Algorithms for Verifying Deep Neural Networks

Other titles in Foundations and Trends® in Optimization

Atomic Decomposition via Polar Alignment: The Geometry of Structured Optimization
Zhenan Fan, Halyun Jeong, Yifan Sun and Michael P. Friedlander
ISBN: 978-1-68083-742-1

Optimization Methods for Financial Index Tracking: From Theory to Practice
Konstantinos Benidis, Yiyong Feng and Daniel P. Palomar
ISBN: 978-1-68083-464-2

The Many Faces of Degeneracy in Conic Optimization
Dmitriy Drusvyatskiy and Henry Wolkowicz
ISBN: 978-1-68083-390-4

Algorithms for Verifying Deep Neural Networks

Changliu Liu
Carnegie Mellon University
cliu6@andrew.cmu.edu

Tomer Arnon
Stanford University
tarnon@stanford.edu

Christopher Lazarus
Stanford University
clazarus@stanford.edu

Christopher Strong
Stanford University
castrong@stanford.edu

Clark Barrett
Stanford University
barrett@cs.stanford.edu

Mykel J. Kochenderfer
Stanford University
mykel@stanford.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Optimization

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett and M. J. Kochenderfer. *Algorithms for Verifying Deep Neural Networks*. Foundations and Trends® in Optimization, vol. 4, no. 3-4, pp. 244–404, 2021.

ISBN: 978-1-68083-787-2

Foundations and Trends® in Optimization

Volume 4, Issue 3-4, 2020

Editorial Scope

Topics

Foundations and Trends® in Optimization publishes survey and tutorial articles in the following topics:

- algorithm design, analysis, and implementation (especially, on modern computing platforms
- models and modeling systems, new optimization formulations for practical problems
- applications of optimization in machine learning, statistics, and data analysis, signal and image processing, computational economics and finance, engineering design, scheduling and resource allocation, and other areas

Information for Librarians

Foundations and Trends® in Optimization, 2020, Volume 4, 4 issues. ISSN paper version 2167-3888. ISSN online version 2167-3918. Also available as a combined paper and online subscription.

Contents

Algorithms for Verifying Deep Neural Networks

Changliu Liu[1], Tomer Arnon[2], Chris Lazarus[3], Christopher Strong[4], Clark Barrett[5] and Mykel J. Kochenderfer[6]

[1] *Carnegie Mellon University; cliu6@andrew.cmu.edu*
[2] *Stanford University; tarnon@stanford.edu*
[3] *Stanford University; clazarus@stanford.edu*
[4] *Stanford University; castrong@stanford.edu*
[5] *Stanford University; barrett@cs.stanford.edu*
[6] *Stanford University; mykel@stanford.edu*

ABSTRACT

Deep neural networks are widely used for nonlinear function approximation, with applications ranging from computer vision to control. Although these networks involve the composition of simple arithmetic operations, it can be very challenging to verify whether a particular network satisfies certain input-output properties. This article surveys methods that have emerged recently for soundly verifying such properties. These methods borrow insights from reachability analysis, optimization, and search. We discuss fundamental differences and connections between existing algorithms. In addition, we provide pedagogical implementations of existing methods and compare them on a set of benchmark problems.

Changliu Liu, Tomer Arnon, Chris Lazarus, Christopher Strong, Clark Barrett and Mykel J. Kochenderfer (2021), "Algorithms for Verifying Deep Neural Networks", Foundations and Trends® in Optimization: Vol. 4, No. 3-4, pp 244–404. DOI: 10.1561/2400000035.

1

Introduction

Neural networks [1] have been widely used in many applications, such as image classification and understanding [2], language processing [3], and control of autonomous systems [4]. These networks represent functions that map inputs to outputs through a sequence of layers. At each layer, the input to that layer undergoes an affine transformation followed by a simple nonlinear transformation before being passed to the next layer. These nonlinear transformations are often called *activation functions*, and a common example is the *rectified linear unit* (ReLU), which transforms the input by setting any negative values to zero. Although the computation involved in a neural network is quite simple, these networks can represent complex nonlinear functions by appropriately choosing the matrices that define the affine transformations. The matrices are often learned from data using stochastic gradient descent.

Neural networks are being used for increasingly important tasks, and in some cases, incorrect outputs can lead to costly consequences. Traditionally, validation of neural networks has largely focused on evaluating the network on a large collection of points in the input space and determining whether the outputs are as desired. However, since the input space is effectively infinite in cardinality, it is not feasible to check all possible inputs. Even networks that perform well on a large sample of inputs may not correctly generalize to new situations and may be vulnerable to adversarial attacks [5].

This article surveys a class of methods that are capable of formally verifying properties of deep neural networks over the full input space. A property can be formulated as a statement that if the input belongs to some set $\mathcal{X}$, then the output will belong to some set $\mathcal{Y}$. To illustrate, in classification problems, it can be useful to verify that points near a training example belong to the same class as that example. In the control of physical problems, it can be useful to verify that the outputs from a network satisfy hard safety constraints.

The verification algorithms that we survey are *sound*, meaning that they will only report that a property holds if the property actually holds. Some of the algorithms that we discuss are also *complete*, meaning that whenever the property holds, the algorithm will correctly state that it holds. However, some of the algorithms compromise completeness in their use of approximations to improve computational efficiency.

The algorithms may be classified based on whether they draw insights from these three categories of analysis:

1. *Reachability.* These methods use layer-by-layer reachability analysis of the network. Representative methods are ExactReach [6], MaxSens [7], NNV [8], SymBox [9], Ai2 [10], and ERAN [11]–[14]. Some other approaches also use reachability methods (such as interval arithmetic) to compute bounds on the values of the nodes.

2. *Optimization.* These methods use optimization to falsify the assertion. The function represented by the neural network is a constraint to be considered in the optimization. As a result, the optimization problem is not convex. In *primal optimization*, different methods are developed to encode the nonlinear activation functions as linear constraints. Examples include NSVerify [15], MIPVerify [16], and ILP [17]. The constraints can also be simplified through *dual optimization.* Representative methods for dual optimization include Lagrangian dual methods such as Duality [18], ConvDual [19], and LagrangianDecomposition [20], and

semidefinite programming methods such as Certify [21] and SDP [22].

3. *Search.* These methods search for a case to falsify the assertion. Search is usually combined with either reachability or optimization, as the latter two methods provide possible search directions. Representative methods for *search and reachability* include ReluVal [23], Neurify [24], DLV [25], Fast-Lin [26], Fast-Lip [26], CROWN [27], nnenum [28], and VeriNet [29]. Representative methods for *search and optimization* include Reluplex [30], Marabou [31], Planet [32], Sherlock [33], Venus [34], PeregriNN [35], and BaB [36] and its extensions [20], [37], [38]. Some of these methods call Boolean satisfiability (SAT) or satisfiability modulo theories (SMT) solvers [39] to verify networks with only ReLU activations.

Scope of this article. This article introduces a unified mathematical framework for verifying neural networks, classifies existing methods under this framework, provides pedagogical implementations of existing methods,[1] and compares those methods on a set of benchmark problems.[2]

The following topics are not included in the discussion:

- neural network testing methods that generate test cases [44]–[47];
- white box approaches that build mappings from network parameters to some functional description [48];
- verification of binarized neural networks [49]–[51];

[1]Our implementation is provided in the Julia programming language. We have found the language to be ideal for specifying algorithms in human readable form [40]. The full implementation may be found at https://github.com/sisl/NeuralVerification.jl.

[2]There have been other reviews of methods for verifying neural networks. Leofante, Narodytska, Pulina, *et al.* review primal optimization methods that encode ReLU networks as mixed integer programming problems together with search and optimization under the framework of Boolean satisfiability and SMT [41]. Xiang, Musau, Wild, *et al.* review a broader range of verification techniques in addition to safe control and learning [42]. Salman, Yang, Zhang, *et al.* review and compare methods that use convex relaxations to compute robustness bounds of ReLU networks [43].

- closed-loop safety, stability and robustness by executing control policies defined by neural networks [52], [53], or verification of recurrent neural networks [54];
- training or retraining methods to make a network satisfy a property [19], [21], [55];
- robustness of the verification algorithm under floating point arithmetic [12];
- simplification or compression of the network to improve verification efficiency [56], [57].

Chapter 2 discusses the mathematical problem for verification. Chapter 3 gives an overview of the categories of methods that we will consider. Chapter 4 introduces preliminary and background mathematics. Chapter 5 discusses reachability methods. Chapter 6 discusses methods for primal optimization. Chapter 7 discusses methods for dual optimization. Chapter 8 discusses methods for search and reachability. Chapter 9 discusses methods for search and optimization. Chapter 10 compares those methods. Chapter 11 concludes the article.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT Press, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," p. 529, *Nature*, vol. 518, no. 7540, 2015.

[5] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

[6] W. Xiang, H.-D. Tran, and T. T. Johnson, "Reachable set computation and safety verification for neural networks with relu activations," *ArXiv*, no. 1712.08163, 2017.

[7] W. Xiang, H. Tran, and T. T. Johnson, “Output reachable set estimation and verification for multilayer neural networks,” pp. 5777–5783, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, Nov. 2018.

[8] H.-D. Tran, X. Yang, D. M. Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, and T. T. Johnson, “NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems,” in *International Conference on Computer-Aided Verification (CAV)*, Jul. 2020.

[9] J. Li, J. Liu, P. Yang, L. Chen, X. Huang, and L. Zhang, “Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification,” in *Static Analysis*, Cham, 2019.

[10] T. Gehr, M. Mirman, D. Drashsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, “Ai2: Safety and robustness certification of neural networks with abstract interpretation,” in *IEEE Symposium on Security and Privacy (SP)*, 2018.

[11] G. Singh, R. Ganvir, M. Püschel, and M. Vechev, “Beyond the single neuron convex barrier for neural network certification,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev, “Fast and effective robustness certification,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[13] G. Singh, T. Gehr, M. Puschel, and M. Vechev, “An abstract domain for certifying neural networks,” in *ACM Symposium on Principles of Programming Languages*, 2019.

[14] G. Singh, T. Gehr, M. Puschel, and M. Vechev, “Boosting robustness certification of neural networks,” in *International Conference on Learning Representations*, 2019.

[15] A. Lomuscio and L. Maganti, “An approach to reachability analysis for feed-forward relu neural networks,” *ArXiv*, no. 1706.07351, 2017.

[16] V. Tjeng, K. Xiao, and R. Tedrake, “Evaluating robustness of neural networks with mixed integer programming,” *ArXiv*, no. 1711.07356, 2017.

[17] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, "Measuring neural net robustness with constraints," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[18] K. Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli, "A dual approach to scalable verification of deep networks," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[19] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning (ICML)*, Oct. 2018.

[20] R. Bunel, A. De Palma, A. Desmaison, K. Dvijotham, P. Kohli, P. H. Torr, and M. P. Kumar, "Lagrangian decomposition for neural network verification," *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[21] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *International Conference on Learning Representations*, 2018.

[22] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *ArXiv*, no. 1903.01287, 2019.

[23] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *USENIX Security Symposium*, 2018.

[24] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Efficient formal safety analysis of neural networks," in *Advances in Neural Information Processing Systems*, 2018.

[25] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *International Conference on Computer Aided Verification*, 2017.

[26] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, "Towards fast computation of certified robustness for ReLU networks," in *International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 80, Oct. 2018.

[27] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[28] S. Bak, "Execution-guided overapproximation (ego) for improving scalability of neural network verification," in *International Workshop on Verification of Neural Networks*, 2020.

[29] P. Henriksen and A. Lomuscio, "Efficient neural network verification via adaptive refinement and adversarial search," in *European Conference on Artificial Intelligence (ECAI)*, 2020.

[30] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*, 2017.

[31] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić, *et al.*, "The marabou framework for verification and analysis of deep neural networks," in *International Conference on Computer Aided Verification*, 2019.

[32] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," in *International Symposium on Automated Technology for Verification and Analysis*, 2017.

[33] S. Dutta, S. Jha, S. Sanakaranarayanan, and A. Tiwari, "Output range analysis for deep neural networks," *ArXiv*, no. 1709.09130, 2017.

[34] E. Botoeva, P. Kouvaros, J. Kronqvist, A. Lomuscio, and R. Misener, "Efficient verification of neural networks via dependency analysis," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[35] H. Khedr, J. Ferlez, and Y. Shoukry, "Effective formal verification of neural networks using the geometry of linear regions," *ArXiv*, no. 2006.10864, 2020.

[36] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A unified view of piecewise linear neural network verification," in *Advances in Neural Information Processing Systems*, 2018.

[37] R. Bunel, J. Lu, I. Turkaslan, P. Kohli, P. Torr, and M. P. Kumar, "Branch and bound for piecewise linear neural network verification," *Journal of Machine Learning Research*, vol. 21, no. 2020, 2020.

[38] J. Lu and M. P. Kumar, "Neural network branching for neural network verification," in *International Conference on Learning Representations*, 2020.

[39] C. Barrett and C. Tinelli, "Satisfiability modulo theories," in *Handbook of Model Checking*, Springer, 2018, pp. 305–343.

[40] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," pp. 65–98, *SIAM Review*, vol. 59, no. 1, 2017.

[41] F. Leofante, N. Narodytska, L. Pulina, and A. Tacchella, "Automated verification of neural networks: Advances, challenges and perspectives," *ArXiv*, no. 1805.09938, 2018.

[42] W. Xiang, P. Musau, A. A. Wild, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, and T. T. Johnson, "Verification for machine learning, autonomy, and neural networks survey," *ArXiv*, no. 1810.01989, 2018.

[43] H. Salman, G. Yang, H. Zhang, C.-J. Hsieh, and P. Zhang, "A convex relaxation barrier to tight robustness verification of neural networks," in *Advances in Neural Information Processing Systems*, 2019.

[44] Y. Sun, X. Huang, and D. Kroening, "Testing deep neural networks," *ArXiv*, no. 1803.04792, 2018.

[45] K. J. Hayhurst, D. S. Veerhusen, J. J. Chilenski, and L. K. Rierson, "A practical tutorial on modified condition/decision coverage," 2001.

[46] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *Symposium on Operating Systems Principles*, 2017.

[47] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *International Conference on Software Engineering*, 2018.

[48] J. D. Olden and D. A. Jackson, “Illuminating the “black box”: A randomization approach for understanding variable contributions in artificial neural networks,” pp. 135–150, *Ecological Modelling*, vol. 154, no. 1-2, 2002.

[49] C.-H. Cheng, G. Nührenberg, and H. Ruess, “Verification of binarized neural networks,” *ArXiv*, no. 1710.03107, 2017.

[50] N. Narodytska, S. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, “Verifying properties of binarized deep neural networks,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[51] C.-H. Cheng, G. Nührenberg, C.-H. Huang, and H. Ruess, “Verification of binarized neural networks via inter-neuron factoring,” in *Verified Software. Theories, Tools, and Experiments*, 2018.

[52] W. Xiang, H. Tran, J. A. Rosenfeld, and T. T. Johnson, “Reachable set estimation and safety verification for piecewise linear systems with neural network controllers,” in *American Control Conference (ACC)*, Jun. 2018.

[53] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, “Learning and verification of feedback control systems using feedforward neural networks.,” in *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS)*, 2018.

[54] M. E. Akintunde, A. Kevorchian, A. Lomuscio, and E. Pirovano, “Verification of rnn-based neural agent-environment systems,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[55] M. Mirman, T. Gehr, and M. Vechev, “Differentiable abstract interpretation for provably robust neural networks,” in *International Conference on Machine Learning (ICML)*, 2018.

[56] Y. Y. Elboher, J. Gottschlich, and G. Katz, “An abstraction-based framework for neural network verification,” in *International Conference on Computer Aided Verification*, 2020.

[57] P. Prabhakar and Z. R. Afzal, “Abstraction based output range analysis for neural networks,” in *Advances in Neural Information Processing Systems*, 2019.

[58] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling, “JuliaReach: A toolbox for set-based reachability,” in *ACM International Conference on Hybrid Systems: Computation and Control*, 2019.

[59] M. E. Akintunde, A. Lomuscio, L. Maganti, and E. Pirovano, "Reachability analysis for neural agent-environment systems," in *International Conference on Principles of Knowledge Representation and Reasoning*, 2018.

[60] B. G. Anderson, Z. Ma, J. Li, and S. Sojoudi, "Tightened convex relaxations for neural network robustness certification," *ArXiv*, no. 2004.00570, 2020.

[61] H. Zhang, P. Zhang, and C.-J. Hsieh, "Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications," in *AAAI Conference on Artificial Intelligence (AAAI)*, Dec. 2019.

[62] H.-D. Tran, D. M. Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Star-based reachability analysis of deep neural networks," in *International Symposium on Formal Methods*, 2019.

[63] H.-D. Tran, P. Musau, D. M. Lopez, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Parallelizable reachability analysis algorithms for feed-forward neural networks," in *IEEE/ACM International Conference on Formal Methods in Software Engineering (FormaliSE)*, 2019.

[64] X. Yang, H.-D. Tran, W. Xiang, and T. Johnson, "Reachability analysis for feed-forward neural networks using face lattices," *ArXiv*, no. 2003.01226, 2020.

[65] H.-D. Tran, S. Bak, W. Xiang, and T. T. Johnson, "Verification of deep convolutional neural networks using imagestars," *ArXiv*, no. 2004.05511, 2020.

[66] W. Xiang, H.-D. Tran, and T. T. Johnson, "Specification-guided safety verification for feedforward neural networks," *ArXiv*, no. 1812.06161, 2018.

[67] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," pp. 295–320, *SIAM Review*, vol. 59, no. 2, 2017.

[68] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma, "Strong mixed-integer programming formulations for trained neural networks," pp. 1–37, *Mathematical Programming*, 2020.

[69] Y. Zhang and Z. Zhang, "Dual neural network," in *Repetitive Motion Planning and Control of Redundant Robot Manipulators*, pp. 33–56, Springer, 2013.

[70] V. Rubies-Royo, R. Calandra, D. M. Stipanovic, and C. Tomlin, "Fast neural network verification via shadow prices," *ArXiv*, no. 1902.07247, 2019.

[71] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *Proceedings of the 1988 Connectionist Models Summer School*, vol. 1, 1988.

[72] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.

[73] S. Bak, H.-D. Tran, K. Hobbs, and T. T. Johnson, "Improved geometric path enumeration for verifying ReLU neural networks," in *International Conference on Computer-Aided Verification (CAV)*, Jul. 2020.

[74] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," in *International Conference on Learning Representations*, May 2018.

[75] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[76] K. Julian, M. J. Kochenderfer, and M. P. Owen, "Deep neural network compression for aircraft collision avoidance systems," pp. 598–608, *AIAA Journal of Guidance, Control, and Dynamics*, vol. 42, no. 3, 2019.

[77] "Neural Network Verifications Workshop, VNN-COMP," *International Conference on Computer-Aided Verification*, 2020. [Online]. Available: https://sites.google.com/view/vnn20/vnncomp.