Epistemic Accountability: Rigorously Investigating Autonomous Decisions

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Drew Goldman

Spring, 2023

Technical Project Team Members

Dr. Ruzica Piskac

Samuel Judson

Katrine Bjorner

Filip Cano

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Briana Morisson, Department of Computer Science

Epistemic Accountability: Rigorously Investigating Autonomous Decisions

CS4991 Capstone Report, 2023

Drew Goldman
Computer Science, Economics
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
dag5wd@virginia.edu

Abstract
As autonomous agents are deployed with greater independence and less complete information into more complicated environments, the risks of these agents taking actions with adverse consequences increases; consequently, accountability of those agents is necessary. On a team with Yale professors, PhD candidates, and other undergraduate students, I have been conducting independent research for the last six years. We employed formal methods for human-in-the-loop analysis of decisions drawn from program verification, symbolic execution, and program synthesis by investigating the classification of a neural network trained on MNIST handwritten digit images. The results demonstrate the effectiveness and scalability of understanding intention by synthesizing mimic programs for neural networks utilizing the SMT-based cvc5 synthesizer. Further work may include investigating how this method can be tailored to work in tasks with different types of data and qualitatively evaluating the usefulness of this interpretation compared to standard statistical methods.

1. Introduction

For the past six years, a team of Yale professors, PhD candidates, undergraduate students and I have been conducting independent research into a system for utilizing algorithms to ensure the accountability of autonomous agents. This system, originally conceived by a Yale doctoral candidate encapsulates probability, cryptography and computer logic.

On May 6th, 2010, the US-based stock market suddenly saw massive and irrational price swings, leading to trillions of dollars of cumulative losses, all within the course of minutes. The collapse was not the result of fundamental economic uncertainty. Rather, a complex sequence of human and automated decision-making turned one slightly anomalous order into a breakdown in the proper functioning of the markets (CFTC and SEC, 2020). This “flash crash” provoked both regulatory and criminal investigations, which struggled to assign responsibility for this near calamity. Black-box autonomous agents do not uniquely generate risk in finance, however. In 2016 a fatal accident involving a self-driving Tesla car signified the need for accountability in autonomous vehicles. The incident sparked debate about the safety and liability of self-driving technology and raised questions about the responsibility of the car manufacturer, software developer, and even the vehicle owner. If autonomous entities are propagated into society with amplified liberty, the requisite for liability only intensifies.

Assessment of human culpability requires inferring the reasoning underlying harmful

decisions. Experts do so by interpreting the knowledge, beliefs, desires, and intentions of the decision maker. Formal accountability processes, such as trials and accident review boards, rely upon an investigation of the observed events that led to the injury, with counterfactual analysis playing a critical role. Just as there is a meticulous process of ensuring accountability in human matters, it is equally important to promote the answerability of autonomous agents.

2. Related Works

Explaining the decision processes of deep neural networks has become a major focus of research into their accountability and the trustworthy use of machine learning (Amina and Berrada, 2018). Amina and Berrada emphasize the need to understand decision-making processes in complex AI systems, such as deep neural networks, to ensure accountability and trustworthiness, which has been a guiding principle for our own research. Furthermore, they provide an overview of techniques to increase transparency and prevent negative consequences of opaque decision-making.

Notwithstanding that black-box models generally express highly-complex and non-interpretable transformations across the full input space, locally the model can often be represented by simpler hypotheses (Ribeiro, et al, 2016). Whether programming-by-example can serve as a viable alternative to statistical learning methods for producing local explanations of black-box models is a question we sought to answer. This line of inquiry is crucial as it seeks to shed light on the decision-making processes of such models and promote their accountability. Regardless of the field in which an autonomous agent operates, a proper system should be able to provide guidance in assigning accountability in the event of any unintended adverse outcomes (Poirier, 2012).

3. Project Design

What follows is a brief overview of the project. To investigate and explain the behavior of autonomous agents, we chose to experiment on the MNIST handwritten image database. We derived an underlying decision procedure for a simple neural network trained on the database to classify the images with high accuracy and precision. Using input-output test data pairs of the neural network as input to a synthesizer called cvc5, we were able to generalize the decision making of the neural network in the form of a decision tree. Our approach supports agents that rely upon both logical (algorithmic) and statistical (machine learning) reasoning.

3.1 MNIST Database

The MNIST database is a canonical benchmark in machine learning. We chose MNIST because it allowed us to evaluate our (non-bootstrapped) method on a simple but non-trivial multiclass-classification task of a well-understood and self-contained nature. Each MNIST instance is comprised of a feature vector, encoding a 28x28 grayscale image, with ten possible output classes, each corresponding to a different base-ten Hindu-Arabic numeral.

3.2 Convolution Neural Network

For our model, we implemented a convolutional neural network architecture from a popular tutorial on solving MNIST (Chollet 2015). The neural network implemented the Tensorflow (keras) package, and the tutorial employed multiple convolution and pooling layers before flattening and ultimately classifying via a softmax activation. We integrated the general 80/20 training/testing split. While the neural network classified all the test images as either a {0, 1, …, 9}, we greatly simplified the problem by only examining the decision boundary between just two of the classifications—1 and 7. That is, we

discarded all the images that were not either a 1 or a 7. Therefore, the convolution neural network (CNN) classified images correctly as either a 1 or a 7, or the model misclassified 1s as 7s or 7s as 1s. The neural network achieved an accuracy of almost 99%, which is unsurprising as this is a relatively easy classification task in the machine learning literature, by today's standards.

We sorted the images via the probability with which the model believed an image to be a 7. It is implied that the complement probability is its best guess of whether the image was a 1. Essentially, that sorted list contained the images (input) and the models' classification (output) on which the model had the lowest confidence. The images likely had features which made them look ambiguous. We chose 1 and 7 because they are the two Hindu-Arabic numerals that look most similar and therefore are closest to the decision boundary of the neural network. It is particularly these edge-cases that provide the greatest challenge for autonomous black-box agents. We subsequently input these input-output data pairs into a synthesizer program.

3.3 CVC5 Synthesizer

CVC5 is an automated theorem prover and SMT (Satisfiability Modulo Theories) solver. It is a software tool that can be used to check the validity of logical formulas and to find solutions to mathematical problems. In the context of our project, we wrote grammar in SMT-LIBv2 format on which the CVC5 program operated. We constrained the program to compare the input pixels of images using only less than or equal to operators and joined those comparisons together using only AND, OR, and NOT logical operators. The CVC5 program synthesized a decision tree for the neural networks' classifications of the ambiguous images.

4. Results

The synthesis revealed a decision procedure that the neural network created through its training. Figure 1 below demonstrates, in a simplified manner, how the model determined whether the image was a 1 or a 7. In this model created by the synthesizer, the decision tree has depth of 6 with 19 branching nodes and the model compares the value of only 23 pixels out of the 28x28 = 784 pixels.

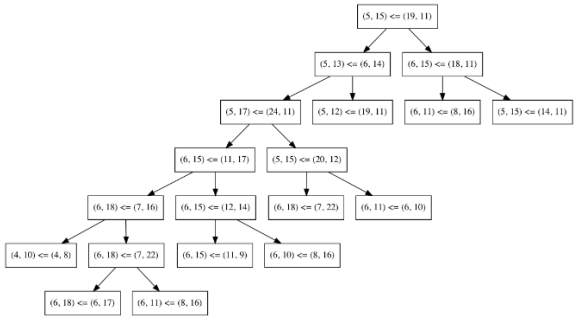

Figure 1: Decision Tree forClassification of 7

The heatmap, depicted in Figure 2 below illustrates which pixels are used by the model shown in Figure 1. The darkness of each pixel corresponds to the number of times it appears in a guard within the model. The heatmap shows the relative importance of given pixels when the model decides between classifying the image as a 1 or a 7. Figure 2 captures an intuition that the distinguishing features between the two digits are the width of the horizontal stroke as well as the slant and depth of the vertical.

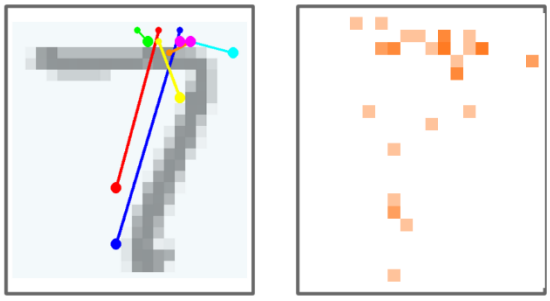

Figure 2: Heatmap for 7 Classification

Finally, we evaluated the accuracy of our synthesized mimic programs. While our

approach guarantees that the model is exact-by-construction over the pixels, we wanted to investigate how well the model generalized to the entire test set. We evaluated the accuracy twice, once with respect to the classifications of the opaque model and once with respect to the `ground truths' (the true classification given in the dataset) in the testing data. We found the difference between accuracy and recall to be very small, likely as the baseline accuracy of the opaque model is already at 99%.

5. Conclusion

Our research highlights the importance of accountability in the deployment of autonomous agents. The use of formal methods for human-in-the-loop analysis of decisions drawn from program verification, symbolic execution, and program synthesis showed promising results in understanding the intentions of neural networks, which could be applicable to a wide range of decision-making systems. Moreover, the study demonstrated the effectiveness and scalability of the SMT-based cvc5 in synthesizing mimic programs for neural networks. In addition to its potential applications and benefits, the project provided me with valuable data science and symbolic logic skills. Overall, this research serves as an endorsement of the critical role that accountability and formal methods play in the development and deployment of autonomous systems.

6. Future Work

Additional work is needed to test the effectiveness of the method on data sets beyond the MNIST handwritten digit images. This could include real-world data sets, in the biomedical, financial, or environmental realms. Moreover, the potential for the synthesized mimic programs to be used in the interpretability and explainability of autonomous decision-making systems could be further investigated. These more comprehensive assessments of the method's effectiveness and scalability, as well as its potential for use in a wide range of autonomous decision-making systems, unlock more possibilities for generalizing the insights in this study.

7. Acknowledgments

I would like to sincerely thank my mentors Dr. Ruzica Piskac and doctoral candidate Samuel Judson for their many years of guidance. I would also like to thank Filip Cano and Katrine Bjorner for their consistent encouragement and assistance on this project.

References

[1] CFTC and SEC. 2010. Findings Regarding the Market Events of May 6, 2010: Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues. Retrieved from https://www.sec.gov/news/studies/2010/marketevents-report.pdf.

[2] Elon Musk. 2016. A Tragic Loss. *Tesla*. Retrieved from https://www.tesla.com/blog/tragic-loss

[3] Isabelle Poirier. 2012. "High-Frequency Trading and the Flash Crash: Structural Weaknesses in the Securities Markets and Proposed Regulatory Responses." *Harvard Business Law Journal* 8, 2. Retrieved from https://repository.uchastings.edu/hastings_business_law_journal/vol8/iss2/5.

[4] Anass Amina and Mohamed Berrada. 2018. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)." *IEEE Access* 6: 52138-60. DOI: https://doi.org/10.1109/ACCESS.2018.2870052.

[5] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd *ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, 1135-1144. ACM. DOI: https://doi.org/10.1145/2939672.2939778.

[6] François Chollet. 2015. MNIST Convolutional Neural Network (CNN) example. *Keras Documentation*. Retrieved from https://keras.io/examples/vision/mnist_convnet