Text classification using convolutional neural network

L E Sapozhnikova[1], O A Gordeeva[1]

[1]Samara National Research University, Moskovskoye shosse, 34, Samara, Russia, 443086

e-mail: sapozhnikova111@gmail.com

Abstract. In this article, the method of text classification using a convolutional neural network is presented. The problem of text classification is formulated, the architecture and the parameters of a convolutional neural network for solving the problem are described, the steps of the solution and the results of classification are given. The convolutional network which was used was trained to classify the texts of the news messages of Internet information portals. The semantic preprocessing of the text and the translation of words into attribute vectors are generated using the open word2vec model. The analysis of the dependence of the classification quality on the parameters of the neural network is presented. The using of the network allowed obtaining a classification accuracy of about 84%. In the estimation of the accuracy of the classification, the texts were checked to belong to the group of semantically similar classes. This approach allowed analyzing news messages in cases where the text themes and the number of classification classes in the training and control samples do not equal.

1. Introduction

Today, volume of stored and used information continuously increases, and one of automatic text processing tasks is the problem of classifying text data, which allows separating texts into various thematic catalogs (categories, classes). Sites, documents, letters, appeals, news are classified for optimal storage and usage.

Text classifiers are used to recognize the emotional coloring of the text during reviews and comments processing. Text classification is used in antispam systems and contextual advertising via the analysis of user activity in the network and the classification of sites, which the user viewed.

Various methods and technologies can be used to classify textual information. Classification of textual information can be based on the assessment of the meaning of the text [1], on frequency analysis [2]. BigData technology is often used for solving of classification problem for text and media [3]. This article discusses the method based on the convolutional neural network. Some aspects and features of its application to solve the problem of text classification, as well as the results of applying this classification method are presented.

2. The formulation of the text classification problem

In general, the text classification problem is formulated as follows:

There is a set of objects (texts) and a beforehand defined set of classes with which objects can be compared. For some of the objects it is known which class they belong to. This subset is a training

sample. For the rest of the objects the classes are not defined. It is necessary to determine which class each object (text) from a set of objects belongs to.

The problem of text classification can be formalized as follows [4]:

There is a set of texts $D = \{d_1, ...d_n\}$. Each text $d_i \in D$ is a sequence of words $Wd = \{w_1, ..., ...\}$. A finite set of classes $C = \{c_1, ..., c_m\}$ are given. Ideal classifier which translates an object d to its class c_j can be termed as $\Phi(d)$. The task is to build another classifier $\tilde{\Phi}(d)$ which able to classify an arbitrary object d and is closest to the ideal classifier $\Phi(d)$.

The solving process includes the following main steps:

- Pre-processing of text including tokenization and vector representation of words.
- Building a classifier.
- Estimation of misclassification probability.

Text classification methods [5]:

- probabilistic (naive Bayes classifier);
- metric (k-nearest neighbours method);
- logical (decision tree classifier);
- linear (logical regression);
- methods based on neural networks

3. Text classification using convolutional neural network

3.1. General architecture of a convolutional neural network

Convolutional neural networks are very effectively used to solve the problem of text classification [6]. The result of the classification is the distribution of the probabilities that the text belongs to beforehand defined classes.

The basic architecture of the convolutional neural network consists of the following layers [7].

1. A convolutional layer, which is a set of attribute maps (matrices), each map has a convolution kernel, which is a filter (or window) that slides over the entire area of the attribute map. The set of filters determines the dimension of the new matrix. The error backpropagation algorithm for convolutional networks is also a convolution, but with spatially inverted filters.

2. Sub-sampling layer, which reduces the size of the matrix. On this layer the most frequently used method is the maximum element method (max-pooling).

3. A fully connected layer in which each neuron is connected to all neurons at the previous layer, and each connection has its own weight.

4. The output layer, which is connected with all neurons of the previous layer. The number of neurons corresponds to the number of classification classes.

3.2. The architecture of the used convolutional neural network

In this research, the specific model of the convolutional neural network was determined. The input data is words that are represented by vectors of semantic attributes. In the representation the words close in meaning are located at close distance in the vector space.

The vector $x_i \in R^k$ is k-dimensional vector corresponding to the i-word in the sentence. Then the sentence with length = n can be defined as [6]:

$$x_{1..n} = x_1 \oplus x_2 \oplus ... \oplus x_n, \quad (1)$$

where $\oplus$ is concatenation operation.

The term $x_{i..i+j}$ means the concatenation of words $x_i, x_{i+1}, ..., x_{i+j}$. Convolution uses the filter $w \in R^{hk}$ which is applied to the window containing h words to create a new attribute. For example, the attribute c_i will be generated from the window of words $x_{i..i+h-1}$ as

$$c_i = f\left(w * x_{i..i+h-1} + b\right), \quad (2)$$

where $b \in R$ is the offset step, f is nonlinear activation function.

This filter applies to every possible window of words in a sentence $\{x_{1..h}, x_{2..h+1}, ..., x_{n-h+1..n}\}$ to produce a new map of attributes

$$c = \{c_1, c_2, ...c_{n-h+1}\}, \quad (3)$$

where $c \in R^{n-h+1}$.

Then the operation of combining the set of values is applied, the maximum value of $\hat{c} = \max\{c\}$ (the most important attribute for each convolution) is selected.

In the process, the network uses several filters with different window sizes to obtain a set of attributes. These attributes from the penultimate layer are transferred to the last layer, and the output data is the probability of the distribution of attributes into classes.

As an activation function Leaky ReLU function is used [7]. The formula is:

$$f(x) = 1(x < 0) * (\alpha x) + 1(x \geq 0)(x), \quad (4)$$

where α is a constant with a small-scale value.

This function has a higher convergence rate compared to other activation functions and is also quite simple to calculate.

To regularize the neural network (to prevent overfitting), L2 regularization [7] is used in combination with the dropout [8].

L2-regularization is implemented by the penalization of the neural network by increasing the loss function. For each weight w the loss function λw^2 is added, where λ is the strength of regularization.

L2-regularization prevents a strong increase in weights of neurons and leads to a redistribution of weight values. This causes the neural network to use all neurons at least to a small extent.

Dropout is the random disconnection of neurons. At each level of training, some neurons are excluded from the network. It helps to avoid the dependence between neutrons during training.

The combination of L2-regularization and dropout allows avoiding a situation where the network shows excellent results on a training sample but is ineffective when tested on a control sample.

The architecture of the network which was used is shown in figure 1. The first layer solves the problem of the vector representation of words. Next, three convolutional layers are created (conv2d_1, conv2d_2, conv2d_3). The figure also shows the Leaky ReLU activation function for each layer, the dropout and the sub-sampling layer (max-pooling). Then the layers merge, and at the end, a fully connected layer is obtained (dense_1). The architecture is designed using the TensorBoard visualization system [9].

3.3. The input data formatting

The testing of the described classification method was carried out on the materials of the RIA Novosti news portal on the Internet. This portal contains a huge number of publications with well-defined themes. It allows to define categories (classes) for classification, as well as to generate a sufficient number of text fragments for neural network training.

Eight classes were determined - *culture*, *events*, *religion*, *society*, *economy*, *politics*, *science*, *world*. It was received 8,000 articles from the materials of the portal for each class. In total, 64,000 news articles related to one of the 8 classes were received. For data acquisition the software was developed with Node.js platform. These data were preliminarly processed before starting the training of the neural network.

3.4. Text preprocessing

The neural network input data formatting was performed in Python 3.6 using Keras and Jupyter Notebook libraries.

The maximum text size was limited to 1000 characters. The texts of a greater number of characters were divided into parts and assigned to the same class. For the training of a neural network, it is important to have the same number of examples of each class, otherwise, the neural network will ignore the semantic meaning of the text and will take into account the a priori probability of the appearance of each class articles.

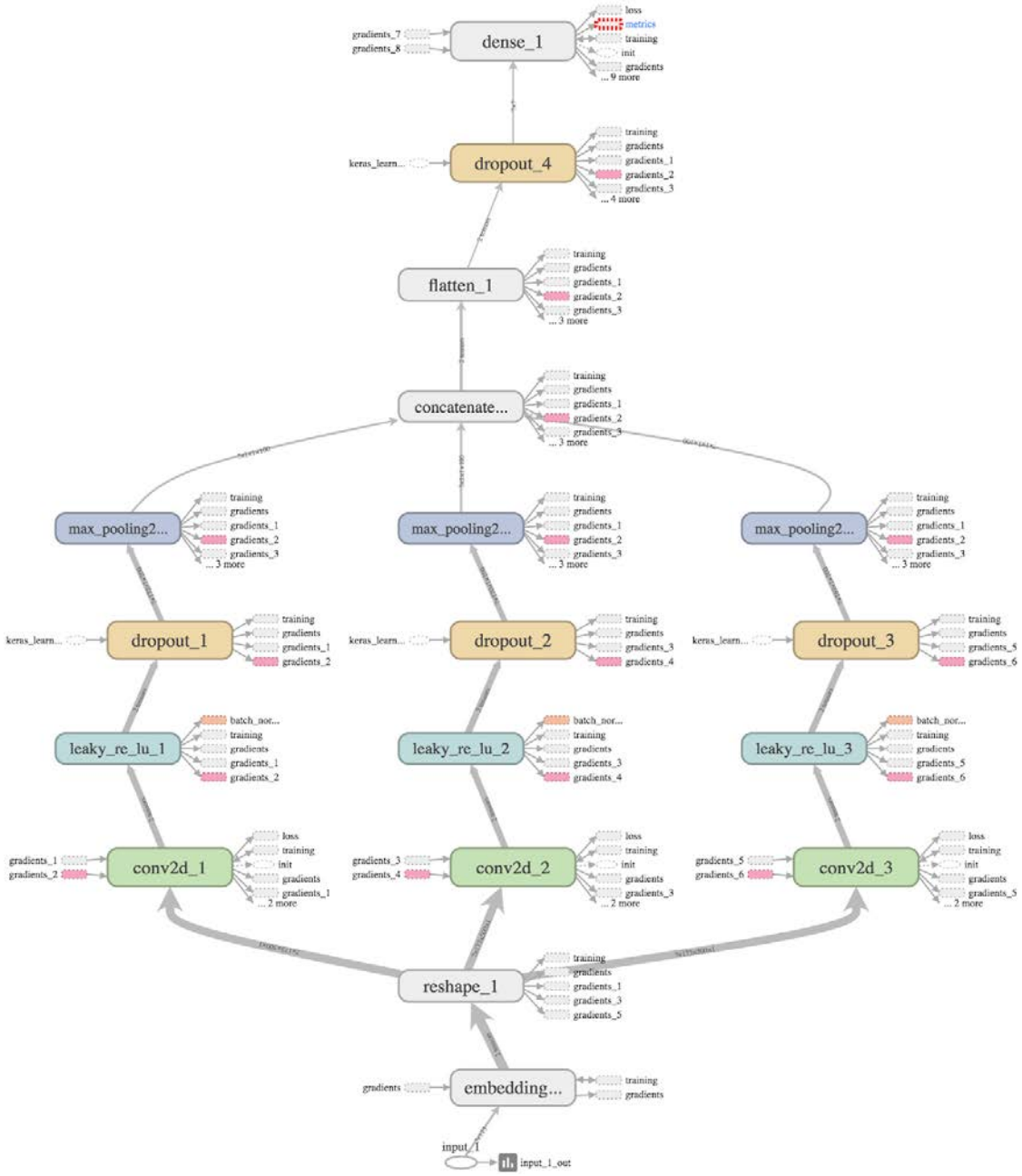

Figure 1. The architecture of the convolutional neural network.

To avoid this situation, the volumes (number of texts) of the classes were equalized with the class of the smallest volume - 11,733 texts. A total of 93,864 texts were obtained for 8 classes.

Then all texts were divided into three samples - training (60% of texts), validation (20% of texts) and control (20% of texts).

After that, the tokenization of texts in order to find unique words was carried out. As a result, 282,972 unique tokens (words) were found. The input data was formatted for the neural network with Python 3.6 programming language using the Keras and Jupyter Notebook libraries. The tokens have to be translated into vector representations of attributes for network training.

3.5. Vectors of attributes formation

Vector representation describes the dependencies between words. In vector space, similar words will have similar vectors. Research in this field was carried by Thomas Mikolov [10]. For vector

representations, he obtained important results in syntactic and semantic meaning. As a result of his research, it was determined that the resulting vector representations of words have significant syntactic and semantic patterns that are implemented in existing models and network libraries. In particular, there is a constant vector displacement between pairs of words indicating a specific relationship. Moreover, even more complex relationships were discovered. For example, you can define the relationship between words as the difference of their vectors. Also, the feminine and masculine words will have a constant vector difference.

The most popular practical way to obtain a vector of attributes is to use word2vec language models created using neural networks. These models are trained in very large volumes of natural language words in various grammatical forms, including words in different kinds, cases, inclinations, and so on. In addition, they also include stop-words. Therefore, such models do not require specific preliminary actions as removing stop-words, stemming or lemmatization, which are necessary for classification by other methods. Keeping different word forms may even increase the accuracy of the classification. As a result, semantically similar words and synonyms will have similar vectors of semantic attribute values.

In this research, the open model word2vec [11] is used. This model represents a word as a vector of 500 values of attributes. The model is pre-trained and it was designed for the Russian language. To begin with, a series of experiments of vector construction was conducted for similar words. It confirmed the ability to use the model. As a result, it was found that this vectorization model constructs similar attribute vectors for words that are close in meaning. Consequently, this trained model was found to be suitable to translate the words of the experimental sample into the vector of attributes without applying stemming or lemmatization and without removing the stop-words.

3.6. Training and use of the network

The convolutional neural network which was shown in figure 1 and described in paragraph 3.2 was trained with parameters presented in table 1. These parameters were selected based on the results of a series of experiments to investigate the dependence of classification accuracy on the parameters of the neural network.

Table 1. Neural network parameter values.

Parameter	Value
Number of learning epochs	5
Activation function	Leaky ReLU with α=0,1
Convolutional layers	3 layers with filter size = 2, 4 and 5
The number of filters	100
Regularization L2	0,1
Dropout for convolutional layers	0,5
Dropout for the fully connected layer	0,6
The speed of learning	0,001

It should be noted that with each epoch the accuracy of the classification of the training sample increases and the accuracy of the classification of the validation sample ceases to increase after the 5^{th} epoch. It indicates that the network is overfitted. Thus, after the 5^{th} epoch, the accuracy of the classification of the validation sample is 84%. All neuron weights were saved for further use in text classification tasks. The classification accuracy for the control sample, which was separated from the total set of texts at the step of text preprocessing, shows that the trained model of the neural network with the parameters specified in table 1 solves the problem of text classification with an accuracy of 84%.

One of the difficulties in text classification is that it is not always possible to determine the probability that a text belongs to a particular class if the classes are semantically related. For example, the text of the news “In country X, the president issued a decree on raising taxes” has elements of politics, economics, and, possibly, international news when it comes to a foreign country. Therefore, there is a certain subjective character of classification by both a human and automated systems of

different Internet-portals, which have different algorithms for deciding whether a text belongs to a particular class.

For the more detailed research of the classification accuracy, three control samples were compiled from various news sites to test the neural network processing on other control samples. Texts of news from portal RIA Novosti (18773 texts), Mail.ru news portal (7740 texts) and TASS news agency (16835 texts) were acquired. While the texts of the control sample from the RIA Novosti website were not used in network training.

It should be noted that the network was trained for 8 thematic classes used by RIA-Novosti: *culture*, *events*, *religion*, *society*, *economics*, *politics*, *science*, and the *world*. However, for Mail.ru news, the following classes are used: *events*, *society*, *economy*, *politics* (4 classes), and for TASS - *culture*, *events*, *society*, *economy*, *politics*, *science*, *the world* (7 classes).

The number of classes does not match, so for the sample from Mail.ru the estimated classification accuracy is slightly more than 40%, for TASS news texts the classification accuracy was 67%. As mentioned earlier, the classification accuracy for RIA Novosti data was 84% - the greatest accuracy, since the classes of the training and control samples are the same.

Next experiment the same classes in the training and classification for all three data sets was used. Four classes were chosen - *accidents*, *society*, *economics*, and *politics*. The neural network was retrained. The results of the experiments are presented in table 2.

Table2. Accuracy of control sample classification.

Sample source	RIA Novosti	Mail.ru	TASS
Training sample – 8 classes			
Number of classes for the control sample	8	4	7
Classification accuracy	84 %	46 %	65,5 %
Training sample – 4 classes			
Number of classes for the control sample	4	4	4
Classification accuracy	84 %	66 %	73 %

As can be seen from table 2, for the texts of RIA Novosti, the classification accuracy remained unchanged (84%), for Mail.ru and TASS news it increased significantly and amounted to 68% and 75%, respectively. However, these values are still lower than for the texts of RIA Novosti.

Such low values of classification accuracy are related to the fact that some of the texts of other sources can be initially attributed to semantically similar classes, and, although the classification accuracy is low, it does not indicate the poor quality of the neural network processing.

The results of the experiment show that in the presence of semantically similar classes and the classification of data from sources other than the sources of the training sample, the accuracy of the classification may be incorrectly underestimated.

To eliminate this effect, groups of semantically similar classes were identified, and the calculation of the classification accuracy was adjusted so that the classified text was checked for belonging to a group of semantically similar classes, not to a particular class.

Groups of semantically similar classes are presented in table 3.

Table 3. Groups of semantically similar classes.

Group	Classes in the group
1	the world politics
2	society events
3	culture religion
4	economy
5	science

The results of the estimation of the classification accuracy by belonging to a group of semantically similar classes are presented in Table 4.

Table 4. Comparison of the accuracy of the control sample classification without using and using groups of semantically similar classes.

Sample source	Mail.ru	TASS
Training sample – 8 classes		
Number of classes for the control sample	4	7
Classification accuracy without groups of semantically similar classes	46 %	65,5 %
Classification accuracy with groups of semantically similar classes	57,5 %	73,5 %
Training sample – 4 classes		
Number of classes for the control sample	4	4
Classification accuracy without groups of semantically similar classes	66 %	73 %
Classification accuracy with groups of semantically similar classes	73,5 %	78,5 %

As can be seen from table 4, the classification accuracy has increased significantly in all cases, both when teaching in eight classes, and when teaching in four classes (up to 12%). As a general result, an increase in the classification accuracy for control samples from other sources averaged 8–10%.

Thus, the most objective value of classification accuracy is obtained for the classification of a control sample from a source that is also a source of a training sample. However, when estimating the accuracy of the classification for control samples from other sources, the most optimal results are achievable with an equal number of classes for training and control samples and when checking to belong the control sample texts to a group of semantically similar classes.

4. Conclusion

The problem of classification is the current direction in the processing of text data. Text classification processes are implemented in various areas: classifiers for various characteristics (themes, style, the emotional coloring of the text), spam filtering, contextual advertising, and so on.

In this paper, the applicability of a convolutional neural network for solving the problem of text classification was researched. The convolutional neural network was trained using the already trained neural network for translation of words in vectors of selected attributes which represent universal semantic meanings that can be used to classify the texts of the natural Russian language.

The constructed neural network is able to classify news and other texts by themes and to provide a distribution of the probability of belonging of the text to 8 predetermined classes. The accuracy of the classification is estimated by the control sample and is 84%. For the selected number of classes, this classification result indicates the effectiveness of using a convolutional neural network for text classification.

Processing of the constructed and trained neural network is researched by control samples from various sources - RIA Novosti, TASS, Mail.ru. The highest classification accuracy was obtained on a control sample of RIA Novosti articles and amounted to 84%. To classify the data of the two other samples, using an equal number of classes of training and control samples, as well as using groups of semantically similar classes for estimation of classification accuracy, allowed to obtain a more objective and higher result. The neural network has shown its effectiveness in solving the problem of text classification. The downside of its use is the need for large amounts of data for training and validation.

5. References

[1] Mikhaylov D V, Kozlov A P and Emelyanov G M 2015 An approach based on tf-idf metrics to extract the knowledge and relevant linguistic means on subject-oriented text sets *Computer*

Optics **39(3)** 429-438 DOI: 10.18287/0134-2452-2015-39-3-429-438
[2] Mikhaylov D V, Kozlov A P and Emelyanov G M 2016 An approach based on analysis of n-grams on links of words to extract the knowledge and relevant linguistic means on subject-oriented text sets *Computer Optics* **40(4)** 572-582 DOI: 10.18287/2412-6179-2016-40-4-572-582
[3] Rytsarev I A, Kirsh D V and Kupriyanov A V 2018 Clustering of media content from social networks using bigdata technology *Computer Optics* **42(5)** 921-927 DOI: 10.18287/2412-6179-2018-42-5-921-927
[4] Eprev A S 2010 Automatic classification of text documents *Math. Struct. and modeling* **21** 65
[5] Batura T V 2017 Methods of automatic text classification *Programming products and systems* **1** 85
[6] Kim Y 2014 Convolutional Neural Networks for Sentence Classification *Proc. of EMNLP* (Doha Qatar) 1746-1751
[7] Karpathy A 2019 CS231n: Convolutional Neural Networks for Visual Recognition *Stanford CS class* URL: http://cs231n.github.io
[8] Budhiraja A 2018 *Dropout in (Deep) Machine learning* URL: https://medium.com/amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5
[9] Tensorboard URL: https://www.tensorflow.org/programmers_guide/summaries_and_tensorboard
[10] Mikolov T, Yih W and Zweig G 2013 Linguistic Regularities in Continuous Space Word Representations *Proc. of NAACL-HLT* 746-751
[11] Model word2vec URL: http://panchenko.me/data/dsl-backup/w2v-ru