь с каждой итерацией настолько быстро, насколько это возможно.

Предложенная нейросетевая модель также может найти применение при проектировании автоматизированных информационных систем оценки отзывов и рецензий в различных предметных областях.

Список литературы

[1] Ильвовский Д., Черняк Е. Глубинное обучение для автоматической обработки текстов // Открытые системы. СУБД. 2017. № 02. URL: https://www.osp.ru/os/2017/02/13052221/ (дата обращения 05.09.2019).

[2] Google github. Overview seq2seq [Электронный ресурс]. URL: https://google.github.io/seq2seq/ (дата обращения: 05.09.2019).

[3] Sepp Hochreiter, Jürgen Schmidhuber. Long short-term memory // Neural Computation. 1997. Vol. 9. № 8. P. 1735–1780. DOI:10.1162/neco.1997.9.8.1735.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research. 2014. Vol. 15 (Jun). P. 1929-1958. URL: http://jmlr.org/papers/v15/srivastava14a.html (дата обращения: 05.09.2019).

[5] Мешок слов и сентимент-анализ на R. [Электронный ресурс]. URL: https://habr.com/ru/post/255143/ (дата обращения: 05.09.2019)