ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Proposal for a Master's thesis

Comparing and enhancing deep-learning models with temporal logics

Supervisors: Joshua Schneider and Martín Ochoa
Professor: Prof. David Basin

Prerequisites

- Strong skills in formal methods.
- Some familiarity with Deep Learning.

Background

In recent years, many breakthroughs in artificial intelligence have been achieved through the use of deep neural networks. Several previously challenging tasks, such as defeating the Go champion or translating natural language have been made possible by machine learning techniques. However, it is also known that machine learning techniques are not well suited for all artificial intelligence tasks. For instance, it has been argued that deep neural networks require an excessive amount of training data, does not generalize well, and struggles with integrating existing knowledge [3]. Symbolic reasoning, on the other hand, can have an advantage in situations that benefit from abstraction and logical inference. Moreover, deep neural networks are complex and it is difficult to explain their inner workings, whereas logical formulas maybe more intuitive for humans in some scenarios.

Many of the problems that have been successfully solved using deep learning techniques are so-called *classification* problems: given a training set of samples belonging to two classes A and B, can we successfully classify previously unseen samples? For instance, can we detect future intrusions into a system, given a set of system logs that are marked as being the result of prior attacks, and a set of logs that represent normal system executions?

If the patterns that constitute an intrusion are well understood, we can encode them in logical formulas, and view this problem as an instance of the *monitoring problem* [4]: Given a stream of events and a property formulated in a specification language, check whether the property is satisfied at every position in the stream. For example, consider a property that guards against a specific kind of attack on a webserver: "the webserver process must not write to any file that is later used to spawn a new process." The idea is that a violation of this property could indicate an attacker successfully exploiting a vulnerability in the server and injecting a payload into an executable file. Already this simple property requires an expressive specification language. Specifically, it requires quantification over objects ("any file") and Boolean ("must not") and temporal connectives ("later"). Linear temporal logic (LTL) provides these features except for quantification, which in this case can be handled by trace slicing [7]. Various extensions of LTL, such as MTL and MFOTL, further increase the expressiveness.

What can be done if the monitoring conditions are not known a priori? Can we learn such logical formulas from examples, and if so, how does this process and the resulting classification accuracy compare to deep learning techniques?

For LTL, learning algorithms with strong guarantees have been developed [2, 5]. Given finite sets of positive and negative examples (which are finite event streams), they find a formula of minimal size that is consistent with *all* examples. These algorithms work by constructing a propositional formula ϕ_n that tests the results of all LTL formulas of size n on the examples, starting with $n = 1$. Then they invoke an off-the-shelf SAT solver to find a satisfying assignment for ϕ_n, from which the desired LTL formula is reconstructed. If ϕ_n is unsatisfiable, the algorithm restarts with size $n + 1$. Reference [5] combines this approach with decision tree learning to improve the performance (at the cost of losing minimality).

The approach described above results in concise models and it works well even if only few examples are available. However, it is difficult to generalize the reduction to SAT to more expressive logics in a scalable manner. It is also not resilient to noisy training data, which is less of a concern for machine learning techniques.

Objective

The objective of this thesis is to compare and enhance deep learning models with temporal logics. For this, it is necessary to understand work in learning LTL formulas and apply such work to case studies suggested in the literature and novel scenarios that involve classification of event streams. Then, compare the performance of obtained formulas against a solution involving neural networks. Finally suggest ways in which the two techniques could be use complementary to enhance either classification accuracy or interpretability.

Tasks

This project can be subdivided into the following tasks:

1. Familiarize yourself with the work in [2, 5] and choose one of the two LTL learning algorithms.
2. Reproduce the behavior classification case study from [2] using the chosen algorithm.
3. Apply the LTL learning technique in a novel classification scenario. One possibility is to infer sequential API contracts [6] using open data sets, e.g. [8].
4. Train ML models (e.g., LSTM neural networks as in the evaluation of [2]) for each of the two scenarios and compare their classification accuracy with the learned LTL formulas.
5. Develop, implement, and evaluate a metamodel combining both approaches, or some other combination of deep learning and temporal logics.
6. **(optional)** Enrich the comparison by considering further case studies.
7. **(optional)** Develop an extension of these techniques to tackle MFOTL or other temporal logics. A possible starting point is [1].
8. Write the final report and prepare the presentation.

Deliverables

The following deliverables are due at the end of the project:

Final report The final report should consist of an introduction; an overview over related work; a description of each implemented learning technique; one or more sections discussing the evaluation; and a conclusion.

Implementation Submit the code of your implementation, all generated formulas and models, and all datasets used in the evaluation.

Presentation At the end of the project, a presentation of 30 minutes must be given during an InfSec group seminar. It should give an overview and discuss the most important highlights of the work.

References

[1] Ezio Bartocci, Luca Bortolussi, and Guido Sanguinetti. Data-driven statistical learning of temporal logic properties. In Axel Legay and Marius Bozga, editors, *Formal Modeling and Analysis of Timed Systems*, pages 23–37, Cham, 2014. Springer International Publishing.

[2] Alberto Camacho and Sheila A. McIlraith. Learning interpretable models expressed in linear temporal logic. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29(1):621–630, May 2021.

[3] Gary Marcus. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631, 2018.

[4] Prasad Naldurg, Koushik Sen, and Prasanna Thati. A temporal logic based framework for intrusion detection. In David de Frutos-Escrig and Manuel Núñez, editors, *FORTE 2004*, volume 3235 of *LNCS*, pages 359–376. Springer, 2004.

[5] Daniel Neider and Ivan Gavran. Learning linear temporal properties. In Nikolaj Bjørner and Arie Gurfinkel, editors, *FMCAD 2018*, pages 1–10. IEEE, 2018.

[6] Martin P. Robillard, Eric Bodden, David Kawrykow, Mira Mezini, and Tristan Ratchford. Automated API property inference techniques. *IEEE Trans. Software Eng.*, 39(5):613–637, 2013.

[7] Grigore Rosu and Feng Chen. Semantics and algorithms for parametric monitoring. *Log. Methods Comput. Sci.*, 8(1), 2012.

[8] Anand Ashok Sawant and Alberto Bacchelli. A dataset for API usage. In Massimiliano Di Penta, Martin Pinzger, and Romain Robbes, editors, *MSR 2015*, pages 506–509. IEEE Computer Society, 2015.