Neural Networks for Safety-Critical Applications - Challenges, Experiments and Perspectives

Chih-Hong Cheng, Frederik Diehl, Gereon Hinz, Yassine Hamza, Georg Nuehrenberg,
Markus Rickert, Harald Ruess, Michael Truong-Le
fortiss - Landesforschungsinstitut des Freistaats Bayern, Germany

***Abstract*—We propose a methodology for designing dependable Artificial Neural Networks (ANNs) by extending the concepts of understandability, correctness, and validity that are crucial ingredients in existing certification standards. We apply the concept in a concrete case study for designing a highway ANN-based motion predictor to guarantee safety properties such as impossibility for the ego vehicle to suggest moving to the right lane if there exists another vehicle on its right.**

***Index Terms*—autonomous driving, neural network, dependability, certification, formal verification, research challenges**

I. Introduction

The recent burst of applying artificial neural network (ANN) technologies has created an impact on applications such as autonomous driving. Although using ANN-based techniques had shown great promise (e.g., substantially superior image recognition [5]) compared to classical approaches, there have been huge barriers in using neural networks in safety-critical domains (e.g., report from NASA [2]). In this paper, we propose a methodology for enabling the usage of ANNs by considering reasonable extensions for existing safety standards (Sec. II). We examine the technology readiness of our proposed methodology by conducting a case study on highway motion prediction for autonomous driving (Sec. III), and address further research needs (Sec. IV).

II. Certification Considerations for Dependable Neural Networks

For certification of safety-critical systems, safety is established by rigorous *engineering processes*. In safety engineering, the basic principle of (1) ensuring that the specification is correct and (2) ensuring that an implementation satisfies the specification is well perceived. Table I summarizes three critical aspects of the underlying intention of certifying safety-critical systems, namely *specification validity*, *implementation understandability*, and *implementation correctness*.

- The specification validity is important to ensure that "the right system is built". Several methods can be used in this regard, such as prototyping, design-time analysis and reviews, or product acceptance tests.
- The well-behaving of an implementation is captured by two aspects: (1) understandability via requirement-to-code traceability, and (2) correctness via extensive testing, with coverage criteria such as Modified Condition / Decision Coverage (MC/DC).

Although these approaches are valid for classical engineering, e.g., using V-models, applying them on neural networks has created the following issues:

- (*Black-box structure*) For ANN-based systems, implementations consist of layers of neurons operating on and transforming high-dimensional vectors. This makes understandability arguments such as fine-grained requirement-to-code traceability difficult.
- (*Testing for correctness claims*) ANNs often use non-linear piecewise activation functions, e.g., the piecewise-linear ReLU function. Since an invocation of a piecewise function can be interpreted as if-then-else statement and every neuron contains an activation function, MC/DC is intractable, as branching possibilities are exponential in the number of neurons.
- (*Implicit specification*) For implementing systems using ANNs, the specification refers to a combination of data (which specifies input-output behaviors) as well as classical specifications for domain knowledge such as traffic or safety rules. The "specification knowledge" inside the data is *implicit*, compared to cases such as traffic rules.

Based on the above issues, Table I further summarizes our considered additions towards safety certification of ANNs.

(A) (*Neuron-to-feature understandability*) One should provide confidence regarding the meaning of a neural network by associating individual neurons with conditions (features) when they can be activated. This is an adaption of the specification-to-implementation traceability.

(B) (*From testing to formal analysis*) The result of certification should provide (best effort) correctness claims over the (partially incomplete) classical specification, such as obeying traffic rules or ensuring road safety. As testing approaches its limitation, we suggest to apply formal methods such as static analysis or symbolic reasoning.

(C) (*Validating the "implicit specification"*) One needs to check the *validity of the data*, to ensure that only sanitized data will be used in training. E.g., in autonomous driving, one needs to enhance raw data with sure guarantees that the training data for the maneuvering control of the vehicle does not contain risky driving behavior.

III. Case Study: Highway Motion Prediction for Autonomous Vehicles

We outline how we applied the strategy above in verifying a highway overtaking ANN-based motion predictor used in autonomous driving (developed by Lenz et al. [6]). Figure 1 provides a snapshot on the simulation of the vehicle.

The predictor is a feed-forward neural network with piecewise linear activation functions in its hidden layers. It takes

TABLE I

Extending the concept in certify safety-critical systems to new opportunities brought by neural networks.

Implementation understandability	Existing standard	Fine-grained specification-to-code traceability
	Adaptation for ANN	(+) Fine-grained neuron-to-feature traceability
Implementation correctness	Existing standard	Verification based on testing and classical coverage criteria such as MC/DC
	Adaptation for ANN	(−) coverage criteria such as MC/DC (+) formal analysis against safety properties
Specification validity	Existing standard	Validation via prototyping, design-time analysis, validity and product acceptance test
	Adaptation for ANN	(+) Validating data as a new type of specification

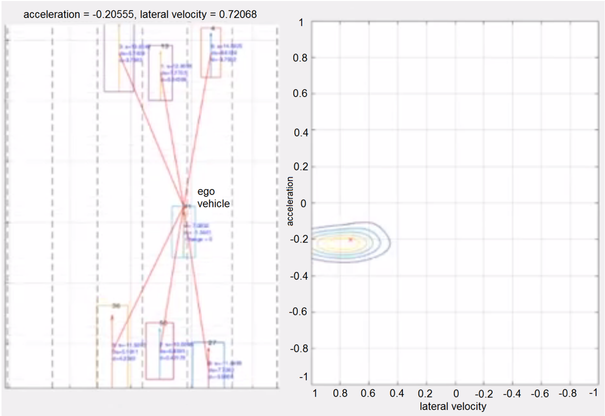

Fig. 1. Simulation of the vehicle (left) and the switch-lane motion suggested by the neural network (right).

three categories of inputs: (i) its own speed profile, (ii) parameters of its nearest surrounding vehicles for each orientation, and (iii) the road condition. The total number of input variables to the network is 84. Given the current state of the perceived environment, it produces in real-time a probability distribution over all possible actions for a vehicle, characterized as a Gaussian mixture model. The action of the ego vehicle is decomposed into two parts: (i) indicator over possible *lateral velocity* (i.e., if it is feasible to switch lanes), and (ii) indicator over *longitudinal acceleration* (i.e., if it is feasible to accelerate). In Figure 1, the motion predictor on the right suggests to slightly decelerate and to switch to left lanes, as the generated Gaussian mixture is within the lower left part.

One of the most critical safety requirements is to ensure that if there is a vehicle in the left of the ego vehicle, the predictor never suggests a large lateral velocity to the left of the ego vehicle, since such a scenario may lead to crashes. In this example, the safety property is that the mean value of the probability distribution should be limited to certain threshold.

We performed formal verification (as in Sec. II (B)) following the methodology developed by Cheng et al. [3], which encodes verification problems as mixed integer linear programs. We were able to successfully verify safety properties on a Google VM with 12 cores. For some ANN-predictors, we validated that the training data never contains problematic inputs (as in Sec. II (C)). The first column of Table II shows various ANN-predictors with four hidden layers and different number of neurons in each layer. The first three ANNs were trained using validated data; others are trained using the original data set. We found that validating the data before training the ANN may play a role: Out of 1.2 million data points, only 32 were problematic. Dropping these 32 data points has drastically changed the behavior of the ANN, where the maximum possible lateral velocity, when a left vehicle exists, increases from 0.273333 m/s (safe for case $I^{\text{validated}}_{4\times25}$) to 2.10916 m/s (unsafe for case $I^{\text{original}}_{4\times25}$). However, only relying on validating data is not sufficient, as can be observed by the case $I^{\text{validated}}_{4\times40}$, which may still demonstrate risk behavior even when being trained with validated data.

TABLE II

Results of verifying ANN-based motion predictors.

ANN	maximum lateral velocity, when exists a vehicle in the left	verification time
$I^{\text{validated}}_{4\times25}$	0.273333	1089.0s
$I^{\text{validated}}_{4\times40}$	2.06022	3611.7s
$I^{\text{validated}}_{4\times50}$	1.41809	18324.2s
$I^{\text{original}}_{4\times10}$	0.688497	5.4s
$I^{\text{original}}_{4\times20}$	0.467385	549.1s
$I^{\text{original}}_{4\times25}$	2.10916	28.2s
$I^{\text{original}}_{4\times40}$	1.95859	645.9s
$I^{\text{original}}_{4\times50}$	1.72781	13351.2s
$I^{\text{original}}_{4\times60}$	`n.a.` (unable to find maximum)	`time-out`
$I^{\text{original}}_{4\times60}$	Prove that the lateral velocity can never be larger than 3 m/s	11059.8s

IV. Concluding Remarks

The proposed certification methodology, during the case study, has also indicated further research needs.

(i) During the feasibility study, we found that implementation understandability can only be partially achieved by technologies such as deconvolution [7].

(ii) Scalability of automated verification requires improvement. Recent results on quantized neural networks [4] might make verification more scalable via an encoding to bitvector theories in SMT.

(iii) Apart from verification, another important direction is to consider training under known properties on the target function (known as *hints* [1]), such as safety rules.

References

[1] Y. S. Abu-Mostafa. Hints. Neural Computation 7(4), 1995.

[2] S. Bhattacharyya, D. Cofer, D Musliner, J. Mueller, and E. Engstrom. Certification considerations for adaptive systems. In: *ICUAS*, IEEE, 2015.

[3] C.-H. Cheng, G. Nührenberg, and H. Rueß. Maximum resilience of artificial neural networks. In *ATVA*, Springer, 2017.

[4] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. In: *arXiv:1609.07061*, 2016.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[6] D. Lenz, F. Diehl, M. Troung Le, and A. Knoll. Deep neural networks for markovian interactive scene prediction in highway scenarios. In: *IV*. IEEE, 2017.

[7] M. Zeiler, G. W Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, IEEE, 2011.