IEEE Transactions on Evolutionary Computation, Vol. 11, No. 2, April 2007

Developing Complex Systems Using Evolved Pattern Generators

Vinod K. Valsalam, James A. Bednar, and Risto Miikkulainen

***Abstract*— Self-organization of connection patterns within brain areas of animals begins prenatally, and has been shown to depend on internally generated patterns of neural activity. The neural structures continue to develop postnatally through externally driven patterns, when the sensory systems are exposed to stimuli from the environment. The internally generated patterns have been proposed to give the neural system an appropriate bias so that it can learn reliably from complex environmental stimuli. This paper evaluates the hypothesis that complex artificial learning systems can benefit from a similar approach, consisting of initial training with patterns from an evolved pattern generator, followed by training with the actual training set. To test this hypothesis, competitive learning networks were trained for recognizing handwritten digits. The results demonstrate how the approach can improve learning performance by discovering the appropriate initial weight biases, thereby compensating for weaknesses of the learning algorithm. Because of the smaller evolutionary search space, this approach was also found to require much fewer generations than direct evolution of network weights. Since discovering the right biases efficiently is critical for solving large-scale problems with learning, these results suggest that internal training pattern generation is an effective method for constructing complex systems.**

***Index Terms*— Artificial neural networks, competitive learning, complex systems, evolutionary computation, pattern generator, self-organization, spontaneous activity**

I. Introduction

THE tradeoff between bias and variance is a well-known issue in machine learning [28], [85]. Given a finite set of example inputs and outputs (the training set), a learning system needs to construct a mapping that produces correct outputs for new examples (the test set). There is usually a large, or even infinite, number of possible mappings consistent with the training set, providing different outputs for the same test inputs. Which mapping will be selected is determined by the bias of the learner. The best results are obtained if the bias matches the problem and is strong [34]. That way, the outputs for new examples are likely to be correct, and the same mapping is selected with different training sets and even when the training examples are noisy. Such a learner has a low variance in its behavior.

Unfortunately, it is usually not clear what the right bias is, making it necessary to make the bias weaker. Which mapping is selected then depends more on the training data. As a result, the variance is increased: The selection of the mapping becomes unpredictable, depending on which examples were included in the training set and the noise in those examples. Choosing an appropriate point in the bias/variance tradeoff is therefore critical for building large-scale systems. The choice depends on how much is known about the problem in advance, and whether such knowledge can be formulated as an appropriate bias for the learning algorithm. In fact, for many complex problems, establishing such a bias could be as hard as solving the problem itself [28].

This paper is inspired by how nature handles this bias/variance tradeoff. Complex neural systems are constructed in a developmental process that combines both genetic and environmental information, as opposed to a pure hardwiring of neural connections (strong bias, small variance) or a pure environment-induced learning process (weak bias, large variance). Internally generated spontaneous activity in many cortical and subcortical sensory areas before birth (see [62], [70], [93] for reviews) may be one of the developmental mechanisms by which nature deals with this bias/variance tradeoff. These activity patterns may be specified genetically as pattern generators, allowing the system to use the same activity-dependent learning mechanisms both prenatally to establish bias, as well as postnatally to adapt to the environment. These observations suggest that learning driven by both internal and external inputs can be used to build structures that would be too complex to determine directly genetically and too fragile to learn directly from highly variable external inputs [4], [5].

The main hypothesis studied in this paper is that artificial learning systems may achieve the same benefits as biological learning systems using a similar approach for establishing bias. This approach involves computationally evolving a pattern generator and using the generated patterns in addition to real-world data for training. Learning based on pattern generation has already been successful in explaining computationally how biological mechanisms for orientation processing and face detection may develop [5], [54]. The advantages of this approach for building complex systems are studied systematically in this paper.

In order to test this hypothesis and show how a learning algorithm can benefit from this method, a winner-take-all competitive learning neural network architecture was studied in the task of recognizing handwritten digits. Experiments were devised to evaluate the relative merits of three learning approaches: (1) competitive learning alone on a set of training data, (2) evolving (i.e., hardcoding through simulated evolution) the network connection weights directly based on the same training data, and (3) competitive learning on patterns produced by an evolved pattern generator, followed by competitive learning on the real training data. Best possible performance in the domain was not the main goal in these

V. Valsalam and R. Miikkulainen are with the Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: vkv@cs.utexas.edu; risto@cs.utexas.edu).

J. Bednar is with the Institute for Adaptive and Neural Computation, The University of Edinburgh, 5 Forrest Hill, Edinburgh, EH1 2QL UK (e-mail: jbednar@inf.ed.ac.uk).

experiments; rather, the aim was to understand how these three learning processes differ and how they might scale up in more complex tasks.

The results show that competitive learning alone is weaker than the other two methods. Both direct evolution and pattern generation achieve good classification accuracy, but pattern generation reaches the same level of performance in many fewer generations. These results suggest that pattern generation may indeed help in constructing complex artificial systems. This paper presents a proof of concept of this approach, paving way for possible applications in future work.

The remainder of the paper is organized as follows. Section II reviews biological and computational background and related research on learning and evolution. The general hypothesis studied in this paper, i.e., that prenatal training based on evolved pattern generators is an effective way to build complex artificial learning systems, is formulated in Section III. The learning architecture and the algorithms used to evaluate the hypothesis are presented in Section IV, and experimental results on the handwritten digit recognition task in Section V. In order to understand the benefits of the pattern generation approach better, a simpler task of line categorization is analyzed in Section VI. Finally, discussions and possible directions for future work are presented in Section VII.

II. Background and Related Work

In the following subsections, the biological motivation for pattern-generation-based prenatal learning is reviewed. This review is not meant to be comprehensive, but intended to serve as the inspiration for formulating the computational approach presented in this paper. Studies verifying the advantage of combining learning with evolution in computation, and previous computational work involving pattern generators, are briefly reviewed as well.

A. Biological Motivation

Many researchers have argued that much of the structure of the primary visual cortex in mammals develops through self-organization of input connections from the thalamus, driven by visual experience (see e.g. [57], [73] for reviews). A number of classic experiments by Hubel, Wiesel and others showed that altering the visual environment, especially during a critical period of early life, can dramatically change the organization of the visual cortex [40], [41]. For instance, if kittens are raised in environments containing a single orientation during the critical period, a disproportionately large number of their primary visual cortex neurons become responsive to that orientation compared to the orthogonal orientation [8], [71]. Even in normal adult animals, the distribution of orientation preferences is slightly biased toward horizontal and vertical contours [14], [20]. Such a bias would be expected if the neurons learned orientation selectivity from typical environments, which have a similar orientation bias [83]. Conversely, kittens who were raised without patterned visual experience at all, e.g. by suturing their eyelids shut, have few orientation-selective neurons in their visual cortex as an adult [9], [22]. Thus, visual experience can influence how the neural circuitry for orientation processing develops.

However, relying on environmental input alone has two inherent weaknesses: (1) Self-organization takes time, and the animal would not be able to act on visual input until the process is almost complete. (2) The self-organized structure depends critically on the specific input patterns available: if the visual environment is variable, the organism may not develop predictably, and what the learning algorithm discovers may not be the information most relevant to the organism. Therefore, the neonatal visual system needs to have the proper bias to address these issues.

There is significant evidence that such a bias exists as genetically determined visual cortex structures [31], [44]. For example, orientation-selective cells are found in newborn kittens and ferrets even before they open their eyes; an altered environment has only limited effects [9], [16], [39]. Psychological studies further suggest that human newborns can already discriminate between patterns based on orientation [78], [79]. Experiments also show that large-scale orientation processing structures exist prior to visual experience, and that these cortical structures have many of the same features found in adults [17], [22], [29]. Prenatal or pre-visual ocular dominance structures (corresponding to binocular vision processing) have also been observed in animal brains [23], [38], [65]. Thus, although environmental input clearly influences visual cortex structure, many aspects of the visual system appear to be constructed from a specific blueprint encoded in the genome.

What are the biological mechanisms responsible for establishing such genetically hardcoded biases, while also allowing the system to learn and adapt based on the environment? The large-scale structures of the brain, such as the division into different brain areas, are constructed primarily through chemical gradients [56], [66], [89], [91] (see [31] for a review). These gradients direct the growing connections to a general location on the cortical sheet. The gradients are largely unaffected by environmental stimuli, making the bias very strong. Incorporating environmental information into this process would be difficult, requiring a transduction mechanism between an environmental stimulus and the developmental hardware.

On the other hand, at the level of individual neurons and connections between small groups, sensory systems act as just such a transduction mechanism. In a sensory system, patterns in the environment are represented as patterns in neural activity, and these patterns in turn change how the orientation, ocular dominance, and similar map-level organization in the cortex develop (as discussed above). At this level, the question becomes how *genetic* cues could be expressed to give the system its bias. First, the system is structured to utilize information in input activity; second, the amount of information necessary to specify individual connections may be too large to store genetically.

The recent discovery of spontaneous activation provides an important clue to how such genetic bias is expressed: Much of the neural activity in developing sensory systems is not caused by the external environment, but generated internally in many cortical and subcortical sensory areas, such as the

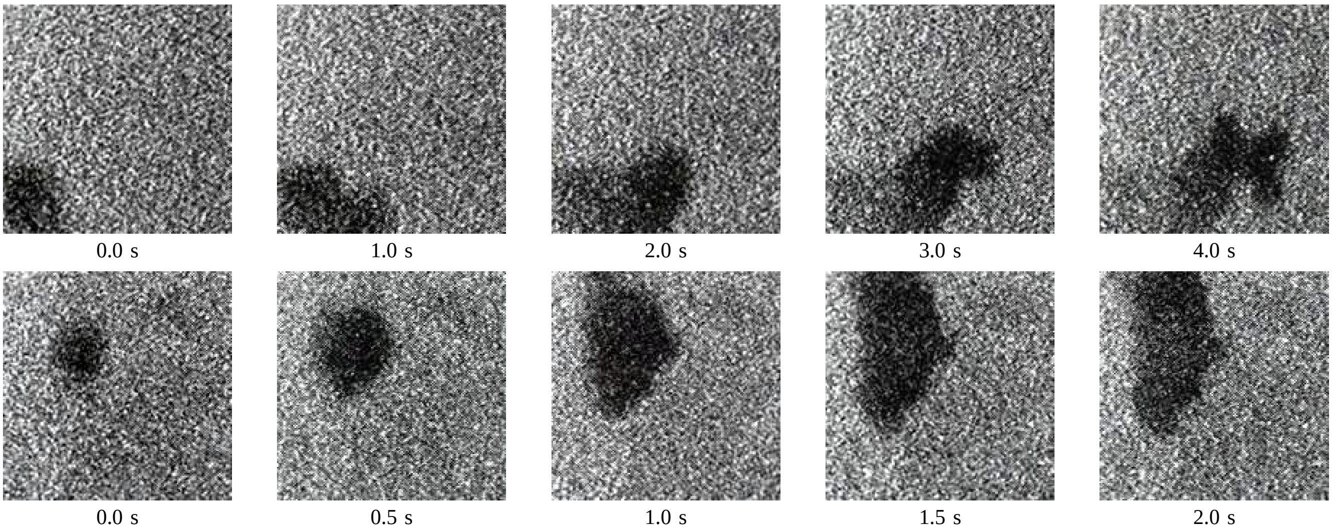


Fig. 1. *Retinal wave patterns.* Each of the frames shows a calcium concentration image of approximately 1 mm^2 of newborn ferret retina, measuring how active the retinal cells are. Dark gray indicates areas of increased activity. This activity is spontaneous (internally generated), because the photoreceptors have not yet developed at this time. From left to right, the frames on the top row form a 4-second sequence showing the start and expansion of a wave of activity. The bottom row shows a similar wave 30 seconds later. Such activity could be responsible for biasing the visual system before learning begins from environmental experience. Reprinted with permission from [26], copyright 1996 by the American Association for the Advancement of Science.

visual cortex, the retina, the auditory system, and the spinal cord ([26], [48], [53], [63], [92], [94]; see [62], [70], [93] for reviews). Such activity may serve several developmental functions, such as guiding cell migration, cortical innervation, cortical patterning, and others that are still unknown [95]. However, one possible role for it is to express a genetic bias within a system that is designed to learn from the environment [19], [49], [50], [67], [72], [74]. The genetic information is represented in the same way as environmental information at the neural level: as patterns of activity in the input seen by a brain area. In this way, "hardwiring" may actually be learned. The genome thus needs to specify only a simple pattern generator, a mechanism capable of producing activity patterns, rather than specifying billions of individual connections.

The bias constructed through such prenatal self-organization can guarantee that each organism has a rudimentary level of performance from the start and that initial development does not depend solely on the details of the external environment, while retaining the flexibility of the neural system to adapt to environmental input. Thus, internally generated patterns can preserve the benefits of a blueprint, within a learning system capable of much higher complexity and performance. Evolution has therefore determined a point in the bias/variance tradeoff that allows constructing a reliable but flexible system by combining genetic and environmental information.

Although spontaneous activity can arise at all levels of the developing nervous system, retinal waves and ponto-geniculo-occipital (PGO) waves are the best known, and can be used to illustrate the general properties of internally generated patterns.

1) Retinal Waves: In the developing retina of e.g. cats and ferrets, internally generated activity occurs as intermittent, local waves across groups of ganglion cells [53], [77], [92]. Fig. 1 shows such an activity in the retina of a newborn ferret. The waves begin before photoreceptors have developed [49], so they cannot result from visual input. Instead, they arise from spontaneous recurrent activity in networks of developing amacrine cells that provide input to the ganglion cells [13], [26], [74]. Like visual images, these waves are locally coherent in space and time and thus they could act as visual-like training input for the developing visual cortex [72].

Recent experiments have focused on whether spontaneous activity is merely permissive for development, perhaps by keeping newly formed connections alive until visual input occurs, or whether it is truly instructive, determining how the structures develop [15], [21], [45], [55], [64], [81], [82], [84]. One such experiment showing an instructive role involved introducing artificially correlated activity into the developing visual pathway of ferrets. As a result, the number of orientation selective cells in the primary visual cortex was significantly reduced [90]. This result shows that spontaneous activity cannot be only permissive; it has a specific instructional role at least in shaping receptive field tuning properties of cortical neurons.

2) Ponto-Geniculo-Occipital Waves: Ponto-geniculo-occipital (PGO) waves have been shown to be the hallmark of rapid-eye-movement (REM) sleep in cats, ferrets, monkeys, and humans (see [80] for a review). REM sleep has long been believed to be important for development, for two reasons: Developing mammalian embryos spend a large percentage of their time in states that look much like adult REM sleep, and the duration of REM sleep is strongly correlated with how plastic the neural system is, both over development and across different species [42], [67], [76].

During and just before REM sleep, PGO waves originate in the brainstem and travel to the LGN, many areas of the visual cortex, and a variety of subcortical areas (see [11] for a review). In adults, PGO waves are strongly correlated with eye movements and with vivid visual imagery in dreams, suggesting that they activate the visual system as if they were visual inputs [50]. Experimental studies also suggest that PGO waves are under genetic control: They elicit different activity patterns in different species [24], and the eye movement patterns that are associated with PGO waves are more similar in identical twins than in unrelated age-matched subjects [18]. Thus, PGO waves are another possible source for genetically controlled training patterns for the visual system.

In sum, the hypothesis that internally generated patterns play a role in self-organization of the visual system is well established biologically. This idea has already inspired computational explanations of developmental phenomena, as will be reviewed next.

B. Computational Modeling of Pattern Generators

The pattern-generation hypothesis was previously tested using a computational map model of the visual cortex called HLISSOM [5]–[7], [54]. Two developmental phenomena were studied: (1) How orientation-selective responses develop in the visual cortex prenatally and postnatally, and (2) how human newborns come to prefer face-like visual input prenatally and how these preferences change in early life.

The HLISSOM orientation model resulted in detailed connectivity patterns that match known biological orientation processing circuitry in animals. Training the model prenatally with three-dot input patterns in turn caused it to respond preferentially to pictures of faces, and these preferences changed as they do in infants in later training with visual images. The experiments with HLISSOM therefore elucidate computationally how self-organization based on internal pattern generation can account for the observed biological structures, resulting in species-specific biases such as face preferences. Building on the prior work with HLISSOM, the goal of this paper is to study the pattern generator approach as a machine learning technique for constructing complex artificial learning systems. Instead of designing the generator by hand, generators are evolved computationally to provide a suitable bias for final learning in the actual task.

C. Computational Merging of Learning and Evolution

A related idea that has been explored extensively by several researchers is combining evolution with learning from the environment. In many such approaches, initial connection weights are evolved for a neural network that later adapts them through learning. Evolution selects individuals with weight patterns that have the capacity to learn good performance, rather than individuals with good performance at birth [37], [58]. In other words, learning establishes exploration in the local vicinity of the genetically specified solution. Evolution can search a large search space efficiently, leaving local optimization to the learning algorithm rather than having to find the correct weight patterns directly. This process results in the Baldwin effect [2]: learning influences evolution even though acquired characteristics are not inherited.

A second approach is to use genetic algorithms to select weight values for “teaching units” in a network rather than the actual network weights [59], [60]. These units specify the target outputs for training, which are not known in advance. In essence, the genetic algorithm transforms the reinforcement learning problem into a supervised learning problem, for which strong learning algorithms exist. Along the same lines, the outputs of existing well-performing networks in the population can be used as targets for supervised training of newly evolved networks [51], [52]. Such training indirectly selects both for good performance and the ability to train other networks.

In each of the above cases the genetic algorithm is used to select specific connection weights, either those that are later tuned by learning or those that generate the target outputs. Such a direct genetic encoding limits the network size and complexity, because the search space of all possible connection weights in a fully connected network grows exponentially with the number of neurons [61]. Thus it may take a prohibitively long time for the genetic algorithm to explore the search space. A third approach aims at solving this problem by drawing motivation from nature, where the size of the genome is several orders of magnitude smaller than the number of connections it is coding [10], [43], [47], [75], [87]: the actual neural connections are established in a developmental process driven by the genome. These *developmental* encoding schemes map genotype to a phenotype [12], [32], [33], [61] through a mechanism that imitates how biological organisms develop. The genome encodes a process for constructing an individual through interaction with its environment, rather than simply specifying the individual itself.

The pattern-generation hypothesis tested in this paper is based on the developmental approach as well: the genome encodes a pattern generator which is then used together with the environment to construct the individual. This approach is described in more detail in the next section.

III. Hypothesis

The simulations with HLISSOM demonstrated how genetic and environmental influences may interact in the developing visual system. A more general hypothesis is evaluated in this paper: Learning from generated patterns followed by learning from the actual training data is a general-purpose problem-solving approach that can be used to construct complex artificial systems effectively. Following the biological analogy, the two phases are called prenatal and postnatal learning in this paper. The following subsections develop this approach and discuss how it facilitates designing an appropriate bias for complex systems. The hypothesis is tested experimentally and the results analyzed in the later sections.

A. Approach

In the most straightforward approach, the pattern generator can be designed specifically for the task, as was done with HLISSOM. Such a generator allows the engineer to express a desired bias without having to hard-code it into a particular, inflexible architecture. Biasing the learning system in this way may allow it to solve problems that would otherwise be difficult to solve. For example, simple patterns can be learned before real data, which establishes the necessary bias by moving the starting point closer to the solution, thus avoiding local optima in the search space of solutions [25].

More significantly, the pattern generator can be constructed automatically using evolutionary algorithms (EA). In this approach, detailed domain-specific knowledge necessary to design the generator by hand is not needed. For instance, studying real faces may lead one to suggest that a three-dot configuration would be a good training pattern to bootstrap a face detector as was done with HLISSOM. However, often such knowledge can only be obtained through trial and error, and it would be desirable to have an algorithm to do it automatically. Indeed, the learning system, the pattern generator, and the EA together can be considered a single general-purpose adaptive algorithm.

This approach consists of three parts: (1) evolving the pattern generator, (2) prenatal learning from generated patterns, and (3) postnatal learning from actual training data. As in nature, the combination of learning and evolution in this approach represents a balance between adaptation at different time scales (i.e., determines a proper tradeoff between bias and variance; Section I). Short-term learning (prenatal followed by postnatal learning) allows an individual learner to become well suited for the particular tasks on which it is tested. Long-term adaptation (i.e., selection by the EA), along with prenatal learning ensures that postnatal learning is properly biased, avoiding getting stuck in local optima, and therefore more likely to succeed.

B. Constructing Complex Systems

The bias/variance discussion in Section I suggests that choosing the right bias for learning systems is both important and difficult. As an illustrative instance of such a system, consider the task of learning to recognize two-dimensional objects from training examples. This task is difficult for several reasons. First, the number of objects (target classes for classification) that the system needs to be able to distinguish from each other can be very large. Second, the system needs to be able to identify these objects in a transformation-invariant manner, i.e., regardless of their location, orientation, and scale. Third, a host of other variables such as noise, occlusions, and background changes can further distort the input. Let us next see how combining learning with evolved pattern generators facilitates designing the appropriate bias to make learning such a complex system possible.

1) Expressing bias in the input space: In the pattern-generator approach, domain-specific knowledge is specified through the pattern generator, i.e., in the input space, rather than in the parameter space of the learner. This approach is effective because knowledge of the problem is usually available in the input space. For example, in the two-dimensional object recognition domain, we may be told that all objects have straight edges, thus restricting the number of target classes the system needs to identify. In this case, we may choose to encode a pattern generator that generates straight line segments or combinations of them as prenatal training patterns. Similarly, in constructing a face-recognition system, three-dot patterns may be used to bias the system towards the essential facial features (i.e. the eyes and the mouth) before training the system with images of real faces. Learning such patterns prenatally gives the system the required bias for successfully learning more complex shapes in the postnatal phase. In contrast, it is not immediately clear how such knowledge could be expressed as constraints on the parameters of the learning system.

2) Optimizing bias by evolutionary search: Postnatal training with actual training examples, when combined with evolution, further refines the bias that was initially encoded in the pattern generator. In the example above, evolution might discover that prenatal training patterns containing intersecting line segments leads to better postnatal training performance in identifying polygons, and might consequently include such features in the generated patterns. The features could appear in different locations, orientations, and scales, allowing the learning system to adapt to various geometric transformations. The result is a pattern generator that is able to express a more specific bias, making postnatal learning even more effective. Extracting the features needed for expressing such a bias by manually examining the training set is generally very difficult.

3) Compact encoding of bias in generator encoding: Pattern generators can generally be specified in a highly compact manner. For instance, line segments can be specified in terms of their end points and Gaussian blobs in terms of location and width. Such compact encodings facilitate evolutionary search, making it possible to arrive at different input patterns by few simple manipulations. Efficient data representations of this type are an essential part of designing the right bias for complex systems [1], [28].

4) Transforming bias to parameter space through learning: The bias must be specified in terms of the learning system parameters before learning from actual training examples can begin. This transformation is achieved through prenatal learning. Depending on the sophistication and complexity of the learning algorithm, it is possible to synthesize biases having complex and emergent properties in the parameter space from simple input patterns. For example, using the “noisy disk” model of retinal waves as prenatal training patterns, HLISSOM experiments demonstrated how complex orientation processing circuitry similar to that seen in the visual cortex of animals can develop [5], [54]. Thus, prenatal training makes it possible to express the bias in complex parameter spaces, which may be difficult to achieve by other means. Moreover, the bias established by prenatal learning is likely to be robust against distortions in the input, because the learning system becomes better tuned to the relevant features in the input.

The above four properties make it possible to establish the proper bias efficiently. As a result, the pattern-generator approach may allow constructing complex systems that would be infeasible to build by applying postnatal learning directly. Instead of pattern generation, complex systems may also be built using more traditional approaches, such as evolving them directly. The obvious disadvantage of the direct EA approach is that it requires optimizing a large number of parameters, which could take prohibitively long. In contrast, pattern generators are evolved in the smaller space of few generator parameters, and such a system is therefore expected to find a good solution in fewer generations. Thus, the approach should be able to perform better than direct EA in constructing

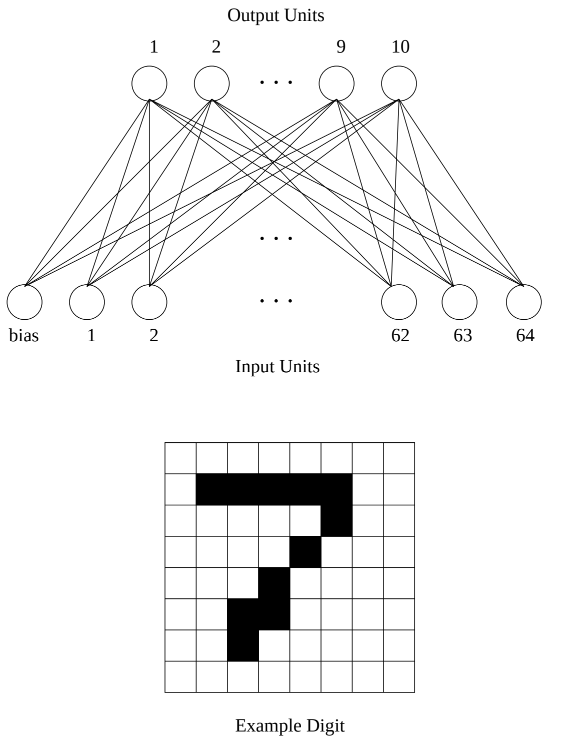


Fig. 2. *The architecture of the competitive learning network.* The binary activations from the input pattern consisting of 64 pixels are fed to the input units of the network, which also contains a bias unit. The 10 output units each correspond to a classification of the input as one of the 10 digits; the one with the highest activation is chosen as the answer of the network. During training, the weights to this unit are adjusted towards the input pattern, making that unit more likely to win similar patterns in the future.

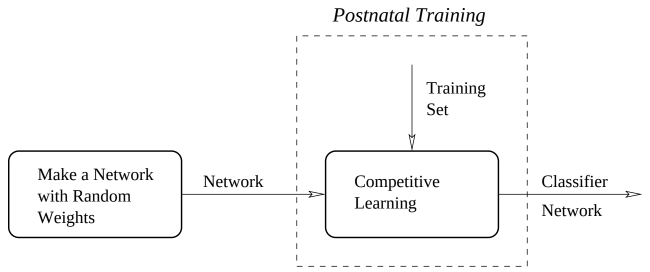


Fig. 3. *The postnatal learning approach.* Inducer1 produces a classifier network by performing competitive learning on the set of training examples. This method corresponds to postnatal learning (there is no prenatal learning phase).

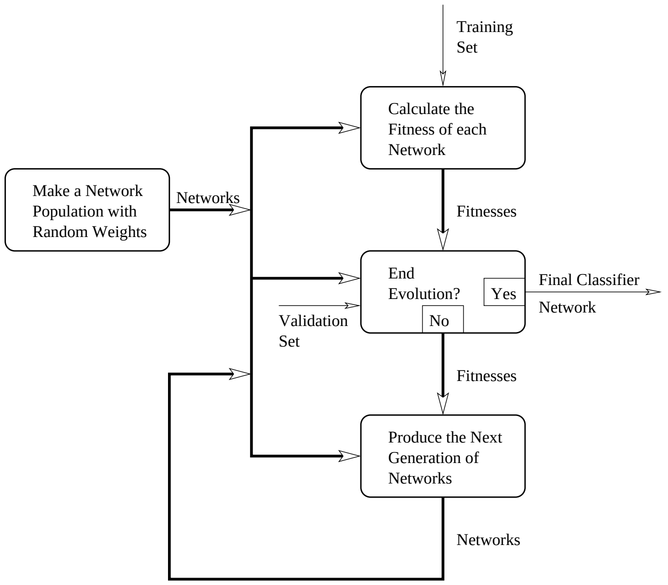

Training
Set
Calculate the Fitness of each Network
Make a Network Population with Random Weights
Networks
Fitnesses
End Evolution?
Yes
No
Final Classifier
Network
Validation
Set
Fitnesses
Produce the Next Generation of Networks
Networks

Fig. 4. *The direct evolution approach.* Inducer2 produces a classifier network by evolving the weights of a network that has the same architecture as that produced by Inducer1. This method corresponds to hardwiring the behavior genetically (there is no learning). The thick lines indicate the entire population of networks.

complex systems.

IV. Testing the Hypothesis

The hypothesis is tested in this paper on the task of constructing a single-layer artificial neural network to identify the handwritten digits 0 to 9. The network receives the digits as its input and produces the classification of each digit as its output. Classification accuracy and learning effort is compared for postnatal learning, direct evolution, and pattern generator approaches. These experiments are not designed to produce the best possible performance for classification of handwritten digits, but to verify the hypothesis that an appropriate bias for learning can be established using the pattern-generator approach.

A. Competitive Learning

The hypothesis is tested using competitive learning [35], [69] as the learning algorithm. Even though other algorithms may be more powerful in classification tasks in general, competitive learning is a good choice for four reasons: (1) it is a well known abstraction of biological learning, based on Hebbian adaptation of synaptic efficacies and winner-take-all competition [36], and a good surrogate for a whole class of learning algorithms; (2) it is sensitive to initial weight settings, i.e., prenatal training is likely to have a significant effect; (3) it is relatively simple, so that analyzing and understanding this effect is possible; and (4) it is a self-organizing, unsupervised algorithm, which makes the pattern generator design simpler by not having to produce targets for the inputs it generates for prenatal learning.

The digits are written in an 8×8 grid of pixels (Fig. 2). The inputs to the network consist of the binary activations at the 64 grid locations and a bias unit. The network has 10 outputs, one for each of the 10 digits to be recognized. Each output unit is connected directly to each of the inputs (including the bias).

Learning starts by initializing the network connection weights w_{ij} between an input unit i and an output unit j randomly, and normalizing such that the squares of the weights to each output unit sum to one:

$$w_{ij} = \frac{w_{ij}}{\sqrt{\sum_u w_{uj}^2}}. \tag{1}$$

When the network is presented with an input pattern, each output unit j computes the weighted sum s_j of its input

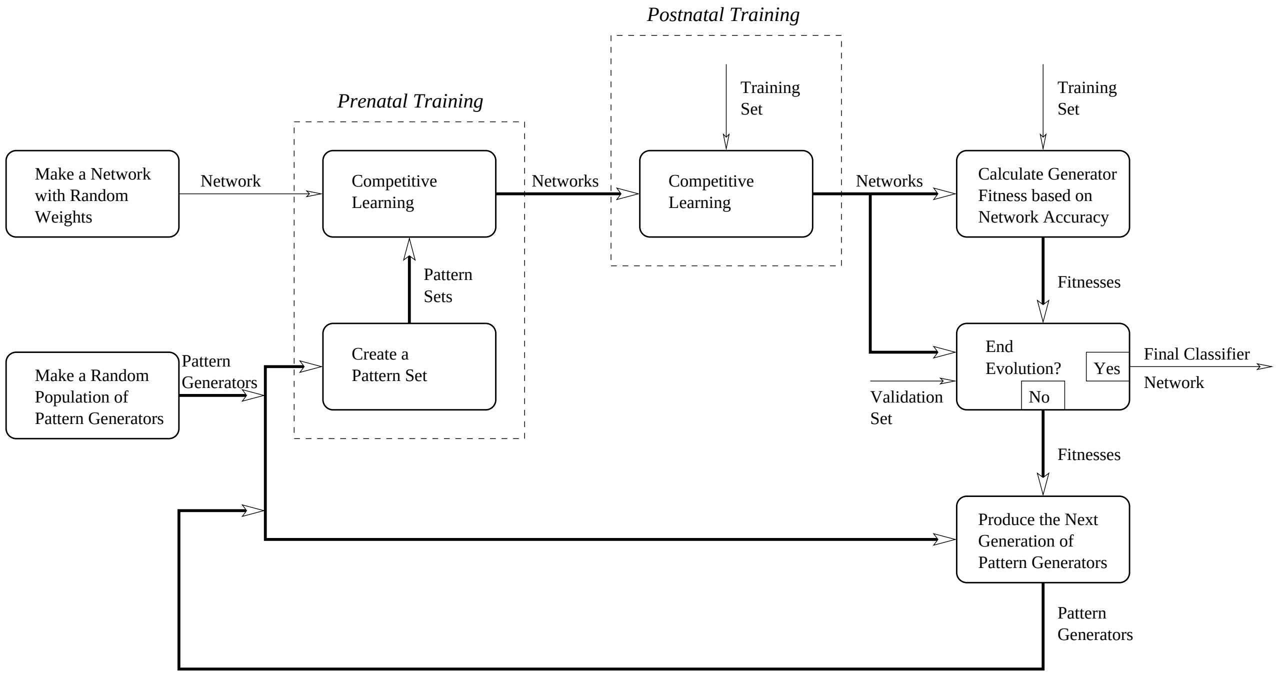


Fig. 5. *The pattern generator approach.* Inducer3 evolves pattern generators in its main computational loop. The patterns produced by each generator train a competitive learning network in the prenatal training phase. This phase is followed by a postnatal competitive learning phase where the network is trained further with a training pattern set, like Inducer1. The evolution run produces two results: the champion pattern generator and the classifier network trained using it. The thick lines indicate the entire population of pattern generators or corresponding networks.

activations x_i:

$$s_j = \sum_i w_{ij} x_i. \tag{2}$$

The output unit with the highest sum is the winner for that pattern. The weights to this unit v are then updated as

$$w_{iv}(t+1) = w_{iv}(t) + \eta(x_i - w_{iv}(t)), \tag{3}$$

where η is the learning rate. After the update, the weights to this unit are again normalized such that their squares sum to one. This process constitutes a basic competitive learning method that is at the core of many unsupervised learning algorithms [3], [27], [30], [46], [88].

B. Experiments

To evaluate the benefits of training with generated patterns, three different ways of constructing the neural network are compared.

1) Inducer1: First, a network is trained using competitive learning alone (Fig. 3). This process involves initializing the network with random weights and training it using a set of examples until its weights converge. In other words, learning is carried out without any intentional bias. Using a biological analogy, this method corresponds to an organism whose learning is entirely postnatal, without any specific genetically determined biases.

2) Inducer2: In the second method (Fig. 4), the connection weights of the network are evolved directly; there is no competitive learning phase at all. However, the network architecture is the same as in Inducer1, and classification of examples is done by competition between the output units as in Inducer1. A population of networks is initialized with random weights at the beginning of evolution. In any given generation, the classification accuracy of each network on the training set is measured to compute its fitness, after which the next generation of networks is produced. The evolution is terminated once the fitness on a validation set begins to level off, and the network at that point is output as the classifier. The encoding and evolution of the network weights are discussed in detail in Section IV-C. This method demonstrates a direct evolutionary algorithm in constructing complex systems; in biological terms, it represents an organism whose behavior is entirely genetically determined.

3) Inducer3: Constructing the third classifier network involves evolving a pattern generator (Fig. 5). The network architecture is the same as in Inducer1. The process begins by initializing a population of pattern generators with random parameter values. In any given generation of evolution, a network initialized with random weights is produced for each pattern generator in the population. This network is then trained (using competitive learning) with a set of patterns produced by the pattern generator during the prenatal training phase. After the prenatal training is complete, the resulting network is trained on a set of examples during the postnatal training phase. After postnatal training, the fitness of the pattern generator is calculated based on how well the final network performs on the training set. After all pattern generators in the population have been evaluated in this manner, the next generation of pattern generators is created. The evolution is terminated once the fitness on a validation set begins to level off, and the network and the corresponding pattern generator at that point are output as the result of evolution. The encoding and evolution of the pattern generator are discussed in detail in Section IV-D. This method implements the approach outlined in Section III-A for constructing a complex system based on evolved pattern generators; biologically, it corresponds to determining the bias of the learning system genetically.

The expected outcome of these comparisons is that the environmentally driven learner (Inducer1) is likely to get stuck in suboptimal local optima, because it will start far from the desired solution, without any bias toward it. The direct EA (Inducer2), on the other hand, would require a prohibitively large number of iterations to produce a successful classifier network, because it has to search in an extremely high-dimensional space of network weights (650 weights, as explained below in Section IV-C). In contrast, the pattern-generator-driven system (Inducer3) should be able to discover a solution quickly compared to the direct EA approach because it only needs to evolve a small number of parameters of the generator (10 parameters, as explained below in Section IV-D).

The competitive learning method used in Inducer1 was already described completely in Section IV-A; the evolutionary approach of Inducer2 and Inducer3 will be described in the next two subsections.

C. *Evolving Networks Directly*

In order to evolve the weights of the network directly, in the Inducer2 approach each gene is coded as an array of 65 weight values (corresponding to 64 inputs and 1 bias) associated with an output unit. The weights are floating point values between 0 (inclusive) and a specified maximum bound of 1 (exclusive). The genes for all the output units are concatenated to form a chromosome, which constitutes an individual in the population. Each chromosome therefore consists of 10 genes, one for each output unit of the network. This encoding contributes a total of 650 weight values for evolution to search through.

The weights are mutated by applying Gaussian perturbations on the floating-point values. The standard deviation of the perturbation is calculated as the product of a “mutation factor” and the maximum value allowed for weights. If the mutated value lies outside the allowed legal range of values, the mutation is ignored and the weight is not changed. The probability of mutation is controlled by a “mutation rate” evolution parameter.

For each individual chosen for mating, a partner is selected randomly from the population and an offspring is created through uniform crossover at two levels: individual weight values and whole genes. That is, genes and weight fields in the genes of the parent are randomly selected and replaced by the corresponding piece of the genome from the partner to produce the offspring. This process is controlled by a “mating rate” parameter for gene fields and another one for whole genes.

In every generation, all individuals are given a chance to improve their fitness either through mutation or by mating with a randomly chosen individual. If the fitness improves as a result of mutation or mating, then the offspring replaces the parent in the population for the next generation; otherwise the parent is retained in the population, keeping the population size constant.

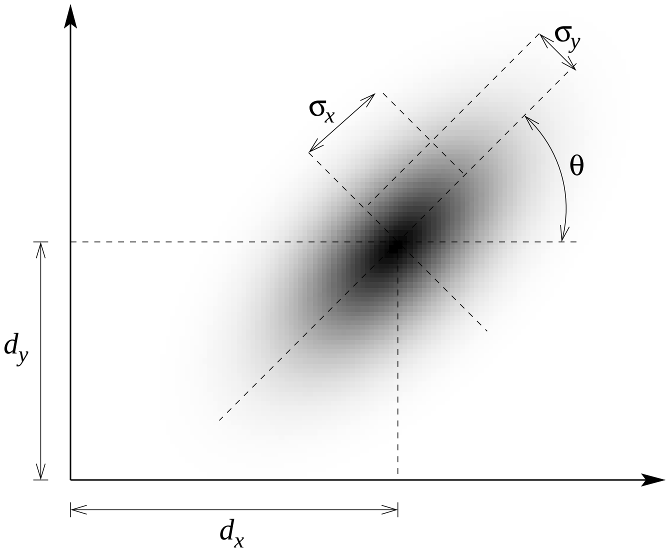


Fig. 6. *Parameterization of prenatal training patterns.* Oriented Gaussian patterns are parameterized by σ_x, the standard deviation in the x-direction; σ_y, the standard deviation in the y-direction; θ, the rotation angle; d_x, the displacement in the x-direction; and d_y, the displacement in the y-direction. By varying these parameters, a variety of prenatal training patterns can be produced.

D. *Evolving Pattern Generators*

As prenatal training patterns in the Inducer3 approach, single oriented two-dimensional Gaussian patterns of floating point values between 0 and 1 were used. By varying the parameters of the Gaussian, a variety of shapes can be produced. The parameters include the size, orientation, and center of the Gaussian, and are defined as: σ_x, the standard deviation of the Gaussian in the x-direction; σ_y, the standard deviation in the y-direction; θ, the rotation angle; d_x, the displacement in the x-direction; and d_y, the displacement in the y-direction (Fig. 6).

Each pattern generator encodes a distribution of such Gaussians. For each of the five parameters, a normal probability distribution is encoded as a mean and variance pair. A pattern is generated by obtaining values for the Gaussian parameters by sampling these distributions. Such an encoding allows evolution to control which Gaussians are generated by manipulating their probabilities.

The mean and variance parameters in the encoding are constrained to restrict the search space, making evolution more efficient. The means of the distributions from which d_x and d_y are drawn are constrained to lie inside the 8×8 pixel grid, and the variances are not allowed to be more than twice the size of the grid. These constraints ensure that the centers of most of the generated patterns lie within the grid. The means and variances of the σ_x and σ_y distributions are constrained to lie in the intervals $[0, 2)$ and $[0, 4)$ respectively, so that most of the generated patterns are smaller than the grid. Similarly, the mean and variance of the distribution of θ are constrained to lie in the ranges $[0, \pi)$ and $[0, \frac{\pi^2}{4})$ respectively, so that most samples lie in $[0, \pi)$, which covers the entire range of orientations.

The pattern generator chromosome is a simple string of

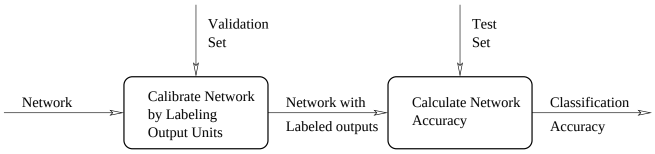


Fig. 7. *Measuring classification accuracy.* The output units of the network are first labeled using a validation set. Using these labels, the classification accuracy of this network on the test set can be determined, by counting how often a unit with the right label wins.

numbers, instead of being divided into genes like in Inducer2, since it contains only one set of Gaussian specifications. The encoding can in principle be generalized to multiple Gaussians, each encoded in a separate gene. Such an encoding can produce more complicated patterns; however, single Gaussians were found to be sufficient to test the hypothesis. Selection, mutation, and crossover (at the level of mean and variance parameters) are performed as in the Inducer2 approach.

E. Estimating Classification Accuracy

Two methods are used to estimate how well the networks perform in their task. The first measures classification accuracy directly, and the second measures fitness more continuously. Both methods require computing a 10×10 confusion matrix, whose (i, j) entry is the number of times output unit j won when examples of digit i were presented to the network.

1) Percentage Correct: The first method calculates the percentage of examples that are correctly recognized from the test set (Fig. 7). Since neither competitive learning nor the evolved neural network have any inherent labels on the output units to indicate which digits they each represent, the labeling must be done after learning, based on the performance of the network on the validation set. Each output unit j is assigned the label of the first row with the highest value in column j of the confusion matrix on the validation set. In some cases, the same label is assigned to multiple output units, some digits may not be represented by any output unit, and some units may not get labeled at all (if they do not win any inputs). After labeling, the classification accuracy on the test set can be determined.

2) Fitness Estimation: The second method measures classification accuracy based on how close to orthogonal the rows of the confusion matrix are. If the classifier is perfect then there can be only one non-zero entry in each column, corresponding to the digit that the unit recognizes. The average angle between the rows can therefore be used as a measure of classification accuracy without having to label the output units.

During evolution, the confusion matrix is calculated from the training set, and the average angle is used as fitness of networks and pattern generators. Such a fitness provides a smoother fitness landscape for evolution than the percentage-correct method. It rewards changes in the confusion matrix that may not result in any immediate increase in the percentage accuracy, but are likely to do so when accumulated over several generations. In contrast, the performance of final networks produced by all three approaches is measured using percentage correct.

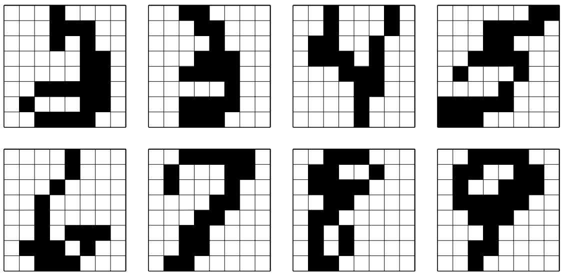


Fig. 8. *Example inputs in the handwritten digit recognition domain.* The original 39×39 gray-scale images from the NIST database were downsampled and thresholded to form a simple but challenging set of examples for an 11-fold crossvalidation experiment.

TABLE I

PARAMETERS FOR EVOLUTION AND LEARNING

	Inducer1	Inducer2	Inducer3
Prenatal learning rate			0.005
Prenatal max. epochs			100
Postnatal learning rate	0.0005		0.0005
Postnatal max. epochs	1000		1000
Mutation factor		0.2	0.2
Mutation rate		0.9	0.5
Mating rate (fields)		0.1	0.05
Mating rate (genes)		0.05	0.025
Number of genes		10	1
Size of pattern set			100
Population size		25	25
Max. generations		9000	9000

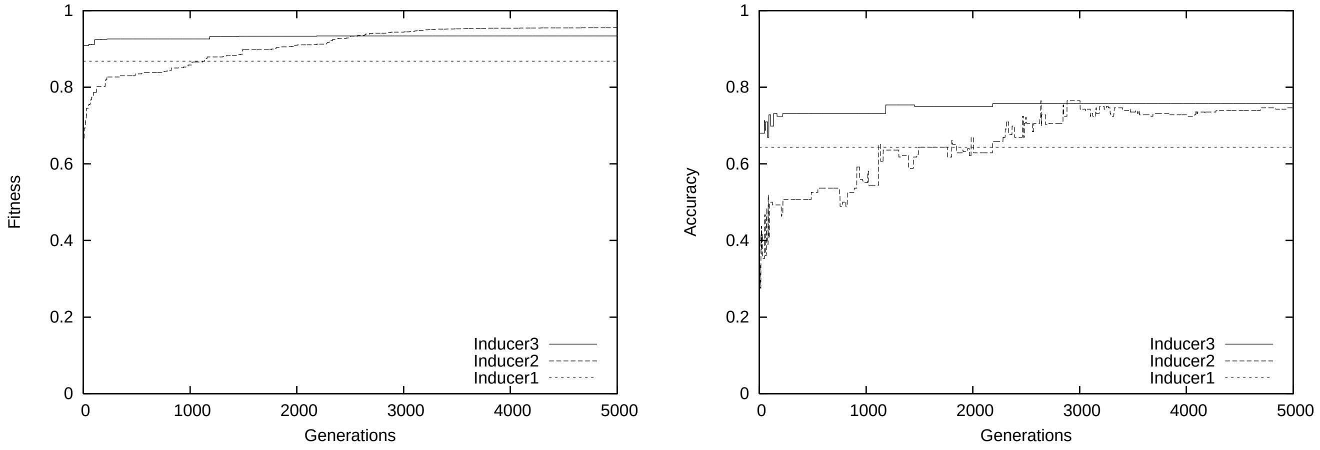
V. RESULTS

In order to compare the performance of the three approaches, a crossvalidation experiment was run on the handwritten digit recognition domain. In this section, the dataset and the parameterization of the experiment will be described, the accuracy of the resulting networks measured, and the performance of the networks analyzed in terms of evolved patterns and network weights.

A. Method

The three classifier network inducers were evaluated using a 2992 image subset of the National Institute for Standards and Technology (NIST) handwritten digit database. The images were originally gray-scale values on a 39×39 grid, and were downsampled to 8×8 binary values to make the inputs simpler to code and represent in the network, and at the same time to make the recognition task more challenging. A few examples from the resulting dataset are shown in Fig. 8. The dataset was shuffled and split into 11 equal-size parts so that a 11-fold crossvalidation experiment could be run on it. In each of the 11 splits, a different part was used for testing the classifier accuracy, another different part for validation (i.e., determining when to stop evolution), and the remaining nine parts for training. The validation set was also used for labeling the output units, in order to obtain labels that generalize well to unseen examples.

Suitable values for the evolution and competitive learning parameters were determined experimentally prior to the experiment (Table I). Competitive learning in Inducer1 and Inducer3



(a) Fitness

(b) Accuracy

Fig. 9. *Improvement in fitness and accuracy during one representative evolution run of Inducer2 and Inducer3.* The final performance of Inducer1 is also shown for comparison. The pattern-generator-based learner (Inducer3) reaches the same good level of performance much earlier than the direct evolution learner (Inducer2), suggesting that it may be an effective strategy for constructing complex systems.

TABLE II
AVERAGE CLASSIFICATION ACCURACIES

	Inducer1	Inducer2	Inducer3
Average Accuracy(%)	63.37	75.77	72.23
Average Generations		2781	337

was continued until all weights changed less than 10^{-3} in an epoch, or when a maximum number of epochs was reached, and the network of the final epoch was taken as the result. The training examples were presented in a different random order in each epoch. Evolution in Inducer2 and Inducer3 was continued until the fitness on the validation set leveled off, i.e., did not improve by more than 10^{-5} over the next 700 generations, and the champion of this generation was then selected as the result of evolution.

B. Performance

The average classification accuracy over the 11-fold cross-validation experiment is shown in Table II. The first result is that Inducer1 is significantly less accurate than either Inducer2 or Inducer3 (the difference is statistically significant according to pairwise Student's t-test, with $p < 0.01, df = 10$). Apparently, Inducer3 is able to discover suitable biases that allow competitive learning to avoid local optima. Second, Inducer3 achieves the same high level of accuracy in a much smaller number of generations than Inducer2 (their difference is statistically not significant according to pairwise Student's t-test, with $p > 0.18, df = 10$). Evolution only needs to discover a suitable bias for learning instead of the final network, which can be done much quicker. These results are further illustrated in Fig. 9, where fitness and accuracy are plotted over generations for one example evolution run. Inducer3 achieves the same good level of performance much earlier than Inducer2, and both are significantly better than Inducer1.

C. Analysis of Patterns and Networks

Because each output unit in the competitive learning network is connected directly to each input unit, it is possible to visualize its connection weights in the same way as the input patterns, i.e., on a 8×8 grid. Such a visualization makes it clear what kinds of input patterns that output unit is most likely to win in the competition (Equation 2).

The connection weights of the network initialized with random weights, used as the starting point for postnatal learning in Inducer1 and prenatal learning in Inducer3, are shown in Fig. 10. The final network weights for one example split of the dataset are shown for Inducer1 and Inducer2 in Figs. 11 and 12. The prenatally trained weights and final weights of Inducer3 are shown for the same split in Fig. 14, and a few example patterns produced by the pattern generator that was evolved in this run are shown in Fig. 13. In all the weight figures, if an output unit wins a large number of examples of a particular digit from the test set (i.e., at least 75% of the largest number of wins for that digit by any unit), then that digit is shown on top of that unit. Thus, if the network is a good classifier, such as the final network of Inducer3 (Fig. 14(b)), a different single digit will be shown on top of every unit, indicating that each digit is recognized as a separate class. In contrast, in a poor classifier, such as the random initial network (Fig. 10), some units do not represent any digits at all, while other units represent multiple digits, and some digits are represented by multiple units.

Several interesting observations can be made based on these figures. First, note that during each input presentation, the competitive learning algorithm makes the weight vector of the winning unit more similar to that input (Equation 3). Therefore, with both Inducer1 and Inducer3, the learned weights end up visually resembling the digits the unit wins in the competition. That is, they approach the mathematical average of the bitmaps for all the examples of that digit in the training set.

On the other hand, the weights of Inducer2 look very

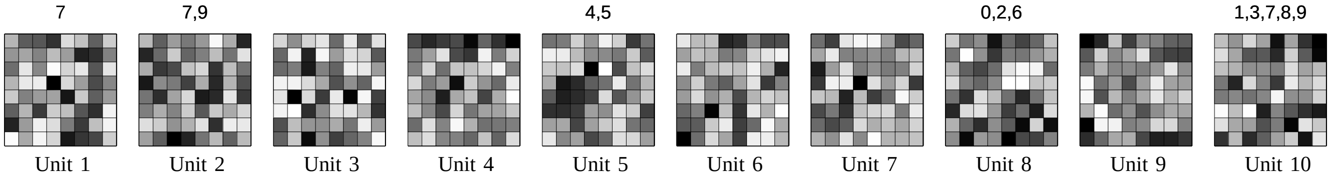


Fig. 10. *Random weights to each output unit of the initial network used in Inducer1 and Inducer3.* The weights are arranged in an 8×8 grid corresponding to the pixels in the input image. Darker squares represent stronger weights. A digit on top indicates that this unit wins a large number of examples of that digit. The assignment of digits to units is uneven, indicating that this network is a poor classifier.

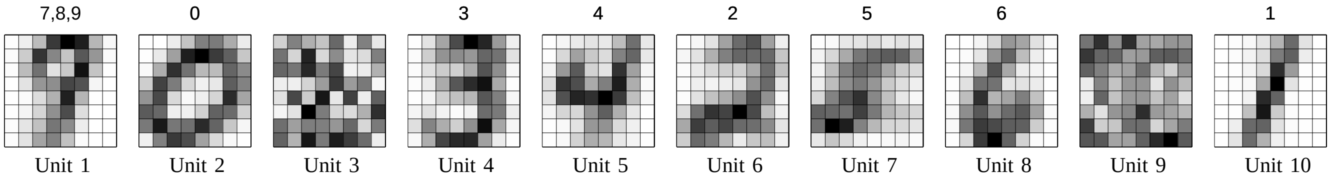


Fig. 11. *Final weights for each output unit of Inducer1 for an example split.* Most of the weights have converged to a configuration that imitates the input digit patterns; however, two units still have unorganized weights, while unit 1 represents a combination of digits 7, 8 and 9. This result demonstrates how competitive learning can get stuck in a local optimum when it does not start with an appropriate initial bias.

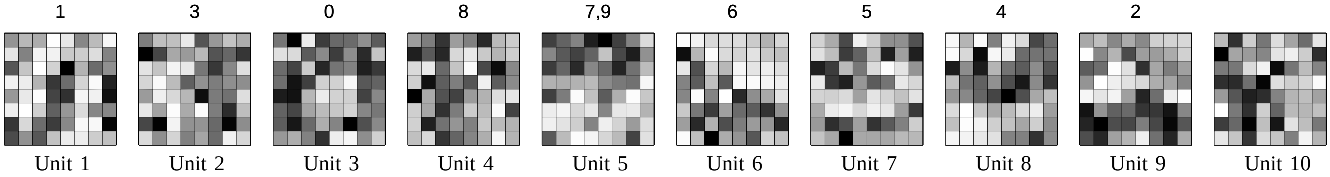


Fig. 12. *Final weights for the output units of Inducer2 for the same split as in Fig. 11.* All digits except 7 and 9 are properly represented; however, since the weights are evolved directly, evolution has discovered weight patterns that represent features crucial for classification, rather than the digits themselves.

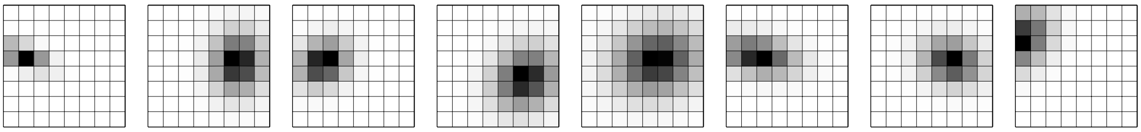
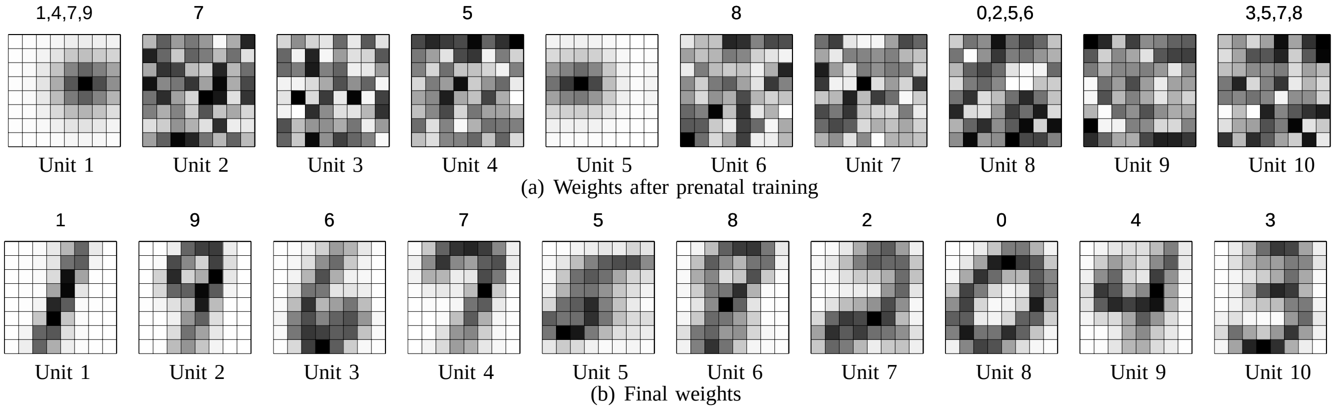


Fig. 13. *A few example patterns produced by the pattern generator in the same split as Figs. 11 and 12.* Each pattern is formed by sampling the distribution for each of the five parameters for oriented Gaussians. These patterns tend to be located around the middle and slightly horizontal. The weights resulting from prenatal training with such patterns are shown in Fig. 14(a).



(a) Weights after prenatal training

(b) Final weights

Fig. 14. *Weights for each output unit of Inducer3 for the same split as in Figs. 11 and 12, trained prenatally with patterns like those in Fig. 13.* Comparing these results to the random weights network in Fig. 10, it is clear that only two of the ten units learn a significant bias (a). Yet, these biases are sufficient for postnatal training to perform better than in Inducer1 and all digits are represented well by the final weight patterns (b).

different. They are adjusted by the evolutionary algorithm, which searches the weight space globally, and finds a weight combination that achieves high fitness. There is no pressure to make the weights look like the inputs, as long as they result in the right classifications on the training set. Indeed, on close inspection it is possible to identify certain strategically set weight values that cause the units to respond more strongly to particular digits than others. For example, digit 0 is represented by a loop of strong weights enclosing a small weaker region in unit 3, and the bottom part of digit 6 is represented by strong weights at the bottom of unit 6. However, digits 7 and 9 are still being confused in this run because the strong region in the upper half of unit 5 matches both digits. Evolution was not able to find the few weight settings that help distinguish these two rather similar digits, although it succeeded in separating them in some other runs. Thus, the partial and approximate matches that the weight patterns have with particular digits give them sufficient advantage to win examples of those digits in the competition with other units. In effect, evolution has discovered a set of reliable features that allows performing the classification task reliably, without forming internal models of the digits themselves.

The final weight patterns of Inducer1 illustrate some of the weaknesses of the competitive learning algorithm. Digits 7, 8 and 9 are particularly difficult for it to distinguish, because many pixels are common between them in the handwriting samples of many people (Fig. 8). For example, digits 7 and 9 are sometimes distinguished only by whether the top part of the digit is open or closed. Lacking proper initial weight biases that would allow the network to respond to such small differences, Inducer1 is likely to get stuck in a local optimum with one unit representing multiple digits. Since only the unit that wins gets to adapt, another unit that may have had the potential to represent one of the digits will not get the chance to learn that digit. For the same reason, some units may never learn; they remain unorganized, still retaining their initial random weights, even at the end of training like units 3 and 9 in Fig. 11.

These problems are avoided to a large extent in Inducer2 and Inducer3, resulting in better classification accuracy. Inducer2 is based on a very different approach: finding features that allow separating these digits to different units, as discussed in the previous subsection. However, the process in which Inducer3 overcomes these problems is interesting, and upon close examination gives us insights into how prenatal training can establish just the right bias for successful competitive learning, as will be discussed next.

VI. Effect of Prenatal Training

The most obvious way to establish an appropriate bias would be to separate each digit to a different unit as much as possible already in prenatal training, so that postnatal training would find it easier to complete the separation. However, this effect is typically not seen in the prenatal training phase of Inducer3. Only units 1 and 5 adapt their weights, and units 1, 8 and 10 end up representing several different digit classes. How does such seemingly counterproductive initial bias make postnatal learning easier?

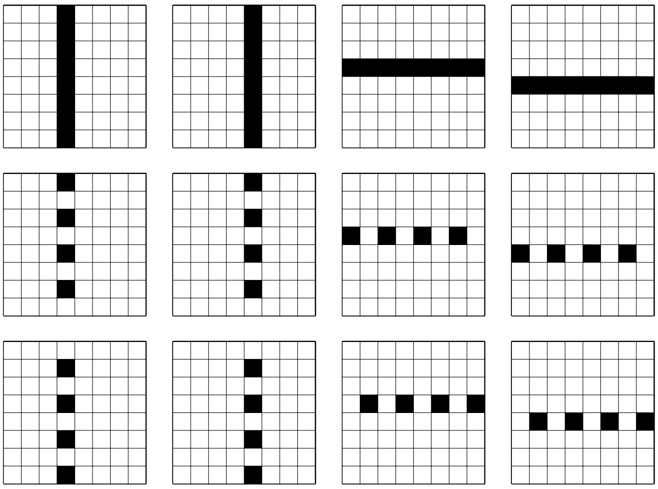


Fig. 15. *Training examples for categorization of vertical and horizontal lines.* There are four categories and three examples in each category: a solid line and two complementary dotted lines. This design makes the effect of prenatal training explicit, as shown in Figs. 16 – 20.

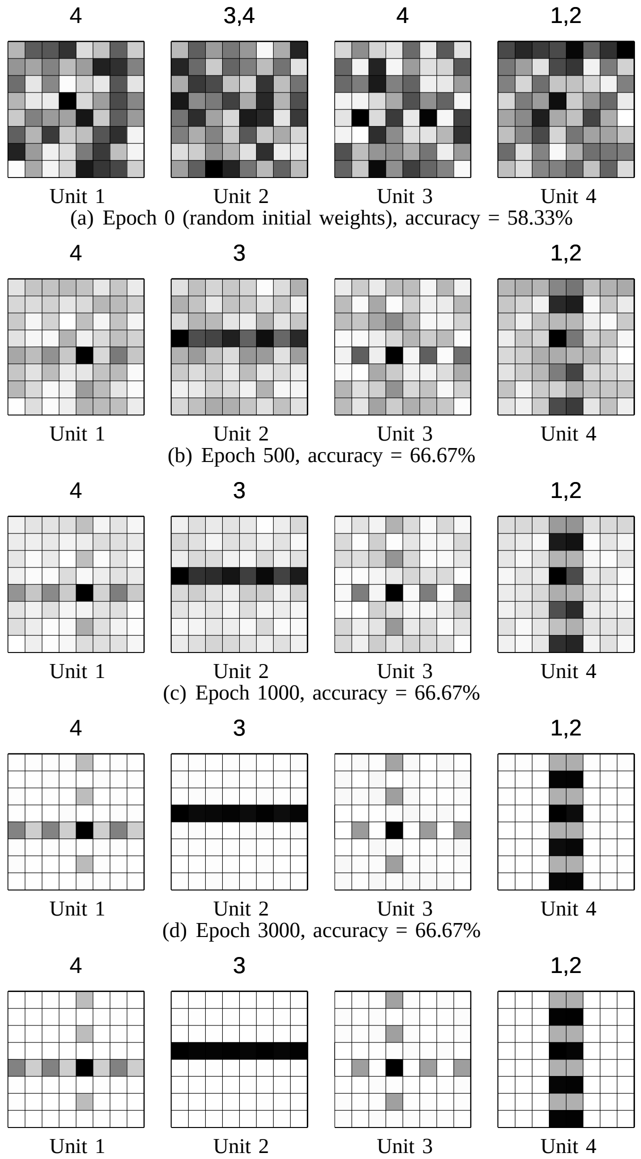
A. Illustrative Example

To illustrate more clearly the effect of prenatal learning in the digit recognition task, which we will return to in Section VI-D, let us first consider a simpler problem: that of learning to categorize vertical and horizontal lines. There are four categories to be learned: A vertical line in column 4 of the input grid, a vertical line in column 5, a horizontal line in row 4, and a horizontal line in row 5 (Fig. 15). The training set consists of 12 examples in total, with three examples for each of the four categories: A solid line and two dotted lines that are pixel complements of each other.

Competitive learning is likely to categorize examples based on how similar they are, i.e., how many pixels they have in common. The examples of a given vertical or horizontal category have several pixels in common, and it should be possible to learn to categorize them correctly. However, a common pixel also exists between a vertical and a horizontal line. If an output unit exists with particularly high weights on that pixel, the learning algorithm may learn to map them both to that unit. The learning may also fail if an output unit has initial weights that allow it to win examples of two categories, even if these categories have nothing in common. If there are no viable competitors for these categories, the unit will gradually learn to respond strongly to both of them.

B. Inducer1 Learning

Competitive learning in Inducer1 fails in exactly these two ways in the line classification problem (Fig. 16). The learned weight patterns for each output unit are gradually seen to emerge from epoch 500 onward. With careful observation, it is possible to see that initial biases for these patterns already existed in the initial random weights (epoch 0), and as the learning continues, these biases get stronger. When the weights converge around epoch 5000, only unit 2 has learned a clean category, recognizing exclusively all three examples of the



(a) Epoch 0 (random initial weights), accuracy = 58.33%

(b) Epoch 500, accuracy = 66.67%

(c) Epoch 1000, accuracy = 66.67%

(d) Epoch 3000, accuracy = 66.67%

(e) Epoch 5000, accuracy = 66.67%

Fig. 16. *Inducer1 weights at various stages of learning to categorize vertical and horizontal lines.* The network gets stuck in a local optimum, where unit 4 has strong representation for two categories, while unit 3 has none.

horizontal line in row 4. Units 1 and 3 succumb to the first pitfall of combining two categories because they have one pixel in common. Unit 4 demonstrates the second pitfall by learning two disjoint patterns for which there is no viable competition from other units. There is only one categorization change from initial to the final state: One of the examples of category 4 is reclassified from unit 2 to unit 1 by epoch 500, which improves classification accuracy from 58% to 67%.

C. Inducer3 Learning

Inducer3 was applied to the same classification task using a manually constructed prenatal training pattern shown in Fig. 17. This pattern was designed to produce clustering of different categories on one unit, similar to the clustering of digits seen on some units of Fig. 14(a). Snapshots of the

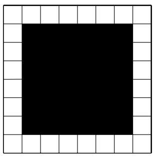
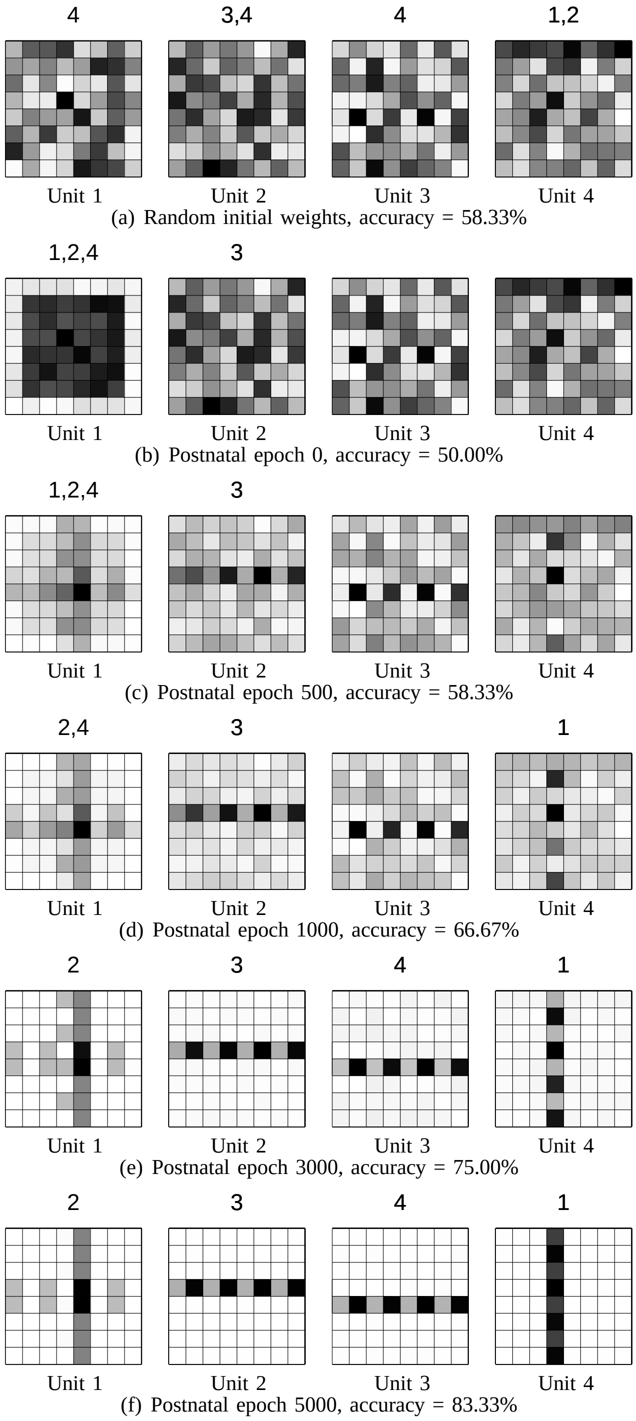


Fig. 17. *A good pattern for prenatal training of Inducer3, designed to produce a beneficial clustering effect.*



(a) Random initial weights, accuracy = 58.33%

(b) Postnatal epoch 0, accuracy = 50.00%

(c) Postnatal epoch 500, accuracy = 58.33%

(d) Postnatal epoch 1000, accuracy = 66.67%

(e) Postnatal epoch 3000, accuracy = 75.00%

(f) Postnatal epoch 5000, accuracy = 83.33%

Fig. 18. *Inducer3 weights at various stages of learning to categorize vertical and horizontal lines.* After prenatal training, unit 1 wins most of the patterns. Based on the remaining patterns, the other units develop distinct categories, and eventually the whole system converges to good categorization.

network weights during the whole of postnatal training reveal how biasing the network in this manner allows it to avoid local optima and separate the categories better than Inducer1 (Fig. 18).

The prenatal training pattern has been learned by output unit 1; the other units maintained their random weights until the end of prenatal training (postnatal epoch 0). As with Inducer1, the slight random biases of units 2, 3 and 4 gradually strengthen during postnatal training with examples. However, unlike with Inducer1, the effect of these biases is diminished by unit 1, which wins most of the examples in the beginning, leaving only a few examples for the other units. As a result, units 2, 3 and 4 specialize to recognize only those examples for which they initially had the highest bias, while the other examples are captured by unit 1. As units 2, 3 and 4 specialize, they gradually start winning other similar examples over unit 1 as well. Simultaneously, unit 1 becomes more specialized to examples for which it does not have significant competition from the other units. This process allows units 2, 3 and 4 to incrementally learn and specialize to examples that are originally represented by unit 1. A good separation of examples into different units results, in contrast to the confused categories learned by Inducer1.

The clustering of categories on unit 1 reduced accuracy from 58% with the random initial weights to 50% at the end of prenatal training. Yet Inducer3 eventually achieves a classification accuracy of 83%. This result demonstrates that prenatal training indeed establishes a suitable starting point for postnatal training, making it possible for the system as a whole to achieve good final performance.

Postnatal training is highly sensitive to the choice of prenatal training patterns. For example, although the pattern in Fig. 19, which has two additional active pixels on both sides, produces a clustering effect as well, it does not lead to successful postnatal learning (Fig. 20). The unit that learns it (unit 2) matches too many postnatal training patterns. While there are enough leftover patterns for units 3 and 4 to develop unique categories, unit 1 does not win any examples. Instead, unit 2 represents two categories, resulting in large error. The final accuracy achieved by the network is only 67%.

Ahead of time it would have been difficult to guess that the pattern in Fig. 17 is successful while that in Fig. 19 is not. In other words, the learning is sensitive to the right bias, and the patterns that establish them are not obvious. Therefore, a learning method like evolution is useful for discovering the appropriate patterns for prenatal training.

D. Effect in More Complex Tasks

In the line-categorization task it is easy to see how the initial bias established by prenatal training allowed Inducer3 to succeed while Inducer1 failed. In a more complicated task like digit recognition, where there are many categories and the examples are not as clearly defined as the horizontal and vertical lines, it is harder to trace the exact learning path taken by the algorithm. This path can be highly convoluted, with output units changing their labels multiple times before converging to a particular digit. However, the basic mechanisms through

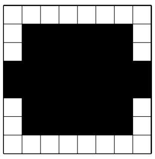
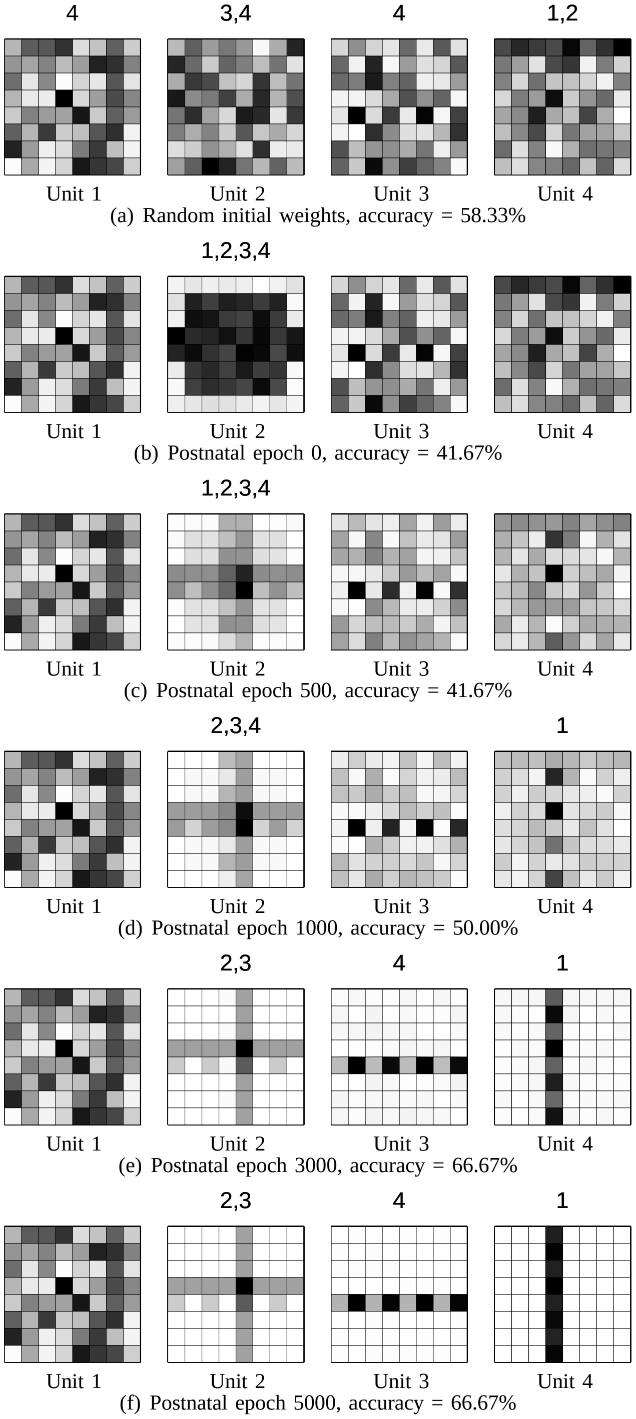


Fig. 19. *A bad pattern for prenatal training of Inducer3, designed to produce a detrimental clustering effect.*



(a) Random initial weights, accuracy = 58.33%

(b) Postnatal epoch 0, accuracy = 41.67%

(c) Postnatal epoch 500, accuracy = 41.67%

(d) Postnatal epoch 1000, accuracy = 50.00%

(e) Postnatal epoch 3000, accuracy = 66.67%

(f) Postnatal epoch 5000, accuracy = 66.67%

Fig. 20. *Inducer3 weights at various stages of learning using a bad prenatal training pattern.* The clustering effect is too strong, preventing unit 1 from learning anything. This example shows that the system is sensitive to prenatal training, and that the right patterns are difficult to discover, making evolution of pattern generators a useful approach.

which prenatal training allows postnatal training to succeed are the same: they establish an appropriate bias as a good starting point from which the solution can be reached easily.

Interestingly, even when the pattern generators can in principle evolve to produce more complex patterns, such as those resembling the actual digits, evolution does not make use of this possibility. In preliminary experiments with patterns consisting of multiple Gaussians, the best generators that evolved were still relatively simple, only establishing biases similar to those with single Gaussians. Only a few output units learned during the prenatal phase, while the remaining units maintained their initial random weights. Apparently, such learning biases are a more efficient way to achieve good performance on this task than constructing the recognition network directly. This principle illustrates an important and surprising synergy of evolution and learning.

VII. Discussion and Future Work

As the classification accuracies in Table II show, Inducer1 is the least accurate of the three methods. Since the weights are initialized randomly, the network is not biased in favor of any learning path. Without a proper bias, it frequently gets stuck in local optima. On the other hand, the networks in Inducer3 are prenatally trained with the generated patterns. Evolution converges on a pattern generator that establishes biases on a few output units, making it easier for the network to separate the categories during the postnatal learning phase. The result is a significantly better classification accuracy for Inducer3. In this section, the benefits of Inducer3 in constructing complex systems over direct evolution of weights (Inducer2) and over direct weight initialization methods are analyzed. The biological insights from the computational studies are discussed and directions for future research outlined.

A. Benefits Over Direct Evolution

The direct-evolution approach of Inducer2 discovers good solutions eventually, and can be seen as a possible alternative to the pattern-generator approach of Inducer3. As a matter of fact, Inducer2 takes less computation time than Inducer3 because fitness evaluation of networks in Inducer2 does not involve learning. However, when constructing complex systems such as those in evolutionary robotics, fitness evaluation of an individual requires it to live a “lifetime” and interact with its environment. The individual can either be fixed as in Inducer2, or it can learn from the environment and improve its fitness, as in Inducer3. In both cases the duration of a lifetime (i.e., a generation) is the same. In this respect, the cost of evaluating an individual is the same for both Inducer2 and Inducer3, and only the number of generations needs to be compared, as was done in this paper.

According to this measure, Inducer2 reaches a good level of accuracy eventually, if evolution is allowed to proceed long enough. However, Inducer3 reaches this same level in much fewer generations than Inducer2, as shown by the fitness and accuracy plots in Fig. 9. In other words, constructing the digit recognition network is much more efficient through prenatal pattern generation.

Without a proper bias, Inducer2 requires evolution to search a large space of possible solutions. The digit recognition experiment with such small networks is still within the limits of direct evolution, and a solution will eventually be found. However, larger problems with larger search spaces may no longer be tractable for Inducer2. For instance, if the network were expanded to use the full resolution (39×39) of digits available in the NIST dataset, the size of the evolutionary search space for Inducer2 would increase from 650 to 15220, whereas that for Inducer3 would remain constant at 10. Thus, the results in this paper suggest that the pattern-generator approach may turn out to be crucial in solving such large-scale problems.

B. Benefits Over Direct Initialization

Since prenatal learning in Inducer3 establishes the right bias by initializing the weights for postnatal learning, couldn’t the same benefits be achieved by setting the weights directly? Indeed, it is possible to envision such algorithms, e.g., based on direct evolution. However, they would have to represent the large parameter space by a compact encoding so that searching for a good set of parameters becomes manageable.

To analyze this approach, an experiment similar to Inducer3 was designed with such a compact encoding. The weights to the output units were initialized to oriented Gaussian patterns, and evolution was used to search through such weight initializations. A set of 10 Gaussians were evolved, each represented by the same five parameters as those in Inducer3 (Fig. 6). The final performance was similar to that of Inducer3 (average classification accuracy of 73.33% in 547 generations over the 11 splits). Thus, direct weight initialization accomplishes essentially the same biasing as prenatal learning from generated patterns.

However, it is often difficult to construct such parameter-space encodings, especially when the architecture of the learner is complex and irregular. It was only possible in the above experiment because the network architecture is so simple, associating connection weights one-to-one with input pixels. In contrast, the pattern generator is a compact encoding in the input space, which is likely to be smaller and more regular than the parameter space. For example, in a vision processing task, the input is the visual field, which is easier to approximate by a compact encoding than the large and irregular parameter space of a complex learning algorithm that processes the input. Thus, the pattern-generator approach may be successful in establishing the proper bias even on complex learning systems that are difficult to initialize directly.

C. Biological Insights

The pattern-generator approach is inspired by observations from nature (Section II-A), and may in turn be used to gain a deeper understanding of biological learning systems. The results obtained using this approach support the biological hypothesis that genetically determined, internally generated activity patterns are indeed an efficient mechanism for establishing the appropriate bias, which is crucial for constructing complex systems such as those seen in biology (Section I).

The computational implementation can be further refined to make more detailed biological predictions, e.g., by using more biologically realistic learning algorithms and pattern generators. Such extensions constitute an important direction of future work.

Devising such biologically realistic computational models also involves modeling the hierarchical organization of the visual cortex [86]. The lower levels of this hierarchy have small retinal receptive fields and perform simple functions such as orientation processing, while the higher levels have larger receptive fields and perform more complex functions such as face detection [43], [68]. Evidence for pattern generation occurring at the low levels was reported in Section II-A. Evidently, these biases enable the system to efficiently detect features in the visual input such as orientation, which are then used by the higher cortical levels to perform more sophisticated processing. Such features condense the information to only what is relevant, making high-level learning easier. A similar hierarchical approach should be useful in building artificial learners as well. For example, a hierarchical extension of the digit recognition network in Fig.2 could be constructed as a two-layer network. In this architecture, the lower layer would learn the bias necessary to detect features in the prenatal phase, and these features would then form the inputs to the upper layer during postnatal learning. As a result, the system should be able to learn more complex behaviors than what is possible in a single-layer network.

Alternative ways to establish a proper bias exist in biology as well. For example, axon growth guided by molecular gradients has been found to play a crucial role in the early development of structures in animal nervous systems (see [31] for review), and could result in a genetically determined bias. While pattern generation and axonal growth can both in principle achieve a similar effect, little is known about how they compare and possibly interact. Computational models can be built in the future to gain insights into such questions as well.

In the quest to discover the mechanisms behind development of the brain, studying the patterns in cortical organization might prove useful. The network plots in Section V-C show how the particular weight adaptation mechanisms affect the organization of the network. The weight plots produced by competitive learning are smooth and resemble the digits they represent, while those produced by direct evolution look grainy and represent only critical features of digits. This observation suggests that it may be possible to identify which mechanisms are responsible for particular structures by studying the structure and organization of the cortex. For example, the smooth, regular pattern of orientation processing cells observed in the visual cortex may suggest that such organization is primarily a result of activity-dependent self-organization, and less of genetic hardwiring.

D. Extensions to Large Systems and Changing Environments

The competitive learning network for handwritten digit recognition is used in this paper to demonstrate how the combination of evolved pattern generators and prenatal training can compensate for the shortcomings of learning algorithms by discovering an appropriate bias. In this sense, the study is a proof of concept for the pattern-generator approach. In future work, this approach needs to be applied to more complex learners and domains, verifying that it scales up and is useful more generally. In the specific case of competitive learning, evolution discovers biases that allow learning to avoid local optima, such as unorganized output units and units with clustered digits, by providing a good starting point through prenatal learning. For other learning algorithms, prenatal learning can establish biases and avoid local optima in other ways. These biases depend on the weaknesses of that particular learner, and specifically those weaknesses that are exposed by the experimental conditions and problem domain. If successful, this effort could ultimately pave the way for the engineering of complex systems that are otherwise difficult or impossible to construct.

In addition to establishing the right biases for postnatal learning, internal pattern generation could serve a significant role in adult adaptation to changing environments. If the system learns only from experience after birth, the prenatally established biases are soon overridden, and the system would have difficulty adapting to new environments. However, if internally generated activity was interleaved with experience-based learning (as may happen during REM sleep in mammals [11]), some of the prenatal organization would be retained, making further adaptation more effective. Such postnatal patterns may explain why animals can learn altered environments only partially [71], and why the animal spends so much time in REM sleep during the time when its neural structures are most plastic [67]: Postnatal internally generated patterns may help ensure that the learning system does not become too closely adapted to a particular environment. Applying this idea to constructing artificial systems for changing environments is a most interesting direction of future work.

VIII. Conclusions

Research on brain development in animals has led to insights on how complex brain structures are constructed prenatally and postnatally. Spontaneous activity in the brain before birth may be responsible for rudimentary structures that are found in most animals at birth. Such prenatal training may have been discovered by evolution to establish a proper bias, so that the system can learn efficiently from environmental inputs after birth. This paper demonstrates how the same approach could work more generally for building complex systems. The hypothesis is that pretraining a system with patterns from an evolved generator will establish the required bias to make learning from the actual data easier. Experiments in the handwritten digit recognition domain support this hypothesis, suggesting that complex systems can be effectively constructed in this way.

Acknowledgments

This research was supported in part by the NIH under Human Brain Project Grant 1R01-MH66991 and in part by the NSF under grant EIA-0303609.

References

[1] J. A. Anderson and E. Rosenfeld, Eds., *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press, 1988.

[2] J. M. Baldwin, "A new factor in evolution," *The American Naturalist*, vol. 30, pp. 441–451, 536–553, 1896.

[3] H. B. Barlow, "Unsupervised learning," *Neural Computation*, vol. 1, pp. 295–311, 1989.

[4] J. A. Bednar, "Learning to see: Genetic and environmental influences on visual development," Ph.D. dissertation, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 2002, technical Report AI-TR-02-294.

[5] J. A. Bednar and R. Miikkulainen, "Pattern-generator-driven development in self-organizing models," in *Computational Neuroscience: Trends in Research, 1998*, J. M. Bower, Ed. New York: Plenum Press, 1998, pp. 317–323.

[6] ——, "Learning innate face preferences," *Neural Computation*, vol. 15, no. 7, pp. 1525–1557, 2003.

[7] ——, "Constructing visual function through prenatal and postnatal learning," in *Neuroconstructivism, Vol. 2: Perspectives and Prospects*, D. Mareschal, M. H. Johnson, S. Sirois, M. Spratling, M. S. C. Thomas, and G. Westermann, Eds. Oxford, UK: Oxford University Press, 2005.

[8] C. Blakemore and G. F. Cooper, "Development of the brain depends on the visual environment," *Nature*, vol. 228, pp. 477–478, 1970.

[9] C. Blakemore and R. C. van Sluyters, "Innate and environmental factors in the development of the kitten's visual cortex," *The Journal of Physiology*, vol. 248, pp. 663–716, 1975.

[10] D. Burger and J. R. Goodman, "Billion-transistor architectures," *IEEE Computer*, vol. 30, pp. 46–48, 1997.

[11] C. W. Callaway, R. Lydic, H. A. Baghdoyan, and J. A. Hobson, "Pontogeniculooccipital waves: Spontaneous visual system activity during rapid eye movement sleep," *Cellular and Molecular Neurobiology*, vol. 7, no. 2, pp. 105–149, 1987.

[12] A. Cangelosi, D. Parisi, and S. Nolfi, "Cell division and migration in a 'genotype' for neural networks," *Network: Computation in Neural Systems*, vol. 5, pp. 497–515, 1994.

[13] M. Catsicas and P. Mobbs, "Waves are swell," *Current Biology*, vol. 5, no. 9, pp. 977–979, 1995.

[14] B. Chapman and T. Bonhoeffer, "Overrepresentation of horizontal and vertical orientation preferences in developing ferret area 17," *Proceedings of the National Academy of Sciences, USA*, vol. 95, pp. 2609–2614, 1998.

[15] B. Chapman, I. Gödecke, and T. Bonhoeffer, "Development of orientation preference in the mammalian visual cortex," *Journal of Neurobiology*, vol. 41, no. 1, pp. 18–24, 1999.

[16] B. Chapman and M. P. Stryker, "Development of orientation selectivity in ferret primary visual cortex and effects of deprivation," *The Journal of Neuroscience*, vol. 13, no. 12, pp. 5251–5262, Dec. 1993.

[17] B. Chapman, M. P. Stryker, and T. Bonhoeffer, "Development of orientation preference maps in ferret primary visual cortex," *The Journal of Neuroscience*, vol. 16, no. 20, pp. 6443–6453, 1996.

[18] G. Chouvet, R. Blois, G. Debilly, and M. Jouvet, "La structure d'occurrence des mouvements oculaires rapides du sommeil paradoxal est similaire chez les jumeaux homozygotes [The structure of the occurrence of rapid eye movements in paradoxical sleep is similar in homozygotic twins]," *Comptes Rendus des Seances de l'Academie des Sciences – Serie III, Sciences de la Vie*, vol. 296, no. 22, pp. 1063–1068, 1983.

[19] M. Constantine-Paton, H. T. Cline, and E. Debski, "Patterned activity, synaptic convergence, and the NMDA receptor in developing visual pathways," *Annual Review of Neuroscience*, vol. 13, pp. 129–154, 1990.

[20] D. M. Coppola, L. E. White, D. Fitzpatrick, and D. Purves, "Unequal representation of cardinal and oblique contours in ferret visual cortex," *Proceedings of the National Academy of Sciences, USA*, vol. 95, no. 5, pp. 2621–2623, 1998.

[21] M. C. Crair, "Neuronal activity during development: Permissive or instructive?" *Current Opinion in Neurobiology*, vol. 9, pp. 88–93, 1999.

[22] M. C. Crair, D. C. Gillespie, and M. P. Stryker, "The role of visual experience in the development of columns in cat visual cortex," *Science*, vol. 279, pp. 566–570, 1998.

[23] J. C. Crowley and L. C. Katz, "Early development of ocular dominance columns," *Science*, vol. 290, pp. 1321–1324, 2000.

[24] S. Datta, "Cellular basis of pontine ponto-geniculo-occipital wave generation and modulation," *Cellular and Molecular Neurobiology*, vol. 17, no. 3, pp. 341–365, 1997.

[25] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett, *Rethinking Innateness: A Connectionist Perspective on Development*. Cambridge, MA: MIT Press, 1996.

[26] M. B. Feller, D. P. Wellis, D. Stellwagen, F. S. Werblin, and C. J. Shatz, "Requirement for cholinergic synaptic transmission in the propagation of spontaneous retinal waves," *Science*, vol. 272, pp. 1182–1187, 1996.

[27] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*, ser. Lecture Notes in Biomathematics 45, S. Amari and M. A. Arbib, Eds. Berlin: Springer, 1982, pp. 267–285.

[28] S. Geman, E. L. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[29] I. Gödecke, D. S. Kim, T. Bonhoeffer, and W. Singer, "Development of orientation preference maps in area 18 of kitten visual cortex," *European Journal of Neuroscience*, vol. 9, no. 8, pp. 1754–1762, Aug. 1997.

[30] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, pp. 23–63, 1987.

[31] E. A. Grove and T. Fukuchi-Shimogori, "Generating the cerebral cortical area map," *Annual Review of Neuroscience*, vol. 26, no. 1, pp. 355–380, 2003.

[32] F. Gruau and D. Whitley, "Adding learning to the cellular development of neural networks: Evolution and the Baldwin effect," *Evolutionary Computation*, vol. 1, pp. 213–233, 1993.

[33] F. Gruau, D. Whitley, and L. Pyeatt, "A comparison between cellular encoding and direct encoding for genetic neural networks," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds. Cambridge, MA: MIT Press, 1996, pp. 81–89.

[34] D. Haussler, "Quantifying inductive bias: AI learning algorithms and Valiant's learning framework," *Artificial Intelligence*, vol. 36, pp. 177–221, 1988.

[35] S. Haykin, *Neural Networks, A Comprehensive Foundation*. Upper Saddle River, New Jersey: Prentice Hall, 1999.

[36] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. Hoboken, NJ: Wiley, 1949.

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

[38] J. Horton and D. Hocking, "An adult-like pattern of ocular dominance columns in striate cortex of newborn monkeys prior to visual experience," *J. Neurosci.*, vol. 16, no. 5, pp. 1791–1807, 1996.

[39] D. H. Hubel and T. N. Wiesel, "Receptive Fields of Cells in Striate Cortex of Very Young, Visually Inexperienced Kittens," *Journal of Neurophysiology*, vol. 26, no. 6, pp. 994–1002, 1963.

[40] ——, "Sequence regularity and geometry of orientation columns in the monkey striate cortex," *The Journal of Comparative Neurology*, vol. 158, pp. 267–294, 1974.

[41] D. H. Hubel, T. N. Wiesel, and S. LeVay, "Plasticity of ocular dominance columns in monkey striate cortex," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 278, pp. 377–409, 1977.

[42] M. Jouvet, "Paradoxical sleep and the nature-nurture controversy," in *Adaptive Capabilities of the Nervous System: Proceedings of the 11th International Summer School of Brain Research*, ser. Progress in Brain Research, P. S. McConnell, G. J. Boer, H. J. Romijn, N. E. van de Poll, and M. A. Corner, Eds. Amsterdam: Elsevier, 1980, vol. 53, pp. 331–346.

[43] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, 4th ed. New York: McGraw-Hill, 2000.

[44] M. Kaschube, F. Wolf, T. Geisel, and S. Lowel, "Genetic Influence on Quantitative Features of Neocortical Architecture," *The Journal of Neuroscience*, vol. 22, no. 16, pp. 7206–7217, 2002.

[45] L. C. Katz and C. J. Shatz, "Synaptic activity and the construction of cortical circuits," *Science*, vol. 274, pp. 1133–1138, 1996.

[46] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer, 1995.

[47] E. S. Lander *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[48] W. R. Lippe, "Rhythmic spontaneous activity in the developing avian auditory system," *The Journal of Neuroscience*, vol. 14, no. 3, pp. 1486–1495, 1994.

[49] L. Maffei and L. Galli-Resta, "Correlation in the discharges of neighboring rat retinal ganglion cells during prenatal life," *Proceedings of the National Academy of Sciences, USA*, vol. 87, pp. 2861–2864, 1990.

[50] G. A. Marks, J. P. Shaffery, A. Oksenberg, S. G. Speciale, and H. P. Roffwarg, "A functional role for REM sleep in brain maturation," *Behavioural Brain Research*, vol. 69, pp. 1–11, 1995.

[51] P. McQuesten, "Cultural enhancement of neuroevolution," Ph.D. dissertation, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 2002, technical Report AI-02-295.

[52] P. McQuesten and R. Miikkulainen, "Culling and teaching in neuroevolution," in *Proceedings of the Seventh International Conference on Genetic Algorithms*, 1997, pp. 760–767.

[53] M. Meister, R. O. L. Wong, D. A. Baylor, and C. J. Shatz, "Synchronous bursts of action-potentials in the ganglion cells of the developing mammalian retina," *Science*, vol. 252, pp. 939–943, 1991.

[54] R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh, *Computational Maps in the Visual Cortex*. Berlin: Springer, 2005.

[55] K. D. Miller, E. Erwin, and A. Kayser, "Is the development of orientation selectivity instructed by activity?" *Journal of Neurobiology*, vol. 41, pp. 44–57, 1999.

[56] Z. Molnár, S. Higashi, and G. López-Bendito, "Choreography of early thalamocortical development," *Cerebral Cortex*, vol. 13, no. 6, pp. 661–669, 2003.

[57] J. A. Movshon and R. C. van Sluyters, "Visual neural development," *Annual Review of Psychology*, vol. 32, pp. 477–522, 1981.

[58] S. Nolfi, J. L. Elman, and D. Parisi, "Learning and evolution in neural networks," *Adaptive Behavior*, vol. 2, pp. 5–28, 1994.

[59] S. Nolfi and D. Parisi, "Auto-teaching: Networks that develop their own teaching input," in *Proceedings of the Second European Conference on Artificial Life*, J. Deneubourg, H. Bersini, S. Goss, G. Nicolis, and R. Dagonnier, Eds., 1993, pp. 845–862.

[60] ——, "Desired answers do not correspond to good teaching inputs in ecological neural networks," *Neural Processing Letters*, vol. 1, no. 2, pp. 1–4, 1994.

[61] ——, "Genotypes for neural networks," in *Handbook of brain theory and neural networks*, M. Arbib, Ed. MIT Press, 1995, pp. 431–434.

[62] M. J. O'Donovan, "The origin of spontaneous activity in developing networks of the vertebrate nervous system," *Current Opinion in Neurobiology*, vol. 9, pp. 94–104, 1999.

[63] A. Peinado, R. Yuste, and L. C. Katz, "Extensive dye-coupling between rat neocortical neurons during the period of circuit formation," *Neuron*, vol. 14, no. 7, pp. 103–114, 1993.

[64] A. A. Penn and C. J. Shatz, "Brain waves and brain wiring: The role of endogenous and sensory-driven neural activity in development," *Pediatric Research*, vol. 45, no. 4, pp. 447–458, 1999.

[65] P. Rakic, "Prenatal genesis of connections subserving ocular dominance in the rhesus monkey," *Nature*, vol. 261, pp. 467–471, June 1976.

[66] ——, "Specification of cerebral cortical areas," *Science*, vol. 241, pp. 170–176, 1988.

[67] H. P. Roffwarg, J. N. Muzio, and W. C. Dement, "Ontogenetic development of the human sleep-dream cycle," *Science*, vol. 152, pp. 604–619, 1966.

[68] E. T. Rolls, "Functions of the primate temporal lobe cortical visual areas in invariant visual object and face recognition," *Neuron*, vol. 27, no. 2, pp. 205–218, 2000.

[69] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 151–193.

[70] F. Sengpiel and P. C. Kind, "The role of activity in development of the visual system," *Current Biology*, vol. 12, no. 23, pp. R818–R826, 2002.

[71] F. Sengpiel, P. Stawinski, and T. Bonhoeffer, "Influence of experience on orientation maps in cat visual cortex," *Nature Neuroscience*, vol. 2, no. 8, pp. 727–732, 1999.

[72] C. J. Shatz, "Impulse activity and the patterning of connections during CNS development," *Neuron*, vol. 5, pp. 745–756, December 1990.

[73] ——, "The developing brain," *Scientific American*, vol. 267, no. 3, pp. 61–67, 1992.

[74] ——, "Emergence of order in visual system development," *Proceedings of the National Academy of Sciences, USA*, vol. 93, pp. 602–608, 1996.

[75] G. M. Shepherd, *The Synaptic Organization of the Brain*, 5th ed. Oxford, UK: Oxford University Press, 2003.

[76] J. M. Siegel, "The evolution of REM sleep," in *Handbook of Behavioral State Control: Cellular and Molecular Mechanisms*, R. Lydic and H. A. Baghdoyan, Eds. Boca Raton, FL: CRC Press, 1999, pp. 87–100.

[77] J. Sirosh, "A self-organizing neural network model of the primary visual cortex," Ph.D. dissertation, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 1995, technical Report AI95-237.

[78] A. Slater and S. P. Johnson, "Visual sensory and perceptual abilities of the newborn: Beyond the blooming, buzzing confusion," in *The Development of Sensory, Motor and Cognitive Capacities in Early Infancy: From Perception to Cognition*, F. Simion and G. Butterworth, Eds. East Sussex, UK: Psychology Press, 1998, pp. 121–142.

[79] A. Slater, V. Morison, and M. Somers, "Orientation discrimination and cortical function in the human newborn," *Perception*, vol. 17, pp. 597–602, 1988.

[80] M. Steriade, D. Paré, D. Bouhassira, M. Deschênes, and G. Oakson, "Phasic activation of lateral geniculate and perigeniculate thalamic neurons during sleep with ponto-geniculo-occipital waves," *The Journal of Neuroscience*, vol. 9, no. 7, pp. 2215–2229, 1989.

[81] M. Sur, A. Angelucci, and J. Sharma, "Rewiring cortex: The role of patterned activity in development and plasticity of neocortical circuits," *Journal of Neurobiology*, vol. 41, pp. 33–43, 1999.

[82] M. Sur and C. A. Leamey, "Development and plasticity of cortical areas and networks," *Nature Reviews Neuroscience*, vol. 2, no. 4, pp. 251–262, 2001.

[83] E. Switkes, M. J. Mayer, and J. A. Sloan, "Spatial frequency analysis of the visual environment: Anisotropy and the carpentered environment hypothesis," *Vision Research*, vol. 18, no. 10, pp. 1393–1399, 1978.

[84] I. Thompson, "Cortical development: A role for spontaneous activity?" *Current Biology*, vol. 7, pp. R324–R326, 1997.

[85] P. Utgoff and T. Mitchell, "Acquisition of appropriate bias for inductive concept learning," in *Proceedings of the Second National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 1982, pp. 414–417.

[86] D. C. Van Essen, C. H. Anderson, and D. J. Felleman, "Information processing in the primate visual system: An integrated systems perspective," *Science*, vol. 255, pp. 419–423, 1992.

[87] J. C. Venter *et al.*, "The sequence of the human genome," *Science*, vol. 291, no. 5507, pp. 1304–1351, 2001.

[88] C. von der Malsburg, "Self-organization of orientation-sensitive cells in the striate cortex," *Kybernetik*, vol. 15, pp. 85–100, 1973.

[89] C. von der Malsburg and D. J. Willshaw, "How to label nerve cells so that they can interconnect in an ordered fashion," *Proceedings of the National Academy of Sciences, USA*, vol. 74, no. 11, pp. 5176–5178, 1977.

[90] M. Weliky and L. C. Katz, "Disruption of orientation tuning in visual cortex by artificially correlated neuronal activity," *Nature*, vol. 386, no. 6626, pp. 680–685, Apr 17 1997.

[91] D. J. Willshaw and C. von der Malsburg, "A marker induction mechanism for the establishment of ordered neural mappings: Its application to the retinotectal problem," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 287, pp. 203–243, 1979.

[92] R. O. L. Wong, M. Meister, and C. J. Shatz, "Transient period of correlated bursting activity during development of the mammalian retina," *Neuron*, vol. 11, no. 5, pp. 923–938, Nov 1993.

[93] R. O. L. Wong, "Retinal waves and visual system development," *Annual Review of Neuroscience*, vol. 22, pp. 29–47, 1999.

[94] R. Yuste, D. A. Nelson, W. W. Rubin, and L. C. Katz, "Neuronal domains in developing neocortex: Mechanisms of coactivation," *Neuron*, vol. 14, no. 1, pp. 7–17, Jan. 1995.

[95] L. I. Zhang and M. M. Poo, "Electrical activity and development of neural circuits," *Nature Neuroscience*, vol. 4 Suppl, pp. 1207–1214, 2001.