Convolutional neural network-based model for web-based text classification

Satyabrata Aich, Sabyasachi Chakraborty, Hee-Cheol Kim
Department of Computer Engineering/Institute of Digital Anti-Aging Healthcare, Inje University, Republic of Korea

Article Info

Article history:

Received Apr 18, 2019
Revised Jul 8, 2019
Accepted Jul 17, 2019

Keywords:

Convolutional neural networks
Deep learning
Text classfication
Text mining
Web-based text classiifcation

ABSTRACT

There is an increasing amount of text data available on the web with multiple topical granularities; this necessitates proper categorization/classification of text to facilitate obtaining useful information as per the needs of users. Some traditional approaches such as bag-of-words and bag-of-ngrams models provide good results for text classification. However, texts available on the web in the current state contain high event-related granularity on different topics at different levels, which may adversely affect the accuracy of traditional approaches. With the invention of deep learning models, which already have the capability of providing good accuracy in the field of image processing and speech recognition, the problems inherent in the traditional text classification model can be overcome. Currently, there are several deep learning models such as a convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long-short term memory that are widely used for various text-related tasks; however, among them, the CNN model is popular because it is simple to use and has high accuracy for text classification. In this study, classification of random texts on the web into categories is attempted using a CNN-based model by changing the hyperparameters and sequence of text vectors. We attempt to tune every hyperparameter that is unique for the classification task along with the sequences of word vectors to obtain the desired accuracy; the accuracy is found to be in the range of 85–92%. This model can be considered as a reliable model and applied to solve real-world problem or extract useful information for various text mining applications.

Corresponding Author:

Hee-Cheol Kim,
Departement of Computer Engineering/Institute of Digital Anti-Aging Healthcare,
Inje University,
197, Inje-ro, Gimhae-si, Gyeongsangnam-do, Republic of Korea 50834.
Email: heeki@inje.ac.kr

1. INTRODUCTION

Categorization or classification of text is considered to be one of the important topics in the field of natural language processing (NLP); it is also an essential tool in diverse fields such as filtering information, categorization of topics, searching the web, and sentimental analysis [1]. One of the simple ways of explaining text classification is described as follows: Given a group of documents containing a group of classes, a function is defined that will assign a value to the group of classes for each document [2]. Text classification or text mining approaches are used to extract important information from a large amount of text data in a short time [3]. From the beginning, text classification has been considered to be a complicated problem because text data are mostly unstructured and contain a lot of text vectors. Previous approaches available are suitable for a small amount of text with less complexity; however, these approaches are not suitable for a large amount of text because this reduces their accuracy. For this purpose, deep learning models are popular because they provide good accuracy when used for a large amount of data and these models have already shown their potential in the field of speech recognition and computer vision [4, 5]. The procedure that most deep learning approaches follow is as follows: First, input sentences are represented as a sequence of words. A term called “one-hot vector” represents each word in that model. Then, a weight matrix is multiplied by the sequence of words and projected to a vector space that is continuous in nature to form a dense vector that contains a sequence of real values. This sequence of words will be considered as input to the deep neural network in which multiple layers predict the desired output. Based on the tuning of suitable hyperparameters and the sequence of word vectors, maximum accuracy can be achieved in the training set [6-8]. Although different deep learning models are available such as long-short-term memory (LSTM), recurrent neural networks (RNNs), and convolutional neural networks (CNNs), we use a CNN because of its simplicity and also because it provides high accuracy for text classification. Therefore, in this study, a CNN model is developed with the best possible tuning of hyperparameters and sequencing of word vectors to improve the classification accuracy of texts into different categories.

The paper is organized as follows: Section 2 provides background and related work. Section 3 describes the methodology. In Section 4, the evaluation of the model is discussed. Section 5 provides conclusion and future work.

2. BACKGROUND

Kim [9] proposed a method to perform classification of a sentence using a CNN by using one convolution layer with the very little tuning of hyperparameters; four different models such as CNN-rand, CNN-static, CNN-non-static, and CNN-multichannel were used and an accuracy ranging from 81.5–89.6% was achieved. The author concluded that prior training of unsupervised word vectors is one of the important aspects while performing NLP-related tasks using deep learning. Zhang *et al.* [10] proposed an empirical study that uses character-level CNN for large-scale datasets; they found that a CNN that uses character-level features as input is effective when subjected to datasets of a certain size, curated text, and choice of alphabets. The result would be different if there is a change in dataset size, curated texts, and choice of alphabets. Johnson and Zhang [11] proposed a method to verify the effect of the word order on text classification accuracy; here, instead of applying a CNN to a low-dimensional word vector, they applied the method to high-dimensional text data to achieve better model accuracy. They used a parallel CNN, in which more than two convolution layers are used in parallel for learning more embedding texts to improve model accuracy; they found this method to be effective. Hughes *et al.* [12] proposed an approach based on the semantic classification performed at the sentence level. They used this method on medical texts and found that deep CNNs facilitate in analyzing the semantics of sentences by generating more optimal features, which indirectly improves the accuracy. They found that this method outperformed other approaches used for tasks related to NLP.

Rios and Kavuluru [13] proposed an approach to perform text classification of biomedical articles by assigning a medical subject heading (MeSH) term to the articles. They found an improvement of approximately 3% when using MeSH terms for classification tasks compared to previous results on public datasets. They mentioned that this method has a strong potential for classification of texts related to biomedical articles. Zhang *et al.* [14] proposed a novel method to perform sentimental analysis on text data by using a CNN and cross-modality consistent regression (CCR) and transfer learning. They used three types of embeddings such as lexicon embedding, semantic embedding, and sentiment embedding to encode the texts. To improve the performance, each CNN model contained one of the embeddings as well as CCR and transfer learning was performed. It has been found that all CNN models perform better compared to the existing models. Wang and Kim [15] proposed an improved CNN model for topic and sentiment classification on four benchmark datasets and found that the performance of the new model outperformed all the previous models.

Moriya and Shibata[16] proposed a CNN technique with deep layers at character level and then used transfer learning method for improvement in the classification accuracy.Nii *et al.*, [17] proposed a framework that used word vector representation of text and CNN for text classification.It was found that the performance of the proposed method is better than the previous methods. Lidong and Hui [18] proposed a technique named as multi mixed CNN for classification of text sentiments.It was found that this method is more effective compared to support vector machine, Naïve Bayesian and other classical methods. Kowsari *et al.*, [19] mentioned a good review about all the classification method and its advantages and disadvantages. From of the abovementioned past work, it can be seen that CNNs have enough potential for text classification-related tasks while achieving high accuracy.

3. CONVOLUTIONAL NEURAL NETWORKS

This section gives an overview of CNNs and their architecture for application to text classification. A typical CNN architecture for recognizing characters is shown in Figure 1 [20]. It was mainly invented for computer vision and nowadays almost every vision system uses CNNs [9].

A CNN is a neural network-based architecture, which basically consists of multiple stages; the network is trainable for performing text classification-related tasks. The stages of a CNN are as follows [21-23]:

a. Convolutional layers: These are some of the important layers of a CNN. These layers contain a number of kernel matrices. In these layers, convolution is usually performed by the kernel matrices on the input and an output as a biased value-added feature matrix is generated. The weight of kernel and biases is learned by using learning procedures because the connection weights are shared among neurons.

b. Pooling layers: These layers are fundamental elements of a CNN. The main objective of these layers is to carry out dimensionality reduction of the input, which reduces the number of randomly generated variables so that the data analytic process is faster and simpler. The subsampling of the convolution layer output is performed by the pooling layer by combining neighboring elements. The max-pooling function is the most commonly used pooling function that usually takes the maximum value among the local neighborhoods.

c. Embedding layer: This is one of the special elements of a CNN to perform text classification-related tasks. The objective of this layer is to convert input text documents into a proper format that is suitable for the CNN. In this layer, each word of the input text document is converted into a dense vector of a fixed size.

d. Fully connected layer: This is a hidden layer of a feed-forward neural network (FNN). This layer can be explicated as a unique convolution layer that contains a kernel matrix of size 1x1. This type of layer is a member of the group that contains the weights of a trainable layer. This layer is mostly used in the last stage of a CNN.

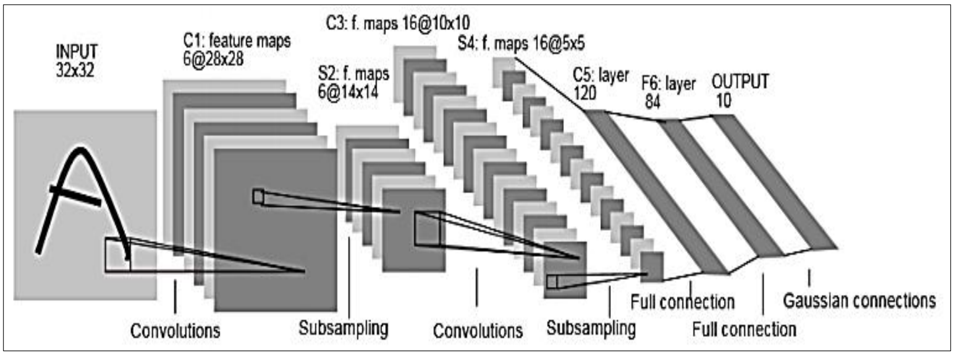


Figure 1. Architecture of a convolutional neural network

A backpropagation algorithm basically uses supervised learning and continuous-valued function. The numerical weights on each input are assigned based on historical data prior to the training process. In the training process, the optimal weight is finalized by reducing the mean square error E_m .The formula for finding the mean square error E_m is as follows [23].

$$E_m = \frac{1}{p_1 * n_o} \sum_{P=1}^{P=P_1} * \sum_{s=1}^{s=n_o} e_s{}^2 \, (p)$$

Where n_o is the number of neurons of the output layer and $e_s^2(p)$ is the error of the sth output neuron for the pth pattern of the training set.

To minimize the error function E_m mini-batch stochastic gradient descent (m-SGD) is widely used [24]. In m-SGD, the model coefficients and model error estimations are performed by dividing the training sets into small number of batches. This type of algorithm has the advantages of both stochastic gradient descent algorithm as well as batch gradient descent algorithm, i.e. robustness from the stochastic gradient descent algorithm and efficiency the batch gradient descent algorithm. This algorithm is the most widely used algorithm in the field of deep learning [25].

4. METHODOLOGY

The flowchart of the methodology used in this study is shown in Figure 2.

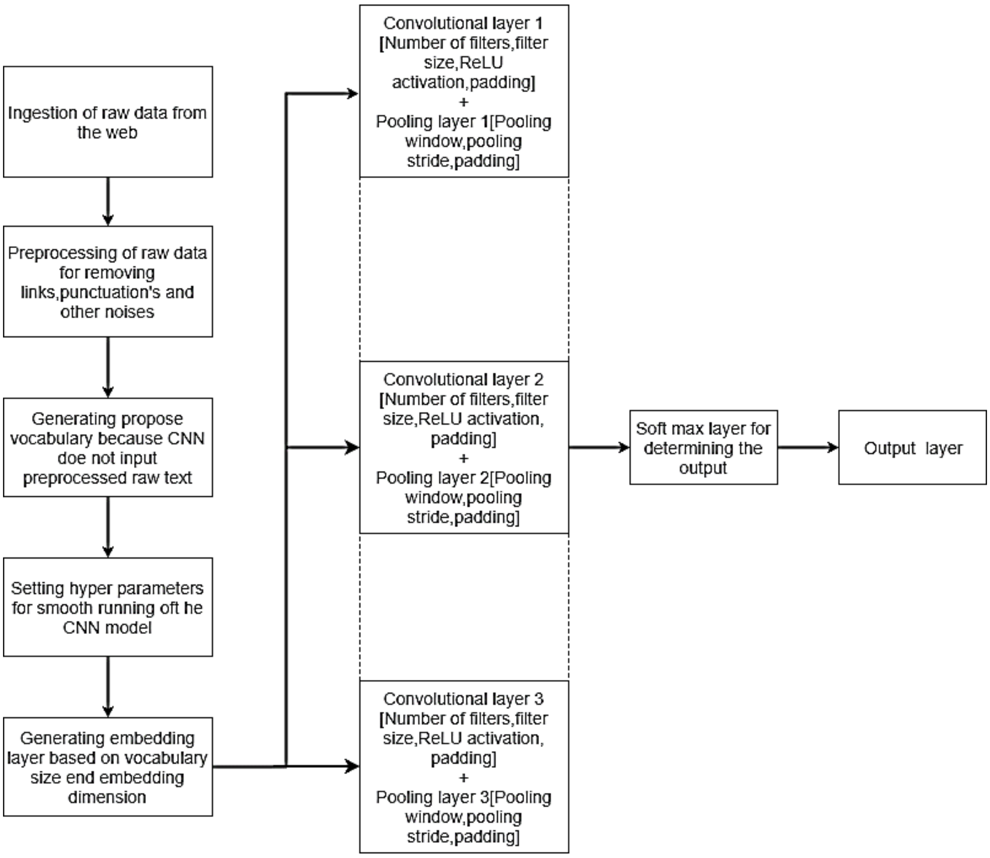

Ingestion of raw data from the web
Preprocessing of raw data for removing links,punctuation's and other noises
Generating propose vocabulary because CNN doe not input preprocessed raw text
Setting hyper parameters for smooth running oft he CNN model
Generating embedding layer based on vocabulary size end embedding dimension
Convolutional layer 1 [Number of filters,filter size,ReLU activation,padding] + Pooling layer 1[Pooling window,pooling stride,padding]
Convolutional layer 2 [Number of filters,filter size,ReLU activation, padding] + Pooling layer 2[Pooling window,pooling stride,padding]
Convolutional layer 3 [Number of filters,filter size,ReLU activation, padding] + Pooling layer 3[Pooling window,pooling stride,padding]
Soft max layer for determining the output
Output layer

Figure 2. Flowchart of the text classification using CNN

The following steps can be considered as the key steps to perform text classification using CNNs.

a. Data aggregation and ingestion: The initial development of the model requires a huge amount of data to be used for reviving the feature vector to completely convolve through the text matrix and to self-optimize the scoring function that is the weights. The data was collected from an open source repository that contained blogs on four different topics, namely, healthcare, sports, movies, and finance. The data initially contained a lot of noise like advertisement data, images, and links. This noise was removed at the very initial stage to support proper ingestion of data into the model.At the production level, data ingestion is implemented by using the Beautiful Soup object, which accepts a hyperlink to any blog or any kind of webpage. After fetching the link, the Beautiful Soup object is processed with a link of the blog or textual data; initially, it captures complete textual data from the blog by negating all the excess noise such as images and links. After fetching data from the link, the object is processed in the initial stage such as by removal of punctuations and leftover noise; further, two-degree lemmatization is performed on the text to make it suitable to process using the classifier. The scope of the Soup object is such that every time the object runs, it retrains the model further to make it more sophisticated after properly testing using a predefined model.

b. Data preprocessing and generation of vocabulary: This is the initial and the most fundamental part that needs to be performed for the generation of any model of such a use case. Any kind of text that is fetched from the internet contains some noise, which is required to be removed from the input prior to its ingestion into the model. In a CNN, few aspects that always need to be ascertained before going forward with any approach for model development include the length of the document, the padding required for the input matrix, etc. Therefore, the primary aim is always to analyze the type of data. In the very first step, we remove all kinds of punctuations from the text data to develop a consistent vocabulary that can be used for the model. For vocabulary generation, few generic functions are created, which first form a proper vocabulary, because neural networks never take strings as input and require working with numerical inputs. Therefore, any kind of input should be converted to a one-hot input so that it perfectly relates to the response variable or the output variable. For the regularization of all the sentences or word vectors in a textual object, the vectors need to be combined with proper bindings. For maintaining proper binding for all word vectors, we perform padding of sentences such that it renders the length of all vectors to be of the same size. Therefore, we usually have a hyperparameter in our system, MAX_DOC_LENGTH, so that proper padding such as <PAD> (in our case) can be used to regularize all the word vectors into a similar segment in terms of length. Moreover, this padding helps in monitoring new words that the classifiers have not observed previously and convert to a proper segment so that the system maintains consistency.

c. CNN model: First, we generate a sparse matrix from the textual data on the basis of the split ratio of the number of sentences and the number of words. After creating a sparse matrix, we look it up in our vocabulary for matching, which generates the batch size. Our primary default batch size was 10 words, where the smaller sentences were padded with 0 with regards to the previous padding provided to them in the earlier step. Therefore, we conclude that the maximum length of the document ingested into the CNN model should be 45 words. In the embedding layer, which is generally the first layer of a CNN model, we map each vocabulary word to low-dimensional text vectors. On the basis of the embedding layer developed in the previous stage, a convolution layer is developed that takes an input from the embedding layer and passes the scalar products of the functions to the max-pooling layer. For the complete system, we developed two convolution layers followed by two max-pooling layers and fully connected layers. For the first convolution layer, the weight matrix or the filter had a dimension of the window size of each convolution frame and the embedding size; for the second convolution layer, the weight matrix had the dimension of the same window size of the second convolution layer and the size of weight matrix or filter. The sizes of the filter or weight matrix track a major use case in determining the performance of the CNN model as these are an initially randomized sequence of numbers, which are then optimized to a loss gradient to hold the minima. After convolution, the matrix is pooled over by a max-pooling layer, which downsamples the index to a maximum value. The fully connected layer at the end tends to accept an input from the final max-pooling layer to generate the score of each and every class for a batch of word vectors passed into the model. Finally, the output is provided by the output layer. The activation function that was used in the model was the ReLU function to generalize the path of the system for concurring results so that it reinforces itself to different ways for proper prediction.

5. EVALUATIONS

This section compares the performance of our model with some existing state-of-the-art methods. In this study, we used a CNN model using the vocabulary generated from the input text and tuning the hyperparameters; we found the maximum accuracy to be 92%. The hyperparameters used in our study are as follows: number of filters, maximum length of the document, embedding size, window size, pooling window, pooling strides, number of convolution layers, and number of pooling layers. In our case, we achieved maximum accuracy by assigning the values to the abovementioned hyperparameters as follows: number of filters = 10, maximum length of the document = 80, embedding size = 20, window size = 20, pooling window = 4, pooling stride =2, number of convolution layer = 2, and number of pooling layers = 2. With the abovementioned combinations, we achieved higher accuracy.

The method by Hughes *et al*. [12] used a CNN model with their word2vec approach for medical text classification and they achieved a maximum accuracy of 68%. The method by Kim [9] used CNN-rand model, in which words were initialized randomly and modified later in the training phase; CNN-static model, in which pre-trained vectors were static but other parameters were updated based on the performance; CNN-non-static model, where pertained vectors as well as other parameters were updated based on the performance; CNN-multichannel model, in which each channel has its own setting and is tuned separately. The accuracy of the abovementioned models ranges from 81.5–89.6 %. The classification performance is shown in Figure 3. From the Figure 3, it is clear that our proposed method achieves the highest accuracy. The slight margin of improvement, of approximately 3%, can also facilitate in achieving the desired classification objectives.

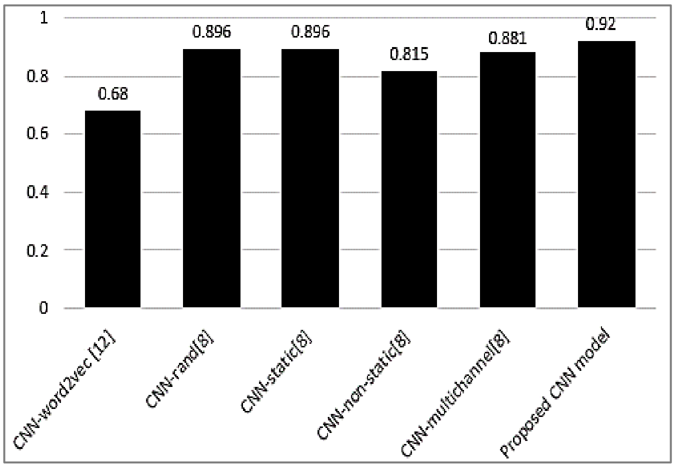


Figure 3. Classification performance of different CNN models

6. CONCLUSION

In this study, we proposed a novel CNN-based method to classify texts belonging to different categories collected from the web with higher accuracy compared to other CNN-based models. The proposed CNN model was built by considering different hyperparameters, which were tuned to optimize the results. We found accuracies ranging from 85–92% based on the hyperparameter tuning and shuffling of the sequence of the text vectors. In the future, we will implement our proposed model at a much larger scale with better fine-grained datasets. We hope that our model initiates further studies as well as helps researchers in the field to classify random texts from the web and extract useful information from it.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1D1A3B04032905). This paper is a revised and expanded version of a paper entitled “A Convolutional Neural Network Approach for Classification of Text based on the Texts Collected from the Web” presented at “International Conference on Future Information & Communication Engineering (ICFICE), Pattaya, Thailand, and 26 June-1 July 2018.

REFERENCES

[1] C. C. Aggarwal, and C. Zhai, “A survey of text classification algorithms,” *In mining text data*, Springer, Boston, MA, pp. 163-222, 2012

[2] M.R. Murty, *et al.*, “Text Document Classification based-on Least Square Support Vector Machines with Singular Value Decomposition," *International Journal of Computer Applications*, vol. 27, pp. 21-26, 2011.

[3] S. Aich, *et al.*, “A text mining approach to identify the relationship between gait-Parkinson's disease (PD) from PD based research articles," *In Inventive Computing and Informatics (ICICI), International Conference on, IEEE*, pp. 481-485, 2017.

[4] A. Krizhevsky, *et al.*, “ImageNet Classification with Deep Convolutional Neural Networks,” *In Proceedings of NIPS*, 2012.

[5] A. Graves, *et al.*, “Speech recognition with deep recurrent neural networks,” *In Proceedings of ICASSP*, 2013.

[6] Y. Kim, “Convolutional neural networks for sentence classification,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational linguistics*, 2014, pp. 1746-1751.

[7] Y. Xiao, and K.Cho, “Efficient character-level document classification by combining convolution and recurrent layers,” *arXiv preprint arXiv: 1602.00367*, 2016.

[8] A. Hassan, and A.Mahmood, “Efficient Deep Learning Model for Text Classification Based on Recurrent and Convolutional Layers,” *In Machine Learning and Applications (ICMLA), 16th IEEE International Conference on, IEEE*, 2017, pp. 1108-1113.

[9] K. Yoon, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv: 1408.5882*, 2014.

[10] X. Zhang, *et al.*, “Character-level convolutional networks for text classification,” *In Advances in neural information processing systems*, pp. 649-657, 2015.

[11] R. Johnson, and T. Zhang, “Effective use of word order for text categorization with convolutional neural networks,” *arXiv preprint arXiv: 1412.1058*, 2014.

[12] M. Hughes *et al.*, “Medical text classification using convolutional neural networks,” *Stud Health Technol Inform*, vol. 235, pp. 246-50, 2017.

[13] A. Rios, and R. Kavuluru, “Convolutional neural networks for biomedical text classification: application in indexing biomedical articles,” *In Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, ACM*, 2015, pp. 258-267.

[14] Z. Zhang, *et al.*, “Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression,” *Neurocomputing*, vol. 275, pp. 1407-1415, 2018.

[15] X. Wang, and H.C. Kim, “Text Categorization with Improved Deep Learning Methods,” *Journal of Information and Communication Convergence Engineering*, vol. 16, pp. 106-113, 2018.

[16] S. Moriya, and C.Shibata, “Transfer learning method for very deep CNN for text classification and methods for its evaluation,” *In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), IEEE*, vol. 2, pp. 153-158, 2018.

[17] M. Nii, *et al*., “Nursing-care text classification using word vector representation and convolutional neural networks,” *In 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), IEEE*, 2017, pp. 1-5.

[18] H. Lidong, and Z.Hui, “A new short text sentimental classification method based on multi-mixed convolutional neural network,” *In 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). IEEE*, 2018, pp. 93-99.

[19] K.Kowsari, *et al.*, “Text classification algorithms: A survey,” *Information*, vol. 10, pp. 150, 2019

[20] Y. LeCun, *et al.*, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.

[21] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[22] S.V. Georgakopoulos, *et al.*, “Convolutional Neural Networks for Toxic Comment Classification,” arXiv preprint arXiv: 1802.09957,2018.

[23] X. Pan, *et al.*, “A comparison of neural network backpropagation algorithms for electricity load forecasting,” *In Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on, IEEE*, pp. 22-27,2013.

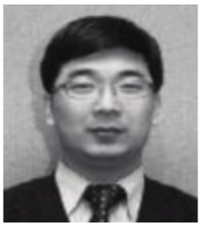
[24] L. Bottou, “Online learning and stochastic approximations,” On-*line learning in neural networks*, vol.17, pp. 142, 1998.

[25] Jason Brownlee, “A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size,” on *Deep Learning*, 2017, [Online], Available: https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/ [extracted on 14 August 2018].

BIOGRAPHIES OF AUTHORS

Satyabrata Aich, is working as a researcher in the field of computer engineering and digital healthcare. He has published many research papers in journals and conferences in the realms of machine learning, text mining, and supply chain management. His research interests are natural language processing, machine learning, supply chain management, text mining, and medical informatics.

Sabyasachi Chakraborty, is working as a master student at Inje University. He has worked in many real life project related to data mining, text mining. He has also published few papers related to data analytics and big data. His research interests are natural language processing, machine learning, big data, and text mining.



Hee-Cheol Kim, received his BSc at Department of Mathematics, MSc at Department of Computer Science in SoGang University in Korea, and PhD at Numerical Analysis and Computing Science, Stockholm University in Sweden in 2001. He is Professor at Department of Computer Engineering and Head of the Institute of. Digital Anti-aging Healthcare, Inje University in Korea. His research interests include machine learning, text mining, and medical informatics.