https://doi.org/10.20378/irb-47277

Concept Enforcement and Modularization for the ISO 26262 Safety Case of Neural Networks

Gesina Schwalbe and Ute Schmid

Cognitive Systems, University of Bamberg, Bamberg, Germany
{forename.lastname}@uni-bamberg.de

Abstract. The ability to formulate formally verifiable requirements is crucial for the safety verification of software units in the automotive industries. However, it is very restricted for complex perception tasks involving deep neural networks (DNNs) due to their black-box character. For a solution we propose to identify or enforce human interpretable concepts as intermediate output of the DNN. Two effects are expected: Requirements can be formulated using these concepts. And the DNN is modularized, thus reduces complexity and therefore easing a safety case. A research project proposal for a PhD thesis is sketched in the following.

Keywords: ISO 26262, neural networks, formal verification, concept enforcement

1 Introduction

The project aims for a safety certification of computer vision (CV) components solved with convolutional neural networks (CNNs) in autonomous driving (AD) applications using post-training methods. The goals of this PhD project are:

- Finding a systematic way to formulate safety requirements for CNNs for CV tasks without major performance losses.
- Giving a method to formally proof that the DNN fulfills these requirements.
- Assessing the contribution of the suggested requirement formulation and corresponding verification methods for a safety case.

2 Problems in the Automotive Safety Certification of Neural Networks

The analysis in [14] reveals that most method proposals from the ISO 26262 automotive functional safety standard [1] are also applicable to units based on a CNN model. However, they are not sufficient to assure safety: In complex tasks and input spaces typical for AD perception, pure *testing* is considered neither practical nor sufficient [15], also due to frequent robustness problems. This needs to be supplemented by formal verification, (manual) inspection, and implementation measures. The incomplete specification using examples is not suitable for

formal verification. There is also no expert knowledge about the algorithm to remedy this, because of the automated modeling and the black-box character. These latter properties, lastly, also invalidate manual *in(tro)spection* methods and *implementation measures*. Hence, for a legally and societally acceptable safety argumentation of CNNs in safety critical automotive applications, the following is needed: a considerable amount of representative safety relevant test cases, robustness measures, and a way to open or simplify the black-box for formal verification and introspection. The last aspect is the point of interest for the proposed PhD thesis topic.

3 Previous Work

One way to simplify a CNN model is topologically, e.g. via pruning of connections or filters [2]. Another is the transformation into interpretable models via rule extraction, either globally, e.g. DeepRED [9], or locally, e.g. LIME-Aleph [12]. Unfortunately, the remaining connections respectively the extracted set of rules operate on pixel level for CV tasks. The resulting model can get large and hardly human interpretable, since intermediate semantic concepts are missing.

Such concepts encoded by the internal representation of a CNN can be found and realized as additional output in two ways. One is to guess concepts represented by a neuron or neuron cluster by means of feature visualization and attribution techniques [11]. The other is to preselect concepts with accordingly labeled data, and to use network dissection to find the neuron(s)/convolutional filter(s) [4], respectively the neuron/filter cluster(s) [7] corresponding to each preselected concept. Or, similarly, identify layers with the best feature representation for a given task [8]. All transfer learn the additional output by attaching one to two trainable dense layers to one DNN layer. An alternative to a post-training analysis is topological enforcement of desirable concepts as intermediate output such as done in ReNN [17].

Once the new outputs are available, these can be used to formulate, check, and enforce formally verifiable rules using solvers [5], as done in ReNN and [13]. Both examples and [10] suggest a positive impact on the performance when including task specific concepts into the internal representation of a CNN, especially when using ones formulated in natural language [3]. As is suggested by the above, there exists a base for concept extraction from CNNs, as well as for formal verification of relations between neuron outputs. To our knowledge, there is no holistic approach to both of them in the context of post-training CNN modularization and automotive formal safety verification.

4 Approach via Concept Extraction

Building on the Net2Vec approach [7], we establish the following understanding of a concept-to-neuron correspondence: The layer of a NN, or any other collection of neurons, spans the vector space of the corresponding neuron outputs, here called the *space of abstract features* of the layer or neuron cluster. Consider a

neuron cluster, an abstract feature vector w of that cluster, the projection p_w to the one dimensional sub-vector space spanned by x, and a semantic concept c. The vector w is said to correspond to c or to be a *concept embedding* of c if the intermediate output of the NN obtained by concatenation of the layer with p_w has a high correlation with the existence of c (in a certain spatial location). This essentially means, w is a mask emphasizing neurons that together predict c well. Note that previous literature was restricted to feature spaces consisting of neuron outputs of exactly one layer, not general clusters. The case of unit vector correspondences (i.e. single neurons) was investigated by [4], that of general spatial clusters within a layer by [7]. The definition proves to be natural, since similar concepts correspond to vectors with small distance, and vector operations yield meaningful relations on the concepts [7].

Our approach pursues the following workflow: Consider a CV task and a corresponding dataset for supervised learning, and additionally a similar (possibly the same) dataset densely labeled with concepts relevant to task. Examples would be the segmented and image level visual concepts in the BRODEN dataset [4]. Assume, a trained CNN is given, and consider a cross-section of it, i.e. a collection of neurons from which the output of one layer can be reconstructed. This can simply be one layer as in [8]. Consider one of the selected concepts. Represent it by an output neuron which is attached to the CNN by connecting it to each neuron of the cross-section. Train and prune the new connection weights on the concept data. This yields the weighted combination of cross-section neurons that correspond to the concept, the concept vector. By iterating different cross-sections and optimizing the loss, the best concept vector representing each concept can be found. For a start improve on the following loss: For segmented concepts and a cross-section consisting of complete filters, the loss is the one indicated in [7] which is the intersection over union of the weighted sum of the thresholded activation maps. For image level concepts (e.g. scenes), use the cross entropy loss as indicated in [8]. If the prediction results of the intermediate concept output are not satisfying, concept enforcement shall be evaluated, i.e. the effect of retraining the network with the additional loss. On the interpretable intermediate output of the DNN, formal analysis is applicable, as well as rule extraction with possibly more human interpretable outcome. The concept vectors might also mark the interfaces of mostly independent modules within the network at which the DNN can be split.

5 Outlook: Empirical Evaluation Setup

Within the project the above approach shall be empirically evaluated. The first simple experiment setup is a traffic sign recognition task on the German Traffic Sign Recognition dataset [16]. It is expected that the image level concepts of digits which are found in the speed limit signs have corresponding feature vectors within any basic convolutional NN trained on the task (see Figure 1). This can be evaluated using a digit dataset like MNIST. Future work will be located in the setting of of pedestrian detection, for which expected concepts are e.g.

Fig. 1. Examples of speed limit signs from the German Traffic Sign Recognition dataset [16, Fig. 3]; encoded concepts which a recongnition algorithm must be capable to distinguish are the digits 0 to 8.

segmentations of body parts as found in the Pascal Parts dataset, part of the BRODEN dataset [4]. An example can be found in Figure 2.

Fig. 2. Examples of concepts labeled in the Pascal Part dataset [6, Fig. 4, p. 6]

Bibliography

[1] 32, I.S.: Road Vehicles — Functional Safety — Part 1: Vocabulary, vol. 1, 2 edn. (2018)

[2] Abbasi-Asl, R., Yu, B.: Interpreting convolutional neural networks through compression. CoRR **abs/1711.02329** (2017)

[3] Andreas, J., Klein, D., Levine, S.: Learning with latent language. In: Proc. Conf. North Amer. Chapter of the Assoc. for Computational Linguistics: Human Language Technologies, vol. 1, pp. 2166–2179 (2018)

[4] Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, pp. 3319–3327 (2017)

[5] Bunel, R.R., Turkaslan, I., Torr, P., Kohli, P., Mudigonda, P.K.: A unified view of piecewise linear neural network verification. In: Advances in Neural Information Processing Systems 31, pp. 4790–4799 (2018)

[6] Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.L.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, pp. 1979–1986 (2014)

[7] Fong, R., Vedaldi, A.: Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, pp. 8730–8738 (2018)

[8] Fuchs, F.B., Groth, O., Kosiorek, A.R., Bewley, A., Wulfmeier, M., Vedaldi, A., Posner, I.: Neural Stethoscopes: Unifying analytic, auxiliary and adversarial network probing. CoRR **abs/1806.05502** (2018)

[9] Hailesilassie, T.: Rule extraction algorithm for deep neural networks: A review. CoRR **abs/1610.05267** (2016)

[10] Kim, J., Rohrbach, A., Darrell, T., Canny, J.F., Akata, Z.: Textual explanations for self-driving vehicles. In: Proc. 15th European Conf. Computer Vision, Part II, vol. 11206, pp. 577–593 (2018)

[11] Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill **2**(11), e7 (2017)

[12] Rabold, J., Siebers, M., Schmid, U.: Explaining black-box classifiers with ILP – empowering LIME with Aleph to approximate non-linear decisions with relational rules. In: Proc. Int. Conf. Inductive Logic Programming, pp. 105–117 (2018)

[13] Roychowdhury, S., Diligenti, M., Gori, M.: Image classification using deep learning and prior knowledge. In: Workshops of the 32nd AAAI Conf. Artificial Intelligence, vol. WS-18, pp. 336–343 (2018)

[14] Salay, R., Queiroz, R., Czarnecki, K.: An analysis of ISO 26262: Using machine learning safely in automotive software. CoRR **abs/1709.02435** (2017)

[15] Shalev-Shwartz, S., Shammah, S., Shashua, A.: On a formal model of safe and scalable self-driving cars. CoRR **abs/1708.06374** (2017)

[16] Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The German Traffic Sign Recognition benchmark: A multi-class classification competition. In: Proc. Int. Joint Conf. Neural Networks, pp. 1453–1460 (2011)
[17] Wang, H.: ReNN: Rule-embedded neural networks. In: Proc. 24th Int. Conf. Pattern Recognition, pp. 824–829 (2018)