Text classification using convolutional neural network committee training

N.A. Krivosheev
Department of Information Technology
National Research Tomsk Polytechnic University
Tomsk, Russia
nikola0212@mail.ru

V.G. Spitsyn
Department of Information Technology
National Research Tomsk Polytechnic University
Tomsk, Russia
spvg@tpu.ru

***Abstract* — The method of classification of textual information based on the apparatus of convolutional neural networks is considered. The word-by-word text conversion into dense vectors is considered. Testing was conducted on the text data of the sample "The 20 Newsgroups", this sample contains texts distributed in 20 classes. The accuracy, the best of the convolutional neural network used in this work, on the test sample was ~ 74%. The accuracy of voting of neural networks using the Bagging algorithm was ~ 81.5%. Based on the review of similar solutions, a comparison was made with the following text classification algorithms: using the support vector machine (SVM, 82.84%), naive bayes classifier (81%), k nearest neighbor algorithm (75.93%), a bag of words.**

Keywords — convolutional neural networks, Bagging, text classification, text database "The 20 Newsgroups".

I. Introduction

At the moment, one of the most popular tasks is to understand the text. This task includes: classification, translation, answers to questions, etc. The task of classification is one of the traditional ones in machine learning, and therefore text databases have been created for training neural networks.

There are many solutions to the problem of text classification [1-4], which use various methods for converting text into vectors, such as character-by-character conversion, N-gram characters, word conversion, Word2vec, bag of words, and others. There are many algorithms for text classification: neural networks, support vector machine (SVM), k nearest neighbors, naive Bayes classifier, etc.

Currently, deep neural networks are very popular for solving classification problems, since they allow achieving maximum accuracy among all known machine learning models. In addition, convolutional neural networks have made a breakthrough in image classification. At the moment, they successfully cope with some tasks of word processing. There are many text classification algorithms: neural networks, naive Bayesian classifier, support vector method, and others. Also, as stated in some studies [5-6], convolutional neural networks are better than recurrent neural networks, which are often used for analysis and text processing tasks. But the use of convolutional networks for the classification of texts has been little studied. Therefore, the study of the use of convolutional neural networks for the task of classifying texts, in comparison with other algorithms, is of practical interest.

This paper deals with the problem of text classification using a multi-layer perceptron and a convolutional neural network. The pre-processing of textual data in the form of word-by-word text-to-vector conversion is considered. Considered voting neural networks using Bagging (Bootstrap aggregating) [7]. The results of training and testing of neural networks on a sample of textual data "The 20 Newsgroups" [8] are presented, in this article a comparison is made with analogues. All programs are implemented in Python, using the keras library.

The article discusses the solution to the problem of text classification based on a sample of text data "20 news groups" [8]. A review was carried out of similar solutions that use other text classification algorithms: the support vector method (SVM, 82.84%), the naive Bayes classifier (81%), k nearest neighbors (75.96%), a bag of words.

II. Training Dataset

For training, testing and comparison with analogues, the training dataset "The 20 Newsgroups" was chosen [8]. This sample contains a collection of approximately 18 846 news documents in English, which is divided (approximately) evenly between 20 different categories. These classes include topics such as religion, science, politics, sports, etc. The "20 newsgroups" collection has become a popular data set for experiments with machine learning techniques for text-based applications, such as text classification.

III. Justification Of The Choice Of A Convolutional Neural Network

In recent years, the convolutional neural network (CNN) has become increasingly popular and is used in solving various problems. CNN performed well in solving problems related to the processing of natural language. This algorithm is very flexible and can be used for classification using various text preprocessing methods. CNN classifies textual data much better than a multilayer perceptron [9]. The main feature of the CNN is the use of filters that are sensitive to a specific sequence of words.

The predecessors of convolutional neural networks were models of cognitron and neocognitron. Modern-day convolutional neural networks were presented in the works of Le Kun [10-12] and A. Krizhevsky, I. Sutskever, GE Hinton [13]. The main layers used in convolutional networks are the convolutional layer, the subsampling layer, and the fully connected layer.

The combination of the classification function with the feature of feature extraction using convolution kernels obtained in the learning process allows to select the optimal set of features. Obtaining this feature set by selecting the method of extracting features manually is an almost impossible task.

Based on the above, it was suggested that the convolutional neural network apparatus can show high efficiency in solving the problem of text data classification.

978-1-7281-2896-2/19/$31.00©2019 IEEE

The obtained results should be compared with other approaches that do not use neural networks in terms of the accuracy of solving the problem.

IV. Data Processing

Primary preprocessing consists in translating text into lower case, in removing uninformative and rare characters. Rare symbols include symbols used in the entire sample no more than a few dozen times. The following characters are replaced with spaces: tabulation, line feed, punctuation marks, a long sequence of identical characters (a set of three or more stars, etc.). Removed stop words. Lemmatization [14] of all words in the text is performed. An example of text preprocessing is shown in Fig. 1.

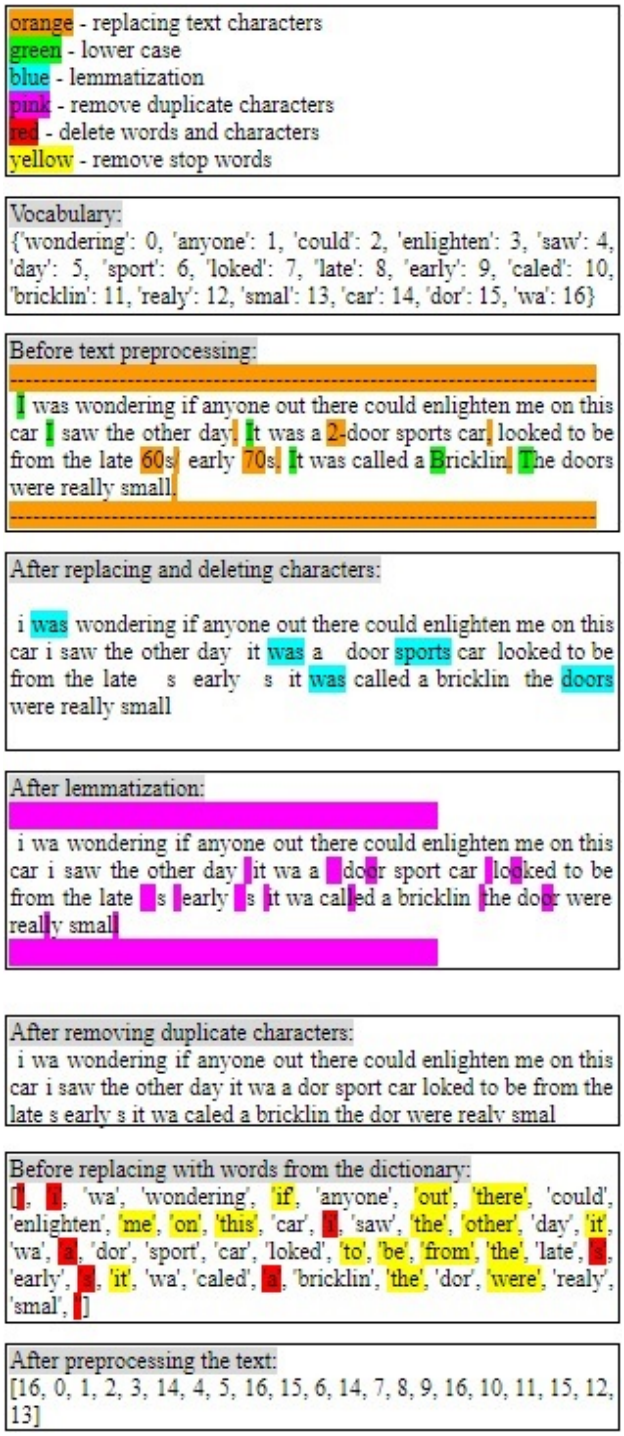

Fig. 1. Text processing example.

In this paper, a word-by-word conversion of text into a vector is performed. A count of the number of references to each word. Very rare and short words are deleted. A dictionary of words is compiled, in which each word is assigned an individual number, for subsequent submission to the neural network.

It should be noted that the accuracy of convolutional neural networks essentially depends on a compiled dictionary, so if you do not delete relatively rare words or select the frequency of mentioning incorrectly, the accuracy of the network decreases significantly and may decrease by almost 2 times. In this work, the threshold for the frequency of mentioning a word in the text is chosen experimentally. If the word is mentioned less than in each 25 * N text (where N is the number of classes), then it is deleted.

The next stage of preprocessing is the conversion of text into a vector, in this paper automatic conversion into dense vectors of fixed size is used. Transformation into dense vectors is performed using the Embedding layer, implemented in the keras library.

After preprocessing, all texts are standardized (cropped or filled) to the specified length. In this paper, all texts are standardized to a text length of 300 words. If the length of the text is less than 300 words, then the missing part of the vector is filled with zeros.

V. The Results Of Learning And Testing Neural Networks

In this paper we tested convolutional neural networks of various topologies. The Bagging algorithm was applied.

All neural network topologies were trained using the NADAM method [15], using the categorical loss function (categorical_crossentropy). In all hidden layers of the neural network, the activation function RELU is used [16]. The output layer uses the softmax activation function [17].

The Bagging algorithm uses the voting of 7 convolutional neural networks. Voting takes place by elementwise multiplication of the output vectors of neural networks. The topology of the best convolutional neural network used in this work is shown in Fig. 2.

The topologies of convolutional neural networks are different (the number of layers, the number of neurons in the layer and the size of the convolution window, etc.).

In this work, we tested neural networks on the test sample “The 20 Newsgroups” [8], using word-by-word text conversion into dense vectors.

The recognition accuracy of the most accurate convolutional neural network on the test sample is ~ 74%. The average accuracy of the used neural networks is ~ 71.9%. When using convolutional neural networks and the Bagging method, the classification accuracy increases to ~ 81.5%.

According to the data obtained, it can be concluded that the Bagging algorithm significantly increased the classification accuracy.

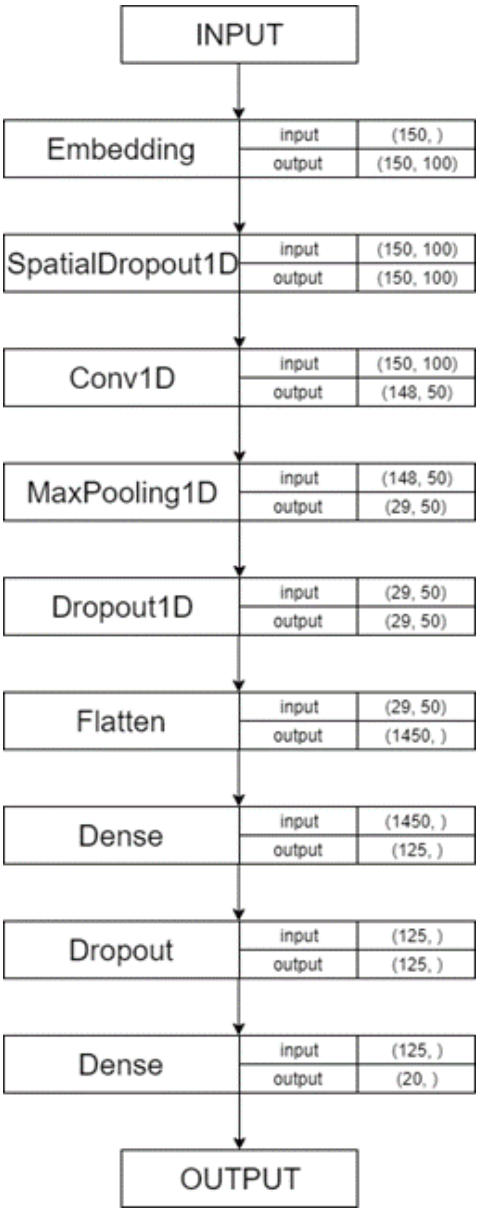

Fig. 2. The topology of the convolutional neural network, the best of those used in this work.

VI. Comparison With Analogues

There are many analogs for solving the problem of text classification, which were tested on the basis of text data "The 20 Newsgroups" [1-3]. The presented analogues use such methods as: multilayer perceptron, convolutional neural networks, naive Bayes classifier, and support vector machine (SVM).

In article [1], the author uses a bag of words and a multilayer perceptron as a classifier. In the article, the author uses three classes from the training set: comp.graphics, sci.space, rec.sport.baseball. The accuracy of the neural network used in the article [1] was 75%.

In this paper, we tested the voting of neural networks (the algorithm described above) according to the data indicated in the article [1]. The accuracy of the algorithm was ~ 95.5%. The voting accuracy of convolutional neural networks far exceeds the accuracy of the algorithm specified in the article [1]. A comparison of the accuracy of the algorithms is shown in Fig. 3:

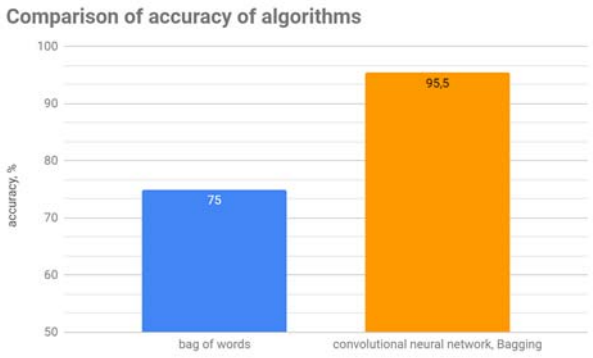

Fig. 3. A comparison of the accuracy of the algorithms [1].

In the article [2], the author tested the following algorithms: support vector machine (SVM), k nearest neighbors, naive Bayes classifier, etc. In the article, the author uses all 20 classes from the training set. The accuracy of the support vector method (SVM) is 82.84%. This algorithm is the best of those presented in article [2] for training and testing on the sample "The 20 Newsgroups" [8]. The accuracy of the support vector method exceeds the accuracy of the classifier of voting-based convolutional neural networks. The accuracy of the naive Bayes classifier presented in article [2] is 81%, which greatly exceeds the algorithm of k nearest neighbors, which was tested in article [2]. The accuracy of the k nearest neighbor algorithm is 75.93%.

In this paper, we tested the voting of neural networks using the Bagging method on the data indicated in the article [2]; the accuracy on the test sample was ~ 81.5%. Voting of neural networks is inferior to the support vector method (SVM). Also, this method proved to be more accurate than other algorithms specified in the article [2]. A comparison of the accuracy of the algorithms is presented in Fig. 4:

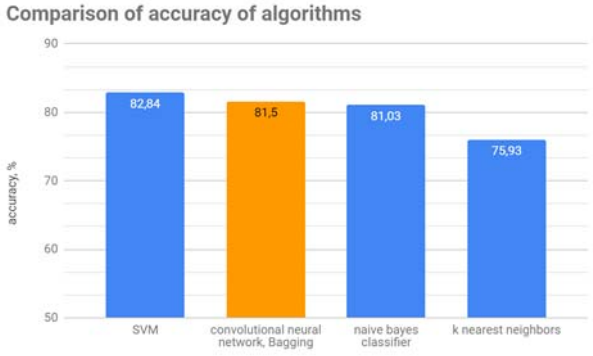

Fig. 4. A comparison of the accuracy of the algorithms [2].

In the article [3], the author tested the following algorithms: a naive Bayes classifier and the support vector machine (SVM) method. In the article, the author uses four classes from the training set: alt.atheism, comp.graphics, sci.med, soc.religion.christian. The accuracy of the naive Bayes classifier used in article [3] was 83.4%. The accuracy of the support vector (SVM) method used in [3] was 91.2%.

In this work, we tested the voting of neural networks using the Bagging method on the data indicated in the article [3], the accuracy on the test sample was ~ 92%,

which slightly exceeds the support vector method (SVM, 91.2%). In addition, the Bagging algorithm far surpasses the naive Bayes classifier (83.4%). A comparison of the accuracy of the algorithms is presented in Fig. 5:

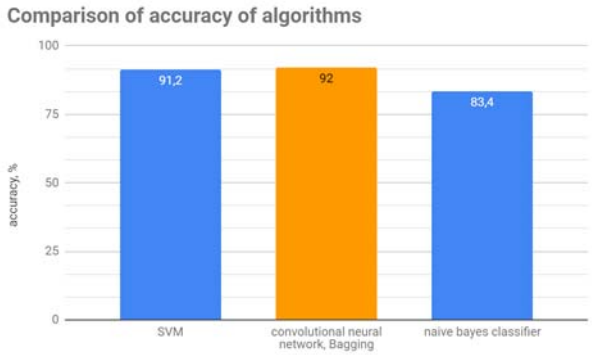

Fig. 5. A comparison of the accuracy of the algorithms [3].

Based on the comparison with analogues, we can conclude that the voting of neural networks using the Bagging method can compete with such methods as the naive Bayesian classifier and the support vector method (SVM).

VII. Conclusion

In this paper, the voting of convolutional neural networks by the Beggig algorithm is implemented and tested. On the basis of the obtained results, it can be concluded that voting of convolutional neural networks using Bagging showed a significant increase in classification accuracy compared to previously obtained results using a multi-layer perceptron [9] and can compete with other presented algorithms intended for text classification. In the future we plan to search for new and improve the methods used to solve this problem.

Acknowledgment

The reported study was funded by RFBR according to the research project № 18-08-00977 A.

Reference

[1] General view on machine learning: text classification using neural networks and TensorFlow [Electronic resource]. - Access mode: URL: https://tproger.ru/translations/text-classification-tensorflow-neural-networks/ (11.21.2018).

[2] Datasets for single-label text categorization [Electronic resource]. - Access mode: URL: http://ana.cachopo.org/datasets-for-single-label-text-categorization (03.06.2019).

[3] Working with text data in scikit-learn [Electronic resource]. - Access mode: URL: https://habr.com/ru/post/264339/ (05.20.2019).

[4] Text classification using Java neural network [Electronic resource]. - Access mode: URL: https://habr.com/post/332078/ (11.21.2018).

[5] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. 2018. arXiv preprint arXiv: 1803.01271

[6] A. Conneau, H. Schwenk, L. Barrault, Y. LeCun, Very deep convolutional networks for text classification. 2017. arXiv preprint arXiv: 1606.01781

[7] Bagging [Electronic resource]. - Access Mode: URL: https://ru.wikipedia.org/wiki/%D0%91%D1%8D%D0%B3%D0%B3%D0%B8%D0%BD%D0%B3 (06.15.2019)

[8] Sklearn.datasets.fetch_20newsgroups [Electronic resource]. - Access mode: URL: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html (11.22.2018).

[9] N.A. Krivosheev, V.G. Spitsyn Algorithms for understanding the text by the methods of deep learning of neural networks. Proceedings of the XVI International Scientific and Practical Conference of Students, Postgraduates and Young Scientists "Youth and Modern Information Technologies" - Tomsk, 2018, p. 82-83.

[10] Y. LeCun Backpropagation applied to handwritten zip code recognition. Neural computation. 1989 Vol. 1 (4), pp. 541–551.

[11] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner Gradientbased learning and document recognition. Proceedings of the IEEE. 1998 Vol. 86 (11), pp. 2278-2324.

[12] Y. LeCun Efficient backprop. Neural Networks: T.C. Montavon, GB Orr, K.-R. Muller (Eds.) - Springer, 2012, pp. 9-48.

[13] A. Krizhevsky, I. Sutskever, Hinton GE Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012, pp. 1097–1105.

[14] Lemmatization [Electronic resource]. - Access Mode: URL: https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F (06.15.2019).

[15] An overview of the gradient descent optimization algorithms [Electronic resource]. - Access mode: URL: http://ruder.io/optimizing-gradient-descent/index.html#nadam (11.22.2018).

[16] You need to know: Key recommendations for deep learning (Part 2) [Electronic resource]. - Access mode: URL: http://datareview.info/article/eto-nuzhno-znat-klyuchevyie-rekomendatsii-po-glubokomu-obucheniyu-chast-2/ (05.20.2019).

[17] Softmax [Electronic resource]. - Access mode: URL: https://ru.wikipedia.org/wiki/Softmax (05.20.2019).