Safety-aware Expansion for Neural Network Repair

Keyvan Majd[1], Georgios Fainekos[2], and Heni Ben Amor[1]

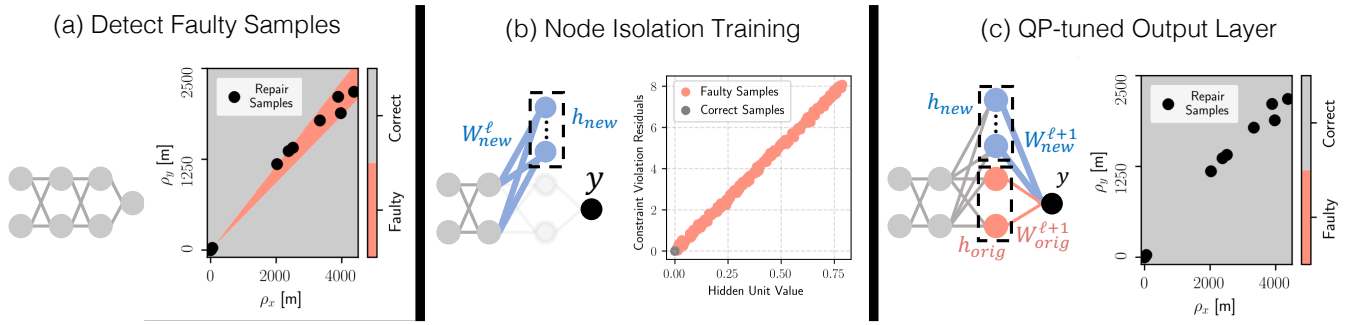

Fig. 1: Overview of deep neural network repair with safety-aware expansion for the HCAS [1] network†. (a) The network returns faulty advisory outputs. We collect a repair dataset that contains erroneous and correct data samples. (b) New hidden units are added to the last hidden layer of the network and undergo isolated training to optimize the correlation between their values and the residual values of constraint violations. (c) The last layer is fine-tuned using quadratic programming to minimize the original loss of the network, subject to hard output constraints that ensure correct classification.

Abstract

Deep Neural Networks (DNNs) have played a critical role in recent advances in the field of robotics by allowing robots to perceive and interact with their surroundings more intelligently and autonomously. However, the application of DNNs in safety-critical contexts is limited, as they might demonstrate unsafe behavior when facing unseen samples after training [2]. Moreover, considering the circumstances, one may need to adjust the output of a trained network to align with the new safety requirements or constraints. In such cases, it becomes essential to repair the neural network to rectify any identified unsafe behavior or to ensure that the network aligns with the new requirements.

Neural network repair involves updating the DNN weights or architecture to satisfy safety requirements. Existing methods approach this problem through **direct weight modification** or **architecture extension**. One approach to direct weight modification is counterexample-guided retraining or fine-tuning [3]–[7]. This approach involves manual sample labeling or relaxed constraints in the objective function. Although these approaches apply to all types of activation functions, they cannot guarantee safety satisfaction even for faulty samples. Methods in [8] and [9] propose provable repair solutions by transforming the repair problem into a satisfiability problem or a mixed-integer quadratic program (MIQP), respectively. However, solving the satisfiability problem and MIQP is a demanding process that scales with the size of network. Moreover, these methods are restricted to DNNs with linear piece-wise continuous (LPWC) activation functions.

An alternative repair strategy is architecture extension [10], [11], focusing on the linear regions of LPWC networks where faults occur. These techniques construct a patch network [11] or decouple networks [10] in the detected faulty linear regions. These techniques introduce a different architecture from the original network, which lacks a seamless solution. Furthermore, these approaches could potentially downgrade the network's original performance within the repaired regions, as maintaining the original performance of the network is not considered during the repair process.

Inspired by the cascade-correlation algorithm introduced in [12], we present a provable repair method by growing neural networks. Given a set of samples $\mathcal{T}$ including both faulty and correct samples, a DNN π, and the constraints of form $g(x, \pi(x)) = 0$, we add a single neuron to the last hidden layer. The incoming weights of the new neuron are initialized randomly, while the outgoing weights are initialized to zero. The new hidden unit adopts the same activation function as the other units within the same layer. Defining the residual error of the constraint violations as $\sum_{x_i \in \mathcal{T}} |g(x_i, \pi(x_i))|$, we first train the incoming weights of the new unit using the gradient descent. We maximize the covariance between the hidden unit's values and the residual errors associated with constraint violations as the objective function. The objective function also encourages the hidden unit to only activate in response to faulty samples, thereby preserving the network's original performance for the correct samples. After isolation training of the new hidden unit, we fine-tune the last layer of the network

[1] K. Majd, and H. Ben Amor {majd, hbenamor}@asu.edu are with SCAI, Arizona State University, Tempe, AZ, USA.
[2] G. Fainekos is with Toyota NA R&D, Ann Arbor, MI, USA.

†HCAS is an aircraft collision avoidance system that outputs one of five possible control advisories ('Strong left', 'Weak left', 'Clear-of-Conflict', 'Weak right' and 'Strong right') given the relative distance and relative heading of ego aircraft to intruder (ρ_x, ρ_y, ϕ).

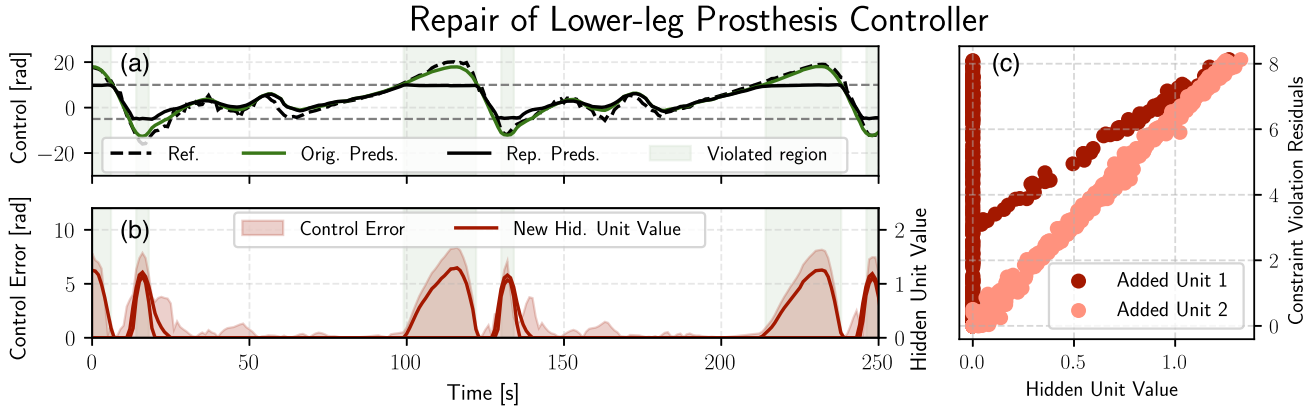

Fig. 2: Using safety-aware expansion in repairing a DNN controller for a lower-leg prosthesis device by adding two hidden units. The original policy is repaired to bound the control output within $[-5, 10]$ *rad*. (a) The control is successfully bounded after the repair (the black signal represents the control values after the repair). (b) The newly added hidden units are activated only for inputs that lead to constraint violations. As a result, the error is nearly zero in regions that were initially correct. (c) This plot demonstrates that the values of the new hidden units are correlated with the residuals after training in isolation, and the added units remain inactive for samples with zero residuals.

using Quadratic Programming (QP). QP minimizes the Mean-Squared Error (MSE) between the network's predicted outputs and the original target values subject to the hard constraint $g(x, \pi(x)) = 0$. In the QP formulation, we also restrict the changes of the original weights to be minimal. If optimization is not feasible, it shows that the added hidden unit cannot cancel all the residual errors from the constraint violations. Therefore, we repeat the repair process by adding more hidden units until QP is feasible. The feasible solution guarantees the satisfaction of the constraints for the faulty samples.

Figure 1 illustrates how our method addresses safety violations in the $N_{1,4}$ HCAS aircraft collision avoidance network [1]. The HCAS system returns a control advisory output ('y_1: Strong left', 'y_2: Weak left', 'y_3: Clear-of-Conflict', 'y_4: Weak right' and 'y_5: Strong right') given the relative distance and heading between the ego aircraft and intruder as inputs. The network should always advise y_1, but it violates this safety property as shown in Fig. 1 (a). To address this, we expand the last hidden layer of the network with a neuron and train it in isolation, as shown in Fig. 1 (b). The new unit is well-trained as it only activates for faulty samples, is strongly correlated with the residual error, and does not activate for correct samples. By fine-tuning the last layer with QP in the final step, we can address any detected unsafe behavior. Our method ensures that safety constraints are met for the faulty samples employed during the repair process. Moreover, Fig. 1 (c) demonstrates the generalization of the repair to the faulty region shown in Fig. 1 (a).

Figure 2 also demonstrates a test case of our repair method for the control of a lower-leg prosthesis device. The DNN controller returns the ankle angle given the lower and upper limb's angle and angular velocities. The original network is trained using imitation learning with a walking dataset from a healthy subject, collected under Institutional Review Board (IRB) approval. Using repair we aim to constrain the control outputs to stay within $[-5, 10]$ *rad*. Figure 2 (a) illustrates that the control signal (shown in black) stays within the desired bounds after repair. Moreover, the units are effectively trained to activate only in the faulty regions, as shown in Fig. 2 (b) and (c). Therefore, the error between the repaired signal and the original signal is nearly zero for the originally correct inputs.

As we presented in this paper, our safety-aware expansion technique provides an effective solution for addressing safety violations in DNNs. This method can be applied across diverse safety-critical domains, from repairing classification errors to ensuring safety in robot learning applications. By augmenting neural network architectures with additional hidden units and employing fine-tuning of the last layer through Quadratic Programming (QP), we address a detected faulty behavior with safety guarantees. Compared to the other architectural extension techniques for repair, our method offers a natural way to extend the network using the same activation functions as the other neurons, making it applicable to networks with different types of activation functions. Furthermore, our approach encourages the maintenance of the network's original performance in the faulty regions while enforcing hard constraints on the output. We achieve this using an efficient algorithm that only trains the incoming weights of newly added units using Gradient Descent and fine-tunes the last layer with QP. The QP can be solved in polynomial time [13].

In our future research, we will provide additional statistical insights by testing of our method across a diverse set of repair benchmarks. Furthermore, we intend to extend our method's capabilities by assessing its performance over higher-order constraints. Finally, we will explore the potential for repair by adding neurons both in depth and width.

REFERENCES

[1] K. D. Julian and M. J. Kochenderfer, "Guaranteeing safety for neural network-based aircraft collision avoidance systems," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, 2019, pp. 1–10.

[2] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[3] A. Sinitsin, V. Plokhotnyuk, D. Pyrkin, S. Popov, and A. Babenko, "Editable neural networks," in *International Conference on Learning Representations*, 2019.

[4] X. Ren, B. Yu, H. Qi, F. Juefei-Xu, Z. Li, W. Xue, L. Ma, and J. Zhao, "Few-shot guided mix for dnn repairing," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 717–721.

[5] X. Yang, T. Yamaguchi, H.-D. Tran, B. Hoxha, T. T. Johnson, and D. Prokhorov, "Neural network repair with reachability analysis," *arXiv preprint arXiv:2108.04214*, 2021.

[6] Z. Liang, T. Wu, C. Zhao, W. Liu, B. Xue, W. Yang, and J. Wang, "Repairing deep neural networks based on behavior imitation," *arXiv preprint arXiv:2305.03365*, 2023.

[7] G. Dong, J. Sun, X. Wang, X. Wang, and T. Dai, "Towards repairing neural networks correctly," in *IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 714–725.

[8] B. Goldberger, G. Katz, Y. Adi, and J. Keshet, "Minimal modifications of deep neural networks using verification," in *23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, vol. 73, 2020, pp. 260–278.

[9] K. Majd, G. M. Clark, T. Khandait, S. Zhou, S. Sankaranarayanan, G. Fainekos, and H. Amor, "Safe robot learning in assistive devices through neural network repair," in *6th Annual Conference on Robot Learning*. PMLR, 2022.

[10] M. Sotoudeh and A. V. Thakur, "Provable repair of deep neural networks," in *42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021, pp. 588–603.

[11] F. Fu and W. Li, "Sound and complete neural network repair with minimality and locality guarantees," in *10th International Conference on Learning Representations (ICLR)*, 2021.

[12] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," *Advances in neural information processing systems*, vol. 2, 1989.

[13] Y. Ye and E. Tse, "An extension of karmarkar's projective algorithm for convex quadratic programming," *Mathematical programming*, vol. 44, pp. 157–179, 1989.