A Gene Regulatory Model for the Development of Primitive Nervous Systems

Yaochu Jin, Lisa Schramm, Bernhard Sendhoff

2008

Submitted to ICONIP'08

A Gene Regulatory Model for the Development of Primitive Nervous Systems

Yaochu Jin[1] and Lisa Schramm[2] and Bernhard Sendhoff[1]

[1] Honda Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach, Germany
[2] Technische Universität Darmstadt, Karolinenplatz 5, 64289 Darmstadt, Germany

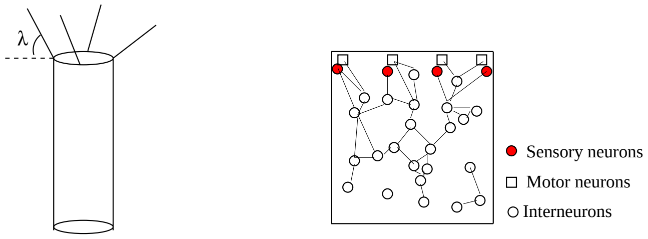
Abstract. This paper presents a model for the development of primitive nervous systems in a hydra-like animat controlled by a gene regulatory network. The gene regulatory network consists of structural genes that simulate the main cellular events during neural development at an abstract level, namely, cell division, cell migration, and axon growth, and regulatory genes that control the expression of the structural genes. The developmental model is evolved with an evolutionary algorithm to achieve the correct developmental order. After the genetically controlled neural development is completed, the connectivity and weights of the neural networks are further adapted to perform simple behaviors such as food catching of a hydra. Our preliminary results suggest that the proposed developmental model is promising for computational simulation of the evolution of neural development for understanding neural organization in biological organisms.

1 Introduction

Understanding the evolution of biological neural organization using computational models has attracted increasing attention. Two issues are considered essential for this body of research to be biologically plausible. First, the evolution of neural organization should be coupled with that of the body plan [1,4,7]. Second, the influence of neural development on the evolution of the nervous system should also be taken into account [9].

A few computational models for neural development have been reported. Cangelosi et al. [2] suggested a developmental model for cell division and migration that uses a rule-rewriting grammar. A model for neurogenesis that includes metabolic reactions and diffusion was proposed by Kitano [8], though no functionality of the developed neural network has been considered in the paper. A recurrent artificial neural network was used for modeling the development of a spiking neural network for the control of a Khepera robot [5].

This work suggests a computational model for neural development based on a gene regulatory network (GRN) for morphological development [13]. The gene regulatory network is composed of a number of structural genes whose expression is controlled by regulatory genes. The gene regulatory model is evolved to achieve the correct developmental order, i.e., first cell division, then cell migration, and finally axon growth [10]. The neurons are distributed over the surface



(a) Body−plan (b) Nervous system on the unfolded body plan

Fig. 1. The hydra-like animat. (a) The body plan with four tentacles, and (b) the nervous system shown on the 2-D area representing the unfolded body surface.

of a cylinder that simulates the body of hydra, which are phylogenetically the first to have a nervous system. Neurons are then connected according to a probability inversely proportional to the distance between them. In order to perform a food-catching behavior, the connectivity and weights of developed neural system are further adapted. Part of this work has been reported in [11].

Section 2 describes the body plan of the hydra-like animat, the gene regulatory model for neural development, as well as the neural network dynamics based on the integrate and fire spiking neural model. The experimental setup and simulation results for evolving the developmental order are presented in Section 3. The developed neural system is further adapted to catch as many pieces of food as possible while minimizing energy consumption, which is presented in Section 4. The paper concludes with discussions in Section 5.

2 The Model

2.1 A Hydra-like Animat

The physical body of the hydra-like animat simulates very loosely that of a hydra, which consists of a cylinder with four tentacles equally distributed on its top. Each tentacle is driven by a muscle cell controlled by a neural network composed of motor neurons, interneurons and sensory neurons, all of which are modeled with integrate and fire (IAF) neurons, refer to Fig. 1. To simplify the experimental setup, four sensory neurons and four motor neurons are put on the top of the body, while the number and position of the interneurons are determined by the developmental process. During the behavior adaptation, four pieces of food drop sequentially around the body and the closest sensory neuron will generate a number of spikes as the input to the neural network.

2.2 Genome for Neural Development

The genome in our model consists of structural units (SUs) representing genes that can result in certain cellular behaviors, such as cell division, cell migration, axon growth, and cell death, or can produce transcription factors (TFs) for intra-

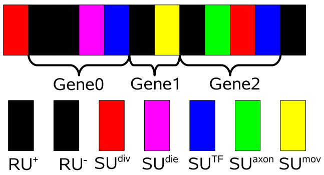


Fig. 2. The example of the genome for neural development.

and inter-cellular interactions. The SUs are preceded by a number of regulatory units (RUs), refer to Fig. 2. Note that multiple RUs may regulate the expression of a single or multiple SUs and that RUs can be activating (RU^+) or repressive (RU^-). Three different cell types are simulated in the model: stem cells, interstitial cells (after division) and neurons (after axon growth), depending on the activation status of the structural units of the cell.

Each RU and TF has a certain affinity value that determines whether a TF can regulate a RU. If the difference between the affinity values of a TF and a RU is smaller than a predefined threshold ϵ (in this work ϵ is set to 0.2), the TF can be bound to the RU to regulate. The affinity overlap ($\gamma_{i,j}$) between the i-th TF and j-th RU is defined by:

$$\gamma_{i,j} = \max\left(\epsilon - \left|\mathrm{aff}_i^{\mathrm{T}F} - \mathrm{aff}_j^{\mathrm{R}U}\right|, 0\right). \tag{1}$$

If $\gamma_{i,j}$ is greater than zero, then the concentration c_i of the i-th TF is checked to see if it is above a threshold ϑ_j defined in the j-th RU:

$$b_{i,j} = \begin{cases} \max\left(c_i - \vartheta_j, 0\right) \text{ if } \gamma_{i,j} > 0 \\ 0 \qquad\qquad\qquad\quad \text{else} \end{cases} \tag{2}$$

Thus, the activation level contributed by the j-th RU (denoted by $a_j, j = 1, ..., N$) can be calculated as follows:

$$a_j = \sum_{i=1}^{M} b_{i,j}, \tag{3}$$

where M is the number of TFs that regulate the j-th RU. Assume the k-th structural gene is regulated by N RUs, the expression level of the gene can be defined by

$$\alpha_k = 100 \sum_{j=1}^{N} h_j a_j (2s_j - 1), \tag{4}$$

where $s_j \in (0, 1)$ denotes the strength and sign (positive for activating and negative for repressive) of the j-th RU, which is subject to evolution, and h_j is an additional parameter representing the regulatory strength of j-th TF. If $\alpha_k > 0$, then the k-th gene is activated and its corresponding behaviors are performed.

A SU that produces a TF encodes all parameters related to the TF, such as the affinity value, the decay rate, the diffusion rate, as well as the amount of TF to be produced:

$$A = \beta \frac{2}{1 + e^{-20 \cdot f \cdot \alpha}} - 1, \tag{5}$$

where f is a parameter also encoded in the SU, and β is a parameter for expression rate of the SU.

A TF produced by a SU can be internal (for inter-cellular interactions) and external (intra-cellular interactions). Internal TFs stay inside a cell and do not diffuse. External TFs can diffuse into the whole simulation area, a process that depends on the diffusion parameter of the TF. To determine how much of a produced TF is external, a percentage ($p^{\text{ex}} \in (0, 1)$) is also encoded in the corresponding gene. Thus, $p^{\text{ex}} \cdot A$ is the amount of external TF and $(1 - p^{\text{ex}}) \cdot A$ is that of the internal TF.

External TFs are put on four grid points around the center of the cell, which undergoes first a diffusion and then decay process:

$$\begin{aligned} \text{Diffusion: } & \mathbf{u}_i(t) = \mathbf{u}_i(t-1) + 0.1 \cdot D_i^f \cdot (\mathbf{G} \cdot \mathbf{u}_i(t-1)), & (6) \\ \text{Decay: } & \mathbf{u}_i(t) = \min\left((1 - 0.1 \cdot D_i^c)\, \mathbf{u}_i(t),\, 1\right), & (7) \end{aligned}$$

where $\mathbf{u}_i$ is a vector of the concentrations of the i-th TF at all grid points, matrix $\mathbf{G}$ defines which grid points are adjoining, D_i^f is the diffusion parameter, and D_i^c is the decay rate.

There are also a few parameters defined by the SUs that generate cellular behaviors. The SU for cell division specifies the angle of division, indicating whether the daughter cell is placed above or below the mother cell. For cell migration, two parameters are encoded, one for direction (moving up or down) and the other for moving velocity. The SU for axon growth encodes the expected lifetime (t_{ttd}) of a neuron and three parameters (c_1, c_2, c_3) determining the probability threshold for axon growth. The limited lifetime of the neuron, t_{life}, is defined by the following Gaussian distribution:

$$t_{\text{life}} \sim \mathcal{N}(20 t_{ttd},\, 4). \tag{8}$$

The threshold for whether the i-th neuron is to be connected to the j-th neuron is calculated as follows:

$$\varphi_{ij} = \frac{c_1}{1 + e^{c_2 \cdot (d_{ij} - 10 c_3)}}, \tag{9}$$

where d_{ij} is the distance between the i-th and j-th neurons. The distance is computed in the 2D region (refer to Fig. 1(b)), which is the distance between the two neurons along the surface of the cylindrical body. Then, a random number p ($p \sim \mathcal{N}(0, 1)$) is generated, and if $p < \varphi_{ij}$, a connection between the two neurons will be generated. The connectivity between the interneurons and the sensory neurons, as well as between the interneurons and motor neurons is determined in the same way. Note, however, that there is no direct sensory-sensory, motor-motor, and sensory-motor connection.

2.3 Neural Network Dynamics

The dynamics of the neural network is modeled by IAF spiking neurons [3]:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{\text{resting}}) + R_m I(t), \tag{10}$$

where τ_m is the membrane time constant, R_m is the membrane resistance, $I(t)$ is the external current, $V(t)$ is the membrane potential, V_{resting} is the resting potential. When the membrane potential $V(t)$ is larger than a given threshold V_{th}, a spike is generated. The spiking event is characterized by its firing time t^f. After spiking, the membrane potential is reset to V_{reset} for a refractory period τ_{ref}. In an IAF neural network, the membrane potential of a neuron (V_k) can be calculated by:

$$V_k(t) = \sum_{i=1}^{H} w_{ik} y_i(t), \tag{11}$$

where w_{ik} is the weight between neurons k and i, N is the total number of its presynaptic neurons, and $y_i(t)$ is the unweighted contribution of the i-th presynaptic neuron:

$$y_i(t) = \varepsilon(t - t_i^f - d_{ik}), \tag{12}$$

where ε is a spike response function modeling the post-synaptic potential, t_i^f is the firing time of neuron i, and d_{ik} is the synaptic delay.

The spiking neural network in this work is simulated using the NEural Simulation Technology (NEST) [6].

3 Evolving the Developmental Order

An evolutionary algorithm is applied to evolve the genome to achieve the correct developmental order, i.e., the genes for cell division, cell migration and axon growth should be activated sequentially during the development. In addition, the gene for division should be deactivated before the gene for migration is activated, as the gene for migration should be deactivated before the gene for axon growth is activated. Gene transposition, gene duplication, and mutations are employed as the genetic variations. If gene transposition is performed, two randomly chosen units (both SUs and RUs are possible) are marked, and all units between these two marked units are cut out and pasted at another randomly chosen position. A gene duplication differs with gene transposition in that all units between the markers are copied and pasted at another randomly chosen position. All real-valued parameters (scaled between [0,1]) in the genome are adapted using an evolution strategy (ES) [12]. A repairing operator is introduced so that a single gene contains only one type of SU related to cellular behaviors.

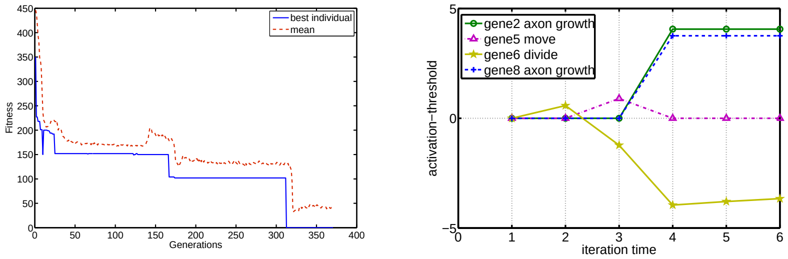
The following fitness function is used to penalize wrong developmental orders:

- If two cellular behaviors are active simultaneously, a penalty of 2 is applied.
- If three cellular behaviors are active simultaneously, a penalty of 5 is applied.
- If axon growth is followed by cell division, 2 is added.

- If axon growth is followed by cell movement, 1 is added.
- If cell migration is followed by cell division, 1 is added.

The target of the evolution is to minimize the fitness function. In the simulations, the parent and offspring population sizes are set to 1000 and 3000, respectively. The probability for gene transposition and duplication is set to 0.05, and the mutation step-size is initialized to $1e^{-4}$, with its lower bound being set to $1e^{-6}$ during evolution. The development begins with a single stem cell put in the center of the simulation area.

The fitness profile is shown in Fig. 3(a). We can see that the correct developmental order is achieved after over 300 generations. The activation levels of the gene for cellular behaviors (refer to Eq. 4) are shown in Fig. 3(b). A value above zero indicates that corresponding gene is activated and repressed if it drops below or equal to zero.



(a) (b)

Fig. 3. (a) Fitness profile. (b) Activation levels of the genes.

4 Evolution of the Neural Network for Food-Catching

Twelve stem cells with the genome that performs the cellular behaviors in the correct order are distributed on the simulation area so that a neural network is developed. For the neural network to perform the food-catching behavior, we employ an ES to adapt its connectivity and weights. In this work, we assume that only one tentacle is needed to catch a piece of food. Thus, the target of the behavior adaptation is that the motor neuron closest to the dropping food should fire as strong as possible to maximize the possibility to catch the food, while the activity of other motor neurons should be minimal to reduce energy consumption.

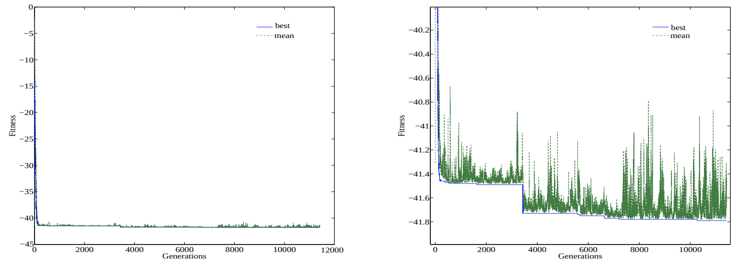
To achieve the above-mentioned target, the following fitness function is defined for the i-th motor neuron (output neuron):

$$F_i = -f_i + \sum_{j!=i} ns_j, \text{for } i, j = 1, ..., 4, \tag{13}$$

where ns_j is the number of spikes generated by other motor neurons in the simulated time period, penalizing unnecessary energy loss, f_i reflects the performance of the i-th neuron measured by the time (t_i^c) for the corresponding tentacle to stand vertically, i.e., when the angle $\lambda = 90$, refer to Fig. 1, or by the maximum λ during in the simulation period, if λ never reaches 90 degree during that time. The angle of the i-th tentacle is calculated as follows:

$$\lambda_i(t) = \lambda_i(t - 0.1) - g * 0.1 + s_i(t),\ \lambda_i \in [0, 90]. \tag{14}$$

The second term on the right side of Eq. 14 simulates the passive dropping-down of tentacle due to gravity (g), $t \in [0, 30]$ is the simulation time in millisecond in behavior adaptation, $s_i(t)$ equals 0 or 1 depending on whether there is a spike at time t. The final fitness is summed over four runs with different food dropping conditions. We use a $(30, 200)$-ES and the initial mutation step-sizes are set to 0.1 with a lower bound of $1e^{-6}$. The maximal weight is set to 3000, which can be excitatory or inhibitory, and the synaptic delays are fixed to 1. All parameters are scaled between zero and one. The fitness curve for behavior optimization is shown in Fig. 4. The spikes of the input and output neurons of the best adapted individual is shown in Fig. 5. From the figure, we can see that in three of the four cases, the neural network performs optimally. In one case (last panel in Fig. 5), outputs 2 ($o2$) and 4 ($o4$) also fire, though only output 3 ($o3$) is assumed to spike. Nevertheless, $o3$ does fire the strongest.



(a) (b)

Fig. 4. (a) Fitness profile, and (b) fitness after zoomed in.

5 Discussions and Conclusions

This paper suggests a neural developmental model based on a gene regulatory network. The GRN has evolved successfully to achieve the correct developmental order, i.e., cell division, cell migration and axon growth in sequence in a simulated evolution that undergoes gene transposition, duplication and mutation. After the gene regulated neural development is complete, the connectivity and weights of the neural network are further adapted using an ES for performing a food-catching behavior in a hydra-like animat with success.

The current developmental model can be improved in various respects. Since no gene deletion is implemented in the model, the complexity of the genome tends to become more complex due to gene duplications. The cell migration

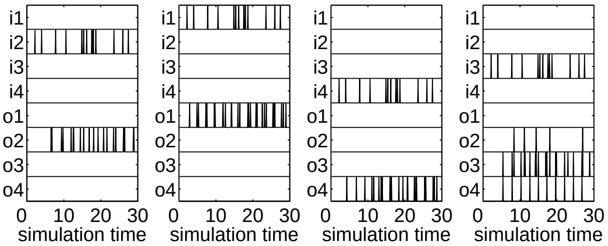


Fig. 5. Spikes at the input and output neurons for four different inputs.

behavior resulted from the current model is quite deterministic, as there is little local interactions between the cells. In the long run, a gene regulatory model for both neural and morphological development will also be investigated.

Acknowledgments

Authors are grateful to J. Eppler for his assistance in using NEST.

References

1. D. Dellaert amd R.D. Beer. Toward an evolvable model of development for autonomous agent synthesis. In *Artificial Life IV*, pages 246–257, 1994.
2. A. Cangelosi, S. Nolfi, and D. Parrisi. Cell division and migration in a 'genotype'for neural networks. *Networks - Computational in Neural Systems*, 5:479–515, 1994.
3. P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems.* The MIT Press, Cambridge, MA, 2001.
4. K.L. Downing. Supplementing evolutionary developmental systems with abstract models of neurogenesis. In *Genetic and Evolutionary Computation Conference*, pages 990–996, 2007.
5. D. Federici. Evolving developing spiking neural networks. In *Congress on Evolutionary Computation*, volume 1, pages 543–550, 2005.
6. M.-O. Gewaltig and M. Diesmann. NEST. Scholarpedia, 2007.
7. B.H. Jones, Y. Jin, B. Sendhoff, and X. Yao. Evolving functional symmetry in a three dimensional model of an elongated organism. In *Artificial Life XI*, pages 305–312, 2008.
8. H. Kitano. A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos. *Artificial Life*, 2(1):79–99, 1995.
9. S. Psujek and R.D. Beer. Developmental bias in evolution: Evolutionary accessibility of phenotypes in a model of evo-devo systems. *Evolution and Development*, 10(3):375–390, 2008.
10. D.H. Sanes, T. Reh, and W.A. Harris. *Development of the Nervous System.* Academic Press, Amsterdam, 2006.
11. L. Schramm. A model for nervous systems development controlled by a gene regulatory network. Diploma thesis, Technische Universität Darmstadt, 2007.
12. H.-P. Schwefel. *Evolution and Optimum Search.* John Wiley, 1994.
13. T. Steiner, L. Schramm, Y. Jin, and B. Sendhoff. Emergence of feedback in artificial gene regulatory networks. In *Congress on Evolutionary Computation*, pages 867–874. IEEE, 2007.