2024 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-7488-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMLA61862.2024.00137

Approximate Bisimulation Relation Restoration for Neural Networks Based On Knowledge Distillation

Jie Chen[*], Zihao Mo[†], Tao Wang[*], Weiming Xiang[†]
[*]School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China
[†]School of Computer and Cyber Sciences, Augusta University, Augusta, GA, 30901, USA

***Abstract*—This paper employs knowledge distillation to optimize neural network compression processes via reducing the approximate bisimulation error between two neural networks. The paper calculates the approximate bisimulation error between two neural networks and derives the relationship between the approximate bisimulation error and the soft loss of knowledge distillation processes. Then, we propose a knowledge distillation optimization framework to further reduce the approximate bisimulation error between the original neural network and its compressed version. This method can significantly enhance the trustworthiness of the neural network compression methods as the approximate bisimulation error is reduced.**

***Index Terms*—Approximate Bisimulation Relation, Knowledge Distillation, Neural Network Restoration, Reachability.**

I. Introduction

To address the complexity of external environments, high dimensionality, and uncertainty of control systems, neural networks have continued to expand in scale and complexity. While this expansion improves performance and accuracy, it also poses significant challenges to the computational and storage capacities of devices. This is especially true in scenarios involving high computational loads, storage costs, and complex models, where the efficient deployment of models faces major challenges [1]. Additionally, this increases the time cost of safety verification. Verifying even simple properties is an NP-complete problem [2], and verifying neural networks remains a significant challenge. The complex structure and activation functions of large neural networks make them nonlinear, non-convex, and difficult to understand, verification methods are developed to verify properties of neural network systems [3]. To address the challenges posed by large-scale and highly complex neural networks, we compress the original neural network while ensuring that the output of the compressed neural network maintains a high level of matching with the output of the original network within a specified accuracy threshold.

Experimental evidence shows that by pretraining on a large-scale neural network, smaller neural networks can achieve results comparable to those of larger networks [4]. This suggests that a complex model in terms of scale does not necessarily imply complex representations. In other words, complex tasks can be learned using simpler networks that can replace larger networks. Mainstream model compression techniques include knowledge distillation to train lightweight model architectures, pruning to remove redundant neurons, and quantization to reduce representation precision [5], [6]. Knowledge distillation, in particular, can be flexibly applied across a range of important learning paradigms [7], highlighting its versatility. It can be integrated into major learning paradigms, including adversarial learning, automated machine learning, label noise filtering, lifelong learning, and reinforcement learning. The combination of knowledge distillation with other learning methods holds promise for addressing forthcoming real-world challenges, underscoring the considerable potential of knowledge distillation in model compression. However, the optimization problem in knowledge distillation remains challenging to solve effectively using current methods. The matching accuracy of compressed models may be relatively low, whether on training data or test data [8].

Therefore, in this paper, we use the knowledge distillation model compression algorithm to enhance the similarity between two neural networks which increases the trustworthiness of neural network compression. Reachability analysis allows the calculation of the maximum difference between the outputs of the compressed neural network and the original neural network, referred to as the approximate bisimulation error [9]. This error reflects the similarity between the neural networks. By targeting specific training to reduce the approximate simulation error, the accuracy of knowledge distillation-based model compression can be improved.

The structure of this paper is as follows. Section II introduces the approximate bisimulation bisimulation relation and knowledge distillation, and reveals the relationship between them. In Section III, the approximate bisimulation bisimulation relation restoration framework based on knowledge distillation is developed. Section IV evaluates the developed approach via experiment. Finally, Section V presents the conclusions.

II. Approximate Bisimulation Relation and Knowledge Distillation

A. Approximate Bisimulation Relation

Given a neural network in the description of $y = \Phi(u)$ where, $u \in \mathbb{R}^{n_u}$ is the input and $y \in \mathbb{R}^{n_y}$ is the output, and $\Phi : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ denotes the neural network. the reachable set of neural network Φ is defined as follows.

Definition 1: [9] Given a neural network Φ and an input set $\mathcal{U} \subset \mathbb{R}^{n_u}$, the following set

$$\mathcal{Y} = \left\{y \in \mathbb{R}^{n_y} \mid y = \Phi(u),\ u \in \mathcal{U}\right\}, \tag{1}$$

is called the output reachable set of neural network Φ.

1946-0759/24/$31.00 ©2024 IEEE
DOI 10.1109/ICMLA61862.2024.00137

Consider two neural networks Φ_1 and Φ_2, the approximate bisimulation relation between two neural networks formally characterizes the the maximum difference between outputs.

Definition 2: [9] Given two neural networks Φ_1 and Φ_2 with an input set $\mathcal{U} \subset \mathbb{R}^{n_u}$, we define the following metric to characterize the output discrepancy of two neural networks

$$d(\Phi_1, \Phi_2) = \begin{cases} \rho(y_1, y_2) & \text{if } u_1 = u_2 \\ +\infty & \text{otherwise} \end{cases}, \tag{2}$$

where

$$\rho(y_1, y_2) = \sup_{y_1 \in \mathcal{Y}_1, y_2 \in \mathcal{Y}_2} \|y_1 - y_2\|, \tag{3}$$

in which $\mathcal{Y}_i$, $i \in \{1, 2\}$ are output reachable sets defined by (1) for two neural networks .

Definition 3: [9] Given two neural networks Φ_1 and Φ_2 with an input set $\mathcal{U} \subset \mathbb{R}^{n_u}$, and let $\varepsilon \geq 0$ and, a relation $\mathfrak{R}_\varepsilon \subset \mathbb{R}^{n_{y_1}} \times \mathbb{R}^{n_{y_2}}$ is called an approximate simulation relation between Φ_1 and Φ_2, of precision ε, if for all $(y_1, y_2) \in \mathfrak{R}_\varepsilon$

1) $d(\Phi_1(u), \Phi_2(u)) \leq \varepsilon$, $\forall u \in \mathcal{U}$;
2) $\forall u \in \mathcal{U}$, $\forall \Phi_1(u) \in \mathcal{Y}_1$, $\exists \Phi_2(u) \in \mathcal{Y}_2$ such that $(\Phi_1(u), \Phi_2(u)) \in \mathfrak{R}_\varepsilon$;
3) $\forall u \in \mathcal{U}$, $\forall \Phi_2(u) \in \mathcal{Y}_2$, $\exists \Phi_1(u) \in \mathcal{Y}_1$ such that $(\Phi_1(u), \Phi_2(u)) \in \mathfrak{R}_\varepsilon$

and we say neural networks Φ_1 and Φ_2 are approximately bisimilar with precision ε, denoted by $\Phi_1 \sim_\varepsilon \Phi_2$. Furthermore, the approximate bisimulation error is defined by

$$d(\Phi_1, \Phi_2) = \sup\{\varepsilon \mid \Phi_1 \sim_\varepsilon \Phi_2\}. \tag{4}$$

In [9]–[12], a neural network merging approach was developed to compute the approximate bisimulation error using reachability tools for various neural networks such as feedforward neural networks, convolutional neural networks, etc.

B. Knowledge Distillation

To perform knowledge distillation, it is essential to first define the concept of knowledge, which can be abstractly described as the mapping relationship from input to output. Distillation, in this context, refers to the methodology of transferring knowledge from large-scale neural networks to more compact neural network models [13]. This process involves the utilization of a teacher-student model, where the pre-trained large-scale model functions as the teacher, Φ_L, and the compact model serves as the student, Φ_S. The core of this approach lies in leveraging the soft targets predicted by Φ_L, in conjunction with hard targets, to train Φ_S. Knowledge distillation employs a loss function, measured through cross-entropy relative to the output distribution, to quantify the compression effect. A typical knowledge distillation framework is illustrated in Fig. 1.

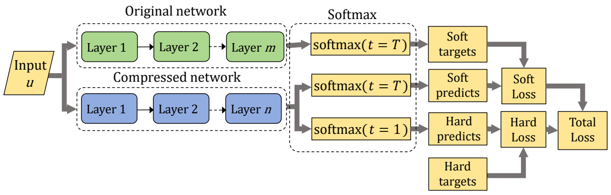

Fig. 1. General Knowledge Distillation Framework.

The use of hard-target supervision is deemed necessary as soft-targets may also incur prediction errors, necessitating correction through hard-targets. L_H measures the loss between the ground truth values and the predicted values. Regarding L_S, the reachable input set is utilized to generate soft targets by employing Φ_L at a high temperature *T*. At temperature T, the cross-entropy between the outputs of Φ_L and Φ_S yields the soft loss. The cross-entropy between the original ground truth values and the outputs of the compressed neural network serves as the hard loss. The cross-entropy between the softmax output and the soft-targets at the same elevated temperature *T* is outlined below

$$L_S = -\textstyle\sum_i p_i \cdot \log(q_i), \tag{5}$$

where

$$p_i = \frac{\exp(y_{L_i}/T)}{\sum_j \exp(y_{L_j}/T)}, \; q_i = \frac{\exp(y_{S_i}/T)}{\sum_j \exp(y_{S_j}/T)}, \tag{6}$$

in which p_i represents the value in the softmax output of the original neural network Φ_L at temperature *T*, q_i represents the value in the softmax output of the compressed neural network Φ_S at temperature *T*. The hard loss in the diagram can be represented as

$$L_H = -\textstyle\sum_i c_i \cdot \log(q_i), \tag{7}$$

where c_i represents the reference value. $c_i \in [0, 1]$, takes the value of 1 for corresponding outputs and 0 otherwise, and q_i for L_H is with $T = 1$. These two losses are combined and weighted to obtain the total loss.

To minimize the total loss, the knowledge distillation process will reduce both soft and hard losses. The hard loss reflects the prediction accuracy as it is the cross-entropy between the original ground truth values and the outputs as in a normal neural network training process. The soft loss is featured by knowledge distillation to represent the difference between the teacher model Φ_L and the student model Φ_S. In the following, we will explore the relationship between knowledge distillation and approximate bisimulation relation, particularly in the view of soft loss.

Taking the partial derivative of L_S with respect to each logit y_{S_i} further yields:

$$\frac{\partial}{\partial y_{S_i}} L_S = \frac{1}{T} \left(\frac{\exp(y_{S_i}/T)}{\sum_j \exp(y_{S_j}/T)} - \frac{\exp(y_{L_i}/T)}{\sum_j \exp(y_{L_j}/T)} \right), \tag{8}$$

which can be approximated as below when the temperature parameter T is chosen to be relatively large

$$\frac{\partial}{\partial y_{S_i}} L_S \approx \frac{1}{T} \left(\frac{1 + y_{S_i}/T}{N + \sum_j y_{S_j}/T} - \frac{1 + y_{L_i}/T}{N + \sum_j y_{L_j}/T} \right). \tag{9}$$

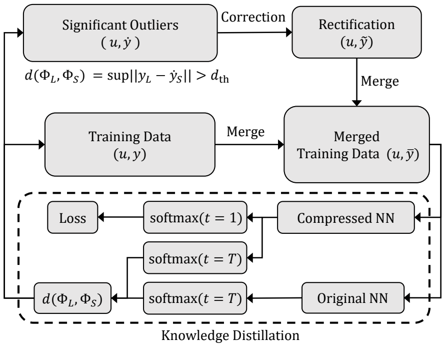

Fig. 2. Approximation Bisimulation Relation Restoration Framework.

Assuming that the logits are centered around zero for each sample, we can yield a further approximation as follows

$$\frac{\partial}{\partial y_{S_i}} L_S \approx \frac{1}{NT^2}(y_{L_i} - y_{S_i}). \tag{10}$$

As a result, the knowledge distillation is equivalent to minimizing $(y_{L_i} - y_{S_i})^2$ in high temperature. On the other hand, Definition 3 of approximate simulation relation error between Φ_L and Φ_S implies that $\varepsilon \approx \max_i \|y_{L_i} - y_{S_i}\|$. Therefore, it implies that the knowledge distillation can reduce and minimize the approximate simulation relation error ε.

III. Approximate Bisimulation Relation Restoration

To address the problem of approximation bisimulation relation restoration, we define the input domain with approximation bisimulation error between two neural networks as the large error input set domain, denoted as $\mathcal{U}_e$. The definition can be tailored to specific application scenarios, for example, if the error between the outputs of two neural networks exceeds $d(\Phi_L, \Phi_S)/2$, the input is considered part of $\mathcal{U}_e$.

Definition 4: Consider two neural networks Φ_L and Φ_S, if their outputs y_L, $\dot{y}_S$ for input u satisfies

$$d(\Phi_L, \Phi_S) = \|y_L - \dot{y}_S\| \geq d_{\mathrm{th}}, \tag{11}$$

where the data pair $(u, \dot{y}_S)$ is referred to as significant outliers. d_{th} represents the threshold to determine significant outliers.

As illustrated in Fig. 2, after determining significant outliers, we have to correct them to data pairs $(u, \tilde{y}_S)$ to avoid data variability and merge them into the retraining dataset as pairs of $(u, \bar{y}_S)$ for knowledge distillation. Specifically in the correction process, we multiply the correction $d(\Phi_L, \Phi_S)$ by a coefficient in the range of $(0, 1)$ to mitigate significant outliers data variability. The function for correct significant outliers is in the following form of

$$\tilde{y}_S = \dot{y}_S + \beta \cdot \sup \|y_L - \dot{y}_S\|, \tag{12}$$

where $\beta \in (0, 1)$ denotes the correction coefficient.

The approximation bisimulation relation restoration procedure aims to reduce the approximation bisimulation error ε, which can be summarized as follows:

- **Initialization:** For a given original neural network Φ_L and its compressed version Φ_S, the approximation bisimulation error ε between the two is calculated. Input-output data pairs (u, y_S) corresponding to the approximation bisimulation error exceeding the threshold d_{th} are considered as significant outliers $(u, \dot{y}_S)$, requiring correction.
- **Correction:** Identify significant outliers $(u, \dot{y}_S)$, apply the correction function (12) to rectify them to be correcting data pairs $(u, \tilde{y}_S)$, and construct a retraining dataset $(u, \bar{y}_S)$ with both corrected and normal data pairs.
- **Distillation:** By utilizing the distillation-based restoration, $\hat{\Phi}_S$ is repeatedly trained by adjusting its weights and biases so that the output y_S of $\hat{\Phi}_S$ converges as closely as possible to the output y_L of Φ_L, aiming to minimize the approximate bisimulation error between $\hat{\Phi}_S$ and Φ_L. This process is intended to make the new network $\hat{\Phi}_S$ closely resemble Φ_L.
- **Evaluation:** After distilling the compressed model, calculate the approximation bisimulation error ε and compare it with the predefined threshold. Until the condition (11) is not satisfied, the evaluating, correction, and distillation process concludes. Otherwise, iterate through the distillation process again.

IV. Experiments and Evaluation

In random numerical experiments, an original neural network Φ_L of the size of $2 \times 20 \times 20 \times 20 \times 2$ is randomly constructed. Subsequently, a smaller-scale neural network Φ_S was obtained through knowledge distillation, with a size of $2 \times 5 \times 5 \times 2$. The activation function used for hidden layers is the Rectified Linear Unit (ReLU).

The small-scale neural network Φ_S obtained from the original neural network employs the approximate bisimulation error $d(\Phi_L, \Phi_S)$ from Eq. (4) between the two neural networks. The input set is chosen within the following interval $\mathcal{U} \triangleq [0, 0.5] \times [0, 0.5]$. Using the reachable set computation tool NNV [14], the approximate bisimulation error ε between the two neural networks can be computed through the application of approximation bisimulation error $d(\Phi_L, \Phi_S) = 0.10714$. Hence, it can be inferred that for any sampled input data from the input set $\mathcal{U}$, the distance between the resulting outputs is guaranteed to be less than or equal to 0.10714.

Fig. 3 depicts the simulated output scatter plots for the two neural networks. Observing the output data points generated by both neural networks using the same set of input data, we found that all the output data points from the large-scale neural network Φ_L fall within the circles centered around the corresponding output data points generated by the small-scale neural network Φ_S, with a radius equal to the approximate bisimulation error $d(\Phi_L, \Phi_S)$.

After performing knowledge-distillation-base restoration on the compressed neural network, denoted as $\hat{\Phi}_S$, the sampled outputs are shown in Fig. 4. The approximate bisimulation error was reduced from 0.10714 to 0.068976, which is a 35.62% decrease from the original one as shown in Table I.

TABLE I
APPROXIMATE BISIMULATION ERRORS

	Before Restoration	**After Restoration**
Error $d(\Phi_L, \Phi_S)$	0.10714	0.068976

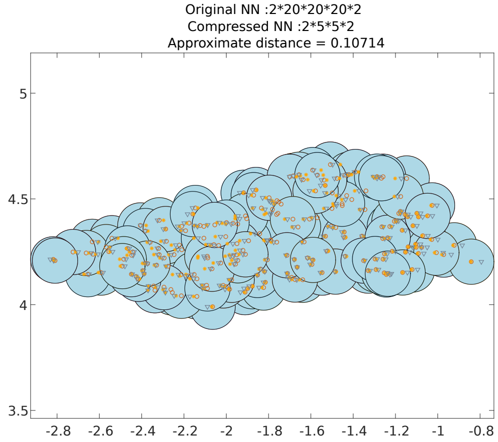

Fig. 3. Approximate Bisimulation Error Before Restoration

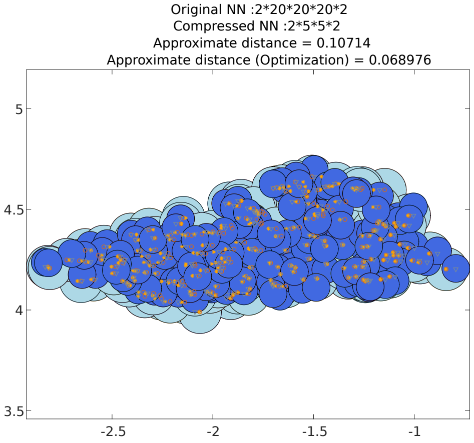

Fig. 4. Approximate Bisimulation Error After Restoration

For each sampled output data point of the optimized compressed neural network, a deep blue circle is drawn with a radius equal to the new approximate bisimulation error, denoted as $d(\Phi_L, \hat{\Phi}_S)$. It can be observed that the range of sampled outputs fitted by the optimized compressed neural network (deep blue region) is smaller than that of the original compressed neural network (light blue region), indicating a smaller approximate bisimulation error for the optimized compressed neural network. This suggests a better performance to reduce the approximate bisimulation error, thereby demonstrating the effectiveness of approximation bisimulation relation restoration algorithms based on knowledge distillation.

V. Conclusions

This paper addresses the neural network restoration through a knowledge distillation model compression algorithm. By calculating the approximate bisimulation error between the two neural networks, the relationship between this error and the knowledge distillation loss is derived which implies that knowledge distillation can reduce approximate bisimulation error. Subsequently, a knowledge distillation optimization framework is developed to minimize the approximate bisimulation error, significantly enhancing the similarity and trustworthiness between the compressed and original models. Finally, the optimized neural network compression model is applied to randomly generated neural networks, validating the effectiveness of the compression optimization method.

Acknowledgment

This work was partially supported by the National Science Foundation, under NSF CAREER Award 2143351, NSF CNS Award 2223035, and NSF IIS Award 2331938.

References

[1] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” *arXiv preprint arXiv:1510.00149*, 2015.

[2] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient smt solver for verifying deep neural networks,” in *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pp. 97–117, Springer, 2017.

[3] H.-D. Tran, W. Xiang, and T. T. Johnson, “Verification approaches for learning-enabled autonomous cyber–physical systems,” *IEEE Design & Test*, vol. 39, no. 1, pp. 24–34, 2020.

[4] J. Ba and R. Caruana, “Do deep nets really need to be deep?,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[5] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, “A survey of model compression and acceleration for deep neural networks,” *arXiv preprint arXiv:1710.09282*, 2017.

[6] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, “Model compression and hardware acceleration for neural networks: A comprehensive survey,” *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485–532, 2020.

[7] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[8] S. Stanton, P. Izmailov, P. Kirichenko, A. A. Alemi, and A. G. Wilson, “Does knowledge distillation really work?,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 6906–6919, 2021.

[9] W. Xiang and Z. Shao, “Approximate bisimulation relations for neural networks and application to assured neural network compression,” in *2022 American Control Conference (ACC)*, pp. 3248–3253, IEEE, 2022.

[10] Z. Mo and W. Xiang, “Maximum output discrepancy computation for convolutional neural network compression,” *Information Sciences*, vol. 665, p. 120367, 2024.

[11] W. Xiang and Z. Shao, “Safety verification of neural network control systems using guaranteed neural network model reduction,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 1521–1526, IEEE, 2022.

[12] W. Cooke, Z. Mo, and W. Xiang, “Guaranteed quantization error computation for neural network model compression,” in *2023 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1–4, IEEE, 2023.

[13] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[14] H.-D. Tran, X. Yang, D. Manzanas Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, and T. T. Johnson, “NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems,” in *International Conference on Computer Aided Verification*, pp. 3–17, Springer, 2020.