

# The Modularity in Freeform Evolving Neural Networks

Shuguang Li

School of Astronautics  
Northwestern Polytechnical University  
Xi'an, Shaanxi, 710072, P.R. China  
Email: lishg81@gmail.com

Jianping Yuan

School of Astronautics  
Northwestern Polytechnical University  
Xi'an, Shaanxi, 710072, P.R. China

**Abstract**—In this paper, we validate whether the network modularity can emerge, and the evolution performance can be improved by varying the environment or evolution process under a more freeform artificial evolution. Previous studies have demonstrated that the modular structure naturally arisen as a response of the variations on environment and selection process, however, since the models they used were relatively simple and with some biasing constraints, the results may lack of generality. In contrast, we evolve more freeform neural networks to address this issue, and an artificial tracer method was employed to quantify the modularity. A series of varying scenarios have been experimented, the results show that the evolution performance have been improved in most cases, however, the modularity never appeared among those scenarios. A further experiment shows that our method has the potentials to produce modular networks but the more advanced methods are still needed to encourage the emergence of modularity on the complex questions.

**Keywords**—modularity; neural networks; evolutionary computation

## I. INTRODUCTION

Modularity is a common property of complex systems, which has been widely discovered in both natural organisms and engineering systems. Particularly, the complex networks, range from biological networks to social networks, also exhibit their modularity through distinguishable community structures within the entire networks [1]. Although it has been well known that the modularity is beneficial for the evolvability and robustness of complex systems, however the origin of modularity remains to be revealed, then how the modularity emerges in complex systems and how it affects the system during the developmental process have been frequently studied [2].

Artificial evolution provides us a platform for investigating above open questions through combining the evolutionary algorithms and computer simulation. During the evolution, the individuals are commonly treated as controllers, and the goal of evolution is to adapt the controller's structure and function for gaining a higher fitness on specific mission. In order to represent the controller and its environment, a simplified linear matrix model has been proposed, which treated the controller, goal, and environment as different matrixes. Artificial neural networks (ANNs) [9], which are inspired by mammalian neural

systems, have also been extensively used to model the controller with complex nonlinear properties for evolutionary simulation [10]. Regardless of different approaches, it is generally believed that the modularity might be an outcome of evolution through natural selection, environment variation and noise. Moreover, some experiments also demonstrated that the modularity could speed up the evolution at some certain conditions [3].

However, previous models, either linear matrixes or ANNs, used in evolution are relatively simple. The controller and environment have been simplified through the low dimension matrixes, and the ANNs models have also been constrained by the predefined structures or the given building rules. Such limitations could significantly decrease the search space of evolutionary algorithms whereas the simulation results then lack of generality. There exists challenge that whether we can find similar conclusions by evolving the more complex ANNs from scratch with a more freeform evolutionary process. Using more complex networks, we can represent more systems at a higher level; furthermore, the freeform evolution is more approximate to the natural process. Thus, the solution of this challengeable question will reveal a clearer mechanism of modularity behind evolution and then bring us a greater field of applications.

We attempt to address this challenge by conducting a well studied retina recognition experiment. The pure topological recurrent neural networks are employed as the controllers and these networks are encoded by a direct graph, and then a group of genetic operators are implemented to train these controllers. The main purposes of this work are to validate whether the modularity will finally arise in these networks and whether the evolution could be accelerated as a response to variation of environment and natural selection. An artificial tracer method is proposed to roughly measure the modularity of networks. Then the characters of modularity and evolution could be investigated by comparing a series of experiments under different conditions.

This paper is organized into five sections: In Section II, We give a brief review of previous studies which related to this issue. The network model, evolutionary algorithm and the quantitative method of network modularity are proposed in Section III. We then describe the different experiments and show their results in Section IV. A discussion of the

experimental study is presented Section V. We finally conclude our work and outline a future plan in Section VI.

## II. BACKGROUND AND PREVIOUS WORK

Artificial evolution approach has been a most popular way to investigate the modularity and its effects of the complex systems. The basic idea is to use a model to represent the system/individual, environment and the specific mission, and then the evolutionary computation method is employed to simulate the natural selection process. This computer based method can then be seen as a simulation of natural evolution.

The linear models of system and a simulated evolution method have been proposed firstly by Lipson et al. [3], they used different linear matrices to represent the individuals/controllers, environments/resource and their goals. By randomly varying the elements of environment matrix, the modularity appeared in the individual's matrix. The relation between varying rate and modularity has also been studied by experiments. The authors claimed that the modularity arises in evolutionary system in response to variation.

According to previous suggestions [3], Kashtan and Alon further pursued the research on environment variation of evolution [4]. The simple feed-forward ANNs were used to perform the retina pattern classification tasks, which had limited connections and small range of weights. The general structural constraints for evolving the ANNs were also given. The results show that the modularity and motifs spontaneously evolved in networks when the goals switched in a modular manner with time during evolution. Their later work also suggested that varying environment could speed up the evolution at certain conditions [5].

To validate whether HyperNEAT could evolve modular neural networks, Clune et al. [6] investigated a series of retina recognition experiments, which were similar to those used by previous studies [4]. The results show that although the HyperNEAT has the potential ability to produce modular structures in some easy case, but unfortunately it was unsuccessful in more complex problems as in previous studies. In order to enable the HyperNEAT to generate modularity networks, Verbancsics and Stanley [8] presented a seeding method towards local connectivity, which successfully encouraged the natural emergence of modular structures and the evolution simulations were accelerated as well.

Instead of changing the environment, HØerstad [7] proposed a method that adding noise to the genotype-phenotype (G-P) mapping. The same retina recognition experiments have been used to test the noised based method. The ANNs and their encoding method were similar to those used by Kashtan and Alon [4]. Based on a large amount of simulation experiments, the author gave a statistical result. It shows that the novel method could trigger the appearance of modularity and finally speeded up the evolution, however, the switch-goal method does not show the same abilities, which totally against the conclusions of previous study [4].

As we can see here, none of those approaches has demonstrated a generality and repeatability; their results are usually in conflict with others even on the same sample

mission. Those may due to the reason that the above experiments are sensitive to the initial conditions, such as the encoding fashions, structural rules and the parameters of simulations. Therefore, a more universal approach with less limitation is needed to investigate the mechanism of modularity and their responses to the variation of environment and evolutionary process. In the following sections, we introduce a more freeform evolution of ANNs and a series of experiments under different conditions.

## III. NETWORKS, EVOLUTIONARY ALGORITHM AND MODULRITY MEASUREMENT

### A. The Pure Topological Neural Networks

To better understand the geometrical properties of complex networks, such as modularity and motifs, a pure topological ANN has been proposed in our previous work [14], which has binary connection weights and free form directed connections of hidden layer (Fig. 1).

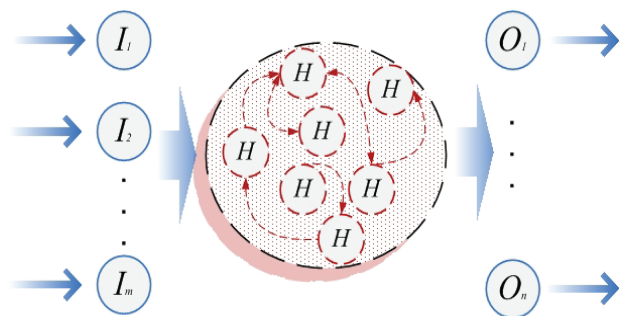


Fig. 1. Pure topological neural networks

As the architecture shows here, there are three groups of neurons: input neurons, hidden neurons and output neurons, represented as  $I$ ,  $H$  and  $O$  respectively. It should be noted that, unlike the traditional recurrent neural networks, there are no “layers” for all hidden neurons within this network. In order to get both positive and negative values, we use a sinusoidal function as an activation of all the hidden neurons; however, all output neurons are modulated by a sigmoid function. Therefore, all hidden neurons’ values and output neurons’ values are updated by equation (1) and (2) respectively. The model can be given as:

$$H_i(t+1) = \sin\left(\sum_j H_j(t)\right) \quad (1)$$

$$O_i(t+1) = \left(1 + \exp\left(-\sum_i H_i(t+1)\right)\right)^{-1} \quad (2)$$

where  $H_i$  denotes the current state value of the  $i$ th hidden neuron, which is relative to all other neurons ( $H_j$ ) who feed their outputs of previous state.  $O_i$  is the  $i$ th neuron’s value of all  $n$  output neurons. Due to the characters of activation functions, we need to normalize all input raw data into range of  $[0, 2\pi]$  before computation. Accordingly, we have to scale the output value from  $[0, 1]$  to the target range as a final step.

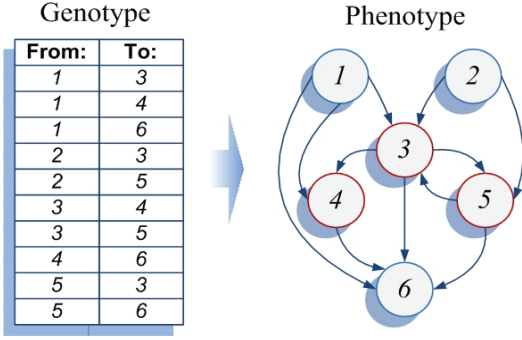


Fig.2. Encoding of neural networks

### B. Training the ANNs by Evolutionary Algorithm

Because of the simplicity, we prefer to use the graph encoding method, which directly encodes connections between two nodes in a “from-to” manner, and then organizes all those connections as a graph vector structure; such a vector could simply represent a neural networks topology (Fig. 2). In the genotype vector, there is no order for all pairs of “from-to” nodes, but in phenotype, especially for hidden neurons (labeled by red cycles), the neurons’ IDs indicate the sequence of signal flow; therefore the recurrences appear along with random connecting.

We designed five typical evolutionary operators to enable our networks to evolve toward an advanced level of fitness as natural lives.

**Elitist replication:** we begin with an elitist copy operation which ranks all candidates and then completely copies the best 25% of individuals to the next generation’s population.

**Roulette wheel selection:** after the elitist copy, based on fitness proportion, we use a classic “roulette wheel selection” method to replicate the promising network candidates from current population to generate the next population.

**Sub-graph crossover:** this operation randomly chooses a hidden node  $n$  of network vectors  $A$ , and  $B$ , and then executes a “one point” crossover operation between  $A$  and  $B$ , which makes  $A$  have all connections before node  $n$  from  $B$ , at the same time,  $B$  is replaced partly by  $A$ ’s connections after node  $n$ . We should note this operator could change the network topology enormously.

**Connection mutation:** a random connection from node  $n_1$  to  $n_2$  is first built by an operator, and then it is searched through the vector of networks, if the vector has one matched to the random connection, then deletes this connection from the vector, otherwise, inserts this random connection into the vector.

**Transposing Mutation:** a pair of nodes which randomly chosen within one network will be transposed by this operator. In consequence, the calculation sequence of nodes will be significantly changed, and then it further influences the network’s outputs.

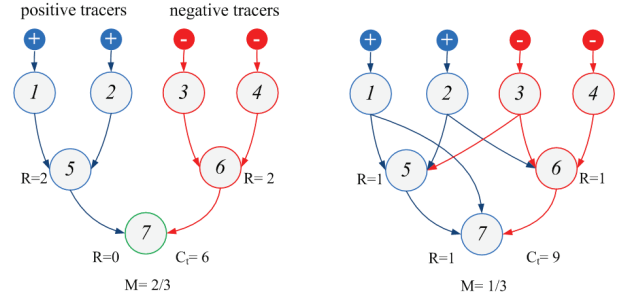


Fig.3. Artificial tracer method

### C. Modularity Measurement by Artificial Tracer

Although many quantitative algorithms for analyzing the network modularity have been proposed, we still desire a simpler and more specific way to measure the modularity of our ANNs. For this purpose, we propose an artificial tracer method inspired by the chemical tracer, isotopic tracer and radioactive tracer, etc. We create the digital tracer elements with different polarities, such as positive tracers and negative tracers and even other polarities. To measure the modularity, we first inject the different tracers into each input node of network according to their attributes. All tracers will then pass through other nodes along the directions of information flow. The output connection will pass a tracer to next node with the same polarity as the parent node. The annihilation will happen when two tracers with different polarities meet at a node of network. We then can roughly calculate the modularity by following equation:

$$M = \frac{\sum_{i=0}^n R_i}{C_i} \quad (3)$$

where  $M$  represents the quantity of network modularity, it has a value ranging from 0 to 1; the higher value means the network structure is more modular.  $R_i$  denotes the number of remaining tracers at the  $i$ th node after annihilations. We summarize their values as the equivalent of total amount of living connections. It should be noted that the  $R_i$  does not include all input nodes,  $i$  starts counting from the first hidden node.  $C_i$  is the total number of connections within this network. This calculation reveals the essential of modularity, which defined as a relation between the inter-connections and intra-connections of elemental modules. An illustration of this procedure is shown in Fig.3.

## IV. EXPERIMENT AND RESULTS

### A. The Retina Pattern Recognition Task

In this study, we investigate all issues using a classic retina pattern recognition test. The retina pattern recognition experiment has been frequently used in previous work as a challenging benchmark. In those studies, the ANNs have been evolved to recognize and classify an artificial retina. Each object retina consists of eight pixels (4-pixel-wide by 2-pixel-high), which is equally divided into left side and right side, four

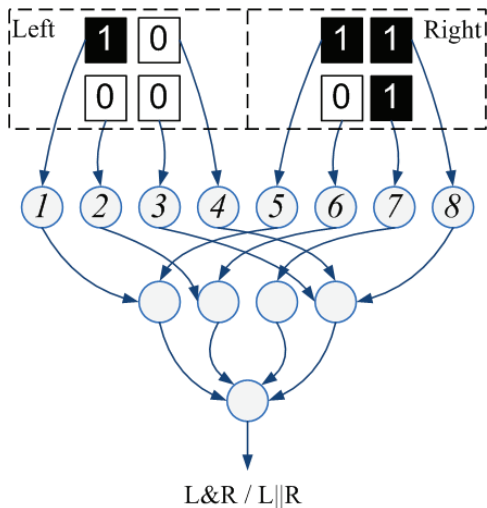


Fig. 4. Retina recognition mission

pixels per side. The goal is to use the ANNs to recognize objects in the left and right sides of this retina (Fig.4). As defined in [4], a left object is defined by three or more black pixels or one or two black pixels in the left column only. A right object is defined in a similar way, with one or two black pixels in the right column only. Those eight pixels each could be abstracted as 1 or 0, then those eight binary values could be treated as a group of input signals for the ANNs. Finally, the ANN has to use the single output value (0 or 1) to decide whether the retina fits the given Boolean logic questions “L AND R”, or “L OR R”. The “L AND R” is true only if the object exists at both sides of the retina, whereas if the object appears in left side or right side or even both sides, the “L OR R” function is then true.

### B. Experiments and Results

In most of our experiments, the data set used to train the ANNs consisted of 100 independent retina patterns which were randomly generated before training. The general fitness was designed to reflect the ratio of correct recognition over all 100 samples. To reduce the computational complexity, we also constrained the network size within 30 nodes in which 8 nodes were assigned for receiving pixels’ values, one node was as the final output, and the remaining nodes, up to 21, were free to build any structures through evolving towards a given task. As for the simulation, similar to previous work, we set the maximum generation as 10,000 and the population included 600 candidate networks. The modularity was estimated as well as the fitness, the best networks’ structures of each generation were recorded. We evolved the ANNs under a group of different regimes, and we run each test 10 times independently for various experimental scenarios, a list of experiments is shown in Table I.

We first evolved the ANNs to recognize the patterns of “L AND R” from predefined data set. Then, similar to previous work [4][6][7], we pursued an interesting MVG regime, in which the recognition goal switched between “L AND R” and “L OR R” every 50 generations. A varying environment regime (VE) was also tested. Instead of varying environmental

TABLE I. LIST OF EXPERIMENTS

Experiment	Description
FG-AND	Evolving networks to solve the fixed goal L AND R
MVG	The goal switched between “L AND R” and “L OR R” every 50 generations.
MVG-C70	Same as MVG, but with a threshold value of 70 for network connections
MVG-C100	Same as MVG, but with a threshold value of 100 for network connections
MVE	The data set changed between 100 samples and randomly selected 50 samples every 50 generations.
VS	The selection mechanism alternated between proportion-based roulette selection and random selection.
FG-M	Same as FG-AND, but the fitness function coupled with the value modularity

elements [3], we temporally changed the demission of data set (environment) as the variation of environment. Additionally, following the suggestions of [3], we designed the VS scenario as the variations of selection process, the proportion-based roulette selection mechanism sometimes got a failure during evolution, and then the random selection played a key role for producing offspring.

The comparisons of results on different regimes are shown in figs, all results shown here are the average values over 10 runs. Fig.5 shows the best networks’ fitness records on all regimes, as we can see, the MVE exhibit a significant higher fitness than others, and it approaches 0.95 within 8,500 generations, whereas the MVG does not show any advantages either in fitness value or evolution speed, its fitness value stays under 0.9 even worse than FG.

The connection number of networks can be seen in Fig.6. It is interesting that all regimes except MVG have a very close amount (85) of the networks’ links. Nevertheless, MVG regime evolved networks with less links than others; at the end of evolution only about 30 connections remained in networks.

Fig.7 presents the modularity estimation results for the best evolved networks of all regimes. The figure indicates a quiet disappointed result that a highly modular structure never arisen among all tests. For most of regimes, the modularity values keep under a low level of 0.35. Still, in MVG regime, we

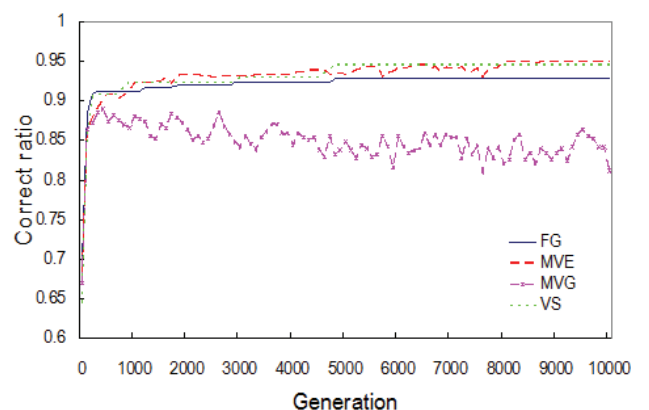


Fig. 5. Results of correct ratio

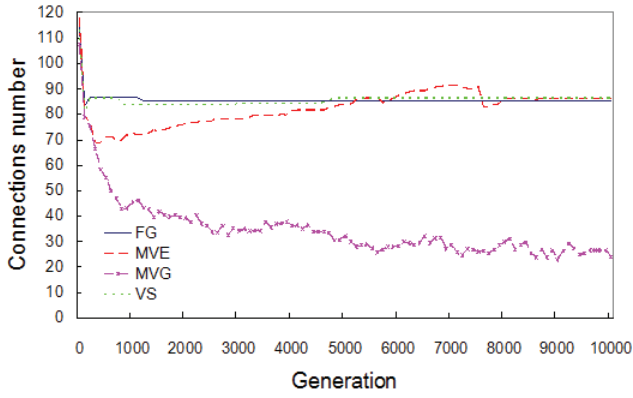
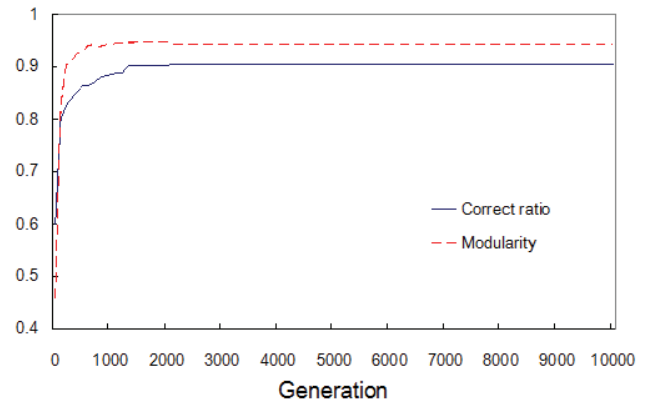


Fig.6. Connections number of different cases



(a)

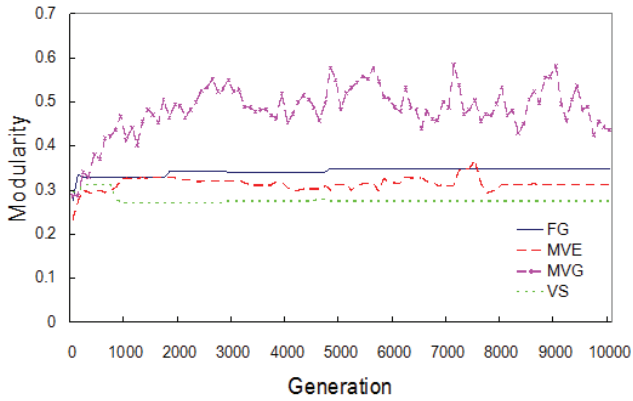


Fig.7. Modularity of different cases

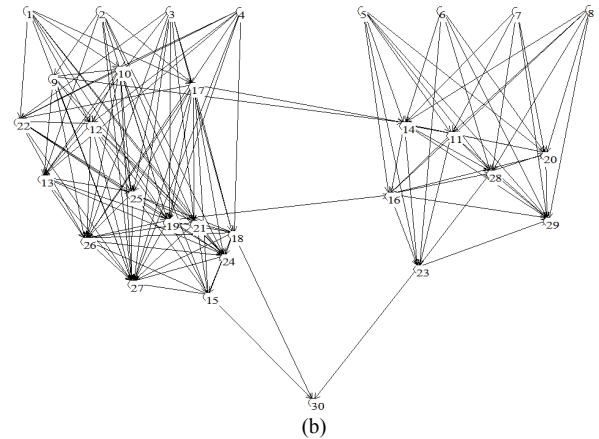


Fig.9. Evolution results: (a) correct ratio and modularity ;(b) the structure of the best network

observe a relatively higher value around 0.5.

In addition to free evolved MVG regime, we also tested two MVG regimes with the constraints of minimum connections, 70 for the MVG-C70 and 100 for the MVG-C100. In those tests, if the links numbers become less than the threshold values (70 and 100, respectively), the fitness value will be punished according to the shortages of threshold values. As shown in Fig.8, the modularity values of those new regimes are close to other regimes and lower than MVG.

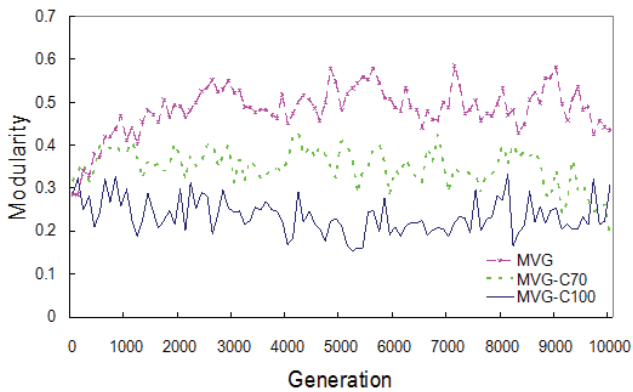


Fig.8. Modularity of different MVG cases

In order to demonstrate these ANNs can produce modular structure by the evolutionary method. We built a new fitness function for the evolution, which coupled the correct recognition ratio with the modularity through a simple multiplying relation. A new group of tests has been run under a similar regime to FG, called FG-M. The correction ratio and modularity are shown in Fig.9 (a). It could be easily found that the correction ratio and modularity, they both approach a relative high level at 0.93 and 0.95 respectively. We also visualize the structure of one evolved network in Fig.9 (b) by Mfinder [15]. As we can see, the network has obvious two separate modules, the inter links are much less than intra links, and thus the structure exhibits a high modularity.

## V. DISCUSSION

As used in previous works, the same retina recognition tasks were performed by ANNs under different conditions. It is found that compared to the fixed goal evolution the evolution processes were accelerated under most of variable scenarios such as MVE, and VS. This could probably attribute the dynamics of those schemes. In the MVE test, the solution space is reduced by portioning the object into a half dimension, and then the entire computational power is driven to solve the simpler partial question towards a same fitness function;

eventually, the local maxima will be overcome faster. The VS scheme seems to push the whole population to a random direction, then the diversity of population is maintained, thus it gives the opportunity to pull the solutions from local maxima.

In contrast, the MVG scheme exhibited an even worse performance than FG scheme, the similar result is also shown in [6][7]. The common questions are now arisen, why did the ANNs perform worse under MVG scheme? What is the hidden mechanism of MVG case? The connection number may explain those by comparing with other successful cases. In Fig.6, all other schemes show a similar trend on the connection number, which finally stands at the value about 85 throughout the evolution; however, for MVG scheme, it seems to fluctuate around the small value of 30. In fact, the amount of connections could reflect the complexity level of network, and the complexity is essential for achieving a complex mission. It is obviously that the networks evolved under MVG model lacked of complexity. The MVG scheme may have the side effect to simplify the networks rather than perfect them.

According to results shown in Fig.7, only MVG scheme has a slightly higher modularity value beyond others. We then guessed that this was probably also caused by the decrease of connections. The results shown in Fig.8 well validate this hypothesis, with the constraints of complexity, the modularity values of MVG cases return to a level closed to others.

It was very doubted if the proposed ANNs could not produce modular structures by evolutionary method. Fortunately, the results of FG-M indicate that the correct ratio based fitness has no confliction with modularity, since the evolved network (Fig.9 (b)) presents a nearly perfect modular structure with a high correct ratio (Fig.9 (a)). However, we have to admit that none of varying schemes had successfully evolved a modular network. This failure can be attributed to two possible reasons. First, we have just run limited tests for each scheme; the better results might arise within a large amount of runs as shown in [7]. Second, since we are challenging a relatively harder mission than others, the current simple methods, such as varying environment, etc. are not powerful enough for triggering the emergence of modularity in our case. Therefore, the more effective methods, for instance the seeding method [8] are desired in next stage of our work.

## VI. CONCLUSION AND FUTURE WORK

Previous works suggest that the ANN's modularity could be encouraged by varying the environment or the evolution process. To validate such viewpoint, we proposed an ANN accompanying with a freeform evolutionary training method to perform a well studied retina pattern recognition task. An artificial tracer method is proposed to estimate the modularity of network.

Many different experimental schemes have been run and the modularity and other features are analyzed in this paper. The results show that most of the varying schemes have the

positive effects on the fitness and evolving speed of ANNs, however the MVG (Modular varying goal) test brings a negative influence on the correct ratio of recognition, and it also has the danger to break the complexity of networks. Nevertheless, the modular structure never appeared among all tests, this exposes that the current variation-based methods lack of generality on more complex scenarios. Finally, a modified fitness function the modularity bias was used to evolve the network; the evolved modular networks raise the hope of autonomously emerging the modularity through more natural ways.

In future, more experiments have to be run and the evolution features of ANNs will be given based on a statistical result. A motifs-seeding method will also be implemented to further encourage an appearance of modularity.

## REFERENCES

- [1] M. E. J. Newman, "Modularity and community structure in networks," *Proceeding of National Academy of Sciences*, vol. 103, no. 23, pp. 8577-8582, 2006
- [2] G. P. Wagner, and L. Altenberg, "Complex adaptations and the evolution of evolvability". *Evolution*. vol.50, pp.967-976, 1996
- [3] H. Lipson, J.B. Pollack, and N.P. Suh, "On the Origin of Modular Variation," *Evolution*, vol.56, no.8, pp.1549-1556,2002
- [4] N. Kashtan, and U. Alon, "Spontaneous Evolution of Modularity and Network Motifs," *Proceeding of National Academy of Sciences*, vol.102, no.39, pp.13773-13778, 2005
- [5] N. Kashtan, E. Noor, and U. Alon, "Varying environments can speed up evolution," *Proceeding of National Academy of Sciences*, vol.104, no.34, pp.13711-13716, 2007
- [6] J. Clune, B. E. Beckmann, P. K. McKinley, and C. Ofria, "Investigating whether hyperneat produces modular neural networks," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2010, pp.635-642
- [7] B. A. Höverstad, "Noise and the Evolution of Neural Network Modularity," *Artificial Life*, vol.17, no.1,18, 2011
- [8] P. Verbancsics and K. O. Stanley, "Constraining Connectivity to Encourage Modularity in HyperNEAT," Department of Electrical Engineering and Computer Science, University of Central Florida, UCF Dept. of EECS Technical Report CS-TR-10-10, 2010.
- [9] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR Upper Saddle River, NJ,1994
- [10] X. Yao, "Evolving artificial neural networks", *Proceedings of the IEEE*, vol. 87, pp. 1423-1447, 1999
- [11] D. Floreano, P. Dürri, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47-62, 2008
- [12] D.E. Moriarty, and R. Miikkulainen, "Forming Neural Networks Through Efficient and Adaptive Coevolution," *Evolutionary Computation*, vol. 5, no. 4, pp. 373-399,1997
- [13] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 54-65, 1993
- [14] S. Li, J.Yuan, X. Yue and J. Luo, "The binary-weights neural network for robot control," in *Proceedings of the International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2010, pp. 765-770
- [15] N. Kashtan et al, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics*, vol. 20, no. 11, pp. 1746-1758, 2004