Automated machine learning for neural network verification

König, H.M.T.

Chapter 1

Introduction

Neural networks have become increasingly prominent ever since Alan Turing first proposed his idea of *unorganised machines* – computer programs based on trainable networks of largely randomly connected, neuron-like elements [105]. Nowadays, neural networks can be found in various applications, ranging from healthcare to generating artworks, and have enabled the rise of big AI companies, such as OpenAI or Tesla. These neural networks typically consist of millions or even billions of parameters and are commonly referred to as *deep neural networks*.

With the increased adaption of deep neural networks comes the call for safety and trustworthiness of the systems in which they are employed. However, deep neural networks are highly complex and generally suffer from poor explainability; *i.e.*, it often remains unclear how their output was reached. At the same time, they are inherently fragile, and their behaviour is sometimes unexpected and, even more concernedly, unintended (see, *e.g.*, [101]). In some cases, this unintended behaviour can lead to severe consequences, *e.g.*, in the case of the misclassification of traffic signs. Therefore, it becomes necessary to provide formal guarantees about their behaviour; these guarantees can be obtained via *formal verification*. In general, formal software verification seeks to prove or disprove the correctness of a computer program with respect to a certain pre-defined *property* or formal *specification*, using mathematically rigorous techniques.

The most commonly studied verification property of deep neural networks is the *local robustness* property [37, 89]. Local robustness means that a trained neural network produces the same (correct) output when small perturbations are applied to its inputs. Informally speaking, a local robustness property could specify that the image of a speed limit sign is not confused with that of a yield sign due to a small input perturbation,

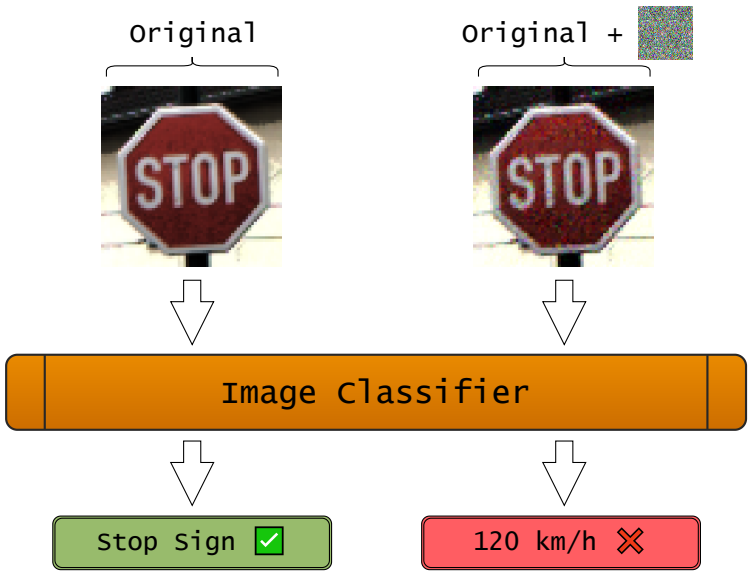

Figure 1.1: An example of an adversarial attack. The initial image is accurately identified as a stop sign. However, applying a specific perturbation to this image can lead the image classification model to produce an incorrect prediction, despite the image evidently showing a stop sign. Source: `https://kennysong.github.io/adversarial.js`

as illustrated in Figure 1.1. This specific phenomenon of misclassification due to minor input variations is canonically referred to as *adversarial attack*. Notice that the local robustness property is flexible in the sense that the degree of perturbations as well as output specification can be adjusted to the given use case.

The key challenge of the verification task is to formally describe the behaviour of the neural network model. However, in a deep learning setting, we typically do not know the concept underlying the learning task. For example, we do not know, which pixel values or features make an image belong to a certain class in an image classification setting. Hence, if we cannot define the task the network is supposed to learn, we also cannot prove whether the network correctly learned the intended concept.

Instead, to formally prove its correctness with respect to a given property, we must encode the neural network. There are several ways to encode the network or, in other words, formally specify the network and its function, enabling us to reason over these encodings with respect to a given correctness property. However, at the time of writing this dissertation, there exists a plethora of encoding techniques as well as reasoning methods. Furthermore, the neural network verification task has been shown to be NP-complete [55]. In this work, we seek to provide a better overview of state-of-the-art

neural network verification methods (with a focus on local robustness properties), to understand better the strengths and weaknesses of existing algorithms, and to present meta-algorithmic approaches to reduce the computational costs involved with tackling neural network verification tasks.

1.1 Research Questions

Neural network verification with respect to local robustness is a highly diverse research area, and existing methods rely on a broad range of techniques. This raises the question which verification algorithm is most suitable for solving specific types of instances of the verification problem, and what constitutes the state of the art in neural network verification overall, also taking into account different hardware specifications, as some methods rely on CPUs, while others utilise GPU acceleration. There might exist a single verifier dominating in performance over other methods, or it might depend on the exact problem type under consideration. In essence, we seek to answer the following research question:

RQ1 (Chapter 3) What constitutes the state of the art in neural network verification?

In general, different problem types require different solving approaches, or well-calibrated adaptations of the same approach. This raises the automated algorithm configuration problem. In the context of neural network verification, this becomes especially relevant since some verification approaches rely on mixed integer linear programming (MIP), where it is well known that state-of-the-art solvers (*e.g.* CPLEX) employed by these systems are highly sensitive to the setting of their hyper-parameters. At the same time, configuring a MIP solver embedded into a neural network verification engine introduces new challenges and considerations, such as the heterogeneity of problem instances, which makes it hard to select a single configuration that works well on every instance. Moreover, this introduces the following research question:

RQ2 (Chapter 4) How can we improve the performance of a MIP-based verification system, leveraging automated algorithm configuration techniques?

Given the inherent complexity of the neural network verification task, solving these problems remains a resource-intensive task even when using state-of-the-art and carefully tuned algorithms. The complexity is further amplified in a portfolio setting, where multiple algorithms run in parallel, introducing inefficiencies through the allocation of computational budget to less effective solvers that run concurrently with

the optimal one until a solution is found. Additionally, there exists the possibility that all algorithms in the portfolio may fail to solve certain instances. This introduces the more general problem of spending compute resources on instances that eventually turn out to be unsolvable within a set cutoff time, *i.e.*, the maximum allowable running time after which the algorithm is terminated. Consequently, a verification system would operate more efficiently if compute resources were allocated towards problem instances that can be solved within the given cutoff time. This leads to the following research question:

RQ3 (Chapter 5) To which extent can we predict the running time of a given verification algorithm for a specific problem instance?

So far, we have considered the task of adapting an appropriate neural network verification method to a given problem instance (or set of problem instances), where a problem instance is composed of a neural network and verification property. However, when performing verification of a neural network model (or any other kind of performance assessment), we are typically interested in finding a model that achieves optimal performance. In a verification context, we typically measure robust accuracy. This introduces the model selection problem, which is concerned with selecting the best-performing model from a set of candidates on the basis of a predefined performance criterion. Therefore, we are also interested in efficiently performing robust model selection by leveraging meta-algorithmic approaches. Thus, we arrive at the following research question:

RQ4 (Chapter 6) How can we efficiently select the neural network model from a given set of models that achieves the highest certified robust accuracy?

In summary, this thesis seeks to improve the state of the art in neural network verification systems by leveraging recent advances in meta-algorithmic approaches, such as automated algorithm configuration, portfolio construction, running time prediction and model selection techniques for optimised resource allocation.

1.2 Contributions of this thesis

The core technical content of this thesis has been published in the form of research papers, with each thesis chapter aligning with a specific paper.

In Chapter 3, we investigate the highly diverse landscape of neural network verification algorithms with respect to local robustness properties and present a detailed

overview of current algorithmic approaches. To enable a principled analysis, we define several criteria for defining the state of the art, and perform an empirical performance analysis of selected methods. For this performance analysis, we created a new and diverse benchmark consisting of neural network verification problem instances and divided this benchmark into subcategories based on different neural network activation functions. In addition, we introduce specific measures capturing not only the stand-alone performance of a given verification algorithm but also their performance in relation to others. Using these *complementarity metrics*, we show that no single best algorithm dominates performance across all verification problem instances and illustrate the potential of leveraging algorithm portfolios. Furthermore, we show that some activation functions are highly under-supported by existing verification methods. The research presented in this chapter has given rise to the following research articles:

- Matthias König, Annelot W Bosman, Holger H Hoos, and Jan N van Rijn. Critically Assessing the State of the Art in Neural Network Verification. *Journal of Machine Learning Research*, 25(12):1–53, 2024.

The paper above is an extension of the following workshop paper:

- Matthias König, Annelot W Bosman, Holger H Hoos, and Jan N van Rijn. Critically Assessing the State of the Art in CPU-based Local Robustness Verification. In *Proceedings of the Workshop on Artificial Intelligence Safety 2023 (SafeAI 2023) co-located with the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI2023)*, pages 1–9, 2023. **[Best Paper Award]**

In Chapter 4, we present a concrete approach to leverage algorithm portfolios for neural network verification in combination with automated algorithm configuration. Specifically, we consider neural network verification based on mixed integer linear programming (MIP) encodings, where the verification property is treated as a minimisation problem and solved by a commercial MIP solver. We show that by using automated algorithm configuration and portfolio construction techniques, the performance of a MIP-based verification system can be substantially improved in terms of running time as well as the total number of solved problem instances within a given time budget. The research presented in this chapter has given rise to the following research articles:

- Matthias König, Holger H Hoos, and Jan N van Rijn. Speeding up neural network robustness verification via algorithm configuration and an optimised mixed integer linear programming solver portfolio. *Machine Learning*, 111(12):4565–4584, 2022.

- Matthias König, Holger H Hoos, and Jan N van Rijn. Speeding Up Neural Network Verification via Automated Algorithm Configuration. In *ICLR Workshop on Security and Safety in Machine Learning Systems*, pages 1–4, 2021.

In Chapter 5, we introduce novel features describing instances of the neural network verification problem. These features take into account the given instance as well as internal mechanics of the verification algorithm used. We focus on several state-of-the-art verification algorithms and show that our features enable the reliable prediction of timeouts; *i.e.*, cases in which a specific instance cannot be solved within the given time budget. This prediction is performed by a supervised machine learning model trained on these features. Using this timeout prediction model, we can substantially reduce the computational costs demanded by the verification system via early termination of verification queries that would otherwise result in a timeout. The research presented in this chapter has given rise to the following research article:

- Konstantin Kaulen, Matthias König, and Holger H Hoos. Dynamic algorithm termination for branch-and-bound-based neural network verification. In *To appear in Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI-25)*, pages 1–9, 2025.

Lastly, in Chapter 6, we consider the task of robust model selection. Specifically, this task involves selecting the neural network model from a given set of candidate models that shows the highest degree of adversarial robustness. Towards this end, we propose a racing algorithm that leverages the estimated likelihood of an instance to be robust and prioritises those during the model evaluation procedure. This enables an early elimination of candidate models after verifying only a small number of input instances. We show that our approach reduces the computational burden of selecting the most robust neural network model by up to two orders of magnitude on standard benchmarks from the literature, compared to an exhaustive evaluation (*i.e.*, standard) approach. The research presented in this chapter has given rise to the following research article:

- Matthias König, Holger H Hoos, and Jan N van Rijn. Accelerating Adversarially Robust Model Selection for Deep Neural Networks via Racing. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI-24)*, pages 21267–21275, 2024.

Altogether, the contributions of this thesis enable the scaling of state-of-the-art neural network verification algorithms to problem instances that were previously

unsolved as well as the usage of the these algorithms in a more resource-efficient manner.

1.3 Other Work by the Author

From the following papers, the work presented by König et al. [65] is directly related to the contents of this thesis, as it applies the concept of automated algorithm configuration to linear bounding techniques for incomplete neural network verification. The remaining papers are not directly related.

- Matthias König, Xiyue Zhang, Holger H Hoos, Marta Kwiatkowska, and Jan N van Rijn. Automated Design of Linear Bounding Functions for Sigmoidal Nonlinearities in Neural Networks. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2024.
- Bruno Veloso, Luciano Caroprese, Matthias König, Sónia Teixeira, Giuseppe Manco, Holger H Hoos, and João Gama. Hyper-Parameter Optimization for Latent Spaces in Dynamic Recommender Systems. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 249–264, 2021.
- Matthias König, Holger H Hoos, and Jan N van Rijn. Towards Algorithm-Agnostic Uncertainty Estimation: Predicting Classification Error in an Automated Machine Learning Setting. In *ICML Workshop on Automated Machine Learning*, pages 1–6, 2020.