Provable Fairness for Neural Network Models Using Formal Verification

Giorgian Borca-Tasciuc[1,*,1], Xingzhi Guo[1,1], Stanley Bak[1] and Steven Skiena[1]

[1]*Department of Computer Science, Stony Brook University, NY, USA*

Abstract

Machine learning models, extensively deployed in critical decision-making roles, necessitate verification for potential gender or racial biases from training data. Traditional fairness strategies focus on curating training data and subsequent statistical evaluation of model fairness. We, however, propose techniques that offer formal proof of fairness, leveraging recent advancements in neural network model verification. Our methods offer robust guarantees and uniquely, they eliminate the need for explicit training or evaluation data, often proprietary, to analyze a trained model. Experimental results from the well-known ADULTS dataset demonstrate that appropriate training can decrease unfairness by an average of 65.4% with less than 1% cost in AUC score.

Keywords

Fairness Metrics, Provable Fairness, Neural Network Verification

1. Introduction

Machine learning models are increasingly utilized in critical decision-making tasks, and it is vital to ensure they are devoid of gender or racial biases [1]. Traditional fairness strategies focus on curating training data and post-hoc evaluation on testing data. However, these methods often require proprietary data and do not provide concrete proof of fairness.

We propose techniques to *formally prove* fairness in neural network models, without needing explicit training or testing data [2, 3, 4, 5]. We explore fairness specifications and apply formal methods for *classification tasks with explicit input labels* that encode sensitive properties such as race, gender, or age. Fairness in this context implies that model M should behave similarly across all values of these sensitive fields. Importantly, eliminating these inputs during training does not ensure fairness, as sensitive properties can be inferred from other variables, such as zip codes.

Our approach is primarily applied to fully connected networks with ReLU activation functions [6]. In these networks, binary decision models divide possible inputs into geometric polytopes, and all points within a polytope share the same label. Fairness properties involve demonstrating that these regions are similar for distinct labels or groups (Figure 1). We define

EWAF'23: European Workshop on Algorithmic Fairness, June 07–09, 2023, Winterthur, Switzerland

gborcatasciu@cs.stonybrook.edu (G. Borca-Tasciuc); xingzguo@cs.stonybrook.edu (X. Guo); stanley.bak@stonybrook.edu (S. Bak); skiena@cs.stonybrook.edu (S. Skiena)

https://giorgianb.github.io/ (G. Borca-Tasciuc); https://stanleybak.com/ (S. Bak); https://www3.cs.stonybrook.edu/~skiena/ (S. Skiena)

CEUR Workshop Proceedings (CEUR-WS.org)

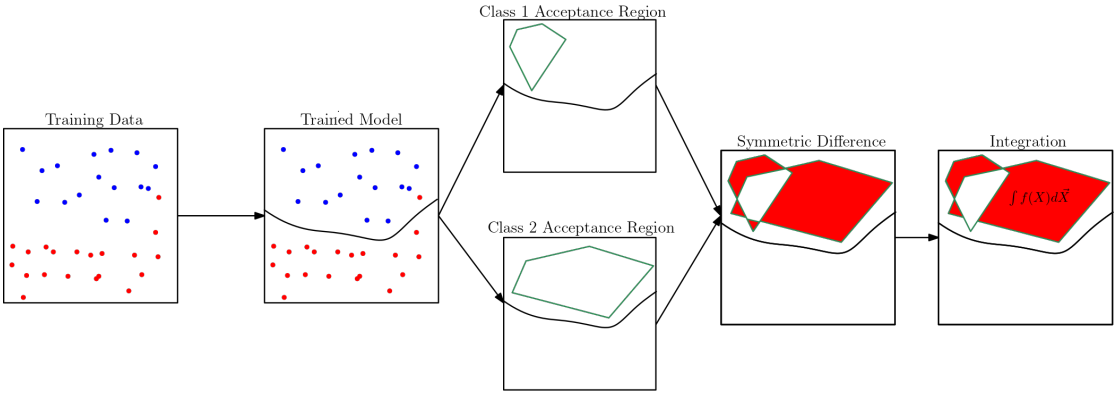

Figure 1: In our verification approach, each partition Θ of the network input space defines a domain for integration over a given input probability density function P, calculating a probability of the input space with a fixed output classification. By repeating and summing over all partitions, we can evaluate fairness metrics for the network.

fairness in M using three provable properties: *Volumetric symmetric difference*, *Probability-weighted symmetric difference*, and *Net Preference*, each depending on specific fairness criteria. Our key contribution is:

- *Provable Fairness Guarantees* ***without*** *Training or Evaluation Data* – We define a class of fairness guarantees for neural classification models that require only black box access to the model. Our metrics either require no training data or knowledge of the class features' distribution, or only require knowledge of the features' probability distribution, which can be estimated using demographic studies.

A full overview of the contributions and a more in-depth analysis of the results is available in the full paper.[7]

2. Related Work

Formal verification methods for neural networks: Formal verification methods conduct set-based analysis of networks, eschewing the execution of individual input samples [8, 2, 9]. Given a neural network $f_{NN} : \mathbb{R}^n \to \mathbb{R}^m$, an input set $X \subseteq \mathbb{R}^n$, and an unsafe set of outputs $U \subseteq \mathbb{R}^m$, the *open-loop NN verification problem* aims to ascertain that for all inputs $x \in X$, the output $f_{NN}(x)$ does not belong to the unsafe set U. The problem is typically resolved by computing the range of the neural network function for an input domain X and checking if the range intersects the unsafe states U [10, 11]. In this paper, we repurpose this range computation approach to examine how probability distributions propagate through the networks.

Two core operations in our work are: (1) computing the range of the neural network as a union of polytopes, and (2) computing the volume of the output polytopes. Unlike the conventional usage of formal verification methods for safety and motion planning, we explore their application for neural network fairness measurement.

Limited research exists on provable fairness. Some frameworks focus on proving *dependency fairness* [12, 13], aiming to ensure outputs are not influenced by certain input features through forward and backward static analysis and input feature partitioning. Another approach [14] emphasizes *individual fairness*, where similar individuals receive similar treatments. Our provable fairness metrics, however, are defined over geometric properties of the network outputs, such as the symmetric difference between the ranges with different values of sensitive inputs.

3. Fairness Verification via Formal Methods

3.1. Acceptance Region

To evaluate model fairness, we scrutinize the acceptance regions, denoted as $\rho(C)$. Given a neural network assigning the label $f(\vec{x})$ for input value $\vec{x}$, $\rho(C)$ represents the input space where $\vec{x} \in C \wedge f(\vec{x}) = \ell$. The defining label ℓ is user-determined based on specific neural network properties of interest.

3.2. Fairness Metrics

Our fairness evaluation relies on metrics embodying legal notions of disparate treatment and impact. We develop and use three metrics: WSD (weighted symmetric difference), VSD (volumetric symmetric difference), and NP (net preference). WSD precisely captures the proportion of individuals affected by class-specific rules in an unfair model. The VSD is an approximation of the WSD that can be computed without access to the distribution of class features. The NP solely captures disparities arising from class-agnostic rules. WSD and VSD measure disparate treatment, and NP measures disparate impact. Details of the metrics are explained in the full paper.

3.3. Verification Approach

Our verification technique extends set-based neural network execution to encompass probability distributions. We propagate a set of potential inputs through the network to determine the possible output range. We use the linear star set representation [15, 16] for state set propagation. Post-analysis, the input set is divided into polytopes, each mapping to outputs with an identical label.

This representation enables a straightforward fairness verification process for neural networks, assuming the direct encoding of C_1 and C_2 into the input features by setting one or more of the features to the appropriate values. The procedure includes:

1. Using neural-network reachability analysis to determine acceptance regions for each class: $\rho(C1)$ and $\rho(C2)$, then calculating $\rho(C_1) \cap \rho(C_2)$.

2. Performing the appropiate integral over the acceptance region depending on the desired metric.

4. Experimental Results

We evaluate various models' fairness, trained on the ADULTS dataset [17], using our proposed metrics (Section 3.2). Our method, distinct from those in Section 2, does not necessitate extensive data sampling or testing data. The ADULTS dataset, comprising both categorical and continuous variables, includes at least one protected feature (e.g., Race or Sex). We consider this a potential source of unfair bias during model training, and use three training strategies (race-permute, augment data, remove data) designed to mitigate this bias. Training strategy details are described in the full paper.

We note the following results. Baseline models exhibit unfairness without fairness training. A significant WSD for all the baseline models indicate that many individuals are labeled differently due to class-specific rules in the model. The VSD serves as a proxy for WSD when input feature probability distributions are unavailable. In 11/16 trials, the model with the lowest WSD also had the lowest VSD. Finally, it is possible to significantly improve fairness while maintaining - or even improving - accuracy.

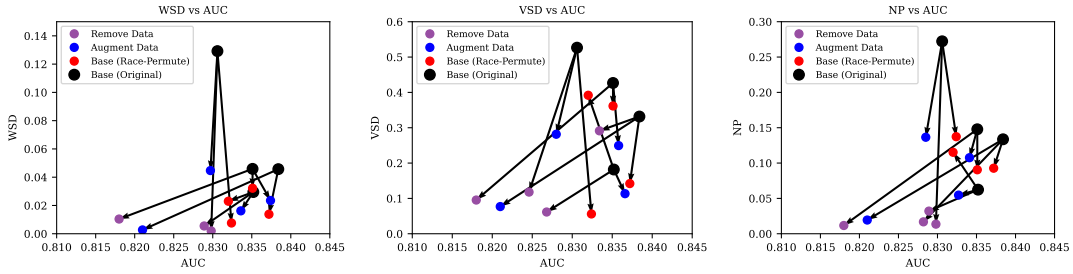

Figure 2: ADULTS Dataset: Model performance/AUC vs Model fairness/NP under three fairness-sensitive training methods. Arrows down and to the left reflect models that are fairer but less accurate than the original mode. We obtain large improvements in fairness at little cost in accuracy.

5. Conclusion

This paper presents a novel technique for formal fairness analysis of neural network models, circumventing the need for access to the model's training data. This broadens the scope for model fairness evaluation. Our evaluation of various models highlights the substantial unfairness in models trained without intervention. We establish that a probability-free approach can effectively substitute when input probability distribution is inaccessible. Due to the formal methods applied, our method provides strong guarantees about the fairness of the model and its potential impact.

References

[1] J. Angwin, J. Larson, S. Mattu, L. Kirchner, Machine bias, in: Ethics of Data and Analytics, Auerbach Publications, 2016, pp. 254–264.

[2] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, Algorithms for verifying deep neural networks, arXiv preprint arXiv:1903.06758 (2019).
[3] H.-D. Tran, W. Xiang, T. T. Johnson, Verification approaches for learning-enabled autonomous cyber-physical systems, IEEE Design & Test (2020).
[4] G. Katz, C. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer, Reluplex: An efficient smt solver for verifying deep neural networks, in: International Conference on Computer Aided Verification, Springer, 2017.
[5] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, M. Vechev, Ai2: Safety and robustness certification of neural networks with abstract interpretation, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 3–18.
[6] S. Bak, C. Liu, T. T. Johnson, The second international verification of neural networks competition (VNN-COMP 2021): Summary and results, CoRR abs/2109.00498 (2021). URL: https://arxiv.org/abs/2109.00498. arXiv:2109.00498.
[7] G. Borca-Tasciuc, X. Guo, S. Bak, S. Skiena, Provable fairness for neural network models using formal verification, 2022. arXiv:2212.08578.
[8] W. Xiang, P. Musau, A. A. Wild, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, T. T. Johnson, Verification for machine learning, autonomy, and neural networks survey, arXiv preprint arXiv:1810.01989 (2018).
[9] A. Albarghouthi, Introduction to Neural Network Verification, verifieddeeplearning.com, 2021. arXiv:2109.10317, http://verifieddeeplearning.com.
[10] S. Bak, H.-D. Tran, K. Hobbs, T. T. Johnson, Improved geometric path enumeration for verifying ReLU neural networks, in: 32nd International Conference on Computer Aided Verification, Springer, 2020.
[11] J. A. Vincent, M. Schwager, Reachable polyhedral marching (rpm): A safety verification algorithm for robotic systems with deep neural network components, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 9029–9035.
[12] C. Urban, M. Christakis, V. Wüstholz, F. Zhang, Perfectly parallel fairness certification of neural networks, Proceedings of the ACM on Programming Languages 4 (2020) 1–30.
[13] S. Galhotra, Y. Brun, A. Meliou, Fairness testing: testing software for discrimination, in: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, 2017, pp. 498–510.
[14] A. Ruoss, M. Balunović, M. Fischer, M. Vechev, Learning certified individually fair representations, arXiv preprint arXiv:2002.10312 (2020).
[15] P. S. Duggirala, M. Viswanathan, Parsimonious, simulation based verification of linear systems, in: International Conference on Computer Aided Verification, Springer, 2016, pp. 477–494.
[16] H.-D. Tran, S. Bak, W. Xiang, T. T. Johnson, Verification of deep convolutional neural networks using imagestars, in: Proceedings of the 32nd International Conference on Computer Aided Verification, Springer, 2020.
[17] D. Dua, C. Graff, UCI machine learning repository, 2017. URL: http://archive.ics.uci.edu/ml.