Analysis of training of deep neural networks with heterogeneous architecture while detecting malicious network traffic

V A Chastikova, S A Zherlitsyn, Y I Volya and V V Sotnikov

Department of Computer Technologies and Information Security, Kuban State Technological University, Krasnodar, 350072, Russia

E-mail: chastikova_va@mail.ru

Abstract. The given article presents the study of the training process of a composite heterogeneous neural network with deep architecture. A brief description of the architecture of the analyzed neural network system has been given. The key nodes of computational load distribution on each of the layers of the neural network have been tracked. The monitoring results have been analyzed and the following conclusions have been made: the most resource-intensive part of the system during training is the LSTM network. The results of the study of structural features of the neural network hidden layers have been presented. Graphs have been constructed and there has been carried out the study of a statistical distribution of weights on each unique layer of the architecture: on a fully connected layer, on the first hidden layer of the subnet-encoder, on the second hidden layer of the subnet-encoder, on the first and second fully connected output layers of the neural network. Based on the research results, a qualitative assessment of the effectiveness and accuracy of the entire neural network system has been given.

1. Introduction

Machine learning is currently widely used to solve a lot of applied problems. Models built using such technologies are widely used in all fields of science and practical activities. The wider the range of tasks to be solved becomes, the more complicated these tasks get; therefore it becomes more labor-intensive to construct and train a system satisfying the requirements [1].

The given paper presents the study of the training process of a deep heterogeneous neural network model with the LSTM network module designed to solve the problem of detecting malicious network traffic [2]. One of the largest currently freely distributed abnormal traffic datasets, CSE-CIC-IDS2018 [3], has been used as a training sample. It contains data from network exchange packets and their characteristics, as well as the type of traffic that contains them.

2. Relevance

The availability of computing resources is growing rapidly and some studies estimate such growth as exponentially time-dependent [4], but often there are not enough available capacities to meet emerging needs. In such conditions, it is essential to choose the most suitable tools, as well as to evaluate and distribute the available computing potential in the optimal way.
The given study is aimed at analyzing the distribution of computational load, as well as internal neural network processes, which is especially relevant in the conditions of the current development of machine learning systems and their carriers [5].

3. Formulation the problem

Recurrent neural networks are often used to analyze data having a stream structure or temporal correlations. One of the subclasses of such networks is LSTM (long short-term memory) networks. They are one of the most efficient neural network models for processing ordered datasets (along with Gated Recurred Units networks [6, 7]).

In order to concentrate sparse data and reduce redundancy in the internal representations of the neural network, the given paper suggests using an embedding neural network. It is a multilayer perceptron with a sharply decreasing number of neurons in each subsequent layer, in relation to the input one [8].

The model obtained in the course of the study [8] has the following architecture: embedding network for non-numerical packet characteristics decomposed into binary classes [9]; a parallel fully connected layer [10] for numerical characteristics; LSTM-network [2, 11, 12], that receives concatenation of the outputs of the two previous layers as input. The Focal Loss function [13] is used as a loss function, and Adam [14] is used as a learning rule.

The given deep neural network architecture is heterogeneous [15] and has been developed in the course of the current study specifically to solve the problem of detecting malicious network traffic.

A logging and monitoring module has been developed to ensure the effective study of the learning process. Based on the results of its work, an analysis has been carried out, the results of which are presented below.

4. Theoretical

Since data for monitoring is generated for each batch (lot, fragment of the training sample), it becomes possible to analyze the processes occurring in the neural network during its training [16, 17, 18]. Figure 1 shows the change in the weights of one of the fully connected layers over time, starting with random initialization and ending with a trained model; the figure clearly shows the “specializations” of individual neurons.

![Figure 1](image1.png)

*Figure 1. Changing in the neuron weights of the Dense layer during training.*

Based on the results of the system work, it can be concluded that the maximum computational load falls on the last layers of the neural network — two resulting fully connected layers and a bidirectional LSTM layer — as well as on the neural network training module [19, 20]. According to the testing results of the proposed model, the largest memory consumption is caused by the bidirectional LSTM layer and the neural network training module. A more detailed analysis of the neural network training module shows that the maximum memory consumption is accounted for by the unit for calculating the gradient of the backpropagation of the loss function in the module of the bidirectional LSTM layer. The study of the characteristic structural features of the hidden layers of the subnet-encoder (embedding neural network) and the fully connected layer concatenating with the output of the encoder attracts the
greatest interest in the analysis of statistical data on the weights of neurons. A histogram of the statistical parameter distribution of an individual fully connected layer is presented in figure 2.

![Figure 2. Histogram of the statistical parameter distribution of an individual fully connected layer.](image)

A subnet-encoder contributes less to the detection of abnormal network traffic than a bidirectional LSTM layer. However, in contrast to the layer under consideration, data on the process of its training is much easier to interpret due to a relatively small number of parameters and a simple logical structure. A histogram of the statistical parameter distribution of the first hidden layer of the subnet-encoder is shown in figure 3.

![Figure 3. Histogram of the statistical parameter distribution of the first hidden layer of the subnet-encoder.](image)

A comparative analysis shows that the statistical distribution of the parameters of the first hidden encoder layer is much closer to the normal distribution than the same distribution for an individual fully connected layer, shown in figure 2.
This is due to the fact that the encoder processes a large but sparse array of data encoded with a unitary code. To study the compactification process of this data, it is necessary to consider the next hidden layer of the encoder, shown in figure 4.

The statistical distribution of the weights of the second hidden layer is much closer to the distribution of the second fully connected layer, which allows us to draw a conclusion about the success of the compactification process of information features. The effectiveness of heterogeneous network architecture with a subnet-encoder is also proved by the way the statistical parameter distribution of its second layer changes over time during the learning process: starting with the subnormal one, the distribution converges to values similar to those of an individual fully connected layer, shown in figure 3.

Due to the fact that the output layer of the subnet-encoder is similar in its distribution law to an individual fully connected layer, it seems relevant to compare them with the histograms of the statistical parameter distribution of the two output layers of the resulting deep neural network, shown in figures 5 and 6, respectively.

The statistical distribution of the parameters of the first output layer is somewhat similar to the parameter distribution of the second hidden layer of the subnet-encoder: at the first stage of the neural
network training, the layer parameters form a subnormal distribution, which becomes closer to a sawtooth shape as the network is being trained. Such behavior of a deep neural network indicates an even greater compaction of the processed data arrays and the emergence of the neuron “specializations”, which consist in processing a certain type of hidden informational features.

A histogram of the statistical parameters distribution of the second - and last - output layer of the resulting neural network is shown in figure 6.

The resulting histogram is not similar to any of those that correspond to other layers of the implemented neural network. This is due to the function performed by the output layer - unlike the other layers that performed the task of representation training, encoding information features into a more compact form – the given layer converts an array of hidden information features into specific classes of anomalous network traffic.

5. Conclusion
Training and analysis of the accuracy of the described deep neural network have been carried out using the CSE-CIC-IDS2018 dataset [3]. As a result, high accuracy figures have been obtained. While testing the processing speed, a deep neural network showed good results due to its highly parallel software architecture and the support for parallel computing.

References
[1] Sokolov A V and Chulok A A 2012 Russian Science and Technology Foresight – 2030: Key Features and First Results Foresight 6(1)
[2] Chastikova V A, Malykhina M P, Zherlitsyn S A and Volya Y I 2017 Comparative analysis of some algorithms of swarm intelligence when detecting network attacks by neural network methods Polythematic Network Electronic Scientific Journal of Kuban State Agrarian University 129 106-15
[3] CSE-CIC-IDS2018 on AWS https://www.unb.ca/cic/datasets/ids-2018.html
[4] Denning P J and Lewis T G 2017 Exponential Laws of Computing Growth Communications of the ACM 60(1) 54-65
[5] Chastikova V A, Zherlitsyn S A and Volya Y I 2018 Neural network method of identification by unformalized semantic characteristics News of Volgograd State Technical University 8(218) 63-7
[6] Gers F A, Schmidhuber J and Cummins F 2000 Learning to Forget: Continual Prediction with LSTM Neural Computation 12(10) 2451-71
[7] Glazkova A V 2019 Comparison of neural network models for classifying text fragments containing *Software & Systems* **32**(2) 263-7

[8] Chastikova V A, Zherlitsyn S A, Volya Y I and Sotnikov V V 2020 Neural network technology for detecting anomalous network traffic *Caspian journal: Control and High Technologies* **1**(49) 20-32

[9] Kharakhinov V A and Sosinskaya S S 2018 The effect of dimension reducing on classification results of leaves of various plant species *Software Engineering* **9**(2) 82-90

[10] Haykin S 2006 Neural Networks: A Comprehensive Foundation *Williamams* **2** 1104

[11] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural computation* **9**(8) 1735-80

[12] Chen G, Ye D, Xing Z, Chen J and Cambria E 2017 Ensemble application of convolutional and recurrent neural networks for multi-label text categorization *Intern. Joint Conf. on Neural Networks (IJCNN)* (USA: IEEE) p 2377-83

[13] Lin T-Y, Goyal P, Girshick R, He K and Dollár P 2017 Focal Loss for Dense Object Detection *International Conf. on Computer Vision (ICCV)* (Italy: IEEE)

[14] Ruder S 2016 An overview of gradient descent optimization algorithms *Cornell University Library*

[15] Malikov S N, Safonov V L and Chudinov S M 2012 Heterogeneous computing systems of digital processing of signals *Scientific reports of Belgorod State University* (Belgorod: BelGU) p 125-8

[16] Sozykin A V 2017 An Overview of Methods for Deep Learning in Neural Networks *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering* **6**(3) 28-59

[17] LeCun Y, Bengio Y and Hinton G 2015 Deep Learning *Nature* **521** 436-44

[18] Wilson D R and Martinez T R 2003 The general inefficiency of batch training for gradient descent learning *Neural Networks* **16**(10) 1429-51

[19] Geron A 2017 Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems *O'Reilly Media* 574

[20] Wang T, Chen P, Amaral K and Qiang J 2016 An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification *Cornell University Library* 8