IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.16, No.4, October 2022, pp. 423~434
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.78664

Sentiment Analysis Using Backpropagation Method to Recognize the Public Opinion

I Komang Arya Ganda Wiguna*[1], Putu Sugiartawan[2], I Gede Iwan Sudipa[3], I Putu Yudi Pratama[4]
[1,2,3,4] Program Studi Teknik Informatika, Institut Bisnis dan Teknologi Indonesia, Bali, Indonesia
e-mail: ***[1] kmaryagw@instiki.ac.id,** [2]putu.sugiartawan@instiki.ac.id,
[3]iwansudipa@instiki.ac.id, [4]opitopit33@gmail.com

Abstract

Dewasa ini pertumbuhan industri perhotelan di Indonesia mengalami pertumbuhan yang pesat setiap tahunnya, industri pariwisata yang menjadi bagian dalam industri perhotelan juga tidak luput dari pertumbuhan yang pesat. Penulis ingin memberikan solusi untuk menigkatkan kualitas layanan pelaku pariwisata dengan cara melakukan analisis sentimet pada platform-platform digital yang dimiliki pelaku bisnis pariwisata dan mengumpulkan sentimen yang negatif untuk menigkatkan kualitas layanan dari usaha yang dimiliki para pelaku bisnis pariwisata. Salah satu metode untuk melakukan sentiment analisis yang akurat adalah Backpropagation Neural Network. Berdasarkan dari hasil pengujian yang dilakukan pada jaringan neural network didapatkan akurasi terbaik saat menggunakan 1 hidden layer dengan hidden layer pertama 10 neurons, learning rate bernilai 0,000002 dimana didapatkan akurasi senilai 71,630%, Semakin banyak melakukan epoch tidak menjamin akan mendapatkan akurasi yang lebih baik. Berdasarkan dari hasil penelitian yang telah dilakukan, saran untuk peneliti selanjutnya adalah melakukan analisis pada metode pengolahan dataset ulasan sehingga mendapatkan dataset yang lebih bersih dan di harapkan bisa meningkatkan akurasi yang lebih baik.

Kata kunci— *Sentimen analisis, neural network, backpropagation*

Abstract

Improve the service quality of tourism actors by conducting sentiment analysis on digital platforms owned by tourism businesses and collecting negative sentiments to improve the quality of services from companies owned by tourism businesses. The growth of the hospitality industry in Indonesia is experiencing rapid growth every year. The tourism industry, part of the hospitality industry, also does not escape the influence of positive and negative sentiments. One method to perform accurate sentiment analysis is Backpropagation Neural Network. Based on neural network testing, the best accuracy when using a hidden layer with the first layer is ten neurons, which produces a learning rate of 0.000002 and an accuracy value of 71.630%; more epochs do not guarantee improved accuracy. Based on the results of the research that has been done, suggestions for further researchers are to analyze the review dataset processing method so that it gets a cleaner dataset and is expected to improve better accuracy.

Keywords— *Sentiment Analysis, Neural Network, Backpropagation*

Received October 26th,2022; Revised October 30th, 2022; Accepted October 31th, 2022

1. INTRODUCTION

Currently, the growth of the tourism industry in Indonesia is experiencing rapid growth every year. Data released by the Center for Data and Information of the Ministry of Tourism and Creative Economy and the Central Statistics Agency (BPS) states that until the end of 2018, the total number of foreign tourists visiting Indonesia was 15,810,305 people with foreign exchange earnings of 16.4 billion USD, up 25% from the previous year.

This rapid growth in the tourism industry in Indonesia also significantly impacts hoteliers because hotels are the primary choice of places to stay and rest for tourists. To maintain the experience experienced by visitors and are expected to make their next visit, hotel management must know soon if there are customers. Who has complaints while staying at the hotel so they can be handled directly to prevent losing customers? To be able to handle complaints submitted by customers immediately, an artificial intelligence program is needed that can detect negative sentiments on the digital platform owned by the company so that hoteliers can immediately follow up on complaints submitted by their customers so that no customers are reluctant to visit because of an incident [1], [2].

One method to create a program that can perform accurate sentiment analysis is Backpropagation Neural Network [3], [4]. Several studies have been produced regarding the use of backpropagation neural networks for research related to backpropagation neural networks, namely in research conducted regarding sentiment analysis on the Job Creation Law, the best accuracy when classification is 98% [5]. Another study on backpropagation neural networks was carried out, analyzing public figures' sentiment using backpropagation neural networks obtained an accuracy rate of 62.3% with five epochs and two hidden layer nodes [6].

2. METHODS

2.1 Sentiment Analysis

Sentiment analysis is a classification task that classifies a text into a positive or negative orientation. Sentiment analysis is a reasonably complex research, sentiment analysis is the process of using text analytics to obtain various data sources from the internet and social media platforms to determine the emotional tone or opinion of users on the platform [7], [8]. Sentiment analysis is also often referred to as opinion mining because it is often used to explore the emotions behind every user's words when using the internet or in conversations on social media [8], [9].

2. 1.1 Text Preprocessing

Text preprocessing is a method to reduce text complexity before doing sentiment analysis. Text Preprocessing aims to produce higher accuracy when conducting sentiment analysis [10], [11]. Term weighting is a method to perform feature extraction on text data by assigning a value/weight. The weight of the term represents the extent to which a term means the contents of the document. The greater the weight of a term, the more effective the term is in describing the document's contents [12]–[14].

One method of term weighting is Term Frequency – Inverse Document Frequency (TF-IDF). TF-IDF is a word weighting algorithm that multiplies two-term weighting concepts, namely the frequency of occurrence of terms in a document (term frequency) and the distribution of terms in document collections (inverse document frequency). Term Frequency is defined in Equation (1).

$$tf_{a.b} = \frac{\text{number of occurrences of term a in document b}}{\text{sum of all terms in document b}} \tag{1}$$

Inverse Document Frequency is defined in Equation (2)

$$idf_a = log\frac{\text{number of documents}}{\text{sum of all documents which have } \textit{term} \text{ a}} \tag{2}$$

Term Frequency - Inverse Document Frequency defined in Equation (3)

$$tfidf_{ab} = tf_{a.b} \times idf_a \tag{3}$$

2. 1.2 Backpropagation Neural Network

Backpropagation is a supervised machine-learning algorithm. Backpropagation performs two stages of calculation, namely forward propagation and backward propagation. Backpropagation works by updating the weights and biases continuously to produce a minimum error value. The smaller the error value, the higher the accuracy generated by the neural network [15]–[17]. The steps for calculating the Backpropagation algorithm are as follows:

1. Initialize the Neural Network parameters (input layer, hidden layer, output layer, and learning rate).
2. Initialize weights and biases with random numbers.
3. Each node in the input layer receives the input signal and is forwarded to each node in the hidden layer by applying Equation (4),

$$h_i = \sum_{i=1} x_i w_{ih} + b_{hi} \tag{4}$$

1. After getting the hi-value, apply the binary sigmoid activation function to each value with Equation (5),

$$y_i = h_i + \frac{1}{i + e^{-h_i}} \tag{5}$$

2. Each value in the hidden node is forwarded to each node in the output layer using Equation (4).
3. After getting the value at the output node, apply the binary sigmoid activation function to each value with Equation (5) to get the output value.
4. Calculate the derivative of the error value at yi using the binary cross entropy method with Equation (6),

$$ce = -\frac{1}{n}\left(\sum_{i=1}^{n}(t_i \times \log(y_i)) + ((1-t_i) \times \log(1-y_i))\right) \tag{6}$$

5. Calculate the derivative of the error value at each output (y_i) using Equation (7),

$$\frac{\partial E_i}{\partial y_i} = -1 \times \left(t_i \times \frac{1}{y_i}\right) + (1-t_i) + \left(\frac{1}{1-y_i}\right) \tag{7}$$

6. Calculate the derivative of each oi using the binary sigmoid derivative Equation with Equation (8),

$$\frac{\partial y_i}{\partial o_i} = f(o_i) \times (1-f(o_i)) \tag{8}$$

7. Calculate the change in weight v_{ho} and bias b_{oi} using Equation (9) for weight calculation and Equation (10) for bias calculation:

$$\delta v_{ho} = \frac{\partial E_i}{\partial o_i} = \frac{\partial E_i}{\partial y_i} \times \frac{\partial y_i}{\partial o_i} \times z_i \tag{9}$$

$$\delta b_{oi} = \frac{\partial E_i}{\partial o_i} = \frac{\partial E_i}{\partial y_i} \times \frac{\partial y_i}{\partial o_i} \tag{10}$$

8. Calculate the value of the new v_{ho} weight and bias b_{oi}
9. Calculate the derivative of each hidden node using the binary sigmoid derivative Equation with Equation (8).
10. Calculate the error value for each hidden node.
11. Calculate the change in weight and bias connecting the input layer with the hidden layer by multiplying the results obtained from steps 12 and 13 using Equation (9) for weight calculation and Equation (10) for bias calculation.

After all the weights have been updated, the algorithm will return to the forward propagation stage and stop at the maximum epoch value. Epoch is a condition where the entire training dataset has gone through the training process and is returned to the initial stage for the next round (one epoch is equal to one training dataset being trained on the neural network) [18]. The common condition as the epoch increases is the reduction in the error value from the training process until the error value has reached convergence.

2. 1.3 Confusion Matrix

Evaluation is carried out to test the results and classifications obtained by measuring the performance value of the already created. The test parameter used for evaluation is accuracy, whose calculation is based on the confusion matrix table. The confusion Matrix works by comparing the original class with the predicted class [5]. The form of the Confusion Matrix is depicted in Table 1.

Table 1 Confusion Matrix

Actual Value	Prediction	
	Positive	Negative
Positive	*True Positive (TP)*	*False Negative (FN)*
Negative	*False Positive (FP)*	*True Negative (TN)*

Accuracy is the ratio of correct predictions (TP+TN) to the overall data.

3. RESULTS AND DISCUSSION

This section will explain text preprocessing, model training, model testing, and the final results.

3.1 Data collection technique

Training the neural network in this study using 515,739 rows of data consisting of seventeen columns, while the value of each column consists of variables;

[Hotel_Address, Additional_Number_of_Scoring, Review_Date, Average_Score, Hotel_Name, Reviewer_Nationality, Negative_Review,Review_Total_Negative_Word_Counts, Total_Number_of_Reviews, Positive_Review, Review_Total_Positive_Word_Counts, Tota l_Number_of_Reviews Reviewer_Has Given, Reviewer Score, Tags, days_since_review, lat, lng].

The aim is to be used as training data to identify positive or negative sentiments in a review. The data used is hotel review data in English, obtained from the website www.kaggle.com. Data training must then convert the data obtained from .csv into a data frame form. This Neural Network Backpropagation model uses a flowchart diagram to illustrate how the model will be designed to carry out the data analysis process. The model will be trained first using raw data from a Comma Separated Values (.csv) file with a predetermined sentiment. An overview of model-making can be seen in Figure 1 below.

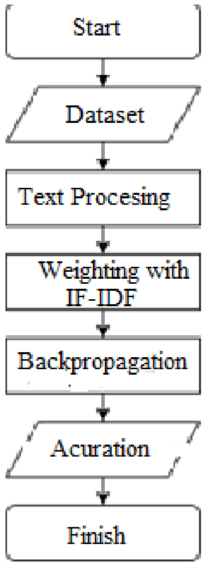

Figure 1 Overview of Modeling

The dataset used in this study is English hotel review data. The data used is Comma Separated Values (.csv) which consists of 515,739 positive and negative review data that have not been normalized; the data will be divided into 80% as training data and the remaining 20% as testing data.

3.2 Text Preprocessing

Stages of doing. Text preprocessing aims to reduce word complexity and increase model training accuracy. An overview of Text preprocessing can be seen in Figure 2, and the results of Text preprocessing can be seen in Table 2 below.

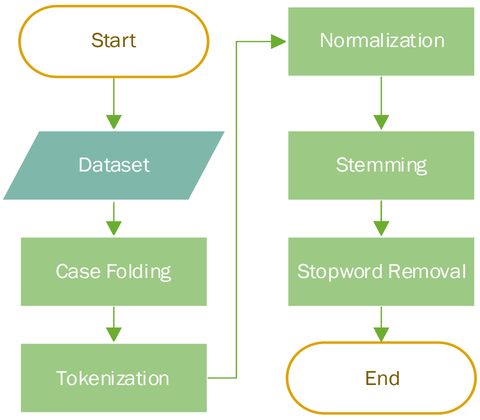

Figure 2 Text Preprocessing Architecture

Table 2 Dataset table after text preprocessing

Raw_Review	Review
Only the park outside of the hotel was beauti…	[park, outsid, hotel, beauti]
I am so angry that i made this post available….	[angri, made, post, avail, via, possibl, site, …]
No real complaints the hotel was great great…	[real, complaint, hotel, great, great, locat, …]
Location was good and staff were ok It is cut…	[locat, good, staff, ok, cute, hotel, breakfast, ….]
Rooms are nice but for elderly a bit difficul…	[room, nice, elderli, bit, difficult, room, tw, …]

3.2 TF-IDF Weighting

Sentences that have passed the Text Preprocessing stage will be separated into words or tokens. Then the TF-IDF weighting is carried out as many as the selected and processed vocabulary so that each vocabulary has its own weight.

Table 3 Dataset after weighting

Review	Review_Tokenize
[park, outsid, hotel, beauti]	[0.9321194954489584, 1.0611754323328455, 0.422]
[angri, made, post, avail, via, possibl, site, …]	[0.08917177914660462, 0.08568871610029234, 0.0]
[real, complaint, hotel, great, great, locat, …]	[0.13682031779472786, 0.190730913523471, 0.128]
[locat, good, staff, ok, cute, hotel, breakfast, …]	[0.15910871043526187, 0.10750167826728615, 0.3, …]
[room, nice, elderli, bit, difficult, room, tw...]	[0.761461127222173, 0.3010046991484013, 0.8981...]

At this stage, all the words in the dataset will be weighted, utilizing each word represented by a numerical value for the needs of the model training process. An overview of the dataset after the weighting process can be seen in Table 3.

3.2 Neural Network Backpropagation

After the weighting is done, then the data is entered into the Backpropagation algorithm for training on the data. This algorithm is used in the process of classifying positive and negative sentiments. Neural Network Backpropagation has two parts, Propagation and Weigh update, where propagation consists of forward and backward. With these two propagations, pattern recognition will obtain optimal results with a tiny epoch. After each word is weighted, the process of equalizing the number of words in each sentence is carried out. A process is carried out to find the average number of words contained in one sentence and find the number of 110 words per sentence.

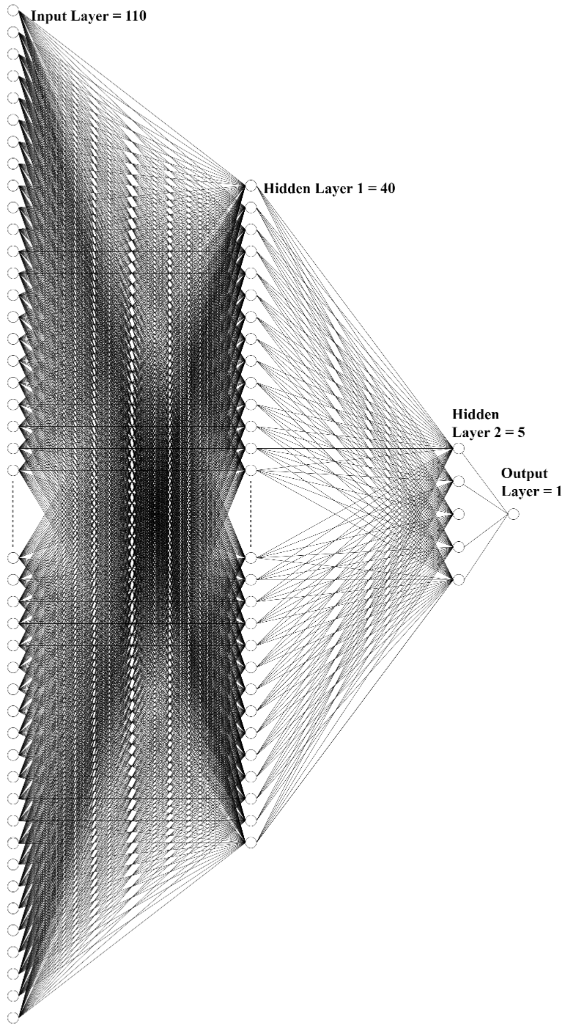

Figure 3 Architecture backpropagation

At this stage, the dataset will be divided into two parts where 80% of the dataset will be used as training data, and the remaining 20% will be used as testing data. Training data consists of 50% positive sentiment and 50% negative sentiment. After the weighting is done, then the data is entered into the Neural Network for training on the data. Neural Network Backpropagation is used to classify positive and negative sentiments. The following are details of the architectural design of the network used.

1. The first layer created is the Input Layer with a length of 110 neurons
2. The second layer, a hidden layer with 40 neurons
3. The third layer, a hidden layer with five neurons
4. The last layer is output with one neuron
5. Learning rate 0.000001

3.2.1 Testing the Number of Neurons on Hidden Layer 1

Testing the Number of Neurons on Hidden layer 1, The number of neurons in each layer affects the accuracy value when training the dataset. Initial testing was carried out using one hidden layer with the number of neurons tested as many as 10 to 100. The selected model is the model that has the highest accuracy value. Then the model chosen will be used in the classification trial using two hidden layers. The test results of hidden layer one neurons can be seen in Table 4.

Table 4 The results of testing the value of neurons on the curation of two

Neuron	Epoch	TP	TN	FP	FN	curation
10	20	13052	75651	1650	64249	57,375%
	40	33251	71059	6242	44050	67,470%
	60	46674	63831	13470	30627	71,477%
	80	51110	59209	18092	26191	71,357%
	100	52451	57662	19639	24850	71,224%
15	20	14107	75426	1875	63194	57,912%
	40	32491	70993	6308	44810	66,936%
	60	44386	64897	12404	32915	70,687%
	80	49554	60657	16644	27747	71,287%
	100	51424	58760	18541	25877	71,269%
…	…	…	…	…	…	…
30	20	14572	75374	1927	62729	58,179%
	40	31125	70832	6469	46176	65,948%
	60	41041	66596	10705	36260	69,622%
	80	46679	62783	14518	30622	70,802%
	100	49568	60433	16868	27733	71,151%

Table 4 shows that the best accuracy is 71.477% in the number of neurons from 10 to 60 epochs, with activation function relu in the hidden layer and sigmoid for the output layer. Accuracy changes from the 1st epoch to the 100th epoch on neuron.

3.2.2 Testing the Number of Neurons on Hidden Layer 2

In this test, ten neurons in hidden layer 1 are used. This is based on the highest accuracy produced in the previous test. This number of neurons will be used as the number of neurons in the first hidden layer when testing using 2 hidden layers. The results of the second test can be seen in Table 5.

The highest accuracy results were obtained by 54.067% in 10 hidden layer 1 neurons and 10 neurons in hidden layer 2. These results are still below the accuracy of using only 1 hidden layer.

Table 5 The results of testing the value of hidden layer 2 neurons on accuracy

Neuron		Epoch	Tp	Tn	Fp	Fn	Akurasi Relu+Sigmoid+
1	2						
	2	20	0	77301	0	77301	50,000%
		40	0	77301	0	77301	50,000%
		60	0	77301	0	77301	50,000%
		80	516	77206	95	76785	50,272%
		100	6625	76536	765	70676	53,790%
	3	20	0	77301	0	77301	50,000%
		40	0	77301	0	77301	50,000%
		60	0	77301	0	77301	50,000%
		80	516	77206	95	76785	50,272%
		100	6740	76529	772	70561	53,860%
		…	…	…	…	…	…
	10	20	0	77301	0	77301	50,000%
		40	0	77301	0	77301	50,000%
		60	0	77301	0	77301	50,000%
		80	539	77202	99	76762	50,285%
		100	7090	76498	803	70211	54,067%

3.2.3 Testing Total Learning Rate

The third test can be seen in Table 6 where Dataset will test the architecture with the best precision by making changes to the learning rate. From the test results in Table X below, the best accuracy results are obtained when the learning rate is 0.000002 with an accuracy of 71.624%.

Table 6 The results of testing the value of learning rate on accuracy

Learning-Rate	Epoch	Tp	Tn	Fp	Fn	Akurasi
0,000001	1	77301	0	77301	0	50,000%
	10	0	77301	0	77301	50,000%
	20	13052	75651	1650	64249	57,375%
	…	…	…	…	…	…
	100	52451	57662	19639	24850	71,224%
0,000002	1	0	77301	0	77301	50,000%
	10	12975	75760	1541	64326	57,396%
	20	34153	70948	6353	43148	67,982%
	30	47272	63460	13841	30029	71,624%
	…	…	…	…	…	…
	100	60195	41823	35478	17106	65,988%
…	…	…	…	…	…	…
1.8e-06	1	0	77301	0	77301	50,000%
	10	10715	76072	1229	66586	56,136%
	20	30212	72250	5051	47089	66,275%
	…	…	…	…	…	…
	100	53451	56378	20923	23850	71,040%

3.2.4 Testing the Number of input layers

The learning rate value has found the highest accuracy of 71.624% on ten hidden layers of one neuron. With a learning rate of 0.000002, in the fourth test, it can be seen in Table 5, where the architecture with the best accuracy will be re-tested its accuracy by making changes to the number of input layers. From the test results, the best accuracy results obtained based on changes in the number of input layers are 71.630% in the input layer 200 or 150 neurons and ten hidden layers, one neuron and with a learning rate of 0.000002.

Table 7 The results of testing the number of input layers

Input-Layer	Epoch	Tp	Tn	Fp	Fn	Akurasi
209	1	0	77301	0	77301	50,000%
	10	12980	75759	1542	64321	57,398%
	20	34161	70945	6356	43140	67,985%
	30	47268	63474	13827	30033	71,630%
	…	…	…	…	…	…
	100	60596	41048	36253	16705	65,746%
200	1	0	77301	0	77301	50,000%
	10	12980	75759	1542	64321	57,398%
	20	34161	70945	6356	43140	67,985%
	…	…	…	…	…	…
	100	60596	41048	36253	16705	65,746%
75	1	0	77301	0	77301	50,000%
	10	12933	75761	1540	64368	57,369%
	20	34118	70849	6452	43183	67,895%
	…	…	…	…	…	…
	100	58507	44801	32500	18794	66,822%

Final Result Structure, showing the final structure of the neural network model with the best accuracy obtained at 71.630%. The input layer consists of 200 nodes with activation, the hidden layer consists of 10 nodes with sigmoid activation, and the output layer consists of 1 node with a learning rate of 0.000002. The final architecture is shown in figure 4.

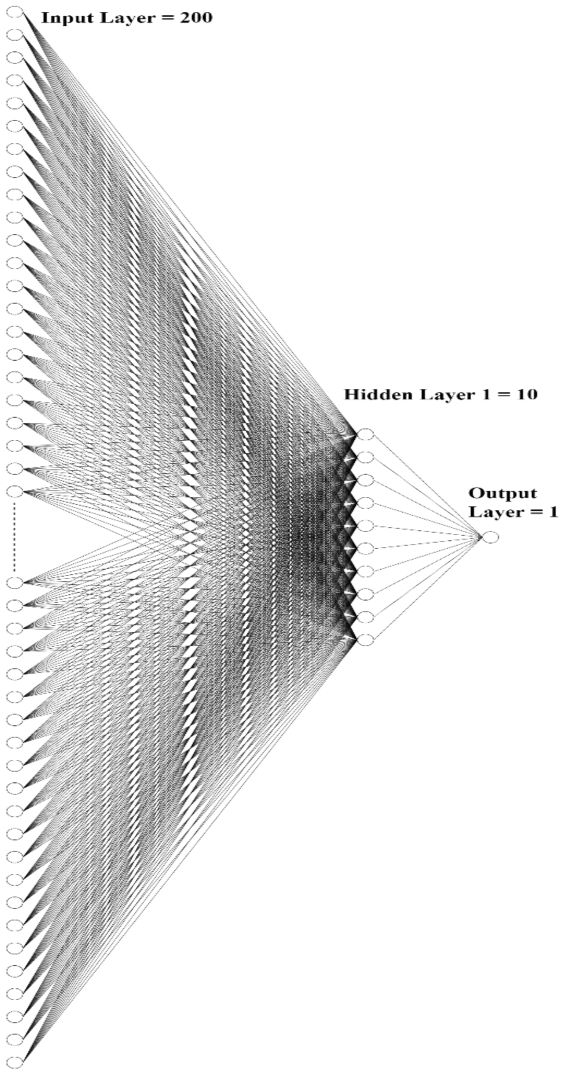

Figure 4 Final Overview of Neural Network Structure

4. CONCLUSIONS

Based on the results of tests that have been carried out on neural networks using several different architectures ranging from the number of hidden layers, the number of neurons, the number of epochs, and the learning rate value, there are several conclusions, including with the existing dataset the best accuracy is obtained when using one hidden layer. With the first ten neurons' hidden layer, the learning rate is 0.000002, where the accuracy is 71.630%. More epochs than guaranteed will get better accuracy. The effect of iteration duration on the tested momentum is not directly proportional to the value of the momentum. This backpropagation method needs to be compared with other word weighting methods, such as Word2Vec, BM25, or Latent Semantic Indexing (LSI), to determine which method can provide better accuracy.

REFERENCES

[1] J. Ye *et al.*, "Jou," *Knowledge-Based Syst.*, p. 110021, 2022.

[2] I. Almalis, E. Kouloumpris, and I. Vlahavas, "Sector-level sentiment analysis with deep learning," *Knowledge-Based Syst.*, vol. 258, p. 109954, 2022.

[3] P. Tao, J. Cheng, and L. Chen, "Brain-inspired chaotic backpropagation for MLP," *Neural Networks*, vol. 155, pp. 1–13, 2022.

[4] G. Shen, D. Zhao, and Y. Zeng, "Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks," *Patterns*, vol. 3, no. 6, p. 100522, 2022.

[5] A. Glushchenko, V. Petrov, and K. Lastochkin, "Backpropagation method modification using Taylor series to improve accuracy of offline neural network training," *Procedia Comput. Sci.*, vol. 186, pp. 202–209, 2021.

[6] P. Shang, L. Yang, Y. Yao, L. (Carol) Tong, S. Yang, and X. Mi, "Integrated optimization model for hierarchical service network design and passenger assignment in an urban rail transit network: A Lagrangian duality reformulation and an iterative layered optimization framework based on forward-passing and backpropagatio," *Transp. Res. Part C Emerg. Technol.*, vol. 144, no. October 2021, p. 103877, 2022.

[7] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, "Multimodal Sentiment Analysis: A Systematic review of History, Datasets, Multimodal Fusion Methods, Applications, Challenges and Future Directions," *Inf. Fusion*, 2022.

[8] N. Leelawat *et al.*, "Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning," *Heliyon*, vol. 8, no. 10, p. e10894, 2022.

[9] Y. Bian, R. Ye, J. Zhang, and X. Yan, "Customer preference identification from hotel online reviews: A neural network based fine-grained sentiment analysis," *Comput. Ind. Eng.*, vol. 172, no. PA, p. 108648, 2022.

[10] R. Chiong, G. S. Budhi, S. Dhakal, and F. Chiong, "A textual-based featuring approach for depression detection using machine learning classifiers and social media texts," *Comput. Biol. Med.*, vol. 135, no. February, p. 104499, 2021.

[11] S. Demir and B. Topcu, "Graph-based Turkish text normalization and its impact on noisy text processing," *Eng. Sci. Technol. an Int. J.*, no. xxxx, p. 101192, 2022.

[12] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, "Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms," *Array*, vol. 13, p. 100123, 2022.

[13] S. Tehseen, F. U. Ahmed, Z. H. Qureshi, M. J. Uddin, and T. Ramayah, "Entrepreneurial competencies and SMEs' growth: the mediating role of network competence," *Asia-Pacific J. Bus. Adm.*, vol. 11, no. 1, pp. 2–29, 2019.

[14] A. M. Saeed, S. R. Hussein, C. M. Ali, and T. A. Rashid, "Medical dataset classification for Kurdish short text over social media," *Data Br.*, vol. 42, p. 108089, 2022.

[15] Z. Wu, D. Tang, Y. Jiang, Y. Lu, and Y. Qiao, "Learned modified perturbation backpropagation for fiber nonlinear equalization in high-symbol-rate transmission systems," *Opt. Commun.*, vol. 521, no. June, p. 128612, 2022.

[16] S. Mouloodi, H. Rahmanpanah, S. Gohari, C. Burvill, and H. M. S. Davies, "Feedforward backpropagation artificial neural networks for predicting mechanical responses in complex nonlinear structures: A study on a long bone," *J. Mech. Behav. Biomed. Mater.*, vol. 128, no. December 2021, p. 105079, 2022.

[17] C. Lubis, E. Sutedjo, and B. Setiadi, "Prediksi Harga Saham Dengan Menggunakan," *Network*, vol. 2005, no. Snati, pp. 17–19, 2005.

[18] I. G. Made, N. Desnanjaya, and P. Sugiartawan, "Controlling and Monitoring of Temperature and Humidity of Oyster Mushrooms in Tropical Climates," no. x.