Attention Based Evolutionary Approach for Image Classification

Ajay Prem, Anirudh Joshi, Haritha Madana, Jaywanth Gowda and Arti Arya

January 15, 2023

Attention based Evolutionary Approach for Image Classification

Ajay Prem
Dept of CSE.
PES University
Bengaluru, India
ajay01.prem@gmail.com

Anirudh Joshi
Dept of CSE.
PES University
Bengaluru, India
joanirudh@outlook.com

Haritha Madana
Dept of CSE.
PES University
Bengaluru, India
harithamadana@gmail.com

Jaywanth J
Dept of CSE.
PES University
Bengaluru, India
jaywanth.gowda11@gmail.com

Arti Arya
Dept of CSE.
PES University.
Bengaluru, India
https://orcid.org/0000-0002-4470-031

***Abstract*—Lately, evolutionary algorithms have gained traction due to their ability to produce state-of-the-art deep learning architectures for a given data set, even though they require considerable amount of compute resources, they are a heavily researched domain because of the complexities involved in designing deep learning architectures. Currently, none of the evolutionary approaches available have incorporated the attention mechanism, which is a proven technique to improve the performance of image classification and language models. This paper posits a neuroevolutionary technique coupled with the use of Convolution Block Attention Module for image classification. As technology progresses, it's inevitable that there will be massive advancements leading to cheaper and more available computing making evolutionary approaches a promising avenue to develop task specific deep learning models. The proposed approach evolves a topology that achieves a high fitness of 87.44% on the CIFAR-10 benchmark, using fewer parameters as compared to previous approaches. This results in a superior fitness score compared to most past approaches, despite being evolved for just few generations.**

***Index Terms*—Neuro-Evolution, Genetic Algorithms, topology evolution, Attention, CoDeepNEAT, Convolutional Block Attention Module(CBAM), CIFAR-10**

I. Introduction

The widespread adoption of deep learning in recent years can be attributed to the availability of large amounts of data and the development of hardware technology, including high-performance GPUs, that enables the training of large and complex deep learning models quickly. Deep learning has had a significant impact on everyday life, with early applications including web search and online advertising. As the technology has advanced, deep learning models are being used for more complex tasks with a noticeable impact on everyday life, including facial recognition in mobile phones and drug discovery. As problems have become more complex, the deep learning models used to solve them have also become more complex. There is a widespread belief in the data science community that the most accurate models for a given problem are increasingly complex and difficult to interpret. Neural network architectures, also known as "black boxes," are difficult to interpret and design for specific tasks because it is unclear how the input variables are combined to produce the final predictions. This makes designing task-specific deep learning architectures challenging.

Due to the increase in complexity of designing large architectures by hand, there has been a considerable amount of research into evolutionary approaches that use genetic algorithms to come up with intricate architecture designs without human interference. There have been several approaches to addressing the problem of evolving neural networks, including NEAT, DeepHyperNEAT, and CoDeepNEAT. While these approaches have made progress in evolving networks that are able to perform tasks like image classification and image captioning with some success, they have not incorporated the latest state-of-the-art techniques. This is a significant limitation, as using the most advanced techniques available can lead to the generation of neural networks that are more efficient and effective learners. Without incorporating these techniques, neural networks may suffer from over-fitting and require more time and resources for training. Therefore, it is essential to continuously incorporate the latest techniques and methods from deep learning research in order to improve the performance of evolutionary algorithms.

Building on the existing capabilities of the CoDeepNEAT algorithm, this paper proposes the incorporation of an attention mechanism called the Convolution Block Attention Module (CBAM) in order to enhance the ability of the generated CNN topologies to learn the distribution of the training data. By adding CBAM, a state-of-the-art attention mechanism, to CoDeepNEAT, this approach aims to improve the efficiency and effectiveness of the evolved neural networks in learning from the training data.

The attention mechanism [6] is an important development that significantly improved the performance of image classifiers by allowing the neural network to focus on specific parts of the image to make a classification. To the best of our knowledge, the CBAM [10] has not been explored with the CoDeepNEAT algorithm to produce architectures with high fitness scores.

II. Related Work

A. Neuro-evolutionary architectures

In [5], Edgar Galvan et al. elaborate on how evolutionary algorithms can be beneficial towards the design of the architectures of deep neural networks, owing to the fact that they are based on the populations involved and do not depend upon gradients. The search space can be traversed quicker because of the parallelization mechanism offered. One such population based method is Neuroevolution [21] [22] [23], which is an evolutionary algorithm that enables automated training of DNNs along with their automated configuration. A survey has been presented that summarizes how Neuroevolution in DNNs is used to generate neural networks, some being- Recurrent Neural networks, Convolutional Neural Networks, Deep Belief Networks as well as Autoencoders. There is also a mention of the Neuroevolution of Augmenting Topologies [1] (NEAT) algorithm - a very useful technique that can be used to evolve optimal neural networks. After comparison and analysis of the various evolutionary algorithm techniques in the aforementioned survey, NEAT was found to be more suitable due to its structure as well as its available variations.

Kenneth O. Stanley et al. [1] provide meaningful methods of crossover and mutation inspired by the natural evolution process. The NEAT algorithm consists of a genotype which in turn consists of two kinds of genes- one for node representation i.e neurons and the other gene is used to represent the connections between these neurons. The concept of species is also introduced where networks are divided into species based on their complexity. The algorithm starts from a minimally structured population in order to maintain light dimensions in the solution. Although NEAT is unable to develop complex structures and can ideally be used for simple tasks, it serves as a foundation to its extensions such as DeepHyperNEAT [2] as well as CoDeepNEAT [3].

Felix A. Sosa and Kenneth O. Stanley [2] describe DeepHyperNEAT- an extension of HyperNEAT which in turn is an extension of NEAT. The compositional pattern producing networks(CPPNs) in HyperNEAT are augmented with new information, along with mutations to HyperNEAT which make use of the new information augmented. DeepHyperNEAT solves the issue of absence of a method to evolve topology along with the connections in HyperNEAT. DeepHyperNEAT (as well as HyperNEAT) solves the time complexity of direct encoding in NEAT by using an indirectly encoded network but it still has a lot more scope for research- which limits the experiments that can be done with the combination of DeepHyperNEAT and attention.

Through CoDeepNEAT [3], Risto Miikkulainen et al. aim at providing an automated method to configure network architecture- that comprises number of layers, number of neurons, activation functions, etc. which is an alternative to the manual configuration done based on experiments. NEAT is scaled up to work with layers instead of neurons- resulting in DeepNEAT. DeepNEAT is combined with the idea of co-evolution where every generation consists of two populations- modules, which represent deep neural networks and blueprints represent the connection between the modules. CoDeepNEAT takes advantage of the repetitive structure of deep neural networks and also allows for a lot of extensions due to the presence of modules. CoDeepNEAT utilizes CNNs for image related tasks like image captioning- whose accuracy could be improved by involving an attention mechanism.

The effectiveness of CoDeepNEAT for tasks like topology as well as hyperparameter selection is verified in [4] by Jonas da Silveira Bohrer et al. The paper presents an implementation of CoDeepNEAT with the Keras framework as its backend. The implementation has been experimented and run on the CIFAR-10 dataset [20]. The best network generated for CIFAR-10 is obtained at generation number 40 with a training accuracy of 86.5% and a validation accuracy of 79.5%. CoDeepNEAT is shown to produce solutions that are good. This implementation however, cannot be modified easily in order to introduce the aspect of attention in CoDeepNEAT.

B. Attention Mechanism

Attention [6] is an everyday aspect of human evolution often neglected but plays a vital role in our perception of distinct objects. Xu et. al [7] employ attention to the image captioning problem and introduce the categorization of attention based approaches to image datasets . The image captioning model comprises an encoder-decoder architecture with the encoder generating words based on vocabulary and fed to a convolutional neural network to ensure vectors depicting features are generated. The vectors are the earliest depiction of attention in the paper as lower convolutional layers are prioritized ensuring only certain aspects of the image are highlighted.The decoder is a Long Short Term Memory Network(LSTM) [13] ,where a context vector depicting the parts of the image to focus on is generated as an intermediary step. The annotation vectors from the encoder are used to generate the context vectors using a specific mechanism and this in turn gives rise to the development of the two fundamental types of attention in practice for image related tasks or convolutional neural networks.

The attention mechanism makes use of a probability density function(attention map) and performs actions over the image feature maps.If the resultant of the action results in a weighted sum, it is known as soft attention as this depicts coinciding areas of the image divided across regions distinguished by focus(bright or dark) as per time-step.If the resultant is sampled to a specific feature, it is known as hard attention and is stochastic as the selection of region is accomplished by the means of some sampling techniques.

C. Neuroevolution of Image Classifiers

Real et. al [11] began the adoption of evolutionary algorithms applied to image classifiers, hand-made models specifically Convolution Neural Networks gave favourable results to image classification tasks but the implementation and understanding required superseded the time required to generalize such models .Evolutionary approaches had helped cause a shift towards allowing the networks to rapidly improve and generally come close to the state-of-the-art accuracies. However, significant investigation into image classifier based neural networks had always developed a significant research gap. With the advantage of enough computational power, the authors prove that evolutionary based image classifiers come close to existing well perceived models by the deep learning community.

The authors evolve the architectures using elementary crossover and mutation operations on base neural networks with poor performing accuracies or fitness scores. Two networks selected from the population of networks are compared in a tournament selection fashion [5]. The better of the two develops into a parent which has control to the evolutionary operations to produce better networks. This allows for a "survival-of-the-fittest" fashion of great neural networks. The pairwise comparison allows for eager participation from all the networks involved in the population and helps generate a truly evolutionary approach to developing networks. The authors performed model analysis on the CIFAR-10 Dataset and CIFAR-100 Dataset [20] with the former producing accuracy comparable with existing hand-developed models. The authors highlight the weight-sharing nature of neuroevolution allowing for potential parallelism and concurrency of evolutionary jobs. Emphasis on the minimal configurations required for setting up equivalent scaled job allow for exciting future prospects. However, they produce the initial steps of research into the subject matter without much emphasis on the limitations of computing power and hence indicates a research gap into methods of minimizing training time.

D. Optimization of Existing CNN based image Classifiers

Prior work in convolutional neural networks has always placed an emphasis on the spatial nature [8] of CNN's through techniques such as pooling which benefit and help attend the problem of local minima optimization. However, CNN's comprise the channel features as well, which determine the elementary nature of the objects in images. Hu et al [14]. take into account this research gap and propose inter-dependency relations between the channel relations allowing for more tuning of features itself. This results in alteration of the features allowing for selective consideration of only the required parts of the image. The methodology comprises of squashing or squeezing the current spatial features into encodings which are fed to aggregation modules and help produce evolving weights. These weights represent the features and allow the network to condense its options based on the relative scale and requirement of the image classification task. The modules known as Squeeze-Excitation Networks [12] (SENets) are also stackable and allow for improvement in testing cores with minimal increase in computation requirements. The paper highlights the advantages of the shift in prioritization towards channel features and allows for a preliminary direction towards optimization of existing convolutional neural network developments.

NeuroEvolution of Augmenting Topologies (NEAT) algorithm [5] generated neural networks by relating similar parts of different neural networks through mutations and crossovers with a historical marking mechanism. DeepNEAT [1] which is a follow up on NEAT can generate deeper neural networks as it treats an entire layer as a gene in contrast with NEAT that considers a single neuron while forming neural networks. NEAT would pick neurons and the connections between these neurons preventing the algorithm from generating larger and deeper architectures which are required to solve complex problems. On the other hand, DeepNEAT is concerned with the composition of the layers. NEAT and DeepNEAT both employ a historical marking mechanism, which is used to keep track of the changes made in the genetic algorithm structure represented by a graph, these changes occur in the graph over the course of several generations through mutation in which nodes of the graph or the connections between these nodes i.e. the edges of the graph are altered leading to new structures. Using the historical markings mechanism the similarity between these structures is measured and similar structures are called a species, which are subpopulations in a single generation. Every generation the quality of the species is measured using a shared fitness function of its members. The final step is the sharing of genetic information among the fittest members, the surviving species evolve separately and the members of each species exchange genetic information through crossovers.

Despite the effectiveness of the above mentioned neuroevolutionary approaches [1] [2] [3], none have incorporated an attention mechanism. After conducting a thorough literature review, CBAM (Convolutional Block Attention Module) emerged as the most effective attention mechanism for image classification tasks. CBAM outperforms generic attention masks, which only extract specific regions of an image for the classifier to focus on. CBAM's strength lies in its use of both channel and spatial attention modules, which make it more robust and reliable.

III. Background

A. Evolutionary Algorithms

CoDeepNEAT [1] is built on the principles of DeepNEAT as it focuses on layers instead of neurons but unlike DeepNEAT which evolves only the structure of the neural network, CoDeepNEAT splits the construction of neural networks into two sets of populations, the blueprint and the module. Blueprints are graphs that represent the connection between nodes of the graph, where the nodes are the modules. The final network is the result of assembling the modules as per the blueprint which are both evolved separately. CoDeepNEAT follows the same principles of DeepNEAT except for the addition of evolution of modules separately and the evaluation of fitness scores by fitness functions. In CoDeepNEAT a score is assigned to the blueprint and to the modules, the score of

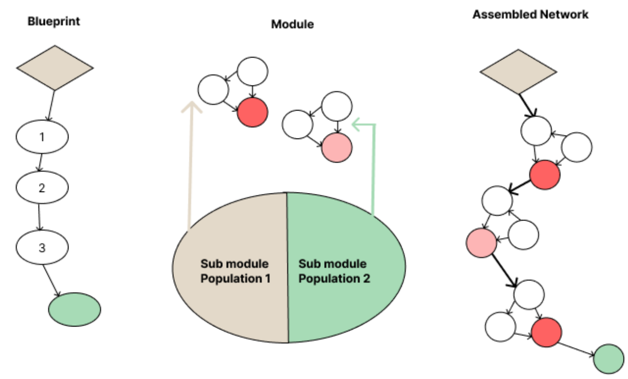

Fig. 1: Structure of CoDeepNEAT

the blueprint must be considered while assigning a score to the modules that make up that blueprint.

In this approach, an additional attention module is added to the CoDeepNEAT algorithm namely the Convolution Block Attention Module (CBAM) [10], a development of Bottleneck Attention Module [9], the intention of adding this module to the CoDeepNEAT algorithm is to raise the bar of the networks being generated as the fitness of the best network is only as good as the modules it's comprised of. The algorithm can pick between the older convolution module or the CBAM module while picking a layer for the neural network.

B. Attention Mechanism

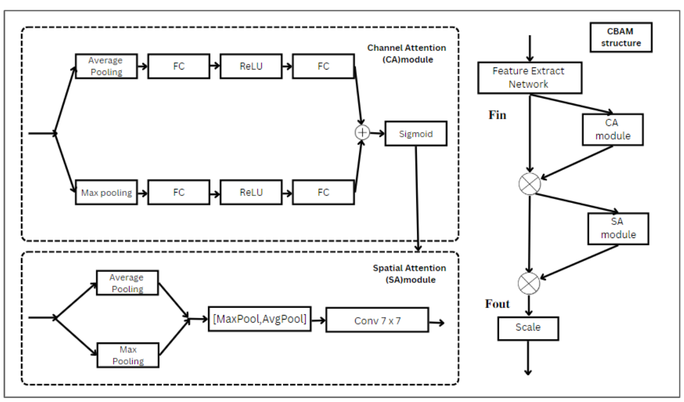

Fig. 2: Structure of CBAM's channel attention and spatial modules, Structure of CBAM

The inclusion of attention mechanisms in evolutionary algorithms is in order to increase the representation power and improve the features selected. The process of feature extraction in convolution operations consists of extraction of the useful features by the merging of spatial as well as channel information. The module network- Convolutional Block Attention Module (CBAM) considers the useful features along both spatial as well as channel dimensional axes. In CBAM [10], there are two modules considered- spatial and channel attention- sequentially applied, thus helping with the flow of information within the network.

The channel attention map is generated by taking advantage of the relationship between channels of features. Channel attention gives importance to "what" is meaningful in the given image that has been sent as input. Max and Average pooling are used in order to form an aggregate of the spatial information- in order to increase efficiency of the channel attention. Two descriptors are produced- one for max- pooled features and one for the average pooled features. These are then sent to a shared Multi Layer Perceptron- that is used on every descriptor. After this, the output feature vectors are combined through addition done element-wise.

The spatial attention map is produced by utilizing the relationship between spatials of features. The spatial attention focuses on "where" the informative parts lie. The methodology to compute spatial attention maps starts with applying max pooling, average pooling along the axis of the channel, followed by concatenation which results in a feature descriptor. Then, the feature descriptor is subjected to a convolutional layer- which generates the resultant spatial attention map.

CBAM is provided with an intermediate feature map **F**. It then goes along the sequence of the channel attention map **MC** , followed by the spatial attention map **MS**. The process can be represented as follows [10]:

$$\boldsymbol{F}' = M_c(F) \otimes F \tag{1}$$

$$\boldsymbol{F}'' = M_s(F') \otimes F' \tag{2}$$

In the above equation, **F"** represents the final output and there is multiplication that is done element wise. The attention values are broadcasted as- channel along spatial dimension as well as spatial along the channel dimension.

IV. Proposed Approach

The goal of this approach is to investigate how incorporating a deep learning technique called attention can improve the efficiency of an evolutionary approach for developing models. The focus is on evaluating the impact of adding attention to the evolutionary process by using the task of image classification. The Convolution Block Attention Module (CBAM) is selected as the attention mechanism to be used because it has been shown to be effective in image classification tasks.

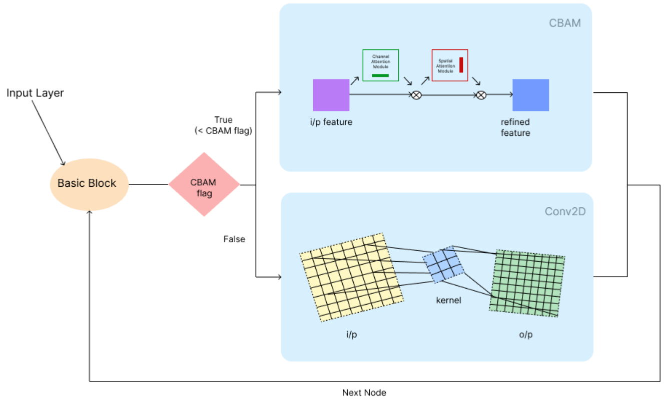

Fig. 3: Algorithm choosing between CBAM and Conv2D

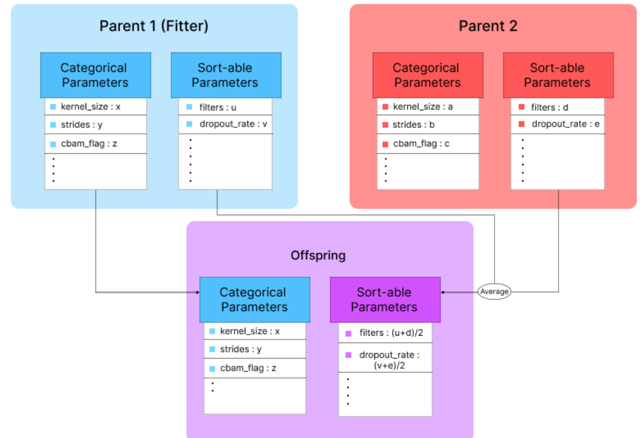

Fig. 4: Crossover Operation: Generation of offspring parameters from parent parameters

This approach involves incorporating the CBAM block into CoDeepNEAT in a way that does not interfere with the overall functioning of the algorithm, allowing CoDeepNEAT to operate freely and independently as it would without the addition of the CBAM block. Essentially, the CBAM block is integrated into CoDeepNEAT without altering the formation of the network topology within the algorithm. This ensures that CoDeepNEAT can continue to evolve and develop efficient models without any constraints or limitations imposed by the inclusion of the CBAM block.

This integration is accomplished by the addition of a hyper-parameter called the CBAM flag. When the evolution process begins, (1) The algorithm must decide which type of layer to use for each neural network in the initial population. The value of the CBAM flag determines the probability of the CBAM block being chosen as a layer in the initial population. (2) The algorithm generates a random decimal number between 0 and 1. If this number is greater than the CBAM flag, a Convolutional layer is chosen, otherwise the CBAM block is selected. This means that a lower value of the CBAM flag results in a lower probability of the CBAM block being chosen in the early stages of the evolution process.

However, as the evolution continues and fitter individuals are produced through crossover as shown in Fig. 4. The categorical parameters and sort-able parameters of the two parents undergo crossover operation. The offspring takes categorical parameter from the fitter parent amongst the two and sort-able parameters are the average values of the sorta-able parameters two parents. It is likely that the number of genomes with CBAM layers will increase compared to those with basic convolution layers. This allows the algorithm to determine whether or not attention is beneficial for the neural networks it is evolving. By gradually introducing the CBAM block through the evolution process, the algorithm can assess its impact on the efficiency and performance of the neural networks being developed.

V. Experiments

While the proposed approach may be computationally more expensive due to the inclusion of the CBAM module, it offers the advantage of not requiring time-consuming techniques such as data augmentation and hyperparameter optimization using Bayesian or random optimization. This trade-off between computational cost and time-saving may make the proposed approach a viable option, depending on the specific needs and resources of the project.

The evolutionary computation for this work was carried out using Google Colab, a cloud-based platform that enables the use of powerful GPUs such as the Tesla P100 with 16GB VRAM and the A100 with 40 GB of VRAM. The evolution took place over a single GPU instance and was performed on the CIFAR-10 dataset with a batch size of 512 and 30 generations. Each generation consisted of 100 genomes, which were trained for 8 epochs using the Stochastic Gradient Descent (SGD) optimizer along with Nesterov momentum set to 0.69. The upper limit of the learning rate for SGD was set slightly higher to compensate for the shortened training time of each genome. Additionally, a CBAM flag of 0.2 was used to incorporate the computationally intensive CBAM layer, although it was set low to mitigate the impact on performance. Overall, this setup allowed for efficient and effective evolution of the topology of the neural network.

It is important to distinguish between the training of genomes during the evolutionary process and the training of the best genome at the end of evolution. During evolution, the fitness scores of the genomes serve as a heuristic or guide to help the evolutionary algorithm identify topologies that are more likely to be effective. However, the best genome at the end of evolution requires more extensive training, as it will be the model that is ultimately deployed in practice.

Due to the time-intensive nature of evolution and the limited run-time available on Google Colab, this approach included the use of deserialization to save and restore the state of

TABLE I: Test Accuracy on CIFAR-10 Dataset

Work in Literature	Parameters	Accuracy	Hyperparameter Optimization	Data Augmentation
Aszemi et al. [15]	-	80.62	Yes	No
Zhao et al. [16]	-	86.33	No	Yes
Evangelista et al. [17]	-	84.6	No	No
Thite et al. [18]	-	75.59	Yes	No
Real et al. [19]	5.4M	94.6	-	-
Proposed Approach	2.625M	87.44	No	No

the evolutionary process between run-times. This allowed the evolution to continue over multiple run-times, covering all 30 generations.

VI. Results & Discussion

The proposed approach was trained on a single GPU instance and the evolution of a generation took approximately one hour on average. The evolution process ran for a total of 30 generations, and the fittest topology was evolved in the 14th generation. During the evolution process, each genome was trained for 8 epochs using stochastic gradient descent with momentum as the optimizer. A total of 100 genomes were included in each generation of evolution. The fittest genome, identified at the end of the evolution process, was further trained for 200 epochs using the default configuration of the SGD optimizer. This resulted in an accuracy score of 87.44% on the CIFAR-10 test set.

One of the major advantages of the proposed approach is its efficiency and effectiveness in generating highly performant CNN topologies. This approach requires fewer generations of evolution compared to past approaches, yet it still produces results that are superior to most of these approaches. Additionally, this approach requires fewer computational resources compared to other approaches, making it a more efficient and cost-effective solution. Even though the approach by Real et al. [19] outperformed this proposed approach, the fittest model resulting from their approach contained more than twice the number of trainable parameters compared to the model generated by this approach. This demonstrates the potential for this proposed approach to generate highly performant models with a smaller number of trainable parameters, which can be beneficial for reducing computational cost and improving generalization.

An additional significant piece of information that researchers who wish to research evolutionary algorithms must consider is that the evolution of neural networks can be guided by factors important to the real-world application as the networks are only partly trained during evolution. Which means that, this evolution set-up favours faster learners instead of topologies that may have higher fitness over a larger training period. Thus, it's possible to direct evolution such that the resulting network is suitable for the requirements of the real-world, in cases where accuracy may be traded off for training time or memory requirements that the network must fulfil. Today, the cost of computational resources is a non-issue for all the big tech players in the market making evolutionary approaches perfectly suited to the needs of building neural networks that can be employed on grand scale with minimal fuss and the findings of this paper support this conclusion and it's likely that using this approach neural network topologies could be evolved that are of a higher-calibre compared to networks designed by hand.

VII. Conclusion & Future Scope

The incorporation of the CBAM block, a proven technique for improving the performance of neural networks, into evolutionary algorithms has been shown to be an effective way to improve their performance. These advanced techniques enable the evolution of neural networks that are more efficient learners, resulting in models that give comparable or even superior performance to past approaches with fewer trainable parameters, shorter training times, and lower computational requirements. This provides strong evidence for the effectiveness of using state-of-the-art techniques in evolutionary algorithms, as they can lead to the generation of highly performant neural networks that can learn effectively from training data.

There are several directions for future work on the proposed algorithm. One possibility is to enhance its ability to process unstructured data, such as text, which could be used in Natural Language Processing tasks like image captioning and text classification. To achieve this, the proposed approach could be modified to include LSTM layers and a new attention mechanism for encoder-decoder/sequence-to-sequence architectures. However, this expansion may require additional computational resources to maintain efficiency as a new attention mechanism for encoder-decoder/sequence-to-sequence architectures needs to be introduced. Additionally, further research could be conducted to optimize the algorithm and improve its performance on complex real-world tasks. By broadening the range of operations and optimizing computational efficiency, the proposed approach has the potential to be a powerful tool for addressing a variety of challenges in the field of Artificial Intelligence.

References

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

[2] Sosa, Felix A., and Kenneth O. Stanley. "Deep HyperNEAT: Evolving the size and depth of the substrate." Tech. Rep. (2018).

[3] Miikkulainen, Risto, et al. "Evolving deep neural networks." Artificial intelligence in the age of neural networks and brain computing. Academic Press, 2019. 293-312.

[4] J. Da, S. Bohrer, B. I. Grisci, and M. Dorn, "NEUROEVOLUTION OF NEURAL NETWORK ARCHITECTURES USING CODEEPNEAT AND KERAS SBCB LAB -SBCB.INF.UFRGS.BR," Arxiv.org, 2020.

[5] Galván, Edgar, and Peter Mooney. ”Neuroevolution in deep neural networks: Current trends and future challenges.” IEEE Transactions on Artificial Intelligence 2.6 (2021): 476-493.
[6] Bahdanau, D., Cho, K., Bengio, Y. ”Neural machine translation by jointly learning to align and translate” , ICLR (2015)
[7] Xu, Kelvin, et al. ”Show, attend and tell: Neural image caption generation with visual attention.” International conference on machine learning. PMLR, 2015.
[8] Chen, Long, et al. ”Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[9] Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module. In: Proc. of British Machine Vision Conference (BMVC). (2018)
[10] Woo, Sanghyun, et al. ”Cbam: Convolutional block attention module.” Proceedings of the European conference on computer vision (ECCV). 2018.
[11] Y. Sun, B. Xue, M. Zhang and G. G. Yen, ”Evolving Deep Convolutional Neural Networks for Image Classification,” in IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, pp. 394-407, April 2020, doi: 10.1109/TEVC.2019.2916183.
[12] Hu, Jie, Li Shen, and Gang Sun. ”Squeeze-and-excitation networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[13] Yu, Yong, et al. ”A review of recurrent neural networks: LSTM cells and network architectures.” Neural computation 31.7 (2019): 1235-1270.
[14] W. Chen, D. Xie, Y. Zhang, and S. Pu, “All you need is a few shifts: Designing efficient convolutional neural networks for image classification,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[15] Aszemi, Nurshazlyn Mohd, and P. D. D. Dominic. ”Hyperparameter optimization in convolutional neural network using genetic algorithms.” International Journal of Advanced Computer Science and Applications 10.6 (2019)
[16] Zhao, Alan, Harsha Uppili, and Gabriel Meyer-Lee. ”Comparing Evolutionary Algorithms for Deep Neural Networks.”
[17] Evangelista, Lucas Gabriel Coimbra, and Rafael Giusti. ”Short-term effects of weight initialization functions in Deep NeuroEvolution.” Evo* 2021: 21.
[18] Thite, Anish, et al. ”Concurrent neural tree and data preprocessing AutoML for image classification.” Proceedings of the Genetic and Evolutionary Computation Conference Companion. 2021.
[19] Real, Esteban, et al. ”Large-scale evolution of image classifiers.” International Conference on Machine Learning. PMLR, 2017.
[20] Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
[21] Stanley, Kenneth O., et al. ”Designing neural networks through neuroevolution.” Nature Machine Intelligence 1.1 (2019): 24-35.
[22] Baldominos, Alejandro, Yago Saez, and Pedro Isasi. ”On the automated, evolutionary design of neural networks: past, present, and future.” Neural computing and applications 32.2 (2020): 519-545.
[23] Stanovov, V., Sh Akhmedova, and E. Semenkin. ”Neuroevolution of augmented topologies with difference-based mutation.” IOP Conference Series: Materials Science and Engineering. Vol. 1047. No. 1. IOP Publishing, 2021