Journal of Computer and Knowledge Engineering, Vol. 9, No. 1, 2025. (41-54)

Ferdowsi University of Mashhad

Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

Information and Communication Technology Association of Iran

Structure Optimization in Deep Neural Networks with Synaptic Pruning Based on Connection Appraisal*

Research Article

Aghil Ahmadi[1], Reza Mahboobi Esfanjani[2]

10.22067/cke.2025.88039.1113

Abstract *Deep neural networks typically require predefined architectures, which can lead to overfitting, underfitting, high computational costs, and storage overhead. Dynamic structure optimization through pruning can reduce network redundancy but often results in performance degradation. In this study, we propose a novel pruning method inspired by biological synaptic pruning that adaptively optimizes deep neural network structures. The proposed method continuously monitors the contribution of each connection during training using a dynamic efficiency criterion that evaluates the relative importance of each connection within its layer. Connections are not removed immediately; instead, only those consistently falling below a predefined threshold are pruned, ensuring stability and robustness. Simulation validation is conducted on an industrial distillation column dataset under noisy conditions and the MNIST benchmark dataset. The results demonstrate improved accuracy, enhanced generalization, and faster learning, with an average pruning rate of 53%. Compared to conventional and state-of-the-art pruning techniques, our method achieves superior performance in terms of compression rate and accuracy while effectively mitigating overfitting.*

Key Words Deep Neural Networks, Synaptic Pruning, Distillation Column, Connection Evaluation.

1. INTRODUCTION

In the fields of data science and artificial intelligence, machine learning has experienced tremendous growth. Among its various tools, artificial neural networks (ANNs) have become some of the most reliable and widely used methods, owing to their parallel distributed architecture, learning capability, and generalization potential [1]. These features enable neural networks to effectively handle complex tasks such as automatic control, system identification, and pattern recognition.

A neural network's structure (comprising the number of hidden layers and associated weights) plays a crucial role in determining its overall performance. Both excessively small and overly large networks pose challenges: small networks lack sufficient capacity to model complex relationships, making them difficult to train, while large networks suffer from overfitting, reduced generalization, and increased computational burden [2], [3], [4]. Achieving an optimal network size is thus vital for creating models that are not only accurate but also efficient and interpretable. The recent success of deep neural networks (DNNs) in various machine learning applications has further highlighted this trade-off. Despite their superior performance, DNNs typically demand substantial memory and processing power, making them difficult to deploy in environments with limited computational resources, such as mobile devices and embedded systems [5], [6]. Consequently, methods to reduce the complexity of these networks without sacrificing accuracy have become essential. One widely adopted solution is neural network pruning, which systematically removes unnecessary parameters from a trained network to simplify its structure. Pruning can effectively reduce computational and storage overhead while maintaining acceptable levels of accuracy.

* Manuscript received 2024 May 12, Revised 2025 May 2 Accepted 2025 June 28.

[1] M.Sc. Department of Electrical and Computer Engineering, Sahand University of Technology, Tabriz, Iran.
Email: ag_ahmadi@sut.ac.ir

[2] Corresponding Author. Professor,Department of Electrical and Computer Engineering, Sahand University of Technology, Tabriz, Iran. **Email:** mahboobi@sut.ac.ir

Although pruning has been explored since the late 1980s [7], its relevance has resurfaced with the growing depth and complexity of modern networks. A fundamental challenge in pruning is identifying which connections are suitable candidates for removal. Traditional methods often rely on the magnitude of the connection weights, assuming that smaller weights contribute less to the network's output and thus can be safely pruned. However, both theoretical studies and empirical evidence have shown that this assumption can be misleading [2], [3]. Important connections may exhibit small weight magnitudes due to specific data distributions or network dynamics. Consequently, relying solely on weight magnitude as a pruning criterion risks discarding valuable connections and potentially degrading network performance.

Recent research has emphasized the need for more robust evaluation criteria that go beyond simple weight magnitude. However, many existing methods still assess connection importance in a static, single-phase manner without continuously monitoring their contribution during the training process. Furthermore, most of these approaches focus primarily on optimization and regularization objectives, lacking a biologically plausible foundation [4]. In contrast, the human brain offers a compelling model for effective pruning. During development, the brain undergoes synaptic pruning, a process where redundant or weak synaptic connections are gradually eliminated based on their activity levels [8]. This activity-dependent mechanism strengthens frequently used synapses while removing those that are rarely activated, leading to a more efficient and specialized network [9], [10], [11]. Incorporating such biologically inspired strategies into artificial neural network pruning can potentially enhance both effectiveness and robustness. In this paper, we propose a novel method for optimizing the structure of deep neural networks by integrating brain-inspired synaptic pruning mechanisms with connection evaluation based on network error contribution. Unlike traditional methods that rely on weight magnitude, our approach dynamically monitors the actual influence of each connection on the network's performance. Connections with persistently weak contributions are gradually eliminated, mirroring the brain's "use it or lose it" principle. This strategy not only reduces the risk of removing valuable connections but also improves the network's ability to handle noisy and uncertain data. The remainder of this paper is organized as follows: Section 2 reviews related works in neural network pruning and structure optimization; Section 3 presents the proposed pruning method; Section 4 provides comparative results and discussion; and finally, Section 5 concludes the paper.

2. RELATED WORKS

The primary distinction between shallow and deep neural networks lies in the number of hidden layers. Shallow networks typically consist of a single hidden layer, whereas deep networks comprise multiple hidden layers (at least three), enabling hierarchical feature extraction and improved representation of complex data. This hierarchical structure enhances robustness in managing uncertainties and allows deep networks to model more precise functions, making them superior in applications requiring complex feature learning, such as industrial process modeling and control.

One of the earliest solutions for reducing the computational complexity of neural networks (NNs) is knowledge distillation, in which a smaller model is trained to mimic the behavior of a larger, well-trained model [12]. Despite its effectiveness, this approach requires predefined architectures for student networks, which limits flexibility.

Another extensively studied method is network pruning, where neurons or connections with minimal contribution are systematically removed. Traditional pruning techniques often rely on thresholding weight magnitudes, assuming that smaller weights are less significant [13], [14]. However, this approach has been questioned, as critical connections might occasionally have small weight magnitudes depending on the data and network dynamics [15].

To address these limitations, more advanced pruning criteria have been introduced. For instance, Molchanov et al. [16] proposed utilizing feature map statistics and mutual information to evaluate the relevance of connections. Other researchers have adopted Taylor series expansions for sensitivity analysis, such as the first-order approach by Molchanov et al. [16] and second-order methods by LeCun et al. [17] and Hassibi and Stork [18], using Hessian approximations for more accurate significance estimation.

Beyond individual weights, filter-level pruning methods have also emerged. He et al. [19] proposed a geometric median-based method for removing redundant filters. Yu et al. [20] introduced the Neuron Importance Score Propagation (NISP) technique, propagating importance values backward through the network layers. Li et al. [21] focused on pruning filters with lower weights, and He et al. [22] introduced soft filter pruning, allowing pruned filters to recover through retraining. While these methods improve computational efficiency, they often lack biological plausibility, focusing on mathematical heuristics rather than biologically inspired mechanisms. Furthermore, many existing approaches perform one-time static evaluations without continuously monitoring the dynamic role of connections during training. Regularization methods such as dropout [23] and dropconnect [24] have been effective in preventing overfitting by randomly deactivating neurons or weights during training. However, they do not reduce network complexity at inference time, as all connections are reactivated. Similarly, techniques like meProp [25] sparsify gradients during backpropagation to accelerate

training but do not alter the network's structure.

Another prominent line of research involves evolutionary algorithms, which simultaneously optimize network topology and weights. Evolutionary strategies employ fitness functions that consider accuracy and network complexity [26], [27], [28]. Genetic algorithms, in particular, have been used to prune networks and discover efficient topologies [29], [30], [31], [32]. Despite their adaptability, these methods are computationally intensive and suffer from convergence uncertainties due to the vast search space.

Recently, several pruning methods have been proposed to enhance the efficiency of deep neural networks without significantly compromising performance. In the Lottery Ticket Hypothesis (LTH) method, the concept of "winning tickets" was used to train small subnetworks that can match the performance of the original network if initialized properly [33]. SNIP presents a pre-training pruning strategy based on connection sensitivity to loss, allowing efficient identification of crucial weights before training [34]. GraSP further improves pruning by preserving the gradient flow essential for learning [35]. Movement Pruning is a dynamic pruning method applied during fine-tuning, focusing on the directional movement of weights to identify unimportant connections [36]. Additionally, Global Magnitude Pruning selects the weakest weights across the entire network rather than layer by layer, achieving a better balance between sparsity and accuracy [37]. Despite their success, most of these methods rely heavily on initial weight magnitudes or static criteria, whereas our proposed method continuously monitors the dynamic contribution of each connection to the network's error during training, inspired by biological synaptic pruning mechanisms.

In summary, while significant progress has been made in neural network pruning with the advent of the Lottery Ticket Hypothesis (LTH), SNIP, GraSP, Movement Pruning, and Global Magnitude Pruning, these approaches still primarily rely on static evaluations or single-shot sensitivity analyses. They often assess connection importance based on initial weight magnitudes, gradient sensitivity, or weight movement trends, with limited adaptation during the training process. Moreover, most SOTA methods lack a biologically inspired mechanism to guide pruning decisions dynamically. These gaps highlight the necessity for pruning strategies that can adapt to the evolving structure and error dynamics of the network. Our proposed method addresses these limitations by continuously monitoring the real-time contribution of each connection to the overall network error and gradually pruning redundant connections, inspired by the synaptic pruning process observed in biological neural systems. This dynamic and brain-inspired approach ensures more robust pruning decisions and greater resilience to noisy and uncertain data, pushing beyond the capabilities of existing SOTA techniques.

3. PROPOSED PRUNING METHOD

We present an innovative Brain-Inspired Connection Evaluation Pruning technique in this section. In the first stage of the proposed algorithm, the real value of each connection in the network is determined, which essentially reflects the importance and contribution of that connection to the overall network performance. In this context, the "real value" is assessed based on an error-driven criterion, where the impact of omitting each connection on the network's output error is evaluated. This allows for a more accurate measurement of each connection's significance beyond simple weight magnitudes.

This evaluation is based on neglecting each connection and computing the error that results from its removal. To measure the true value of the neurons, the current output of the network must be brought closer to the ideal values. In other words, connections that lead to a deviation of the output from the ideal values increase errors. We arrange the connections according to the value of training errors produced when they are eliminated. In the process of pruning, our goal is to make the network lighter and smaller, but we must note that the accuracy of the network should not decrease too much. Therefore, pruning candidates include a subset of connections that have produced the minimum value of errors. We will delete connections inspired by the pruning process in the human brain as follows: brain pruning involves making stronger connections with a higher frequency of use and weaker connections with a lower frequency of use [38]. A connection will be deleted if, over the course of several steps, its strength falls below a predetermined threshold. Namely, if the weak score of a connection persists, it will be eliminated. Fig. 1 depicts the process of synaptic pruning. It is evident that we need to specify two crucial parameters. The first parameter is the threshold, which indicates which connections may need pruning. The second parameter is the warning time, which indicates how long the related connection will remain active before being deleted. However, here the criterion is training error instead of connection weights.

Use it or lose it: neuroscientists refer to the decrease in spine density as "synaptic pruning." Through this process, weaker structures are eliminated, reallocating resources to the surviving ones so they can become stronger and more stable. As it became abundantly evident that synaptic activity directs appropriate pruning, scientists focused on identifying the cellular processes that might control the remodeling [38].

We determine a threshold value for the acceptable error in order to guide the pruning process. This threshold serves as a benchmark to evaluate the significance of each connection within the network. During each iteration, we closely monitor the connections whose removal results in

minimal increases in error compared to the previous step. These connections, having demonstrated a consistently low impact on overall network performance, are considered potential candidates for removal. To ensure a cautious and reliable pruning process, we introduce a control mechanism known as the "warning number." This parameter defines the required consistency of a connection's low contribution across multiple evaluations. Specifically, connections that remain below the error threshold for a certain number of consecutive iterations (defined by the warning number) are identified as weak contributors and selected for pruning. This progressive evaluation prevents the premature removal of connections that might exhibit temporary fluctuations in importance due to network dynamics. This method allows for a gradual and robust reduction in network complexity, as only the connections with persistently negligible impact are pruned. By continuously reassessing the error contribution of each connection, the proposed approach mimics biological pruning mechanisms, ensuring that only truly redundant connections are eliminated. The procedure of the proposed pruning technique is illustrated in Fig. 2, which visually represents the step-by-step process, including error evaluation, candidate selection, application of the warning number criterion, and final pruning decisions.

The pruning pseudo-code is presented in detail in Table 1. This combined pruning method is presented to address the disadvantages of existing pruning methods as mentioned in the previous sections: relying only on the weighted domain is not sufficient, and there is a high probability that some very important network connections are omitted. We addressed this weakness by sorting the connections, and after finding the connections susceptible to deletion, the removal is not done in one step by decreasing the value once. We successively caution the pruning candidates and prune them based on these warnings.

In summary, the evaluation of all network connections is carried out based on their contribution to the overall network error. Specifically, we determine the error introduced by individually removing each connection and then rank the connections according to the magnitude of these errors. Connections associated with the least error increases are considered for removal, guided by a pruning rate defined by the designer. Consequently, our pruning strategy incorporates two key elements: evaluating connections based on training error and tracking their iterative weak scores. Ultimately, this process yields a pruned network that significantly outperforms the original configuration. The motivation behind the proposed pruning strategy stems from the limitations observed in existing methods. Most conventional pruning techniques rely heavily on static evaluations, primarily based on weight magnitude or sensitivity analyses performed either before or after training. Such static approaches often fail to capture the dynamic behavior and real-time importance of connections throughout the learning process, leading to the risk of pruning significant but low-magnitude connections and potentially degrading network performance.

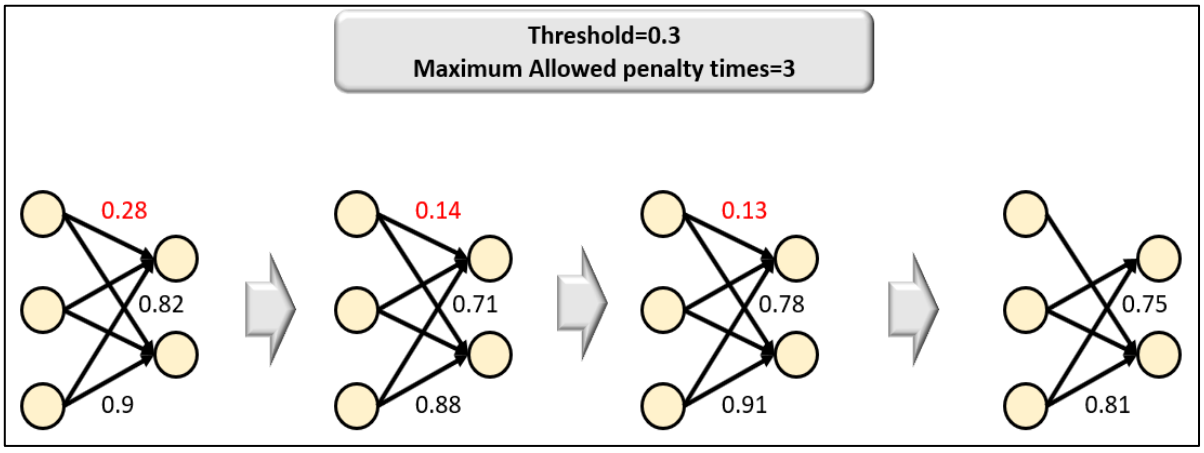

Fig. 1. Process of synaptic pruning

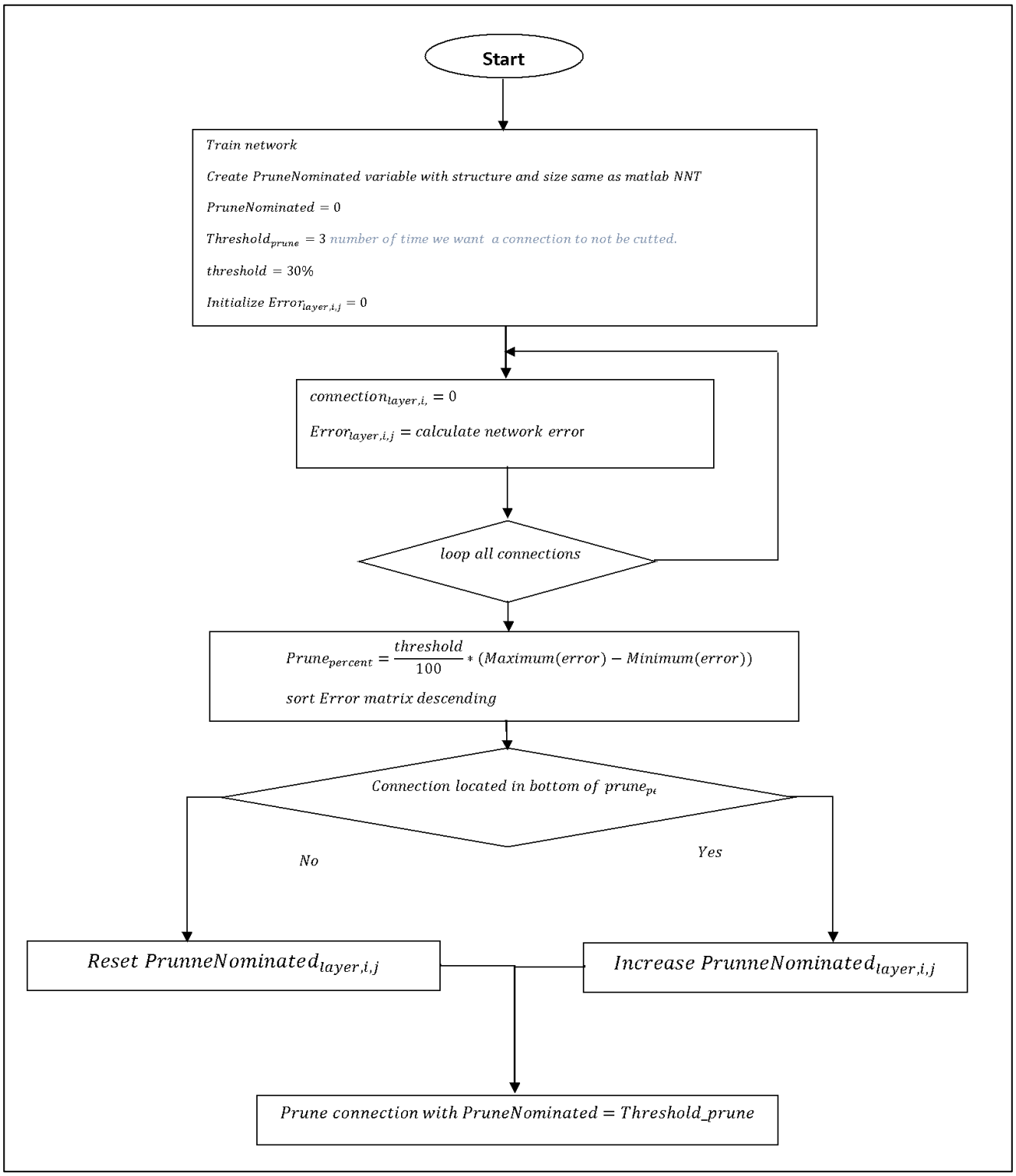

Fig. 2. Flowchart of the proposed pruning

TABLE1
Pruning pseudo code

Train network

Create PruneNominated variable with structure and size same as matlab NNT

$\text{PruneNominated} = 0$

$\text{Threshold}_{\text{prune}} = 3$ number of time we want a connection to not be cutted.

$\text{threshold} = 30\%$

Initialize $\text{Error}_{\text{layer,i,j}} = 0$

loop steps each second

loop connections

$\text{connection}_{\text{layer,i,}} = 0$

$\text{Error}_{\text{layer,i,j}} = \text{calculate network error}$

end loop

$\text{Prune}_{\text{percent}} = \frac{\text{threshold}}{100} * (\text{Maximum}(\text{error}) - \text{Minimum}(\text{error}))$

sort Error matrix descending

for error counter $= 1$ to $\text{count}_{\text{connections}} - \text{Prune}_{\text{percent}} * \text{count}_{\text{connections}}$

$\text{PruneNominated}_{\text{connectiont}_{\text{error counter}}} = 0$

end for

for error counter $= \text{count}_{\text{connections}} - \text{Prune}_{\text{percent}} * \text{count}_{\text{connections}} + 1$ to end

$\text{PruneNominated}_{\text{connectiont}_{\text{error counter}}} + +$

end for

loop connections

if $\text{PruneNominated}_{\text{connection}} = \text{Threshold}_{\text{prune}}$

Prune this connection

end if

end loop

end loop

In contrast, the human brain undergoes synaptic pruning based on continuous monitoring of synaptic activity, gradually eliminating weak and unused connections while reinforcing the strong ones. Inspired by this biological process, our approach integrates a dynamic evaluation criterion that monitors the real-time contribution of each connection to the overall training error. By focusing on the impact of each connection on network performance rather than solely on its weight magnitude, we ensure that only truly redundant connections are pruned.

Moreover, the introduction of a "warning number"—requiring multiple consecutive evaluations before pruning— prevents the premature removal of connections due to temporary fluctuations, thus enhancing the robustness of the pruning process. This feature becomes particularly crucial in noisy or uncertain environments, such as industrial process modeling, where data variability can affect the stability of traditional pruning methods.

Therefore, the proposed method not only addresses the shortcomings of static and heuristic-driven pruning approaches but also offers a biologically plausible, adaptive, and noise-resilient solution for optimizing deep neural network architectures. These attributes make it a highly appropriate choice for complex, real-world applications.

4. COMPARISON RESULTS AND DISCUSSION

In this section, we apply the suggested method to a neural network model of a refinery process's distillation tower in order to assess its efficacy. The objective is to investigate how, in the case of ideal and noisy data, the proposed algorithm can enhance identification accuracy and convergence speed.

The distillation tower, which is a multi-input, multi-output (MIMO) nonlinear system, is a general and inseparable part of a refinery. A distillation column is a device for separating components of a solution. In fact, in the distillation tower, the components of a solution are separated based on their volatility and boiling point differences. Industrial distillation towers are widely used in various process industries, but one of their main uses is crude oil refinement. In the oil industry, different hydrocarbons are separated based on their volatility by the distillation method. The ethane-ethylene distillation column is one of the most widely used towers. Due to its significance, high-purity ethylene production is required. Our data belongs to an ethane-ethylene distillation column identification experiment. There are four series in the data [39]:

U_dest, Y_dest: without noise (ideal series)
U_dest_n10, Y_dest_n10: 10 percent additive white noise
U_dest_n20, Y_dest_n20: 20 percent additive white noise
U_dest_n30, Y_dest_n30: 30 percent additive white noise

There are 90 samples for neural network training. The following describes the inputs and outputs:

Inputs:

1) The proportion between feed flow and reboiler duty
2) The relationship between feed flow and reflux rate
3) Proportion between the feed flow and the distillate
4) Composition of input ethane
5) Top pressure

Outputs:

1) Top ethane composition
2) Bottom ethylene composition
3) Top-bottom differential pressure.

Therefore, we use a deep network with 5 inputs and 3 outputs and also 90 connections (Fig. 3). We can leverage the capabilities of the deep network, provided that we first have correct weight training and, secondly, to increase the speed of the network and prevent overfitting, we find the best possible structure for the network through our structural optimization scheme.

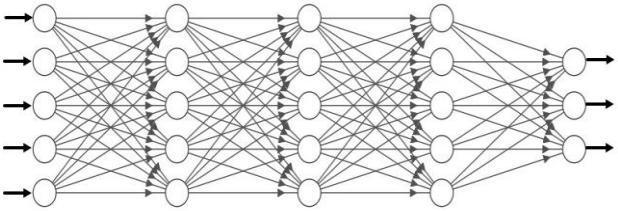

Fig. 3. Applied Deep Neural Network

First, we train the network with the data we have. Fig. 4 shows how the network performance changes (performance function value) each time the network is trained. It includes three curves with different colors for training, validation, and test data. The value of the performance function on the data in each category is displayed in each plot. The horizontal axis label indicates the number of times (epochs) the network has been trained. Also, the title of this graph shows that the best performance of the network (on training and validation data) was achieved in the second epoch, along with the value of the performance function at this point. This optimal point is also marked by two crossed dotted lines whose intersection is at the optimal point, and a green circle is drawn around this point. Furthermore, the regression charts for the training, validation, and test data are given in Fig. 5.

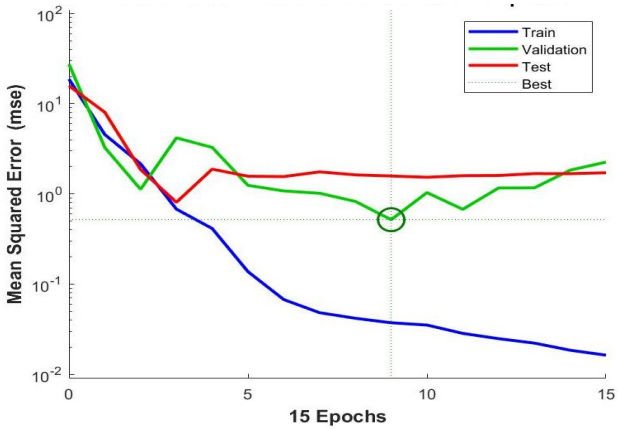

Fig. 4. Performance of the deep neural network

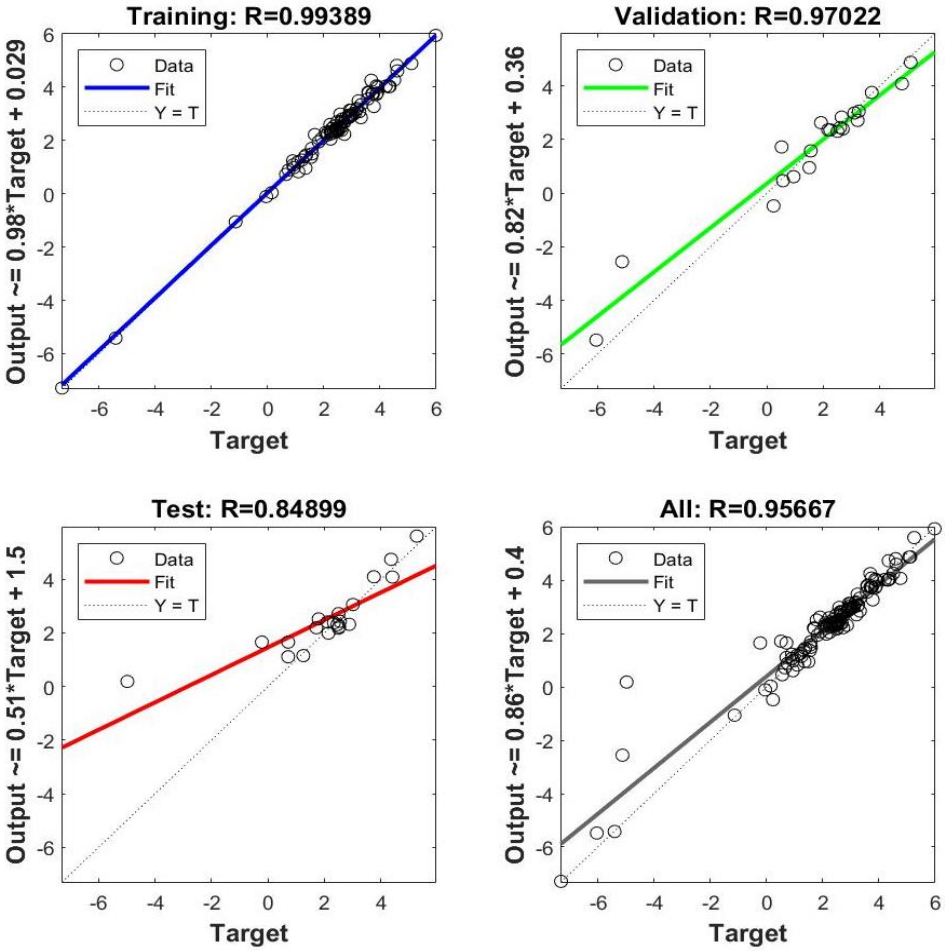

Fig. 5. Regression for the training, validation and test data

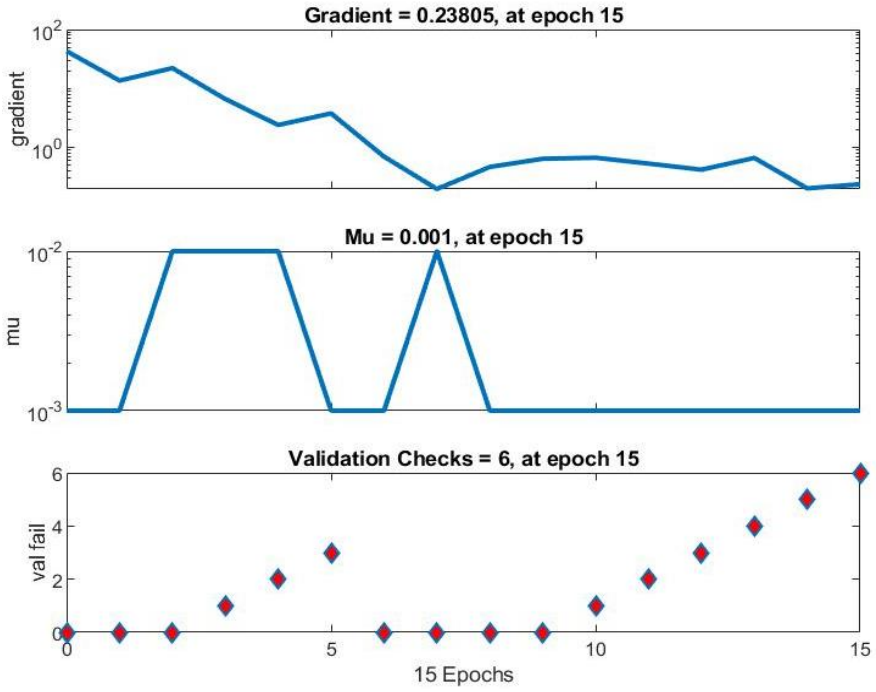

Fig. 6. Training state

In the training state visualization shown in Fig. 6, more information from the training is displayed; for example, the "val fail" graph shows in which epoch the evaluation of the validation data was rejected. This graph shows the cumulative number of failed evaluations. Training stops whenever the network fails six consecutive evaluations.

Comparing different pruning techniques to assess how far the field has come in recent years is a challenging task. Nonetheless, two significant metrics are typically employed and presented here. The compression ratio is defined as the new size divided by the original size. The theoretical speedup is defined as the ratio of the initial number of multiply-adds to the new number. The performance function is the function on which the performance of the network is measured. In this problem, our performance function is MSE (mean square error). In Table 2, comparative data are shown for different scenarios. We study networks with different topologies (shallow and deep) and also compare our approach to the dropout method [24], which is a powerful technique to prevent overfitting under similar circumstances.

As seen, we accelerated network performance and training by utilizing an inventive pruning technique. It is simple to expand the suggested pruning method to other intelligent process industries. Noisy data, which is commonly encountered in real-world industrial settings, is one of the most significant issues in measurement and control. This work aims to investigate whether the proposed algorithm can enhance the speed of convergence and identification accuracy even in cases where a large number of connections are ignored and, more crucially, the data is noisy.

The results of the deep network pruned using the proposed approach, presented in Fig. 7, are compared with those of the shallow network when dealing with data that is noise-free, with 10%, 20%, and 30% noise. It is evident that the proposed structure performs noticeably better, particularly when handling noisy data.

Concisely, a deep network pruned with the proposed method is used to model the distillation tower, and its efficiency was demonstrated compared to the shallow network. Additionally, we compared (Table 3) the RMSE criterion between the proposed model and three other structures in order to compare it with other neural network-based models. The mentioned structures are: nonlinear auto-regressive with exogenous inputs (NARX)-based ANFIS and NARX structure-based neural networks (using both the Levenberg–Marquardt and the Steepest Descent algorithms) [40]. The comparison of errors amply demonstrates the superiority of the proposed method over alternative structures.

TABLE 2
Comparative results

NN type	**Shallow NN (1 hidden layer)**			**NN (2 hidden layers)**			**Deep NN (3 hidden layers)**		
Parameters	Initial	ropout	PROPOSED	Initial	Dropout	PROPOSED	Initial	Dropout	PROPOSED
Accuracy (%)	76.62	77.10	77.94	81.35	82.70	84.73	82.63	83.87	85.89
Net. Compression %)	-	47	47.26	-	53	53.16	-	58	58.71
Execution Time (ms)	15	17.4	13.5	16	18.7	14.45	17.5	19.2	15

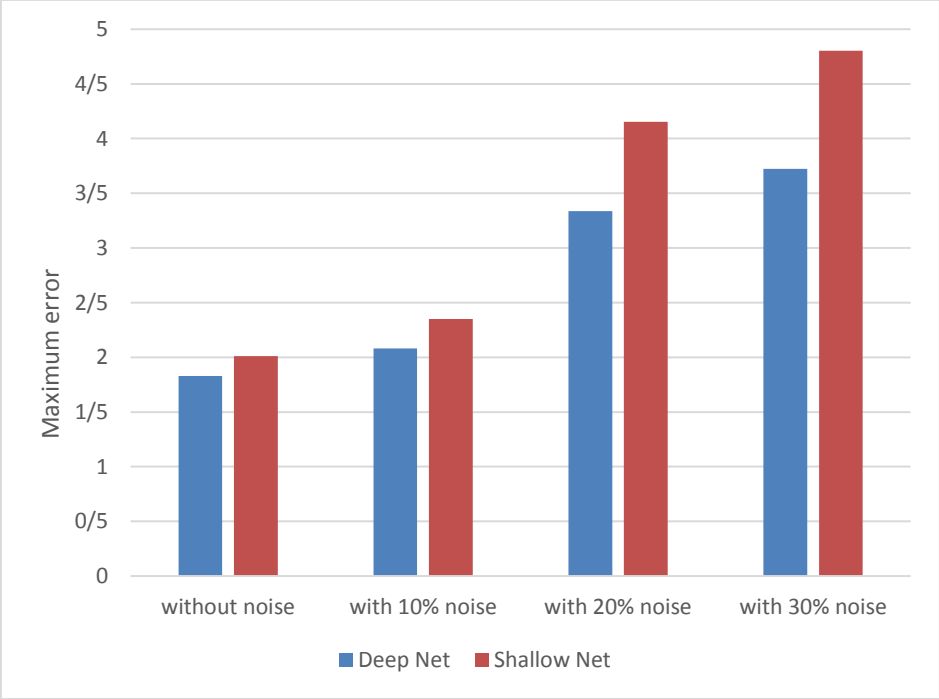

Fig. 7. Deep and Shallow networks comparison in noisy data management

TABLE 3
RMSE for neural networks models, ANFIS and the proposed

Outputs	**Steepest Descent**	**Levenberg Marquardt**	**ANFIS**	**PROPOSED**
Top Composition	0.639	0.2090	0.0421	0.0233
Bottom Composition	1.3127	0.4913	0.031	0.024
Pressure Difference	1.0053	0.2480	0.0189	0.0117

4.1. Generalization Capability

Although this study focused on the distillation column dataset, the underlying principles of the proposed pruning method are generalizable to other complex, nonlinear systems. The dynamic evaluation of connection contributions ensures that the method adapts to diverse data patterns, making it applicable to various domains where overfitting and redundancy are significant concerns. The progressive, biologically inspired pruning strategy further enhances the model's ability to handle unseen data, supporting its potential use in broader industrial and scientific applications.

4.2. Run Time Complexity Analysis

From a computational perspective, the proposed method introduces additional overhead during training due to continuous connection evaluation. However, this overhead is strategically balanced by the significant reduction in network size, which directly impacts inference speed and computational resource requirements. The results in Table 2. highlight that despite the added complexity in the training phase, the overall execution time is reduced post-pruning. This trade-off is particularly beneficial in real-time applications where inference speed is critical. Additionally, the pruning process does not require retraining from scratch, which further mitigates computational costs. By focusing on preserving high-contribution connections, the method ensures efficiency without compromising accuracy, positioning it as a practical solution for resource-constrained environments.

4.3. Comparative Analysis of Pruning Methods

To provide a broader and more comprehensive perspective, we compared our proposed pruning method with several state-of-the-art (SOTA) approaches in the field. These include the Lottery Ticket Hypothesis (LTH), SNIP, GraSP, Movement Pruning, and Global Magnitude Pruning.

The comparison focuses on key characteristics such as the use of dynamic monitoring, biological inspiration, timing of pruning during the learning process, and robustness to noisy data.

As seen in Table 4, most of the SOTA methods focus on static or pre-training evaluations and are not inspired by biological processes. Furthermore, they generally lack robustness when dealing with noisy data, which is common in real-world industrial applications. In contrast, our proposed method incorporates dynamic monitoring of connection contributions throughout training, guided by brain-inspired synaptic pruning principles. This dynamic evaluation not only enables more precise pruning decisions but also enhances the model's ability to handle noisy datasets, as demonstrated by the experimental results. We have included a quantitative comparison between the proposed method and a conventional pruning method (Global Magnitude Pruning) and a recent state-of-the-art method, SNIP. Comparisons are made on both the industrial distillation column dataset and the MNIST benchmark dataset. The results clearly indicate that the proposed method consistently achieves higher accuracy and compression rates across both datasets. This highlights the method's potential for broader application in domains where data quality and model efficiency are critical.

The results in Table 5 indicate that our proposed method consistently outperforms both traditional and recent state-of-the-art pruning techniques in terms of accuracy and compression rate across different datasets.

TABLE 4
Comparison of pruning methods based on key characteristics

Method	**Dynamic Monitoring**	**Biologically Inspired**	**Pretraining/Post-training**	**NoisyData Robustness**
Lottery Ticket Hypothesis (LTH)	×	×	Post-training	×
SNIP	×	×	Pre-training	×
GraSP	×	×	Pre-training	×
Movement Pruning	✓	×	During fine-tuning	×
Global Magnitude Pruning	×	×	Post-training	×
Proposed Method	✓	✓	During training	✓

TABLE 5
Quantitative evaluation of the proposed method versus recent approaches

Dataset	Method	Accuracy (%)	Compression Rate (%)
Distillation Column	Global Magnitude	83.2	50%
Distillation Column	Proposed Method	85.9	58.7%
MNIST	SNIP	98.2	40%
MNIST	Proposed Method	98.5	52%

4.4. Generalization and Overfitting Control

In addition to improving model efficiency, pruning methods play a critical role in enhancing generalization by reducing network complexity. The proposed brain-inspired dynamic pruning approach continuously monitors and removes redundant connections during training, leading to a more compact network structure with fewer parameters. This reduction in the model's capacity limits its ability to overfit the training data and facilitates better generalization to unseen samples. The results reported in Table 5 further support this claim, showing minimal gaps between training and testing performance across different datasets, including the industrial distillation column and the MNIST benchmark. Such consistency in performance demonstrates that the proposed pruning strategy effectively mitigates overfitting and improves the network's generalizability, even under noisy and complex conditions.

4.5. Limitations and Future Work

While the current study provides comprehensive validation on the distillation column dataset, future work will focus on applying the proposed method to other datasets to further validate its generalizability. Nevertheless, the algorithm's foundation, rooted in connection contribution evaluation and brain-inspired pruning, is inherently adaptable to a wide range of neural network architectures and application domains.

5. CONCLUSION

This study introduced a dynamic pruning method inspired by synaptic pruning in the human brain to optimize deep neural network architectures. By continuously monitoring the real-time contribution of connections during training, the method preserves important neurons and gradually eliminates redundant ones. Simulation results demonstrated improved or preserved accuracy, significant network compression, and faster training times. Additionally, the method showed robustness against noisy data, highlighting its practical applicability. A key advantage of our method is its ability to enhance generalization by reducing network complexity, thereby mitigating overfitting. The minimal gap between training and testing performance across different datasets confirms this capability. Furthermore, comparative analysis indicated that our approach outperforms both conventional pruning techniques and some recent state-of-the-art methods, in terms of accuracy and compression rates. Overall, the findings demonstrate that the proposed pruning strategy is efficient for optimizing neural networks. Future work will explore its extension to more complex architectures and broader application domains, along with further validation on additional benchmark datasets.

6. REFERENCES

[1] Z. Allen-Zhu, Y. Li, and Y. Liang. (2019, Dec.). Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers. *Advances in Neural Information Processing Systems*. [Online]. *32*, pp. 1–13.

[2] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. (2017, Jul.). Adanet: Adaptive Structural Learning of Artificial Neural Networks. Presented at International conference on machine learning (ICML). [Online].

[3] Y. Chauvin. (1990, Feb.). Generalization Performance of Overtrained Back-Propagation Networks. Presented at European Association for Signal Processing Workshop. [Online]. pp. 45-55. Available: https://doi.org/10.1007/3-540-52255-7_26

[4] G. G. Towell, M. W. Craven, and J. W. Shavlik. (1991, Jun.). Constructive Induction in Knowledge-Based Neural Networks. Presented at Proceedings of the Eighth International Conference (ML91), pp. 213-217. [Online]. Available: https://doi.org/10.1016/B978-1-55860-200-7.50046-5

[5] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, and Y. Wu. (2019, Dec.). GPipe: Efficient Training of Giant Neural Networks Using Pipeline Parallelism. *Advances in Neural Information Processing Systems*. [Online]. *32*, pp. 1–13.

[6] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer. (2017, Dec.). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*. [Online]. *105(12)*, pp. 2295–2329. Available: https://doi.org/10.1109/JPROC.2017.2761740

[7] S. A. Janowsky. (1989, Jun.). Pruning Versus Clipping in Neural Networks. *Physical Review A*. [Online]. *39(12)*, p. 6600.

[8] G. Chechik, I. Meilijson, and E. Ruppin. (1999, Jun.). Neuronal Regulation: A Biologically Plausible Mechanism for Efficient Synaptic Pruning in Development. *Neurocomputing*. [Online]. *26–27*, pp. 633–639. Available: https://doi.org/10.1016/S0925-2312(98)00161-1

[9] A. Pascual-Leone, A. Amedi, F. Fregni, and L. B. Merabet. (2005, Jul.). The Plastic Human Brain Cortex. *Annual Review of Neuroscience*. [Online]. *28(1)*, pp. 377–401. Available: https://doi.org/10.1146/annurev.neuro.27.070203.144216

[10] C. A. Mangina and E. N. Sokolov. (2006, Apr.). Neuronal Plasticity in Memory and Learning Abilities: Theoretical Position and Selective Review. *Psychophysiology*. [Online]. *60(3)*, pp. 203–214. Available: https://doi.org/10.1016/j.ijpsycho.2005.11.004

[11] M. V. Johnston, A. Ishida, W. N. Ishida, H. B. Matsushita, A. Nishimura, and M. Tsuji. (2009, Jan.). Plasticity and Injury in the Developing Brain. *Brain and Development*. [Online]. *31(1)*, pp. 1–10. Available: https://doi.org/10.1016/j.braindev.2008.03.014

[12] G. Hinton, O. Vinyals, and J. Dean. (2015, Mar.). Distilling the Knowledge in a Neural Network. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1503.02531

[13] G. Chechik, I. Meilijson, and E. Ruppin. (1998, Oct.). Synaptic Pruning in Development: A Computational Account. *Neural Computation*. [Online]. *10(7)*, pp. 1759–1777. Available: https://doi.org/10.1162/089976698300017124

[14] S. Han, J. Pool, J. Tran, and W. Dally. (2015, Dec.). Learning Both Weights and Connections for Efficient Neural Network. *Advances in Neural Information Processing Systems*. [Online]. *28*, pp. 1–13.

[15] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz. (2019, Jun.). Importance Estimation for Neural Network Pruning. *Presented at Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11264–11272. [Online].

[16] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. (2016, Nov.). Pruning Convolutional Neural Networks for Resource Efficient Inference. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1611.06440

[17] Y. LeCun, J. Denker, and S. Solla. (1989, Dec.). Optimal Brain Damage. *Advances in Neural Information Processing Systems*. [Online]. *2*, pp. 598–605.

[18] B. Hassibi and D. Stork. (1992, Dec.). Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. *Advances in Neural Information Processing Systems*. [Online]. 5, pp. 164–171.

[19] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang. (2019, Jun.). Filter Pruning via Geometric Median for Deep Convolutional Neural Network Acceleration. Presented at Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 4340–4349. [Online].

[20] R. Yu, A. Li, C. F. Chen, J. H. Lai, V. I. Morariu, X. Han, M. Gao, C. Y. Lin, and L. S. Davis. (2018, Jun.). NISP: Pruning Networks Using Neuron Importance Score Propagation. Presented at Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 9194–9203. [Online].

[21] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. (2016, Aug.). Pruning Filters for Efficient ConvNets. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1608.08710

[22] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang. (2018, Aug.). Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1808.06866

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. (2014, Jun.). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. [Online]. *15*, pp. 1929–1958.

[24] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. (2013, Jun.). Regularization of Neural Networks Using DropConnect. Presented at International Conference on Machine Learning (ICML), pp. 1058–1066. [Online].

[25] X. Sun, X. Ren, S. Ma, and H. Wang. (2017, Jun.). MeProp: Sparsified Back Propagation for Accelerated Deep Learning with Reduced Overfitting. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1706.06197

[26] P. J. Angeline, G. M. Saunders, and J. B. Pollack. (1994, Jan.). An Evolutionary Algorithm That Constructs Recurrent Neural Networks. *IEEE Transactions on Neural Networks*. [Online]. *5(1)*, pp. 54–65. https://doi.org/10.1109/72.265960

[27] X. Yao and Y. Liu. (1996, Mar.). Evolving Artificial Neural Networks Through Evolutionary Programming. Presented at Proceedings of the 5th Annual Conference on Evolutionary Programming, pp. 257–266. [Online].

[28] J. C. Park and S. T. Abusalah. (1997). Maximum Entropy: A Special Case of Minimum Cross-Entropy Applied to Nonlinear Estimation by an Artificial Neural Network. *Complex Systems*. [Online]. *11*, pp. 289–307.

[29] E. Vonk, L. C. Jain, and R. Johnson. (1995, Dec.). Using Genetic Algorithms with Grammar Encoding to Generate Neural Networks. *Presented at Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1928–1931. [Online]. Available: https://doi.org/10.1109/ICNN.1995.488965

[30] I. Ioan, C. Rotar, and A. Incze. (2004, Sep.). The Optimization of Feed-Forward Neural Network Structure Using Genetic Algorithms. Presented at Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics (ICTAMI), Thessaloniki, pp. 223–234. [Online].

[31] F. Zhao, T. Zhang, Y. Zeng, and B. Xu. (2017, Oct.). Towards a Brain-Inspired Developmental Neural Network by Adaptive Synaptic Pruning. Presented at Proceedings of the International Conference on Neural Information Processing, pp. 182–191. [Online]. Available: https://doi.org/10.1007/978-3-319-70093-9_19

[32] A. Ahmadi and B. Mashoufi. (2012, May). New Optimized Approach for Artificial Neural Networks Training Using Genetic Algorithms and Parallel Processing. *International Review on Computers and Software*. [Online]. *7(5)*, pp. 2232–2238.

[33] J. Frankle and M. Carbin. (2019, Mar.). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1803.03635

[34] N. Lee, T. Ajanthan, and P. H. Torr. (2019, Feb.). SNIP: Single-Shot Network Pruning Based on Connection Sensitivity. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.1810.02340

[35] C. Wang, G. Zhang, and R. Grosse. (2020, Apr.). Picking Winning Tickets Before Training by Preserving Gradient Flow. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.2002.07376

[36] V. Sanh, T. Wolf, and S. Ruder. (2020, May). Movement Pruning: Adaptive Sparsity by Fine-Tuning. *arXiv preprint*. [Online]. Available: https://doi.org/10.48550/arXiv.2005.07683

[37] S. Han, J. Pool, J. Tran, and W. J. Dally. (2015, Dec.). Learning Both Weights and Connections for Efficient Neural Networks. *Advances in Neural Information Processing Systems*. [Online]. *28*, pp. 1135–1143. Available: https://doi.org/10.48550/arXiv.1506.02626

[38] R. Morini, M. Bizzotto, F. Perrucci, F. Filipello, and M. Matteoli. (2021, Feb.). Strategies and Tools for Studying Microglial-Mediated Synapse Elimination and Refinement. *Frontiers in Immunology*. [Online]. *12*, no. 640937. Available: https://doi.org/10.3389/fimmu.2021.640937

[39] R. P. Guidorzi, M. P. Losito, and T. Muratori. (1982, Oct.). The Range Error Test in the Structural Identification of Linear Multivariable Systems. *IEEE Transactions on Automatic Control*. [Online]. *27(5)*, pp. 1044–1054. Available: https://doi.org/10.1109/TAC.1982.1103068

[40] E. Abdul Jaleel and K. Aparna. (2015). Identification of Ethane-Ethylene Distillation Column Using Neural Network and ANFIS. Presented at Proceedings of the 5th International Conference on Advances in Computing and Communications (ICACC). [Online].