Towards formal methods and software engineering for deep learning: Security, safety and productivity for dl systems development

Gaétan Hains, Arvid Jakobsson, Youry Khmelevsky

IEEE SYSCON2018, Vancouver, Canada 2018-04

Towards formal methods and software engineering for Deep Learning

Security, safety and productivity for DL systems development

Gaétan Hains and Arvid Jakobsson
Huawei Parallel and Distributed Algorithms Lab.
Huawei Paris Research Center
Boulogne-Billancourt, France

Youry Khmelevsky
Okanagan College Computer Science Department
and U.B.C. Okanagan
Okanagan (B.C.) Canada

***Abstract*— Deep Learning (DL) techniques are now widespread and being integrated into many important systems. Their classification and recognition abilities ensure their relevance for multiple application domains far beyond pure signal processing. As a machine-learning technique that relies on training instead of explicit algorithm programming they offer a high degree of productivity. But recent research has shown that they can be vulnerable to attacks and the verification of their correctness is only just emerging as a scientific and engineering possibility. Moreover DL tools are not integrated into classical software engineering so software tools to specify, modify and verify them would make them even more mainstream as software-hardware systems. This paper surveys recent work and proposes research directions and methodologies for this purpose.**

Keywords*— *deep-learning systems, neural networks, vulnerability of deep learning, security, verification, software engineering.

I. INTRODUCTION

As research unit of a leading vendor of information and communication systems, Huawei's Central Software Institute (CSI) is developing high-performance deep learning (DL) systems for image classification [18] and other image recognition functions. As the application domain of self-driving cars [14] highlights it, correct operation (safety) and attack resistance (security) of DL systems is an absolute necessity. Moreover, the very advantage of a neural network (NN) over explicitly programmed algorithms makes their maintenance, modularization and general maintenance less well understood than for general software: how can one specify precisely its behavior, how can layers from two NN be combined into a new one with new functionality, etc. Specification, verification and security of DL is a relatively novel area so new knowledge and new techniques are being actively developed in this direction. This paper surveys existing work and proposes specific research directions to improve the general safety, security of NN while improving the human productivity of their developers.

The next sections survey existing work on

- Attacks against DL systems
- Testing, training and monitoring DL systems for safety
- The verification of DL systems

Then we propose new work in the directions of specification, verification and more generally software engineering for DL systems.

II. SECURITY: ATTACKS AND THEIR PREVENTION

An *adversarial example* for a NN classifier is a slightly perturbed input that generates a different, hence wrong, classification from the desired one. In recent years many have been identified and specific solutions designed for each one. But the general problem remains of formally guaranteeing in advance the absence of adversarial example.

Carlini et al.'s paper [3] is motivated in this manner by safety-security (absence of accidental or intentional adversarial examples) and the need to verify it. They introduce the notion of a *ground truth*, or adversarial example with minimal change in input value. This is useful for two things: judging the quality of an attack by comparing it to the ground truth, and judging the quality of a defense by the amount it increases the distortion in the new ground truth.

The authors of [16] present and articulate technical arguments that appear to show that intentional adversarial examples can be countered, in the area of image processing, by a kind of "multi-sensor" approach. Like attacks on face recognition can be countered by 3D or multiple-angle 2D images, adversarial examples would become ineffective in the presence of multiple-angle or time-sequenced images of the same object(s).

III. TESTING, TRAINING AND MONITORING FOR SAFETY

Concrete progress has been made by authors who propose to adapt training and testing with specific safety-conscious properties and techniques.

The survey paper by B. Taylor et al. [22] takes a very general human-level definition of AI safety. It defines eight very challenging wanted properties of machine learning systems like NN but most of them relate to the human

application of DL systems so, in our opinion, they are premature to consider before the *science and engineering* of DL becomes more mature. One of their eight properties is more amenable to purely technical developments "inductive ambiguity identification" with special case "active learning". An active learner can interact with humans during its leaning phase so as to ask them for additional data (e.g. images) that would break some automatically detected ambiguity in classification. Active learning can thus be considered a design goal for improving the safety of DL systems.

The authors of [24] consider the application of an (unrelated) automatic testing tool called DeepTest to self-driving cars. It can be considered an elementary but meaningful tool for structured testing. As such it has the advantages and limitations of testing methods: easy to design and implement, incomplete by design.

Leofante, Pulina and Tacchella [15] present recent work in the definition and verification of machine-learning safety, namely the guarantee that the input-output function defined by a trained system will behave "according to specification". They also quote model-checking results for verifying this property, its computational costs but do not detail the methodology for doing this. Their notion of *global correctness* is based on *stability*: limited input sample variations lead to limited output variations. This is a well-defined and apparently verifiable type of specification, but it does open two related and deep questions: how can designers be certain that their reference datasets are in some sense correct and complete? How to choose the metric that measures the input or output variations? The notion of active learning, presented in [22] could lead to a practical solution to the first question. But the general problem of global correctness certainly needs more powerful mathematical tools than stability theory: NNs must interact with general algorithms, if only for such operations as sorting results, and the whole system correct and complete specification is thus a classical pre-condition, post-condition pair of local expressions on the system state. In the (very common) application of area of image processing NN-specific predicates could specify that image recognition is, for example rotation invariant. To the best of our knowledge this problem of mixing signal-processing with software specification is unexpolored. Stability predicates would then be an important but incomplete tool to ensure system correctness.

Wicker, Huang and Kwiatkowska [26] present a sophisticated approach that allows black-box testing of NNs i.e. with consideration of features being detected but ignorance of the NN's structure. They search a game space where an agent adversary attempts to use normally/fool/randomly use the detection of features. The method is considered competitive with white-box methods.

Yerramalla, Mladenovski and Fuller [28] applies continuous control theory to design a monitor for ensuring that "unstable" learning can be detected. Their notion of stability is specific to an application where a fixed dataset of images is replaced by an airplane's onboard NN that is trained dynamically through in-flight cameras. This work can be considered as mathematical support for dynamically generated datasets, or abstractly: dynamically generated specifications for the DL system.

But again, testing is by design an incomplete approach and the "specification" of a DL system relies on the experimental definition of its training dataset.

IV. VERIFICATION AND SIMULATION

Other authors have investigated formal and even automatic methods for safety verification. This line of research has been accelerating in recent years.

Broderick [1] uses simulation in the area of flight on-board online-learning NNs. It does not take a formal approach to verification but applies statistical techniques. The white paper [25] defines high-level requirements for "formal" (mathematically-based) verification of similar systems from the point of view of control theory.

Fuller, Yerramalla and Cukic [8] model the learning of a NN as a dynamical system where training adjustments are discrete differential equations on the states that are neurons and weights. Lyapunov stability analysis is then applicable to detect stable states in the dynamical system. Stability in this theory thus amounts to the absence of adversarial examples. It is shown how to apply this concept to (shallow) NNs of fixed topology and also to dynamic ones.

Survey paper [2] compares methods for verifying NNs with piecewise linear structures. It compares methods based on SMT solvers, mixed integer programming and a new branch-and-bound method. The tools are able to verify 100-500 properties for networks for 2-6 layers. Correctness is defined as a form of stability and verification, in theory exhaustive testing, is accelerated by assuming piecewise-linear state spaces.

Katz et al. [10, 12] describe SMT-based work on describing safety properties of systems using *simplified activation functions (ReLU)* as linear functions, and finding solutions with a modified simplex algorithm. This approach checks domain specific safety specifications expressed as SMT formulas. Using SMT with specialized theory for handling "Rectified Linear Units" activation functions. Domain specific safety specification must be found manually. Scalability is a concern for this technique.

Cheng, Nührenberg and Ruess [4] verify DNNs by translating non-linear (input-output) constraints generated by ReLU activation functions using big-M encoding. Then standard techniques for linear optimization are applied to verification.

In [6], an optimization technique is proposed to accelerate verification problems that are difficult for SMT and ILP solvers. It assumes so-called *feed-forward* NNs that allow the addition of a global linear approximation of the overall network behavior.

Blog entry [9] is a general discussion of the importance of safety for DL systems, with arguments in favor of formal verification as opposed to testing.

Huang et al. [11] present work on verifying the absence of adversarial inputs in Feed-forward multi-layer neural networks:

inputs which deceive the network. The paper contains many convincing examples of such perturbed images. The verification method finds adversarial inputs, if they exist, for a given region and a family of manipulations. The technique is based on a transformation to an SMT solver.

Katz et al. published in [13] their efforts to prove adversarial robustness of NNs, i.e. the absence of misclassification due to small perturbations. They propose a new notion of "global robustness" quantifying the robustness of a DNN. Intuitively, a network is globally robust if any two neighbors in the input are also neighbors in the output. Robustness is thus a non-limit form of continuity as in:

$$d_1(x, y) \leq \delta \rightarrow d_2(NN(x), NN(y)) \leq \varepsilon$$

where NN is the neural net's inference function, d_1 is a standard metric in the input domain, d_2 a suitable metric in the output domain and δ, ε are experimentally chosen error bounds where ε could be zero, e.g. if the output is a discrete space of features. They then show how to encode this property and verify it using Reluplex. However, it is challenging to verify, and the result only extends to DNN with a few dozen nodes.

Narodytask et al. [17] present the first exact Boolean representation of a deep NN so that a binarized network is faithfully represented as a Boolean formula. They are then able to leverage the high efficiency of modern SAT solvers for the formal and automatic verification of the NNs behavior, in particular resistance to adversarial perturbations.

Pulina and Tacchella [19] present CETAR: a Counter-Example Triggered Abstract Refinement verification approach for DNNs. Performance is not demonstrated on large NNs (only 20 nodes are used).

Paper [20] by the same authors describes and evaluates the tool NeVeR that verifies the safety of ANNs by encoding them as SMT-formula with linear inequalities. Furthermore, to improve scalability, the authors apply the abstraction refinement scheme presented in their earlier work.

Xiang, Tran and Johnson [27] present a verification method for multi-layer NNs and apply it to robotics. Their simulation-based method for the estimation of the output set of a NN, is applicable to networks with monotone activation functions. The verification problem is formulated and solved as a chain of optimization problems for estimating a reachable set of states.

Dutta et al. [5] study the automatic estimation of the output-range for deep NNs. A key concept of theirs is that sets of possible inputs are compactly represented by convex polyhedral. They compute the guaranteed output range for DNNs by successive optimizations.

V. SPECIFICATION AND FUTURE SOFTWARE TOOLS

The above set of research results indicate a strong convergence towards automatic and formally-based methods for verifying the input-output behavior of DL systems. But a serious problem appears to remain in balancing the guarantees of exhaustive search as in model checking with reasonable compute times. This situation is familiar to users of linear solvers and indeed several authors use linear equations and solvers to tackle DL safety problems. But the intrinsic combinatorial nature of the problem is a serious scalability hindrance.

J. Taylor et al.'s paper [23] discusses in a very high-level way the problem of specifying the behavior of a machine-learning system for example through the objective function of its training phase. It covers an interesting set of research targets one of whom has specific meaning for specification of DL system behavior. *Inductive ambiguity identification* is defined as the goal of creating systems that can detect inputs for which their inference or classification would be highly under-determined by training data. Future safety-verification methods should address this problem that is akin to the need for attaching confidence levels to DL-system outputs.

Foerster et al. [7] present a very innovative approach where the NNs come from a specific sub-family: without nonlinearities or input-dependent recurrent weights. For this family the linear representation of input-output behavior is not an approximation but an exact encoding. As a result verification can benefit from fast linear-algebra operations. The balance between this restricted family of NNs and their expressive power is illustrated on a very-large NLP example. This approach could either become a breakthrough or a less-significant approach for niche applications. But the general idea of a compact and efficiently-processed specification is probably a core element of future theories and tools.

The white paper by Russel, Dewey and Tegmark [21] reasserts, among many other things, that formal verification and security and absolute necessities for all AI systems. They propose that AI systems (among them DL systems) should allow the verification of their behavior, of their designs (in particular their specification), allow how to distinguish their software-hardware components, and also the modular verification of their parts.

In view of the existing research it appears that more work is necessary in the direction of automatic or semi-automatic formal verification of NN behavior, applied to DL systems. As we have seen, today's specification of their behavior is very similar to what hardware verification faces: enumerated sets of input-outputs, and a low-level definition of distance between inputs or outputs, as if it were uniquely a matter of numerical precision. But small input perturbations may have deep meanings for example in images. Moreover the accepted / adversarial perturbations may not be compact in the sense of topology: consider for example a mixed-color table whose mixture of pixel colors indicates that the fabric of the table is made of two different chemicals. A purely local Euclidian notion of distance would not express correct detection of such a feature. So, without re-considering the existing scientific basis for specification, it appears necessary to let designers and implementers write higher-level specifications from which the low-level ones can be generated and systematically

verified. A defined advantage of such concepts and tools would also be that theorem proving could be applied under certain conditions, eliminating the need for exhaustive solvers and their lack of scalability for today's very large NNs.

We therefore propose new research sub-direction as follows:

- Domain-specific languages (DSL) for high-level description and manipulation of the input-output specification. For examples grammar-based DSLs for NLP applications, DSLs with discrete-geometry semantics for images where features are geometric etc.
- A DSL sub-language defining the distance function that is the basis for defining perturbations.
- Tools that translate those DSLs into low-level specifications for given datasets, including tools to compare datasets, analyze them for their distance-function statistics etc.
- UML class diagrams for representing datasets, others for replacing the DSLs in industrial applications.
- Visual tools that let application-domain experts interact easily but precisely with the specifications.
- Model-based testing tools based on the above high-level techniques. Such techniques are already being applied to extensional descriptions (the neural net itself) [10,12] but they would become more scalable and efficient if a format resembling source-code would describe the NN e.g. with parallel loops and indices for repeated weights and neurons.
- Theorem-proving techniques are still far in the future because they require (a) clear and expressive logical specifications and (b) a source-code like format for the NN as hypothesized above.

VI. CONCLUSIONS

Safety of DL systems is a serious requirement for real-life systems and the research community is addressing this need with mathematically-sound but low-level methods of high computational complexity. To turn DL system design into a broad industry, methods inspired by software engineering must be applied to complement and sometimes replace the low-level ones as for theorem proving replacing model checking. Our survey of the area has shown the acceleration of the line of work, the general agreement for its mathematical and low-level methods. We have also shown how other recent surveys point to the need for more expressive specifications, a scope for symbolic verification and generally more designer productivity. We have made early but clear and feasible proposals for new research in this direction.

ACKNOWLEDGMENT

Arvid Jakobsson is supported by a CIFRE industrial PhD contract between Huawei Technologies France and LIFO, Université d'Orléans, funded in part by A.N.R.T. The authors thank the reviewers for many relevant and/or deep questions and corrections.

REFERENCES

[1] R. L. Broderick, "Adaptive verification for an on-line learning neural-based flight control system," in *24th Digital Avionics Systems Conference*, vol. 1, C.2–61–10 Vol. 1, 2005, pp. 6-15.

[2] R. Bunel, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar, Piecewise Linear Neural Network verification: A comparative study, *arXiv preprint* arXiv:1711.00455, 2017 [Online] Available: https://arxiv.org

[3] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, Ground-Truth Adversarial Examples, [Online] *arXiv preprint* arXiv:1709.10207, 2017 [Online] Available: https://arxiv.org

[4] C.-H. Cheng, G. Nührenberg, and H. Ruess, Verification of binarized neural networks, *arXiv preprint* arXiv:1710.03107, 2017 [Online] Available: https://arxiv.org

[5] S. Dutta, S. Jha, S. Sanakaranarayanan, and A. Tiwari, Output Range Analysis for Deep Neural Networks, *arXiv preprint* arXiv:1709.09130, 2017 [Online] Available: https://arxiv.org

[6] R. Ehlers, Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks, *arXiv preprint* arXiv:1705.01320, 2017 [Online] Available: https://arxiv.org

[7] J. N. Foerster, J. Gilmer, J. Sohl-Dickstein, J. Chorowski, and D. Sussillo, "Input Switched Affine Networks: An RNN Architecture Designed for Interpretability," in *International Conference on Machine Learning*, 2017, pp. 1136–1145.

[8] E. J. Fuller, S. K. Yerramalla, and B. Cukic, "Stability Properties of Neural Networks," in *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*, Springer, Boston, MA, 2006, pp. 97–108.

[9] I. Goodfellow and N. Papernot, The challenge of verification and testing of machine learning, *Cleverhans-blog,* 2017 [Online] Available: http://www.cleverhans.io/security/privacy/ml/2017/06/14/verification.html

[10] D. Gopinath, G. Katz, C. S. Pasareanu, and C. Barrett, Deepsafe: A data-driven approach for checking adversarial robustness in neural networks, *arXiv preprint* arXiv:1710.00486, 2017 [Online] Available: https://arxiv.org

[11] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, Safety Verification of Deep Neural Networks, *arXiv preprint* arXiv:1610.06940 [cs, stat] Oct. 2016 [Online] Available: https://arxiv.org

[12] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, *arXiv preprint* arXiv:1702.01135, 2017 [Online] Available: https://arxiv.org

[13] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, Towards proving the adversarial robustness of deep neural networks, *arXiv preprint* arXiv:1709.02802, 2017 [Online] Available: https://arxiv.org

[14] B. Kisačanin, "Deep Learning for Autonomous Vehicles," *2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL)*, Novi Sad, 2017, pp. 142-142.

[15] F. Leofante, L. Pulina, and A. Tacchella, "Learning with Safety Requirements: State of the Art and Open Questions.," in *RCRA@ AI* IA*, pp. 11–25, 2016.

[16] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles, *arXiv preprint* arXiv:1707.03501 [cs], Jul. 2017 [Online] Available: https://arxiv.org

[17] N. Narodytska, S. P. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, Verifying properties of binarized deep neural networks, *arXiv preprint* arXiv:1709.06662, 2017 [Online] Available: https://arxiv.org

[18] S. B. Park, J. W. Lee and S. K. Kim, "Content-based image classification using a neural network.", *Pattern Recognition Letters*, Elsevier, vol. 25, pp. 287-300, 2004.

[19] L. Pulina and A. Tacchella, "An Abstraction-Refinement Approach to Verification of Artificial Neural Networks," *International Conference on Computer Aided Verification*, 2010, pp. 243–257.

[20] L. Pulina and A. Tacchella, "NeVer: a tool for artificial neural networks verification," *Annals of Mathematics and Artificial Intelligence*, vol. 62, no. 3–4, pp. 403–425, 2011.

[21] S. Russell, D. Dewey, and M. Tegmark, "Research Priorities for Robust and Beneficial Artificial Intelligence," *AI Magazine*, vol. 36, no. 4, pp. 105–114, Dec. 2015.

[22] B. Taylor and M. Darrah and C. Moats, "Verification and validation of neural networks: a sampling of research in progress", *Intelligent Computing: Theory and Applications, Proceedings SPIE*, Priddy & Angeline Eds., Vol. 5103, 2003.

[23] J. Taylor, E. Yudkowsky, P. LaVictoire and A. Critch, Alignment for advanced machine learning systems, *Machine Intelligence Research Institute*, 2016. [Online] Available: https://pdfs.semanticscholar.org/7ac7/b6dbcf5107c7ad0ce29161f60c2834a06795.pdf

[24] Y. Tian, K. Pei, S. Jana, and B. Ray, DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, *arXiv preprint* arXiv:1708.08559, 2017 [Online] Available: https://arxiv.org

[25] P. Van Wesel, Challenges in the Verification of Reinforcement Learning Algorithms, *Nasa Langley Research Center*, NASA/TM{2017{219628, June 2017 [Online] Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170007190.pdf

[26] M. Wicker, X. Huang, and M. Kwiatkowska, Feature-Guided Black-Box Safety Testing of Deep Neural Networks, *arXiv preprint* arXiv:1710.07859, 2017 [Online] Available: https://arxiv.org

[27] W. Xiang, H.-D. Tran, and T. T. Johnson, Output reachable set estimation and verification for multi-layer neural networks, *arXiv preprint* arXiv:1708.03322, 2017 [Online] Available: https://arxiv.org

[28] S. Yerramalla, Stability Monitoring and Analysis of Online Learning Neural Networks, Doctoral Dissertation, West Virginia University, Morgantown, WV, USA, 2005.