

Abstract

Neuroevolution algorithms are usually used to optimize one Artificial Neural Network. Composite Neuroevolution optimizes multiple Artificial Neural Networks which are integrated into a complete structure in a pre-defined way which called compose method. Composite Neuroevolution contributes to achieve emergence behavior. This paper compares three types of structures on the cat landing problem. Their advantages and disadvantages are discussed.

Keywords: Cat landing problem, Posture control, Composed behavior, Composite Neuroevolution

1 Introduction

Neuroevolution algorithms are usually used to optimize one Artificial Neural Network(ANN). In composed behavior methodology, multiple ANNs are optimized one after another and integrated to a complete structure. This method is done with human interference which guarantees that each ANN has explicit function. This method helps to solve a given task where the optimization falls into the local minimal easily[1]. However, the human interference makes this method inconvenient.

Composite Neuroevolution optimizes multiple ANNs simultaneously without human interference. It has the same effect as composed behavior methodology, so it contributes to achieve emergence behavior. This paper compares three types of compose methods, also referred as structures in this paper, on cat landing problem. Their advantages and disadvantages are discussed.

2 Cat Landing Problem

The cat has ability to rotate itself to face the ground in the air when landing. We construct a virtual cat robot by Physics Modeling, and optimize its controller to perform this motion. All simulations are done in PhysX.

The cat is composed of a sphere and seven cuboids which represent the head, the tail, four legs, the front part of the body and the back part of the body which are respectively connected by seven joints. Actuators are placed on joints at the spine and legs. Their moving range is ± 60 degree on two axes. Sensors are the direction and angular velocity of bodies and legs of cat. The controller outputs angular velocity of two axes to each actuator. Fig.1 shows our model.

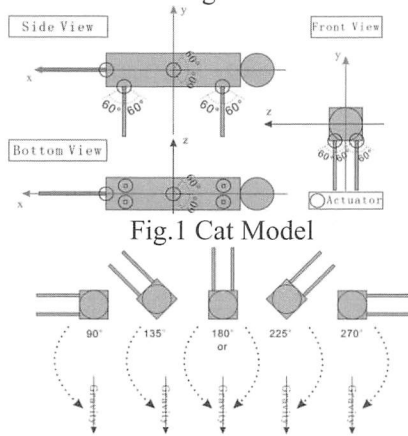


Fig.1 Cat Model



Fig.2 Sub-tasks

The task includes 5 different sub-tasks with different initial states: the cat is rotated 90, 135, 180, 225, 270

degree. Fig.2 shows sub-tasks and their desired motions. The 180° sub-task has two acceptable motions. Each sub-task simulates for 100 steps. The cat is placed at 3m height, and the gravity is set to 0. The fitness function for 'i' sub-task is shown by Eq.1.

$$f_i = 2 - \min(MD_i(k))$$

$$MD_i(k) = \max(D_i(\text{front}, k), D_i(\text{back}, k)) \quad \text{Eq.1}$$

$D_i(\text{part}, k)$ is the distance of y-direction between denoted part and gravity in step k. The total fitness is $F = \min(f_i)$. The optimization falls into local minimal easily because it is much easier to find a motion, which only has the ability to rotate the cat from one side no matter what the initial state is, than the desired motion.

3 Composite Neuroevolution

Composite Neuroevolution optimizes a structure which usually integrated of multiple ANNs. ANNs are coded one by one in a linear gene and optimized by a Neuroevolution algorithm simultaneously. The ANNs are integrated to a structure in a pre-defined way. To emphasize its advantage, the selection and gene operator are same as CNE[2]. However, it is possible to use other Neuroevolution algorithm like CMA-ES[3], NEAT[4] or ESP [5]. There are lots of possible structures. This paper compares three types of structures: compose structure, compete structure and cluster structure. All experiments are discussed based on cat landing problem.

3.1 Compose Structure

The compose structure is shown in Fig.3. It has two kinds of ANNs which have the same input, but different output. Output of working ANN is the signal for actuators. And output of compose ANN is a priority vector: each priority value is related to a working ANN. The working ANN with the highest priority value is selected to transmit its output as the output of structure. The output of structure is $O = w_{i \in \{0,1\} \arg \max(o_i)}$.

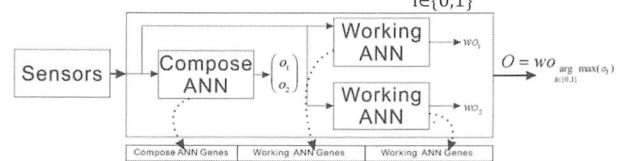


Fig.3 Compose Structure

3.2 Compete Structure

The compete structure is shown in Fig.4. Each ANN has an additional output. The ANN with the biggest additional output value is selected to transmit its output as the output of structure. So the output of structure is $O = w_{i \in \{0,1\} \arg \max(o_i)}$.

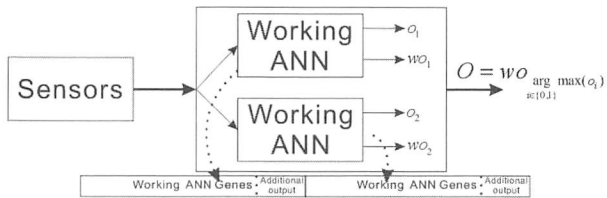


Fig.5 Compete Structure

3.3 Cluster Structure

The cluster structure is shown in Fig.5. In the chromosome, a vector is coded with each ANN. Vectors have same dimension as input of ANN. The distance between vector and input is calculated in each step. The ANN with the minimal distance value is selected to transmit its output as the output of structure. So the output of structure is $O = w_{\text{arg max}(d_i)}$.

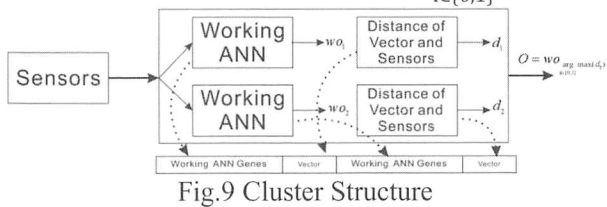


Fig.9 Cluster Structure

4 Experiments

We run CNE, NEAT, ESP, CMA-ES to optimize one ANN as the controller of cat, and run Composite Neuroevolution of three structures. The number of population is 100 for CNE, NEAT, 300 for ESP, 10 for CMA-ES, 50 for three Composite Neuroevolution methods. Their fitness curves are shown in Fig. 6, 7.

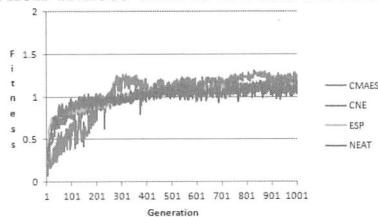


Fig.6 Fitness of CNE,NEAT,ESP,CMA-ES

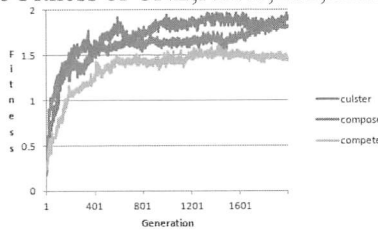


Fig.7 Fitness of three structures

CNE, ESP, NEAT, CMA-ES reach fitness around 1.2. So they fall into local minimal in this task. The compose and cluster structure reach a fitness of around 2. The compete structure reaches a fitness around 1.5, so it also falls into local minimal but better than CNE, etc. All three structures overcome the local minimal problem in varying degrees without human interference. So they contribute to achieve emergence behavior.

The selected ANN of structures in each simulation step shows the information distribution among ANNs, as shown in Fig 8-10. The x-axis is the simulation step and the y-axis is the selected ANN in denoted step. The compose and compete structures change selected ANN

many times in each sub-task, so they tend to store the related information among two working ANNs. The cluster structure basically uses only one ANN in each sub-task, so it stores related information in the same ANN. So it is possible to distinguish which ANN has which function.

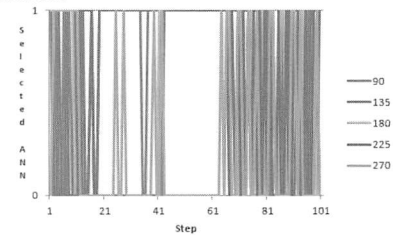


Fig.8 Selected ANN of compose structure

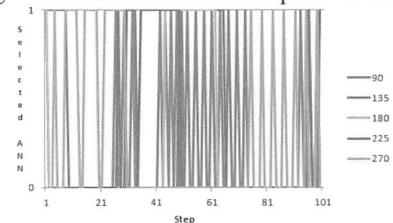


Fig.9 Selected ANN of compete structure

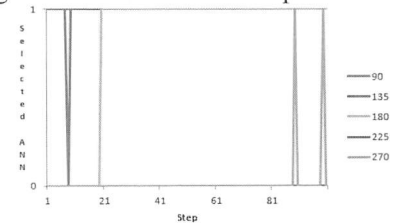


Fig.10 Selected ANN of cluster structure

5 Conclusions

This paper proposes the concept of Composite Neuroevolution and compares three structures: the compose, compete and cluster structure. All three structures contribute to achieve emergence behavior. Compose and cluster structures show similar performance in our experiment. But compete structure is worse than two others. The cluster structure can store related information in the same ANN. This feature is helpful to identify the function of each ANN.

Reference

- [1] Keisuke YONEDA, Ikuo SUZUKI, Masahito YAMAMOTO, Masashi FURUKAWA, Acquisition of Locomotion for Modular Robot With Annular Shape Using Evolutionary Computation, Journal of the Japan Society for Precision Engineering, 76(12): 1351-1357(2008).
- [2] D. Floreano, P. Durr, and C. Mattiussi, Neuroevolution: From architectures to learning, Evolutionary Intelligence, 1:47-62(2008).
- [3] Hansen N, Ostermeier. A Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation, 9(2):159-195(2001).
- [4] Stanley K, Miikkulainen R. Evolving neural networks through augmenting topologies. Evolutionary Computation, 10(2):99-127(2002).
- [5] Gomez F, Miikkulainen R. Incremental evolution of complex general behavior. Adaptive Behavior. 5(3-4):317-342(1997).