Deep Neural Network for Short-Text Sentiment Classification

Xiangsheng Li[1], Jianhui Pang[1], Biyun Mo[1], Yanghui Rao[1(✉)], and Fu Lee Wang[2]

[1] Sun Yat-sen University, Guangzhou, China
{lixsh6,pangjh3,moby5}@mail2.sysu.edu.cn, raoyangh@mail.sysu.edu.cn
[2] Caritas Institute of Higher Education, New Territories, Hong Kong
pwang@cihe.edu.hk

Abstract. As a concise medium to describe events, short text plays an important role to convey the opinions of users. The classification of user emotions based on short text has been a significant topic in social network analysis. Neural Network can obtain good classification performance with high generalization ability. However, conventional neural networks only use a simple back-propagation algorithm to estimate the parameters, which may introduce large instabilities when training deep neural networks by random initializations. In this paper, we apply a pre-training method to deep neural networks based on restricted Boltzmann machines, which aims to gain competitive and stable classification performance of user emotions over short text. Experimental evaluations using real-world datasets validate the effectiveness of our model on the short-text sentiment classification task.

Keywords: Neural network · Restricted Boltzmann machine · Pre-training · Short-text sentiment classification

1 Introduction

With an explosive growth of social media services, many online users can conveniently express their feelings through various channels. Facing such large-scaled sentimental documents, it is important for us to detect the sentiments from them automatically. Sentiment classification aims to identify and extract the user attitude towards an object. Unlike classifying sentiments over normal documents, short-text sentiment classification tends to be ambiguous without enough contextual information. Thus, conventional machine learning algorithms are difficult to be applied to short text directly.

Neural networks, although were applied to various natural language processing tasks, their performance were highly dependent on the adaptiveness of initializations if trained by raw features with random initialized weights [1]. Recently, Hinton and Salakhutdinov [2] indicated that deep neural networks have a better

H. Gao et al. (Eds.): DASFAA 2016 Workshops, LNCS 9645, pp. 168–175, 2016.
DOI: 10.1007/978-3-319-32055-7_15

ability of learning features, and the learned features have a better description of the original data, which are more appropriate for classification. However, deep neural network also has its disadvantages such as instability, gradient vanish and overfitting. Thus, Bengio et al. [3] proposed a method of greedy layer-wise training for deep networks which could effectively alleviate the problems of neural networks. In this paper, we employ the greedy layer-wise training as a pre-training method of our deep neural network model, in addition to validate its effectiveness on sentiment classification of short text. The general process of our model is as follows: First, we convert the high dimensional data into low dimensional expressions, which has the ability to reconstruct the original input vectors. Second, we use the logistic regression classifier at the last layer to predict emotional labels. Experimental evaluations validate that our model based on the pre-training method outperforms the baselines, as well as obtains much more stable performance than the conventional neural network on sentiment classification over short text.

The remainder of this paper is organized as follows. In Sect. 2, we summarize the related work on sentiment classification, short text modeling and deep neural network models. In Sect. 3, we describe our restricted Boltzmann machine and its training algorithms. Experimental evaluation is conducted on two datasets in Sect. 4. In Sect. 5, we present our conclusions.

2 Related Work

2.1 Sentiment Classification

Given unlabeled documents, the objective of sentiment classification is to estimate a score for each emotional label automatically. Pang et al. [4] evaluated three state-of-the-art machine learning algorithms on sentiment classification for movie reviews. Katz et al. [5] proposed a supervised unigram model (i.e., SWAT) to exploit user emotions with individual words. However, the performance of these methods are quite limited. Recently, Bao et al. [6] developed an emotion term (ET) method and an emotion topic model (ETM) to associate emotions with words and topics, respectively. Rao et al. [7,8] also focused on detecting user emotions by topics. The limitation of these models is that they need abundant features to gather enough statistics.

2.2 Short Text Modeling

To tackle the issue of lacking contextual information and the sparsity of content in short text, Sahami and Heilman [9] utilized the external documents collected from the web to expand the features. Banerjee et al. [10] applied the existing knowledge bases such as WordNet or Wikipedia to mine the semantic association between words. However, these methods are dependent on the quality of the external documents or knowledge bases, but we need a model which can automatically extract features implied from the sentence.

2.3 Deep Neural Network

As one typical stream of classification models, neural networks take the frequency of words in the short sentence or normal documents as the input and transfer it to a sequence of layers. The neural network can automatically extract features from the word-level up to the sentence and document levels. To compactly represent highly non-linear and highly-varying functions, deep multi-layer neural networks with more levels of non-linearities have been developed [3]. However, deep neural networks were found to be more difficult to train and the performance was even worse than neural networks with one or two hidden layers [11].

3 Restricted Boltzmann Machine

Restricted Boltzman machine (RBM) [2] is a generative stochastic neural network that is widely employed in deep learning and many other areas. As a variant of Boltzmann machines, RBM restrains that the neurons must form a bipartite graph. Figure 1 presents the framework of RBM models, which is a symmetric structure with a visible layer, a hidden layer and a bias unit.

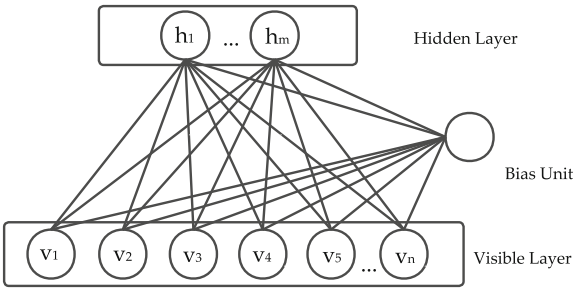

Fig. 1. Framework of RBM models

As shown in Fig. 1, $\mathbf{v}$ and $\mathbf{h}$ denote the visible layer and the hidden layer, which represent the input data and low dimensional features, respectively. $\mathbf{w}$ is the connective weight between the two layers. A joint configuration ($\mathbf{v}$, $\mathbf{h}$) of the whole RBM has an energy [12] as follows:

$$E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i h_j w_{ij}, \tag{1}$$

where v_i and h_j are binary states of visible unit i and hidden unit j, with biases a_i and b_j, and the weight w_{ij}. $\boldsymbol{\theta} = \{w_{ij}, a_i, b_j\}$ represents the parameters of RBM, which are all real numbers. Given these parameters, the joint distribution between the hidden layer and the visible layer is

$$P(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) = \frac{\mathrm{e}^{-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}, \tag{2}$$

where Z is a partition function by summing all possible pairs of visible and hidden vectors, i.e., $Z(\boldsymbol{\theta}) = \sum_{\mathbf{v},\mathbf{h}} \mathrm{e}^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}$.

Note that there is no direct connections between hidden units in RBM, the states of all hidden units are conditionally independent given the visible states. Due to the symmetrical structure of RBM, the states of all visible units are also conditionally independent given the hidden states. Thus, the activation probabilities of the hidden unit and the visible unit are represented as:

$$P(h_j = 1|\mathbf{v}, \boldsymbol{\theta}) = \sigma(b_j + \sum_{i \in visible} v_i w_{ij}), \tag{3}$$

$$P(v_i = 1|\mathbf{h}, \boldsymbol{\theta}) = \sigma(v_i + \sum_{j \in hidden} w_{ij} h_j). \tag{4}$$

where $\sigma(x)$ is the logistic sigmoid function, i.e., $1/(1 + exp(-x))$.

3.1 Objective Function

Our objective is to make the visible vectors close to the original input vectors as much as possible. The likelihood function of the input vectors given $\boldsymbol{\theta}$, i.e., the probability that RBM assigns to a visible vector is estimated by

$$P(\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\boldsymbol{h}} \mathrm{e}^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}. \tag{5}$$

Thus, the optimization problem is to minimize the negative log-likelihood function on the train data, as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{t=1}^{T} -logP(\boldsymbol{v}^{(t)}|\boldsymbol{\theta}), \tag{6}$$

where T is the size of batch training.

We then apply the stochastic gradient descent to solve the above function. The gradient of the log probability of the training vector is given by

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = -\sum_{t=1}^{T} \left(\langle \frac{\partial}{\partial \boldsymbol{\theta}} E(\mathbf{v}^{(t)}, \mathbf{h}|\boldsymbol{\theta}) \rangle_{data} + \langle \frac{\partial}{\partial \boldsymbol{\theta}} E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) \rangle_{model} \right), \tag{7}$$

where the angle brackets are used to denote the expectation under the distribution specified by the subscript. Finally, we can estimate the gradient in terms of w_{ij}, a_i and b_j by

$$\frac{\partial P(\boldsymbol{v}|\boldsymbol{\theta})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}, \tag{8}$$

$$\frac{\partial P(\boldsymbol{v}|\boldsymbol{\theta})}{\partial a_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}, \tag{9}$$

$$\frac{\partial P(\boldsymbol{v}|\boldsymbol{\theta})}{\partial b_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}. \tag{10}$$

Next, we employ the contrastive divergence gradient algorithm [13] to conduct pre-training for deep neural networks.

3.2 Pre-training Process

The pre-training process embedded in our model aims to make the reconstructed data close to the original data as much as possible, in which, all layers except the last classifier layer could be considered as encoder machines. Given a closer vector expression of the original data, the last layer for classification will become more robust. Algorithm 1 presents the method of pre-training via the contrastive divergence gradient [13]. The whole process of our model can be summarized as pre-training using the contrastive divergence gradient algorithm, and fine-tuning by the traditional back-propagation algorithm.

Algorithm 1. Pre-training using contrastive divergence gradient

Input:

1. $\boldsymbol{x_0}$: training samples
2. T: the size of batch training
3. m, n: the amounts of hidden and visible units

Output:

Gradient approximation: $\boldsymbol{\theta} = \{w_{ij}, a_i, b_j\}$

Initialize $\boldsymbol{v}_1 = \boldsymbol{x}_0$; $\boldsymbol{w}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are random values
for $t = 1$ to T **do**
  for $j = 1, \cdots, m$ **do** sample $h_{1j}^{(t)} \in \{0, 1\} \sim P(h_{1j}^{(t)} = 1 | \boldsymbol{v}_1^{(t)})$
  for $i = 1, \cdots, n$ **do** $v_{2i}^{(t)} = P(v_{2i}^{(t)} = 1 | \boldsymbol{h}_1^{(t)})$

Update parameters with a learning rate ϵ:
$\Delta \mathrm{w} \leftarrow \Delta \mathrm{w} + \epsilon(P(\boldsymbol{h}_1 = 1 | \boldsymbol{v}_1)\boldsymbol{v}_1 - P(\boldsymbol{h}_2 = 1 | \boldsymbol{v}_2)\boldsymbol{v}_2)$
$\Delta \mathrm{a} \leftarrow \Delta \mathrm{a} + \epsilon(\boldsymbol{v}_1 - \boldsymbol{v}_2)$
$\Delta \mathrm{b} \leftarrow \Delta \mathrm{b} + \epsilon(P(\boldsymbol{h}_1 = 1 | \boldsymbol{v}_1) - P(\boldsymbol{h}_2 = 1 | \boldsymbol{v}_2))$

Some strategies are also adopted in pre-training [14]. First, the hidden states are represented as the binary value rather than probabilities, because the hidden neuron is suitable to convey only one bit information as in biology. Second, the visible states are encoded by probabilities which may reduce the reconstruction error of the input data. After the process of pre-training, we use the traditional back-propagation algorithm to fine-tune the whole network, in which, the simple logistic regression is used in the last layer for sentiment classification.

4 Experiments

4.1 Datasets

SemEval is an English dataset used in the 14th task of the 4th International Workshop on Semantic Evaluations [15]. The dataset include news headlines and user scores over emotions of $anger, disgust, fear, joy$, $sadness$ and $surprise$.

Among 1,246 news headlines with the total score larger than 0, the first 1,000 and the rest 246 of them are used for training and testing, respectively.

ISEAR is a collection of 7,666 sentences annotated by 1,096 participants with different cultural backgrounds [16]. The emotional labels of this dataset are $anger, disgust, fear, joy, sadness, shame$ and $guilt$. We randomly select 60 percent of sentences as the training set, 20 percent as the validation set, and the rest as the testing set.

4.2 Experiment Setting

The experiment is conducted to validate the effectiveness of the deep neural network with pre-training (DNN) on sentiment classification of short text. We also implement the conventional neural network with only one hidden layer (NN_one) and a deeper structure with two hidden layers (NN_two) without pre-training for comparison. Gaussian random values with a mean value of 0 and different variances are used to initialize these models, so as to evaluate the stability of performance. To make an appropriate comparison with other baseline models, the accuracy at top 1 (*Accu*@1) is employed as the performance indicator [6].

4.3 Results and Analysis

The performance of DNN, NN_one and NN_two with different variances of initializations on *SemEval* and *ISEAR* is presented in Tables 1 and 2, respectively. The results indicate that DNN is more stable than others and NN_one outperforms NN_two for both datasets. In terms of *Accu*@1 values, NN_one performs better on the small-scaled *SemEval* dataset than *ISEAR*. Compared to NN_two, we also observe that the process of pre-training embedded in DNN with two hidden layers not only improves *Accu*@1 values but also alleviates the problem of instability. The reason is that pre-training could improve the effectiveness of parameter estimation for the deep neural network model, where the features approach an approximate low dimensional expression.

Table 1. Mean *Accu*@1(%) on *SemEval*

Variance	0.05	0.1	0.2	0.3	0.4	0.5
NN_one	36.2	38.5	37.9	37.3	35.2	35.2
NN_two	36.2	36.2	22.7	20.2	19.5	24.2
DNN	36.2	36.2	36.2	36.2	36.2	36.2

We also compare the performance of these methods with other existing models in Table 3, in which the best variance of initialization is determined by cross-validation for DNN, NN_one and NN_two. Compared to the baseline models of SWAT [5], ET and ETM [6], the *Accu*@1 of DNN improves 8 %, 16.8 %, 41.4 % on *SemEval* and 100 %, 2 %, 8 % on *ISEAR*, respectively.

Table 2. Mean *Accu*@1(%) on *ISEAR*

Variance	0.05	0.1	0.2	0.3	0.4	0.5
NN_one	42.4	50.1	49.0	47.2	27.9	35.4
NN_two	14.0	14.3	17.6	17.6	20.4	15.0
DNN	49.3	52.7	51.2	50.5	47.6	47.1

Table 3. Performance of different models

(a) *SemEval*

Model	*Accu*@1(%)
NN_one	38.5
NN_two	36.2
DNN	36.2
SWAT	33.5
ET	31.0
ETM	25.6

(b) *ISEAR*

Model	*Accu*@1(%)
NN_one	50.1
NN_two	20.4
DNN	52.7
SWAT	26.3
ET	51.7
ETM	48.8

5 Conclusion

In this paper, we incorporated a pre-training method into the deep neural network model for sentiment classification of short text. To evaluate the effectiveness of our model, we compared it with conventional neural network models and other existing methods using two different datasets. The results indicated that the deep neural network with pre-training performed competitively in terms of accuracy and robustness. This is because pre-training makes the output of each layer become a low dimensional code of the original data, which is more significant than conventional outputs.

For future work, we plan to improve the structure of our deep neural network by refining the sparse code of layers. With the rapid development of online communications, the method of extracting significant information from a non-standard short document also deserves further research.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (61502545, 61472453, U1401256, U1501252), the Fundamental Research Funds for the Central Universities, and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS11/E06/14).

References

1. Bengio, Y.: Learning deep architectures for ai. Found. Trends® Mach. Learn., vol. 2(1), pp. 1–127 (2009)
2. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Sci. **313**(5786), 504–507 (2006)

3. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. Adv. Neural Inf. Process. Syst. (NIPS) **19**, 153 (2007)
4. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
5. Katz, P., Singleton, M., Wicentowski, R.: Swat-mp: the semeval- systems for task 5 and task 14. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval), pp. 308–313(2007)
6. Bao, S., Xu, S., Zhang, L., Yan, R., Su, Z., Han, D., Yu, Y.: Mining social emotions from affective text. IEEE Trans. Knowl. Data Eng. **24**(9), 1658–1670 (2012)
7. Rao, Y., Li, Q., Mao, X., Wenyin, L.: Sentiment topic models for social emotion mining. Inf. Sci. **266**, 90–100 (2014)
8. Rao, Y., Li, Q., Wenyin, L., Wu, Q., Quan, X.: Affective topic model for social emotion detection. Neural Netw. **58**, 29–37 (2014)
9. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th International Conference on World Wide Web (WWW), pp. 377–386 (2006)
10. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 787–788 (2007)
11. Tesauro, G.: Practical issues in temporal difference learning. Mach. Learn. **8**(3–4), 33–53 (1992)
12. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. **79**(8), 2554–2558 (1982)
13. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. **14**(8), 1771–1800 (2002)
14. Hinton, G.: A practical guide to training restricted boltzmann machines. Momentum **9**(1), 926 (2010)
15. Strapparava, C., Mihalcea, R.: Semeval- task 14: Affective text. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), pp. 70–74 (2007)
16. Scherer, K.R., Wallbott, H.G.: Evidence for universality and cultural variation of differential emotion response patterning. J. Pers. Soc. Psychol. **66**(2), 310 (1994)