


Letter

# Variable Chromosome Genetic Algorithm for Structure Learning in Neural Networks to Imitate Human Brain

Kang-moon Park <sup>1,†</sup>, Donghoon Shin <sup>2,†</sup>  and Sung-do Chi <sup>3,\*</sup>

<sup>1</sup> Department of Computer Science, College of Natural Science, Republic of Korea Naval Academy, Changwon-si 51704, Korea

<sup>2</sup> Mercedes-Benz Research & Development Korea, Advanced Driver Assistance Systems, Seoul 04637, Korea

<sup>3</sup> Department of Software, Korea Aerospace University, Goyang-si 10540, Korea

\* Correspondence: sdchi@kau.ac.kr; Tel.: +82-2-300-0184

† These authors contributed equally to this work.

Received: 13 June 2019; Accepted: 31 July 2019; Published: 5 August 2019



**Featured Application:** Neural Network Structure Learning is expected to overcome difficulty of constructing a neural network structure.

**Abstract:** This paper proposes the variable chromosome genetic algorithm (VCGA) for structure learning in neural networks. Currently, the structural parameters of neural networks, i.e., number of neurons, coupling relations, number of layers, etc., have mostly been designed on the basis of heuristic knowledge of an artificial intelligence (AI) expert. To overcome this limitation, in this study evolutionary approach (EA) has been utilized to automatically generate the proper artificial neural network (ANN) structures. VCGA has a new genetic operation called a chromosome attachment. By applying the VCGA, the initial ANN structures can be flexibly evolved toward the proper structure. The case study applied to the typical exclusive or (XOR) problem shows the feasibility of our methodology. Our approach is differentiated with others in that it uses a variable chromosome in the genetic algorithm. It makes a neural network structure vary naturally, both constructively and destructively. It has been shown that the XOR problem is successfully optimized using a VCGA with a chromosome attachment to learn the structure of neural networks. Research on the structure learning of more complex problems is the topic of our future research.

**Keywords:** variable chromosome genetic algorithm; deep learning; structure learning; artificial neural network; neural architecture search

## 1. Introduction

The purpose of artificial intelligence (AI) is to imitate the human intelligence [1]. In order to imitate human intelligence, the research of AI has been developed as an algorithm to figure out a certain issue. Artificial neural network (ANN) is one of these algorithms. In other words, the development of ANN is to imitate the process of the human brain [2]. The ANN is already widely used for an estimation of automotive brake pressure, which has a great potential to achieve a simple design of the braking control system without sensors [3].

The compute unified device architecture (CUDA) is a parallel processing technology that used GPU. Since the introduction of the CUDA, deep learning has been rapidly developed and applied to various fields in the industry. Recently, big tech companies and car makers such as Google, BMW, and Tesla have expanded their interests to research in deep learning [4]. In particular, the convolutional

neural network (CNN) is widely implemented to image recognition [5,6] and the recurrent neural network (RNN) is used for various applications such as sentence and voice recognition [7].

As ANNs have been scaled up and improved, they have become much more complex [8]. Even to realistically imitate a human brain, a system must be able to acquire the individual personality models of that individual [9]. Although human engineers are able to tune and optimize parameters through experimentation, neural networks are prone to have complex topologies or hyper-parameters. Previously, the design of neural networks was based on heuristics [10]. Since the structure significantly varies according to the number of neurons or the purpose of the problem, the optimal structure is hard to find.

Currently, employed architectures have mostly been developed by human experts, which is a time-consuming and error-prone process [11]. As deep learning has scaled up to more challenging tasks, the architectures have become difficult to design by hand [8]. For these reasons, various studies have been carried out to generate the network architecture automatically [11,12]. Structure learning is a very useful instrument that is able to automatically find an appropriate ANN architecture [13]. For this reason, the structure learning algorithm is used to generate the topology of ANN in this study.

This study proposed a chromosome attachment as a new genetic operation to implement the structure learning of ANN. It makes the ANN vary both constructively and destructively. Through this operation, a structure learning method in the process of a neural architecture search (NAS) is possible. NAS has three challenges: Designing a network structure, putting components into the network topology, and hyper-parameters tuning [8]. The proposed method in this study presents a challenge through structure learning.

Evolutionary approach (EA) is usually used to automatically design the topology of an ANN topology. There are many neural network architecture search algorithms with an EA. Generally, constructive or destructive algorithms are used to automatically design architectures [14]. However, most EAs only use a constructive algorithm.

A typical constructive algorithm starting with a minimal network is the genetic algorithm (GA). GA has been the most common method used for designing neural network architecture [11–18]. In these studies, GA was used to generate the network structures depending on their purposes. It can be applied to a system having a network structure, such as a Bayesian network or a neural network.

In most research focused on the automatic generation of the neural network structure, they were carried out to control the number of neurons through a genetic algorithm in a network structure that has one hidden layer [13,16,17]. In these studies, the neurons belonging to the hidden layer are increased or decreased according to generations while searching the suitable number of neurons.

NeuroEvolution of Augmenting Topologies (NEAT) is a representative EA methodology. NEAT outperforms the best fixed topology methodology on challenging benchmarking reinforcement learning [15]. NEAT is used as a typical evolutionary approach [8]. The constructive algorithm uses GA, starting with a minimal structure of artificial neural network. The initial neural network has a minimal structure, however it becomes gradually more complex to search an optimal architecture with NEAT methodology.

On the other hand, NEAT methodology only can evolve the neural network architecture to a larger structure because it has to start with a minimal structure. If the neural network architecture in the solution area is too large to increase gradually, it is time consuming to search a right answer. Thus, destructive operation is necessary to design a more complex or larger neural network.

Variable chromosome genetic algorithm (VCGA) is applied to this method to implement the structure learning of ANN so as to imitate a human brain. The previous structure learning method of the ANN could only evolve constructively. However, VCGA uses chromosome attachment for a system that is able to evolve the structure of an ANN, not only constructively but also destructively. Destructive method like pruning is necessary. Since the size of the ANN is unknown at the beginning, design method such as pruning is required. Furthermore, the structure learning of a big neural network spends a large amount of time [16,19].

The proposed method is applied to the problem of generating a XOR network. Although this task is a simple problem, it requires growing and pruning hidden neurons [15].

The rest of this paper is organized as follows. Section 2 describes the proposed methodology of applying VCGA to neural network structure learning. Section 3 analyzes the results of simple case study. Finally, conclusions and future works are drawn in Section 4.

## 2. Methodology

As discussed in Section 2.2, previous methodologies typically start with a minimal structure of neural network in order to become an optimized architecture. In contrast, VCGA with a chromosome attachment is used to generate a neural architecture both constructively and destructively in this study. The chromosome attachment operation gives a destructive approach to GA. It makes possible for neurons and ls to increase or decrease various directions in order to search an optimized architecture on a large solution area. Thus, it is more suitable to search a large and complex architecture of neural network.

### 2.1. Structure Learning Based on Variable Chromosome Genetic Algorithm

In this study, VCGA is used to search the neural network architecture. Figure 1 provides an overview of methodology used in this paper in the structure learning process. In Phase 1, the chromosome set is generated first. These chromosomes represent neurons and connections. An individual has multiple chromosomes and it composes an ANN structure. In Phase 2, each generated ANN structures are simulated. Simulation results are calculated by fitness function and ANN structures are sorted by result score. In Phase 3, genetic operations are implemented to make offspring: Selection, cross over, mutation, and attachment. These new generated chromosome set make new individuals and an ANN structure on Phase 1. Equation (1) shows a fitness function.

$$Fitness\ value = \frac{1}{2}score - \frac{1}{2}D \times N\{2 \times (score - 0.5)\} \tag{1}$$

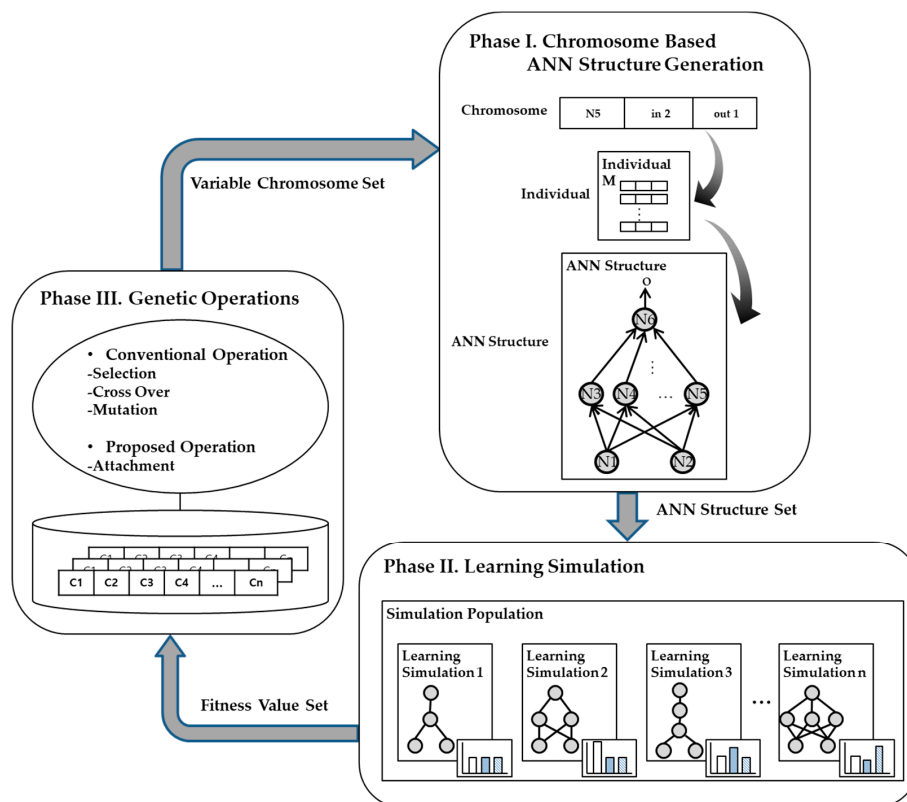


Figure 1. Overall methodology.

Score means the accuracy of predicted output value of the ANN structure and it has a real value between 0.0 and 1.0.  $D$  means a dependency on the number of chromosomes. The larger  $D$  is, the more important the minimum structure of the ANN is. And the smaller  $D$  is, the more important the accuracy of the predicted output value is.  $N$  means relative number of chromosomes;  $(\text{number of own chromosomes})/(\text{number of average chromosomes})$ . In this case study, the constant  $D$  was set to 0.2.

2.2. Chromosome Type of Artificial Neural Network

In order to encode an ANN into a chromosome, ANN is represented as two cases of chromosome types. A chromosome can be expressed as an ANN topology [20].

Figure 2 shows two types of chromosome type: Neuron information and connection information. The chromosome type of neuron information consists of the unique number of neuron, the number of the input ports, and the number of the output ports as shown in Figure 2a.

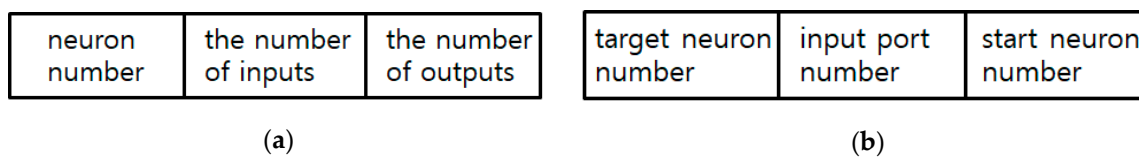


Figure 2. Chromosome type of neural network: (a) Chromosome type of neuron information and (b) chromosome type of linking information.

The chromosome type of connection information consists of the unique number of target neuron, the number of the port to input, and the unique number of the starting neuron as shown in Figure 2b.

Figure 3 shows examples of chromosomes and ANN topologies. The chromosome type of the neuron N5 is shown in Figure 3b, assuming that there is a neural network of the ANN topology shown in Figure 3a. It means the neuron N5 has two input ports and one output port. The chromosome type of connection information is shown in Figure 3c. It is a connection from N3 to N5 by port 1 of N5.

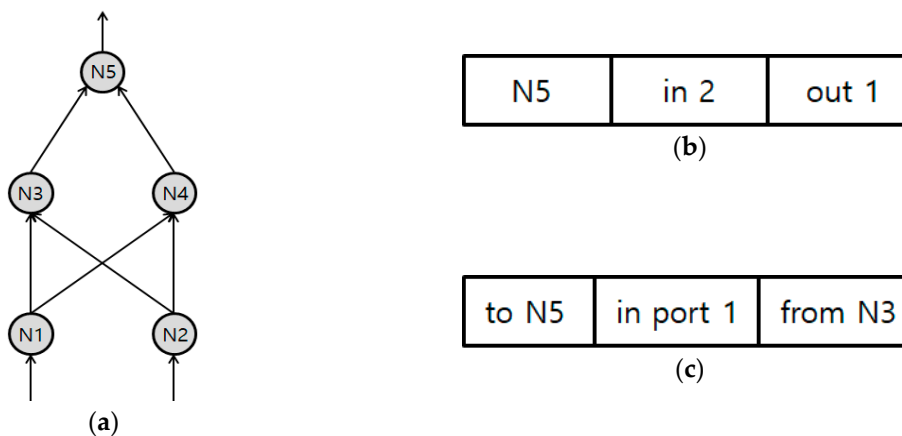


Figure 3. Example of artificial neural network (ANN) topology and chromosomes: (a) Example of ANN topology; (b) example of neuron information ANN topology; and (c) example of linking information ANN topology.

Chromosomes composed of these two types of chromosome constitute a single individual. The number of chromosomes in an individual means the sum of the number of chromosomes that store neuron information and the number of chromosomes that store connection information.

Figure 3 shows examples of chromosome types and ANN topologies. The chromosome type of the neuron N5 is shown in Figure 3b, assuming that there is a neural network of the ANN topology shown in Figure 3a. It means the neuron N5 has two input ports and one output port. And the connection from N3 to N5 has a chromosome type of the form shown in Figure 3c.

Chromosomes composed of these two types of chromosome types constitute a single entity. The number of chromosomes in an individual means the sum of the number of chromosomes that store neuron information and the number of chromosomes that store link information.

### 2.3. Application of Genetic Operation

The genetic operations applied in VCGA are mutation, cross-over, and chromosome attachment [21]. Mutation and cross-over can change the contents of chromosomes as shown in Figure 4. It changes the number of input ports or output ports of existing neurons, and target or start the neuron of existing connections. For example, mutation modifies a selected gene of a chromosome randomly as shown in Figure 4a and cross-over exchanges a gene of each parent as shown in Figure 4b.

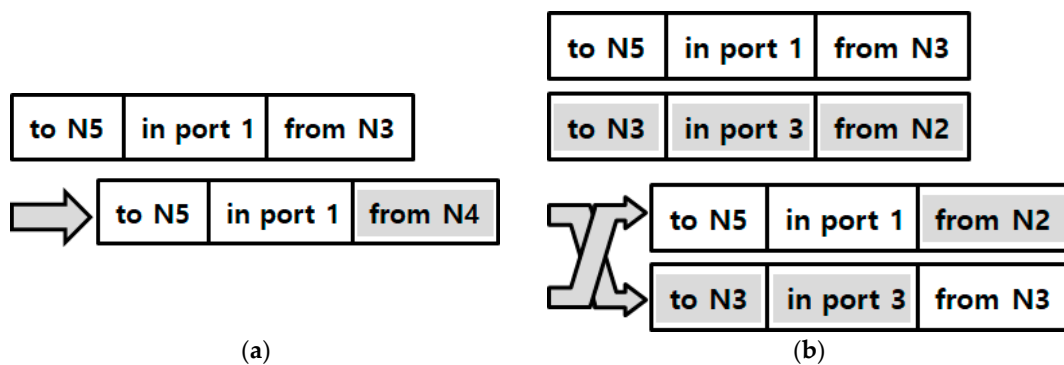


Figure 4. Mutation and crossover: (a) Example of mutation and (b) example of crossover.

On the other hand, chromosome attachment adds or discards a neuron or a connection between neurons by changing the number of chromosomes in an individual as shown in Figure 5. The mutation in biology has two types: Gene mutation and chromosome mutation. Well-known mutations (in biology) like mutation (in GA) or cross-over can be treated as a kind of gene mutation which changes the base sequence of DNA [22]. However, chromosome mutation only changes the number of chromosomes. Chromosome attachment is a chromosome mutation [23]. It is also called nondisjunction in biology [24]. It makes a change that varies the number of chromosome offspring. This is a kind of mutation that causes chromosome aberration. However, through this operation both new neurons and connections may appear or existing neurons may disappear. Since there are no restrictions on the layer, theoretically there may also be multi-hidden layers and skip connections. This operation is first applied to the GA and EA through this study.

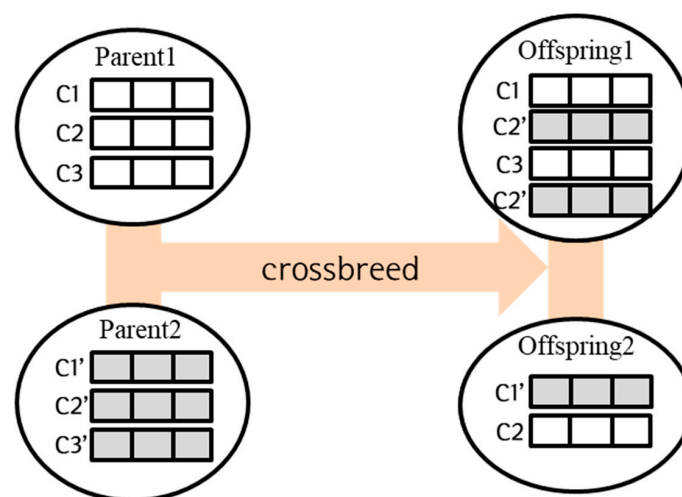


Figure 5. Chromosome attachment.

### 3. Case Study: XOR Problem

As a case study, the proposed methodology was applied to solve the XOR problem [25]. The XOR problem is the most fundamental form of the problem in multilayer perceptron. Although the correct answer is already a well-known problem, it is very important for the imitation of the human brain. The proposed methodology is compared to NEAT. The simulation automatically found the optimized structure for the XOR without giving any information about the correct answer. If it succeeds in finding the correct answer, it should have five chromosomes of neuron information, six chromosomes of connection information, and 11 chromosomes in total.

Two simulations for verifying a chromosome attachment operation applied on EA of ANN were implemented. The first simulation was start from a minimal structure and was compared to the NEAT methodology. The second simulation started from a large structure. It verified that the chromosome attachment operation was useful for auto-generating large and complex ANN structure.

#### 3.1. Initialization

NEAT began with an initial population of uniform topology. In contrast, the proposed methodology began with a random population. The initial populations generated 200 individuals. The average number of chromosomes in the first generation was about 8 in the first simulation. In the second simulation, the average number of chromosomes in the first generation was about 19 since they have larger structures than the first simulation. The probabilities of mutation, cross-over, and attachment were all 0.2 and the learning rate of the neural network was 0.2. This value was approximate to NEAT [15]. The simulation was implemented for a total of 300 generations in the first simulation and 2000 generations in the second simulation.

#### 3.2. Simulation Result

Figure 6 plots the score and the number of chromosomes of the neural network structure with the highest score for each generation of the first simulation. The score of the highest performing species began from 0.65 and increased to 0.95. If a score of 0.9 or higher was obtained, the generated neural network could correctly obtain the exclusive or (XOR) value. This system found a structure for XOR in 48 generations. This solution structure had 32 chromosomes. As the generation progressed, the number of chromosomes gradually decreased. It took 218 generations to find the optimal form with 11 chromosomes for the first time. Since then, it appeared to oscillate and converge.

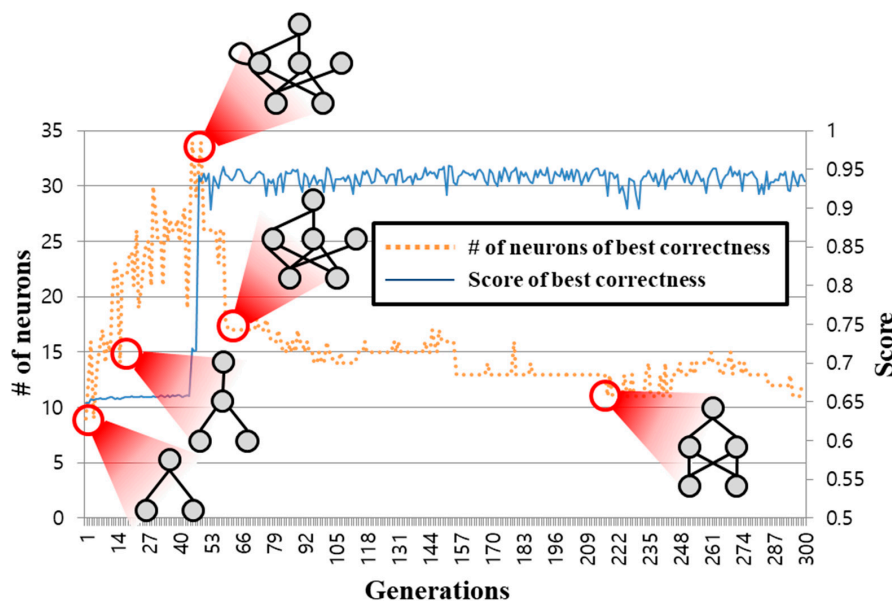


Figure 6. Simulation result of first simulation.

Figure 7 plots the results of the second simulation. In this simulation, species of the first generation were fully connected with 5 neurons. The score of the highest performing species began from 0.97 and decreased to 0.95, however, it increased again to 0.97. This system started from a completed ANN structure. However, it decreased the number of chromosomes after starting. Neurons and connections decreased together until the number of chromosomes reached 11.

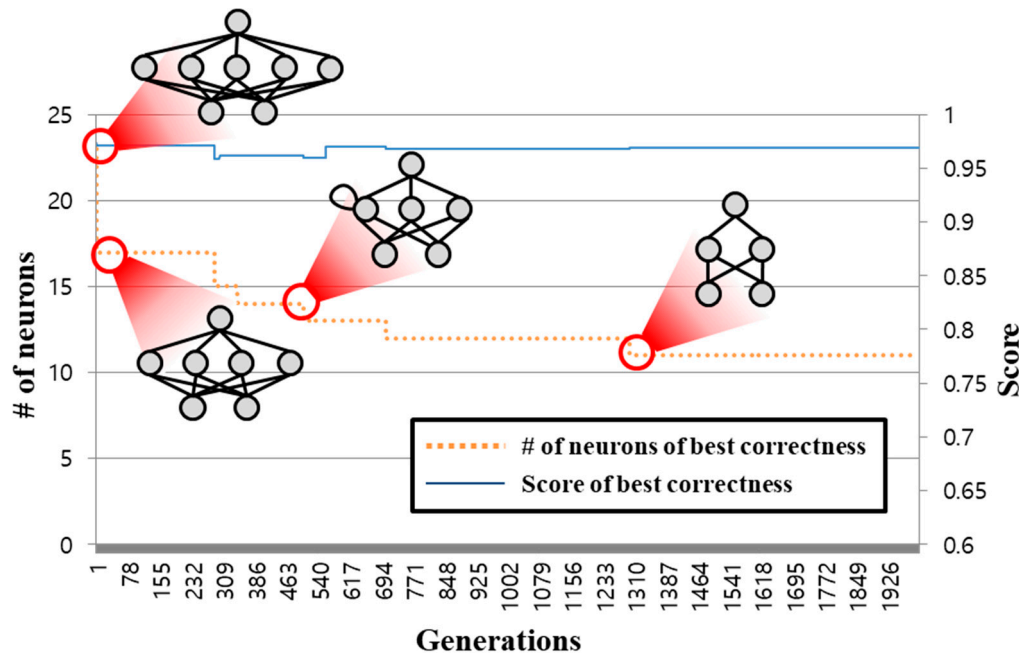


Figure 7. Simulation result of second simulation.

Table 1 shows a comparison between the results of structure learning with NEAT and the proposed method about the XOR problem. The NEAT methodology found a solution faster than the proposed methodology using VCGA. However, NEAT evolved networks with 2.35 hidden neurons and 7.48 disabled connections, while the proposed method’s solutions used 2 hidden neurons and 0 disabled connections. Since NEAT only uses the add operation, it cannot evolve with the destructive approach. On the other hand, the proposed methodology with VCGA allows decreasing neurons or connections with the chromosome attachment. The result is important because it shows that a methodology using VCGA has the potential to be applied to large and complex problems while performing as well as NEAT.

Table 1. Comparison with NeuroEvolution of Augmenting Topologies (NEAT) in exclusive or (XOR) problem.

Method	Average of Generation Finding Solution	Average Number of Hidden Nodes	Average Number of Disabled Connections	Evolution Directed to Decreasing
NEAT	32	2.35	7.48	impossible
using variable chromosome genetic algorithm (VCGA)	about 40	2	0	possible

#### 4. Conclusions

In this study, the structure learning methodology for an ANN using a VCGA with a chromosome attachment was proposed. Previous methods including NEAT only use the constructive approach. Thus, it only can design an ANN from a minimal structure. Due to this limit, the previous methodology

has difficulty in designing large structures. However, the proposed method enabled the evolutionary algorithm to destructive approach by using the chromosome attachment. As a result, the given initial ANN could be tuned to auto-design the structure after structure learning, whether the initial ANN was a minimum structure or not. It is important to note that the approach could learn a larger ANN.

The proposed methodology evolved the ANNs, starting with a minimal structure and completed structure. It succeeded in finding the right answer to the XOR problem without any information as part of a simple case study that compared it with the previous method. On the other hand, using the proposed methodology, it was possible to generate more complex neural network structures faster because of the bout constructive and destructive approach. It is expected that the cost of using specialists in designing ANN structures will also reduce.

The expansion of searching space to more complex ANN such as image recognition and situation awareness of auto driving system is the topic of our future research.

**Author Contributions:** Conceptualization, K.-m.P., D.S., and S.-d.C.; Methodology, K.-m.P.; Software, K.-m.P.; Validation, K.-m.P. and D.S.; Formal Analysis, D.S.; Investigation, K.-m.P.; Resources, S.-d.C.; Data Curation, K.-m.P.; Writing—Original Draft Preparation, K.-m.P.; Writing—Review and Editing, D.S.; Visualization, K.-m.P. and D.S.; Supervision, S.-d.C.; Project Administration, K.-m.P.; Funding Acquisition, S.-d.C.

**Funding:** This research was supported by the 2019 National Research Projects of Naval Research Center, Republic of Korea Naval Academy.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Sun, T.; Geng, G.H. Image Retrieval Method for Deep Neural Network. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2016**, *9*, 33–42.
2. Das, B.; Roy, S.; Rai, R.; Saha, S. Studies on Effect of Cutting Parameters on Surface Roughness of Al-Cu-TiC MMCs: An Artificial Neural Network Approach. *Procedia Comput. Sci.* **2015**, *45*, 745–752. [CrossRef]
3. Shin, D.; Park, K.; Park, M. Effects of Vehicular Communication on Risk Assessment in Automated Driving Vehicles. *Appl. Sci.* **2018**, *8*, 2632. [CrossRef]
4. Survey of Neural Networks in Autonomous Driving. Available online: [https://www.researchgate.net/publication/324476862\\_Survey\\_of\\_neural\\_networks\\_in\\_autonomous\\_driving](https://www.researchgate.net/publication/324476862_Survey_of_neural_networks_in_autonomous_driving) (accessed on 20 July 2019).
5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process Syst.* **2012**, *25*, 1097–1105. [CrossRef]
6. Lawrence, S.; Giles, C.; Tsoi, A.C.; Back, A. Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Netw.* **1997**, *8*, 98–113. [CrossRef] [PubMed]
7. Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent Neural Network Based Language Model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, 26 September 2010.
8. Miiikkulainen, R.; Liang, J.; Meyerson, E.; Rawal, A.; Fink, D.; Francon, O.; Raju, B.; Shahrzad, H.; Navruzyan, A.; Duffy, N. Evolving Deep Neural Networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 293–312.
9. Hamanaka, M.; Goto, M.; Asoh, H.; Otsu, N. A Learning-Based Jam Session System that Imitates a Player's Personality Model. In Proceedings of the International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9–15 August 2003; pp. 51–58.
10. Anderson, M. Using AI to Make Better AI. Available online: <https://spectrum.ieee.org/tech-talk/computing/networks/using-ai-to-make-better-ai> (accessed on 12 April 2019).
11. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *J. Mach. Learn. Res.* **2019**, *20*, 1–21.
12. Poza, M.; Yurramendi, Y.; Murga, R.; Kuijpers, C.; Larrañaga, P. Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 912–926.
13. Kwok, T.Y.; Yeung, D.Y. Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Trans. Neural Netw.* **1997**, *8*, 630–645. [CrossRef] [PubMed]
14. Yao, X. Evolving artificial neural networks. *Proc. IEEE* **1999**, *87*, 1423–1447.



15. Stanley, K.O.; Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. *Evol. Comput.* **2002**, *10*, 99–127. [[CrossRef](#)] [[PubMed](#)]
16. Yang, S.H.; Chen, Y.P. An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications. *Neurocomputing* **2012**, *86*, 140–149. [[CrossRef](#)]
17. Miller, G.F.; Todd, P.M.; Hegde, S.U. Designing Neural Networks using Genetic Algorithms. *Proc. ICGA* **1989**, *89*, 379–384.
18. Kandasamy, K.; Neiswanger, W.; Schneider, J.; Poczos, B.; Xing, E.P. Neural architecture search with bayesian optimisation and optimal transport. *Proc. Adv. Neural Inf. Process. Syst.* **2018**, *31*, 2016–2025.
19. Kwok, T.Y.; Yeung, D.Y. Constructive feedforward neural networks for regression problems: A survey. *HKUST-CS95* **1995**, *43*, 1–29.
20. Morris, G.M.; Goodsell, D.S.; Halliday, R.S.; Huey, R.; Hart, W.E.; Belew, R.K.; Olson, A.J. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.* **1998**, *19*, 1639–1662. [[CrossRef](#)]
21. Park, K.M.; Shin, S.H.; Chi, S.D. The Genetic Algorithm using Variable Chromosome with Chromosome Attachment for decision making model. *J. Korea Soc. Simul.* **2017**, *26*, 1–9.
22. Berg, J.M.; Tymoczko, J.L.; Stryer, L. Mutations Involve Changes in the Base Sequence of DNA. In *Biochemistry*, 5th ed.; W. H. Freeman and Company: Bethesda MD, USA, 2002.
23. Rieder, C.L. Kinetochores are transported poleward along a single astral microtubule during chromosome attachment to the spindle in newt lung cells. *J. Cell Biol.* **1990**, *110*, 81–95. [[CrossRef](#)] [[PubMed](#)]
24. Shonn, M.A. Requirement of the Spindle Checkpoint for Proper Chromosome Segregation in Budding Yeast Meiosis. *Science* **2000**, *289*, 300–303. [[CrossRef](#)] [[PubMed](#)]
25. Minsky, M.L.; Papert, S.A.; Perceptrons, F. *Mass*, rev. ed.; The MIT Press: Cambridge, MA, USA, 1969.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).