CAV Project Proposal: Kendall Ahern

Formal Verification of Neural Network Robustness Using SMT-Based Methods

Since becoming a student at CU, I have found a passion for, and want to make a career out of AI and machine learning. All of my other coursework is in that field of study, so I would love to find a way to incorporate that into this project. This is the best idea that I could come up with that relates the two and is feasible in the timeline, but would love feedback and am open to any suggestions you might have.

Machine learning models, especially neural networks, often behave unpredictably when exposed to small disturbances or variance in the input. In systems where safety is probably the most important component (autonomous vehicles or medical devices), this lack of robustness can have serious consequences. Formal verification techniques, I think, could offer a way to mathematically guarantee safety properties of these models rather than relying solely on empirical testing.

My hope with this project will explore formal verification of a small feedforward neural network using SMT solvers in Python. My plan is to use Z3, because that is what we have done on homework's – but I also saw a tool called Marabou that has been used in this field and would love your input - to verify whether a trained neural network satisfies a robustness property. Basically, I don't want those small input variances to change the class that model predicts. My goal with this project is to gain hands-on experience with some of the symbolic verification methods we discussed in class by applying them applying them to AI systems – which is where I hope to take my career. At the end of the day, I want to evaluate the practicality and limitations of SMT-based verification for neural networks.

In terms of objectives and milestones:

1. Train a small neural network on a simple dataset – probably from MNIST.
2. Formally specify some robustness property (Ex: *"for all inputs within k neighborhood of x, the predicted label remains the same")*
3. Encode that property into an SMT solver
4. Figure out if property holds, or produce counterexample if it doesn't
5. (Time permitting) Look at runtime and scalability with respect to network size/structure.

References

1. Katz, G., Barrett, C., Dill, D. L., Julian, K., & Kochenderfer, M. J. (2019). *The Marabou framework for verification and analysis of deep neural networks. Computer Aided Verification (CAV)*. https://arxiv.org/abs/1902.03847
2. Ehlers, R. (2017). *Formal verification of piece-wise linear feed-forward neural networks. Automated Technology for Verification and Analysis (ATVA).* https://doi.org/10.1007/978-3-319-68167-2_5
3. Katz, G., et al. (2017). *Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. Computer Aided Verification (CAV).* https://arxiv.org/abs/1702.01135