NeuroEvolution of Augmenting Topologies for Interpretable Control Policy

Lishuang Wang
wanglishuang22@gmail.com
Southern University of Science and Technology
Shenzhen, Guangdong, China

Mengfei Zhao
12211819@mail.sustech.edu.cn
Southern University of Science and Technology
Shenzhen, Guangdong, China

Ran Cheng*
ranchengcn@gmail.com
Southern University of Science and Technology
Shenzhen, Guangdong, China

ABSTRACT

We used the NeuroEvolution of Augmenting Topologies (NEAT) algorithm in the Gymnasium's Walker2d environment to develop action strategies that balance score and interpretability. Our final strategy is a network with 17 connections and 23 weights. We expressed the forward inference process of the network using mathematical formulas, and it achieving a score of 3300 in sufficient evaluations.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**.

KEYWORDS

Neuroevolution, Interpretable Policy, Continuous Control

ACM Reference Format:
Lishuang Wang, Mengfei Zhao, and Ran Cheng. 2024. NeuroEvolution of Augmenting Topologies for Interpretable Control Policy. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Control systems play a crucial role in the management and regulation of various processes and devices across a wide range of application domains. Their ubiquity makes them a cornerstone of modern technology. Particularly for safety-critical applications, control systems not only need to be efficient but also interpretable to ensure trustworthiness, reliability, and accountability.

We believe that the interpretability of a policy is directly related to its complexity. To find policies with low complexity, we used the NeuroEvolution of Augmenting Topologies (NEAT) algorithm, which evolves both the topology and weights of the network simultaneously. We conducted experiments in the Walker2d environment from Gymnasium and ultimately obtained a neural network with 17 edges. This network achieved an average score of 3300+ over sufficient evaluations with only 23 weights. Due to the small size of the network, we can easily convert its forward inference process into mathematical formulas.

The results and code are available at: https://github.com/WLS2002/interpretable-control-competition-walker-submit.

*Corresponding Author

Conference'17, July 2017, Washington, DC, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

2.1 NEAT

The NeuroEvolution of Augmenting Topologies (NEAT) algorithm [1] is an evolutionary method used for developing artificial neural networks. It can evolve both the weights and the topological structure of the network simultaneously. The NEAT algorithm introduces a historical marking mechanism for network nodes, successfully solving the problem of crossover between different networks. Additionally, NEAT maintains population diversity through speciation, protecting promising network structures and promoting competition between different network structures.

2.2 Walker2d

Walker2d environment is a reinforcement learning task in Gymnasium [2], where a bipedal robot must learn to walk forward. It challenges the agent to balance, coordinate leg movements, and maximize its forward speed, testing control and stability in a simulated 2D environment.

3 EXPERIMENT & RESULTS

Our experiment are running on a server with Intel(R) Xeon(R) Platinum 8463B with 192 cores. We choose TensorNEAT library [3] as it enable hardware acceleration. Some important hyperparameters are shown in table 1, and experiment settings can be check at our source code. Our algorithm used totally $500 \times 400 = 200,000$ environment evaluations.

Parameter	Value
Number of generations	500
Population size 4	500
Evaluation repeat times	1
Node add prob	0
Node delete prob	0
Conn add prob	0
Conn delete prob	0.8-0.1

Table 1: Important hyperparamter settings

$$
\begin{aligned}
h_0 &= \tanh\left(0.93i_0 + 0.48i_1 - 0.24i_{10} - 0.3i_{12} + 0.01i_{13} + 0.22i_{14} - 0.18i_{15} + 0.08i_2 - 0.15i_5 + 0.83i_7 - 0.05i_8\right)\\
h_1 &= \tanh\left(0.3i_0 + 1.1i_1 - 0.21i_{10} - 0.43i_{12} + 0.37i_{13} + 0.54i_{14} + 0.51i_{15} - 0.17i_{16} - 0.12i_2 - 0.26i_4 + 0.82i_5 + 0.98i_7 + 0.02\right)\\
h_2 &= \text{clip}(0.61i_0 - 0.16i_{12} - 0.35i_{13} - 0.32i_{14} + 0.3i_{16} - 0.4i_5 + 1.13i_7 - 0.27i_9 + 0.08, -1, 1)\\
h_3 &= -\tanh\left(0.72i_0 + 0.25i_{12} - 0.45i_{16} + 0.18i_2 - 0.04i_6 - 0.11i_9 + 0.02\right)\\
h_4 &= -\tanh\left(0.39i_0 - 0.87i_{10} - 0.31i_{11} + 0.43i_{14} + 0.94i_{16} + 0.61i_3 - 0.15i_5 + 0.09i_6 - 0.11i_7 + 0.91i_9 + 0.17\right)\\
h_5 &= 0.39i_{10} - 0.04i_{14} + 0.15i_{15} - 0.02i_{16} + 0.47i_5 + 0.21i_8 - 0.13\\
o_0 &= \tanh\left(1.24h_1 + 1.25h_2 - 1.12h_3 + 0.9h_4 + 1.35\right)\\
o_1 &= \tanh\left(1.63h_0 + 0.49h_1 + 1.6\right)\\
o_2 &= \tanh\left(-0.41h_1 + 1.58h_2 - 0.16h_3 + 0.8h_4 + 1.31\right)\\
o_3 &= -\tanh\left(1.18h_0 + 2.41h_1 + 1.62h_2 - 0.9h_3 + 1.0h_4 + 2.39h_5 + 0.62\right)\\
o_4 &= \tanh\left(2.04h_0 - 0.88h_3 + 0.43\right)\\
o_5 &= \tanh\left(0.38h_0 + 1.54h_1 - 0.86h_4 + 0.14h_5 + 0.78\right)
\end{aligned}
$$

Figure 1: The mathematical expression for the forward inference process of the final network. The weights are displayed with two decimal places.

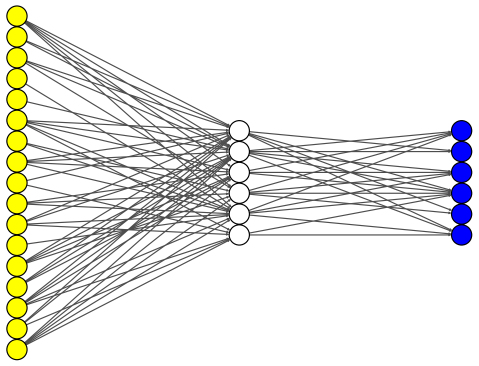

Figure 2: The structure of the final network. Yellow nodes represent input nodes, and blue nodes represent output nodes.

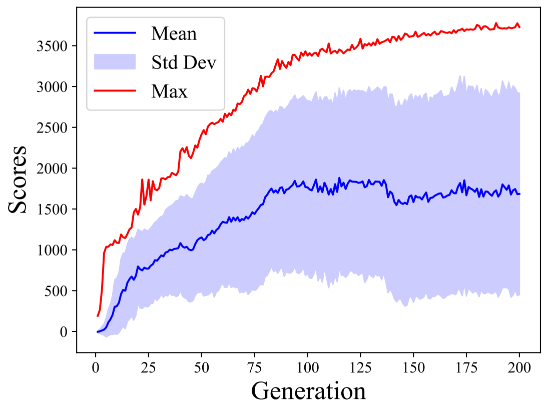

Figure 3: The scores for each generation during the evolving process.

To ensure the stability for the evolving process and the interpretability of the final solution, we improve the NEAT algorithm in the following ways:

(1) **Observation Normalization**. Before the evolution starts, we sample the Walker2d environment using a random policy to obtain the mean and variance of the environment observations. During the evolution process, the observations received by each network undergo z-score normalization.
(2) **Connection Pruning**. Unlike the conventional NEAT algorithm, where the network topology gradually grows, we perform connection pruning during the evolution process. At the start of the evolution, we use a two-layer fully connected network, with each layer having the same number of nodes as the input and output of the Walker2d environment, i.e., 17 and 6. The network initially has $17 \times 6 = 102$ connections. During each generation's mutation process, individuals have a probability of losing a connection. The probability of losing a connection is related to the number of existing connections.
(3) **Elite Pool**. Instead of using the traditional elitism mechanism of the genetic algorithm, we use an Elite Pool. Individuals in the Elite Pool are evaluated multiple times to reduce the variance caused by the environment.
(4) **Fitness Shaping**. In the Elite Pool, individuals that achieve a certain score in the Walker2d environment receive additional fitness rewards. The fewer connections an individual has, the higher the additional reward.

Our algorithm has achieved remarkable results. Figure **??** shows the highest score and average score of each generation's population. The shaded area indicates the range of scores within the population. After 400 generations of evolution, the best individual has only 17 edges and a total of 23 weights. Figure **??** illustrates the network structure of the best individual. Due to the small scale of the network, we can easily convert its forward inference process into mathematical formulas, as shown in Figure **??**. We used this formula as the control strategy for the Walker2d environment, achieving an average score of 3300 over 1200 evaluations with different random seeds.

4 CONCLUSION

We believe that the interpretability of the strategy is related to its complexity. Using the NEAT algorithm, we ultimately evolved networks with low complexity, which also achieved good scores in the Walker2d environment.

REFERENCES

[1] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10, 2 (2002), 99–127.

[2] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. 2023. Gymnasium. https://doi.org/10.5281/zenodo.8127026

[3] Lishuang Wang, Mengfei Zhao, Enyu Liu, Kebin Sun, and Ran Cheng. 2024. Tensorized NeuroEvolution of Augmenting Topologies for GPU Acceleration. (2024).