The Baldwin Effect in Developing Neural Networks

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

ABSTRACT

The Baldwin Effect is a very plausible, but unproven, biological theory concerning the power of learning to accelerate evolution. Simple computational models in the 1980's gave the first *constructive proof* of its potential existence, and subsequent work in evolutionary computation has shown the practical, computational, advantages of hybrid evolution-learning systems . However, the basic theory, particularly its second phase (involving genetic assimilation of acquired characteristics) is difficult to reconcile in systems controlled by neural networks, particularly those that arise from their genotypes via a complex developmental process. Our research uses new evidence of the blurred distinction between development and learning in natural neural systems as the basis for an abstract model displaying the Baldwin Effect in artificial neural networks that evolve, develop and learn.

Categories and Subject Descriptors
I.2.6[**Learning**]: Connectionism and Neural Nets
General Terms
Algorithms

1. INTRODUCTION

The Baldwin Effect (B.E.), an intriguing theory as to learning's ability to accelerate evolution [3, 16], sporadically generates interest in the field of evolutionary computation, both from theoretical and practical perspectives [15, 2]. Concerning theoretical evolutionary biology, the veracity of B.E. is difficult to test, due to the complications of evolving populations of organisms - with quantifiable learning abilities - over many generations. For more practical computational issues, such as improving search-based problem solvers, B.E. provides an easily-implemented paradigm for combining (global) evolutionary computation with (local) learning in a manner that often accelerates the overall search process. In turn, these computer models shed light on the (still rather vague) theoretical details of B.E.

Notwithstanding the myriad adaptive mechanisms in living organisms, the most potent forms of learning in nature are normally associated with networks of neurons (i.e. nervous systems). Thus, to understand the full biological ramifications of B.E., one must consider it in the context of neural-network-based learning systems. In nature, these networks emerge from a complex developmental process that (loosely speaking) uses DNA instructions to produce brains; but nowhere do DNA segments (i.e. genes) code for individual neurons nor the connections between them: the mapping from genotype to phenotype is extremely *indirect*; and this confounds B.E by greatly reducing the possibility for typical learned associations (often realized as synaptic modifications) to eventually be expressed in the genotype and transmitted to future generations. It is this transfer of phenotypic to genotypic modifications - in the opposite direction of development - that is prerequisite for the Baldwin Effect. Although B.E. adherents view this reverse-transfer process as very indirect in its own right, there must, at some level, exist a relationship between phenotypic and genotypic change for B.E. to remain plausible.

In this work, we examine the mechanisms traditionally associated with neural-network development and learning. We find that the border between the two processes contains a significant gray area with respect to the creation of new neurons (a.k.a. neurogenesis) and synapses (a.k.a. synaptogenesis), along with the tuning of those synapses. To wit, neurogenesis and synaptogenesis are not restricted to early neurodevelopment, as once believed. Recent evidence [14] shows that neurons can be generated and inter-connected throughout life, depending upon an animal's mental (and physical) challenges. Thus, neuro- and synaptogenesis can be shared between development and learning, with the genome brokering the actual division of labor. Furthermore, shifts in this distribution that transfer some of the burden from mature (learning) stages of life to early (developmental) stages, support the reverse-transfer requirements of the Baldwin Effect.

Motivated by these biological findings and their implications for the B.E., we devise an abstract model in which a) artificial neural networks (ANNs) evolve, develop and learn, and b) one fundamental aspect of learning involves neuro- and synaptogenesis (to handle unsolved problem instances). By monitoring the evolving division of neuron-generating labor between development and learning, we observe this more flexible interpretation of the Baldwin Effect.

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

2. THE BALDWIN EFFECT

One of the first proposals that learning could accelerate evolution was Jean-Baptiste Lamarck's (1744-1829) *inheritance of acquired characteristics*, wherein physical and mental changes incurred during one's lifetime could be passed on directly to offspring. In terms of our contemporary understanding of the germ-soma distinction, Lamarckianism implies a reverse transcription of the modified phenotype back into the genotype, a process that is fully realizable and often useful in evolutionary algorithms, but biologically unrealistic.

In 1896, James Baldwin postulated an indirect mechanism for the eventual inheritance of acquired characteristics [3]. This *Baldwin Effect* involves two stages. In phase I (Figure 1), assume a set of genotypes spread uniformly about a sub-optimal region, D1, of the fitness landscape. If phenotypes have plasticity, then each can essentially perform local search in the fitness landscape (as shown by the circles with horizontal arrows), and a rough estimate of phenotypic fitness will be the time-averaged landscape locations of the phenotype. Clearly, those phenotypes lying near the base of the optimal peak will have better opportunities to learn their way to higher fitness. Hence, they will have a selective advantage, and the population distribution will move from D1 to D2, in the direction of P*, the optimal phenotype.

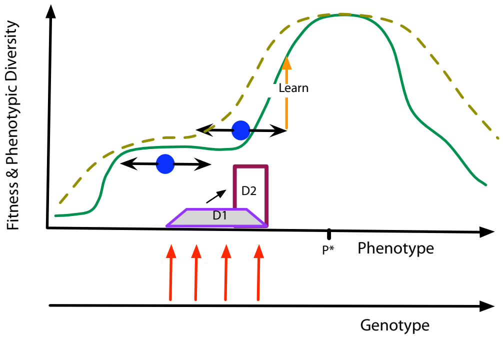

Figure 1: The Baldwin Effect Phase I: Learning speeds evolution by effectively smoothing (from the solid to the dashed curve) the fitness landscape.

To move, and not merely redistribute, the genotype pool, evolution relies on genetic operators (mutation, crossover, inversion, etc.). If the genotype and phenotype space are well correlated [12], then genetics can initiate the emergence of innately optimal phenotypes, *natural born P*s*, and, in general, lead to a flattening of distribution D2 into D3 (Figure 2). Additionally, if learning has a cost, as it normally does [12], then the *P* learners* will pay it but the natural-born P*s will not, thus giving the latter a selective advantage and moving the population distribution from D3 to D4, where the previously-learned phenotype, P*, becomes fully innate.

Thus, in the Baldwin Effect, learning accelerates evolution; and then, if the fitness landscape is static, *and the genotype and phenotype spaces are well correlated*, evolution obviates learning via genetic assimilation.

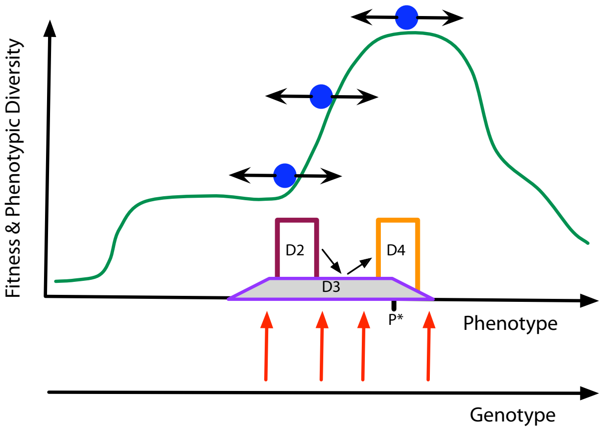

Figure 2: The Baldwin Effect Phase II: Genetic assimilation of P*.

3. NEURAL DEVELOPMENT

Unfortunately, in nature, the developmental process greatly complicates the genotype-phenotype mapping, as discussed more thoroughly in [4]. Simple counting arguments show the impossibility of genes coding for individual neurons or synaptic strengths, so learning-driven synaptic change can only find very indirect parallels in development. For example, Figure 3 provides a plausible correlation between a learning change and a genetically-encoded developmental result. As shown, if the synaptic strength between a neuron in region A and one in B is modified during learning (thick arrow), this can be roughly approximated by genetically-controlled A-B connectivity changes in later generations. With more (or less) connections between A and B in the future, there are more (or less) opportunities for synaptic enhancement via Hebbian learning. Since standard Hebbian learning (via long-term potentiation) relies on correlated firing (between neurons in regions A and B in this case), there is a greater chance of such correlation (and thus a promising starting-point for synaptic strengthening) when more synapses (of even weak efficacy) link areas A and B.

The scenario of Figure 3 relies heavily upon the common characterizations of development and learning in neural networks, based both on traditional neuroscientific explanations [10] and practical issues of artificial neural network deployment. Here, development involves the generation and linking of neurons, while learning consists solely of the fine-tuning of synaptic strengths; and the two processes occur in lock step, with development ending before learning begins. Furthermore, development is largely controlled by the innate genome (a.k.a nature), while the environment (a.k.a nurture or experience) predominantly governs learning.

Another interesting perspective on development and the Baldwin Effect stems from evidence that the lock-step model, though correct to a rough approximation, neglects new neurobiological evidence of temporal overlap between neurogenesis, synaptogenesis, and synaptic tuning. For example, many studies (summarized in [13]) find high levels of long-term potentiation (LTP) and long-term depression (LTD) - both forms of synaptic tuning - during development. In fact,

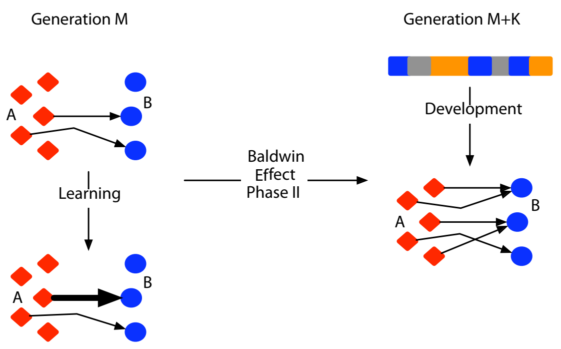

Figure 3: Genetic assimilation of a synaptic change during learning (thickened arrow) into a similar developmental outcome, which could facilitate (even accelerate) the original learning result.

the rates of LTP and LTD (i.e. learning rates) are actually very high during development and much lower during adult life. In addition, recent work by Shors [14] reveals that a) neurogenesis occurs throughout life, particularly in the dentate gyrus (DG) of the hippocampus, but b) those neurons only hook up to other neurons (and ultimately survive) if the organism subsequently performs cognitively-challenging tasks.

This new evidence motivates a reinterpretation of the Baldwin Effect in neural networks. Instead of viewing the second phase as one of converting synaptic-strength changes (i.e. classic learning) into genomic codes for controlling neurogenesis and synaptogenesis (i.e. classic development) - which represents the reverse encoding of the results of one process into two dramatically different processes - we propose an alternative that involves a quantitative, rather than a qualitative, conversion.

To wit, the second (assimilation) phase of the Baldwin Effect may only involve a change in the *rates* of neurogenesis, synaptogenesis and LTP/LTD across an organism's life stages, as shown in Figure 4, where the four makeshift graphs roughly illustrate the distribution of effort among these processes at different evolutionary and developmental stages. Under the view that adaptive changes in later life are predominantly governed by the environment, not the genome, a Baldwinian modification could simply be to move more of that adaptive change into earlier life stages, where genomic control may dominate.

For example, when the biochemical bases for LTP and LTD arose in evolution, both processes may have been very active throughout life, requiring constant environmental signaling to tune neural circuitry. However, over many (thousands of) generations, genomic changes could have arisen such that the early stages of development utilized neurogenesis, synaptogenesis and high LTP/LTD to form much of this circuitry with a minimum of environmental influence. Similarly, the rates of neurogenesis and synaptogenesis could have originally been much less variable throughout life, but evolution has gradually found genomes coding for an acceleration of these processes in early development; and thus, more of these activities became governed by genomic rather than environmental factors. The neural plasticity that remains in today's adult genomes (of any species), may represent that flexibility which evolution found optimal with respect to factors such as a) the coding limits of the genome, b) constraints of the animal's brain and body, and c) earth's environment and the rates of change associated with it.

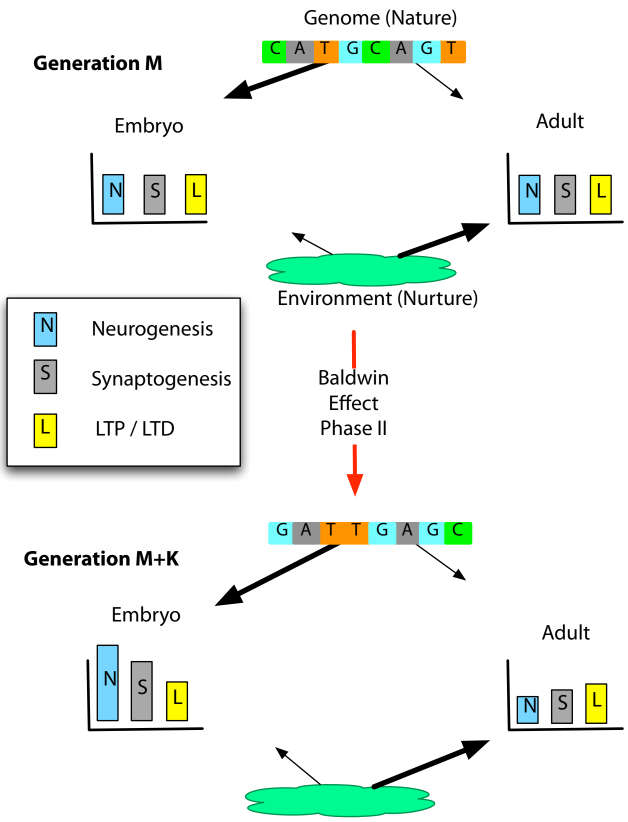

Figure 4: An alternate, quantitative interpretation of the Baldwin Effect, wherein the brunt of neuro- and synaptogenesis moves into the embryonic phase.

4. THE BALKO MODEL

Our system, BALKO, explores this alternate interpretation of the Baldwin Effect using Kohonen networks [11]. It employs a population of evolving nets whose topologies are jointly determined by development and learning, with the balance between these two processes governed by the genome.

As shown in Figure 5, a BALKO genotype codes for the essential parameters of the development and learning algorithms. The former determines the initial number of nodes in the Kohonen (self-organizing) layer - which is organized as a ring - along with the connectivity pattern between the input and self-organizing layers. It also specifies the range of initial weights for all connections. The learning parameters include the initial neighborhood radius for self-organized map (SOM) formation, the derivative of the radius, and the actual learning rate used during weight modification. These are considered *tuning* factors, while a more substantial *growth* aspect of learning consists of the dynamic addition of new nodes (drawn as dotted circles in Figure 5) to the Kohonen layer in situations where none of the existing nodes fires hard enough on an input pattern. So along with synaptic tuning, learning in BALKO encompasses some degree of neuro- and synaptogenesis.

To assess the fitness of a phenotype network, a set of normalized (originally binary) input patterns are sequentially fed into the network, with each instigating normal SOM competitive learning in the Kohonen layer. After this learning period (whose duration is genetically determined), the individual's fitness is derived from three primary factors: 1) the ability to differentiate among inputs (quantified as entropy), 2) the topological nature of the SOM, and 3) the total learning effort, as shown at the bottom of Figure 5.

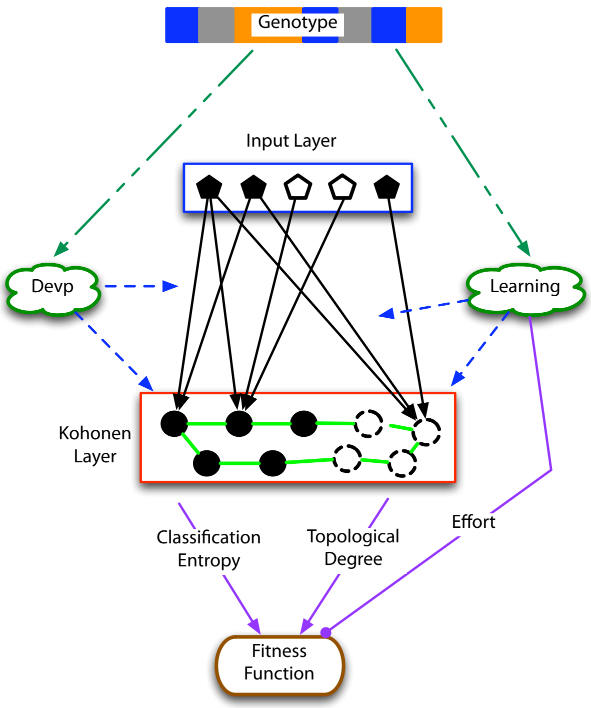

Figure 5: Overview of the BALKO model.

4.1 Evolution

BALKO uses a standard genetic algorithm [8], with a bit-vector chromosome, single-point crossover, and bit-flipping mutations. Bit sequences of varying lengths (i.e. genes) code for a variety of developmental and learning parameters, as summarized in Table 1.

For the simulations described in this paper, the (single-point) crossover rate is 0.8, while the bit-wise mutation probability is 0.03. Tournament selection of size 3 with a random-winner probability of 0.1 is employed, along with an elitism value of one individual.

Symbol	Name	Range
	Development	
K_{init}	Init Size of Kohonen Layer	[5, 20]
x_{seed}	Init value for Ω	[0,1]
R_Ω	Rate value for Ω	[3,4]
Ω_{iter}	Iteration value for Ω	[0,10]
D_s	Developmental steps	[1,20]
	Learning	
R_{init}	Init neighborhood radius	[0,3]
δR	Derivative of neighborhood radius	[-1, 0]
λ	Learning rate	[0,1]
	Both	
w_i	Max init connection weight	[0,1]

Table 1: Genes in a BALKO chromosome. Ω is the logistic map used for determining connectivity during development.

4.2 Development

To determine the topology of connections between the input and Kohonen layers, BALKO employs the classic logistic map (below) as a useful abstraction, designed not to mimic biology in a detailed manner, but to stand in for any number of complex, non-linear, neurodevelopmental processes:

$$\Omega(x) : x_{i+1} \leftarrow R_\Omega x_i(1 - x_i) \tag{1}$$

with an evolving rate value (R_Ω) restricted to the range [3, 4], which determines the degree of order or chaos in the computed pattern. As shown in Table 1, the initial value of x (x_{seed}), as well as the iteration factor (Ω_{iter}), also evolve. The core of development is the *generate-offsets* algorithm:

```
define generate-offsets ($N_t$)
    x ← $x_{seed}$; offsets ← {} ;
    For step = 1 to $D_s$:
        For i = 1 to $\Omega_{iter}$: x ← $\Omega(x)$
        offsets ← offsets ∪ {x*$N_t$}
    return offsets
```

This uses the logistic map to produce a list of offsets, which is employed by *generate-indices* for each of the Kohonen neurons.

```
define generate-indices ($index_0$, offsets, $N_t$)
    index ← $index_0$ ; indices ← {}
    For z ∈ offsets:
        index ← [round(index + z) modulo $N_t$]
        indices ← indices ∪ { index }
    return indices
```

The returned indices refer to the neurons (in a target group) to which a focal neuron will connect. N_t is the cardinality of this target-neuron set, while $index_0$ is the starting point within the target set. In BALKO, the target group is the input-layer. For each neuron, n_i in the Kohonen layer, *generate-indices* is called with the index of the corresponding input-layer neuron (which is the ith neuron when the two layers have the same size but is otherwise scaled to the size differences). The returned indices denote those input neurons that send afferents to n_i; multiple occurrences of an index entail a stronger initial synaptic strength from that particular input neuron. During the ensuing learning stage, only these connection weights can be modified.

Depending upon the values of the evolved developmental parameters, the connection patterns can be dense or sparse, regular or chaotic. For example, the logistic map is known to be periodic when $3.4 \leq R_\Omega \leq 3.6$ and chaotic when $3.6 \leq R_\Omega \leq 3.8$.

4.3 Learning

BALKO uses the standard self-organized map learning algorithm [11], with the winner node (i.e. that with the highest output value) and all neighbors (within a gradually shrinking radius) having their existing connections tuned to more faithfully detect the current input pattern, via the following update rule:

$$\triangle w_{i,k} = \lambda(I(i) - w_{i,k}) \tag{2}$$

where I(i) is the output of the ith input node, and $w_{i,k}$ is the weight on the connection from the ith input to the kth Kohonen-layer node. The neighborhood radius, with initial value R_{init}, changes by δR after each training instance. Learning continues until $R \leq 0$. Since the genome determines λ, R_{init} and δR, evolution controls the tempo, extent and duration of learning.

If none of the Kohonen neurons fires above a particular threshold (of 0.8 for the current simulations) for an input pattern, P, then the network is assumed to *fail to detect* P. If the Kohonen layer has not reached its maximum size (of 20 neurons in these simulations), then a new neuron, n_k is added; it is strongly biased toward detecting P by adding connections from exactly those input neurons that fired above a second threshold (0.1 in these runs) on P.

In BALKO, synapses with weights below 0.01 are removed, as are Kohonen neurons without afferents. So learning can modify synapses to the point that connections and even neurons vanish from the simulation. This can have a positive effect upon fitness, since the dendritic density of Kohonen neurons increases the total learning effort, which negatively impacts fitness (as shown below).

4.4 Input Patterns

Input vectors for the network have a variable (parameter-controlled) degree of similarity to one another, thus abstractly representing the fact that an organism's natural sensory input is far from random, but reflects the structure of the real world. To this end, BALKO uses a Hopfield network [9] of size I_s (the number of bits in an input pattern) to generate patterns. It begins by randomly choosing the bi-directional weights between each pair of neurons (from the range [-1, 1]), thus forming an explicit *bias* as to relationships between pattern bits. Next, it generates a random set of on/off states for the I_s nodes. Then, simulated annealing is applied to the network, with the probability of a state change governed by both the sum of weighted inputs and the current temperature. This process is performed N times to produce N patterns, each of which is normalized prior to presentation to the Kohonen network. The length of the annealing process positively influences the similarity of the N patterns.

4.5 Fitness Testing

Three factors contribute to the fitness of a BALKO phenotype:

1. Classification entropy (H_C) - the degree to which wins are evenly distributed about the Kohonen nodes.

2. Topological degree (C_W) - the amount of correlation among the input weight vectors of neighboring nodes in the ringed Kohonen layer.

3. Learning effort (E_L) - a weighted combination of the tuning and growth efforts during learning.

The complete fitness function is:

$$F = H_C + C_W - E_L \tag{3}$$

Classification entropy is defined as:

$$H_C = \frac{-\sum_{i=1}^{K_f} p_i * log(p_i)}{log(K_f)} \tag{4}$$

where K_f is the number of neurons in the Kohonen layer at the end of learning, and p_i is the fraction of input cases for which the ith Kohonen-layer neuron *wins* (i.e. has the highest activation level). The numerator is thus a standard entropy calculation, while the denominator scales it by the maximum possible entropy for the given layer size.

The topological degree of the Kohonen layer, C_W, is the average correlation among the input vectors of neighboring neurons. Since weight vectors are normalized, the dot product of two vectors reflects the cosine of the angle between them, with a dot product of 1 denoting the maximum similarity.

$$C_W = \frac{1}{K_f S_N} \sum_{i=1}^{K_f} \sum_{j \in N(i)} \vec{w}_i \bullet \vec{w}_j \tag{5}$$

where $\vec{w}_i$ is the normalized input-weight vector for the ith neuron of the Kohonen layer, N(i) is the neighborhood of the ith neuron, and S_N is the size of each neighborhood. $S_N = 4$ (two neurons on each side) for the simulations reported herein. C_W quantifies the degree to which nearby Kohonen neurons serve as detectors for similar input cases.

E_L, the learning effort, consists of two terms related to growth (g) and tuning (t):

$$E_L = k_g E_{grow} + k_t E_{tune} \tag{6}$$

where $k_g = 0.1$ and $k_t = 0.033$ for the simulations reported below. Intuitively, the cost of neurogenesis plus synaptogenesis (during the learning phase) exceeds that of tuning existing synapses. The growth effort (E_{grow}) and tuning effort (E_{tune}) are calculated as follows:

$$E_{grow} = \Psi K_+ \tag{7}$$

where $\Psi \in [0, 1]$ is the average density of incoming connections to the Kohonen neurons, and K_+ is the number of Kohonen neurons added during learning.

Tuning effort involves connection density, learning rate, and tuning duration (based on the shrinking speed of the Kohonen neural neighborhood):

$$E_{tune} = \Psi\lambda\frac{R_{init}}{\mid \delta R \mid} \quad (8)$$

Thus, a highly-fit individual manages to a) separate input patterns into same-sized clusters, b) treat similar inputs similarly, and c) minimize learning effort (by moving much of topology generation into development).

5. RESULTS

For the simple, abstract scenario described above, a small population of 20 individuals run for 50 generations suffices to evoke the B.E.. Figure 6 shows the results of a typical run. In this and related figures, the key variables are the maximum fitness and the number of initial nodes (K_{init}), which are the solid lines in the upper and lower graphs, respectively. Evolution begins with a dramatic drop in K_{init} and a gradual fitness rise, indicating a low investment in development and a high reliance upon learning (primarily in terms of growing new nodes and connections). However, the 3 key learning parameters must also evolve, so their early values are often insufficient for attaining high values of H_C and C_W. Furthermore, since E_{grow} and E_{tune} incur fitness costs, evolution cannot simply ramp up the learning effort, but must do so cautiously and in concert with changes to the developmental strategy - so that the increased learning cost is offset by H_C and C_W rises.

Because H_C involves the scaling factor, $log(K_f)$, there is no immediate advantage to increasing K_{init} or K_f. In fact, it is easier to achieve a reasonably even distribution (of wins) over a small neuron population than over a large one. This appears to be a key factor in forcing K_{init} down during early generations. Since the K_+ neurons added during learning are wired up to match input cases, they tend to positively influence H_C, but can easily have negative effects upon C_W if tuning effort is low. However, the early reliance on low K_{init} plus high E_{grow} proves to be a successful combination in this and many other runs. This strategy represents B.E. phase I, wherein the best fitness stems from a substantial learning effort (via high E_{grow}) combined with less reliance on the genome and developmental.

To progress beyond B.E. phase I, evolution must find a superior combination of development and learning such that Kohonen nodes receive a) configurations of afferents that match the structure of the input cases, or b) *enough afferents* to tune to fit that structure. The latter solution occurs most often in BALKO runs. It involves increasing the number of developmental steps and finding a combination of logistic-map parameters such that continued iteration of the map gives many unique indices, and thus many synapses onto each Kohonen neuron from a diversity of input neurons.

The afferent density, Ψ, is also important to consider. It positively influences both learning-effort costs, so keeping it low is advantageous; but afferents are certainly necessary for proper input segregation (thus boosting H_C). Since Ψ is measured at the end of an individual's lifetime, a high synaptic-tuning effort can pay double dividends by both decreasing Ψ and adjusting the remaining afferents to better separate inputs.

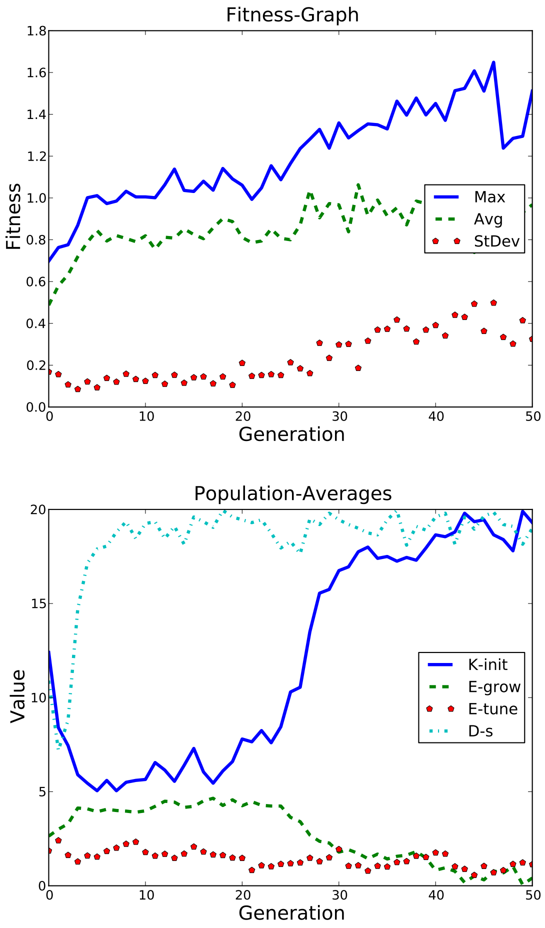

Figure 6: (Top) Fitness progression for a single run of BALKO using 20 input patterns. (Bottom) Population averages of 2 genes (K_{init} and D_S) and 2 effort measures (E_{grow} and E_{tune}) for the same run.

So the first 20-25 generations involve slowly increasing fitness and languishing K_{init}, until evolution works out the proper balance between K_{init} and other genes. Once accomplished, this spurs an often-dramatic rise in K_{init}, marking the transition to B.E. phase II. In this stage, the individual relies on a genome-governed developmental process to produce neurons and an initial connection topology. Learning effort is then devoted primarily to synaptic tuning, with very little late-life neuro- or synaptogenesis. These, more-costly, post-natal adaptive processes have been assimilated into the developmental scheme, a hallmark of B.E. phase II.

Overall, the total learning effort clearly decreases from phase I to II. However, the most noticeable shift is the reduction in E_{grow}, while E_{tune} increases slightly.

The same general trends in Figure 6 reoccur in the 20-run averages of Figure 7, but with less pronounced transitions between phases I and II of B.E. This occurs for at least 2 reasons: 1) the proper constellation of developmental and learning genes never arises, and thus the population remains highly dependent upon late-life neuro- and synaptogenesis

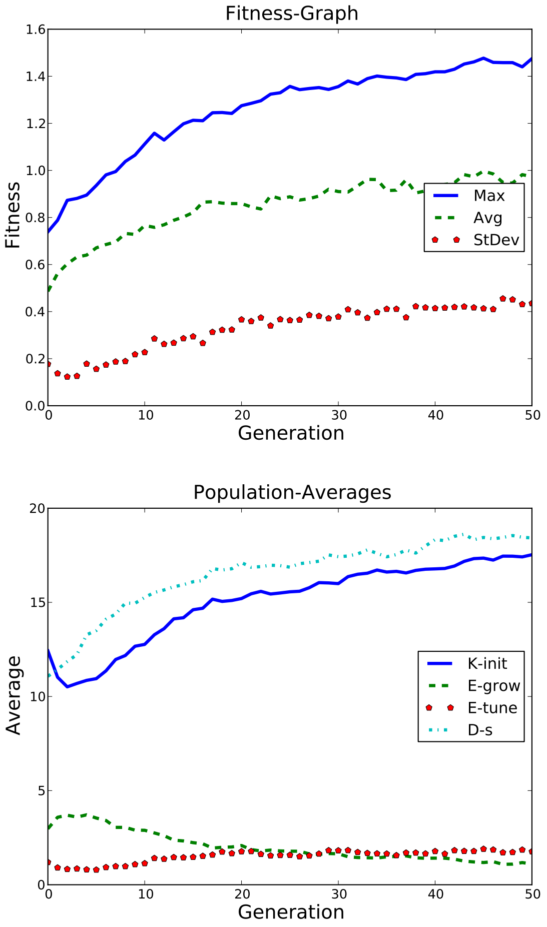

Figure 7: (Top) 20-run average fitness progression for a size-20 population using 20 input patterns. (Bottom) Population averages of 2 genes (K_{init} and D_S) and 2 effort measures (E_{grow} and E_{tune}) over the 20 runs.

to attain high fitness; K_{init} stays low through all 50 generations, or 2) pattern diversity is so high that it behoves individuals to generate a high number of Kohonen neurons during development, even before the other adaptive parameters have been found. Similarly, an increase of input patterns from 20 to 40 heightens the need for Kohonen-layer neurons to classify each case. With 40 patterns, a low K_{init} incurs a higher E_{grow} than with 20 patterns (even though the maximum cardinality of Kohonen neurons is 20 in all runs). Furthermore, with more patterns, the classification entropy, H_C, experiences less difficulty in scaling up to higher K_f values. Both of these factors facilitate a rise in K_{init}, although it still dips during the early generations (diagrams omitted).

In general, the similarity among input patterns strongly affects the degree of B.E.. If annealing runs too long, many patterns are similar or identical. This clearly eases the burden on the Kohonen network, allowing smaller neuron groups to achieve high entropy and topology values, without much E_{grow}. Hence, K_{init} can remain small and phase II of B.E. rarely occurs (data omitted). Conversely, for low pattern similarity phase I never occurs , as many innate Kohonen neurons give an immediate fitness advantage, as mentioned above.

Finally, the development algorithm was extended with a second set of evolved logistic-map parameters (Ω^2_{iter} and R^2_Ω) to add stochasticity into the choice of the *corresponding* input neuron for each Kohonen neuron (i.e. the $index_0$ argument in calls to *generate-indices*). This allows more intricate developmental patterns to arise. A similar B.E. occurs in these cases, although the onset of Phase II is often delayed due to the need to tune more developmental parameters before increasing K_{init}.

6. RELATED WORK

Though extremely difficult to test in biological lab settings, the B.E. is easily amenable to evolutionary computation. Hinton and Nowlan's [7] diabolically simple (yet elegant) set of simulations first illustrated the emergence of the B.E.. They showed that early learning helped guide evolution toward a difficult goal (B.E. phase I), but as the population approached the target, the flexible portions of the phenotype became hard-wired to the correct values, thus jettisoning the (costly) learning capabilities (B.E. phase II). Their model involved simple bit-string genotypes, which essentially doubled as phenotypes, so no development nor neural networks were involved. Ackley and Littman [1] took a big step by showing the B.E. in evolved pairs of interacting neural networks, one of which learned by back-propagation.

Upon adding learning (in neural networks) to his seminal work in cellular encoding, Gruau [5] (with Whitley) observed the confounding effects of development upon the B.E.. Downing [4] later extended the Hinton and Nowlan model to include an abstract developmental process based on a Turing machine (whose specifications were encoded in the genome). Those experiments showed the scaffolding effect that development can manifest to reduce the learning burden and thus support B.E. phase II.

Although researchers occasionally combine evolution, development and learning in their simulations (most recently [6]), few observe the B.E. unless a) they are looking for it, and b) their model's learned behaviors can also be ontogenetically formed.

7. DISCUSSION

This research represents an initial attempt to reconcile the Baldwin Effect with development in neural networks. It requires a reinterpretation of B.E. - in particular, the differences between phases I and II - as more quantitative than qualitative: cross-generational changes in the pre- and post-natal **rates** of neuro- and synaptogenesis can transfer adaptive effort from learning to development, with the latter more closely governed by the genome and less by environmental factors. As in [4], the effects of learning are not reverse-engineered into the genome, but strong learning (in this case, E_{grow}) *buys evolutionary time* until proper developmental scaffolding reduces the overall learning costs and elevates fitness to peak levels (see Figure 8).

In the runs reported above, the evolutionary search required to find proper parameters for the logistic map coarsely represents a process that we envision in nature: the develop-

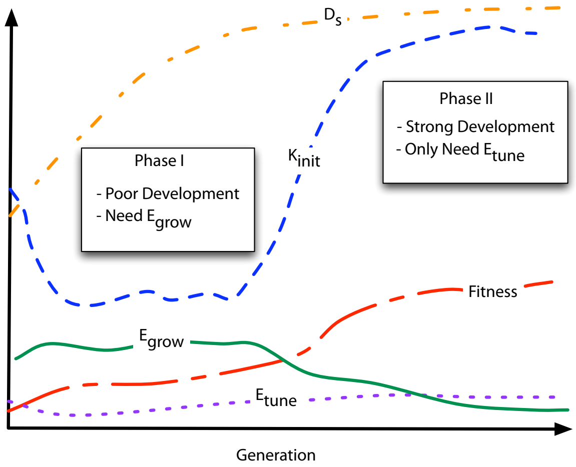

Figure 8: Summary of the Baldwin Effect in the BALKO experiments.

mental recipe gradually evolves to reduce some of the postnatal adaptive burden. As future work, we intend to experiment with other developmental schemes that maintain this non-linearity but exhibit more biological realism, including the effects of environment on development. This will force a generalization of our conceptual foundations from the Baldwin Effect to West-Eberhard's genetic accommodation theory [17], since, in a more complex model, many of the BALKO results could stem from environmentally induced developmental changes without actual genetic change.

Another important continuation of this work is the deployment of these networks in functioning agents, with fitness determined solely by their behavior. BALKO gleans fitness from the structure and primitive (segregating) behavior of the Kohonen net, under the assumption that both classification entropy and topological organization are important characteristics of neural networks. The most convincing examples of the Baldwin Effect, however, should span the full spectrum from genes to ethology.

Due to its extreme complexity, the Baldwinian puzzle will probably not buckle under to a single neuroscientific finding nor theoretical insight. However, the interplay between the biosciences and computation, particularly evolutionary computation, should continue to flesh out plausible interpretations of this intriguing hypothesis, with the proper version quite possibly filling sizeable gaps in our understanding of the evolution of intelligence. This is a rather difficult and esoteric pursuit, but one that artificial-life researchers are well equipped to handle, on both counts.

8. REFERENCES

[1] D. H. Ackley and M. L. Littman, *Interactions between learning and evolution*, in Artificial Life II, C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, eds., Reading, Massachusetts, 1992, Addison-Wesley, pp. 487–509.

[2] J. Bala, K. DeJong, J. Huang, H. Vafaie, and H. Wechsler, *Using learning to facilitate the evolution of features for recognizing visual concepts*, Evolutionary Computation, 4 (1996), pp. 297–311.

[3] J. M. Baldwin, *A new factor in evolution*, The American Naturalist, 30 (1896), pp. 441–451.

[4] K. L. Downing, *Development and the Baldwin effect*, Artificial Life, 10 (2004), pp. 39–63.

[5] F. Gruau and D. Whitley, *Adding learning to the cellular development of neural networks*, Evolutionary Computation, 1 (1993), pp. 213–233.

[6] S. Harding, J. Miller, and W. Banzhaf, *Evolution, development and learning using self-modifying cartesian genetic programming*, in Proceedings of the 11th Genetic and Evolutionary Computation Conference, D. T. et al., ed., Montreal, Canada, ACM, 2009, pp. 699–706.

[7] G. E. Hinton and S. J. Nowlan, *How learning can guide evolution*, Complex Systems, 1 (1987), pp. 495–502.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The MIT Press, Cambridge, MA, 2 ed., 1992.

[9] J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proceedings of the National Academy of Sciences, 79 (1982), pp. 2554–2558.

[10] E. Kandel, J. Schwartz, and T. Jessell, *Principles of Neural Science*, McGraw-Hill, New York, NY, 2000.

[11] T. Kohonen, *Self-Organizing Maps*, Springer, Berlin, 2001.

[12] G. Mayley, *Landscapes, learning costs and genetic assimilation*, Evolutionary Computation, 4 (1996), pp. 213–234.

[13] D. Sanes, T. Reh, and W. Harris, *Development of the Nervous System*, Elsevier Academic Press, Burlington, MA, 2006.

[14] T. J. Shors, *Saving new brain cells*, Scientific American, 300 (2009), pp. 40–48.

[15] P. Turney, L. D. Whitley, and R. W. Anderson, *Introduction to the special issue: Evolution, learning, and instinct: 100 years of the Baldwin effect*, Evolutionary Computation, 4 (1997), pp. iv–viii.

[16] B. Weber and D. Depew, eds., *Evolution and Learning: The Baldwin Effect Reconsidered*, The MIT Press, Cambridge, MA, 2003.

[17] M. J. West-Eberhard, *Developmental Plasticity and Evolution*, Oxford University Press, New York, NY, 2003.