Nature inspired techniques for optimization of Neural Networks: A comprehensive review

Dr. Shweta Sharma[1], Dr. Vivekanandam[2], Dr. Vijayalakshmi Kakulapati[3]
[1] Lincoln University Collage Malaysia; [2] Lincoln University Collage Malaysia; [3] Sreenidhi Institute of Science and Technology, India
shweta_sharma0287@yahoo.com
vivekanandam@lincoln.edu.my
vldms@yahoo.com

Abstract: Optimization of neural networks is a central research topic in artificial intelligence with the goal of enhancing accuracy, efficiency, and scalability. Conventional optimization techniques are typically plagued with local minima, convergence rate, and hyperparameter adjustment. Nature-inspired algorithms, motivated by biological, physical, or evolutionary phenomena, provide robust and general-purpose solutions. This paper presents a systematic survey of prominent nature-inspired optimization techniques, such as genetic algorithms, particle swarm optimization, ant colony optimization, simulated annealing, artificial bee colony, and other novel techniques. We discuss their theoretical foundations, applications to real-world neural network training, strengths, and weaknesses. We also present a comparative assessment to assist researchers in choosing suitable techniques for particular neural network optimization problems.
Keywords: Nature Inspired Algorithm, Neural network, Optimization, network optimization

Introduction

Neural networks (NNs) have transformed many areas of study, including computer vision, natural language processing, and robotics. Though, improving their performance is a challenging task due to of the high-dimensional search spaces complexity, non-convex loss environments, and the sensitivity of the models to their hyperparameters. The most common methods of optimization are used that including gradient descent and its variants. But they are prone to facing the issues of getting stuck in local minima, low convergence rates, and insignificant performance in solving complex problem spaces [1].

Optimization techniques inspired by natural phenomena and grounded on biological, physical, and evolutionary principles offer workable answers to these problems. The algorithms' global search capability, resistance to local minima, and improved convergence characteristics are a result of their inspiration from natural phenomena, such as thermodynamic annealing, swarm intelligence, and natural selection [2]. The main contributions, performance indicators, and suitability for various tasks for neural network optimization are all highlighted in this review's methodical examination of these approaches.

Related work

Given their ability to do global search, avoid local optima, and handle high-dimensional search spaces with ease, nature-inspired optimization techniques have drawn a lot of interest in neural network training and hyperparameter tuning. Numerous algorithms, such as physics-based algorithms, swarm intelligence algorithms, and evolutionary techniques, have been proposed to enhance various elements of neural networks. The significant contributions from the body of current literature on this topic are covered in the next section.

1. *Evolutionary Algorithms for Neural Network Optimization*

Evolutionary algorithms are very good at optimizing neural networks as they work well for exploration and exploitation. The first evolutionary technique is known as Genetic algorithm (GAs) proposed by Goldberg[1]. Genetic algorithm have been used to enhance the architecture and also optimize the weight of hyperparameters in neural networks. The performance of GA proven well in terms of optimization. Also studies demonstrate that use of GA in neural network architecture is achieving well to improve the model performance and generalization. For example, evolutionary-based hyperparameter optimization has been demonstrated the better execution then state of the art grid search approach in deep learning tasks [3].
Storn and price [4] proposed differential evolution (DE) algorithm for exploration and exploitation using the concept of mutation, crossover and selection phases. In contrast with GA, Differential evolution algorithm demonstrated faster convergence with respect to parameter tuning in neural networks. Due to this, DE become a good choice for researchers to explore the qualities of DE in the application of Deep Learning. To achieve this fast convergence, researchers started applying DE in various neural network not only to optimize the parameters, hyperparameters and architectures but for minimizing function evaluations as well [11].

2. *Swarm Intelligence Techniques in Neural Networks*

Optimization of neural networks successfully done by different swarm intelligence algorithm as well by collective events in biological groups. Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart [2] and mimics the social flight of fish and birds to constantly update the possibility of results. Concerning different phases of neural network training, the approach has been applied to weight optimization and hyperparameter tuning. Research has shown that PSO training outcomes in faster convergence and lower processing costs when compared to conventional gradient descent methods [5].

Now a days, numerous types of hybrid models are developed which combine Evolutionary Algorithms with gradient based optimization. Particle swarm optimization (PSO) become hybridized by Kadry et al. [15] with Backpropagation method to enhance the training process in neural network. The comparison shows better results of hybrid approach in term of convergence and precision. Loshchilov and Hutter [12] also aimed for better performance and uses Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm for hyperparameter tuning of deep neural networks.

Based on searching habits of ants, Ant Colony Optimization(ACO) has been used for optimization of Neural networks. Dorigo and Stützle [3] in their studies shows that ACO can tackle the combination optimizational problems, which also included neural architecture. Recent studies have utilized ACO for network pruning and feature selection, which trims down the size of the model without compromising its accuracy [13].

Artificial Bee Colony (ABC) optimization uses to enhance the parameters of Neural network, proposed by Karaboga and Basturk [7]. The dynamic selection of weights for network weights has been done by ABC that shows effective improvement in classification application in deep neural networks. Comparison demonstrates the outperformance of ABC in terms of model performance and training efficiency than typical metaheuristic approaches [14].

Furthermore, swarm based hybridized optimization algorithm can combine with several other nature inspired algorithm in order to achieve improved performance. A real-time adaptive optimization framework suggested by Jing et al. [18] , combine the approach of evolutionary and PSO algorithms for improve accuracy of deep learning algorithms

Yang [6] proposed another nature inspired algorithm named Firefly Algorithm (FA). The algorithm successfully used for feature selection and training neural networks. FA utilizes bioluminescence signaling mechanisms to optimize and has proven especially powerful for multi-modal optimization problems. Comparisons with PSO and GA have shown better performance in preventing premature convergence [10].

3. *Physics-Based and Other Nature-Inspired Techniques*

Another category of algorithms which inspired by nature is physics inspired algorithm. Simulated Annealing (SA) is one of them and was first presented by Kirkpatrick et al. [5]. Inspired on materials science annealing, SA used to optimize deep neural network parameters. Specifically in reinforcement learning, SA was quite successful in escaping local minima and applied for fine tuning the hyperparameters of deep neural networks.SA has the great potential to avoid local optima [8].

The Gravitational Search Algorithm (GSA) introduced by Rashedi et al. [9] represents optimization as a gravitational system in which solutions are drawn to one another by Newtonian gravity. GSA has been used in the optimization of deep neural networks, and it has proven useful in global search and prevention of local minima. Parallel computing methods have improved the use of GSA in big machine learning problems [17].

Applications in Neural Network Optimization

1. *Weight Optimization*

Nature-inspired methods most optimally tune NN weight configurations efficiently, resulting in better generalization and accuracy. It has been proven via research that PSO-based weight optimization enhances classification within deep learning algorithms with the added benefit of decreased computational complexity [11]. Genetic algorithms have also been employed to optimize weight matrices to increase learning rates and stability of training processes [12].

2. *Hyperparameter Tuning*

Evolutionary and swarm intelligence inspired automatic hyperparameter selection methods minimize human intervention to a large degree and improve model performance. Empirical studies demonstrate that hyperparameter optimization using DE provides more precise results with faster convergence rate compared to conventional grid search methods [13]. In addition to these methods, other hybrid methods such as hybridizing PSO and Bayesian optimization have also been investigated for determining optimal learning rates, batch sizes, and dropout rates for obtaining improved generalization abilities [14].

3. *Architecture Search*

To achieve the stability between the computation cost and performance, the best neural network topologies have been acquired using evolutionary techniques like GA and ACO. GA-based architectural optimization has been shown in experiments to reduce model complexity without sacrificing CNN predictive performance [15]. By recognizing effective network topologies with little to no trial and error, ACO has been effectively utilized to optimize NAS [16]

4. *Feature Selection*

Utilizing strategies like PSO and FA, feature selection enhances performance of the network by eliminating duplicated or redundant inputs. FA-based feature selection has been utilized to improve classification accuracy by choosing the most informative input variables and lowering processing costs [17]. PSO-based feature selection methods have also been utilized in medical image processing, and they have demonstrated better illness identification accuracy than traditional feature selection methods [18].

Table 1. Comparative analysis of nature inspired algorithms with applications in Neural networks

Algorithm	**Advantage**	**Disadvantage**	**Convergence Speed**	**Computational Complexity**	**Scalability**	**Applications**	**Sensitivity**
Genetic Algorithms	Robust, flexible	High computational cost	Moderate	High	Moderate	Weight tuning, architecture search	moderate
Particle Swarm Opt.	Simple, few parameters	Premature convergence	Fast	Low	High	Weight optimization, feature selection	high
Ant Colony Opt.	Effective in discrete optimization	Slower for large-scale NNs	Moderate	Medium	Low	Architecture optimization	Moderate
Simulated Annealing	Escapes local minima	Slow convergence	Slow	Medium	Low	Training optimization	Low
Firefly Algorithm	Effective in feature	Parameter sensitivity	Fast	Medium	High	Feature selection,	high

	selection					weight tuning	
Grey Wolf Opt.	Good balance between exploration and exploitation	Sensitive to parameter tuning	Fast	Medium	High	Deep learning tuning, feature selection	moderate
Whale Opt. Alg.	Strong global search ability	High dependency on initial parameters	Moderate	Medium	Moderate	Weight optimization, hyperparameter tuning	high

Table 1 provides the comparative analysis of different nature inspired algorithm of major factors to optimize neural networks. It gives a better understanding of strengths and weaknesses of applying nature inspired optimization algorithm to neural networks. The table includes advantages, disadvantages, convergence speed, computational complexity, scalability, sensitivity and application. NIAs applied on different phases of neural networks to optimize weights, hyperparameters, architecture and feature selection as given in Table 1.

Future Directions

Furthermore, nature inspired algorithm will be useful in parallel computing, automation robotics and for many such sustainable industrial use like additive manufacturing processes, Smart factories etc. Because of the favorable solutions, hybrid models—which associate gradient-based learning algorithms with nature-inspired heuristics—are predicted to be employed more frequently. The successful combination and hybridization in neural networks model architectures is made possible by algorithmic developments such as Differentiable Architecture Search (DARTS). Another important area of advancement is automated architectural search using evolutionary techniques [14].

Developments in parallel and distributed computing paradigms will further improve the scalability of nature-inspired optimization algorithms. Real-time adaptive optimization techniques, as proposed by Jing et al. [18], hold a promising direction towards optimizing deep learning for efficiency. These techniques have the potential to decrease training time by a considerable amount without compromising on accuracy levels, thus making nature-inspired techniques more feasible for practical deep learning applications.

Conclusion

Nature-based optimization algorithms provide great benefits in optimizing neural networks, especially in solving problems of local minima, convergence rate, and hyperparameter optimization. Despite the continued challenges, including the cost of computation and sensitivity to parameters, continued research into hybrid models and adaptive algorithms indicates promising directions for further enhancement.

References

1. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.
2. J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.
3. M. Dorigo and T. Stützle, Ant Colony Optimization. MIT Press, 2004.
4. R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.
5. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," Science, vol. 220, no. 4598, pp. 671–680, 1983.
6. X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," International Journal of Bio-Inspired Computation, vol. 2, no. 2, pp. 78–84, 2010.
7. M. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of Global Optimization, vol. 39, no. 3, pp. 459–471, 2007.
8. E. Aarts and J. Korst, Simulated Annealing and Boltzmann Machines. John Wiley & Sons, 1989.
9. E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," Information Sciences, vol. 179, no. 13, pp. 2232–2248, 2009.
10. X. S. Yang, "Firefly algorithms for multimodal optimization," International Symposium on Stochastic Algorithms, pp. 169–178, 2009.
11. Y. Bengio, "Gradient-based optimization of hyperparameters," Neural Computation, vol. 12, no. 8, pp. 1889–1900, 2000.
12. I. Loshchilov and F. Hutter, "CMA-ES for hyperparameter optimization of deep neural networks," Proceedings of the International Conference on Learning Representations (ICLR), 2016.
13. T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," Journal of Machine Learning Research, vol. 20, no. 55, pp. 1–21, 2019.
14. H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," Proceedings of ICLR, 2019.
15. M. W. Kadry, A. E. Hassanien, and A. T. Azar, "A hybrid PSO with backpropagation algorithm for feedforward neural network training," Journal of Intelligent Systems, vol. 25, no. 1, pp. 63–78, 2016.
16. X. Cui, Z. Sun, and H. Wang, "Adaptive parameter control in evolutionary algorithms: A survey," IEEE Transactions on Evolutionary Computation, vol. 25, no. 2, pp. 152–169, 2021.
17. H. Yu and S. Zhou, "Parallel computing approaches for evolutionary neural network optimization," Concurrency and Computation: Practice and Experience, vol. 32, no. 10, e5123, 2020.
18. L. Jing, Z. Deng, and Y. Tian, "Real-time adaptive optimization strategies for deep learning models," ACM Computing Surveys, vol. 54, no. 5, pp. 1–35, 2022.