Performance Analysis of Hyperparameters on a Sentiment Analysis Model

Irfan Ali Kandhro
Department of Computer Science
Sindh Madressatul Islam University
Karachi, Pakistan
irfan@smiu.edu.pk

Sahar Zafar Jumani
Department of Computer Science
Sindh Madressatul Islam University
Karachi, Pakistan
sahar@smiu.edu.pk

Fayyaz Ali
Department of Software Engineering
Sir Syed University of Engineering and Technology, Karachi, Pakistan
fayyaz.ali@ssuet.edu.pk

Zubair Uddin Shaikh
Department of Software Engineering
Sindh Madressatul Islam University
Karachi, Pakistan
zubair@smiu.edu.pk

Muhammad Arshad Arain
Department of Software Engineering
Sindh Madressatul Islam University
Karachi, Pakistan
m.arshad@smiu.edu.pk

Aftab Ahmed Shaikh
Department of Software Engineering
Sindh Madressatul Islam University
Karachi, Pakistan
aftab.shaikh@smiu.edu.pk

Abstract—This paper focuses on the performance analysis of hyperparameters of the Sentiment Analysis (SA) model of a course evaluation dataset. The performance was analyzed regarding hyperparameters such as activation, optimization, and regularization. In this paper, the activation functions used were adam, adagrad, nadam, adamax, and hard_sigmoid, the optimization functions were softmax, softplus, sigmoid, and relu, and the dropout values were 0.1, 0.2, 0.3, and 0.4. The results indicate that parameters adam and softmax with dropout value 2.0 are effective when compared to other combinations of the SA model. The experimental results reveal that the proposed model outperforms the state-of-the-art deep learning classifiers.

Keywords—student feedback; sentiment analysis; performance analysis; LSTM

I. INTRODUCTION

In an Education Management System (EMS), assessing the performance of faculty members is becoming an important component. It’s not only helpful for improving the quality of course content and teaching style but it is also used by the faculty annual appraisal process. The course evaluation is typically collected at the end of the semester of each course and a set of questions are answered in Likert scale, open-ended, and self-evaluation approach. The combined response is used as a metric to measure the quality of the teaching staff. The evaluation form also provides room for open feedback which is typically not entertained in the performance appraisal due to lack of automated methods [1-3]. The textual data may contain some important information about the subject understanding, comprehension, regularity, and presentation skills and may also provide clear suggestions for improving the quality of teaching. This kind of information may not get from the Likert scale-based feedback [4]. And conversely, getting sense out and understanding the semantic of text from the textual feedback manually is a painstaking task and as a result, and textual feedback is not properly utilized [3]. The main aim of this paper is to analyze and understand the textual feedback automatically and develop qualitative and quantitative metrics that can estimate the performance of a teacher. This work comes under the promising and emerging area of opinion mining which has gained eminence since the uprising of the World Wide Web. A lot of relevant research has been reported recently. Researchers have extracted sentiments from comments posted online in websites and forums [5], movie and other review sites [6-7], social networking sites [8-9], course comments posted online in websites and forums [5], movie and other review sites [6-7], social networking sites [8-9], course evaluation dataset. The performance was analyzed regarding hyperparameters using student feedback dataset. The main focus of a sentiment analysis model is on extracting and determining the writer’s feelings form a piece of text. The feeling might be his or her opinion, emotion, and attitude. The most valuable step of this analysis is to classify the polarity of the given text as positive, neutral, and negative [5, 11]. Similarly, the obtained work aims to categorize the polarity of student comments in terms of these three labels. This paper suggests suitable hyperparameters for training and testing a Sentiment Analysis (SA) model and provides a comprehensive strategy for investigating the effects of the hyperparameter tuning model with deep learning LSTM approach. The experiment was carried out with different tuning strategies to induce and evaluate the relevance of hyperparameters using student feedback dataset.

II. RELATED WORK

Table I shows some related work carried out using combined methods of machine learning, deep learning, and other conventional techniques. This part briefly summaries studies interrelated to the sentiment analysis of web contents regarding user emotions, opinions, and reviews towards different matters and products using deep learning approaches. The opinion mining task can be powered by various models such as deep learning models. These models include Recurrent Neural Networks, Recursive Neural Networks, Convolutional Neural Networks, and Deep Neural Networks. This section
The adopted system analyzed sentiments from macro and microblogs. The core reason for this study was to get user opinions and attitudes about hot topics and events by implementing CNN. CNN prevails over the problem of explicit feature extraction and learns completely through the training data. To gather the data from the target, the input URL and feature extraction and learns completely through the training comments were collected and divided into three labels: 300 negative, 274 neutral, and 426 positive. This study was compared with previous studies which used SVM, CRF and additional methods to perform SA [26].

### III. METHODOLOGY

The presented methodology classifies the students’ sentiments as positive, neutral and negative. The model workflow is shown in Figure 1 and is analyzed below.

#### A. Data Preprocessing

The collected dataset is not well organized and in order to extract the meaning and information from the text we need strong data preprocessing techniques. There are several steps applied for the removal of spelling errors, grammatical mistakes, and URLs. The details are described below:

- Punctuation consists of the special symbols and numbers, which were removed from the text, as these symbols are useless and only create ambiguity in processing.
- Tokenization is the process of splitting a sentence into words.
- After tokenization, case conversion is performed to convert the uppercase tokens into lower case i.e. (GOOD, goood).
- In NLP, stop words are a set of commonly used words such as determiners, conjunctions, and prepositions. These words are worthless for sentiment analysis and classification, and they are removed before training the model.

#### B. Word Embedding

The word embedding presents a dense representation of words and their relative significance. It can be learned from text data and reused among various applications. The word embedding maintains the relations of words, and captures context and semantics of particular words in text documents. In this model we used a pre-trained Word2vec model as input in our LSTM network and that model produced 300-dimensional vectors for processing the millions of words and get support from the bag-of-words scheme.

---

**Table I. MISCELLANEOUS TECHNIQUES**

| Ref. | Techniques | Dataset | Size | Limitation | Result |
|------|------------|---------|------|------------|--------|
| [1]  | SRS, NRC Lexicon | Students comments (Coursera dataset) | 4000 | It is good only when data are cleaned and formal | Overall 90%, same teacher (85% P and 15% N) |
| [2]  | Deep feed forward neural network | Twitter database | 2000 | Limited sample | 75% |
| [3]  | SVM sentiment classifier LSA-based filtering | Internet blogs Chinese movie reviews | 8000 | Small display capability of cellular devices | 85% |
| [4]  | DCNN+LSTM | Twitter data | 3,813,173 | It cannot process very long sequences if using tanh as its activation function | Better accuracy than SVM and Naive Bayes and less maximum entropy |
| [5]  | Analyzing financial news using lexical approach | Newspapers (The Star Online, National News Agency of Malaysia) | 200 | Word loss | St 76.7% and non st 82.4% |
| [6]  | SVM, Naive Bayes, complement Naive Bayes | Students reviews at the University of Portsmouth | 1036 | Performance comprised when trained class in model | Highest accuracy at 94%, followed by CNB at 84% |
| [7]  | Lexicon based, machine-learning and hybrid approaches | Comments in Spanish | 1000 | Issue with long sentence analysis | The accuracy through the hybrid approach (83.27%) |
| [8]  | Lexicon in Thai, SVM, ID3 and Naive Bayes | Student reviews at Loi Raja Hat University | 175 | Only 10 attributes for positive and negative | 97% highest accuracy of SVM |
| [9]  | Lexicon with 167 positive and 108 negative keywords | Feedbacks by students, obtained from RateMyProfessors.com | 1148 | The small data set of students' comments was utilized to teacher evaluation | The student comment text corpus score should be 0.67 |
| [10] | Late Dirichlet allocation, SVM, Naive Bayes and maximum entropy. | Movie reviews dataset | 75 | Missing values and incomplete data | SVM 82.90%, Naive Bayes 81.50%, maximum entropy 81%, LDA 88.50% |

The presented methodology classifies the students’ sentiments as positive, neutral and negative. The model workflow is shown in Figure 1 and is analyzed below.

**A. Data Preprocessing**

The collected dataset is not well organized and in order to extract the meaning and information from the text we need strong data preprocessing techniques. There are several steps applied for the removal of spelling errors, grammatical mistakes, and URLs. The details are described below:
C. LSTM

The representation of sentence in a sequence form was conducted by using the LSTM network. The first layer used was the embedded layer that contains 32 length vectors to represent every word. The next layer is the LSTM which contains 100 memory units. The final layer is the classification stage, where the model used a dense layer as the output layer with a single neuron. The model used the activation function to give a value between 0 and 1 for the predictions of two classes. The model adopted log or entropy loss to execute and process the binary classification problem, and dropout ratio with LSTM to maintain the learning and convergence of the network.

D. Hyperparameters Testing

In the model, the single hidden layer has 300 nodes which are the dimensions of a word in a form of a vector. The outputs of neurons were shaped with the activation functions (adam, adagrad, nadam and adamax). They push the output results up and down in a nonlinear fashion depending on the magnitude. When the magnitude is high then signals disseminate, and take their part at shaping the final prediction of the network. With the use of the activation function, the overall demonstration of the LSTM model is highly complex and nonlinear, therefore the softmax, softplus, sigmoid and relu optimization functions are used for minimizing the error of the model. Besides, to avoid the risk of overfitting, the regularization or shrinking approach has been used by making coefficients zero (dropout used between 0.1 to 0.4). After testing various combinations, the dense layer has a sigmoid activation function deployed for binary sentiment classification. In the last layer, we implemented the softmax activation function for the multi-class SA problem.

IV. RESULTS AND DISCUSSION

The experiments were conducted on a course evaluation dataset containing 3000 students’ comments [23]. In the dataset each feedback record contains fields such as teacher’s id, course name, comment, label, and semester. The dataset is divided into three groups for training (70%), testing (20%), and validation (10%). The research was conducted on SA where labels were 0 for negative, 1 for positive, and 2 for neutral. The diverse and blend parameters were tested considering regularization, optimization, and activation, in order to achieve the highest accuracy of the model as shown in Figures 2 to 4. The performance of the SA model was greatly feasible and effective as compared to conventional models, and LSTM SA model does not require prior knowledge such as sentiment lexicon and syntactic parsing. Moreover, the LSTM network has a long term memory to the context of the comment, which makes up the cons of the traditional SA. In a similar manner, the model adopted parameters are regularization, optimization, learning rate, and decay. And all these play a large part in reducing overfitting. The model also integrates max pooling, dropout, and normalization approach to reduce overfitting. By reducing dimensionally, max pooling performs best at a size of 2. Dropout layers were assessed at different locations in the network and they were found to be most helpful after max pooling and before normalization. The model implemented the cross-entropy loss function which basically computes the error between the true label and the predicted label. Figure 2 shows the validation accuracy of the model with a dropout value 0.2, “adam” optimizer, and softmax, softplus, sigmoid, and relu activation functions. The results indicate that the accuracy of the model is outstanding with soft plus.

Figure 3 depicts the accuracy of the model with softmax activation, adam optimizer, and dropout values of 0.1, 0.2, 0.3, and 0.4. The results indicate that the accuracy of the model is outstanding with a dropout value of 0.1. In the same way Figure 4 compares the accuracy of the model with softmax activation and 0.2 dropout for adam, adagrad, nadam, and adamax optimization functions. The results indicate that the
accuracy of the model is better with a dropout value of 0.1. Table II shows the experimental results obtained by the model on the student feedback dataset. The study was tested on 80 different combinations of parameters with 64 batch size for 30 epochs. The activation functions used were adam, adagrad, nadam and adamax, and hard_sigmoid, the optimization function were softmax, softplus, sigmoid, and relu, and the dropout regularization ranged between 0.1 to 0.4. The accuracy of the model was found to be higher with softmax, adam, and dropout ratio between 0.1 and 0.2. It has been noted that when the dropout value was 0.1 the model exhibited 89% training and 96% validation accuracy. Similarly, when the dropout value was set at 0.2 the model accuracy improved to 90% training and 97% validation accuracy.

Table II. HYPERPARAMETER COMPARISON

| Activation | Optimization | Dropout | Validation accuracy | Training accuracy |
|------------|--------------|---------|---------------------|-------------------|
| Softmax    | Adam         | 0.1     | 0.89%               | 0.96%             |
| Softmax    | Adagrad      | 0.1     | 0.89%               | 0.92%             |
| Softmax    | Nadam        | 0.1     | 0.88%               | 0.93%             |
| Softmax    | Adamax       | 0.1     | 0.89%               | 0.94%             |
| Softplus   | Adam         | 0.1     | 0.86%               | 0.91%             |
| Softplus   | Adagrad      | 0.1     | 0.86%               | 0.92%             |
| Softplus   | Nadam        | 0.1     | 0.85%               | 0.90%             |
| Softplus   | Adamax       | 0.1     | 0.86%               | 0.91%             |
| Sigmoid    | Adam         | 0.1     | 0.89%               | 0.93%             |
| Sigmoid    | Adagrad      | 0.1     | 0.82%               | 0.87%             |
| Sigmoid    | Nadam        | 0.1     | 0.82%               | 0.87%             |
| Sigmoid    | Adamax       | 0.1     | 0.83%               | 0.89%             |
| Relu       | Adam         | 0.1     | 0.86%               | 0.93%             |
| Relu       | Adagrad      | 0.1     | 0.86%               | 0.91%             |
| Relu       | Nadam        | 0.1     | 0.87%               | 0.90%             |
| Relu       | Adamax       | 0.1     | 0.81%               | 0.91%             |
| hard_sigmoid | Adam    | 0.1   | 0.87%               | 0.89%             |
| hard_sigmoid | Adagrad | 0.1   | 0.86%               | 0.89%             |
| hard_sigmoid | Nadam   | 0.1   | 0.88%               | 0.92%             |
| hard_sigmoid | Adamax | 0.1   | 0.85%               | 0.91%             |
| Softmax    | Adam         | 0.2     | 0.90%               | 0.97%             |
| Softmax    | Adagrad      | 0.2     | 0.90%               | 0.95%             |
| Softmax    | Nadam        | 0.2     | 0.90%               | 0.96%             |
| Softmax    | Adamax       | 0.2     | 0.88%               | 0.95%             |
| hard_sigmoid | Adamax | 0.4   | 0.85%               | 0.91%             |

REFERENCES

[1] S. M. Kim and R. A. Calvo, “Sentiment analysis in student experiences of learning,” presented at the 3rd International Conference on Educational Data Mining, Pittsburgh, PA, USA, Jun. 2010, pp. 111–121.

[2] C. K. Leong, Y. H. Lee, and W. K. Mak, “Mining sentiments in SMS texts for teaching evaluation,” Expert Systems with Applications, vol. 39, no. 3, pp. 2584–2589, Feb. 2012, doi: 10.1016/j.eswa.2011.08.113.

[3] B. Jagtap and V. Dhotre, “SVM & HMM based hybrid approach of sentiment analysis for teacher feedback assessment,” International Journal of Emerging Trends & Technology in Computer Science, vol. 3, no. 3, pp. 229–232, 2014.

[4] J. Ogden and J. Lo, “How meaningful are data from Kim, S. M. & Calvo, R. A. (2010). Sentiment analysis in student experiences of learning,” in Proceedings of the 3rd International Conference on Educational Data Mining, Pittsburgh, PA, USA, 2012.

[5] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[6] A. Minanovic, H. Gabelica, and Z. Krstic, “Big data and sentiment analysis using KNIME: Online reviews vs. social media,” in 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, May 2014, pp. 1464–1468, doi: 10.1109/MIPRO.2014.6859797.
