



# User Perspective Discovery Method Based on Deep Learning in Cloud Service Community

Lei Yu<sup>(✉)</sup>, Yaoyao Wen, and Shanshan Liang

Department of Computer Science, Inner Mongolia University, Hohhot, China  
yuleiimu@sohu.com

**Abstract.** The rapid development of cloud computing has promoted the coordinated integration of resources in various industries. In order to facilitate users' selection and invocation, more and more individuals and organizations have moved local application resources into the cloud service communities in the form of web services. In recent years, more and more people are interested in the emotional attitudes reflected in consumer reviews, but the sentiment analysis using the deep learning method to achieve evaluation of API (Application Programming Interface) services has received little attention. In order to explore the effective information of user's point of view data in the cloud service community, we propose an approach to analyze the user's opinion data using deep learning. We design three deep learning models of Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU). The result shows that the accuracy rate and recall rate of Bi-LSTM model is higher than the LSTM and GRU. Finally, we evaluate the performance of the three deep learning models, and choose the optimal Bi-LSTM model as the model used by the cloud service community in the future. According to the parameter comparison experiment of Bi-LSTM model, we obtained the optimal tuning of the model, and the model achieved the accuracy of 89.68%.

**Keywords:** Sentiment classification · Deep learning · LSTM · Bi-LSTM · GRU

## 1 Introduction

With the rapid development of Chinese cloud computing industry, the number of web services and their users is increasing in the cloud service community [1]. On the one hand, the rich web services provide a wide range of options for target users to meet various service requirements. On the other hand, it also imposes a heavy burden on the service choices of target users, especially when many candidate services have the same or similar functions, service users not only need to know the ability of a particular web service meet functional requirements, but also need to know whether the service can provide a satisfactory user experience [1]. Suppliers use consumer reviews as a new marketing tool to build their credibility. The growing number of online user reviews in the cloud service community, it has been a huge challenge to extract useful information from

a large amount of comment data. Traditional dictionary-based methods and machine-based learning methods have not been able to deal with the emotional classification of massive reviews. Therefore, in order to recommend better services to users and increase the revenue of service providers, it's important to establish an efficient sentiment model.

In this paper, we use deep learning methods to classify user option data in the cloud service community. The more positive comments of user reviews, the higher the trust of the service, and then the service providers and developers can recommend the trusted services to users.

## 2 Related Work

As our society becomes more and more connected, the number of online comments in the cloud service community is growing exponentially each year. The online reviews are the direct expression of the user's point of view, and they greatly affect people's behavior. The method of opinion-based discovery in the cloud service community is mining based on user's review data. With the development and widespread application of deep learning technology, many researchers have begun to apply deep learning models to the field of sentiment analysis, it uses deep learning technology to build an emotion classification model and judges people's implicit emotional tendencies in comment sentences.

### 2.1 Research of Sentiment Analysis

The concept of deep learning first appeared in 2006, limited by large-scale data volume and high-performance hardware, it has not entered people's field of vision until the last 5 years. Deep learning simulates the mechanism of the human brain and builds neural networks to adapt to people's daily affairs. A simple deep learning framework is based on sample data, stacking modules with learning capabilities in multiple layers. Deep learning is a machine learning method with a deep neural network model. It is very good at processing complex high-dimensional data and has been widely used in many fields such as science, business, and natural language processing.

Sentiment analysis is a field of natural language processing. In the process of emotion classification, the high-level representation of deep learning can emphasize important category information while suppressing irrelevant background information. The field of sentiment analysis research has expanded from movie reviews to stock message boards to congressional debates, and the research results have been industrially deployed in systems that measure market reactions and summarize opinions from Web pages, discussion boards, and blogs. With these widely changing fields, researchers and engineers constructing sentiment classification systems need to collect and organize data in each new field they encounter, and annotate a corpus for each field. However, in different fields, the expression of emotions is different, and it is impractical to annotate the corpus for each area of possible interest, especially because the product characteristics will change over time. We study the domain adaptability of sentiment classification, pay attention to online reviews of different types of service products, and combine the most advanced neural language models with sentiment information. This is still an area that needs to be explored.

In sentiment analysis tasks, there are usually three methods based on lexicon, machine learning and deep learning. The method based on sentiment lexicon is a typical unsupervised learning. Kim et al. used existing sentiment lexicons to judge the polarity of texts by adding the sum of emotional vocabulary scores [2]. The establishment of an emotional dictionary requires a large amount of manual participation, which makes it rare to use sentiment lexicon for emotional classification. Machine learning is another important method of sentiment analysis. Pang et al. proposed using machine learning models for sentiment analysis [3], and tried using three classification algorithms: naïve Bayes, support vector machine and maximum entropy. Kale et al. proposed a semi-supervised model, they attempted to use untagged data for rating prediction [4]. Wan et al. proposed a text classification method based on KNN and support vector machine to improve the accuracy of classification [5]. Bing Liu et al. proposed a model for mining and summarizing product reviews [6], which can summarize comments based on their characteristics. However, supervise machine learning methods require a large number of manually labeled training corpora to improve learning ability of the model, which is hindered in practical applications.

The emergence of deep learning methods has better compensated for the shortcomings of the above methods, and has achieved many results in specific topic sentiment analysis tasks.

In 2013, Wang et al. described a neural network architecture that attempts to exploit the CNN and RNN architectures [7]. In 2014, Kim proposed using CNN for sentence modeling, and achieved good results on multiple data sets [8]. Zhang Xiang et al. proposed a CNN model based on character level and used for text sentiment classification [9]. In 2015, Zhang et al. proposed a two-way door-loop neural network model for calculating emotions in target in tweets [10]. Graves et al. completed the identification of the 2013 phone number using the LSTM system [11]. Yao et al. further analyzed the role of the Bi-LSTM model in Chinese word segmentation [12]. Chiu et al. combined Bi-LSTM with CNN and demonstrated the good performance of the model [13]. In 2018, Hao Zhe Lin et al. proposed a model of GRNN to capture the intrinsic link of literature-level sentiment classification evaluation [14].

In China, He Yan Xiang et al. established the MCNN model to classify Chinese microblogs and fuse the emoji matrix in the semantic expression of words, which has practical significance [15]. Liu Rui Mei et al. used a deep convolutional neural network to analyze the emotions of multimedia images [16]. Jing zhi Gang et al. proposed a new sentiment analysis model Bi-LSTMM-B, through the improvement of Bi-LSTM model and Bagging algorithm, which shows that the combination of deep learning model and integrated learning thought can improve the accuracy of sentiment analysis [17]. Xiang Zhang et al. used character-level convolutional networks (Conv Nets) for emotional classification of texts [9].

## 2.2 Research of Service Recommendations

As the distribution model of information systems shifts to the XaaS paradigm, microservices architecture is rapidly emerging, taking the RESTful principle as its API model of choice. Researchers have been studying service data mining on various platforms to promote research on service recommendations and service composition. However,

researchers in the service computing field lack a cloud SaaS platform for hosting real-world services. The cloud service community we built is an independent SaaS platform that can connect API service providers and service users (including secondary developers and service users). The platform aggregates services into categories, and each category contains services with similar functions. The platform is committed to providing users with the most comprehensive and convenient services, as well as helping service providers open services and increase API calls. The platform has assembled more than 90 services required for application development, and unique services aggregate the platform's exclusive image processing, speech synthesis, ancient poetry synthesis, movie query, song query and other services. After registering an account, platform users can call all API services on the platform without jumping to a third-party website for operation. Users can also filter and accept services recommended by the platform, and can freely combine services. We use the cloud service community as a public data source, and users quickly generate and share data by using community information services. The cloud service community is a multi-information service domain consisting of multiple Mashups to meet comprehensive functional requirements and provide additional business value. Software service Web API usually provides black box functions to users through the Internet, which leads to a lack of reliability analysis of internal information.

With the development of popular computing concepts such as cloud computing, pervasive computing, and edge computing. Not only the underlying computing resources, but also enterprises, applications and data are open as Web services (usually Web API), making services on the internet unprecedented prosperity. Web services are black box software services, which provide software components as building blocks for enterprise application integration, effectively promoting the service deployment of cloud computing resource pool. The number of web services in the cloud resource pool are growing rapidly, so we need some new ways to make accurate service recommendations and choices. Providing accurate service recommendations to users is critical to improving the efficiency and success rate of application deployment in the cloud service community. In order to meet people's needs, Web service recommendation systems such as Web Service List, Web Service X, Programmable Web [18], Remote Methods [19] and Service-Repository [20] have appeared on the cloud service market. As the number of user comments increases, it is impossible for service provider to read all the comments. Therefore, it is important to mine user comments and extract valid information. In 2017, H. Wang et al. extracted the fine-grained value characteristics from customer reviews and identified the personalized distribution of each value feature [21], it shows the value preference of specific customers. They proposed a VFmine algorithm based on text mining, it can extract value characteristics from customer reviews.

According to the above analysis, we attempt to use deep learning method to extract fine-grained and value information from user comments and analyze the user's value preference.

### 2.3 LSTM and Bi-LSTM

According to the deliberate design, LSTM is able to avoid long-term dependence, it can remember the long-term information. It has strong timing signal processing capabilities. The difference between LSTM and RNN is that the LSTM has a gate structure. The gate

is a structure that determines whether information can pass, it can control the flow of information, prevent information from fading away, and is more likely to capture long-term dependencies than RNN. The architecture of LSTM is shown on Fig. 1. It has three kinds of gate structures: input gate, output gate and forgetting gate. The distribution is represented by  $i_t$ ,  $f_t$  and  $O_t$ ,  $C_t$  is the cell activation state, and the multiplication gate enables the LSTM memory cell to store and access information for a long time, thereby reducing the gradient disappearance problem. LSTM memory unit algorithm is as follows:

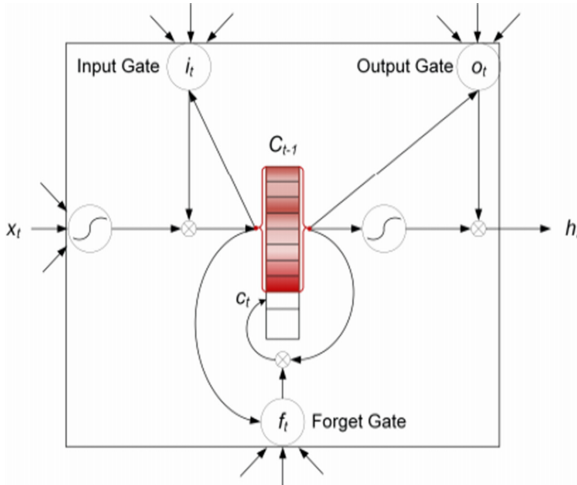


Fig. 1. LSTM single cell door structure

Forgetting gate decides when you need to forget the previous state.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{1}$$

$W$  represents the weight matrix,  $b$  is the offset term,  $\sigma$  is a nonlinear function applied to the hidden layer unit. The forgetting gate is used to output a value between 0 and 1 for each digit in the cell state  $c_{t-1}$  value. The next step is to decide what information to store in the cell state, which consists of two parts: in the first part, the input gate will decide which state to update.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2}$$

Next, the tanh layer adds a new cell state value vector  $c_t$  to the state, and we will combine them to create an update.

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

Then we can update the old cell  $c_{t-1}$  to a new cell state  $c_t$ .

We multiply the old state by  $f_t$  to discard the information that needs to be discarded, and plus  $i_t$  multiply  $\tilde{c}_t$  to get the cell state to get new information.

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \tag{4}$$

Finally, in the output gate, the sigmoid layer is used to determine which part of the information to output.

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{\infty}c_{t-1} + b_o) \tag{5}$$

We have  $x_t^c = [x_t, h_{t-1}]$ , so the formula  $h_t = o_t \times \tanh(c_t)$  can be written as follows:

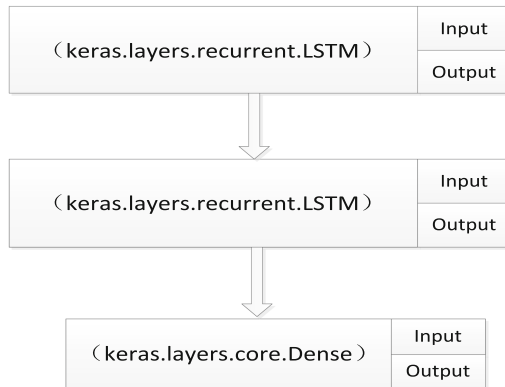
$$f_t = \sigma(W_f x_t^c + b_f) \tag{6}$$

$$i_t = \sigma(W_i x_t^c + b_i) \tag{7}$$

$$\tilde{c}_t = \tanh(W_g x_t^c + b_o) \tag{8}$$

$$o_t = \sigma(W_o x_t^c + b_o) \tag{9}$$

In this paper, we superimposed two LSTM layers, each using 20% Dropout to prevent overfitting, and finally adding a fully connect layer to output the result. Different LSTM layers capture different fluctuations, so the model has the ability to perform higher-level timing expressions. The first layer LSTM needs to return the complete sequence data for the next layer. At the same time, in the second layer LSTM, we do not return the original sequence data, but return the calculated sequence data for the output layer. The model architecture is shown below (Fig. 2).



**Fig. 2.** Stacked neural network architecture

Because LSTM modeling has the problem of not being able to encode information from back to front, we use bidirectional LSTM to capture bidirectional semantic

dependencies. Bi-LSTM is used to upgrade the input information block of the available network. It is composed of forward LSTM and backward LSTM. The structure is shown in Fig. 5. By using two times directions, Bi-LSTM can better understand the context. In this paper, after the vectorized representation of the text, the Bi-LSTM neural network model is used to learn the text vector to obtain the forward hidden state vector  $\vec{h}_t$  and reverse hidden state sequence  $\overleftarrow{h}_t$ . By concatenating these two output sequence vectors, the final output vector is  $h_t = \vec{h}_t \parallel \overleftarrow{h}_t$  (Fig. 3).

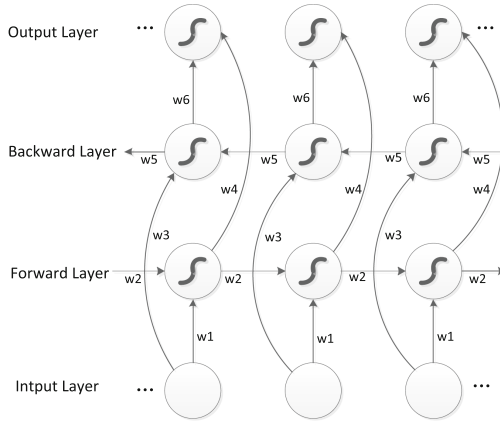


Fig. 3. Bidirectional LSTM

### 2.4 GRU

Because of the complex structure of both forgetting gates and input gates used by LSTM, Chung et al. proposed the GRU in 2014 [22], which preserves the resistance of LSTM to the vanishing gradient problem, but its internal structure is simpler than LSTM, so it is easier to train. Its internal structure is shown in the Fig. 4.

The GRU has an update gate  $z_t$  and reset gate  $r_t$ . The update gate is a combination of the input gate and the forget gate in the LSTM model. The output gate in the LSTM model functions is similar with the reset gate. The update gate  $z_t$  determines the integration of the new input information and the historical information, that is, how much previous information is to be retained, and the reset gate  $r_t$  determines the proportion of the state information in the model, that is, the new input information and the previous information can be well combined. GRU simultaneously combines the cell state and output into one state parameters. As the number of gates changes from 3 to 2, the training parameters decrease and the training speed increases.

Similar to LSTM, GRU uses the hidden state at time t-1 and the input time series value at time t to calculate the hidden state output at time t. The deduction formula of GRU neural network is:

$$z_t = \delta(W_z \times [h_{t-1}, x_t]) \tag{10}$$

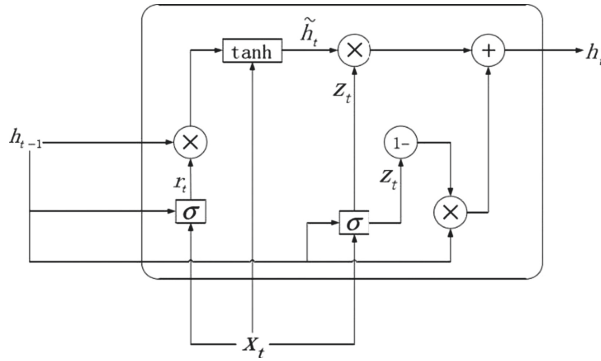


Fig. 4. A cell for GRU

$$r_t = \delta(W_r \times [h_{t-1}, x_t]) \tag{11}$$

$$\tilde{h}_t = \tanh(W \times [r_t \times h_{t-1}, x_t]) \tag{12}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \tag{13}$$

$h_{t-1}$  represents the output of the previous neuron,  $x_t$  represents the input of the current neuron,  $W_z$  represents the weight of the update gate,  $W_r$  represents the weight of the reset gate,  $\tilde{h}_t$  represents the pending output value in this neuron, and  $h_t$  represents the output of the current neuron,  $\delta$  represents the sigmoid function, and  $\tanh$  represents

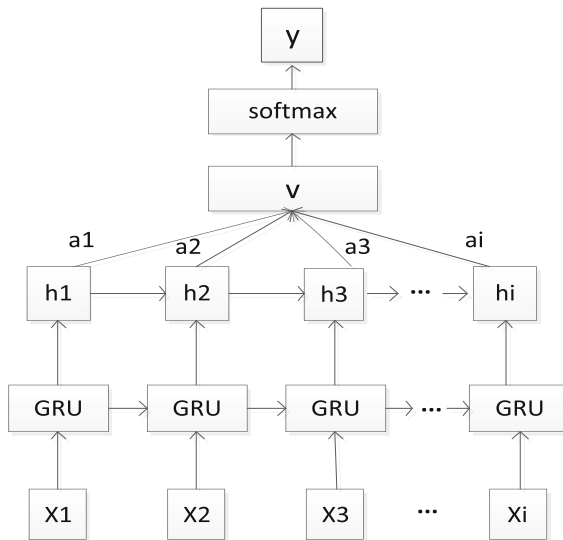


Fig. 5. GRU model structure

the hyperbolic tangent function. In this paper, we construct a conventional GRU neural network in the loop layer. The training flowchart of the model is shown in Fig. 5.

### 3 Experiment

We use the user comment data in the cloud service community as the experimental data set, it has 40411 comments. The community allows users to rate and comment on more than 90 different services such as gold inquiries, mobile phone number inquiries, and weather forecasts in the community. There are 10128 positive comments, 9573 neural comments, and 10710 negative comments. Since the data extracted from the cloud service community contains a lot of irrelevant information, we need to delete the irrelevant data and segment the data. Through data preprocessing, the total number of samples in the review data set is 24082, there are 8050 positive comments, 7902 neural comments and 8103 negative comments.

#### 3.1 Dataset

Before constructing the user perspective propensity model, we divide the review text after data preprocessing and word vector representation into three parts: training set, verification set and test set. The distribution of various types is shown in Table 4. The training set is used to train the deep learning, the validation set is used to adjust the parameters of the model, and the test set is used to verify the results of the model (Table 1).

**Table 1.** Type assignment of a data set

Type of data	Number	Proportion (%)
Training set	12000	49.89
Validation set	7200	29.93
Test set	4855	20.18
Total	24055	100.00

#### 3.2 Experiment Evaluation Criteria

The experiment designed two sets of comparative experiments, one compared to the traditional LSTM model and one compared to the GRU.

In order to select the best model among the three neural network models of Bi-LSTM, LSTM and GRU, we need to evaluate the performance of the three models. We use accuracy as an evaluation index, and use precision, recall and F1 values to compare the performance of the three deep learning models. The following describes each indicator (Table 2):

**Table 2.** Confusion matrix

		Predicted emotion		
		Positive	Neutral	Negative
Actual emotion	Positive	T1	F1	F2
	Neutral	F3	T2	F4
	Negative	F5	F6	T3

$T_i$  ( $i = 1, 2, 3$ ), it represents the number of samples whose sentiment classification is correctly predicted.

$F_i$  ( $i = 1, 2, 3, 4, 5, 6$ ), it represents the number of samples whose sentiment classification is incorrectly predicted.

1) Accuracy.

Accuracy represents the ratio of the number of samples correctly predicted and the total number of samples by the classification model.

$$Accuracy = \frac{\sum_{i=1}^3 T_i}{\sum_{i=1}^6 F_i + \sum_{i=1}^3 T_i} \quad (14)$$

2) Precision.

Precision measures the ability of the classifier that not divide negative sample errors into positive samples.

$$Precision_{positive} = \frac{T1}{T1 + F1 + F2} \quad (15)$$

$$Precision_{neutral} = \frac{T2}{T2 + F3 + F4} \quad (16)$$

$$Precision_{negative} = \frac{T3}{T3 + F5 + F6} \quad (17)$$

3) Recall.

The recall rate is the ability of the metric classifier to find all the correct classification samples.

$$Recall_{positive} = \frac{T1}{T3 + F3 + F5} \quad (18)$$

$$Recall_{neutral} = \frac{T2}{T2 + F1 + F6} \quad (19)$$

$$Recall_{negative} = \frac{T3}{T3 + F2 + F4} \quad (20)$$

## 4) F1.

The F1 value takes into account the precision and recall of the prediction result, it is the weighted harmonic mean of precision and recall.

$$F - measure = 2 \frac{precision \times recall}{precision + recall} \quad (21)$$

The value of the indicators obtained in the final training of the model are shown in Table 3.

**Table 3.** Classification results of the three models

Model	Accuracy	Recall	Precision	F1 value
LSTM	0.8741	0.8728	0.8751	0.8740
Bi-LSTM	0.8968	0.8766	0.8950	0.8854
GRU	0.8798	0.5746	0.8738	0.6905

It can be seen from the data in the table that the accuracy of the three neural network models of LSTM, Bi-LSTM and GRU exceeds 85%. The GRU model has fewer tensors and takes the shortest time, but the F1 value is the smallest. The Bi-LSTM model has exceeded the traditional LSTM model and GRU model, so this paper chooses Bi-LSTM model as the deep learning model used by the cloud service community in the future.

### 3.3 Model Parameter Comparison Experiment

In this paper, the Bi-LSTM model is used as an experimental model to study the effects of different parameters on the model. We designed a comparison experiment for the three parameters of Epoch value, Batchsize value and Dropout value.

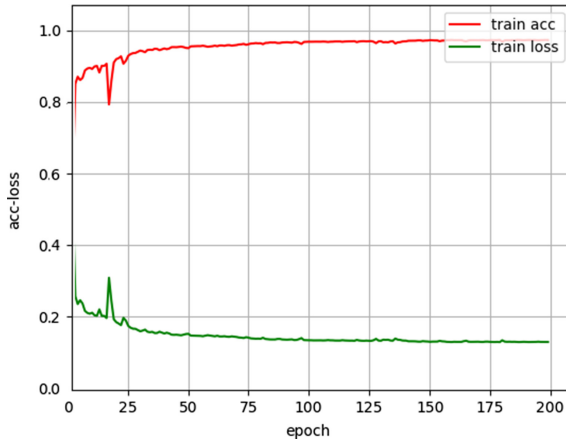
**Experiment 1.** Examine the effect of Epoch parameters on the model result.

Epoch is the number of iterations of training. We keep the parameters unchanged other than Epoch. The accuracy and loss of the neural network model are show in Fig. 6.

Through the above training curve, we found that the training results of the model are good, the loss is decreasing, and the accuracy of the training set is close to 100%. We can see that as the number of iterations increases, the accuracy of the training set increases gradually, and the loss value gradually decreases. From this we can determine that the size of the Epoch value has an effect on the experiment results. In this paper, the Epoch value is 1000 and the accuracy of the training set remains basically unchanged.

**Experiment 2.** Investigate the impact of Batchsize parameters on the model result.

Batchsize is the amount of sample data that passed into the deep learning model each time. The size of the Batchsize value affects the optimization degree and speed of the experiment. If the value is too small, the network will not converge, and too large



**Fig. 6.** Curve of accuracy and loss value variation

**Table 4.** Comparison of experiment results and usage time under different Batchsize

Batchsize	Average time (s/epoch)	Accuracy (%)
8	250	88.62
16	169	88.60
32	165	89.68
64	118	88.81
128	111	88.74
256	107	88.10

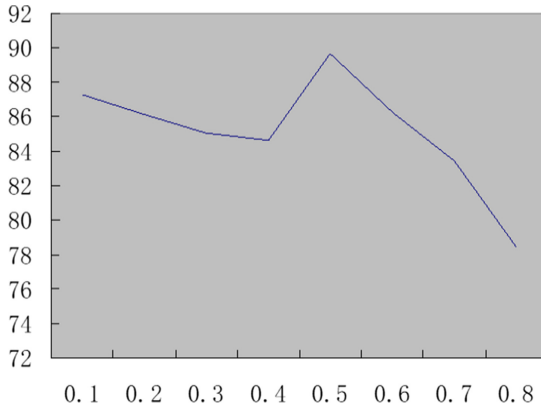
may cause a memory explosion. We set difference Batchsize value while keeping other parameters unchanged. The training results are shown in Table 4.

As we can see from the table, the larger the value of Batchsize is, the longer the average time of each iteration training is. In order to improve the memory utilization and the parallel efficiency of matrix multiplication, and considering the time factor, the Batchsize value selected in this paper is 64.

**Experiment 3.** Examine the effect of Dropout value on the model result.

The feedforward artificial neural network has multiple layers of nonlinear “hidden” units between its input and output. Dropout allows the model to randomly omit each hidden unit from the network during the training process, so that the hidden unit cannot depend on other existing ones. In this paper, we keep the other parameters unchanged other than dropout values. The training result is shown in Fig. 7.

As we can see from the figure, when dropout takes 0.8, the model obtains the minimum value, and the accuracy of the model is lowest. When taking 0.5, the accuracy is the highest. This shows that the deep learning model cannot implement abandonment



**Fig. 7.** Training results under different Dropout conditions

strategy for too many neural units during training, that is, the value of dropout cannot be set too large.

## 4 Conclusions

With the growing number and diversity of web services in the cloud service community, and the number of similarity-featured web services is innumerable, it's difficult to find the most suitable web service for users. In order to recommend more satisfactory services to users, we take users as the center and uses the deep learning method to analyze the online customer reviews of cloud service providers in the community. We designed the performance evaluation and comparison experiments of three deep learning models, and selected the best performance model as the model used by the developers of the cloud service community. The result shows that the performance of the Bi-LSTM model is higher than the LSTM and GRU models.

In future work, we plan to add the convolutional neural network, the deep learning methods used in this article are all cyclic neural networks, so we decide to add convolutional neural networks, they can recognize emotions in comments texts and increase the breadth of user opinions.

**Acknowledgment.** This work was supported by grants from National Natural Science Foundation of China under Grant (NSFC No. 61962040).

## References

1. Qi, L., Xiang, H., Dou, W., Yang, C., Qin, Y., Zhang, X.: Privacy - preserving distributed service recommendation based on locality- sensitive hashing. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 49–56, Honolulu, HI (2017)

2. Gamon, M., Aue, A.: Proceedings of the Workshop on Sentiment and Subjectivity in Text (2006)
3. Bo, P., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10. Association for Computational Linguistics (2002)
4. Liu, X.M., Kale, A., Wasani, J., Ding, C., Yu, Q.: Extracting ranking and evaluating quality features of web services through user review sentiment analysis. In: Proceedings of the 2015 IEEE International Conference on Web Services (ICWS 2015), pp. 153–160 (2015)
5. Wan, C.H., et al.: A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Syst. Appl.* **39**(15), 11880–11888 (2012)
6. Wan, C.H., et al.: A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Syst. Appl.* **39**(15) 11880–11888 (2012)
7. Wang, X., Jiang, W., Luo, Z.: Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: Proceedings of the 26th International Conference on Computational Linguistics, pp. 2428–2437 (2013)
8. Kim, Y.: Convolutional neural networks for sentences classification. arXiv preprint [arXiv: 1408.5882](https://arxiv.org/abs/1408.5882) (2014)
9. Zhang, X., Zhao, J., Yann, L.: Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **28**, 649–657 (2015)
10. Zhang, M., Zhang, Y., Vo, D.: Gated neural networks for targeted sentiment analysis (2015)
11. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE (2013)
12. Yao, Y., Huang, Z.: Bi-directional LSTM recurrent neural network for Chinese word segmentation. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) *ICONIP 2016*. LNCS, vol. 9950, pp. 345–353. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46681-1\\_42](https://doi.org/10.1007/978-3-319-46681-1_42)
13. Chiu, J.P.C., Eric, N.: Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **4**, 357–370 (2016)
14. Lin, H., Fan, Y., Zhang, J., Bai, B.: PRNN: piecewise recurrent neural networks for predicting the tendency of services invocation. In: IEEE International Conference on Web Services (ICWS), pp. 56–64 (2018)
15. He, Y., Sun, S., Niu, F.: An emotional semantic enhanced deep learning model for microblog emotion analysis. *Chin. J. Comput.* (4) (2017)
16. Liu, R., Meng, X.: Emotional analysis of multimedia picture based on deep learning. *Res. Audio-Vis. Educ.* (4) (2018)
17. Jin, Z., Han, Y., Zhu, Q.: An emotional analysis model combining deep learning and integrated learning. *J. Harbin Inst. Technol.* **50**(11), 38–45 (2018)
18. Maximilien, E.M., Ranabahu, A.: The ProgrammableWeb: agile, social, and grassroots computing. In: International Conference on Semantic Computing (ICSC 2007), pp. 477–481, Irvine, CA (2007)
19. <https://loopback.io/doc/en/lb3/Remote-methods.html#19>
20. Weiping, L., Weijie, C., Li, L.: Fuliang, G.: A semantically enhanced service repository for service oriented application system development. In: 2009 World Conference on Services - II, pp. 41–48, Bangalore (2009)

21. Wang, H., Chi, X., Wang, Z., Xu, X., Chen, S.: Extracting fine-grained service value features and distributions for accurate service recommendation. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 277–284, Honolulu, HI (2017)
22. Chung, J., Gulceher, C., Cho, K.H., et al.: Empirical evaluation of gated recurrent neural networks on sequence modeling. Eprint Arxiv (2014)