The Role of Activation Function in Neural NER for a Large Semantically Annotated Corpus

1st Muhammad Saad Amin
Dipartimento di Informatica
Universita di Torino
Torino, Italy
muhammadsaad.amin@unito.it

2nd Luca Anselma
Dipartimento di Informatica
Universita di Torino
Torino, Italy
luca.anselma@unito.it

3rd Alessandro Mazzei
Dipartimento di Informatica
Universita di Torino
Torino, Italy
alessandro.mazzei@unito.it

Abstract*—Information extraction is one of the core fundamentals of natural language processing. Different recurrent neural network-based models have been implemented to perform text classification tasks like named entity recognition (NER). To increase the performance of recurrent networks, different factors play a vital role in which activation functions are one of them. Yet, no studies have perfectly analyzed the effectiveness of activation function on Named Entity Recognition based classification task of textual data. In this paper, we have implemented a Bi-LSTM-based CRF model for Named Entity Recognition on the semantically annotated corpus i.e., GMB, and analyzed the impact of all non-linear activation functions on the performance of the Neural Network. Our analysis has stated that only Sigmoid, Exponential, SoftPlus, and SoftMax activation functions have performed efficiently in the NER task and achieved an average accuracy of 95.17%, 95.14%, 94.38%, and 94.76% respectively.

***Keywords*—activation functions, Groningen Meaning Bank (GMB), named entity recognition, recurrent neural networks**

I. Introduction

In the age of the modern era, traditional classification and recognition approaches have been overlapped by neural networks. This is due to the performance and efficient results of networks in the form of increased accuracy and reduced loss. Based on the type of data, different deep neural networks have been used to perfectly classify or recognize large image, textual, or speech data. Result analysis of deep neural networks is characterized by hyperparameter optimization including the impact of network optimizer and activation function used. For the task of tagging textual data for information extraction, recurrent neural networks (RNN) are often used. RNN based deep models work with the help of non-linear activation functions. The purpose of the non-linear activation function is to update input neurons to conceive data variation which helps model in understanding data structure efficiently.

In the history of neural networks, there is a list of activation functions that have been used to perform specified tasks. Among different flavours of activation functions, step function, sigmoid function, and tangent hyperbolic (tanh) function are considered as classical activation functions. The impact of these activation functions clarifies the learning path for the network [1]. During the training of the network, classical activation functions transform high valued gradients approximately close to zero which implies the overall network inputs and the stochastic gradient descent values are updated with a very small value [2]. Networks facing this issue cannot perform classification or recognition tasks because the network does not learn perfectly from input data due to not significant changes of weights. This problem is referred to as the vanishing gradient problem [3]. To deal with the vanishing gradient problem, updated variants of activation functions in the form of ReLU and LeakyReLU are designed [4-5]. These activation functions conceive positive output as constant value and do not update weights close to vanishing points.

In the modern era of research, designing and implementing activation functions with high accuracy, minimum loss and reduced training time is under deep consideration [6-8]. People working in the field of neural networks admit the effect of activation functions on the performance of the system. Indeed, considering RNN for text tagging, the impact of different activation functions on systematic accuracy and efficiency, at best of our knowledge, has not been extensively explored. To overcome this limitation, we have investigated the impact of activation functions on a modern high-performing neural architecture, that is Bi-LSTM based sequential RNN network, by implementing a text tagging task, referred to as Named Entity Recognition (NER).

NER is a task for tagging proper nouns and recognizing them as belonging to few specific entity categories, as *PERson, ORGanization, GEOgraphical, TIMe, ARTifact, EVEnt, NATural* etc. In particular, we performed the NER tagging experiments on the Groningen Meaning Bank (GMB) dataset [7], that is a large semantic dataset containing various layers of integrated semantic annotation. GMB contains word sense, named entity, syntactic and semantic annotations. Indeed, we decided to us GMB since, as long-term goal, we want to build a high performing modular neural architecture for complete semantic analysis. So, NER tagging is the first step toward this ambitious project.

In the experiments described in this paper, we have implemented 13 different activation functions i.e., *ReLU, Leaky ReLU, Tanh, Binary step function, Linear, SeLU, ELU, Sigmoid, Parametric ReLU, SoftMax, SoftSign, SoftPlus, Swish, and GeLU* separately on each layer of RNN for NER tagging and we analysed their impact in the form of accuracy and loss on the GMB dataset.

The remaining paper is structured as follows. Section 2 discusses related works. Section 3 introduces the RNN architecture. Section 4 introduces the experimental implementation conducted in this paper. Results are discussed in Section 5 and Section 6 is focused on the conclusion of this paper and highlights future directions as well.

II. Background and Related Works

Deep neural networks are the most commonly used models in today's era of modern technology. With the advancement of time, architectures and datasets are becoming more and more complex. To deal with these complex structures, a lot of different neural models have been

introduced that take hours and sometimes days of time to properly train themselves. To improve the performance of neural networks, activation functions play an important role [9]. Effect of activation functions is directly interlinked with hidden neurons that change the biases interleaving within layers [10-13]. Based on the nature of the experiment like image classification, text classification, video sequences distribution, only one activation function is used [14-16]. The choice of the right activation function is a major problem that totally depends on the nature and size of data [17].

Different activation functions have been designed and implemented to date and they can be categorized in three major types. The first type includes rigid activation functions. Linear, ReLU, Heaviside, and Logistic are considered rigid activation functions [18-20]. The second type of activation function is known as Radial activation functions and includes Gaussian, Multiquadric, Inverse multiquadric, and Polyharmonic splines. The third type of activation function is known as folding activation functions and includes SoftMax, softplus, sigmoid, and exponential activation functions. Folding activation functions perform aggregate operations like min, max, mean on input data and help in learning non-linearity.

In this work, we have analysed the effect of changing activation functions on the performance of the network. We have used 13 different types of activation functions and checked network performance by analysing accuracy, loss, validation accuracy, validation loss, and mean loss values. Our experiment of NER has performed very well for Sigmoid, SoftMax, SoftPlus, and Exponential activation functions. Detailed analysis of performance and comparison of results will be discussed in sections given below.

III. Experimental Structure of RNN

Recurrent Neural Network-based Bi-directional Long Short-Term Memory (Bi-LSTM) model having 4 layers embeddings is used for Named Entity Recognition (NER) task. The input sequence is fed into the embedding layer of the sequential model having 2523072 parameters and 64 embedding dimensions. The input length of the sequence is kept at 50 and RNN units are taken as 100 units. CRF layer is used to computer log-likelihood of the model during training. For this purpose, we have used the Viterbi decoding scheme during the prediction of data values. After that, the dense layer is deeply connected with previous layers resulting in 1919 resultant parameters having 19 different classified labels. Finally, the CRF layer is embedded and it is used to predict tagged sequences for final data evaluation. Complete model summary of implemented architecture is listed in Table I.

TABLE I. Implemented Sequential RNN Model

Layers	Output Shape	No. of Parameters
Embedding	(None, 50, 64)	2623072
Bi-directional	(None, 50, 200)	132000
Time distribution	(None, 50, 100)	20100
Dense	(None, 50, 19)	1919
CRF	Multiple	361

The graphical representation of deployed architecture is shown in Fig. 1. Input textual sequence is passed through word embedding phase where data is balanced with respect to embedding layer. In the embedding layer, each word of the input sequence is converted into a vector of fixed length, and the size of the length is self-defined. In a bi-directional layer, RNN is actually connecting two hidden layers of opposite directions to the same output layer.

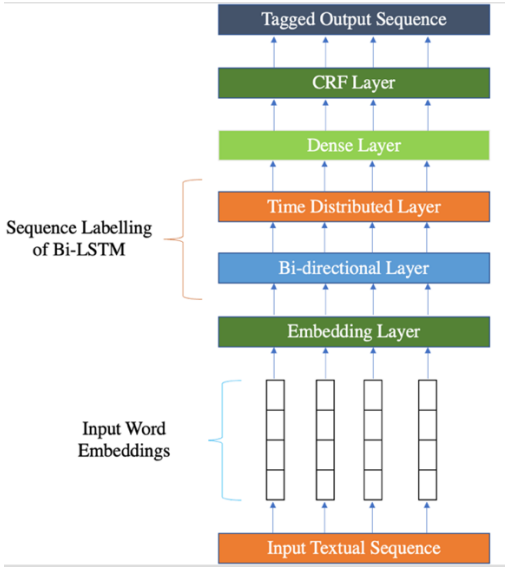

Fig. 1. Graphical representation of implemented architecture with respect to layer embeddings

Feedforward backpropagation mechanism is actually contributing to this layer. Purpose of using *backpropagation* is to deal with and minimize error rate of forward layers. If there is no backpropagation then there is no looping back of the processed information which may lead to miss the important features of the complex network. *Backpropagation* is a type of *gradient decent* technique that works in the reverse direction of feedforward gradients as shown in Fig. 2 given below. *Feedforward backpropagation* works on minimizing cost function by upgrading weights and biases values at each iteration of the network training.

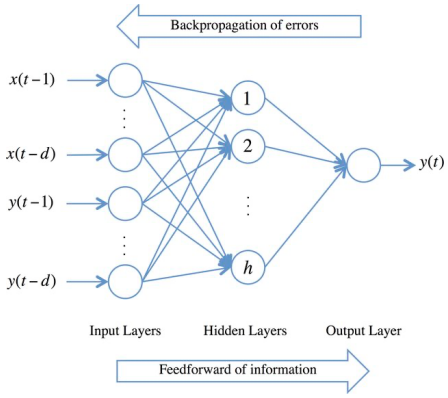

Fig. 2. Workflow of feedforward backpropagation algorithm

So, the past and future states of the recurrent model is working simultaneously in this step. Similarly, pre-processed time series data is handled in the time distribution layer. Instead of dealing with data as several inputs, the time distribution layer deals with one layer applied to each input sequence. This vector data is further processed towards the dense layer. Data dimensionality is altered and dealt with in

this layer. Output generated by dense layer is also a vector having lower dimension but this resultant data is achieved after performing certain operations like scaling, rotation, translation, etc. and finally, tagged data is obtained as output sequences.

IV. Activation Functions

Activation Function (AF) also known as *Transfer Function (TF)* is responsible for controlling and responding to the neuron inside the Neural Network. Depending on the type of input, an activation function helps in computing output values that are being fed into the neurol node. It operates by computing weighted sum of the input values thus adding non-linearity into the model and transferring it to the next hidden or output layer as shown in Fig. 3. The purpose of adding non-linearity into the neural model is to make model understand complexities. If non-linearity is not added into the network, then it is just a linear classifier not capable of dealing with computationally complex problems. Activation functions are categorized into binary, linear, and non-linear activation functions. All types of AFs are used subjectively. One activation function can perform very well in one case and others may not. As Named Entity Recognition is considered as one the complex tasks in Natural Language Processing (NLP), therefore we have focused more on non-linear types of activation functions.

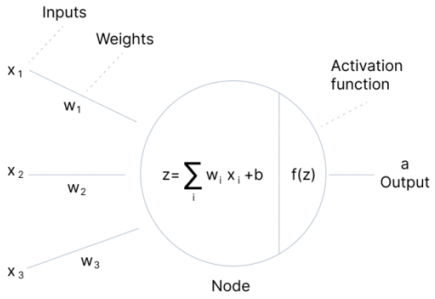

Fig. 3. Weighted sum of input and bias along with activation function inside the neural network

A. Binary Activation Function

Binary activation function act as a switch for the neuron. Based on the threshold value, it decides whether a neuron should be active or not. If input provided is higher than the threshold, neuron is activated otherwise not. Main issue of binary AF is that, it cannot deal with multi-class scenarios. And also, the gradient of values less than the threshold is always zero which is not good for Backpropagation. Mathematical equation of Binary AF is given below.

$$f(x) = \begin{cases} 0 \; for \; x < 0 \\ 1 \; for \; x \geq 0 \end{cases} \quad (1)$$

B. Linear Activation Function

Linearity of AF refers towards no activation i.e., just proportional to the input of NN. Here weighted sum of input is not computed rather it just let the input pass as it is. Linear AF is not considered helpful for backpropagation as the derivative of input is always constant. Secondly, all AF layers are considered as single layer due to linear relation between them. Mathematical equation of Linear AF is given below.

$$f(x) = x \quad (2)$$

C. Non-Linear Activation Function

Problem of backpropagation and updating weights as input to the model is solved with the help of non-linear AFs. With updating every neuron of NN, prediction of complex input sequences is made easy. This non-linearity of AF also supports multi-layered structure due to different weights of neurons on each layer.

For this experiment, we have totally focused on non-linear AFs because of our input data. Collectively we implemented NER task on 13 different activation functions to analyze the high-performance impact in this regard. For text classification task, there is no proper description of the choice of activation function. So, our focus is to recommend high performing AF for NER task. Among 13 different AF (mentioned in introduction), our experiment was successful only in 4 types of AF i.e., *Sigmoid, SoftMax, SoftPlus*, and *Exponential*. Other authors working in this task and focusing on *Hyperparameter Optimization* can choose among these only, despite of looking for all other types of AFs. Specifically, for *Neural-NER* task on *GMB dataset*, the best *Activation Function* is *Sigmoid*. Table II provides the information of mathematical equations of 4 above mentioned AFs used in this experiment.

TABLE II. Mathematical Equations of Non-Linear Activation Functions used in this Experiment

Activation Function	Mathematical Equation
Sigmoid	$f(x) = 1/1 + e^{-x}$
SoftMax	$\sigma(\vec{z})_i = e^{z_j} \Big/ \sum_{j=1}^{K} e^{z_j}$
SoftPlus	$f(x) = log_e(1 + e^x)$
Exponential	$f(x) = \begin{cases} \alpha(e^x - 1) \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$

V. Experimentation

For the implementation of the NER task on the GMB dataset, we have used Bi-LSTM based RNN model with the embedding of the Conditional Random Field (CRF) layer. Structure of input data is completely based on tagged textual data having 62010 sentences in total. Dataset is divided into training, development, and testing split of 80%, 10%, and 10% respectively. The NER layer of the complete GMB dataset, that we have used as training in our experiment, has 2677091 parameters. To train this large dataset we have used an NVIDIA-based GPU having CUDA accelerated libraries embedded in it. Our experiment is completely executed on GPU for fast and efficient computation of results. Since a focus of this experiment is analysing the impact of activation function on the performance of the network, we have trained and tested the RNN model 13 times each one with respect to a new activation function. Considering hyperparameters, one optimizer is taken into consideration at a time and the activation function is changed each time respectively. In this experiment, we have used Stochastic Gradient Decent (SGD) as an optimizer and trained RNN model for all activation functions. Our experimental finding states that only sigmoid, exponential, SoftMax and softplus activation functions help the RNN model in training perfectly. All other activation functions perform very badly thus resulting in very poor accuracy and high loss values. Performance of the network is analysed with respect to the training accuracy, validation accuracy, mean loss, validation loss, and testing accuracy of the network.

Statistical information of the model gives a clear understanding of deployed architecture. This information includes embedding dimensions of the neural network, units used in the network, batch size, model type, length of the input sequence, marginal dropout value, optimizer used, loss category, learning rate, and activation function. The hyperparameters used in the implementation of the current experiment are listed in Table III.

TABLE III. STATISTICAL INFORMATION OF IMPLEMENTED SEQUENTIAL RNN MODEL

Parameters	Values
Embedding Dimensions	64
RNN Units	100
Batch Size	128
Model	Sequential
Input Length	50
Dropout	0.2
Kernel Initializer	He_normal
Optimizer	SGD
Loss	Categorical Cross Entropy
Validation Split	0.2
Learning Rate	1e-3
Activation Functions	Sigmoid, SoftMax, SoftPlus, Exponential

At this moment, we have only used SGD as one type of optimizer and have not changed optimizer. Because the main focus of this experiment is the analysis of activation function on the overall performance of the network. For this reason, we have implemented 13 different types of activation functions on textual data for the NER task. Among all activation functions, only Sigmoid, SoftMax, Softplus, and Exponential activation functions have performed well whereas all remaining activation functions completely distracted the path by updating *false* weight values of the network and performed poorly in making the network understand complex input data sequences.

VI. RESULTS

In this section we discuss the effect of different activation functions after the implementation on the Bi-LSTM based RNN model. Performance of different activation functions on training accuracy, training loss, validation accuracy, validation loss, and mean loss is focused in this experiment. Among a variety of different activation functions, we have used almost 13 different types of activation functions and analysed their effect one by one. Among all, only *SoftMax, Sigmoid, SoftPlus,* and *Exponential* functions produced good results. For hyperparameter optimization, we have focused only on the effect of activation functions by keeping all other hyperparameters fixed. Only SGD optimizer is used to carry out the whole experiment. Performance comparison of mentioned activation functions is displayed in Fig 4.

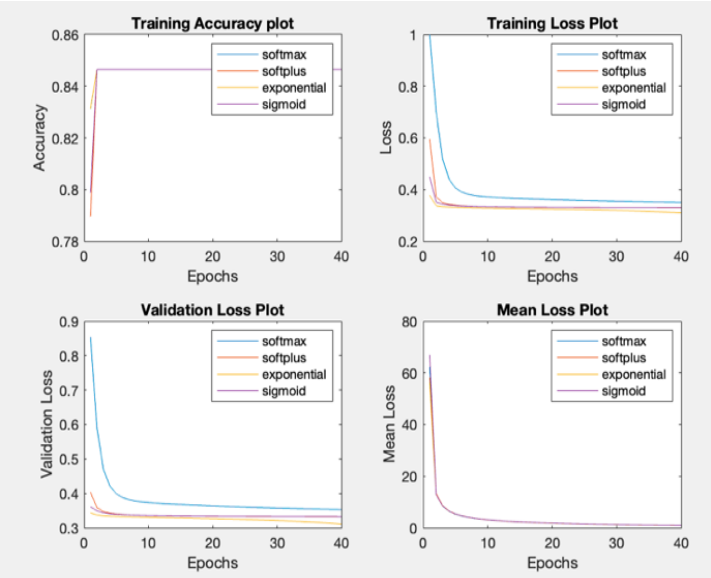

Fig. 4. Comparison of training accuracy, training loss, validation loss and mean loss of implemented Sigmoid, SoftMax, SoftPlus and Exponential activation functions

Accuracy of the neural model is calculated by comparing ground truth values with the predicted values. Here ground truth values correspond to reference labels of NER model and predicted values corresponds to the values generated by our neural model. For the clear understanding of evaluation, a sample of reference data and predicted data is shown below in Fig. 5.

```
Ground Truth: ['B-per I-per O O O O O O B-gpe O O B-org I-org I-org O B-per I-p
er I-per O O B-tim O O <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <O
OV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OO
V> <OOV> <OOV> <OOV> <OOV>']

Prediction: ['B-per I-per O O O O O O B-gpe O O B-per I-per I-per O B-per I-per
I-per O O B-tim O O <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV>
<OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <OOV> <
OOV> <OOV> <OOV> <OOV>']
```

Fig. 5. Accuracy comparison of implemented Sigmoid, SoftMax, SoftPlus and Exponential activation functions

Comparison analysis of activation functions is based on accuracy and loss values. Loss is actually referred to as a prediction error for a deep neural network. A system is considered accurate if it has maximum accuracy and minimum loss. In our experiment, we have analysed that for this specific task: among all activation functions, the accuracy of Sigmoid is the highest one, and the accuracy of SoftMax is the lowest one. Similarly, considering mean loss values, we have found that exponential and SoftMax activation functions have the minimum mean loss than other activation functions. Results based on training accuracy, training loss, validation loss, and mean loss are graphically reported in Fig. 4. Validation accuracy of model is displayed in Fig. 6 and accuracy comparison of mentioned activation functions is reported in Fig. 7.

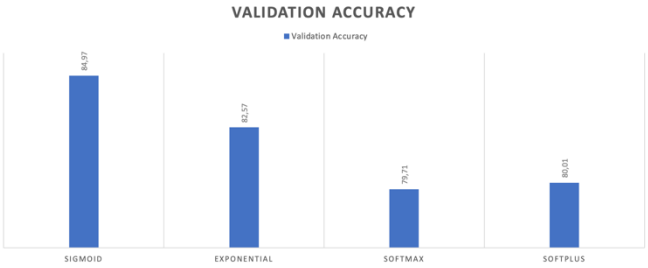

Fig. 6. Validation accuracy comparison of implemented Sigmoid, SoftMax, SoftPlus and Exponential activation functions

Comparison of our results with recent literature is listed in table IV given below. This comparison is based on the accuracy achieved by the models used for Named Entity Recognition task by focusing on LSTM, Bi-LSTM or any variant of these models. For the comparison, we have focused on the papers published recently and using State of The Art (SOTA) models. Our implementation has got more accuracy than all the implemented models.

TABLE IV. IMPLEMENTED SEQUENTIAL RNN MODEL

Ref. No	Model	Accuracy
[1]	RNN Bi-LSTM CRF	92.82%
[2]	BERT Bi-LSTM CRF	80.76%
[3]	Bi-LSTM CRF	84.5%
[4]	Bi-LSTM CNN CRF	89.22%
[5]	LSTM	91.00%
[6]	BERT Bi-LSTM-CNN CRF	94.89%
[7]	BERT Bi-LSTM CRF	89.16
Our Implementation	**Bi-LSTM CRF (Sigmoid)**	**95.17%**
	Bi-LSTM CRF (Exponential)	**95.14%**
	Bi-LSTM CRF (SoftMax)	**94.76%**
	Bi-LSTM CRF (SoftPlus)	**94.38%**

Among these activation functions, best performance of the model is achieved using *Sigmoid* activation function.

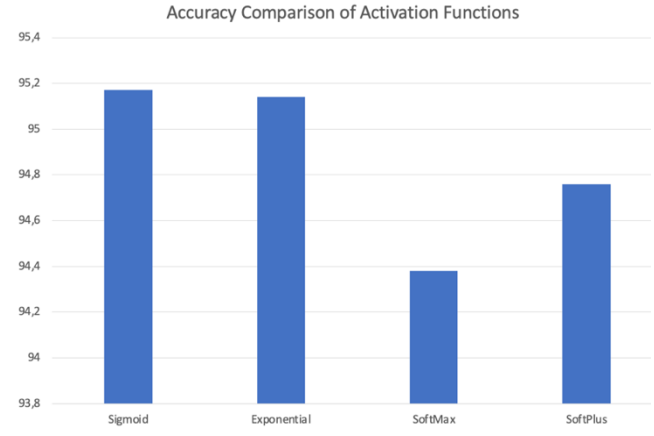

Fig. 7. Accuracy comparison of implemented Sigmoid, SoftMax, SoftPlus and Exponential activation functions

VII. CONCLUSION

The method of choosing the best activation function is considered a hit and trial process. If we have to train complex deep neural architectures with a very large dataset, it will be very difficult to repeat the process for choosing the best activation function. Considering the importance of the role of activation functions in neural networks, we have implemented Bi-LSTM based RNN on the GMB dataset for the NER task and have analysed the effect of activation functions. We have analysed the effect of all activation functions for this specific task and found that only 4 activation functions have produced good results. Sigmoid, SoftMax, SoftPlus, and Exponential activation functions performed very well in terms of accuracy and loss. Based on experimental results, we recommend Sigmoid activation function due to maximum accuracy and minimum loss values. In future experiments, we will tune other hyperparameters and will deeply analyse the effect of optimizers in RNN-based models. Moreover, we will work on higher levels of the GMB annotation schema (e.g. Semantic Role Labelling), in order to build a high-performance semantic analyser.

REFERENCES

[1] Warto, Muljono, Purwanto and E. Noersasongko, "Capitalization Feature and Learning Rate for Improving NER Based on RNN BiLSTM-CRF," 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 2022, pp. 398-403, doi: 10.1109/CyberneticsCom55287.2022.9865660.

[2] J. Wang et al., "Fine-Grained Chinese Named Entity Recognition Based on MacBERT-Attn-BiLSTM-CRF Model," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), 2022, pp. 0125-0131, doi: 10.1109/CCWC54503.2022.9720911.

[3] X. Hu, H. Zhang and S. Hu, "Chinese Named Entity Recognition based on BERTbased-BiLSTM-CRF Model," 2022 IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS), 2022, pp. 100-104, doi: 10.1109/ICIS54925.2022.9882432.

[4] V. Sornlertlamvanich and S. Yuenyong, "Thai Named Entity Recognition Using BiLSTM-CNN-CRF Enhanced by TCC," in IEEE Access, vol. 10, pp. 53043-53052, 2022, doi: 10.1109/ACCESS.2022.3175201.

[5] B. G. Pallavi, E. R. Kumar, R. Karnati and R. A. Kumar, "LSTM Based Named Entity Chunking and Entity Extraction," 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), 2022, pp. 1-4, doi: 10.1109/ICAITPR51569.2022.9844180.

[6] X. Wu, T. Zhang, S. Yuan and Y. Yan, "One Improved Model of Named Entity Recognition by Combining BERT and BiLSTM-CNN for Domain of Chinese Railway Construction," 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), 2022, pp. 728-732, doi: 10.1109/ICSP54964.2022.9778794.

[7] Y. Tian, "Named Entity Recognition in Emergency Domain based on BERT-BILSTM-CRF," 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), 2022, pp. 817-820, doi: 10.1109/ICETCI55101.2022.9832114.

[8] Rosebrock, A. 2017. Deep Learning for Computer Vision with Python. PyImageSearch.

[9] Ide, H., and Kurita, T. 2017. "Improvement of learning for CNN with ReLU activation by sparse regularization," In Proceedings of 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2684–2691. https://doi.org/10.1109/IJCNN.2017.7966185.

[10] Hochreiter, S. 1998. "The vanishing gradient problem during learning recurrent neural nets and problem solutions," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6(02), pp. 107–116.

[11] Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. 2000. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," Nature (405), pp. 947- 951.

[12] Maas, A. L., Hannun, A. Y., and Ng, A. Y. 2013. "Rectifier Nonlinearities Improve Neural Network Acoustic Models," In

Proceedings of the 30th International Conference on Machine Learning.

[13] Agostinelli, F., Hoffman, M., Sadowski, P., and Baldi, P. 2014. "Learning Activation Functions to Improve Deep Neural Networks," ArXiv:1412.6830 [Cs, Stat]. http://arxiv.org/abs/1412.6830.

[14] J. Bos, V. Basile, K. Evang, N. Venhuizen, and J. Bjerva. The groningen meaning bank. In N. Ide and J. Pustejovsky, editors, Handbook of Linguistic Annotation, volume 2, pages 463–496. Springer, 2017.

[15] Hinkelmann, Knut. "Neural Networks, p. 7". University of Applied Sciences Northwestern Switzerland, 2018.

[16] Hodgkin, A. L.; Huxley, A. F. (1952-08-28). "A quantitative description of membrane current and its application to conduction and excitation in nerve". The Journal of Physiology. 117 (4): 500–544. doi:10.1113/jphysiol.1952.sp004764.

[17] Hinton, Geoffrey; Deng, Li; Deng, Li; Yu, Dong; Dahl, George; Mohamed, Abdel-rahman; Jaitly, Navdeep; Senior, Andrew; Vanhoucke, Vincent; Nguyen, Patrick; Sainath, Tara; Kingsbury, Brian (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition". IEEE Signal Processing Magazine. 29 (6): 82–97. doi:10.1109/MSP.2012.2205597. S2CID 206485943

[18] Hendrycks, Dan; Gimpel, Kevin (2016). "Gaussian Error Linear Units (GELUs)". arXiv:1606.08415

[19] Cybenko, G. (December 1989). "Approximation by superpositions of a sigmoidal function". Mathematics of Control, Signals, and Systems. 2 (4): 303–314. doi:10.1007/BF02551274

[20] Snyman, Jan (3 March 2005). Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Science & Business Media. ISBN 978-0-387-24348-1