2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Fast Construction of an Updating System for Intrusion Detection using a Multi-layer Extreme Learning Machine

Daichi NOGUCHI† and Masaharu ADACHI‡

†‡Graduate school of Engineering, Tokyo Denki University 5 Senjyu Asahi-chou, Adachi-ku, Tokyo, 120-8551 Japan
Email: 16kmj42@ms.dendai.ac.jp, adachi_at-mark_eee.dendai.ac.jp

Abstract– Fast construction for an intrusion detection system (IDS) enables rapid detection of, and response to, intrusions into a network. Using deep neural networks is expected to give a high detection rate for an IDS (S.Poluluri, et al., EFTA2016, pp.1-8). However, this requires time-consuming iterative computation. To address this, we propose a method for fast construction of an IDS using a multi-layer Extreme Learning Machine based on Auto Encoder.

1. Introduction

Various methods, including machine learning using neural networks, have been proposed for construction of an intrusion detection system (IDS). Ideally, such a system could handle frequent updates to data [1]. In this article, we aim at improving the detection rate and reducing the time to construct an IDS from a system of deep neural networks based on Auto Encoder (AE). We further aim at improving the detection rates of incidents that are now poorly detected and reducing the learning time for the IDS by using ELM-AE [2], which is based on an Extreme Learning Machine (ELM) and AE. We investigate multi-layer neural networks having the ability to learn huge amounts of data in a short time and detecting intrusions at a high rate.

The rest of the paper is organized as follows. In Section 2, we describe ELM, AE, and ELM-AE [2], which comprises ELM and AE together. Then, in Section 3, we describe the evaluation of the IDS using ELM-AE by one or more layers. Finally, in Section 4, we conclude.

2. Related neural network models

2.1. Extreme Learning Machine

ELM [3] is known to have fast learning capability and high performance.

The fundamental model of ELM supposes a neural network constructed of three layers: input, hidden, and output. Training a network via gradient descent methods requires iteratively reducing the error between the output and target signal until the error satisfies some criterion. ELM trains the system by calculation of the output-layer weight vector $\boldsymbol{\beta}$ as follows.

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \qquad (1)$$

Here, $\boldsymbol{H}^{\dagger}$ is the pseudo inverse matrix of the output matrix of the hidden layer through some non-linear activation functions such as a sigmoid function, and $\boldsymbol{T}$ is the target data vector. The weight coefficient is calculated by applying Eq.(1) once. Therefore, the learning speed of ELM is extremely fast relative to that of a neural network trained using gradient descent methods. The fundamental model of ELM is shown in Fig.1.

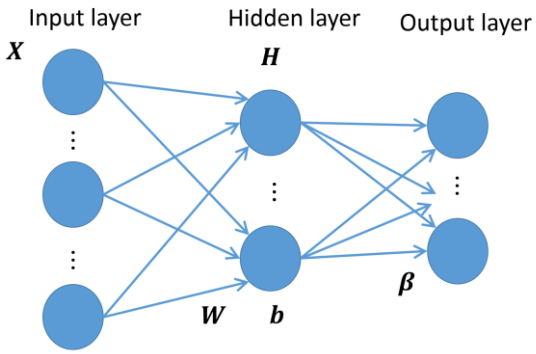

Fig.1 Fundamental architecture of ELM.

In Fig.1, $\boldsymbol{X}$ is the input matrix, $\boldsymbol{W}$ is the weight matrix of the hidden layer, $\boldsymbol{b}$ is the bias of the hidden layer, $\boldsymbol{H}$ is the output matrix of the hidden layer, and $\boldsymbol{\beta}$ is the vector of weights of the output layer.

2.2. Auto Encoder

AE is a neural network having three layers, arranged the same as with ELM. AE trains the hidden layer weights such that the output vectors are equal to the input vectors. This system enables the hidden layers to represent the feature of the input vectors. Typically, AE is trained so that it has a hidden layer with lower dimensionality than the input layer, and so it is often used for the purpose of dimension reduction. Non-linear functions, such as sigmoid, are often used for the activation function of the hidden layer when it is trained. Then, it is trained by a gradient descent method to reduce the error between the data output from the hidden layer and the input data. The architecture of AE is shown in Fig.2.

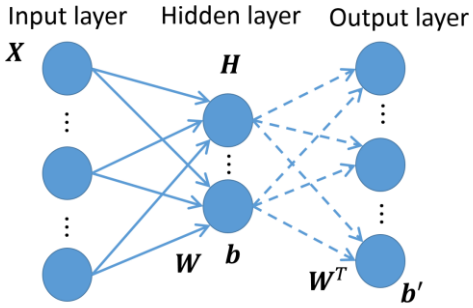

Fig.2 Architecture of AE.

In the figure, $\boldsymbol{b}'$ is the bias vector of the output layer, and the other symbols have the same meaning as in the ELM.

2.3. ELM-AE

ELM-AE [2] is AE trained by applying the training method of ELM to AE without iteration when it extracts the feature using AE. This gives ELM-AE fast learning relative to AE because ELM-AE calculates the hidden-layer weight matrix by a one-time matrix calculation. The output-layer weight matrix β is calculated as follows.

$$\boldsymbol{\beta} = (\boldsymbol{I} / C + \boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{X} \qquad (2)$$

The symbols in Eq.(2) are the following: the input matrix to the hidden layer is $\boldsymbol{X}$, the hidden-layer output matrix is $\boldsymbol{H}$, and the cost parameter is C. Here, $\boldsymbol{I}$ is the identity matrix. The initial hidden-layer weights are given by random values, and these are overwritten by the transposed matrix of $\boldsymbol{\beta}^T$.

2.4. ML-ELM

ML-ELM [2] is a learning machine consisting of multiple layers of ELM-AEs, and so is a multi-layer neural network that is expected to offer fast learning speed and improved classification performance. The output for multi-layer ELM-AE is calculated as follows.

$$\boldsymbol{H}^{(k)} = func(\boldsymbol{H}^{(k-1)}(\boldsymbol{\beta}^{(k)})^T) \qquad (3)$$

Here, the symbols in Eq.(3) are the hidden-layer output matrix and the output weight vector of the kth hidden layer, and a possibly non-linear function *func*.

3. Approach for improving lower detection rate

In this study, we investigated the architecture that would be optimal for IDS using ML-ELM based on AE, aiming at both a high detection rate and a reduced training time. Where we use a sigmoid and softmax function as the activation function for the hidden layer and the output layer. We also evaluated the performance of IDS, using detection rate as a measure of performance. The architecture of the network used in evaluation is shown in Fig.3. The two hidden layers nearest the input-layer side are labeled as hidden layers 1 and 2, and the number of neurons of these layers are called $L^{(1)}$ and $L^{(2)}$, respectively. In this study, the parameters of hidden layer 1 are fixed as $L^{(1)}$ = 840 and $C^{(1)}$ = 10^4 which are increased during searching parameters for the first hidden layer. We suppose that these parameter values are relatively good for all detection rates. In addition, we think the number of hidden nodes L affects the calculation time and it is possible to select the value for desired condition to obtain high detection rates with relatively small computational costs. The parameters of hidden layer 2 are varied.

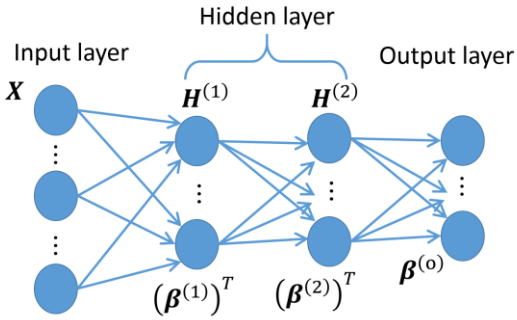

Fig.3 Multi-layer neural network used in simulation.

The symbols in Fig.3 are the following: the weights of hidden layers 1 and 2 are $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\beta}^{(2)}$, respectively; the hidden-layer output matrices are $\boldsymbol{H}^{(1)}$ and $\boldsymbol{H}^{(2)}$, again respectively; and the weight of the output layer is $\boldsymbol{\beta}^{(o)}$. The training time of IDS using ELM-AE, which is based on both ELM and AE, is expected to be short. Therefore, we investigated the construction of the IDS in terms of how well it performs in terms of detection. In the evaluation of detection rate, we used the NSL-KDD dataset [4] and classified each class to one of five classes: Normal, DoS, Probe, R2L, and U2R. We examined how well the IDS could classify data items into the above five classes. The results are shown in Figs.4 and 5. The kinds of correspondence and the samples included in the training dataset and testing dataset used in the simulation are shown in Table 1. For the classifications used as the true classifications, see references [5] and [6]. The detection rates of Normal, DoS, and Probe are shown in Fig.4. In this figure, it can be seen that the detection rate of Normal is the highest, and that DoS is better classified than Probe. It may be necessary to investigate how to stabilize the detection rate of Probe, which fluctuated. The detection rates of R2L and U2R are shown in Fig.5. That figure shows that the detection rates for both R2L and U2R are lower than for the other classes. However, it seems that the two detection rates are relatively higher between L = 640 and L=840 in the second hidden layer than in other ranges for L.

Table 1 NSL-KDD dataset.

Class label	Training samples	Testing samples
Normal	67343	9711
DoS	45927	7460
Probe	11656	2421
R2L	995	2885
U2R	52	67

For reference, we compared ML-ELM with some methods, and we calculated average of 10 execution of the our program. The compared results of detection rates and results of calculation time are shown in Table 2-4 and Table 5. The conditions of each method are shown in Table 6.

Table 2 Average detection rates.

Class label	ELM [%]	ELM-AE [%]	ML-ELM [%]
Normal	76.2	76.3	67.5
DoS	97.7	97.4	84.9
Probe	61.7	57.0	55.0
R2L	0.402	1.99	9.44
U2R	2.53	13.8	21.5

Table 3 Minimum detection rates.

Class label	ELM [%]	ELM-AE [%]	ML-ELM [%]
Normal	71.3	70.9	38.7
DoS	97.5	97.0	8.65
Probe	52.9	49.9	27.3
R2L	0.0347	0.589	0.139
U2R	0.0	4.48	0.0

Table 4 Maximum detection rates.

Class label	ELM [%]	ELM-AE [%]	ML-ELM [%]
Normal	81.2	80.9	77.5
DoS	97.9	97.8	98.3
Probe	77.4	64.1	74.4
R2L	2.36	2.67	50.4
U2R	10.4	25.4	76.1

Table 5 Calculation time.

	ELM	ELM-AE	ML-ELM
Time [sec]	9.90	13.1	40.0

Table 6 Experimental conditions.

Values of parameters	ELM	ELM-AE	ML-ELM
L	500	640	840-740
C	-	10	10^0-10^4

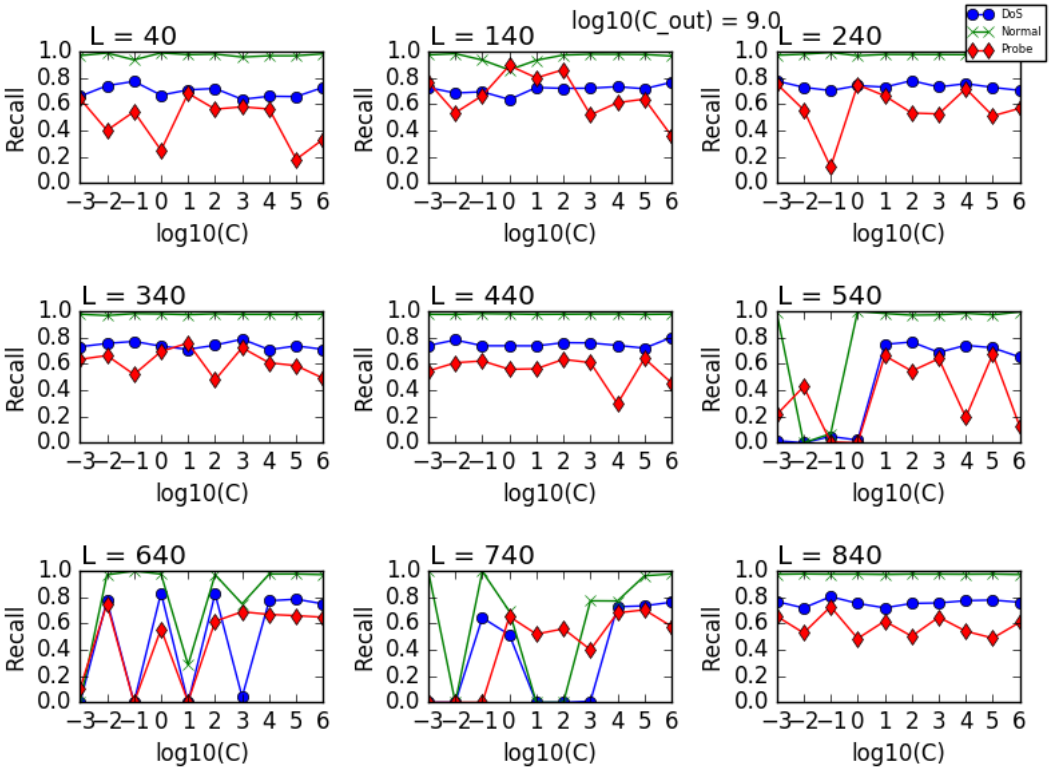

Fig.4 Detection rates of Normal, DoS, and Probe.

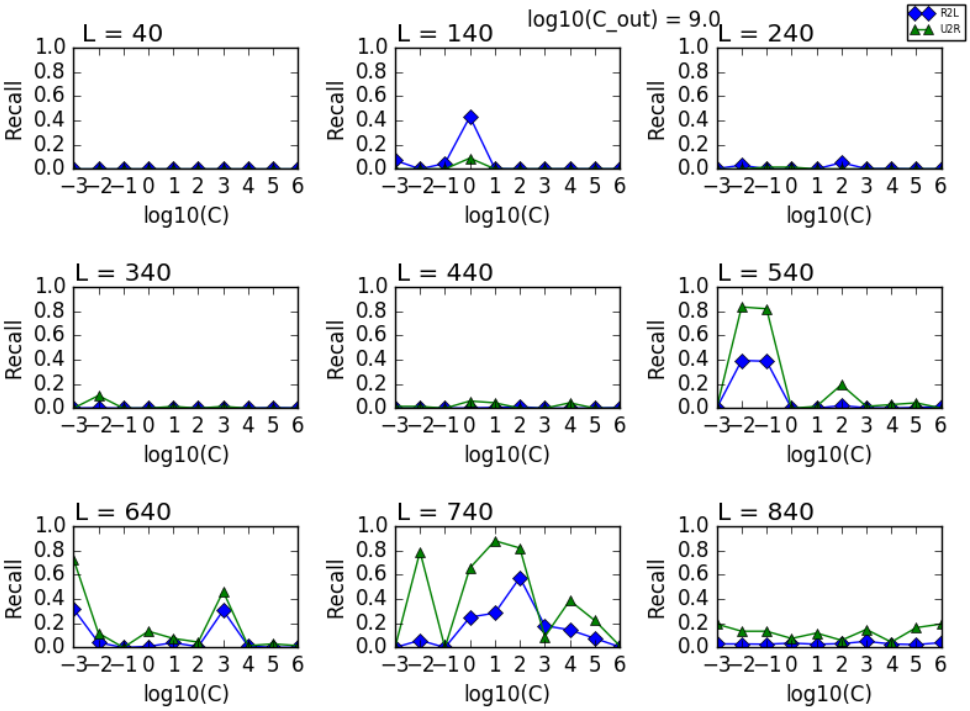

Fig.5 Detection rates of R2L and U2R.

4. Conclusion

In this study, we evaluated the performance of IDS using ML-ELM based on AE [4]. Specifically, we experimentally investigated various conditions that affect detection rate by varying two parameters of the second hidden layer, the cost parameter $C^{(2)}$ and the number of neurons $L^{(2)}$ for a fixed cost parameter $C^{(o)} = 10^9$ in the output layer. In addition, we should investigate how the values of the parameters $L^{(1)}$ and $C^{(1)}$ affect the detection error. The results (see Figs.4–5) show that the detection rates for Normal, DoS, and Probe are higher than the detection for the other two types. This is not a surprise: detection of R2L and U2R is more difficult, and so the detection rates are expected to be lower. We think these results of the two cases are caused by the difference of the number of learning samples or similar characteristics with some classes. We intend to investigate IDS using ELM-AE/ML-ELM as a way to improve the detection rates of R2L and U2R.

In future work, the best values for the number of neurons in each hidden layer and the associated cost parameter should be found while increasing the number of hidden layers. We will refer to the sample-reduction method used in reference [6] when we consider the equalization of samples of five classes during the training step in the future.

Acknowledgements

The authors would like to thank anonymous reviewers for their fruitful comments and suggestions.

References

[1] A.L. Buczak, et al., "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE COMMUNICATIONS SURVEY & TUTORIALS, vol.18, no.2, pp.1153-1176, Secondquarter, 2016.

[2] L.L.C.Kasun, et al., "Representational Learning with ELMs for Big Data," Trends & Controversies Extreme Learning Machines IEEE INTELLIGENT SYSTEMS, vol.28, no.1, pp.31-34, Feb., 2014.

[3] G.Huang, et al., "Extreme learning machine: Theory and applications," NEUROCOMPUTING, Proceedings of 7th Brazilian Symposium on Neural Networks, vol.70, Issues 1-3, pp.489-501, Dec., 2006.

[4] M.Tavallaee, et al, "A Detailed Analysis of the KDD CUP 99 Data Set," Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications pp.1-6, July, 2009.

[5] S.Poluluri, et al., "Accelerated deep neural networks for enhanced Intrusion Detection System," Proceedings of 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp.1-8, Sept., 2016.

[6] R.Singh, et al., "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," Expert Systems With Applications, vol.42, Issue 22 pp.8609-8624, Dec., 2015.