IOP Conf. Series: Materials Science and Engineering **261** (2017) 012008 doi:10.1088/1757-899X/261/1/012008

Chinese Sentence Classification Based on Convolutional Neural Network

Chengwei Gu[1], Ming Wu[2] and Chuang Zhang[3]

[1, 2, 3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications
Xitucheng Road No.10, Beijing, P.R.China

[1]gcw831tjs@163.com, {[2]wuming, [3]zhangchuang}@bupt.edu.cn

Abstract. Sentence classification is one of the significant issues in Natural Language Processing (NLP). Feature extraction is often regarded as the key point for natural language processing. Traditional ways based on machine learning can not take high level features into consideration, such as Naive Bayesian Model. The neural network for sentence classification can make use of contextual information to achieve greater results in sentence classification tasks. In this paper, we focus on classifying Chinese sentences. And the most important is that we post a novel architecture of Convolutional Neural Network (CNN) to apply on Chinese sentence classification. In particular, most of the previous methods often use softmax classifier for prediction, we embed a linear support vector machine to substitute softmax in the deep neural network model, minimizing a margin-based loss to get a better result. And we use tanh as an activation function, instead of ReLU. The CNN model improve the result of Chinese sentence classification tasks. Experimental results on the Chinese news title database validate the effectiveness of our model.

1. Introduction

Sentence classification is the fundamental issue in NLP. Solving this problem opens many doors in NLP, such as information retrieval, machine translation and automatic digest. Since Sentence classification is such significant in NLP, there exists a lot of previous work in this field. The traditional sentence classification method is mainly based on the principle of statistics, using machine learning model. Using manually annotated data sets for supervised training to obtain classifiers, and then classify the new data. Among them, the general learning methods include Naive Bayes Classifier, Maximum Entropy Model, Logistic Regression, and so on. In the semantic classification task, the sentences have the following characteristics: (I) Single sentence language composition is small: each sentence contains a few words, up to dozens of words, some sentences do not even include the complete sentence composition, often pile the words; (II) The emergence of new words for the existing classifier is very difficult to deal with the unknown words (Out of Vocabulary). These characteristics lead to the traditional sentence classification method need to update the dictionary for the sentence data set and features, but this method is a drop in the bucket.

In this paper, we concentrate on classifying Chinese sentences by deep neural network. After a couple of pioneer works [1][2], neural network is showing to be effective in the research and applied to many NLP tasks, such as semantic parsing [3], translation task [4]. In the field of NLP, many deep learning models have been applied to the study of word embedding, and using word embedding to

accomplish a couple of tasks. The word embedding is a sparse V-dimensional (V is the size of the vocabulary) vector mapped to low-dimensional vector through the hidden layer, and each dimension can represent some semantic information. Neural network model was applied to computer vision research achieved excellent results [5], and then obtain good results in sentiment analysis, sentence retrieval, sentence model [6] and other NLP tasks. As a complicated architecture with huge number of parameters, it is rather important to have a careful parameter adjust together with a large scale of training samples. More specifically, we use the Chinese news title database to validate this deep neural model. Experimental results on the Chinese news title database confirm a performance improvement.

Recently, the success of the deep learning in the field of the NLP has attracted more attention. In contrast to the traditional machine learning methods, sentence representations learnt by deep network structure have shown their great potential in various NLP tasks. There are many models in deep learning fields, one of the greatest breakthroughs in sentence classification is deep CNN model. CNN has achieved good results in classifying corpus for short sentences. In this paper, we propose a novel and different CNN to extract high level sentence representations for classification. Compared to traditional models [7], we adjust the network architecture by accomplishing a linear support vector machine to substitute softmax, and optimizing the parameters by minimizing a margin loss. We also alter the activation function and adjust the neural network architecture. And in this paper the task we completed is the Chinese sentence classification, so we also need to segment the Chinese sentence. These changes improved deep learning model not only making up the shortfall in traditional machine learning representation extraction, but also gaining better performance in Chinese sentence classification.

2. Related work

Sentence classification has received great deal of attention in the last few decades [1] [7]. So there is a lot of research on sentiment analysis, or more generally on sentence classification tasks. Traditional methods always have two stage that one is extraction of hand-crafted features followed by a classifier. Typical features include Document Frequency [8], Information Gain, Cross Entropy, TF-IDF [9] and N-grams [10]. The appropriate features take an important role in accomplishing the final algorithm. And then algorithms take the major in training and testing work. So the extraction of features becomes the most important and most challenging work in sentence classification. Choose a good feature is half of the battle. These solutions are defeated by the deep learning models recently. We use Chinese corpora which applied by Sogou Labs for our experiments. At first, we use One-hot Representation to represent a word, the vector size is equal to the the number of vocabulary which can be large. Then, words have been mapped into a low-dimensional space using word2vec [11] which was proposed by Google. And then these word embedding has low vector size which then be put into the classifier. These solutions preform not good, because all words order is neglected.

Approaches that use CNN followed by a classical classifier have been proposed recently. In other words, the CNN herein is to extract features, to obtain good and high level sentence representations. A fairly shallow CNN model has been proposed by Kim [7]. This neural network architecture is rather simple. This model use word2vec toolkit to get word embedding as input. And then one convolutional layer followed by a max pooling layer. In this model, three filters of different sizes are used to complete the convolution. Then followed by a fully connected layer using drop-out to avoid over fitting. A similar model was proposed by Kalchbrenner [12]. A significant difference between these two models is that this model has many temporal k-max pooling layers and much deeper than previous models. This change makes it possible to get the k most vital representations in a sentence regardless of their position. Inspired by the previous work, we proposed a new and novel CNN model with SVM as classifier served as a supervised learning solution.

3. Model Architecture

The CNN model that we proposed is used for Chinese sentence classification. Our design for this model architecture has taken the risk of over fitting into account. The overall architecture of our model

is shown in Figure 1, the neural network consists of word embedding as input layer, two convolutional layers, tanh as activation function followed by two fully-connected layers with a small amount of neurons and SVM layer at last.

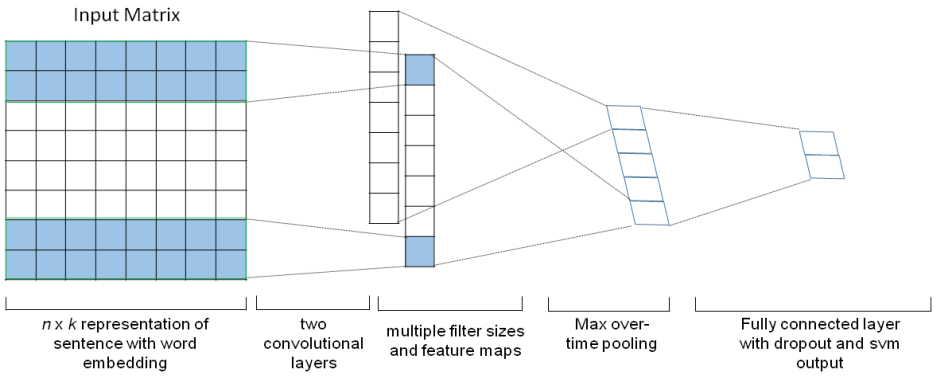

Figure 1. Model Architecture

3.1 Chinese Word Segmentation

In this paper, we use jieba-analysis toolkit to segment the SougouT corpus. We obtain the words after segmenting sentences. And then these words were put into the neural language model to get the word embedding.

3.2 Pre-trained Word Embedding

We initialize our word embedding with that acquired from an unsupervised neural language model. The model is a popular solution to improve performance under the condition that we do not have a very large supervised training set. We use the available word2vec toolkit which proposed by Google [11]. We use the SogouT corpus data set to train the word2vec toolkit to obtained the word embedding. We use the continuous bag-of-words architecture to obtain these vectors. Words that are not in the pre-trained words set are initialized randomly. The word embedding can be fine-tuned in our model. Word embedding in the paper we use are static and non-static. In our model architecture, showed in Figure 1, each filter is applied to two channels that one is static, another is fine-tuned. And then results are added to computed convolutional layer output.

3.3 CNN Architecture

We obtain the word embedding which is the k-dimensional word vector from the previous step. A sentence of length n can be represented as a $n * k$ matrix

$$x_{1:n} = x_1 * x_2 * x_3 \ldots * x_n \quad (1)$$

which * is the cascade operator. Generally, we use $x_{1:n}$ represent as cascading of words x_1, x_2, $x_3 \ldots x_n$, in other words sentence of length n. Then followed by a convolution layer. We use different size of filters to perform convolution on the input matrix. The filter $w \in R^{lk}$, which is applied to a window of length words to obtain a high level representation. For example, we use filter of length l to generate representation by

$$c_i = f(w \cdot x_{i:i+l-1} + b) \quad (2)$$

in the equation (2) b is a bias added to the convolutional layer out and f is a non-linear function called activation function. We use tanh to substitute ReLU in our model, because we acquired the promotion of result in the experiment. The filter is applied to produce a feature map

$$c = \{c_1, c_2, c_3, \ldots, c_{n-l+1}\} \quad (3)$$

with $c \in R^{n-l+1}$.

Most of previous convolution neural network model is fairly shallow which used to implement NLP task. This is due to the convolution of the token's n-gram feature, and the need for different n-gram length to model short and long-span relationship. In this paper, we then apply another convolution layer and a batch norm layer [13] and tanh activation function showed in Figure 2 in order to extract feature map. Next, the model followed by a max-pooling layer on the feature map and obtain the maximum value in the feature map as the representation corresponding to this size of filter. We use this max-pooling layer to to get the most essential feature in each feature map. The max-pooling operation can solve the different length of sentences problem, after this operation can make the output vector size be the size of the number of the filter. We extract one feature from one filter in this novel convolution neural network model. The model uses many filters with different slide window sizes to get a number of features. These features are then put into fully-connected layer, and then the output is fed to a SVM layer to complete training a neural network for classification. layer weights are updated via back propagating the gradients from the SVM layer. We use SVM layer substitute softmax to accomplish classification. The linear SVM layer is widely used for classification.

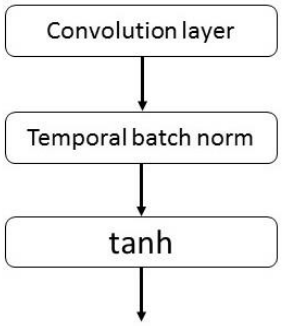

Figure 2. Convolutional layer Architecture

3.4 Regularization

We used dropout on the second last layer with a l2-nomalize constraint of the weight matrix. Dropout prevents over fitting via dropping out some connection randomly. That is, the second last layer output $= \{c_1 , c_2 , c_3, \dots , c_m\}$, instead of using

$$y = w \cdot z + b \tag{4}$$

for out y in the training step, dropout uses

$$y = w \cdot (z \; or \; 0) + b \tag{5}$$

Gradients are back propagated only via the units which are not dropout. At the test, we do not use dropout (not discard any of the network parameters to, then dropout layer output equivalent to input). Then, the learned weight vectors are scaled by p such that $\mathrm{w} = \mathrm{p} \cdot \mathrm{w}$, which p is Bernoulli probability.

4. Experimental Results

The deep neural network model we proposed in the previous chapter was compared with the classical CNN approach and machine learning model on Chinese sentence classification. And in the experiment tensorflow [14] was used to implement our model on the machine with Titan X GPU.

4.1 Dataset and Experimental Setup

We choose SogouT corpus to train neural language model via word2vec toolkit. Then we obtain word embedding as input. We choose Chinese news title corpus as training data and testing data. One news title data set 9k (NT-9k): This data set contains 9000 sets of training data, 1000 sets of cross validation data, and 4000 sets of test data. The data are divided into nine categories: domestic, international, social, financial, stock, technology, sports, military and entertainment. The maximum length of the sentence is 21. The entire data set contains 140195 words (including the placeholder for the empty

word). Another news title data set 16k (NT.16k): This data set contains 16,000 sets of training data, 1000 sets of cross validation data, and 4000 sets of test data.

4.2 Static Input vs. Fine-tune Input

We had expected that the fine-tune input would prevent over fitting (by ensuring that the learned vectors do not deviate too far from the original values) and thus work better than the static model, especially on smaller datasets. The results, however, are mixed, and further work on regularizing the fine-tuning process is warranted. For instance, instead of using an additional channel for the non-static portion, one could maintain a single channel but employ extra dimensions that are allowed to be modified during training. The part of classification results shows in Table 1.

Table 1. Part of classification results

	Domestic	**International**	**Social**	**Financial**	**Stock**	**Technology**
Static	0.842	0.874	0.811	0.823	0.812	0.857
Fine-tune	0.861	0.898	0.892	0.867	0.873	0.884

4.3 L2-SVM vs. Softmax

We compared performance among Naive Bayesian Model, the convolution neural network model with our novel model using SVM layer rather than softmax. All of models are tested using cross validation with the architecture we have introduced.

We may have a look at the validation curve of the softmax vs L2-SVM with the function of weight updates in Figure 3. When it came to the latter half stage of training process, weights were updated slower because learning rate get lowered. But our model with L2-SVM still had a little advantage with a lower averaged error rate.

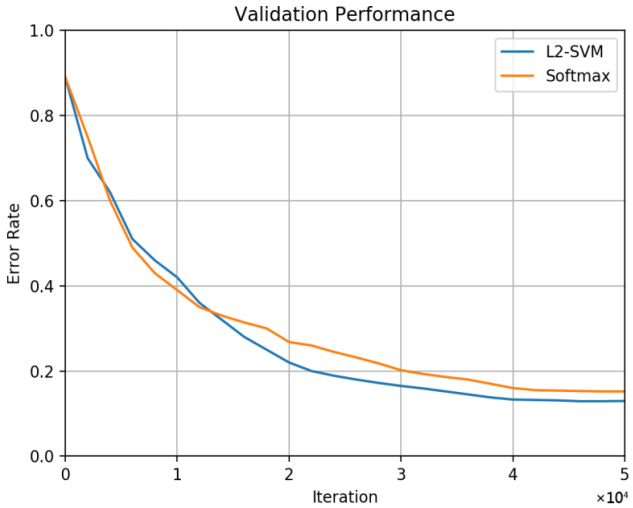

Figure 3. Validation performance between Softmax L2-SVM

4.4 Result

In order to validate our neural network model, we first compared it with traditional solution using traditional features have been mentioned previously, and next we compared our model with convolution neural network that is proposed by Yoon Kim[7], Different methods with its average accuracy on all category is shown as Table 2. Our model has a little promotion than other models. We set batch-size 64 through our experiment.

Table 2. Results of our CNN models against other methods

Models	Accuracy
Naive Bayesian Model	73.6%
CNN(Yoon Kim)	84.3%
CNN(Our Model)	87.1%

5. Conclusion

In the present work, we have proposed a new convolution neural network model architecture built on top of word2vec for Chinese sentence classification where a support vector machine was embedded to get learn a high-level hierarchical representation of a sentence. We train and test our model on the Chinese news title corpus and an experiment demonstrated the promotion that our model achieved. Our model prove that the input word embedding is effective to Chinese sentence classification. Although sentence abide by human rules and images are the original signals from the nature, short sentences and images have the same properties. In this paper, we concentrate on using convolutional neural network to accomplish sentence classification task. Applying similar model to other natural language processing tasks, in particular sentiment analysis is left for future research.

References

[1] Y. Bengio, R. Ducharme, P. Vincent. 2003. Neural Probabilitistic Language Model. Journal of Ma- chine Learning Research 3:1137–1155.
[2] Collobert R, Weston J. A unified architecture for natural language processing:deep neural networks with multitask learning[C]// International Conference. DBLP, 2008:160-167.
[3] Yih W T, He X, Meek C. Semantic Parsing for Single-Relation Question Answering[C]// Meeting of the Association for Computational Linguistics. 2014:643-648.
[4] Collobert R, Weston J, Karlen M, et al. Natural Language Processing (Almost) from Scratch[J]. Journal of Machine Learning Research, 2011, 12(1):2493-2537.
[5] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4):p ágs. 212-223.
[6] Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.
[7] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.
[8] Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF[J]. Journal of Documentation, 2004, 60(5):503-520.
[9] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization[C]// Fourteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. 1997:143-151.
[10] Damashek M. Gauging Similarity with n-Grams: Language-Independent Categorization of Text[J]. Science, 1995, 267(5199):843-8.
[11] Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method[J]. Eprint Arxiv, 2014.
[12] Conneau A, Schwenk H, Barrault L, et al. Very deep convolutional networks for text classification[J]. arXiv preprint arXiv:1606.01781, 2016.
[13] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, pages 448–456, Lille, France.
[14] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems[J]. 2015.