Seminar Formal Methods for Learned Systems, Project proposal

Iterative Marabou

tightening verifiable bounds

Jop Schaap

March 25, 2024

---

1 Introduction

Deep Neural Networks(DNN) show a very high performance on a wide variety of tasks. However, the complexity of DNNs, makes it difficult to know whether the mapping from the inputs to outputs is correct for all possible input output pairs. This in turn prevents us from reasoning whether or not the DNN is correctly functioning over an input space [1]. However, DNNs are often used in safety critical systems, such as for autonomous driving [2]. Thus it is important to find and verify properties of these DNNs.

One such property of DNNs is to know when the network is safe to use. For this project I suggest a framework to iteratively apply formal solvers, to obtain input bounds that are guaranteed to produce a bounded output. I want to use tne Marabou[3] tool, which is an extension of Reluplex[4]. Reluplex (and by extension Marabou) is capable of verifying whether certain outputs of the DNN are impossible to generate given some input space. However, Reluplex is unable to find bounds for input spaces under which restrictions are upheld. The main idea I want to apply here, is to iteratively apply the Marabou solver with larger input bounds until the solver finds a counter example showing the new bound to be too wide.

Thus the research question I would like to answer in this project is:

- How effective is my proposed framework in combination with Reluplex to construct safe input spaces?

Here I want to measure effectiveness in how wide the proposed approach would be able to construct the input space, within a limited time.

2 Preliminaries

In this section I will introduce terms and definition necessary to understand this work. I will start by first describing the inner workings of *Reluplex* and *Marabou*. Hereafter I will discuss the *quick find* algorithm and how this helps our search.

2.1 Reluplex

TODO! Here I will fully explain Reluplex, up to an extend where I find it necesarry

2.2 Marabou

TODO! Here I'll give a quick overview in what ways Marabou

2.3 Quick find

TODO! Explain quickfind and also how to apply this in 2d space

3 Topic

To let the technique work we require that we have at least one point for which the properties hold. Finding of this point is outside the scope of this report and I instead assume this starting point to be 0. For the project I want to create an algorithm as described below:

Algorithm 1 Iterative Marabou algorithm

$min_{LB}[num\_input\_parameters] \leftarrow [-\infty]$
$min_{UB}[num\_input\_parameters] \leftarrow [0]$
$max_{LB}[num\_input\_parameters] \leftarrow [0]$
$max_{UB}[num\_input\_parameters] \leftarrow [\infty]$

while $time\_to\_run > 0$ **do**
    $dim \leftarrow$ random_dim() ▷ Optimize the minimum
    $mid \leftarrow (min_{LB}[dim] + min_{UB}[dim])$ ▷ Get the midpoint as done in quickfind
    addProperty$(input_parameters[dim] \geq mid)$
    $upholds \leftarrow$ check_properties()
    if $upholds$ **then**
        $min_{LB}[dim] \leftarrow mid$ ▷ If the properties still hold we found a better bound
    else
        remove_last_property()
        $min_{UB} \leftarrow mid$ ▷ If the properties don't hold we find a limitation of the bound
    end if

    $mid \leftarrow (max_{LB}[dim] + max_{UB}[dim])$ ▷ Optimize the maximum
    addProperty$(input_parameters[dim] \leq mid)$
    $upholds \leftarrow$ check_properties()
    if $upholds$ **then**
        $max_{LB}[dim] \leftarrow mid$
    else
        remove_last_property()
        $max_{UB} \leftarrow mid$
    end if
end while
return min_{UB}, max_{LB}

4 Conclusion

TODO!

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, *Intriguing properties of neural networks*, Feb. 2014. DOI: 10.48550/arXiv.1312.6199. arXiv: 1312.6199 [cs]. (visited on 03/21/2024).

[2] M. Bojarski, D. Del Testa, D. Dworakowski, *et al.*, *End to End Learning for Self-Driving Cars*, Apr. 2016. DOI: 10.48550/arXiv.1604.07316. arXiv: 1604.07316 [cs]. (visited on 03/22/2024).

[3] G. Katz, D. A. Huang, D. Ibeling, *et al.*, "The Marabou Framework for Verification and Analysis of Deep Neural Networks," in *Computer Aided Verification*, I. Dillig and S. Tasiran, Eds., vol. 11561, Cham: Springer International Publishing, 2019, pp. 443–452, ISBN: 978-3-030-25539-8 978-3-030-25540-4. DOI: 10.1007/978-3-030-25540-4_26. (visited on 03/21/2024).

[4] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 97–117, ISBN: 978-3-319-63387-9. DOI: 10.1007/978-3-319-63387-9_5.