Abstraction-Based Training and Verification of Safe Deep Reinforcement Learning Systems (Extended Abstract)

Min Zhang

Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
zhangmin@sei.ecnu.edu.cn

Abstract. Deep Reinforcement Learning (DRL) systems shall be formally verified when they operate in safety-critical domains. However, their verification is a very challenging problem for two main reasons, i.e., the continuity and infinity of system state space and the inclusion of inexplicable decision-making deep neural networks (DNNs). We propose to first abstract the continuous and infinite state space into a finite set of *abstract states* and then train the system on these abstract states. This *abstract training approach* brings manifold benefits. First, we can build a verifiable formal model based on the same abstraction and verify whether it satisfies the expected safety and functional requirements using off-the-shelf model checkers. The verification results are then used to guide the abstraction refinement repeatedly for further training until all the requirements are satisfied. Second, we can perform a tight and scalable reachability analysis of the trained systems by treating the planted neural networks as black boxes, thus avoiding over-approximating them. Third, we can flexibly fine-tune the granularity in which the system states are abstracted for a better balance between robustness and performance.

Keywords: Abstraction · Deep reinforcement learning · Model checking · Probabilistic robustness · Reachability analysis

Background

Deep Reinforcement Learning (DRL) is an artificial intelligence technique for developing autonomous systems where deep neural networks (DNNs) are planted for decision-making. It has been developing quickly to solve those hard-specifiable systems such as robot control autonomous driving [9]. As some of those domains are safety-critical, their functionality, safety, and robustness shall be formally verified before deployment [4]. However, the verification problem is very challenging due to two main reasons. One is that the system state space is usually continuous and infinite, and the other is that the system is driven by an in-explicable and non-linear deep neural network. The two facts make it difficult to build an efficiently verifiable formal model. Existing verification approaches have to abstract the system state space and over-approximate the network after a system is trained. Such *ex post facto* verification has several limitations. One is that the abstraction and over-approximation introduce

too much overestimation and consequently result in false positives in verification results. Another is that the verification results are hardly utilized to improve the system reliability as further training after verification may cause an unpredictable impact on system properties due to the inexplicability of neural networks [5].

Abstraction-Based Training and Verification

Verification-in-the-Loop Training [8]. As inspired by the importance of abstraction to the formal verification of infinite-state systems [7], we propose to abstract the infinite state space S of a DRL system into a finite set $\mathcal{S}$ of abstract states by defining the abstraction function $\mathcal{A} : S \to \mathcal{S}$ and train it on $\mathcal{S}$. At each training step t, we map the system state $s_t \in S$ to its corresponding abstract state $s = \mathcal{A}(s_t)$ and feed $\mathtt{s}$ into the planted neural network π to compute an action $a = \pi(\mathtt{s})$. A reward is then computed by a predefined reward function based on a and s_t, and the parameters in the neural network are updated correspondingly. The system proceeds to the successor state $s_{t+1} = f(s_t, a)$, where f is the system dynamics.

Under the same abstraction, we can build a state transition system $\mathcal{M}_{\mathcal{S}} = \langle \mathcal{S}, \mathcal{I}, \mathcal{T} \rangle$, where $\mathcal{I} = \{s | s \in \mathcal{S} \wedge \exists s_0 \in S_0 . s = \mathcal{A}(s_0)\}$ and $(s, s') \in \mathcal{T}$ if $s' \in \hat{f}(s, \pi(s))$ for all $s, \mathtt{s}' \in \mathcal{S}$. Here, S_0 is the set of initial states of the system, and $\hat{f}(s, \pi(s)) = \{s' | \exists s \in \mathcal{C}(s) . s' = \mathcal{A}(f(s, \pi(s)))\}$ denotes the set of successor abstract states from $\mathtt{s}$, where $\mathcal{C}$ is the corresponding concretization function of $\mathcal{A}$. $\mathcal{M}_{\mathcal{S}}$ is a simulation of the trained DRL system, i.e., for any transition from s_t to $s_{t+1} = f(s_t, a)$ and abstract state s, if $s = \mathcal{A}(s_t)$, then there exists $s' \in \mathcal{S}$ such that $s' \in \hat{f}(s, \pi(\mathtt{s}))$ and $s' = \mathcal{A}(s_{t+1})$. Since $\mathcal{S}$ is finite, we can leverage *off-the-shelf* model checkers to model check $\mathcal{M}_{\mathcal{S}}$ against the pre-specified properties defined in some temporal logic such as ACTL. When counterexamples are found, they could be spurious due to the abstraction. We then refine the abstract state $\mathcal{S}$ guided by the counterexamples and continue to train the system on the refined abstract state space. We repeat this *verification-in-the-loop* training process until all the properties are verified, and we finally obtain a verified safe DRL system.

Tight and Scalable Reachability Analysis [11]. Reachability analysis is an effective way to verify the safety properties of DRL systems [3, 6]. Given a DRL system with state space S, let R_S be the set of all the reachable states. We have $S_0 \in S$ and $s' \in R_S$ for all $s' \in S$ if there exists some state $s \in R_S$ such that $s' = f(s, N(s))$, where N is a neural network trained on S. Generally, it is an undecidable problem to check whether a state is reachable or not for a DRL system because the reachability problem of most nonlinear systems is undecidable [2]. Due to the infinity of S and the non-linearity of f, we have to over-approximate both N and f to overestimate R_S. This dual over-approximation results in too much overestimation and limited scalability to large neural networks.

By training the system on the abstract state space $\mathcal{S}$, we can over-approximate R_S more tightly and scalably via computing the set of reachable abstract states. Let the overestimated set be $\mathcal{R}_{\mathcal{S}}$, where $\mathcal{I} \in \mathcal{R}_{\mathcal{S}}$ and $s' \in \mathcal{R}_{\mathcal{S}}$ for all $s' \in \mathcal{S}$ if there exists some $s \in \mathcal{S}$ such that $(s, s') \in \mathcal{T}$. Because the neural network π is trained on $\mathcal{S}$, we can avoid

over-approximating π and treat it as a black box when checking whether $(s, s') \in \mathcal{T}$ holds or not. It suffices to compute the corresponding action on s by $a = \pi(\mathbf{s})$ and determine the successor abstract states in $\hat{f}(s, a)$. The concretization of all the reachable abstract states constitutes an overestimated set of reachable actual states. In this process, we only need to over-approximate the dynamics f, which consequently yields a tight and scalable way to overestimate the set of reachable states for the DRL system.

Probabilistic Robustness Training and Evaluation [12]. Training on abstract states is also helpful in developing robust DRL systems. A DRL system is considered *robust* in a state with respect to some perturbation if it takes the same action on all the perturbed states. Under abstract training, a perturbed state may be mapped to the same abstract state and thus have the same action as the original state. The probability of mapping a perturbed state to the same abstract state as the original state can be estimated, yielding an analytical metric called *probabilistic robustness* to indicate the system robustness. The metric only depends on abstraction and thus can be computed analytically but not experimentally. We have proved that the probability increases monotonously with the granularity in which system states are abstracted. Consequently, we can achieve a flexible mechanism to balance the robustness and performance of trained DRL systems by fine-tuning the abstraction granularity of the system states.

The prototypes for the safe and robust training, verification and reachability analysis, and technical documents are available at https://github.com/aptx4869tjx/RL_verification.

Concluding Remarks

We believe that abstraction is a promising solution for connecting formal methods and deep reinforcement learning for developing provably reliable DRL systems. Following the work [1], which shows the feasibility of applying abstraction to the training phase, we demonstrate that abstraction can be utilized simultaneously in both verification and training. Introducing abstraction into both training and verification brings manifold benefits, e.g., simplifying the subsequent verification, utilizing the verification results for further training, computing tight sets of reachable states in a scalable and orthogonal manner to the size, architecture, and type of activation functions of neural networks, and balancing the robustness and performance by flexibly fine-tuning the abstraction granularity. All these benefits are necessary to develop safe and robust DRL systems.

Several problems remain ahead when the abstraction-based training and verification approach is applied to real-world complex DRL systems. One practical problem is extending it to high-dimensional systems, whose states require sophisticated abstractions defined particularly for neural network verification [10] to avoid state explosions in both training and verification phases. Another interesting direction is applying the proposed approaches to other variant DRL systems with non-deterministic and stochastic features, which could be verified using probabilistic and statistical model checking approaches. It is also interesting to explore the possibility of extending the training and verification approaches to classification tasks for training verifiable and robust deep neural networks.

Acknowledgments. The author thanks SETTA 2022 organizers for the invited talk. This work has been partially supported by National Key Research Program (2020AAA0107800), NSFC-ISF Joint Program (62161146001, 3420/21) and NSFC projects (61872146, 61872144), Shanghai Trusted Industry Internet Software Collaborative Innovation Center and "Digital Silk Road" Shanghai International Joint Lab of Trust-worthy Intelligent Software (Grant No. 22510750100).

References

1. Abel, D.: A Theory of Abstraction in Reinforcement Learning. Dissertation, Brown University (2020)
2. Asarin, E., Mysore, V.P., Pnueli, A., Schneider, G.: Low dimensional hybrid systems–decidable, undecidable, don't know. Inf. Comput. **211**, 138–159 (2012)
3. Fan, J., Huang, C., Chen, X., Li, W., Zhu, Q.: ReachNN*: atool for reachability analysis of neural-network controlled systems. In: Hung, D.V., Sokolsky, O. (eds) ATVA 2020. LNCS, vol. 12302, pp. 537–542. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59152-6_30
4. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. **16**, 1437–1480 (2015)
5. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep reinforcement learning that matters. In: AAAI 2018, pp. 3207–3214. AAAI Press (2018)
6. Ivanov, R., Carpenter, T., Weimer, J., Alur, R., Pappas, G., Lee, I.: Verisig 2.0: verification of neural network controllers using Taylor model preconditioning. In: Silva, A., Leino, K.R. M. (eds.) CAV 2021. LNCS, vol. 12759, pp. 249–262 (2021). Springer, Cham. https://doi.org/10.1007/978-3-030-81685-8_11
7. Jackson, D.: Abstract model checking of infinite specifications. In: Naftalin, M., Denvir, T., Bertran, M. (eds.) FME 1994. LNCS, vol. 873, pp. 519–531. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58555-9_113
8. Jin, P., Tian, J., Zhi, D., Wen, X., Zhang, M.: TRAINIFY: a CEGAR-driven training and verification framework for safe deep reinforcement learning. In: Shoham, S., Vizel, Y. (eds.) CAV 2022. LNCS, vol. 13371, pp. 193–218 (2022). Springer, Cham. https://doi.org/10.1007/978-3-031-13185-1_10
9. Kiran, B.R., et al.: Deep reinforcement learning for autonomous driving: a survey. IEEE Trans. Intell. Transp. Syst. pp. 4909–4926 (2021)
10. Singh, G., Gehr, T., Püuschel, M., Vechev, M.: An abstract domain for certifying neural networks. In: POPL 2019. pp. 1–30. ACM (2019)
11. Tian, J., Zhi, D., Wang, P., Liu, S., Katz, G., Zhang, M.: `BBReach`: tight and scalable black-box reachability analysis of deep reinforcement learning systems (2022, submitted)
12. Zhi, D., Tian, J., Wang, P., Liu, S., Wen, X., Zhang, M.: Probabilistic robustness for deep reinforcement learning with provable guarantees (2022, submitted)