DOI: 10.54254/2755-2721/76/20240597

Semantic-based neural network repair for AI models in dynamic environments

Xiaoyu Chen

Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, 999078, China

yuge030226@163.com

Abstract. This paper addresses a critical challenge in the field of artificial intelligence (AI) and deep learning (DL) - the debugging and repair of AI models within dynamic environments. The rapid development of frameworks like TensorFlow or PyTorch can introduce version incompatibilities, potentially breaking existing code. To tackle this issue, it can be seen that a semantic-based neural network repair method that leverages an existing semantics called ExAIS is proposed, defined in the Prolog logic programming language. This method identifies architectural and parameter errors in AI models and generates debug messages to facilitate the repair process. Key contributions of this paper include introducing a novel semantic-based AI model repair method capable of suggesting multiple changes to rectify inefficiencies in AI models. We evaluate the effectiveness of our approach in a real-world context by collecting a set of real bugs from AI developers. The background section provides an overview of AI framework testing, machine learning, neural architecture search (NAS), and deep learning. The methodology section details the repair method based on ExAIS semantics, covering error identification and repair suggestions as the two main steps. The evaluation section showcases the repair effectiveness on automatically generated and manually written models, emphasizing the applicability and efficiency of the approach.

Keywords: Neural Network Repair, Semantic-Based Debugging, ExAIS, Deep Learning, Dynamic Environment.

1. Introduction

Artificial intelligence (AI) seeks to replicate natural intelligence or imbue machines with the ability to learn behaviors. Deep learning (DL), a key approach in AI, employs artificial neural networks to simulate the neurons of the human brain. However, debugging AI systems poses significant challenges, with one major difficulty arising from the dynamic nature of AI frameworks like TensorFlow or PyTorch. As these frameworks undergo rapid development, newer versions may lack backward compatibility, potentially breaking existing code. This report introduces an innovative semantic-based neural network repair method tailored to aid AI developers in addressing prevalent architectural and parameter errors in AI models. Drawing upon an established semantics termed ExAIS, which delineates TensorFlow layers using the Prolog logic programming language, our approach excels at identifying model errors and generating debug messages crucial for facilitating effective repairs. The key contributions of our report include: Introducing a novel semantic-based AI model repair method capable of suggesting multiple changes to rectify inefficiencies in AI models. Collecting a set of real bugs from AI developers to evaluate the efficacy of our approach in a real-world context.

2. Background

This section provides an overview of the techniques and underlying semantics supporting the repair approach.

2.1. Artificial intelligence framework testing

Testing of artificial intelligence (AI) frameworks is a crucial aspect of ensuring their reliability and effectiveness. AI framework testing involves assessing the performance, functionality, and compatibility of various AI frameworks, such as TensorFlow or PyTorch, under different conditions and scenarios. The primary objective of AI framework testing is to identify and address bugs, errors, and inconsistencies that may arise during the development and deployment of AI systems [1].

One of the key challenges in AI framework testing is the dynamic nature of these frameworks [2]. As AI frameworks evolve rapidly to incorporate new features, optimizations, and bug fixes, ensuring backward compatibility becomes essential. Changes in framework versions can potentially introduce incompatibilities with existing codebases, leading to errors and failures in AI applications. Therefore, thorough testing procedures are necessary to detect and mitigate such issues.

2.2. Collecting real bugs for evaluation

In order to evaluate the real-world effectiveness of the proposed repair method, the authors embarked on a comprehensive process of collecting actual bugs encountered by AI developers during their development endeavors [3]. This involved reaching out to AI development teams, both within academia and industry, to solicit reports of bugs and errors encountered in their AI model development projects.

The process of bug collection was conducted with meticulous attention to ensure the diversity and representativeness of the collected bugs. Various channels were utilized for bug solicitation, including online forums, social media platforms, developer communities, and direct communication with AI development teams. Additionally, efforts were made to gather bugs from a wide range of AI domains, encompassing computer vision, natural language processing, reinforcement learning, and other relevant areas [4].

Once a sufficient number of bug reports were gathered, they were carefully curated and categorized based on various attributes, such as the type of error, the affected AI framework, the complexity of the model, and the domain of application. This categorization process helped ensure that the collected bugs represented a diverse and comprehensive set of real-world challenges faced by AI developers.

Furthermore, the authors collaborated closely with AI development teams to validate and verify the authenticity of the collected bugs. This involved reproducing the reported errors in controlled environments, analyzing the underlying causes, and confirming the feasibility of repairing the bugs using the proposed semantic-based neural network repair method.

Overall, the process of collecting real bugs for evaluation was characterized by thoroughness, collaboration, and attention to detail. By leveraging a diverse set of real-world bug reports, the authors aimed to provide meaningful insights into the practical application and performance of the proposed repair method.

2.3. Machine learning

Machine learning is a field within artificial intelligence (AI) and computer science that revolves around using data and algorithms to mimic the learning processes of humans [5]. The goal is to enable machines to learn from experiences and gradually improve their accuracy in performing specific tasks [6]. In simpler terms, it's like teaching a computer to learn and make decisions based on patterns it discovers in data.

Neural Architecture Search (NAS) is an automated technique used in machine learning to design artificial neural networks (ANN). Think of ANNs as structures inspired by the human brain. NAS automates the process of finding the best architecture for a neural network, surpassing what humans might design manually. It's like letting the computer figure out the most effective way for a neural network to function.

Deep learning is a subset of machine learning that specifically deals with artificial neural networks containing multiple layers [7]. These layered networks, often referred to as deep neural networks, aim to extract complex features or patterns from data. In simpler terms, deep learning involves teaching a computer to understand intricate aspects of information by organizing it into layers, much like how our brain processes information in layers.

For someone new to these concepts, it's helpful to picture machine learning as the broader field, with neural architecture search being a method for optimizing the design of neural networks within machine learning. Deep learning is then a specialized approach within machine learning that focuses on these complex, layered neural networks.

3. Methodology

Our repair method, which relies on ExAIS semantics, establishes the capabilities and prerequisites for various TensorFlow layers. The process involves two primary steps: identifying errors and suggesting repairs.

3.1. Recognition error

In this phase of our methodology, we utilize ExAIS to execute the AI model and identify errors. ExAIS allows us to conduct precondition checks, which involve applying logical rules to assess whether the specific requirements of each TensorFlow layer are fulfilled. These requirements encompass various aspects such as input shape, data type compatibility, and parameter constraints.

For instance, if a convolutional layer expects input data to be of a certain shape and data type but receives incompatible input, ExAIS will flag this as an error during the precondition check. Similarly, if a fully connected layer requires a specific number of neurons in the input layer but receives input with a different dimensionality, ExAIS will identify this as an error [8].

Upon detecting an error, our methodology generates detailed debug messages to provide insights into the nature and location of the problem. These debug messages may include information about the specific layer or operation where the error occurred, the expected conditions, and the actual conditions encountered. By providing such detailed information, we aim to facilitate the process of locating and understanding the underlying issues in the AI model effectively.

3.2. Repair suggestions

In the Repair Suggestions phase, our approach provides recommendations for modifying the architecture and parameters of the AI model to address the identified errors. The primary objective is to minimize changes while ensuring that the repaired model remains similar to the original one. This ensures that the model's performance and functionality are preserved to the greatest extent possible.

Recommendations for modifying the model's architecture may include adjustments to the structure of neural network layers, such as adding or removing layers, changing layer types, or modifying layer configurations. These modifications are designed to address architectural errors identified during the Recognition Error phase while maintaining the overall structure and functionality of the model.

Similarly, recommendations for modifying the model's parameters may involve adjusting the values of weights, biases, activation functions, or other parameters associated with neural network layers. These adjustments are aimed at correcting parameter-related errors identified during the Recognition Error phase while preserving the model's ability to accurately learn from data and make predictions.

Throughout the Repair Suggestions phase, our approach prioritizes the minimization of alterations to the original model to prevent unnecessary changes that could potentially impact its performance or functionality negatively. By carefully selecting repair recommendations that preserve the model's essential characteristics and behaviors, we aim to ensure that the corrected model remains effective and reliable for its intended application.

DOI: 10.54254/2755-2721/76/20240597

3.3. Methodology for AI model repair

In university studies, students learn about the development and debugging of AI models. The mentioned methodology introduces a semantic-based approach for repairing AI models. Semantics, often covered in programming language courses, involves the meaning of code. The methodology combines this understanding with TensorFlow, a popular deep learning framework. Students studying software engineering or AI-related disciplines would find this methodology relevant, as it addresses the dynamic nature of AI frameworks and the need for efficient model repair.

Evaluation of Automatically Generated Models: In AI courses, students often encounter the challenge of evaluating models, especially those generated automatically. This involves assessing the model's performance, identifying errors, and proposing repairs. The mentioned evaluation step aligns with the broader concept of model evaluation in AI studies. It emphasizes the importance of automated testing and repair methods, showcasing practical applications in real-world scenarios.

This integration of concepts reflects the interdisciplinary nature of AI, combining knowledge from mathematics, computer science, and engineering to develop and enhance intelligent systems.

Our method successfully repairs 100% of randomly generated models with errors, completing the repair in an average time of 21.08 seconds: This statement indicates that the proposed repair method is highly effective. When applied to randomly generated AI models that contain errors, the method successfully identifies and fixes all issues. Moreover, it does so efficiently, with an average repair time of 21.08 seconds, showcasing the speed and reliability of the approach.

Fix Manually Written Models: This part highlights the versatility of the repair method. It demonstrates its practicality by successfully fixing 15 out of 16 problematic models that were manually written. The term "manually written" suggests that these models were created by developers without automated generation, emphasizing the method's adaptability to various scenarios [8].

4. Applicability and efficiency of the method

This segment underscores that the proposed approach is not limited to specific types of AI models. It is applicable to various AI model types, indicating its versatility. Additionally, the text emphasizes that the method efficiently handles larger-scale networks, showcasing its scalability and efficiency in addressing complex and sizable models.

The method is commended for going beyond addressing tensor shape errors, demonstrating its ability to handle a wide range of issues with minimal changes to standard layers. Moreover, it is noted as the first automatic semantic-based repair method for AI model errors, indicating innovation in its approach to identifying and fixing problems.

In summary, the provided text is presenting the impressive capabilities of the proposed AI model repair method, emphasizing its effectiveness, versatility, and efficiency in comparison to other existing methods.

In the context of related work, the ExAIS repair method stands out when compared to other existing methods. It is distinguished by several unique features that set it apart and contribute to its effectiveness in addressing various challenges encountered in AI model development.

Directness: The ExAIS repair method is recognized for its direct approach to identifying and rectifying errors in AI models. Unlike some existing methods that may involve complex and indirect processes, ExAIS offers a straightforward mechanism for pinpointing errors and providing repair suggestions. This directness streamlines the debugging process, allowing for efficient resolution of issues without unnecessary complications.

Versatility in Addressing Different Issues: Another notable feature of the ExAIS repair method is its versatility in handling a wide range of issues encountered in AI model development. Whether the errors are related to architectural discrepancies, parameter inconsistencies, or other underlying issues, ExAIS is capable of effectively identifying and addressing them. This versatility makes the method applicable across various domains and scenarios, enhancing its utility for AI developers facing diverse challenges [9].

DOI: 10.54254/2755-2721/76/20240597

Unique Ability to Identify Errors Without the Need for Error Messages: One of the key strengths of the ExAIS repair method is its ability to identify errors without relying solely on error messages or explicit indicators. By leveraging the semantics encoded in ExAIS, the method can analyze the underlying structure and behavior of AI models to detect discrepancies and anomalies. This unique approach reduces reliance on error messages, which may not always be comprehensive or accurate, thereby improving the reliability and accuracy of error detection [10].

The method successfully rectifies faults in AI models, achieving a remarkable average repair time of 21.08 seconds. The mention of real-world neural network bugs from developers serves as a practical validation of the method's applicability. The authors express optimism about the broad application prospects of their approach and outline plans to extend semantic-based repair techniques to other AI models.

5. Conclusion

This paper provides a comprehensive exploration of the intricate relationships and extensions within AI frameworks and AI systems. The focus spans various components including machine learning (ML), deep learning (DL), and other relevant areas. Through a meticulous examination, the paper elucidates how these elements interact and evolve within the dynamic landscape of artificial intelligence. It presents novel approaches such as semantic-based neural network repair methods, exemplifying the advancements made in addressing challenges encountered in AI model development. Furthermore, the paper offers valuable insights into the practical application and effectiveness of these methods through rigorous evaluation processes.

While the paper provides a comprehensive overview of AI frameworks, machine learning, and deep learning, there may be opportunities to delve deeper into specific topics. For example, further exploration into advanced techniques within machine learning, such as transfer learning, ensemble methods, or Bayesian approaches, could enhance the paper's depth and breadth.

The paper could benefit from more extensive discussions on the practical applications of the proposed methods. Providing detailed case studies or real-world examples where the semantic-based neural network repair method has been successfully applied would strengthen the paper's credibility and relevance to AI practitioners.

While the paper briefly mentions future prospects for extending semantic-based repair techniques to other AI models, a more detailed exploration of potential research directions and emerging trends in the field could provide valuable guidance for future studies. Identifying specific areas where semantic-based methods could be applied and outlining potential challenges and opportunities would enrich the paper's discussion on future directions.

Overall, by addressing these areas for improvement, the paper can enhance its contributions to the field of AI research and provide a more comprehensive understanding of semantic-based neural network repair methods in dynamic environments.

References

[1] Li Hangyu; Wang Nannan; Zhu Mingrui; Yang Xi; Gao Xinbo. Review of Research Progress in Neural Structure Search. Journal of Software. 2022.
[2] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. Foundations of Neural Networks. Deep Learning. MIT Press, 2016.
[3] Silver, David et al. Adaptation to Dynamic Environments. Mastering the game of Go without human knowledge. Nature, 2017.
[4] Du Wei; Liu Gongshen. A Review of Backdoor Attacks in Deep Learning. Journal of Information Security. 2022.
[5] D.; Schütze, Hinrich. Semantic Analysis in AI Manning Christopher. Foundations of Statistical Natural Language Processing. MIT Press, 1999.
[6] Hybrid-Augmented Intelligence: Collaboration and Cognition.Bi Geng Cultural Communication

DOI: 10.54254/2755-2721/76/20240597

[7] Deng Feiyue[1]; Ding Hao[2]; Lu Haoyang[2]; Hao Rujiang[2]; Liu Yongqiang. A high-speed railway wheel set bearing fault diagnosis method based on lightweight neural network. Engineering Science Journal,2021.

[8] Li Zhenbo; Li Meng; Zhao Yuanyang; Guo Ruohao; Chen Yaru. Freshness evaluation method of chilled pomfret based on improved VGG-19 convolutional neural network. Transactions of the Chinese Society of Agricultural Engineering, 2021.

[9] Hohman, Frederick; Kahng, Minsuk; Pienta, Robert; Chau, Duen Horng. Semantic-Based Approaches in AI Systems. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. IEEE Transactions on Visualization and Computer Graphics, 2019.

[10] Zhang, Y.; Chen, X.; Zhou, D.; Jordan, M. I. Robust Machine Learning by Median-of-Means: Theory and Practice. Annals of Statistics, 2021.