Distribution-Aware Neuro-Symbolic Verification

Faried Abu Zaid[1], Dennis Diekmann[2], and Daniel Neider[3]

[1] Transferlab, appliedAI Institute, Munich
[2] Carl von Ossietzky University, Oldenburg
[3] Verification and Formal Guarantees of Machine Learning, TU Dortmund University

Abstract. We propose distribution aware neuro-symbolic verification as a way to restrict verification processes to the data distribution (or other distributions). We propose the NICE Laplacianizing flow as suitable density model because of two major properties. First, we show that the log-density function associated to a NICE Laplacianizing flow is piece-wise affine and hence applicable for SMT based approaches using linear arithmetic. Second, each NICE flow maps the upper log-density level sets of the data distribution to the upper log-density level sets of the latent Laplacian, which gives rise to potential applications within interval bound propagation methods.

1 Introduction

Neuro-symbolic verification has recently emerged as a new technique to verify semantic properties of neural networks [6,8,10]. In a nutshell, reference networks are used to express high-level properties which can be addressed as predicates in an otherwise logical specification. Popular ways of performing the verification are the reduction to SMT solving or the use of interval-bound propagation methods. Low-level properties such as adversarial robustness are expressible in the neuro-symbolic framework in the same way as high-level properties such as "a self-driving car will always hold in front of a stop sign". In case of a violation of the property, many verification methods are able to provide concrete counter examples. However, searching for counter examples in the entire feature space might return instances which are not in the support of the data distribution. Xie et al. [10] propose an auto-encoder based method to ensure that counterexamples are contained in the data distribution. We build upon this idea and refine it to yield probabilistically interpretable results. We achieve this by replacing the auto-encoder with a density estimator and adjust the verification task to verify properties only for the upper density level set of a specified probability mass. Since computing upper-level density sets for a given estimator is computationally infeasible in general, we employ a special flow architecture based the non-linear independent component estimator (NICE) by Dinh et al. [2] and show that upper-level density sets of the target distribution have very simple latent representations in these flows, which makes them accessible for SMT based on linear arithmetic and interval propagation based techniques.

We believe that restricting verification tasks to the support of a meaningful input distribution is very important for the verification of real world systems

since in practice areas outside of the support of the input distribution might be meaningless and we might therefore not be interested in the behavior of the model in this area. Additionally, the far tail of a distribution is by nature poorly represented in data, which leads to high epistemic uncertainty about the tail even after seeing the data. We would expect that a model which takes uncertainty because of the lack of data into account produces much less confident predictions in the tail of the distribution, which is an behavior that we might want to verify separately.

While auto-encoder implicitly capture the data distribution through the reconstruction error, they are not trained to align the reconstruction error with the underlying density function. Hence, upper- and lower reconstruction error level sets are not probabilistically interpretable. The natural replacement to solve this issue are upper density level sets $L_D^{\uparrow}(t) = \{x \in \mathbb{R}^d \mid p_D(x) > t\}$, where p_D is the density of the input distribution. In this case we can even bound the failure probability relative to the reference distribution: A successful verification of the property on $L_D^{\uparrow}(t)$ implies a failure probability of at most $1 - p_D(L_D^{\uparrow}(t))$.

2 Applications

2.1 Verification within the Data Distribution

This is our motivating example from the introduction. In a practical scenario, we would like to specify acceptable failure probability p rather than an acceptable log-density threshold. Hence, we propose the following abstract procedure for verification of machine learning models within the center of the data distribution:

1. For a given $p \in [0, 1]$, determine the log-density $\log t_p$ with $p_D(L_D^{\uparrow}(t_p)) = p$.
2. Verify that $\forall x : \log p_D(x) > \log t_p \rightarrow \varphi(x)$

Where φ is the neuro-symbolic property that we want to verify. Note that since we are able to sample from our flow model, estimating $\log t_p$ can easily be done empirically with high accuracy [9,1].

2.2 Verification of Correct Epistemic Uncertainty Quantification

As we argued earlier, for the far tail of the data distribution there are usually no samples available. Hence, any model trained purely from data has never gotten information about these areas. Without any inductive bias, the uncertainty estimates given e.g. by a classifier should converge towards a uniform distribution as we move further outwards in the tail [5,4]. However, it is known that many deep neural network training methods produce badly calibrated networks with overconfident predictions, especially in areas of high epistemic uncertainty [3,7]. Our approach can be used to verify that the network takes epistemic uncertainty into account. E.g. for a binary classifier C:

1. For a given (very small) $p \in [0, 1]$, determine the log-density $\log t_p$ with $p_D(L_D^{\downarrow}(t_p)) = p$.
2. Verify that $\forall x : \log p_D(x) \leq \log t_p \rightarrow C(x) \in \left[\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon\right]$ for a given tolerance ϵ.

References

1. Benoît Cadre, Bruno Pelletier, and Pierre Pudlo. Estimation of density level sets with a given probability content. 25(1):261–272.
2. Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *3rd Int. Conf. Learn. Represent. ICLR 2015 San Diego CA USA May 7-9 2015 Workshop Track Proc.*
3. Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proc. 34th Int. Conf. Mach. Learn.*, pages 1321–1330.
4. Eyke Hüllermeier and Willem Waegeman. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. 110(3):457–506.
5. Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, NIPS'17, pages 5580–5590. Curran Associates Inc.
6. Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher Strong, Clark Barrett, and Mykel J. Kochenderfer. Algorithms for Verifying Deep Neural Networks. 4(3-4):244–404.
7. Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the Calibration of Modern Neural Networks.
8. Mark Niklas Müller, Christopher Brix, Stanley Bak, Changliu Liu, and Taylor T. Johnson. The third international verification of neural networks competition (VNN-COMP 2022): Summary and results. abs/2212.10376.
9. A. B. Tsybakov. On Nonparametric Estimation of Density Level Sets. 25(3):948–969.
10. Xuan Xie, Kristian Kersting, and Daniel Neider. Neuro-Symbolic Verification of Deep Neural Networks. In *Proc. Thirty-First Int. Jt. Conf. Artif. Intell.* International Joint Conferences on Artificial Intelligence.