

# Establishing an Appropriate Learning Bias Through Development

Vinod K. Valsalam

*Department of Computer Sciences  
The University of Texas at Austin  
Austin, TX 78712 USA  
vkv@cs.utexas.edu*

James A. Bednar

*Institute for Adaptive and Neural Computation  
The University of Edinburgh  
5 Forrest Hill, Edinburgh, EH1 2QL UK  
jbednar@inf.ed.ac.uk*

Risto Miikkulainen

*Department of Computer Sciences  
The University of Texas at Austin  
Austin, TX 78712 USA  
risto@cs.utexas.edu*

**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. Such activity is genetically controlled and has been proposed to give the neural system an appropriate bias so that it can learn reliably from complex environmental stimuli. This paper demonstrates this idea computationally. A competitive learning network is trained with hand-designed patterns during a “prenatal” developmental phase, and its classification performance in a line categorization task is significantly affected as a result. Plotting and analyzing the network weights during various stages of the learning process reveals the complex dynamics through which the bias is established, and suggests that evolution might be necessary to discover the appropriate pattern generators automatically. This approach is expected to be useful in building complex artificial systems, such as the learning system of a robot with uninterpreted sensors and effectors.

**Index Terms**—Competitive learning, bias and variance, pattern generator, spontaneous activity, self-organization, prenatal development, developmental robotics

## I. INTRODUCTION

The tradeoff between bias and variance is a well-known issue in machine learning [6, 20]. A strong bias that matches the problem being solved and minimizes variance is desired, but the right bias is hard to determine, and could even be as hard as solving the problem itself [6]. The bias is also difficult to express in a learning system. In the perceptual systems of animals, internally generated spontaneous activity acting as training patterns for the prenatal self-organization of many cortical and subcortical sensory areas (see [13, 18] for reviews) may be one of the developmental mechanisms by which nature sets the right bias.

This paper demonstrates this idea computationally. A competitive learning neural network is used for classifying vertical and horizontal lines. Training the network with a hand-designed pattern in a developmental phase prior to learning the actual task gives the network a good initial bias, enabling it to classify the lines better. When the different stages in the learning process are visualized by plotting the network weights, the learning process is found to be dynamic and highly sensitive to initial bias. The results show that the

prenatal biasing approach is effective, and suggest that a mechanism like evolution is necessary to discover appropriate pattern generators for more complex learners and application domains.

The paper is organized as follows. Section II reviews the biological and computational background on pattern based development and learning. The competitive learning architecture and the line categorization task used for the experiments are presented in Section III, and the results analyzing the effect of prenatal biasing on classification accuracy in Section IV. Future work in utilizing this idea in building artificial complex systems is presented in Section V.

## II. BACKGROUND AND RELATED WORK

In the following subsections, the biological motivation for establishing bias through prenatal self-organization from generated patterns is reviewed, and previous computational work involving learning from generated patterns is discussed.

### A. Biological Motivation

Neuroscience researchers have found significant evidence for prenatal bias in the visual cortex of animals in the form of genetically determined structures [8, 10]. For example, experiments have shown that orientation processing structures exist prior to visual experience in ferrets and kittens [4, 7].

The large-scale structures of the brain, such as the division into different brain areas, are constructed primarily through genetically determined chemical gradients (see [8] for a review). The gradients are largely unaffected by environmental stimuli, making the bias very strong. However, for fine-scale structures such as those for orientation processing, another possible source of genetic bias was discovered recently: the spontaneously generated patterns of internal neural activity observed in many cortical and subcortical sensory areas, such as the visual cortex, the retina, the auditory system, and the spinal cord (see [13, 18] for reviews). This activity may express a genetic bias within a system that is designed to learn from the environment [5, 19]. That is, 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

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.

pattern generator, i.e. 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.

### B. Computational Modeling of Pattern Generators

The pattern-generation hypothesis was previously tested using a computational map model of the visual cortex called HLISSOM [2, 12]. Two developmental phenomena were studied: (1) How orientation processing circuitry develops 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 pattern generator approach was studied as a machine learning technique for improving the performance of artificial learning systems [21]. Instead of designing the generator by hand, generators were evolved computationally to provide a suitable bias for learning to recognize handwritten digits in a competitive learning neural network. The approach indeed improved the performance of such networks in this task. However, because the domain is rather complex, it turns out difficult to analyze the learning process to explain why the particular pattern generators discovered by evolution were effective.

Therefore, a simple vertical and horizontal line categorization task is used in this paper along with hand-designed prenatal training patterns to explore the mechanisms behind how prenatal biasing of the network affects further learning. The categorization task and the competitive learning network used in this study are described in the next section.

## III. METHOD

The main goal of this paper is to demonstrate how the bias of an artificial learner can be manipulated by an approach similar to the internally generated patterns in nature. In order to facilitate detailed understanding of the underlying process,

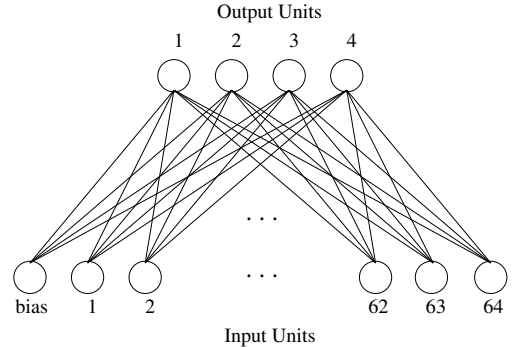


Fig. 1. *The architecture of the competitive learning network.* The binary activations from the input pattern consisting of 64 pixels are fed to the input units of the network, which also contains a bias unit. The four output units each correspond to a classification of the input as one of the four line categories in Fig. 2; 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 that unit more likely to win similar patterns in the future. Because of its simplicity, this network is a good candidate to study the process of prenatal biasing.

the task and learning algorithm chosen are deliberately simple, as is described below.

### A. Competitive Learning

The learning algorithm used in this study is competitive learning [17]. Even though other algorithms may be more powerful in classification tasks in general, competitive learning is a good choice for this study 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 [9], 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 pattern design simpler by not having to produce targets for the inputs for prenatal learning.

The competitive learning network (Fig. 1) must learn to categorize vertical and horizontal lines drawn on an  $8 \times 8$  grid of pixels. The lines fall into four categories as described in Section III-B. The inputs to the network consist of the binary activations at the 64 grid locations and a bias unit. The network has 4 outputs, one for each of the 4 categories 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:

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

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

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

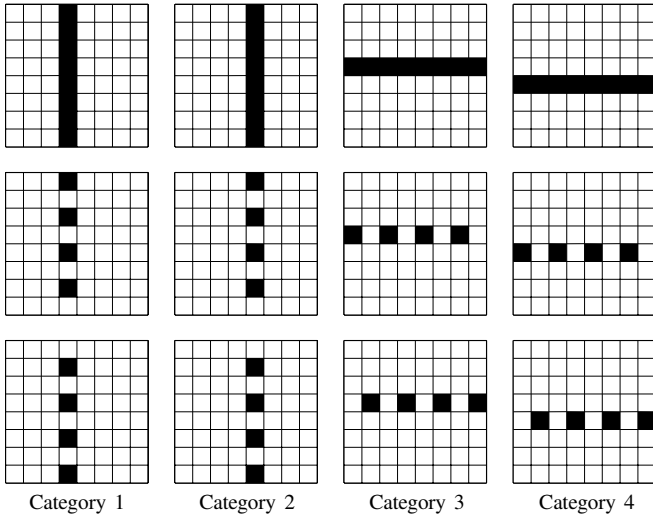


Fig. 2. Training examples for categorization of vertical and horizontal lines. There are four categories and three examples in each category: a solid line and two complementary dotted lines. This design makes the effect of prenatal training explicit, as shown in Figs. 3 – 7.

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

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

where  $\eta$  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 [1, 11].

### B. Line Categorization Task

The task the network is trained to perform consists of categorizing vertical and horizontal lines drawn in the  $8 \times 8$  pixel grid (Fig. 2). 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. The training set consists of 12 examples 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, learning can fail for two reasons. First, because a vertical and a horizontal line share a common pixel, if an output unit exists with particularly high weights on that pixel, the learning algorithm may learn to map them both to that unit. Second, the learning may also fail if an output unit has initial weights that allow it to win examples of two different 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 stronger to both of them.

Thus, the categorization task is designed to permit competitive learning to achieve perfect classification, while at the same time manifesting its weakness of getting stuck in

TABLE I  
PARAMETERS FOR LEARNING

Prenatal learning rate	0.005
Prenatal max. epochs	400
Postnatal learning rate	0.0005
Postnatal max. epochs	5000

local optima. It also allows prenatal learning to affect the final classification performance by manipulating the initial weight biases. Experimental results showing these effects are described in the following section.

## IV. RESULTS

In this section, the results of competitive learning without an intentional initial bias (i.e., random initial weights) are described first. Second, how classification performance can be improved by prenatal training with a hand-designed pattern that gives the network a good initial bias is analyzed. The effects of using a bad prenatal training pattern on performance are shown in the end.

### A. Experimental Setup

Suitable values for the competitive learning parameters were determined experimentally prior to the experiment (Table I). Competitive learning was continued until all weights changed less than  $10^{-5}$  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.

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  $8 \times 8$  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). Such weight plots are used in the following subsections to visualize the convergence path taken by competitive learning and to analyze the effects of prenatal biasing on them.

In all the weight figures, if an output unit wins a large number of examples of a particular category from the test set (i.e., at least 75% of the largest number of wins for that category by any unit), then that category is shown on top of that unit. Thus, if the network is a good classifier, a different single category will be shown on top of every unit, indicating that each category is recognized as a separate class. In contrast, in a poor classifier, some units do not represent any categories at all, while other units represent multiple categories, and some categories are represented by multiple units.

### B. Learning without Prenatal Biasing

Competitive learning with random initial weights fails in the line classification problem in exactly the two ways described in Section III-B. (Fig. 3). 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

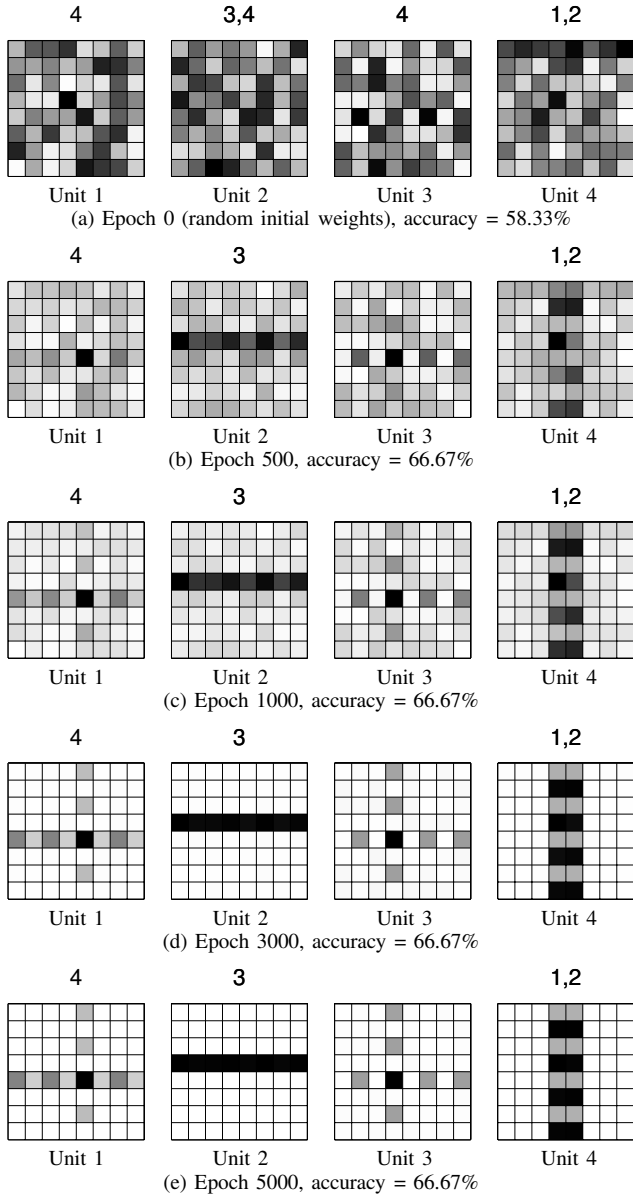


Fig. 3. Weights at various stages of learning without prenatal biasing to categorize vertical and horizontal lines. The network gets stuck in a local optimum, where unit 4 has strong representation for two categories, while unit 3 has none.

5000, only unit 2 has learned a clean category, recognizing exclusively all three examples of the 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. Learning with Prenatal Biasing

Competitive learning with an initial weight bias was applied to the same classification task using a manually constructed

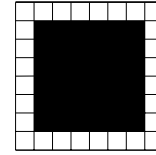


Fig. 4. A good pattern for prenatal training, designed to produce a beneficial clustering effect.

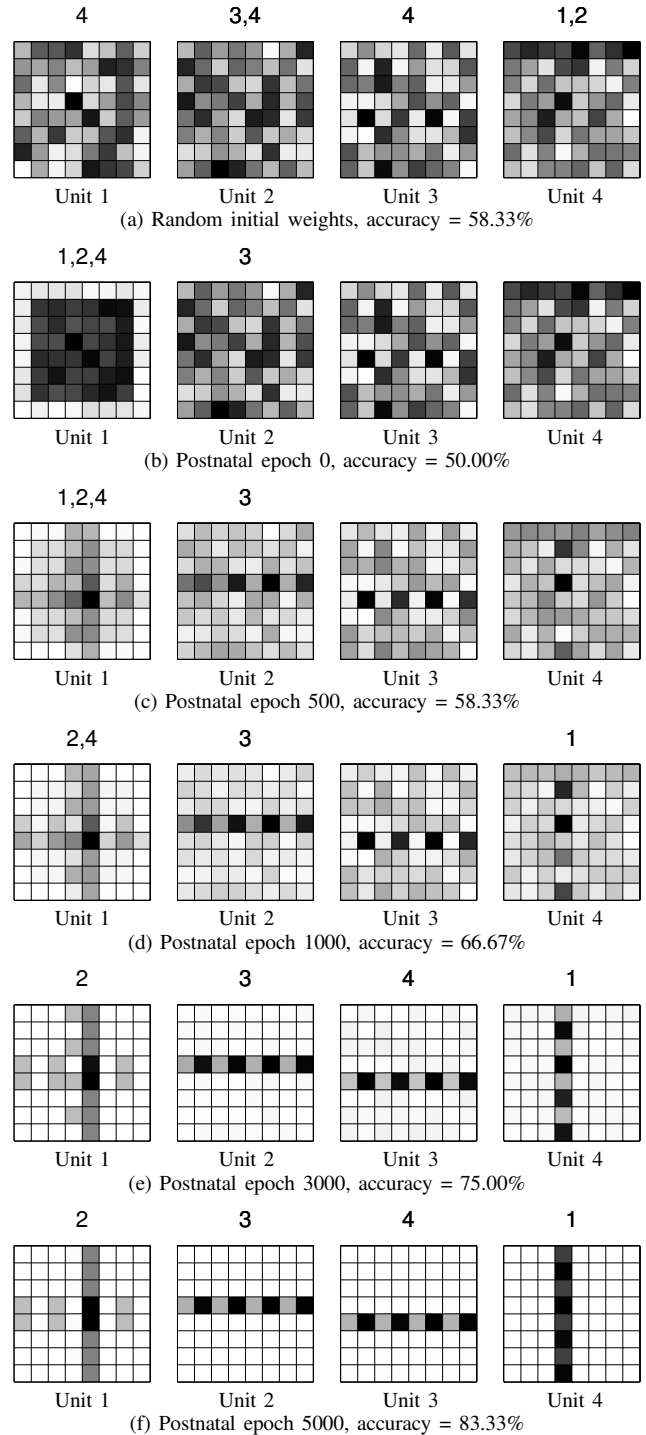


Fig. 5. Weights at various stages of learning with prenatal biasing to categorize vertical and horizontal lines. After prenatal training, unit 1 wins most of the patterns. Based on the remaining patterns, the other units develop distinct categories, and eventually the whole system converges to good categorization.

prenatal training pattern shown in Fig. 4. This pattern was designed to produce clustering of different categories on one unit, similar to the clustering of digits seen on some units in the handwritten digit recognition experiment of [21]. Snapshots of the 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 learning without prenatal training (Fig. 5).

The prenatal training pattern was learned by output unit 1; the other units maintained their random weights until the end of prenatal training (postnatal epoch 0). As in the experiment without prenatal training, the slight random biases of units 2, 3 and 4 gradually strengthen during postnatal training with examples. However, unlike in the experiment without prenatal training, 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 when no prenatal training was used.

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 the learner 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. 6, which has two additional active pixels on both sides, produces a clustering effect as well, it does not lead to successful postnatal learning (Fig. 7). 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%.

## V. DISCUSSION AND FUTURE WORK

Ahead of time it would have been difficult to guess that the pattern in Fig. 4 is successful while that in Fig. 6 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.

The most obvious way to establish an appropriate bias would be to separate each category to a different unit as much as possible already in prenatal training, so that postnatal training

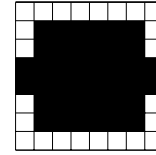


Fig. 6. A bad pattern for prenatal training, designed to produce a detrimental clustering effect.

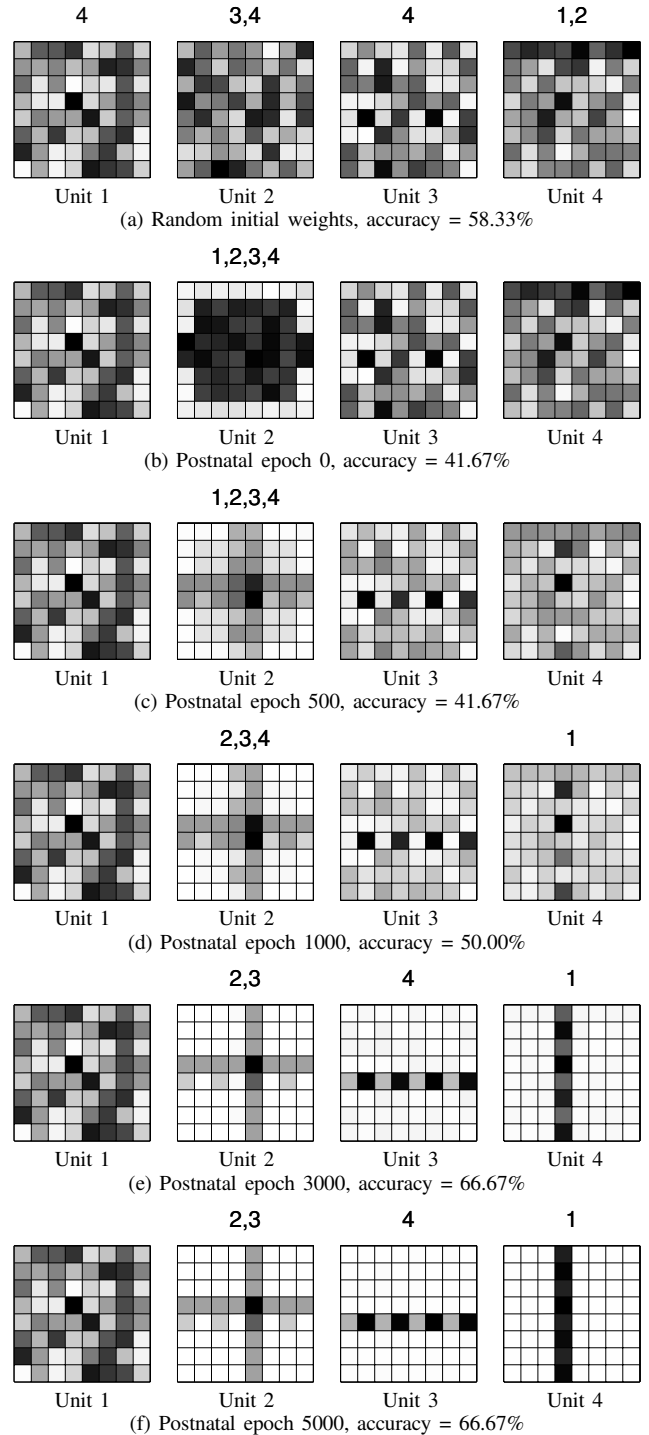


Fig. 7. Weights at various stages of learning when biased with a bad prenatal training pattern. The clustering effect is too strong, preventing unit 1 from learning anything. This example shows that the system is sensitive to prenatal training, and that the right patterns are difficult to discover, making evolution of pattern generators a useful approach.

would find it easier to complete the separation. However, this effect is typically not seen when trained with patterns from evolved generators as in the handwritten digit recognition task of [21]. Some units end up representing several different digit classes. The above analysis of the learning process using the simpler line categorization task shows how such seemingly counterproductive initial bias make postnatal learning easier.

In the line-categorization task it is easy to see how the initial bias established by prenatal training allowed competitive learning to avoid certain local minima. 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 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.

Self-organizing maps (SOM) [11] is a generalization of competitive learning where a neighborhood function is used to determine which units in the topological neighborhood of the winner should learn and by how much. If the neighborhood function is shrunk to include only the winner, a SOM defaults to a competitive learning network. Thus, like competitive learning, self-organizing maps are also susceptible to getting stuck in local optima, although to a lesser extent (see [16] for a discussion of the error function behavior of SOMs). Future work will explore how the performance of SOMs can be improved in a similar manner by learning the appropriate bias from generated patterns.

An interesting application of this approach with SOMs is in developmental robotics. The task involves a robot learning to navigate in its environment with no initial knowledge of what its sensors and effectors mean. The robot must first learn the properties of its own sensorimotor system, which it can do using domain-independent statistical learning methods [14]. However, the same task can in principle be accomplished without the *a priori* knowledge by using SOMs to learn perceptual features from continuous sensor inputs [3, 15]. The performance of these SOMs can potentially be improved by the prenatal biasing technique. The resulting learning systems are likely to be more flexible and easily adaptable to new environments and robots because the bias is established through self-organization and not hardwired by design.

## VI. CONCLUSIONS

Research on brain development in animals has led to insights on how patterns of prenatal spontaneous activity in the brain may be responsible for rudimentary cortical structures necessary for the system to learn efficiently from environmental inputs after birth. Such prenatal training may have been discovered by evolution to establish a proper bias in the learning system. This paper shows how a similar developmental approach based on learning from generated patterns

can give a competitive learning network the appropriate bias for improving its performance in a line categorization task. The results show that the right patterns may be hard to determine manually for more complex learners and tasks, and therefore suggests that a mechanism like evolution is necessary to discover them automatically.

## REFERENCES

- [1] H. B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.
- [2] James A. Bednar and Risto Miikkulainen. Pattern-generator-driven development in self-organizing models. In J. M. Bower, editor, *Computational Neuroscience: Trends in Research, 1998*, pages 317–323. New York: Plenum Press, 1998.
- [3] Douglas Blank, Deepak Kumar, Lisa Meeden, and James B. Marshall. Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems*, 36(2):125–150, 2005.
- [4] Barbara Chapman, Michael P. Stryker, and Tobias Bonhoeffer. Development of orientation preference maps in ferret primary visual cortex. *The Journal of Neuroscience*, 16(20):6443–6453, 1996.
- [5] Martha Constantine-Paton, Hollis T. Cline, and Elizabeth Debski. Patterned activity, synaptic convergence, and the NMDA receptor in developing visual pathways. *Annual Review of Neuroscience*, 13:129–154, 1990.
- [6] Stuart Geman, Elie L. Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
- [7] Imke Gödecke, D. S. Kim, Tobias Bonhoeffer, and Wolf Singer. Development of orientation preference maps in area 18 of kitten visual cortex. *European Journal of Neuroscience*, 9(8):1754–1762, August 1997.
- [8] Elizabeth A. Grove and Tomomi Fukuchi-Shimogori. Generating the cerebral cortical area map. *Annual Review of Neuroscience*, 26(1):355–380, 2003.
- [9] Donald O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, Hoboken, NJ, 1949.
- [10] Matthias Kaschube, Fred Wolf, Theo Geisel, and Siegrid Lowel. Genetic Influence on Quantitative Features of Neocortical Architecture. *The Journal of Neuroscience*, 22(16):7206–7217, 2002.
- [11] Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.
- [12] Risto Miikkulainen, James A. Bednar, Yoonsuck Choe, and Joseph Sirosh. *Computational Maps in the Visual Cortex*. Springer, Berlin, 2005.
- [13] Michael J. O’Donovan. The origin of spontaneous activity in developing networks of the vertebrate nervous system. *Current Opinion in Neurobiology*, 9:94–104, 1999.
- [14] David Pierce and Benjamin Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–227, 1997.
- [15] Jefferson Provost, Benjamin J. Kuipers, and Risto Miikkulainen. Self-organizing perceptual and temporal abstraction for robot reinforcement learning. In *AAAI-04 Workshop on Learning and Planning in Markov Processes*, 2004.
- [16] Helge Ritter and Klaus Schulten. Kohonen’s self-organizing maps: Exploring their computational capabilities. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN’88)*, pages 109–116, 1988.
- [17] David E. Rumelhart and David Zipser. Feature discovery by competitive learning. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 151–193. MIT Press, Cambridge, MA, 1986.
- [18] F. Sengpiel and P. C. Kind. The role of activity in development of the visual system. *Current Biology*, 12(23):R818–R826, 2002.
- [19] Carla J. Shatz. Impulse activity and the patterning of connections during CNS development. *Neuron*, 5:745–756, December 1990.
- [20] 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.
- [21] Vinod Valsalam, James A. Bednar, and Risto Miikkulainen. Constructing good learners using evolved pattern generators. In H.-G. Beyer et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005*, pages 11–18. New York: ACM, 2005.