

Developmental Tuning of Neural Networks using Weighted Function Graphs

Keith L. Downing
Department of Computer Science
The Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
keithd@idi.ntnu.no

ABSTRACT

The Weighted Function Graph (WFG) is introduced as a form of Compositional Pattern-Producing Network (CPPN) that evolves using a very indirect genotype-phenotype mapping. Although CPPNs and WFGs have broad applicability in evolving neural networks, the former are primarily used as mappings from spatial coordinates to connection weights or learning parameters. Conversely, this work uses WFGs to encode developmental activation waves across adjacent layers of neurons, akin to those observed in developing brains, which provide early patterning of synaptic change based on correlated neural firing. This WFG-based developmental tuning is then used to illustrate the Baldwin Effect in neural networks that evolve, develop and learn.

Categories and Subject Descriptors

I.2.6[Learning]: Connectionism and Neural Nets

General Terms

Algorithms

1. INTRODUCTION

Stanley's Compositional Pattern-Producing Networks (CPPNs) [12] are elegant pattern-generating tools, often employed as mappings from spatial coordinates to output properties, such as pixel color intensity. When evolved interactively, CPPNs can produce intricate pictures of high artistic quality [11]. In addition, in their HyperNEAT system [13], Stanley and co-workers have used CPPNs to map the locations of pairs of neurons to the weights of the connections between them, thus enabling a compact CPPN to assign weights to the arcs of arbitrarily large neural networks. The CPPN thus serves as a developmental encoding for synaptic weights, and one with inherent geometric properties that give rise to functionally significant topological patterns in ANN weight matrices [9, 4].

This work, though highly inspired by the CPPN research, moves in a different direction, one involving higher degrees of both implicit/indirect coding and biological plausibility.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

Copyright 2010 ACM 978-1-4503-0072-8/10/07 ...\$10.00.

First, the Weighted Function Graph (WFG) is an alternate implementation of the CPPN, one that relies on a very indirect mapping from a bit-string genotype to a network of interconnected functions (with weights on the arcs between them). Each gene merely codes for a function node and several tags, which indirectly determine the other functions to (from) which it will send (receive) weighted outputs. This exhibits *weak linkage* of modular system components, one of the key enablers of evolvability according to the popular new evolutionary theory known as *facilitated variation* [6]. This contrasts sharply with the highly direct encoding of CPPN links and weights in Stanley's NEAT[5], the evolutionary engine behind CPPNs.

Second, the route from WFG to artificial neural network (ANN) weights is more biologically plausible, and thus more circuitous, in this research than in contemporary applications of CPPNs to evolving ANNs (EANNs). Here, the WFG exploits the fundamental pattern-producing ability of the CPPN to generate simulated waves of neural stimulation. These waves spread across a neural layer, yielding complex, often heterogeneous, activation time series for each of the layer's neurons. When these correlate with wave-induced activations in neighboring layers, Hebbian-based synaptic tuning of the inter-layer connections occurs.

For each neural layer in the EANN, genes code for diverse properties, including the components of a WFG. Thus, each layer includes a wave-generating WFG employed during development to help produce an initial pattern of synaptic strengths, which can be further tuned by standard ANN learning upon exposure of the network to training situations.

This use of WFGs for neural development is motivated by our long-standing interest in the Baldwin Effect [1, 16] and recent attempts [2, 3] to understand the disruptive and facilitative roles that development plays in the Baldwinian interactions between learning and evolution, particularly when these pertain to neural structures, and thus to the evolution of intelligence. This paper continues that investigation by using WFG-based neural development in a simple test of the Baldwin Effect involving EANNs.

2. WEIGHTED FUNCTION GRAPHS

As shown in Figure 1, a WFG resembles a neural network, but with each node housing one of a number of alternative activation functions, as opposed to the standard sigmoids, step functions and hyperbolic tangents of neural-network nodes. For example, the WFG may include Gaussians, absolute values, and sine waves (as well as the common

ANN activation functions). Each WFG connection includes a weight, and all nodes compute the sum of their weighted inputs, which serves as input to the activation function, whose result becomes the node's output.

WFGs have no explicit layered organization (other than pre-designed input and output nodes), so any node can send outputs to any other node; and all nodes (except the inputs, depicted as boxes in Figure 1) can receive weighted outputs. At each timestep, the nodes undergo asynchronous activation, wherein each node simply sums the weighted outputs in its input buffer and feeds that sum to its activation function to produce an output value, which is immediately propagated to the input buffers of all post-synaptic neighbors. After a user-determined number of update rounds, the WFG's outputs are gathered from the output nodes (e.g., the lower node in Figure 1).

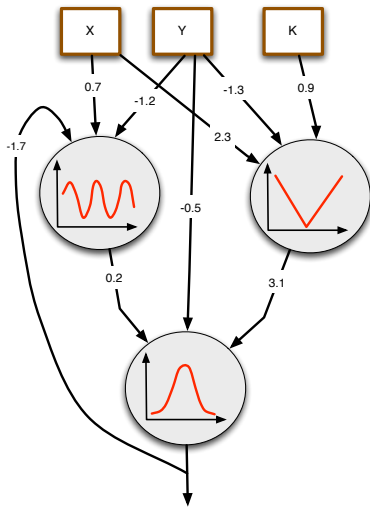


Figure 1: Simple example of a Weighted Function Graph (WFG), with inputs coming from the 3 boxes on the top, and output read from the lower of the 3 function nodes.

2.1 Evolving WFGs

WFGs are evolved via a simple bit-vector chromosome consisting of multiple segments, one for each node in the network, including inputs and outputs. Each segment consists of 5 genes that encode:

- the activation function
- the afferent connection tag
- the efferent connection tag
- the afferent weight tag
- the efferent weight tag

The first is simply an index into a list of possible activation functions, while, as shown in Figure 2, the afferent tags for node N help determine a) which nodes can send input to N, and b) the weights on those incoming arcs. Similarly, the

efferent tags influence a) the nodes to which N can send its output, and b) the weights on those outgoing arcs. The two afferent (efferent) tags constitute the afferent (efferent) *mask* of each node.

More specifically, if the afferent connection tag of node N matches the efferent connection tag of node M (above a user-defined match threshold), then M will send an excitatory connection to N. Conversely, if the match is very poor, and thus below a similar threshold, then M will send an inhibitory connection to N. For medium-strength tag matches, no connection between M and N is created. Then, the strength of an excitatory or inhibitory arc is positively correlated with the matching degree of M's efferent-weight- and N's afferent-weight tags.

For example, in Figure 2, the genotype encodes 4 nodes, labelled A-D. Each has afferent and efferent connection tags as shown (with weight tags omitted for clarity). The match thresholds imply that efferent-to-afferent tags that share 75% or more bits will produce an excitatory link between their respective nodes, while sharing of 25% or less justifies an inhibitory arc; otherwise, no connect is formed. So node A forms an excitatory link to node C, due to the close match between A's efferent and C's afferent mask. Similarly, B links to A in an excitatory manner, as does C to itself. Conversely, C inhibits both B and D due to very poor matches between C's efferent mask and the afferent masks of B and D; and finally, D inhibits A. The absolute values of the weights on each arc are then determined by comparisons of the efferent-to-afferent weight tags, which are omitted from the figure.

WFGs have a pre-defined number of input and output nodes, but the efferent and afferent tags, along with the activation functions, of output nodes can evolve, as can the two efferent tags of each input node. Since recurrent links are permitted in WFGs, both types of tags are relevant for output nodes, while input nodes are restricted to feed-forward signalling. During WFG configuration, once all connections are determined, nodes that either form no connections or, more generally, do not lie along at least one pathway from some input to some output node, are removed.

The indirect coding and tag-controlled configuration process supports a balance between stability and flexibility of interaction between nodes over evolutionary time, as a single tag-bit mutation can lead to major topological changes, although most mutations will cause little or no change. This embodies the concept of *weak linkage* among modular components, a key precursor to evolvability, as described in Gerhart and Kirschner's theory of *facilitated variation* [6]. To wit, modules whose interactions involve simple enabling signals (as opposed to complex, instructive signals) whose general structure matters more than its precise details, are both a) easy to reconfigure, since other modules may evolve to produce a similar enabling signal, and b) robust to many genetic mutations, since small changes to the signal often do not affect its triggering ability.

This attempt at increased biological plausibility in line with facilitated variation increases the indirect nature of the genotype-phenotype mapping, thus increasing the difficulty of finding *needle-in-a-haystack* solution WFGs, in problems where only isolated solutions exist. However, the general enhancement of evolvability accrued by weak linkage has clear

advantages in artificial evolution, as shown in the Enzyme Genetic Programming system [7].

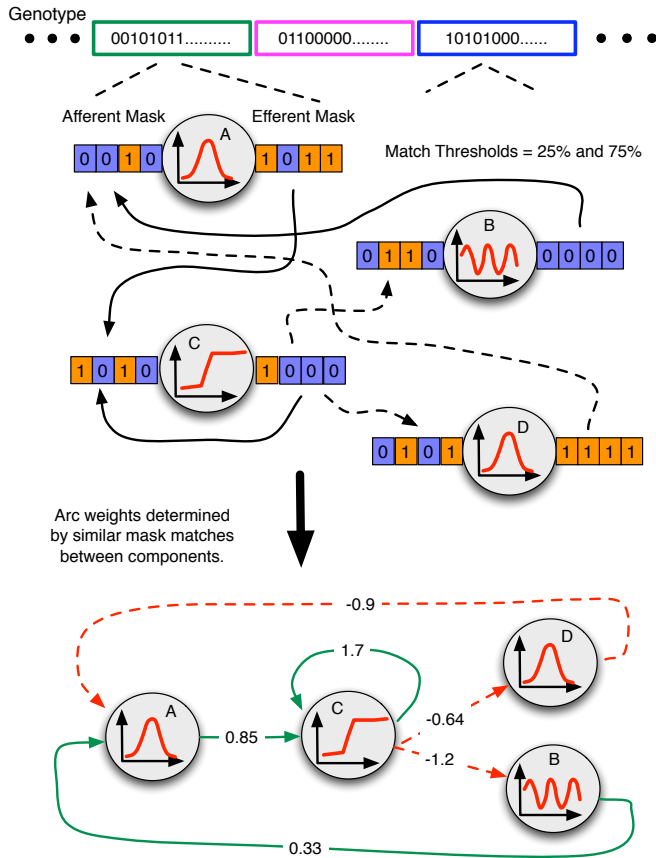


Figure 2: WFGs evolve as sets of nodes, each of which contains a function indicator along with two masks, afferent and efferent, each consisting of connection- and weight-influencing bits. The topology and arc weights are determined by tag matches, with nodes that fail to hook up being discarded. To simplify the illustration, no inputs nor outputs are included, and only the connection tags of each mask are shown. The match thresholds for inhibitory and excitatory connection formation are 25% and 75%, respectively.

3. DEVELOPMENTAL SYNAPTIC TUNING

Neuroscientific studies indicate that spontaneous waves of neural activity, modulated by cyclic-AMP (cAMP) concentrations, lead to early synaptic tuning during development, prior to the exposure to normal sensory inputs. This has shown to play an important role in the binocular segregation of connections from the retina to the lateral geniculate nucleus (LGN) [14], while others [8] postulate similar wave-induced synaptic tuning in the hippocampus, and a variety of evidence (summarized in [10] indicates both a) the presence of these waves throughout the brain during neural develop-

ment, and b) their instructive role in synaptic formation and tuning.

These waves promote neural firing such that neurons in adjacent regions/layers that happen to fire simultaneously (due to the stimulation from their respective activation waves) will have their synaptic connections modified, typically by Hebbian means. Thus, early chemical waves strongly influence the patterning of neuronal connections, prior to the molding effects of normal sensory stimuli. A comprehensive model of this phenomena would include the chemical and physical bases of reaction-diffusion processes, a reasonably straightforward but computationally-intensive endeavor.

A host of interesting neural development models can be found in [15], but this research attempts to incorporate the basic neuroscientific mechanisms into a much more abstract model. Fortunately, WFGs provide an efficient alternative for compactly representing any number of pattern-producing processes found in nature.

Just as a WFG can generate pictures by mapping Cartesian coordinates to pixel intensities, by supplementing these coordinates with an input time step, they can produce a time series of patterns, which can easily depict a dynamic structure such as an activation wave, as shown in Figure 3.

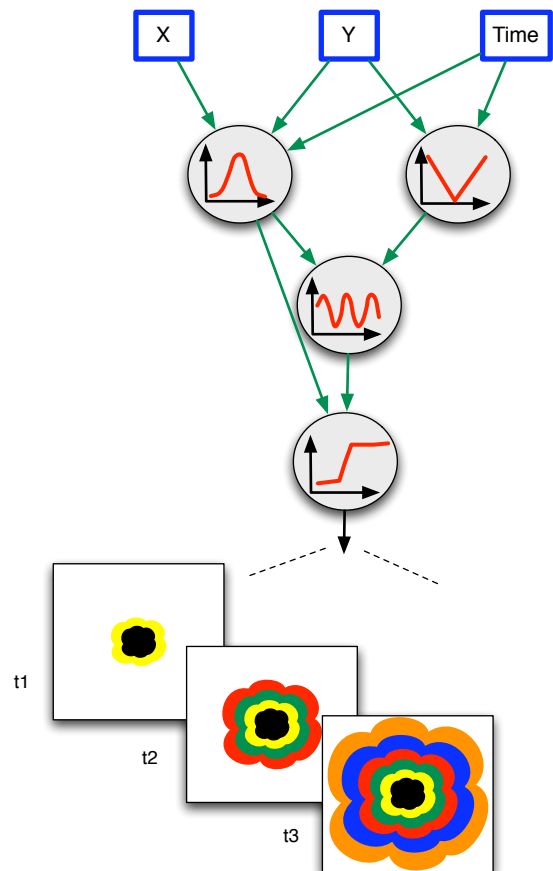


Figure 3: A WFG, when provided with Cartesian coordinates and time as inputs, produces abstract temporal activation patterns.

When neural layers are modeled as 2d surfaces, each neuron (n) has a center coordinate, (x_n, y_n) . Each layer also houses a WFG, whose mapping abstractly represents a 2d activation pattern that will spread across the layer during development. Then, to compute the wave-induced activity of n at time t , simply input x_n, y_n and t to the WFG and interpret the output value as a local activation.

When adjacent layers in an ANN include a WFG, each can be run to produce activation patterns. As shown in Figure 4, when neurons j and k in adjacent layers (J and K) have correlated wave-induced activation, Hebbian-based synaptic tuning on the j - k connection provides an early bias of the network. When the activation waves reflect some aspect of the sensory world (with its motor requirements) to which the organism will eventually be exposed, then this preliminary synaptic tuning should provide a useful *head start* for the neural network and agent.

Hence, by including WFG parameters with the other layer-specific genes in an evolving ANN, any pair of interconnected layers with WFG-based development stimulation can achieve an evolving prenatal bias of its synaptic weights.

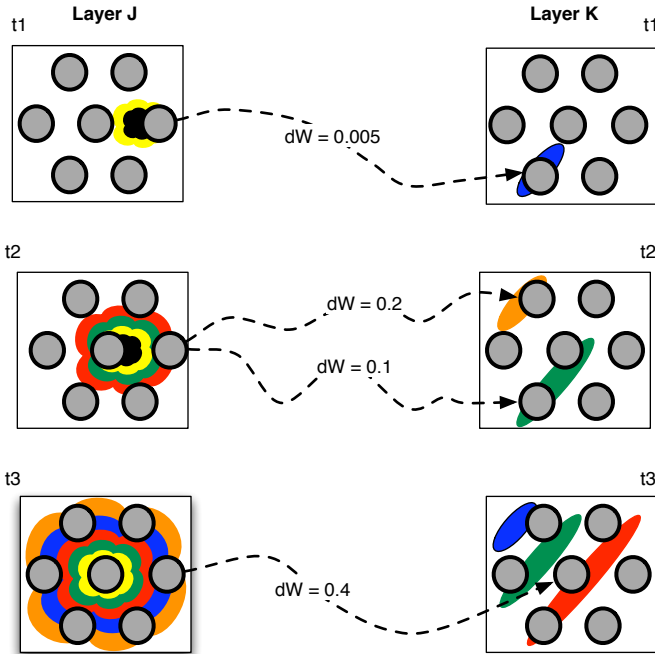


Figure 4: Two adjacent neural layers (J and K) exposed to different WFG-generated activation waves over 3 timesteps. Co-stimulated neurons in each layer induce correlation-based synaptic changes (dW), only a few of which are shown. All colored region indicate some level of stimulation, with red (blue) being most (least) intense.

4. A BALDWIN-EFFECT MODEL

To assess the viability of weighted function graphs as mod-

els of neural activation spreading during development, we use a primitive test of the Baldwin Effect (B.E.).

As described in [1, 16, 3], B.E. concerns the ability of learning to accelerate evolution via a two-stage process. In phase I, individuals with phenotypic plasticity achieve higher fitness than their innate traits could alone. This moves the population distribution toward plastic individuals. In phase II, some of these learned skills become innate by chance mutations. Selection pressure favors this assimilation of plastic features into the genome and developmental process when a) the environment is reasonably static across the generations, and b) learning has a fitness cost, and c) the genotype-phenotype mapping is not overly complex.

Although B.E. seems plausible for some phenotypic traits, such as the size of muscles and the efficacy of physical skills, its relationship to the evolution of intelligence is more tenuous, given contemporary understanding of the brain, neural development and synaptic change. If learning is generally equated with synaptic change, then how can the modification of a few of the (human) brain's 100 trillion synapses be assimilated into DNA consisting of approximately 25 thousand genes? As [3] argues, an interpretation of neural Baldwinism as a heterochronous shift seems more appropriate: B.E. phase II involves the transfer of a certain degree of neurogenesis, synaptogenesis and synaptic tuning from post-natal, experience-driven learning to pre-natal development.

In [3], this heterochronous neural Baldwinism is illustrated by a shift of the brunt of neurogenesis and synaptogenesis from the post-natal (learning) to the pre-natal (developmental) phase of life, as summarized in Figure 5. In this work, we turn to the third key factor, synaptic tuning, and explore the degree to which it can be assimilated into neural development.

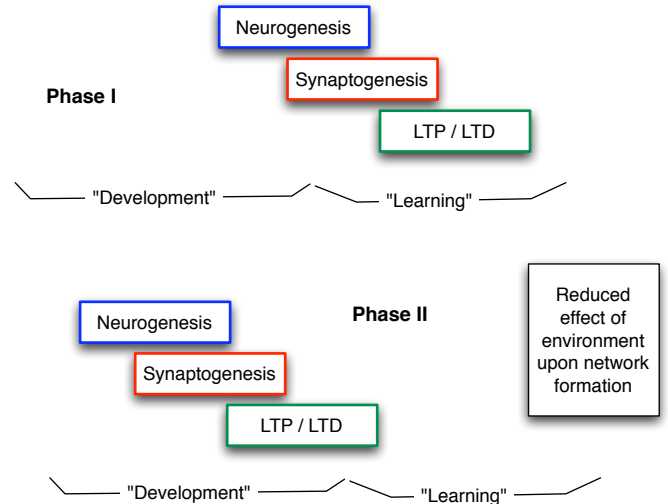


Figure 5: The heterochronous view of the Baldwin Effect (phases I and II) with respect to three key mechanisms of brain formation. Long-term potentiation (LTP) and long-term depression (LTD) are two forms of synaptic tuning.

This investigation utilizes a simple two-layered neural net-

work, with both input and output layers housing evolved WFGs that generate pre-natal (i.e. prior to the presentation of training cases to the network) activation waves, which then facilitate developmental synaptic modification, as depicted in Figure 4.

The task of each network is simply to map binary input patterns to output patterns, with input-output pairs generated randomly at the beginning of each evolutionary run. The network’s fitness is inversely proportional to both a) the total output error across all learning epochs, and b) the learning effort, a simple function of the learning rate and number of epochs, both evolvable parameters.

The general intuition is that networks will originally rely on a high learning effort to reduce output error and thus achieve high fitness, as exemplifies B.E. phase I. Then, as WFGs evolve for the input and output layers, the resulting developmental activation waves and concomitant prenatal synaptic tuning should eventually bias synapses to reduce output error prior to the presentation of any training cases. Thus, learning effort (and its fitness cost) can decrease to achieve B.E. phase II.

4.1 Modeling Details

In all of the scenarios discussed below, the neural network consists of an input and output layer, fully connected in a strictly feed-forward manner, with both containing 16 neurons. Each output neuron has a simple linear activation function: the output equals the sum of the weighted inputs.

Synaptic weights are randomly initialized to values between -0.05 and +0.05, and after each round of synaptic modification, either during development or learning, the absolute values of all incoming weights to each output neuron are normalized geometrically (i.e. so that the weight vector has length 1). The binary input and output patterns are sparse, consisting of one 1 and many 0’s.

Table 1 summarizes the 6 basic characteristics that are evolved for each neural network. Each of the two WFG’s consists of 14 genes of the type shown in Figure 2: 10 for the internal nodes, 1 for the output node, and 3 for the input nodes (coding for x, y and time). The function set for evolved WFG’s consists of: identity, absolute value, sine, gaussian, and the logistic (sigmoid) function, while WFG connection weights are restricted to reals in $[-5, 5]$. Each of the 16 input and output nodes has Cartesian coordinates to use as inputs to the respective WFG during development.

Symbol	Name	Range
<i>Development</i>		
D_r	Developmental tuning rate	[0, 0.5]
D_s	Activation wave steps	[0, 5]
WFG_{in}	WFG for input layer	0-10 internal nodes
WFG_{out}	WFG for output layer	0-10 internal nodes
<i>Learning</i>		
L_r	Learning rate	[0, 0.5]
L_e	Learning epochs	[1, 10]

Table 1: Genes in the genetic-algorithm chromosome for all scenarios described in this article.

During both developmental and training-based synaptic

tuning, the standard correlational Hebbian rule of equation 1 applies:

$$\Delta w_{i,j} = \lambda x_i x_j \quad (1)$$

where $w_{i,j}$ is the weight on the connection from neuron j to neuron i , λ is either D_r or L_r (see Table 1), and x_i and x_j are the outputs of neurons i and j , respectively.

Equation 2 expresses the output error of each neural network:

$$E = \frac{1}{L_e \parallel N \parallel \parallel C \parallel} \sum_e \sum_{c \in C} \sum_{n \in N} |x_n^{e,c} - t_n^c| \quad (2)$$

where N is the set of output neurons, C is the set of training cases, $x_n^{e,c}$ is the output of the n th neuron on case c during epoch e , and t_n^c is the target output for neuron n on case c .

Equation 3 provides the fitness function used in all scenarios:

$$f = \frac{1}{\sqrt{E} + \Theta L_e L_r} \quad (3)$$

where $L_e L_r$ is the learning effort and Θ is the tuning tax, typically 0.05.

The key genetic-algorithm parameters include a population size of 20, full-generational replacement with a rank-based selection mechanism and elitism of one individual, a crossover rate of 0.8 and a mutation rate of 0.05 per bit.

5. RESULTS

5.1 Scenario 1

This scenario employs 10 training cases: sparse input and target-output vectors containing exactly one 1 and 15 0’s. The 50-generation run of Figure 6 illustrates a Baldwin Effect supported by WFG-based synaptic tuning. Early on, the learning rate rises, thus decreasing output error and increasing fitness, thereby achieving B.E. phase I. Then, around generation 25, the learning effort drops dramatically, while maximum fitness continues its slow rise, which ushers in B.E. phase II. The drop in learning effort produces a slight rise in output error, which trades off with learning effort in the fitness function.

What enables learning effort to decrease without incurring a major error increase is the WFG-based developmental process, which evolves into two activation patterns across the input and output layers, as shown in Figures 7 and 8. These reduce the output error (on the 10 training/test cases) from 0.25 prior to developmental synaptic tuning to 0.17 afterwards (but prior to any learning). Since developmental effort has no cost in the model, fitness can increase when any significant amount of synaptic tuning is transferred from learning to development, provided that output error does not increase substantially as a side effect.

Because developmental tuning has no basis in the training data, evolution must search for WFG-generated activation waves that fortuitously bias the synapses to learn those patterns. In examples such as this, the bias is significant: over 30%, and in some examples, it reaches as high as 60%.

In addition to the WFG’s, the four other evolved parameters for this best-of-run individual appear in Table 2, which

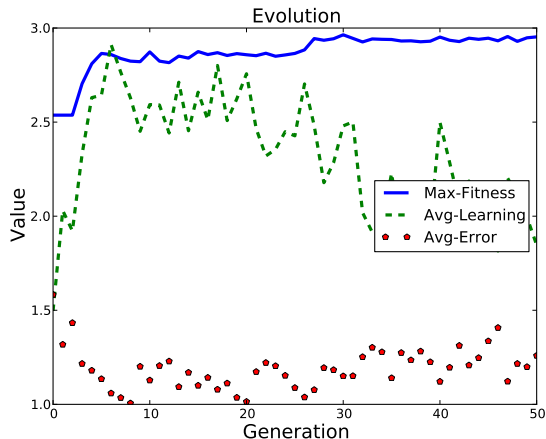


Figure 6: The evolutionary progression of best-of-generation fitness, along with population-average values for learning effort and output error, which is scaled by 10 for ease of viewing. Note the initial increase in learning effort (B.E. phase I) that eventually gives way to a decrease, while fitness continues to rise (B.E. phase II).

reveals a high tuning rate for development, 0.251, compared to the corresponding rate for learning, 0.1666.

D_r	D_s	L_r	L_e
0.251	5	0.1666	10

Table 2: Four evolved parameters for the most fit individual of scenario 1.

Despite the use of elitism, a noticeable jitter appears in the fitness plot of Figure 6. This stems from the random initialization of network weights prior to developmental tuning, which cannot always produce identically-biased synaptic patterns (prior to learning) when given diverse initial-weight distributions. However, in most cases, a top-fitness individual in generation k remains very highly fit when cloned into generation $k+1$.

To test the generality of this result, 20 runs were performed using different sparse data sets of size 10, with exactly one 1 and 15 0’s in each vector. As shown in Figure 9, fitness rises throughout the 50 generations, while error declines. Learning effort begins high but gradually decreases as developmental synaptic tuning evolves to help reduce output error.

5.2 Scenario 2

In this scenario, 20 sparse input-output pairs (again containing exactly one 1 apiece) are given to the network. The population size (20) and all other parameters from scenario 1 are reused. The evolution in Figure 10 clearly shows a gradually increasing fitness while learning effort begins high but gradually declines. In this scenario, the learning decline is less obvious in the population averages than in the (cascading) best-of-generation learning trace.

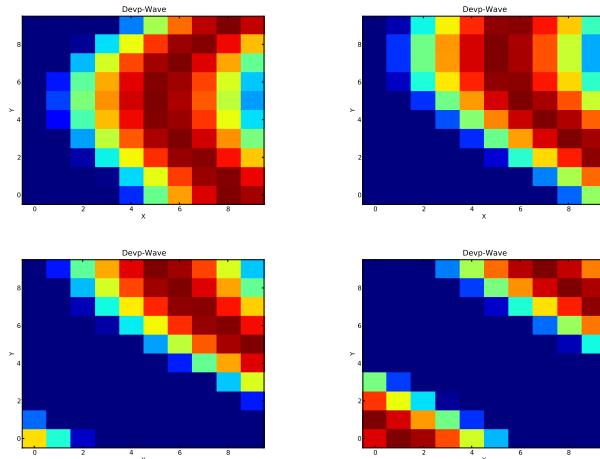


Figure 7: Four sequential (left-to-right, top-to-bottom) snapshots of a WFG-generated activation sequence for the network’s input layer. Lighter colors indicate higher activation.

Figure 11 shows results common to many of the (unpublished) runs: a) the evolved WFG’s ignore the timestep and produce the same pattern for each timestep, and b) the output layer’s WFG produces a monotonic pattern of either high or low activation in all spatial locations. However, this mundane pair of patterns reduces output error by 44% prior to any learning. The remaining phenotypic traits appear in Table 3. Note that only 3 developmental steps and 5 learning epochs were employed by this individual.

D_r	D_s	L_r	L_e
0.492	3	0.148	5

Table 3: Four evolved parameters for the most fit individual of scenario 2.

6. REFERENCES

- [1] J. M. BALDWIN, *A new factor in evolution*, The American Naturalist, 30 (1896), pp. 441–451.
- [2] K. L. DOWNING, *Development and the Baldwin effect*, Artificial Life, 10 (2004), pp. 39–63.
- [3] ———, *The Baldwin effect in developing neural networks*, in Proceedings of the 12th Genetic and Evolutionary Computation Conference, Portland, Oregon, 2010, ACM Press, pp. 555–562.
- [4] J. GAUCI AND K. STANLEY, *Autonomous evolution of topographic regularities in artificial neural networks*, Neural Computation, 22 (2010), pp. 1860–1898.
- [5] KENNETH AND R. MIIKKULAINEN, *Evolving neural networks through augmenting topologies*, Evolutionary Computation, 10 (2002), pp. 99–127.
- [6] M. W. KIRSCHNER AND J. C. GERHART, *The Plausibility of Life: Resolving Darwin’s Dilemma*, Yale University Press, New Haven, CN, 2005.

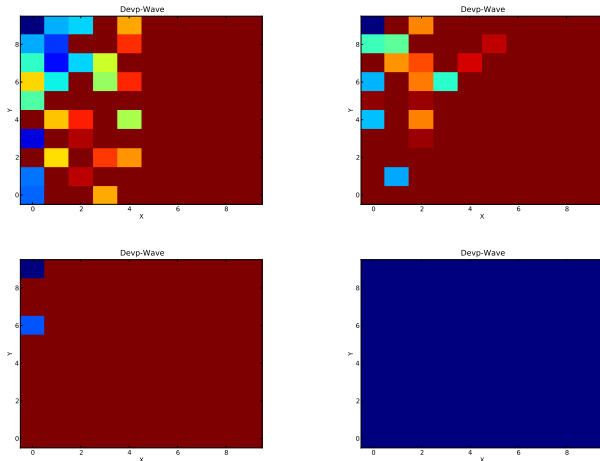


Figure 8: Four sequential (left-to-right, top-to-bottom) snapshots of a WFG-generated activation sequence for the network's output layer.

- [7] M. A. LONES AND A. M. TYRRELL, *Biomimetic representation in enzyme genetic programming*, Genetic Programming and Evolvable Machines, 3 (2002), pp. 193–217.
- [8] B. MCNAUGHTON, L. BATTAGLIA, O. JENSEN, E. I. MOSER, AND M. B. MOSER, *Path integration and the neural basis of the 'cognitive map'*, Nature Reviews Neuroscience, 7 (2006), pp. 663–678.
- [9] S. RISI, J. LEHMAN, AND K. STANLEY, *Evolving and placement and density of neurons in the hyperneat substrate*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010), (2010), p. 8.
- [10] D. SANES, T. REH, AND W. HARRIS, *Development of the Nervous System*, Elsevier Academic Press, Burlington, MA, 2006.
- [11] J. SECRETAN, N. BEATO, D. D'AMBROSIO, A. RODRIGUEZ, A. CAMPBELL, J. FOLSOM-KOVARIK, AND K. STANLEY, *Picbreeder: a case study in collaborative evolutionary exploration of design space*, Evolutionary Computation, 0 (2011), p. to appear.
- [12] K. STANLEY, *Compositional pattern producing networks: a novel abstraction of development*, Genetic Programming and Evolvable Machines: Special Issue on Developmental Systems, 8 (2007), pp. 131–162.
- [13] K. STANLEY, D. D'AMBROSIO, AND J. GAUCI, *A hypercube-based encoding for evolving large-scale neural networks*, Artificial Life, 15 (2009), pp. 189–212.
- [14] D. STELLWAGEN AND C. SHATZ, *An instructive role for retinal waves in the development of retinogeniculate connectivity*, Neuron, 33 (2002), pp. 357–367.
- [15] A. VAN OUYEN, ed., *Modeling Neural Development*, The MIT Press, Cambridge, MA, 2003.
- [16] B. WEBER AND D. DEPEW, eds., *Evolution and Learning: The Baldwin Effect Reconsidered*, The MIT Press, Cambridge, MA, 2003.

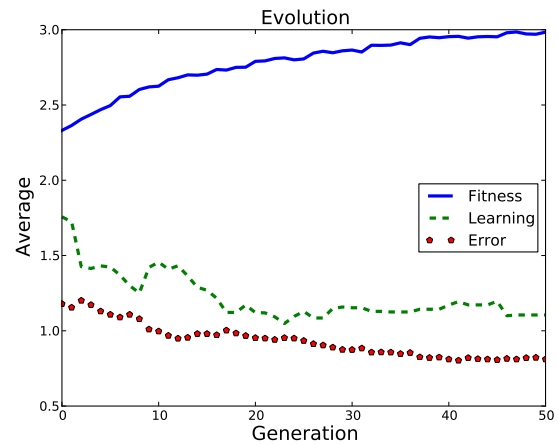


Figure 9: Averages over 20 runs of the best-of-generation network's fitness, learning effort and output error for networks trained on 10 sparse training cases.

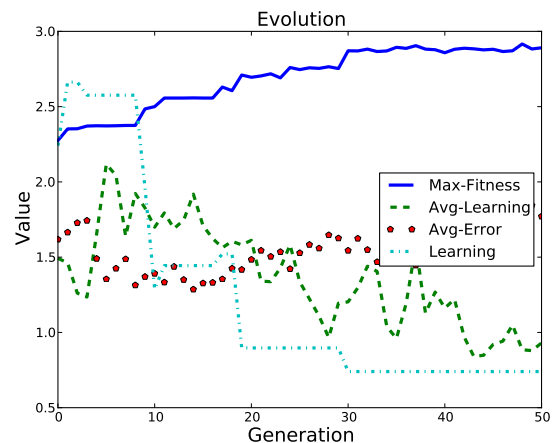


Figure 10: The evolutionary progression of fitness and learning effort for the best-of-generation individuals. Population averages for learning effort and output error (scaled by 10) are also plotted.

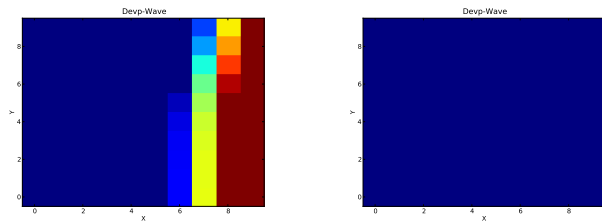


Figure 11: Static (i.e. non-varying for all 3 developmental timesteps) WFG-generated activation patterns for the input (left) and output (right) layers of the neural network for the best-fit individual of scenario 2. These simple patterns reduced the pre-learning output error from 0.25 to 0.14, i.e. by 44%.