Activation-Guided Fuzzing for Neural Network Analysis*

Beomjun Kim[1†] and Dongjae Lee[2]

Kangwon National University, Chuncheon, South Korea
bj030@kangwon.ac.kr, dongjae.lee@kangwon.ac.kr

Abstract

Understanding the internal mechanisms of neural networks remains a challenge when model parameters are inaccessible. We propose an activation-guided fuzzing method that systematically generates inputs to selectively activate specific neurons and infer their downstream effects. Experiments on a PyTorch model demonstrate that this method accurately estimates neuron-to-neuron weight directionality using only activation data.

Keywords: Neural Network Analysis, Activation-Guided Fuzzing, Weight Estimation

1 Introduction

Neural networks often function as opaque systems, making internal interpretation difficult. Inspired by software fuzzing, we propose activation-guided fuzzing [3], which automatically mutates inputs to induce selective neuron activation and observe activation propagation. This approach enables weight inference without direct parameter access, bridging software security and neural network analysis [1]. While prior explainability techniques rely on parameter access, activation-guided fuzzing explores internal behavior through input-level mutation, extending interpretability to inaccessible or secured models [2].

Contributions of this paper This paper introduces an activation-guided fuzzing framework for neural network structure extraction, bridging software security testing and neural network analysis. We demonstrate that observing activation propagation patterns enables inference of weight relationships without direct parameter access, and validate our approach on PyTorch-based toy models.

2 Methodology

The proposed fuzzer iteratively mutates inputs to maximize a target neuron's activation, guided by a dominance-strength scoring function that quantifies its output relative to its peers to promote selective activation. Input mutation was performed through Gaussian noise and feature-wise perturbations guided by the dominance-strength score to ensure stable and meaningful activation changes. When a single neuron is selectively activated, subsequent layer activations approximate a linear relationship, allowing weight estimation via simple linear regression. Repeating this across neurons reconstructs their outgoing weight vectors.

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. P-72, December 16-18, 2025, Sapporo, Japan.

†This work was supported by the Korea Internet & Security Agency (KISA) grant funded by the Korean government (Ministry of Science and ICT, MSIT) through the "Information Security Workforce Development (Information Security Specialized University Program)" in 2025.

3 Experiment and Results

A 4-layer PyTorch model (input 10)-(hidden 5-5-5)-(output 2) was tested for 10,000 iterations targeting one hidden neuron. The proposed scoring successfully induced selective activation (Fig 1), and linear regression on the fuzzer's top inputs precisely estimated inter-layer weights (cosine similarity 1.0, Fig 2).

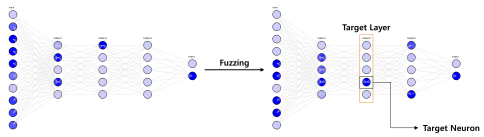

Figure 1: Selective Neuron Activation via Fuzzing

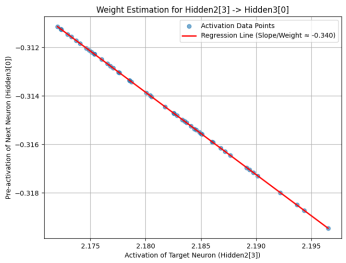

Figure 2: Linear Relationship between Target and Next-Layer Neuron Activations

4 Conclusion and Future Work

This work demonstrates accurate weight estimation via activation-guided fuzzing under full activation observability. Future research will relax the observability assumption in stages: continuous values → binary on/off → partial neuron access, expanding applicability to realistic grey-box security auditing scenarios.

References

[1] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In *International conference on machine learning*, pages 4901–4911. PMLR, 2019.

[2] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *proceedings of the 26th Symposium on Operating Systems Principles*, pages 1–18, 2017.

[3] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT international symposium on software testing and analysis*, pages 146–157, 2019.