Building a Novel News Text Classifier with Supervised Deep Learning Algorithm

Tianchuzi Qin[1, *], Zihan Dai[2, a], Peiyuan Gao[3, b], Yujian Huang[4, c], Ziqi Meng[5, d], Congyao Wang[6, e]

[1]The university of Vermont, VT, USA

[2]New York University Shanghai, Shanghai, China

[3]University of Alberta, AB, CAN

[4]The Hong Kong Polytechnic University, Hong Kong, China

[5]Skidmore College, NY, USA

[6]The Ohio State University, OH, USA

[a]bzd656@nyu.edu, [b]gaopeiyuan347@163.com, [c]eugene.huang0120@gmail.com, [d]mengziqi020814@163.com, [e]1hkhlzys22@gmail.com

[*]Corresponding author: 1727837663@qq.com

These authors contributed equally to this work

Abstract

Nowadays, the rise of big data has made the technology of automatic recommendation to users through data analysis widely used on Internet platforms, which has greatly reduced the work of manual push in the past. Many popular Internet applications such as Toutiao, Douyin and Watermelon Video have used this technology. In this way, their users can always get some content and consultation of the type they like. Therefore, it is very meaningful to use CNN convolutional neural network to analyze and classify a large number of text data types. In this paper, deep learning algorithm is used to build a text classification model, and CNN convolutional neural network algorithm is used to help us build such a model, which not only guarantees the accuracy but also presents the text classification process in a simpler, faster and more intuitive way. This model applies the same structure of many convolutional neural networks in the analysis and classification of images, establishes the multi-layer neural network algorithm for text calculation, and performs word segmentation of text data to make it easier to recognize. In addition, the loading time is greatly shortened. At the same time, Vector, quantization, convolution, Max pooling and full link network are also included in the algorithm system. Eventually, after complete the construction of the whole algorithm system, we also use a lot of real data sets the text of the complete experiment, from ten types of database, for example, we find a large number of text data, and then categorize them by our algorithm, finally the experimental results also show that the success of the laboratory, The efficiency of the text algorithm is more than 90 percent, and the accuracy of the text is more than 95 percent, which can undoubtedly prove that our entire algorithm is complete and successful.

Keywords

Classifier, CNN Convolutional Neural Network Algorithm, Max Pooling, Full Link Network.

1. Introduction

1.1 Significance

The reason we do this selected topic is that,in our normal life will leave a lot of our online search or browse the traces, so in our search terms will appear all sorts of different types of words, these terms are collected and classified by the background will push a lot to us and we search the content related to information and products, While the existence of this algorithm can save a lot of artificial classification and push the time, but also greatly improve the efficiency of the app in background, the intangible between increases the user for application and also improve the user experience, so, the study will be in the future Internet applications of text reading and classification play an important role that cannot be ignored.

1.2 Model

Our text classification is based on the deep learning algorithm. Since CNN (Convolutional Neural Networks) model is proved that it is efficient in dealing with the process of natural language, we use TensorFlow as our foundational frame and CNN as our basic algorithm. The structure of our CNN model consists of input layer, embedding layer, convolutional layer, max pooling layer, full connected layer and output layer. After training, our model is supposed to predict the class of text precisely in some degree.

Specifically, the embedding layer is designed to get a low-dimensional space compared to the original text which is a really high-dimensional space. This can improve the efficiency of our neural network to a great extent. The convolutional layer consists of lots of feature maps. Each of the feature map is a matrix contains several neurons that share the same set of weights. And the shared weights are called the filter, which are parameters to be learned. Our mission is to find out the values for the filters. The max pooling layer is designed to further decrease the dimension of the data so it can be more computational. And the fully connected layer is used to calculate the final results.

1.3 Procedure

Our first step is collecting and preparing the dataset of documents. We need to process the text and its corresponding class. After this, we will get a large size of samples which include a quantity of documents with their relating categories.

Secondly, we need to translate our text data into vectors. We will assign each word with a vector in a predefined vector space. Once we have finished this step, the data will be translated to a matrix with real values so that our algorithm can compute it much more easily and it can be used as our input layer.

Then, our third step is to train the CNN model. By applying the convolutional layer, max pooling layer and fully connected layer, we will get the output layer. Simultaneously, we need to evaluate the performance of our model according to the cross entropy loss. To improve the accuracy of our model, we need to change the parameters used for out model or conduct more times of iterations. We will stop repeating the training once the loss is convergent. And in our design, we stop training before the loss starts to increase in the next iteration.

After several times of iteration, we will get our final model with a loss as small as possible.

1.4 Result

So through the experiment we introduced above structure, the algorithm of our whole system is very complete, in order to verify whether our whole structure existence question and the accuracy of the algorithm, we also chose the ten aspects, such as sports, science and technology, entertainment, has chosen a large number of text content related to experiment, ultimately through test and final test results, After the three iterations were stopped, the accuracy of our final experimental data reached more than 95 percent.

2. Design

2.1 TextCNN (Text Convolutional Neural Network)

We use TextCNN (Text Convolutional Neural Network) as our deep learning algorithm to do text classification.

2.1.1 Embedding

We need to use word embeddings to translate documents to a dense and low-dimensional space. In this way, each word will be represented by a vector in a predefined vector space. According to word embeddings, our neural network can be much more efficient in computation.

To translate words to vectors, we use the TF-IDF (Term Frequency – Inverse Document Frequency) method. This method can represent the relevance between a word and a document.

The way to calculate the TF-IDF is:

$$\mathrm{tf}\,(t, d) = \log\,(1+ \mathrm{freq}\,(t, d));$$

$$\mathrm{idf}\,(t, D) = \log\,(N / \mathrm{count}\,(d \in D : t \in d));$$

$$\text{tf-idf}\,(t, d, D) = \mathrm{tf}\,(t, d) * \mathrm{idf}\,(t, D).$$

According to this, we will represent all the documents by the TF-IDF values relating to each word. Our input to the algorithm will be a matrix which rows represent different sequences (sentences), columns represent different words and cells represent their corresponding TF-IDF values.

Since each sequence may have different length from others, we need to pad them in order to let them have the same length. We use the function pad_sequences provided by Keras to do this. After that, we can get our final embedding layer.

2.1.2 Convolutional Layer

In our algorithm, we use the two-dimensional convolution. We need to multiply the input data by a set of filters which are network parameters to be learned, and we also call them weights. The filter will be applied to the overlapping part of the input data from left to right and top to bottom. The output which is called the feature map computed by the convolutional layer is the dot product between the filters and spatial locations. So, after convolution, we will finally get several feature maps. Each of them is a matrix with neurons that share the same filter or we can say the same weights.

Once we have done this, we need to use ReLU as our activation function. ReLU function is represented in this way:

$$g(z) = \max\,(0, z);$$

We will pass all the values in the feature map through the ReLU function. The advantages of using ReLU as our activation functions are that the model is not linear due to clipping, no expensive functions are required as well as it is not saturate. After applying the activation function to each value in the feature map, we will get the final results as the input for the next layer.

2.1.3 Max Pooling

In Max pooling, it in every text after convolution computation of matrix to select the largest number of numerical, this to ensure that, under the condition of retain the characteristics of the original data, greatly reduce the complexity of the data, and originally produced by convolution of a multi-dimensional data into one-dimensional, so there is difference and start typing, However, it does not affect the final identification results. Next, the data with strong characteristics selected before are integrated in the fully connected method, which is to domesticate different categories and obtain a probability for different classification results after highly integrating the abstract data of the previous quarter. Finally, activate the functions RELU and Softmax to get all the classification probability of these data, and output the final classification result according to the probability.

2.1.4 Loss

Since it is a classification problem based on deep learning, we use cross entropy as our loss function, which can be represented in this way:

$$H (p, q) = \sum x\ p (x) * \log (1 / q (x));$$

We need to minimize the cross entropy loss on training dataset to get the parameters according to gradient descent with a given learning rate. Also, we will use it to measure the performance on validation dataset as well as the test dataset.

2.2 Text Classification

We don't use naive Bayes and KNN because to do that we need to represent the text as a vector, and we need to run the algorithm on the vector, and if we do that we need to take into account the similarity of different words in the text. We use the predefined word embedding provided in the library. In general, if the data is not embedded, there are many open source embedding methods available, such as Glove and Word2Vec. When we take the dot product representing text vectors, they may be zero, even though they belong to the same class, but if you embed the dot product of word vectors and find the similarity between them, then you will be able to find the interrelation of words for a particular class. We then slide filters/kernels over these inserts to find convolution, which further reduces dimension to reduce complexity and maximize pooling of the layer of computation. Finally, we have fully connected layers and an output activation function that will provide values for each class.

2.3 Training and Evaluation

To train the TextCNN model, we apply the training dataset to it. Once these training data goes through the embedding layer, convolutional layer, max pooling layer and fully connected layer, we will get the values of weights in filters in the convolutional layer. At the same time, we also need to calculate the cross entropy loss on the validation dataset. Then, we repeat this procedure until the cross entropy loss on the validation dataset becomes convergent. In our training, we stop repeating training before the loss starting to increase in the next iteration.

Then, we can get the final weights in the features in convolutional layer. Then, we can put these parameters into our model. And we can apply the test dataset to the trained model to check whether the cross entropy loss is small enough as well as to check the values of accuracy, precision, recall and f1-score.

3. Experimental Results

3.1 Text Situation

We design text classification experiment to evaluate the effectiveness and efficiency of our proposed method.The test data contain 150,000 text data records, which consists of pointed label and news data part. The experimental dataset is standard benchmark of Internet news data. The data samples are divided into three parts: training part, test part and verification samples. The training data part has 10 classifications with 65,000 text data sample records per category, which can divide into sports, finance, real estate, home, education, technology, fashion, politics, games and entertainment.After generating subsets from the original data, we only need to use one of them in this experiment. Now after computer execution, our data collections are divided into the following: training samples: 50000*10, test samples: 5000*10, verification samples: 10000*10.

3.2 Preprocessing and Embedding

In order to do experiment, we need to our data set. After reading the file data, we first build the vocabulary with the representation of the character set, which will avoid repeated processing. Then we use read_vocab(), read_category() and to_word() to read the vocabulary, fixed category directory and separate the text in our data-set into words. The next is we need to build word vector and sample matrix after text word segmentation. The process_file() can help us to converting the data set from text to a fixed length ID sequence representation, which can help computer match outcome. The categories convert ID such as the following Table1:

Table1. The categories convert ID

sports [1,0,0,0,0,0,0,0,0,0]	technology [0,0,0,0,0,1,0,0,0,0]
finance [0,1,0,0,0,0,0,0,0,0]	fashion [0,0,0,0,0,0,1,0,0,0]
real estate[0,0,1,0,0,0,0,0,0,0]	politics [0,0,0,0,0,0,0,1,0,0]
home [0,0,0,1,0,0,0,0,0,0]	games[0,0,0,0,0,0,0,0,1,0]
education [0,0,0,0,1,0,0,0,0,0]	entertainment [0,0,0,0,0,0,0,0,0,1]

This process also named build word vector, except category need to build vector, every words that separate from text all have their own word vector. About sample matrix, every sample is a vector, if there are 1hundred thousand sample, that means we have 1hundred thousand times 1hundred thousand dimensional matrix. After the matrix is formed, the computer can run later. Finally, batch_iter() can allow the shuffle batch data trained for the neural network has been preprocessing well. After data preprocessing, the data model are like Table 2:

Table 2. The data model

data	shape	data	shape
x_train	[50000, 600]	y_train	[50000, 10]
x_val	[5000, 600]	y_val	[5000, 10]
x_test	[10000, 600]	y_test	[10000, 10]

3.3 CNN Algorithm

When the data pretreatment is processed, we enter the convolutional neural networks(CNN). We need to build input layer and convolution kernel. One word are represents a neuron, so there are many huge number in the input layer, in order to make sure accurately data, we only use 64 data to train each time.

In the input layer, the embedding word vector dimension is 64, the sequence length is 600, the number of categories is 10. In the convolution kernel, the filter number is 128, the size of filter is 5. After convolution calculation, 128 fully connected neurons were obtained. Even the filter has filter a lot of small or negative weight number, we there are still have many data.

3.4 Max Pooling

Considering that too much data will not bring accurate data, we need to max pooling them.In the max pooling, only the largest features are retained in t, and various features extracted from the filters are filtered out. The data after max pooling process into dropout segment, which there are we use 0.5 retention ratio to filet data again. After dropout, we specify the RELU function(f(x) = max(x,0)) to activate instead of sigmoid function, because this function can remain all positive data and filter all negative data.

3.5 Fully Connected

The next step is softmax,softmax can normalize the values, means that put all the values between (0, 1). This step is for data sequence ID, which mention in the preprocessing. In the end, we use argmax to change the largest value to 1 and the rest to 0 in the matrix, and corresponding to the category of the corresponding ID, so as to get the answer.

Beside, we set the computer learning rate is 1e-3. The computer will train data at speed equal 1e-3. The batch_size is 64, which previously mentioned means that only 64 data are trained each time. In the experiment, the total number of iterations is 10. When the number of training reaches 100, the computer will output a result, so we set print_per batch = 100. When the training batch reaches 10, the results will be saved into tensorboard.

3.6 Experimental Results

For the CNN model, we start training by running python run_cnn.py train. The best outcome obtained on the validation set is 94.12%, and it has stopped after only 3 iterations. The accuracy and error are shown in the figure 1:

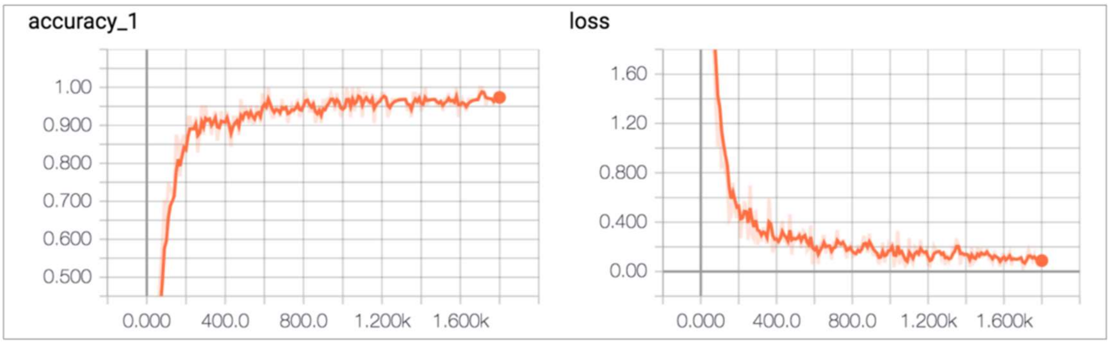

Figure 1. The accuracy and error of the CNN model

After that, we run python run_cnn.py test, and test on the test set. On the test set we get: The accuracy rate has reached 96.04%, and all kinds of precision, recall and f1-score have exceeded 0.9. It can also be seen from the confusion matrix that the classification effect is very good.So we used the following table 3 to show the data we tested.

Table 3. The test data

sports	0.99	0.99	0.99	1000
finance	0.96	0.99	0.97	1000
real estate	1.00	1.00	1.00	1000
home	0.95	0.91	0.93	1000
education	0.95	0.89	0.92	1000
technology	0.94	0.97	0.95	1000
fashion	0.95	0.97	0.96	1000
politics	0.94	0.94	0.94	1000
games	0.97	0.96	0.97	1000
entertainment	0.95	0.98	0.97	1000
avg/total	0.96	0.96	0.96	10000

For the RNN model, we start training by running python run_rnn.py train. The best outcome on the validation set is 91.42%. After 8 rounds of iteration stops, the speed is much slower than CNN. The accuracy and error are shown in the figure 2:

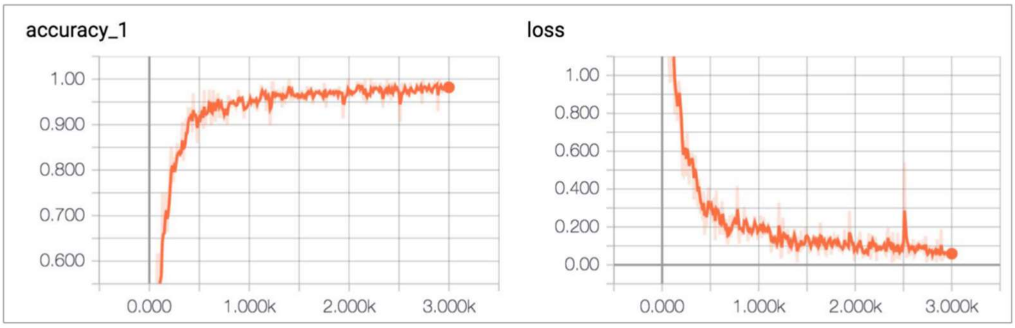

Figure 2. The accuracy and error of the RNN model

Similarly, we also ran python run_rnn.py test and test it on the test set. The accuracy rate on the test set has reached 94.22%, and all kinds of precision, recall and f1-score, except for the home furnishing and fashion, exceed 0.9. The recall and f1-score of home furnishing are only 0.73 and 0.83 respectively, and the precision of fashion is only about 0.89. We can get better results by further adjusting the parameters.

4. Related Work

4.1 Deep Learning

This data analysis and statistical modeling are belong the deep learning, Our target is according deep learning allow the computer have the analysis and learning ability and distinguish word. We choose to use Convolutional neural network(CNN) training to make deep learning. In our experiment, the core idea of convolution neural networks is that there are five layers, namely Embedding layer, convolution layer, pooling layer, full connection layer and soft-max layer. In CNN deep learning, our first step is to enter a word or a sentence, and then we go into the embedding layer, which is the vector of one word in each matrix, in each line. Then it goes to the convolutional layer, where there is a pile of convolutions in which the width of the convolution is fixed, which is the word vector dimension. The height of the convolution is the number of words contained in each window. The convolution of each class then activates the output of the convolution into the function output. The Max-pooling layer is then the maximum value of the results of each type of convolution that outputs several channels and then takes each channel. With the Max pooling layer, the Full Connection layer stitches the output of the Max Pooling layer together into inputs for a fully connected layer, and then the classification results can be obtained by adding a soft-max layer based on the number of categories.

4.2 Text Analysis

Text analysis have been very large use in the business now. TouTiao is a good example for computer text analysis that pushing the same type of news. In order to classify similar information, we use text analysis determines keywords and topics from millions of text data in different files and formats, which means text analysis is that computer extract the characteristics of text or sentence. However, the computer cannot read or analysis the text, which need us to transfer the no structure text to structured information that computer can distinguish. We build a math model instead of text and use vector space to describe text vector, but this makes the dimension of the vector very large. It will be very inefficient to let them learn directly with these vector dimensions. Therefore, the text vector must be further purified. First, we selected ten categories. Each categories have their own ID. The next step is segment word. In the experiment, we use the Chinese text that are do not like the English text having space, so, we need to segment sentence to the words. In order the computer can learning in the CNN model, we vector the data and build matrix. Beside, there are a lot of method that python can run of text analysis like dictionary based analysis, bag-of-word method, supervised model learning, unsupervised machine learning, natural language processing. In our experiment, we use the bag-of word method.

5. Conclusion

This paper applies convolutional neural networks to text classification tasks. In addition to using TF-IDF in embedding layer to transform text into word vector, ReLU function is added in two-dimensional convolution layer to make the network sparse and greatly improve the training speed. The maximum value of the feature value extracted from the fitter shall be retained in the pooling, and all other values shall be discarded. And the features are integrated in the full connection layer to play the role of "classifier". The results show that the operation effect of this simple CNN is good, and the accuracy in the test sample is high after training. It shows that our neural network can classify data sets with different classifications well.

References

[1] Kumar A;Bhatia MPS;Sangwan SR.Rumour detection using deep learning and filter-wrapper feature selection in benchmark twitter dataset,Multimedia tools and applications,2021,1-18.

[2] Huh J;Yetisgen-Yildiz M;Pratt W. Text classification for assisting moderators in online health communities,Journal of biomedical informatics, 2013, 46, 998-1005.

[3] Elhoseiny M;Elgammal A;Saleh B. Write a Classifier: Predicting Visual Classifiers from Unstructured Text, IEEE transactions on pattern analysis and machine intelligence, 2017, 39, 2539-2553.

[4] Fernandes R;D'Souza G L R. A New Approach to Predict user Mobility Using Semantic Analysis and Machine Learning, Journal of medical systems, 2017, 41, 188.

[5] Yuan H;Song Y;Hu J;Ma Y. Design of Festival Sentiment Classifier Based on Social Network, Computational intelligence and neuroscience, 2020, 2020, 8824009.

[6] Ramanujam N;Kaliappan M. An Automatic Multidocument Text Summarization Approach Based on Naïve Bayesian Classifier Using Timestamp Strategy, The Scientific World Journal, 2016, 2016, 17848 27.