

Constructing Good Learners using Evolved Pattern Generators

Vinod Valsalam
Dept. of Computer Sciences
The Univ. of Texas at Austin
Austin, TX 78712 USA
vkv@cs.utexas.edu

James Bednar
School of Informatics
The University of Edinburgh
Edinburgh, EH1 2QL UK
jbednar@inf.ed.ac.uk

Risto Miikkulainen
Dept. of Computer Sciences
The Univ. of Texas at Austin
Austin, TX 78712 USA
risto@cs.utexas.edu

ABSTRACT

Self-organization of brain areas in animals begins prenatally, evidently driven by spontaneously generated internal patterns. The neural structures continue to develop postnatally when the sensory systems are exposed to stimuli from the environment. In this process, prenatal training may give the neural system the appropriate bias so that it can learn reliably under changing environmental stimuli. This paper evaluates the hypothesis that an artificial learning system can benefit from a similar approach, consisting of initial training with patterns from an evolved generator followed by training with the actual training set. Competitive learning networks were trained in recognizing handwritten digits in three ways: through environmental learning only, through evolution only, and through prenatal training with evolved pattern generators followed by environmental learning. The results demonstrate that the evolved pattern generator approach leads to better learning performance, suggesting that complex systems can be constructed effectively in this way.

Categories and Subject Descriptors

I.2.6—Connectionism and Neural Nets; I.2.0—Cognitive Simulation; I.5.2—Classifier Design and Evaluation

General Terms

Algorithms, design, experimentation, performance

Keywords

Competitive learning, evolutionary computation, pattern generators, spontaneous cortical activity, self-organization, complex systems

1. INTRODUCTION

The tradeoff between bias and variance is a well-known issue in machine learning [11, 32]. Given a 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 often a very large number of possible mappings consistent with the training set, and they result in 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. That way, the outputs for new examples are likely to be correct. Also, the same mapping is selected with different training sets and even when the training examples are noisy, i.e. the learner will have a low variance.

Unfortunately, it is usually not clear what the right bias is, making it necessary to make the bias weaker. Which mapping will be selected then depends more on the training data. As a result, the variance is increased: The selection of the mapping becomes unpredictable, determined based on which examples were included in the training set and the noise in those examples. Choosing an appropriate point in the bias-variance tradeoff therefore depends on how much is known about the problem in advance.

In this paper, a method for constructing artificial learning systems that have both good classification accuracy and good learning ability is proposed based on inspiration from nature on how it handles this bias-variance tradeoff. As a result of evolution, nature has created neural learning systems in the brain that undergo a developmental process utilizing both genetic and environmental information, as opposed to a pure hardwiring of neural connections or pure learning induced by the environment.

Experiments since the 1960s have shown that the environment can have a large effect on the structure and function of the early visual areas of the brain (see [23] for a review). For instance, if kittens are raised in environments consisting of only vertical contours during a critical period, most of their primary visual cortex neurons become responsive to vertical orientations [5]. However, there is also significant evidence on the contrary, suggesting that visual cortex structure is also genetically determined. For example, it has been known for a long time that individual orientation-selective cells exist in newborn kittens and ferrets even before they open their eyes [6].

How can the same circuitry be both genetically hardwired, yet also capable of significant learning and adaptation based on the environment? The recent discovery of spontaneous activation provides an important clue: 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 visual cortex, the retina, the auditory system, and the spinal cord (see [25, 27, 33] for reviews). This activity may express a genetic bias within a system that is designed to learn from the environment. The genetic information is represented in the same way at the neural level: as patterns of activity in the input seen by a

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

GECCO'05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

brain area. The same activity-dependent learning mechanisms that can explain postnatal learning may simply be functioning before birth, driven by activity from internal instead of external sources. The genome thus needs to specify only a pattern generator, a mechanism capable of producing visual-like patterns, rather than specifying individual connections.

These results from nature suggest that learning driven by both internal and external inputs can be used to build complex, plastic, robust structures that would be too complex to determine directly genetically and too fragile to learn from external inputs. The main hypothesis studied in this paper is that artificial learning systems can achieve the same benefits as biological learning systems by training using generated patterns in addition to actual training data.

In order to test this hypothesis, a competitive learning neural network architecture is 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 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 training data.

The results show that competitive learning alone is much weaker than the other two methods. Although direct evolution eventually achieves slightly higher classification accuracy than pattern generation, pattern generation reaches a high level of performance in much fewer generations. In real-world applications as well as in nature, such slow learning might not be practical, making pattern generation a particularly useful way of constructing complex systems.

The remainder of the paper is organized as follows. Section 2 reviews biological and computational background on pattern generation and on interaction of 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 systems, is formulated in Section 3. The learning architecture and the algorithms used to evaluate the hypothesis are presented in Section 4, and experimental results on the handwritten digit recognition task in Section 5. The mechanism by which prenatal training helps avoid local minima in postnatal learning is analyzed in Section 6, and possible directions for future work are presented in Section 7.

2. BACKGROUND AND RELATED WORK

In the following subsections, the biological motivation for pattern generation is reviewed, followed by computational studies verifying the advantage of combining learning with evolution and the effects of generating internal patterns.

2.1 Biological Motivation

Many researchers have argued that 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. [29] for a review). A number of classic experiments by Hubel, Wiesel and other researchers showed that altering the visual environment, especially during a critical period of early life, can drastically change the organization of the visual cortex [16, 17]. Such experiments indicate that visual inputs are crucial for normal cortical organization, and suggest that the cortex tunes itself to the distribution of visual inputs.

However, there are two problems with this result. First, self-organization takes time, and the animal would not be able to act on visual input until the process is almost complete. Second, 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.

In contrast, visual development in nature is highly stable, and the visual cortex of most animals is partially organized already at birth (or eye-opening). For example, newborns can already discriminate between two orientations, and animals have neurons and brain regions selective for particular orientations even before their eyes open [8, 31]. These same areas then later adapt consistently to visual experience to construct an adult visual system [6]. Such robustness could be achieved with a specific, fixed genetic blueprint, but there is not enough information available in the genome to represent it.

Recent experimental findings in neuroscience suggest that nature may have found a clever way to utilize self-organization to achieve the same result. Developing sensory systems are now known to be spontaneously active even before birth, i.e. before they could be learning from the environment (see [25, 27, 33] for reviews). This spontaneous, internal activity may guide the process of cortical development, acting as genetically specified training patterns for a learning algorithm.

Although spontaneous activity can arise at all levels of the developing visual system [20, 25, 27], retinal waves are a well-known example and likely to play a role in prenatal self-organization. In the developing retina of e.g. cats and ferrets, internally generated activity occurs as intermittent, local waves across groups of ganglion cells [21]. The waves begin before photoreceptors have developed [19], 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 [7, 10, 30]. Like visual images, these waves are locally coherent in space and time and thus they could act as training input for the developing visual cortex [28].

For a biological species, such training patterns can guarantee that each organism has a rudimentary level of performance from the start. Such training would also ensure that initial development does not depend solely on the details of the external environment. Thus, internally generated patterns can preserve the benefits of a blueprint, within a learning system capable of much higher complexity and performance.

2.2 Computational Studies

The role of spontaneous activity in development has been tested computationally using a computational map model of the visual cortex called HLISSOM [3, 4, 22]. Using prenatal training patterns similar to retinal waves, two developmental phenomena were studied: (1) How orientation maps 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 structure that matches known biological orientation processing circuitry in animals. When trained with three-dot input patterns, the model learned 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.

A related idea that has been explored computationally by several researchers is combining evolution with learning from the environment. In many such approaches, connection weights are evolved

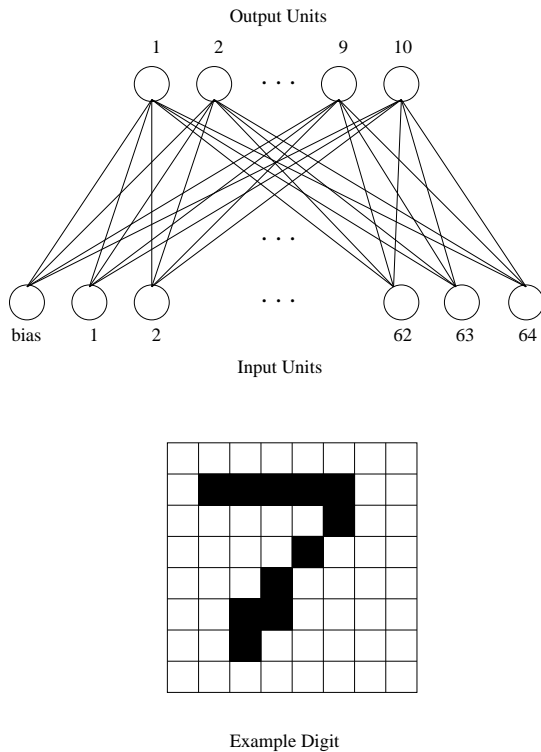


Figure 1: The architecture of the competitive learning network. Binary activations from the input pattern consisting of 64 pixels are fed to the input units of the network, which also contains a bias unit. The 10 output units each correspond to a classification of the input as one of the 10 digits; the one with the highest activation is chosen as the answer of the network. During training, the weights of this unit are adjusted towards the input pattern, making it more likely to win similar patterns in the future.

for a network that later learns. In such a system, evolution selects individuals with weight patterns that have the capacity to learn good performance, rather than individuals with good performance at birth [15, 24]. 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 [1]: learning influences evolution even though acquired characteristics are not inherited.

The above two ideas are brought together in this paper: The pattern generators are evolved to make postnatal learning as effective as possible, as will be described in the next section.

3. GENERAL HYPOTHESIS

How can the idea of internal pattern generation be utilized in constructing complex artificial systems? 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 goal without having to hard-code it into a particular, inflexible architecture. In essence, the engineer will bias the learning system with generated patterns, allowing it to solve problems that would otherwise be difficult for learning systems. For example, simple patterns can be learned before real data,

thereby avoiding local minima in the search space of solutions [9]. Such bootstrapping may also allow the designer to avoid expensive and laborious manual collection and/or tagging of training datasets, as in tasks like handwriting recognition and face detection. For instance, a three-dot training pattern could be used to detect most faces, and only the patterns that were not detected would need to be tagged manually.

More significantly, the pattern generator could be constructed automatically using evolutionary algorithms (EA). In this approach, domain-specific knowledge necessary to design the generator by hand would not be 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; however, often such knowledge can only be obtained through trial and error, and it would be better to have an algorithm to do it automatically. Indeed, the self-organizing system, the pattern generator, and the EA together can be considered a single general-purpose adaptive algorithm.

What benefits would such a system have over other adaptive systems, such as EAs or learning networks alone? Essentially, the combination of learning and evolution represents a balance between adaptation at different time scales (i.e. it determines a proper tradeoff between bias and variance; Section 1). Short-term learning allows an individual network to become well suited for the particular tasks on which it is tested. Long-term adaptation (i.e. selection by the EA) can ensure that short-term learning does not reduce generality. For instance, the EA can select training patterns to ensure that a system is able to handle events that occur rarely, yet are vitally important over the long term. For example, a computer vision system for detecting faults in manufactured devices can be trained both on the typical cases of correct devices, plus specifically generated examples of faults and defects. The EA can also select pattern generators that get the system “in the ball-park,” to increase the chance that learning will succeed. Thus, by combining EAs and learning using pattern generators, it should be possible to evolve systems that perform better than using either approach alone.

4. TESTING THE HYPOTHESIS

The hypothesis is tested in this paper with the task of constructing a single-layer artificial neural network to identify the handwritten digits 0 to 9. The network receives the handwritten digits as its input and produces the classification of each digit as its output. Classification accuracy and learning effort is compared for environment-based learning, direct evolution, and evolved pattern generator approaches.

4.1 Competitive Learning

The learning algorithm used is competitive learning [13, 26]. Even though other neural network learning algorithms may be more powerful in general classification tasks, competitive learning is a good model for learning in biological systems. It is based on Hebbian adaptation of synaptic efficacies [14] and it is a self-organizing, unsupervised algorithm. It therefore captures the kind of learning that is likely to occur in early development, as is appropriate for testing the hypothesis.

The digits are written in an 8×8 grid of pixels (Figure 1). 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 so that the squares of the weights of each output unit

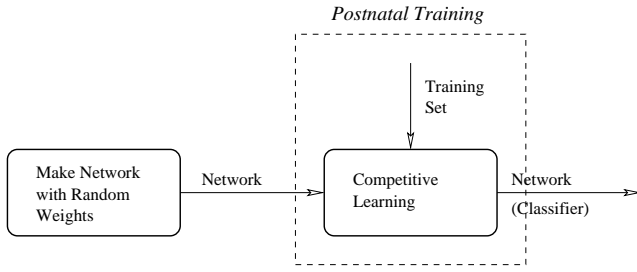


Figure 2: Inducer1 produces a classifier network by performing competitive learning on the set of training examples. This method corresponds to postnatal learning (there is no prenatal learning phase).

sum to one:

$$w_{ij} = \frac{w_{ij}}{\sqrt{\sum_u w_{uj}^2}}. \quad (1)$$

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

$$s_j = \sum_i w_{ij} x_i. \quad (2)$$

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

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

where η is the learning rate. After the update, the weights of this unit are again normalized so that their squares sum to one. This process constitutes a basic competitive learning method that is at the core of many unsupervised learning algorithms [2, 12, 18].

4.2 Experimental Hypothesis

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

4.2.1 Inducer1

First, a network is trained using competitive learning alone (Figure 2). This process involves initializing the network with random weights and training it using a set of examples until its weights converge. This method corresponds to an organism whose learning is entirely postnatal, without any genetically determined biases.

4.2.2 Inducer2

In the second method (Figure 3), the connection weights of the network are evolved directly; there is no competitive learning phase at all. The architecture of the networks is the same as that in Inducer1. The classification accuracy of each network on the training set is estimated to compute its fitness. 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.

4.2.3 Inducer3

Constructing the third classifier network involves evolving a pattern generator (Figure 4). Each generator produces a set of patterns on which a network is trained (using competitive learning) during a prenatal training phase. After the prenatal learning is complete, the resulting network is trained on the training set during a 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 pop-

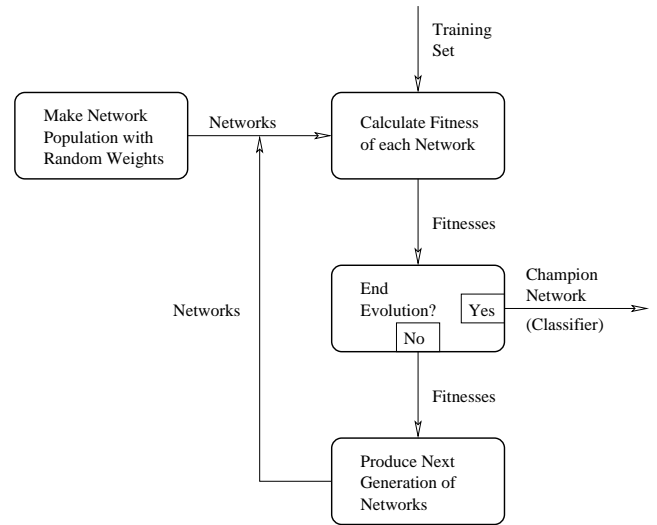


Figure 3: Inducer2 produces a classifier network by evolving the weights of a network that has the same architecture as that produced by Inducer1. There is no prenatal nor postnatal learning in this approach.

ulation have been evaluated in this manner, the next generation of pattern generators is formed. 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 expected outcome of these comparisons is that the direct EA (Inducer2) would require a prohibitively large number of iterations, because it has to search in an extremely high-dimensional space of network weights. The environmentally driven learner (Inducer1), on the other hand, is likely to get stuck in suboptimal local minima, because it will start far from the desired solution, without any bias toward it. In contrast, the pattern-generator-driven system (Inducer3) should be able to discover a solution quickly because it only needs to evolve a small number of parameters of the generator.

4.3 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 + 1 bias) associated with an output unit. The weights are floating point values between 0 (inclusive) and a specified maximum bound (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.

The weights are mutated by applying Gaussian perturbations on the floating-point weight 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.

Mating is done by selecting a partner from the population randomly and performing uniform crossover to produce an offspring. Crossover takes place 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

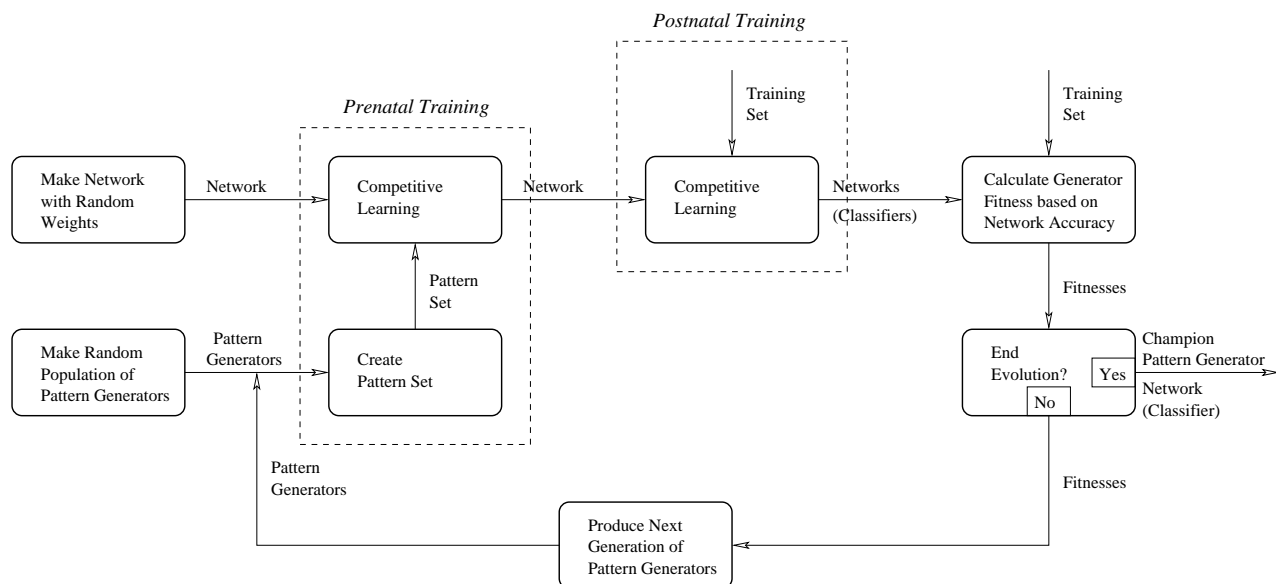


Figure 4: Inducer3 evolves pattern generators in its main computational loop. The patterns produced by each generator train a competitive learning network in the prenatal training phase. This phase is followed by a postnatal competitive learning phase where the network is trained further with a training pattern set, like Inducer1. The evolution run produces two results: the champion pattern generator and the classifier network trained using it.

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. If the offspring has a higher fitness than a parent, 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.

4.4 Evolving Pattern Generators

Each prenatal training pattern in the Inducer3 approach is a single two-dimensional Gaussian of floating point values between 0 and 1. Each pattern generator is coded as a collection of such Gaussians. Each gene in the chromosome specifies one Gaussian based on six floating-point values: 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; d_y , the displacement in the y -direction; and f , the frequency (or probability) with which the Gaussian appears among the generated pattern set.

The values for σ_x , σ_y , d_x , and d_y are constrained so that the generated patterns lie within the 8×8 pixel grid. The θ parameter varies within $[0, \pi)$, which covers all possible orientations of the Gaussian (since orientation is π -periodic). The f parameter varies within $[0, 1)$.

A specified number of genes are packed in a chromosome, which then represents an individual pattern generator in the population. Selection, mutation, and crossover (at the levels of Gaussians and their parameters) are performed as in the Inducer2 approach.

4.5 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.

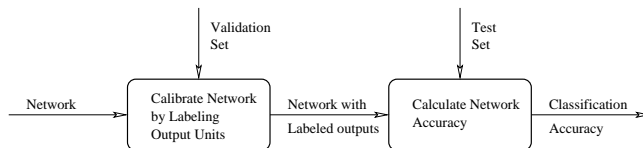


Figure 5: The procedure for calculating the percentage of examples correctly recognized from a test set involves first labeling the output units of the network with the digits they have learned to recognize using a validation set. Once the output units have been labeled, the classification accuracy of the network on a test set can be determined.

4.5.1 Percentage Correct

The first method calculates the percentage of examples that are correctly recognized from the test set (Figure 5). Since competitive learning is an unsupervised method, the network does not have labels on its 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. This measure is used to compare the accuracy of the final networks produced by all three approaches.

4.5.2 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.

Table 1: 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	0.4
Mutation rate		0.9	0.5
Mating rate (fields)		0.1	0.2
Mating rate (genes)		0.05	0.1
Max. network weight		1.0	
Max. displacement			8.0
Max. standard deviation			3.0
Number of genes		10	7
Size of pattern set			100
Population size		25	25
Max. generations		9000	9000

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.

5. EXPERIMENTS

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 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 labeling the output units), and the remaining nine parts for training.

Suitable values for the evolution and competitive learning parameters were determined experimentally prior to the experiment (Table 1). Competitive learning in Inducer1 and Inducer3 was continued until all weights changed less than 10^{-3} in an epoch, or up to a maximum number of epochs. The network with the final converged weights was taken as the result. The training examples were presented in a different random order in each epoch.

Similarly, evolution in Inducer2 and Inducer3 was continued until the fitness on the validation set leveled off, i.e. did not improve by more than 0.009 over the next 1000 generations. The champion of this generation was then selected as the result of evolution.

The average classification accuracy over the 11-fold crossvalidation experiment is shown in Table 2. The differences seen in the table are statistically significant as measured by pair-wise Student’s *t*-test on the crossvalidation runs. As expected, Inducer1 is significantly less accurate than either Inducer2 or Inducer3. Inducer2 achieves a slightly better accuracy than Inducer3, but does so in a much larger number of generations than Inducer3. This conclusion is illustrated in Figure 6, where fitness and accuracy for Inducer2 and Inducer3 are plotted over time for one example evolution run. Inducer3 achieves a good level of performance much earlier than Inducer2.

Table 2: Average classification accuracies

	Inducer1	Inducer2	Inducer3
Average Accuracy(%)	63.37	74.47	71.12
Average Generations		2152	202

In real-world applications as well as in nature, where the search spaces are much larger, it may not be practical to run Inducer2 long enough to achieve its best accuracy. In such domains, Inducer3 is a preferable alternative for constructing complex systems.

6. EFFECT OF PRENATAL TRAINING

How does prenatal learning in Inducer3 allow postnatal learning to perform better than in Inducer1? It turns out that only a few output units learn anything during prenatal training; the other units maintain their initial random weights. Such focused learning establishes just the right bias for postnatal learning so that the system as a whole does not get stuck in local minima like Inducer1 does.

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. To some extent, this indeed happens. For example, during one sample run, one of the output units became biased toward learning digit 7 because of high weights on connections to the top right corner of the input grid, whereas another unit became biased toward 9 because of high weights on connections to the top left corner. These biases were subtle but allowed the network to disambiguate between the digits 7 and 9, which were particularly problematic for Inducer1.

However, such separation is not clear in other cases, and overall the prenatally trained network is quite different from the final network, both in terms of weights and classification behavior. In fact, the confusion matrix after prenatal training often shows a significant clustering of examples to one output unit, instead of separation of each digit to a different unit. The digits are eventually separated during postnatal training, and it turns out that the prenatal clustering plays a crucial role in this process.

Without such clustering, all units have random initial biases. One and the same unit is likely to win most of the examples of similar digits, because the competing units have very different biases and win other kinds of digits. Consequently, the units do not learn to identify the specific features that differentiate the similar digits. This phenomenon is seen frequently with Inducer1, particularly with digits 7, 8 and 9, and it never recovers from it.

In contrast, in Inducer3 one of the output units forms a cluster and a few competing units have just the right biases. These competing units initially represent only the subtle differences between similar digits in the cluster, and win only a few extreme examples. As these units adapt to examples during postnatal learning, they maintain these differences while they gradually become less extreme. In this process, they eventually learn an effective representation for the entire digit category that is distinct from the other similar categories. In this way, evolution of pattern generators discovers a starting point from which it is easy to learn good classification, rather than a starting point that performs well already but from which further progress is difficult.

7. DISCUSSION AND FUTURE WORK

As the classification accuracies in Table 2 show, Inducer1 is the least accurate of the three methods. Since its weights are initialized randomly, the network is not biased in favor of any learning path. Without a proper bias, it regularly gets stuck in local minima. On

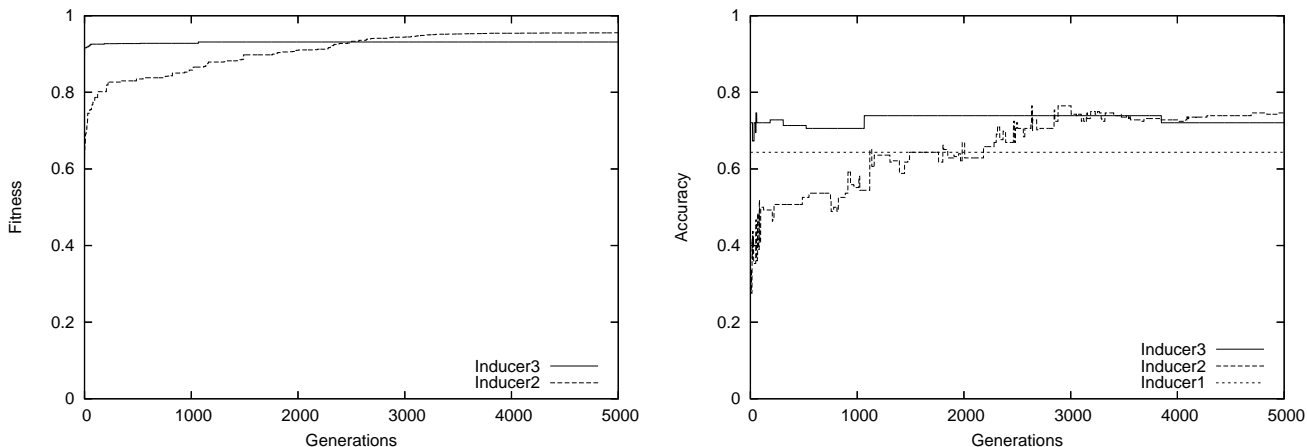


Figure 6: Improvement in fitness and accuracy during evolution of Inducer2 and Inducer3 for one example evolution run. The final accuracy of Inducer1 is also shown for comparison. The pattern-generator-based learner (Inducer3) reaches a good level of performance much earlier than direct evolution learner (Inducer2), confirming that it is an effective strategy for constructing complex systems.

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. The result is a significantly better classification accuracy for Inducer3.

When constructing complex systems such as those in biology and robotics, fitness evaluation of an individual requires it to live a lifetime and interact with its environment. The individual can either be fixed as with Inducer2, or it can learn from the environment and improve its fitness, as with 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 also reaches a good level of accuracy eventually, if evolution is allowed to proceed long enough. Although its final accuracy slightly exceeds that of Inducer3, Inducer3 reaches a good level of performance in much fewer generations than Inducer2, as shown by the fitness and accuracy plots in Figure 6. In other words, constructing a good digit recognition network is much easier 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. Thus, the results confirm the hypothesis that prenatal training with an evolved pattern generator is an efficient way to construct complex systems.

The experiments described in this paper are based on a simple competitive learning network. The domain of handwritten digit recognition of 10 different digits is also simple compared to real-world application domains. Future work will focus on applying these techniques to more complex learners and domains, where the benefits of Inducer3 are expected to be even more pronounced. If successful, this effort will ultimately pave the way for the engineering of complex systems that are otherwise difficult or impossible to construct.

8. 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 proposes the same approach for building complex systems more generally. The hypothesis is that pretraining a system with patterns from an evolved generator will make the learning from the actual data easier. Experiments in the handwritten character recognition domain supported this hypothesis, suggesting that complex systems can be effectively constructed in this way.

9. ACKNOWLEDGMENTS

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

10. REFERENCES

- [1] J. M. Baldwin. A new factor in evolution. *The American Naturalist*, 30:441–451, 536–553, 1896.
- [2] H. B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.
- [3] J. A. Bednar and R. Miikkulainen. Learning innate face preferences. *Neural Computation*, 15(7):1525–1557, 2003.
- [4] J. A. Bednar and R. Miikkulainen. Constructing visual function through prenatal and postnatal learning. In D. Mareschal, M. H. Johnson, S. Sirois, M. Spratling, M. S. C. Thomas, and G. Westermann, editors, *Neuroconstructivism, Vol. 2: Perspectives and Prospects*. Oxford University Press, Oxford, UK, 2005. In press.
- [5] C. Blakemore and G. F. Cooper. Development of the brain depends on the visual environment. *Nature*, 228:477–478, 1970.
- [6] C. Blakemore and R. C. van Sluyters. Innate and environmental factors in the development of the kitten’s

- visual cortex. *The Journal of Physiology*, 248:663–716, 1975.
- [7] M. Catsicas and P. Mobbs. Waves are swell. *Current Biology*, 5(9):977–979, 1995.
- [8] 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*, 279:566–570, 1998.
- [9] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. *Rethinking Innateness: A Connectionist Perspective on Development*. MIT Press, Cambridge, MA, 1996.
- [10] 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*, 272:1182–1187, 1996.
- [11] S. Geman, E. L. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
- [12] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–63, 1987.
- [13] S. Haykin. *Neural Networks, A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey, 1999.
- [14] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, Hoboken, NJ, 1949.
- [15] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, 1:495–502, 1987.
- [16] D. H. Hubel and T. N. Wiesel. Sequence regularity and geometry of orientation columns in the monkey striate cortex. *The Journal of Comparative Neurology*, 158:267–294, 1974.
- [17] 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*, 278:377–409, 1977.
- [18] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.
- [19] 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*, 87:2861–2864, 1990.
- [20] 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*, 69:1–11, 1995.
- [21] 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*, 252:939–943, 1991.
- [22] R. Miiikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer, Berlin, 2005. In press.
- [23] J. A. Movshon and R. C. van Sluysters. Visual neural development. *Annual Review of Psychology*, 32:477–522, 1981.
- [24] S. Nolfi, J. L. Elman, and D. Parisi. Learning and evolution in neural networks. *Adaptive Behavior*, 2:5–28, 1994.
- [25] M. J. O’Donovan. The origin of spontaneous activity in developing networks of the vertebrate nervous system. *Current Opinion in Neurobiology*, 9:94–104, 1999.
- [26] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 151–193. MIT Press, 1987.
- [27] F. Sengpiel and P. C. Kind. The role of activity in development of the visual system. *Current Biology*, 12(23):R818–R826, 2002.
- [28] C. J. Shatz. Impulse activity and the patterning of connections during CNS development. *Neuron*, 5:745–756, December 1990.
- [29] C. J. Shatz. The developing brain. *Scientific American*, 267(3):61–67, 1992.
- [30] C. J. Shatz. Emergence of order in visual system development. *Proceedings of the National Academy of Sciences, USA*, 93:602–608, 1996.
- [31] A. Slater and S. P. Johnson. Visual sensory and perceptual abilities of the newborn: Beyond the blooming, buzzing confusion. In F. Simion and G. Butterworth, editors, *The Development of Sensory, Motor and Cognitive Capacities in Early Infancy: From Perception to Cognition*, pages 121–142. Psychology Press, East Sussex, UK, 1998.
- [32] P. Utgoff and T. Mitchell. Acquisition of appropriate bias for inductive concept learning. In *Proceedings of the Second National Conference on Artificial Intelligence*, pages 414–417, Menlo Park, CA, 1982. AAAI Press.
- [33] R. O. L. Wong. Retinal waves and visual system development. *Annual Review of Neuroscience*, 22:29–47, 1999.