ICAI 2023
12th International Conference on Applied Informatics
Eger, Hungary, 2023

Effective testing of the robustness of ReLU neural networks*

Attila Szász[a], **Balázs Bánhelyi**[b]

[a]Univerity of Szeged
szasza@inf.u-szeged.hu

[b]University of Szeged
banhelyi@inf.u-szeged.hu

Abstract

One of the most important topics in artificial intelligence research today is the verification of neural networks[3]. The accuracy of the networks has continuously increased over the years, so more and more complex neural networks have been created and many tasks could be solved by them. Many modern teaching methods have been developed that have improved the quality of the networks. In certain fields, it is inevitable to be precise and to have fast networks.

In many works, it has been shown that these nets, which are considered to be safe, can also be wrong. It often appears on the input and is invisible to the human eye even a small amount of noise leads to incorrect classification. There are many methods developed by researchers to solve these problems. The methods are mainly divided into 2 classes: robust learning and adversary example detection.

For this reason, neural network verification is an important topic in today's artificial intelligence research. The neural network technique focuses on speed and typically uses floating point arithmetic, while others prefer symbolic methods used for reliability. These systems often have significantly longer run times. In this paper, other methods have been described using the numerical result. These methods have a correct evaluation and a manageable runtime. The system we wrote defines not only the inputs but also the interval of values for the outputs of the given network. When checking to verify a net, these intervals must be as small as possible. During the evaluation, in addition to the nets with the ReLU

*The research has been supported by the grant TKP2021-NVA-09 of the Ministry for Innovation and Technology, Hungary.

activation function, the output widths and the running times were also compared.

Motivation

During our work, we have developed a system based on reliable network assessment. The system supports multiple evaluation methods, in both CPU and GPU environments. Many current contemporary systems use floating-point arithmetic with an emphasis on speed in evaluation. The big disadvantage of this is that some numerical errors in the various operations can accumulate during the evaluation. A good way to get a handle on these errors is to use interval arithmetic. One solution is to compute an interval containing the given value instead of the floating-point number. The method is well suited for neural networks and also for their evaluation since the operations that can be performed on real numbers can be easily extended to intervals. In this case, the inputs of the network are intervals. One of the advantages of the method is that it increases the running time only minimally compared to floating point evaluation. Since it is reliable and robust in class, this method is also used in the evaluation phase.

The network shown in Figure 1. is evaluated according to the rules of naive interval arithmetic. The ReLU activation function is contained in some neurons of the network. The value of the output neuron x_5 lies in the interval $[0; 5]$. It can be observed that the upper limit of 5 would occur only if the neuron x_3 had the value 5 and the neuron x_4 had the value 6. It can be seen that under the given input conditions, these values can never occur simultaneously. As a result, the upper bound of x_5 was never sharp, leading to an overestimation in the evaluation. The main reason for the overestimation is the dependency problem, which can become quite strong and unmanageably wide in the results as the number of layers increases.

The goal of our work is to implement and compare numerically correct systems that handle the dependency problem but do not drastically increase the runtime.

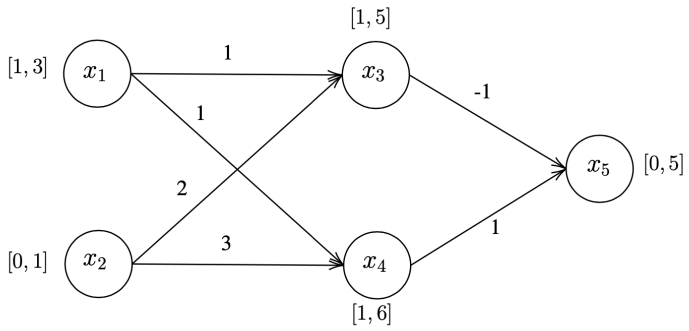
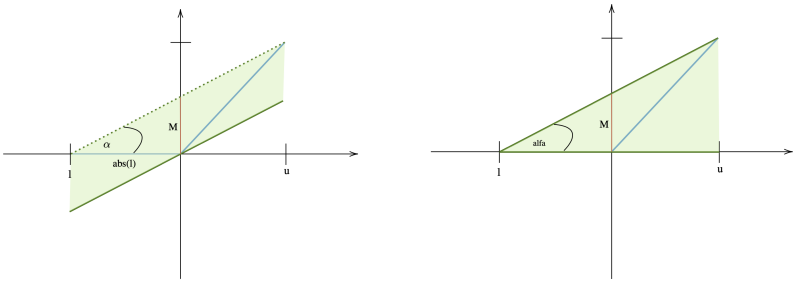


Figure 1. Naive interval arithmetic network evaluation.

Methods

Three techniques were presented, each based on the use of a 1-1 function for the set of values of neurons on the given input set. The first technique is symbolic propagation[4] (see Figure 2a.), where we search for the best-fitting function and then handle the nonlinearity with new variables. In the second case, 2 separate functions were held for each neuron and calculated with them (see Figure 2b). In the third case, the well-known linear affine expression was used.



(a) Symbolic propagation case. **(b)** 2 functions case.

Figure 2. ReLU inclusions.

Results

In this paper, we demonstrate the effectiveness of different techniques on multiple neural networks. The cpu/gpu time of the algorithms was shown on self-trained different sizes networks and was shown on the usual MNIST[1] and ERAN[2] networks. Also, we will separately explain how the computing time develops in the case of networks trained with other robust techniques.

References

[1] L. Deng: *The mnist database of handwritten digit images for machine learning research*, IEEE Signal Processing Magazine 29.6 (2012), pp. 141–142.

[2] *ETH Robustness Analyzer for Deep Neural Networks*, 2022, URL: https://github.com/eth-sri/eran.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus: *Intriguing properties of neural networks*, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, ed. by Y. Bengio, Y. LeCun, 2014, URL: http://arxiv.org/abs/1312.6199.

[4] S. Wang, K. Pei, J. Whitehouse, J. Yang, S. Jana: *Formal Security Analysis of Neural Networks Using Symbolic Intervals*, in: Proceedings of the 27th USENIX Conference on Security Symposium, SEC'18, Baltimore, MD, USA: USENIX Association, 2018, pp. 1599–1614, ISBN: 9781931971461.