Self-Evolving Neural Networks: Networks that Mutate Topologies During Training

Ishaan Singh*[1] and Dr. Sushil Kumar Tiwari[1]

[1]Department of Machine Learning, Indian Institute of Information Techology Lucknow

***Abstract*—**

A. Only a handful of studies focus on modifying self-adaptive evolving structures. Fixed architecture settings dominate the self-adaptive paradigms at the moment. Current researchers seem to have neglected the approaches that are able to automatically evolve during the training phase. In this research, we propose these algorithms which consist of self-evolving neural networks (SENNs) with dynamic topological mutations, like adding, deleting, and rewiring connections, which occur during the learning phase. Such algorithms that perform mutations during learning processes are not well-known in the academic world. Our approach includes the creation of the mutation operators, the implementation of triggering mechanisms through gradient feedback, and the reinforcement of benchmarks comparing SENNs against static models. In a static model, metrics of performance attained are incomparable to those in SENN driven schemes. This experimental thesis achieved strengthened generalization abatement of non-essential parameters and swiftness in convergent tendencies. This research demonstrated the capacity of SENNs for overcoming robustness issues, as well as challenges posed by overfitting. This research proposes sophisticated structures for utilizing dynamic striatal configuration into neural networks.

I. Introduction

The advancement of potent ideas in image recognition, natural language processing, and even autonomous systems of driving have risen alongside the evolvement of Deep Learning[1]. Despite all of the advancements made in these different fields, most of the existing neural networks are created with a fixed topology and a specific neural control, which doesn't showcase maximum performance from data that is diverse, changing, or constantly evolving[2]. On the other hand, biological neural systems are different, as they have the ability to significantly change their structure to fine tune their performance according to any specific stimuli[3]. This Self-Evolving Neural Network (SENNs) research proposes self-developing structures, designs, and frameworks for neural networks with learning based topologies and drive plasticity controlled dynamic mechanisms, eyeing to resolve highly rigid defined architecture frameworks confined systems to become more efficient and flexible[4], [5].

*Name of the company if it is a industry internship

II. Literature Review

A range of methods have been developed to deal with adaptive frameworks in the context of machine learning. The NEAT framework showed the promise of developing network structures using genetic algorithms with neuroevolution[1]. Further developments like CoDeepNEAT and Lamarckian evolution incorporated topology changes along with deep learning models, but these approaches are primarily evolutionary, not gradient-based[1]. On the other hand, NAS, or Neural Architecture Search, customizes a model's architecture, but typically leaves the model unmodifiable after the search is completed[2]. Dynamic sparse training, pruning, growing networks, and other techniques provide adaptable model complexity but concentrate on weight sparsity or network expansion, with no strategy for continuous topology reconfiguration during training[4]. Therefore, the lack of ability neural networks have to alter their topology in real-time while maintaining gradient-based methods indicates a gap in neural network research[3], [4].

III. Research Methods

This document details a study that suggests a hybrid training framework which combines topology evolution and gradient descent optimization.

Creation of Mutation Procedures: Adding nodes in low-activation or high-gradient areas is done to increase the learning capacity. Removing nodes that are too active, therefore, or superfluous is done through salient, gradient map driven, neuron cut detection.

Efficiency and information flow in the network are improved by adding or removing connections to increase access and also aid in streamlining network operations and efficiency.

Mechanism Activation: Triggering mutations during stagnation periods, for example, loss plateaus or very small gradient changes in update the training cycle, are applied at regular intervals.

Control over Change: Hyperparameters equilibrium dominant in exploration and stability govern the magnitude and frequency of topology alterations.

Test Configuration: Robustness assessment uses the noisy synthetic dataset alongside the MNIST and CIFAR-10 datasets.

Primary Comparison Networks: Fixated convolution and feedforward network models.

Evaluation: Generalized gap, convergence pace, accuracy, and number of parameters are reported.

Execution: Dynamic graph changes within backpropagation is incorporated in PyTorch where the models are constructed.

IV. The Results and Discussion

The experimental results emphasize the learning efficiency and flexibility of SENNs:

On MNIST, SENNs attained 99.3

On CIFAR-10, SENNs achieved 82.7

On the synthetic noisy dataset, SENNs exhibited stronger robustness, with lower validation performance variance and greater adaptability to pruning unimportant connections.

Topology mutation activity was found to reduce automatically with training, indicating self-regulation and convergence to optimum structures.

These results show that SENNs not only enhance learning performance but also provide computational efficiency and flexibility benefits. But the stability and control of mutation are still issues, especially in terms of computational overhead and adjusting mutation frequency.

V. Conclusion

This study sought to investigate Self-Evolving Neural Networks that adaptively change their topology throughout training in order to increase flexibility and performance. By combining topology mutation with gradient-based learning, SENNs were demonstrated to gain better accuracy, lower parameter complexity, and greater resilience for various datasets. The research offers a new framework that unites static and evolving architectures and provides theoretical contributions along with practical usage for adaptive deep learning. Future works should aim at improving mutation control mechanisms, incorporating reinforcement learning for architecture evolution, and understanding applications in continual learning and dynamic real-world environments.

Acknowledgment

I would like to express our gratitude to the AI Research Lab at Indian Institute of Information Technology, Lucknow for providing technical assistance. Special thanks to Dr. Sushil Kumar Tiwari for his invaluable guidance and feedback throughout the research process.

References

[1] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[2] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations (ICLR)*, 2019.

[3] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, 2019, pp. 4780–4789.

[4] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature Communications*, vol. 9, no. 1, p. 2383, 2018.

[5] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.

Sign here

(Your Name) (Supervisor Name)