Evaluating Sampling Techniques for Testing Cyber-Physical Systems with Machine Learning Components

Edward Kim Shromona Ghosh Tommaso Dreossi Daniel J. Fremont
Alberto L. Sangiovanni-Vincentelli Sanjit A. Seshia

University of California, Berkeley

Abstract

Machine learning (ML) techniques such as neural networks are increasingly being used within safety-critical cyber-physical systems (CPS) such as autonomous cars. Systematic testing against formal specifications is an important tool for providing higher assurance in these systems. In this paper, we survey a range of sampling-based search techniques for testing CPS with ML components.

Objective

We assume that the system's behavior is unknown but its responses to a given stimulus can be either simulated or measured. The purpose of testing is to find inputs that result in violation of a formal specification. In particular, we focus on sampling-based testing methods where one seeks to efficiently sample the input space to search for such inputs. We compare a number of sampling techniques using accuracy and efficiency as criteria.

The test methods considered here are active or passive samplers. In active samplers, there is feedback from the simulation or measurement to the sampler at each sampling iteration. Random sampling and Halton sequence-based sampling [1] are passive, while Simulated Annealing, Cross-Entropy, and Bayesian Optimization are active samplers. We compare these sampling techniques within the VERIFAI toolkit [2], which provides a framework for design and verification of CPS with ML components. We would like to assess whether active sampling always outperforms passive sampling. In fact, we find that allowing the sampler to use feedback *does not* always increase its accuracy.

Evaluation Method

We tested the samplers on three different case studies provided in VERIFAI: (i) counterexample-guided data augmentation [3], (ii) an end-to-end Neural Network (NN) trained to balance a cartpole in OpenAI Gym [4], and (iii) a semi-autonomous car that includes a Convolutional Neural Network (CNN) and a hybrid controller in the Webots robotics simulator [5].

In the first case study, VERIFAI uses the following testing process: first, sample a high-level description of a scene, then convert this description into a synthetic image using an image renderer, and finally, test the trained CNN on this image. The sampling space consists of a tuple of discrete and continuous variables describing the scene: the x, y position of a car in the 2D image space, the brightness, sharpness, contrast, and color of images, together with discrete variables for the type of car and background. Given ranges for all these variables, the task is to sample scenes where the CNN fails to detect a car on the road. We performed 10 experiments, each consisting of 100 sampling iterations.

In the second case study, a neural network is trained using OpenAI baselines [6] to balance a cartpole. The specification is that the maximum deviation of the pole from vertical should be less than 12° and the maximum deviation from the initial position should be less than 2.4 meters. This specification is encoded as a Metric Temporal Logic (MTL) formula [7]. The sample space consists of the cartpole's position, angle, and length as well as the mass of the cart and pole. In the OpenAI Gym environment, we simulate the cartpole with the sampled initial condition and a trained NN controller, record the trajectory of the cartpole over a defined time interval, and finally evaluate the MTL formula over the trajectory. The objective here is to search for initial conditions that maximize the violation of the specification. We ran 10 experiments each consisting of 100 samples.

Finally, in the Webots simulation case study, we test a semi-autonomous vehicle in a traffic setting where on a three-lane road, the leftmost lane is blocked by a broken car surrounded with three cones to indicate that the lane is blocked. The semi-autonomous car (the "ego" vehicle) also starts in the leftmost lane and has to avoid crashing by changing a lane to the right. This car is equipped with a trained CNN that outputs the distance to the broken car in front, as well as a hybrid controller that switches between cruise control and lane change depending on this distance. The specification, encoded in MTL, is that the car must not hit at any time the broken car or any of the cones. The sample space consists of the initial ego car position, cruising speed, and reaction time to switch control modes, the color and rotation of the broken car, and finally the orientations and 2D locations of the traffic cones. As in the cartpole study, the testing procedure is to simulate first in Webots with the sampled initial condition for a defined time period, collect the trajectory of the ego car, and finally evaluate the MTL formula over the trajectory. The objective is to sample conditions that maximize violation of the formula. We ran three experiments, each with five sampling iterations (we used lower numbers because the

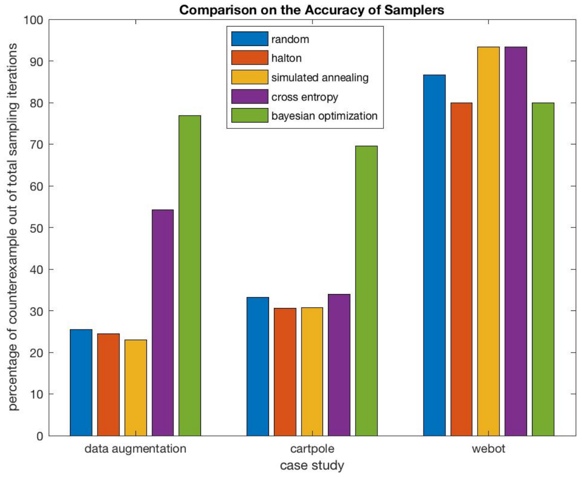

Figure 1: Average percentage of counterexamples found with different sampling techniques.

simulation time for this case study is high).

Preliminary Results and Analysis

Our experiments show an interesting trade-off between active and passive samplers. First, as expected, active samplers, except for simulated annealing, had a higher accuracy in general (in the sense of finding counterexamples) than passive ones as seen in Figure 1. However, we also observed that both Cross-Entropy and Bayesian Optimization tend to find samples in a small local neighborhood in the data augmentation and cartpole case studies. This pattern was especially conspicuous in experiments with accuracy above 50 percent. For example, in the data augmentation case, most of our counterexamples the discrete variables defining the type of scene background (e.g. urban traffic vs. rural road) and the type of the car shown in the image were assigned the same values. Since having a diverse set of counterexamples is critical to identifying which data to augment, the regularity of counterexamples produced by these samplers is potentially problematic. The same is true for the cartpole study, where it is important to test that the controller is robust to diverse initial conditions. Yet, we observed a similar pattern where, for instance, continuous variables such as the length and mass of the pole were fixed in the identified counterexamples while varying all other variables.

On the contrary, we found that Random, Halton, and Simulated Annealing did not suffer from this lack of diversity although they sufferd from lower accuracy. Among the passive samplers, we expected Halton, which seeks to maximize coverage over the space for a given number of samples, to generate the most diverse set of counterexamples. However, in terms of accuracy, it performed about the same as random sampling.

Comparing the run time of the samplers as shown in Figure 2, we observe that although Bayesian Optimization found the most counterexamples, the trade-off in run time outweighs the accuracy benefit. Therefore, we conclude that Bayesian Optimization may not actually be a suitable technique for finding counterexamples due to its high run time and lack of diversity in its search.

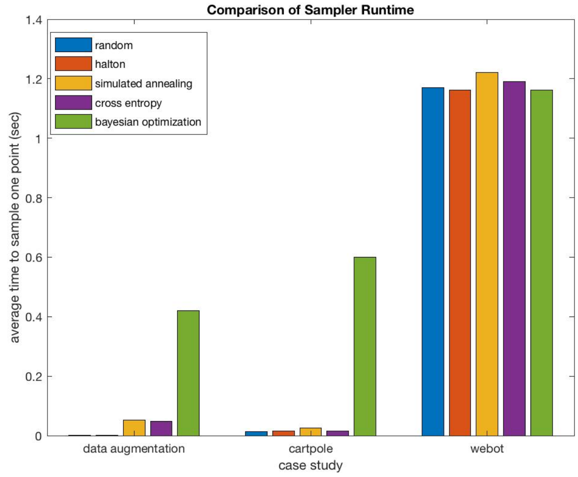

Figure 2: Comparison of sampler run times.

As an alternative, simulated annealing has the strengths of both diverse exploration and local fuzz testing (in the sense of exploring variations around an interesting input). In our experiments, we used a fixed temperature scheduler. However, using dynamic temperature scheduling to enable smart switching between diverse exploration and local search, we may be able to provide both diversity and accuracy. Considering its low run time, comparable to simple random sampling, there is definitely space to further explore simulated annealing in future work.

Acknowledgments: This work was supported in part by the NSF Graduate Research Fellowship Program under Grant No. DGE-1106400, NSF grants 1545216, 1646208, and 1739816, the DARPA BRASS program, the DARPA Assured Autonomy program, the iCyPhy center, and Berkeley Deep Drive.

References

[1] J. Halton, “On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals,” *Numerische Mathematik 2(1), 8490*, 1960.

[2] T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, H. Ravanbakhsh, M. Vazquez-Chanlatte, and S. A. Seshia, “VerifAI: A toolkit for the design and analysis of artificial intelligence-based systems,” *arXiv preprint arXiv:1902.04245*, 2019.

[3] T. Dreossi, S. Ghosh, X. Yue, K. Keutzer, A. Sangiovanni-Vincentelli, and S. A. Seshia, “Counterexample-guided data augmentation,” *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, 2018.

[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[5] Cyberbotics, “Webots,” 2018. `http://www.cyberbotics.com`.

[6] P. Dhariwal and et al., “OpenAI baselines.” `https://github.com/openai/baselines`, 2017.

[7] R. Alur and T. Henzinger, *Real-Time: Theory in Practice*. Springer Berlin Heidelberg, 1992.