The Evolution of Artificial Neurogenesis

Dennis Wilson
dennis.wilson@irit.fr

Sylvain Cussat-Blanc
cussat@irit.fr

Hervé Luga
luga@irit.fr

University of Toulouse
IRIT - CNRS - UMR5505
21 allée de Brienne
31015 Toulouse, France

ABSTRACT

Evolutionary development as a strategy for the design of artificial neural networks is an enticing idea, with possible inspiration from both biology and existing indirect representations. A growing neural network can not only optimize towards a specific goal, but can also exhibit plasticity and regeneration. Furthermore, a generative system trained in the optimization of the resultant neural network in a reinforcement learning environment has the capability of on-line learning after evolution in any reward-driven environment. In this abstract, we outline the motivation for and design of a generative system for artificial neural network design.

Keywords

Artificial intelligence, cognitive science, neural networks

1. INTRODUCTION

Indirect representations of neural networks have proven to be an effective means for designing and evolving networks [5]. Some examples are HyperNEAT [7] and POET [4], which determine the weights between neurons in a network. Indirect encoding can be more suited to evolution, and can compactly express properties such as modularity and hierarchy in the resultant network. Here, a developmental model is proposed as an indirect encoding, where an evolved controller grows a network in a developmental environment.

Developmental systems are well suited towards action in a dynamic environment. Many methods have been proposed to allow a neural network to continuously modify its own weights, such as [6] and RELEARNN [1]. With this capability, a neural network could respond properly to situations not provided during training, and could continue to improve during use. The developmental model proposed is intended for the continuous design of the resultant neural network in a dynamic environment.

We propose a model tailored to these needs. A developmental controller is evolved as an indirect encoding of an artificial neural network, which is evaluated in a number of control scenarios. ANNs are grown in a reinforcement learning environment by the developmental model, and the relative performance over the growth of the network is used as a fitness measure for the developmental model. In this way, individuals are selected that can grow a neural network suited to a reinforcement learning environment without previously knowing the details of the environment.

The design and evaluation of this model is undergoing, but we have presented below the general methods for the model. The developmental environment is presented in § 2, and the controller for the environment is presented in § 3.

2. NEURAL MODEL

A neuron is composed in this model of two separate bodies: a soma and an axon. Dendrites are ignored in this model, and synapses are considered formed when an axon growth cone reaches the exact location of a soma in a discrete 3D space. Growth cones move through the space following morphogen gradients, imitating the behavior of growth cones following chemical cues in the developing brain [8]. In this model, morphogens are emitted by the somas and diffuse spatially throughout the entire 3D space. The emission of the guidance morphogens by the soma allows for a form of indirect communication between neurons and between a soma and axon growth cone, and can lead to complex topologies [3]. Axons are allowed to branch up to a predefined maximum number of branches, creating a new growth cone at their location. This new growth cone begins movement of its own after a delay of one timestep. Finally, axonal growth cones can die or take no action at any time step. The timescale of axonal action and weight modification is scaled by a factor evolved with the controller to mimic the fact that biological synaptic firing is a much faster process than synaptic modification.

In its current design, the 3D space is defined as a simple cube, with soma at each face. Axons grow during evaluation in a problem-based domain, and at each time step the synapses formed by axons with distal ends located at soma cells are translated into a neural network. One face of the cube is considered the input, and another the output. All soma along the input and output faces are considered input and output neurons, respectively. Growth cones at their own soma are not considered as connections, nor are growth cones that reach the input face. This design is appropriate for large networks of neurons with highly spatial input and outputs.

The resultant neural network is an integrate and fire spik-

DOI: http://dx.doi.org/10.1145/2908961.2931671

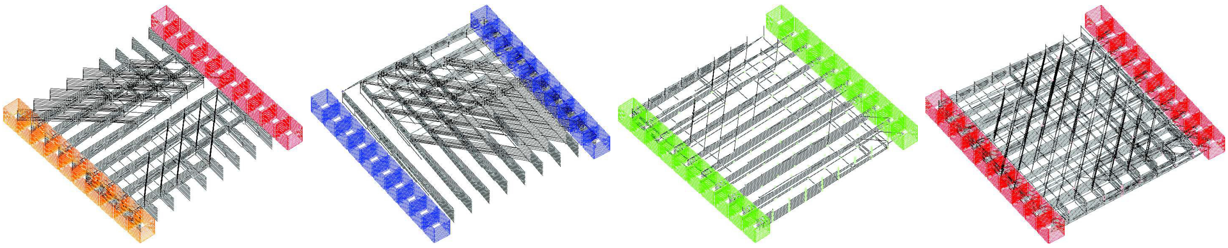

Figure 1: A bird's-eye view of four topologies produced by the model, showing different symmetry and modularity features. Axons from all somas are shown, but only the input and output soma cube faces are rendered. Coloration indicates morphogen concentrations.

ing neural network, which allows for a multitude of information to be presented by the network in the form of different firing patterns and varying gradients of neurotransmitter amount [9]. A single neurotransmitter is distributed throughout the network in a lossless manner, and is provided to the input soma at each timestep, according to the problem inputs. Each neuron has an individual firing threshold, which is modified by its controllers. The weights of the network are similarly individual for each axon and are modified by the axon contollers. The controller is described below.

3. THE CELL CONTROLLER

Axon actions, firing threshold, synaptic weights, and morphogen emission are all controlled by a genetic regulatory network. In nature, a gene regulatory network (GRN) is a network of proteins that controls the behavior of cell. In a living organism, a cell has several functions described in its genome. A gene regulatory network controls their expressions by the use of external signals collected from protein sensors localized on the membrane. These signals activate or inhibit the transcription of the genes, which then determines the cell's behavior.

The GRN model used in this work is a simplified computational model of a real gene regulatory network. It has been designed for computational purposes and not to simulate protein interactions. In it, a gene regulatory network is defined as a set of interacting proteins. The proteins use individual identifiers to determine their various concentrations and, therefore, the behavior of the network. The controller is evolved by modifying the individual protein identifiers, and by adding and removing proteins. More details about gene regulatory networks can be found in [2].

The inputs to the controller are the morphogen concentrations at the axonal growth cone position, the soma position, the current firing threshold, the weight of the axon, the neurotransmitter amount present in the soma, the neurotransmitter amount present in a post-synaptic soma if there is one, and a reward from the problem domain. The outputs are the morphogen concentrations for emission, the change in axon weight, the change in firing threshold, and the axon actions: movement along each morphogen gradient, branching, apoptosis, and quiescence. The maximum output of all the axon action outputs is taken to decide the action.

4. DISCUSSION AND CONCLUSION

Above we have described the basics of a developmental model that can produce symmetry, modularity, and heirarchy. The capabilities of this model are currently being explored by evaluating the performance of the resultant artificial neural networks in a number of reinforcement learning scenarios. An evolved controller that can act in a dynamic environment to grow and modify a neural network in a variety of problems will address some of the current shortcomings of static artificial neural networks and give insight into intelligent controllers that act as secondary parts of a neural network, modifying them based on new situations and information.

5. REFERENCES

[1] T. Brosch, H. Neumann, and P. R. Roelfsema. Reinforcement learning of linking and tracing contours in recurrent neural networks. *PLoS Comput Biol*, 11(10):e1004489, 2015.

[2] S. Cussat-Blanc and W. Banzhaf. Introduction to gene regulatory networks. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, pages 589–601, New York, NY, USA, 2015. ACM.

[3] K. Fleischer and A. H. Barr. A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In *Artificial Life III*, pages 389–416, 1994.

[4] A. Fontana, A. Soltoggio, and B. Wróbel. Poet: an evo-devo method to optimize the weights of a large artificial neural networks. 2014.

[5] T. Kowaliw, N. Bredeche, and R. Doursat. *Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks*. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2014.

[6] F. J. Maldonado and S. Oonk. Self-learning and neural network adaptation by embedded collaborative learning engine (ecle) - an overview. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Aug 2013.

[7] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, Apr. 2009.

[8] M. Tessier-Lavigne and C. S. Goodman. The molecular biology of axon guidance. *Science*, 274(5290):1123–1133, 1996.

[9] S. Thorpe, A. Delorme, and R. V. Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14(6-7):715–725, 2001.