

Laterally Connected Lobe Component Analysis: Precision and Topography

Matthew D. Luciw and Juyang Weng
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan, 48824
Email: {luciwmat, weng}@cse.msu.edu

Abstract—Due to the pressure of evolution, the brains of organisms need to self-organize at different scales during different developmental stages. In early stages, the brain must organize globally (e.g., large cortical areas) to form “smooth” topographic representation that is critical for superior generalization with its limited connections. At later stages, the brain must fine tune its microstructures of representation for “precision” – high-level performance and specialization. But smoothness and precision are two conflicting criteria. The self-organizing map (SOM) mechanisms of self-organization through isotropic updating and other published computational variants have dealt with global to local smoothing and lateral adaptation, but we show in our work that they are insufficient to deliver superior performance. In this paper, we introduce a combination of several mechanisms that, together, address these two conflicting criteria: lateral excitation through adaptive connections, explicit adaptive top-down connections (attention), dually-optimal lobe component analysis (LCA) for synaptic updating, simulated lateral inhibition through winners-take-all, and a developmental schedule that sets the number of winners, which decreases over time. Major performance improvements due to the combination of these mechanisms are shown in the reported experiments.

I. INTRODUCTION

DEVELOPMENT and adaptation of internal representation is a core issue for an autonomous intelligent system. It must be able to interpret its sensory input in order to perform successful actions, in many different contexts. The design of such a system requires an understanding of how such a sufficient representation of real environments can *emerge* (and adapt) in response to incrementally experienced stimuli. Non-adaptive designs (e.g., using hand-designed features) will not work since any such hand designed representational system lacks sufficiency of representation when it interacts with a complex dynamic environment. This is since a highly complex (realistic) environment cannot be modeled.

Towards this end, studying the mechanisms of neuronal learning and adaptation in cortex have lead to a better understanding of some computational mechanisms of development. Based on known cortical circuits [3], multilayer in-place networks (MILN) were developed with aim for high-level performance in real-world tasks. Learning in MILN occurs in a Hebbian manner, as opposed to traditional neural network learning using backpropagation. They were originally applied to invariant visual recognition [17], yet the network design is general purpose. MILN is hierarchically organized from

sensors (at the bottom) to motors (i.e., outputs – at the top). Each layer contains three types of connections – bottom-up, lateral and top-down.

The effect of the explicit top-down (which can be considered as feedback) connections is to influence the feature development and provide temporal context. The top-down connections during development led to a motor-biased organization of the internal representation — class grouping on the neuronal plane [9] — and more discriminating features, which led to better performance. When the top-down connections were enabled in the testing phase, the performance in a challenging object recognition task became nearly perfect [10], when the data was sequential (realistic videos). This was due to the temporal context from the motor, which biased lower level features to be detected next that were learned to correlate with the currently detected abstract class. Intuitively, in the testing phase, the network generated internal expectation using the recent more abstract context of experience. This internal context biased the recognition, but did not totally control it, meaning the network could recover from an incorrect internal context. In this paper, we examine the effect of lateral excitatory connections together with these top-down connections. The effect of explicit lateral connections is a direction that has yet to be fully explored in terms of performance, and the effect of lateral connections in conjunction with the top-down connections has not been examined.

Lateral connectivity has been handled in several different ways in various models. The self-organizing maps (SOM) [8] utilize an isotropic updating function with a scheduled (typically decreasing in size over time) scope. Based on SOM, LISSOM [13] used explicit lateral connections of both excitatory and inhibitory types. The initial scope of the excitatory weights was smaller than the inhibitory, and the excitatory scope and learning rates adapted throughout learning. The excitatory lateral connections helped lead to organization (nearby neurons represent similar or identical features), while major effects of lateral inhibition were to encourage development of different features and to decorrelate the output – leading to a sparse response. But LISSOM did not utilize an output, or “motor” layer, and thus was not tested for performance (e.g., error rate on a classification task). MILN has used the SOM-inspired ideas of winners-take-all and isotropic updating to simulate the lateral inhibition and excitation. The updating

was done in a 3×3 region around each winner neuron. To handle the scheduled scope of lateral excitation, we did not start with a large range function and decrease it; instead a “growing cortex” method was used [9], where the size of the map itself would increase at certain time points, and existing neuronal weights would be cloned to their new neighbors.

In this paper, we investigate the performance effects of adaptive lateral excitatory connections, and study how they can lead to both topography and precision, in conjunction with top-down connections. To our knowledge the performance effects of adaptive lateral connections have not yet been studied, especially in comparison to the other types of lateral excitation described above.

This paper is organized as follows. Section II presents background context. In Section III, the laterally connected lobe component analysis algorithm is presented. Section IV goes into more depth. It provides a more formal view on how LCA’s optimality can help in performance, and characterizes the effects of different computational lateral excitation mechanisms. Section V discusses experiments and results. Conclusions are presented in Section VI.

II. BACKGROUND

A. Biological Topographic Organization

Many cortices, such as the somatosensory, motor, and visual, have been observed to be organized topographically. The topographic organization of orientation selectivity in neurons in primary visual cortex (V1) is classically well known. However, this organization is not completely smooth at a lower scale: “the projection of the world into V1 is smooth and continuous on the macroscopic level, but jittery and occasionally discontinuous on the microscopic scale” ([7] – pg. 78). Maldonado *et al.* [11] measured the features detected in the pinwheel centers of V1 and found a larger variance of feature types, but not significantly larger bandwidths. This implies that features with low correlations can coexist nearby, and averaging of these unrelated nearby features does not necessarily occur. In somatosensory cortex, the organization seems to be more macroscopic than microscopic, with the famous organization of nearby body parts being nearby in cortex, but there is no predictability in receptive field shifts when moving from one minicolumn to the next within each module [6].

At a slightly larger scale, evidence suggests the cortex can develop two adjacent areas to be somewhat modular, but not strictly (e.g., there is no sharp boundary). The fusiform place area (FFA) and the parahippocampal place area (PPA) are adjacent (neighboring) [14] areas found in the inferior temporal (IT) cortex. FFA has been shown to respond preferentially to faces, while PPA has been shown to respond preferentially to images of places. In [14], the responses of each of the FFA and PPA areas in humans were measured using fMRI for stimuli of different morphs between faces and houses. The authors found that the two areas could be considered functionally different modules, since the peaks of the averaged morphed stimuli responses were in one area or the other – there were no areas that responded optimally to the morphed features.

However, it was also concluded that each of the two areas did not strictly define an optimal feature since some response for the morphs could be found in either area. But there is not a smooth interpolation between the two areas. How does the brain develop both an organized cortex and such separation of function?

B. Lateral Excitation and Top-Down Connections

It is generally assumed that lateral excitation is the impetus for topography to emerge. In cerebral cortex, many lateral connections are clustered close by the neuron from which they originate, to other nearby (i.e., neighboring) neurons. There are also strong long-range connections to neurons that detect similar features (e.g., similar or identical orientations). The close connections are not strictly isotropic, but instead “patchy”. Lateral excitation in artificial learning networks was traditionally handled using an isotropic function of updating emanating from the winner neuron(s); however, this is not biologically accurate.

In [13], it was shown that an orientation map with this patchy connectivity can develop by using *adaptive*, limited-range, lateral connections in a self-organizing cortical map, using stimuli such as oriented gaussians or natural stimuli. Lateral excitatory connectivity was also shown to be the cause of topographic organization of the map, meaning the features represented in a small area (containing a few neurons) tended to be similar. In the developed cortex, excitatory connections between neurons that represent dissimilar stimuli (not very statistically correlated) were not typically present, especially for long-range connections but even for nearby neurons. But any explicit performance advantage of this type of lateral connectivity over the isotropic type has not yet been shown to our knowledge.

We propose that a major use in terms of performance of the excitatory lateral connectivity is to develop representative modules – statistically correlated firing groups – and the adaptivity of the local connections serves to decrease the interference between different firing groups. But perhaps the lateral connections cannot do that job on their own, since some combinations of features may not be so correlated from the bottom-up, but will all be useful for a particular task. The top-down connections may also serve a role in the development of such modular feature areas.

The top-down connections are more numerous and generally more diffuse than the bottom-up, and they are assumed to serve a modulatory role, while the bottom-up connections are more directed information carriers [1]. Since top-down originate from the “higher” areas – the more abstract association cortices, and the motor areas, top-down connections could carry information that can be used to bias what features are currently important (internal attention) to the task at hand. The developmental effect is to reduce simultaneous firing for neurons that are not task-correlated, which will suppress lateral connectivity between them.

III. LATERALLY-CONNECTED LCA

The network applied here is a Multi-layer In-place Learning Network (MILN) [17]. So that the system is biologically

plausible, it requires *in-place* learning, meaning each neuron must learn (develop) on its own, while interacting with other neurons, without requirement of a dedicated learner outside the neuron. Structurally, MILN is a set of cortical areas, each containing a set of neurons, arranged in a layered hierarchy. The hierarchy is from sensors to motors — end to end. Each end interacts with the external environment. The number of layers of neurons in between depends on the implementation. It operates at discrete times $t = 0, 1, \dots$. At each t , the sensors gather information from the environment, passing to the network along bottom-up connections. Similarly, an output vector in the motor output space may or may not be present for any given t , for example during supervised learning.

To one neuronal area, there are three sources of stimulation – bottom-up connections, top-down connections, and the lateral connections. If we denote the state of the neuronal area at time step $t - 1$ as $L(t - 1)$, it updates its state from one time step to the next by updating synaptic weights and generating a response through lobe components analysis

$$(\mathbf{y}(t), L(t)) = \text{Cortex}_{LCA}(\mathbf{x}(t-1), \mathbf{y}(t-1), \mathbf{e}(t-1)|L(t-1)) \quad (1)$$

where $\mathbf{x}(t-1)$ is the vector of stimulation from the bottom-up, $\mathbf{e}(t-1)$ is stimulation from the top-down (e for “efferent”), and $\mathbf{y}(t-1)$ is the lateral stimulation from the same level. Note that the variable \mathbf{y} denotes *both* lateral stimulation and cell response. The cell response (firing) stimulates the cells it is laterally connected to at the next time step.

Lobe component analysis (LCA) is the method of synaptic weight updating and how the response $\mathbf{y}(t)$ is generated. A comprehensive treatment of LCA can be found in [18]. We will present the LCA algorithm here, noting the differences between standard LCA and the LCA used in this paper.

A. LCA Algorithms

For a neuronal layer, LCA incrementally updates c neurons represented by the column vectors $\mathbf{v}_1(t), \mathbf{v}_2(t), \dots, \mathbf{v}_c(t)$. In the laterally connected LCA here, we point out that there are three sources of stimulation contained within each vector: bottom-up, lateral and top-down:

$$\mathbf{v}_i = \mathbf{v}_{b,i} \Downarrow \mathbf{v}_{e,i} \Downarrow \mathbf{v}_{l,i} \quad (2)$$

where the \Downarrow symbol is for vertical vector concatenation (the \mathbf{v}_i are column vectors).

“Prenatal” initialization – Sequentially initialize c bottom-up synaptic weights using first c bottom-up inputs: $\mathbf{v}_{b,i}(0) = \mathbf{x}(t)$ and associated top-down inputs: $\mathbf{v}_{e,i}(0) = \mathbf{e}(t)$. When there are lateral connections, initialize them using an isotropic function (such as a Gaussian) and a mask function so that connections beyond a certain distance will always be zero¹. Set each neuron’s maturity $n_i = 1$, for $i = 1, 2, \dots, c$.

“Live.” For $t = c + 1, c + 2, \dots$, do

1. *Neurons compute.* Compute “pre”-response for all neurons: For all i with $1 \leq i \leq c$, compute the pre-competitive

response. The pre-response for a neuron is a function of its three sources of stimulation from the bottom-up $\mathbf{x}(t-1)$, top-down $\mathbf{e}(t-1)$ and lateral $\mathbf{y}(t-1)$ as follows:

$$\hat{y}_i(t) = g_i \left(\alpha \cdot \frac{\mathbf{x}(t-1) \cdot \mathbf{v}_{b,i}(t-1)}{\|\mathbf{x}(t-1)\| \|\mathbf{v}_{b,i}(t-1)\|} + \beta \cdot \frac{\mathbf{e}(t-1) \cdot \mathbf{v}_{e,i}(t-1)}{\|\mathbf{e}(t-1)\| \|\mathbf{v}_{e,i}(t-1)\|} + \gamma \cdot \frac{\mathbf{y}(t-1) \cdot \mathbf{v}_{l,i}(t-1)}{\|\mathbf{y}(t-1)\| \|\mathbf{v}_{l,i}(t-1)\|} \right) \quad (3)$$

and α, β , and γ must sum to one. They control contribution from bottom-up, top-down, and lateral stimulation. For no explicit lateral connections, let $\gamma = 0$. The function g is optional: it can perform thresholding or may be a sigmoidal.

2. *Lateral inhibition for different features and sparse coding.* For computational efficiency use the following top- k rule. Rank $k + 1$ top winners so that after ranking, $\hat{y}_1 \geq \hat{y}_2 \dots \geq \hat{y}_c$, as ranked responses. Use a linear function to scale the response:

$$y_i = (\hat{y}_i - \hat{y}_{k+1}) / (\hat{y}_1 - \hat{y}_{k+1}), \quad (4)$$

for $i = 1, 2, \dots, k$. For superior computational efficiency, this non-iterative ranking mechanism replaces repeated iterations that take place among a large number of two-way connected neurons in the same layer.

The number of winners k can adapt to a *developmental schedule* (see Table I) – a set of time points at which it changes.

3. *Optimal Hebbian learning.* Update only the top k winner neurons for all j in the set of top k winning neurons, using its temporally scheduled plasticity for the three types of weights: bottom-up, (possibly) lateral, and top-down

$$\begin{aligned} \mathbf{v}_{b,j}(t) &= w_1 \mathbf{v}_{b,j}(t-1) + w_2 y_j \mathbf{x}(t-1), \\ \mathbf{v}_{l,j}(t) &= w_{l,1} \mathbf{v}_{l,j}(t-1) + w_{l,2} y_j (\mathbf{m}_j \cdot \mathbf{y}(t-1)), \\ \mathbf{v}_{e,j}(t) &= w_1 \mathbf{v}_{e,j}(t-1) + w_2 y_j \mathbf{e}(t-1), \end{aligned} \quad (5)$$

where the cell’s scheduled plasticity is determined automatically by its two update-age dependent weights, called retention rate and learning rate, respectively:

$$w_1 = \frac{n(j) - 1 - \mu(n_j)}{n_j}, w_2 = \frac{1 + \mu(n_j)}{n_j}, \quad (6)$$

with $w_1 + w_2 \equiv 1$.

Note that the lateral updating uses a different set of learning and retention rates. Weights should be frozen ($w_{l,1} = 1$) until a certain t (e.g., 500), and allowed to start learning afterwards. The \mathbf{m}_j vector is a range limiter on the lateral connections (can only develop nearby).

Update the neuron “age” n_j only for the winners: $n_j \leftarrow n_j + y_j$, $j = 1, 2, \dots, k$.

4. *Lateral Excitation.* This step is how lateral excitation was previously handled. We include it because we compare with this method. When 3×3 updating is used, update the other neurons in a 3×3 neighborhood around every top- k winner, simulating 3×3 lateral excitation. Each neighboring neuron is updated as a fraction $r(d) = 1 - d/2$ of full update, where d is the distance between the updating neuron and the winner.

¹In the experiments, we used a Gaussian with deviation of 3 neighbor neurons, normalized so the max value was equal to one

The learning rate is $r(d)w_2(n)$ with $w_1(n) = 1 - w_2(n)$ and the (real valued) age is advance by $r(d)$.

5. *Long-term memory.* All other neurons i that do not update, keep their age and weights unchanged: $\mathbf{v}_i(t) = \mathbf{v}_i(t - 1)$.

IV. THEORY AND CONCEPTS

A. A Cortical Layer as a Probability Density Estimator

In self-organizing maps, strictly isotropic lateral excitation (such as the 3×3 updating discussed above) can lead to poor representation, meaning neuron weight vectors can get stuck representing areas of low probability in stimuli space. Why is this the case? While Hebbian learning drives neurons to these high probability areas, isotropic updating can be thought of a perturbation that pulls neurons away from one area and towards another. When there is too much pulling (e.g., all neurons pull one another), the overall representation may suffer. At a certain point in time, neurons must “stick” to the high probability areas, possibly ignoring their neighbors.

B. Optimality

Here, we discuss optimality of representation over time. How can we characterize the “best” organization of neuronal weights? The lobe component analysis updating method is concerned with two types of optimality: spatial and spatiotemporal. First, what is the best configuration? Spatial optimality requires that the spatial resource distribution in the cortical level is optimal in minimizing the representational error (distortion). If we have $V = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c)$ so that the expected square approximation error $\|\hat{\mathbf{x}}(V) - \mathbf{x}\|^2$ is statistically minimized:

$$V^* = (\mathbf{v}_1^*, \mathbf{v}_2^*, \dots, \mathbf{v}_c^*) = \arg \min_V E \|\hat{\mathbf{x}}(V) - \mathbf{x}\|^2. \quad (7)$$

and we have spatial optimality² in the above sense. $\hat{\mathbf{x}}(V)$ is the closest lobe component to an observation \mathbf{x} . It is a function of V – the lobe component matrix. Distortion is minimized by selecting the V that minimizes the expected difference between observation \mathbf{x} and closest lobe component $\hat{\mathbf{x}}(V)$. But this global configuration problem is too resource-intensive for the cortex to directly optimize. It can only make incremental improvements.

Each bottom-up weight vector (lobe component) develops to be the first principle component of the samples it updated for [18], [19]. The first principle component is the most *efficient* estimator of many types of distributions (e.g., Gaussian and other exponential distributions) – it minimizes the distortion error, at any t . It is spatially and spatiotemporally optimal.

The solution to Eq. 7 allows neuronal response properties that maximize mutual information between the input and output, which is equivalent to maximizing likelihood [4] under certain conditions³.

²LCA also is concerned with spatiotemporal optimality: how can the best configuration be achieved at *every* time step in development? For analysis of that, see [18].

³E.g., if the firing rate functions of the neurons is the cumulative density function to the stimulus source within its representative area’s p.d.f. (e.g., a sigmoid for a low variance Gaussian)

C. Self-Organization

1) *Non-topographic organization:* How might lateral connections help us reach the optimal representation? Using LCA without any lateral connections will not spatially organize the neuronal weights. However, it will approach (no guarantee to reach) the solution to Eq. 7. In that method, neuron i develops a $\mathbf{v}_{b,i}$ that is the expectation of its *response-weighted input*, $\mathbf{v}_{b,i} = E(y_i \mathbf{x}_i | y_i > 0)$. This is the principal component of these observations, having optimality properties discussed above. Each weight vector is optimal, but the configuration of all weights may not be. Especially, the configuration of neurons may be uneven in terms of how much of the probability space each represents.

2) *Topographic organization through neighbor updating:* Lateral connections can organize neurons more evenly and fairly throughout the input space. Using the idea of neighborhood updating from SOM leads to a topographic LCA, where a winning neuron’s neighbors are updated. This achieves some amount of topographic organization and cortical smoothness, depending on how the learning rate and scope of the neighborhood function are tuned. This can address the problem of poor resource allocation (bad configuration) since non-topographic LCA has through the neuronal density property of updating, which can be beneficial in better estimating probability.

Property 4.1: Neuronal Density Property: A region in the input space that has a higher probability density recruits a denser population of neurons. This property is true for any finite dimension of the input space \mathcal{X} .

Non-topographic LCA does not have this property but topographic LCA does (as does SOM). The methods that use isotropic neighbor updating may introduce a new problem. In that method, the updating equation for neighbor neurons is changed to

$$\mathbf{v}_j(t) = w_1 \mathbf{v}_j(t - 1) + w_2 h(n_{i,j}, t) \mathbf{x}(t - 1), \quad (8)$$

where neuron i is a winner, and $n_{i,j}$ is the distance from neuron i ’s 2D position to the position of neuron j . The kernel function h defines the updating strength.

A problem with Eq. 8 is that a non-winner neuron’s response y_j does *not* depend on its bottom-up and top-down combined weight vector \mathbf{v}_j . Instead, it is simply a function of distance from the winners. Intuitively, this means neurons can fire and update for stimuli that they do not represent well themselves.

With this method, each neuron will no longer converge to the expectation of its response-weighted input but to the expectation of observations that are either response weighted input or neighborhood-function weighted input. Thus, the distribution of observations which each neuron updates for will be greatly diffused (spread out) in comparison to the non-topographic case, leading to a poor approximation of the optimal representation (solution to Eq. 7).

This purely neighbor-based updating leads to problems for efficiently representing real data. Real world, high-dimensional data (e.g., raw pixels from a digital camera) is typically sparsely distributed – there will be large areas in the input

space where a stimulus is extremely unlikely⁴. And, at least with vision, the input space tends to have multiple *disconnected* areas where stimuli are probable. Using the SOM-style method of updating with any type of tuned learning rate leads to neurons representing areas of low probability between multiple high probability areas, since they are “pulled” by their neighbors closer to each of the separate high-probability areas. This phenomenon is well-documented in [5]. An example weight vector between two high probability areas is shown in Fig. 1.



Fig. 1. A layer-one weight vector, around other neighbor weight vectors, viewed as images, of a neuron exhibiting “harmful” interpolation through 3x3 updating.

This phenomenon is problematic for several reasons. First, the approximation of the probability density by response is poorer since less resource is used to represent regions where the data actually lies. Second, the neurons in low probability areas do not send meaningful messages to the next layer when they fire. Since they are “between” several different high probability areas, each presumably with different meaning, their firing does not send the next level an unambiguous message. Their firing is interference between different tasks, classes, etc. This interference can lead to performance errors (depending on the data). It is also interesting to note that in biological cortex – at least in V1 – these types of between-feature averaged representations have not been observed [11].

3) *Topographic organization through adaptive lateral connections*: A better solution is to use explicit lateral connections for excitation, instead of the SOM-inspired neighbor updating. In the proposed LCA with lateral excitatory connections, these connections affect the pre-competitive excitatory “potential” response (which are ranked and used to choose the winners). They take their effect *before* the winners are chosen instead of after, in the SOM-inspired method. The pre-response for a neuron is a function of its three sources from the bottom-up $x(t-1)$, top-down $e(t-1)$ and lateral $y(t-1)$ as in Eq. 3.

We update the neurons with the k largest \hat{y} , considered winners, using Eq. 5. Due to this competitive process, the developing weight vectors will only update for stimuli that they themselves represent well (unless e.g., α is too small). It will be less likely to have damaging interpolation as is seen in the 3×3 updating case when using this method. The interpolation provided by lateral connectivity should now be useful (within high-probability areas).

⁴Consider averaging two images of two different objects. This type of “ghosting” effect (see Fig. 1) is not typically seen in reality.

D. Lateral Weights

1) *Initialization of the lateral weights*: How are the lateral excitatory weights to be initialized? Similar to in LISSOM, and based on observations that most lateral excitatory connections are short-range [2], the scope of connectivity is restricted. For example, a neuron can only excite another neuron up to 5 neural positions away. And the actual values of the weights within the scope of connectivity are determined by an isotropic function such as a Gaussian. These initial weights will organize cortical representation to help pull similar features together in the physical neuron map. However, we also want to adapt the weights so that features with low correlation can exist nearby without interfering with one another – the lateral weight between them will diminish and be cut.

2) *Adaptation of the lateral weights*: Biological lateral connections are strong between functionally similar neurons [12], [15]. While the purpose of lateral excitatory weights early in development is to drive topographic organization, the purpose of later *adaptation* in lateral excitatory weights is to develop weights between nearby neurons that reflect the correlation of firing of those neurons. We can use LCA’s updating equation exactly for lateral weights and we will develop a weight between neuron i (which is a winner and has nonzero firing rate) and j (which is within the scope defined by m but may or may not be a winner) equal to

$$E(y_i(t)y_j(t-1)|y_i(t) > 0). \quad (9)$$

which is the expectation of firing rate of neuron j when neuron i has fired, weighted by neuron i ’s own firing rates. This is simply Hebb’s principle applied to the lateral weight.

3) *Developmental scheduling*: As is necessary for in-place learning, each neuron has a maturity (age), and an age-dependent updating schedule that defines w_1 and w_2 . However, adaptation of the lateral weights must be scheduled differently from the bottom-up and top-down weights. This is due to a simultaneous dependency in development – the bottom-up and top-down weights depend on the lateral connections early on for their development and topographic organization, and the lateral connections cannot reliably begin to reflect Eq. 9 until this organization has settled – meaning the adaptation of bottom-up and top-down weights has settled down. Therefore, the lateral connections must have more plasticity, later, than the bottom-up and top-down connections.

How can a single neuron have two separate updating schedules when it does not know the origin of its synapses? We can consider the lateral connections to be representative of a different cell type which we do not directly model – interneurons dealing with one-to-one connectivity in the same layer. This cell type can then have a different schedule of plasticity.

The performance effect of Eq. 5 is to cut off connections between areas of the stimulus space that do not correlate, thereby avoiding the problems that come from neuronal neighbor pulling. This lateral (and top-down) updating should lead to more *modular* collections of neurons – functionally related neuron groups with lower between-group interference than compared to the 3×3 method. And, due to the lower

interference, each feature should tend to represent the higher-probability areas (where the data is actually observed).

V. EXPERIMENTS

First, note that we will not utilize temporal context (top-down connections in the testing phase over realistic video sequences) in the experiments presented in this paper, as previous work has shown that it aids performance to a nearly perfect level for the data used here. We wish to investigate the effects of lateral connectivity without running into a performance ceiling.

A. MSU Dataset (25 Rotating Objects)



Fig. 2. Sample from each of the 25 objects classes. In the experiments, the training images were 56×56 grayscale.

For experiments, the MSU-25 objects dataset [9] was used. It contains centered images of 25 small objects (see Fig. 2) that rotate. There are 200 images of 56×56 pixels for each object. These 200 images covered about two complete rotations of 360 degrees for each object. Despite the color figure, grayscale was used due to avoid color’s usefulness in class discrimination. Including an additional “empty” (no object but the background sheet changes noticeably as the platform rotates) class, there were $200 \times 26 = 5200$ images total. For our tests, every fifth image in each sequence was set aside for testing. The other 80% were used to train the networks. Training involved random sample selection over 25,000 training samples, and the networks were tested with the disjoint data every 1,000 training samples. All tests were done five times with different sequences of inputs, and results averaged.

We compared the laterally connected LCA algorithm with an LCA method without lateral connections ($\beta = 0$), that instead used 3×3 updating. All tests utilized a neuronal plane size of 20×20 neurons. They connect to a motor layer of 26 neurons, which had top-down projections back to the feature layer. In testing, the highest responding motor neuron was taken as the guessed class.

We used the schedule of inhibition strength, or connection scope (changing k), shown in Table I and did not allow the lateral connections (if they were enabled) to adapt until $t = 500$. Each sample input was repeated for five iterations, and we did not update synapses until after the last iteration per

sample. This allowed the lateral activity to settle. In training, the correct motor output was imposed. The response was reset for each new training or testing sample, since such temporally discontinuous experience (jumping directly to a new image with a new object class having no transition) is not experienced in reality.

TABLE I
SCHEDULING INHIBITORY CONNECTION SCOPE.

Sample number	Number of winners (k)
0	20
1000	15
2000	5
3000	3
4000	1

Results are shown in figures 3 and 4. The version with adaptive lateral connections and adaptive top-down connections ($\alpha = \beta = \gamma = 0.33$) is the best. Previously we have shown the power of using top-down connections during development with this data [9] and 3×3 updating. It is interesting that the laterally connected version *without* using top-down ($\alpha = \gamma = 0.5$) matches the 3×3 version *with* top-down. Why is this the case? Fig. 4 shows the mean per-neuron class-entropy (see Appendix). If this is high, the neurons are firing for more than one class. We claimed this type of interference leads to errors. Indeed the two versions using 3×3 updating have significantly greater entropy on average than the ones using the lateral connections. The effects of the two methods can be visualized by looking at the bottom-up weights as images, as in Fig. 5.

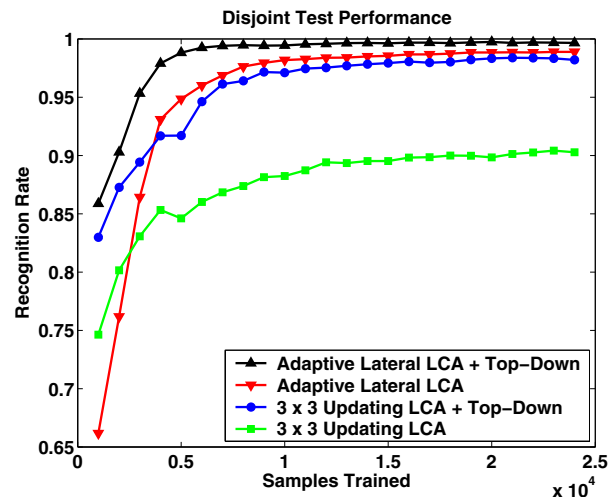


Fig. 3. Performance comparison of LCA using 3×3 updating to LCA with excitatory lateral connections with and without top-down connections.

We must note that the performance is only one of the criteria discussed in the paper, with the topographic organization being the second. Most classifiers are not biologically motivated and do not deliver such an organization.

1) *Comparison with other in-place updating methods:* We also compared with an LCA algorithm with lateral connections

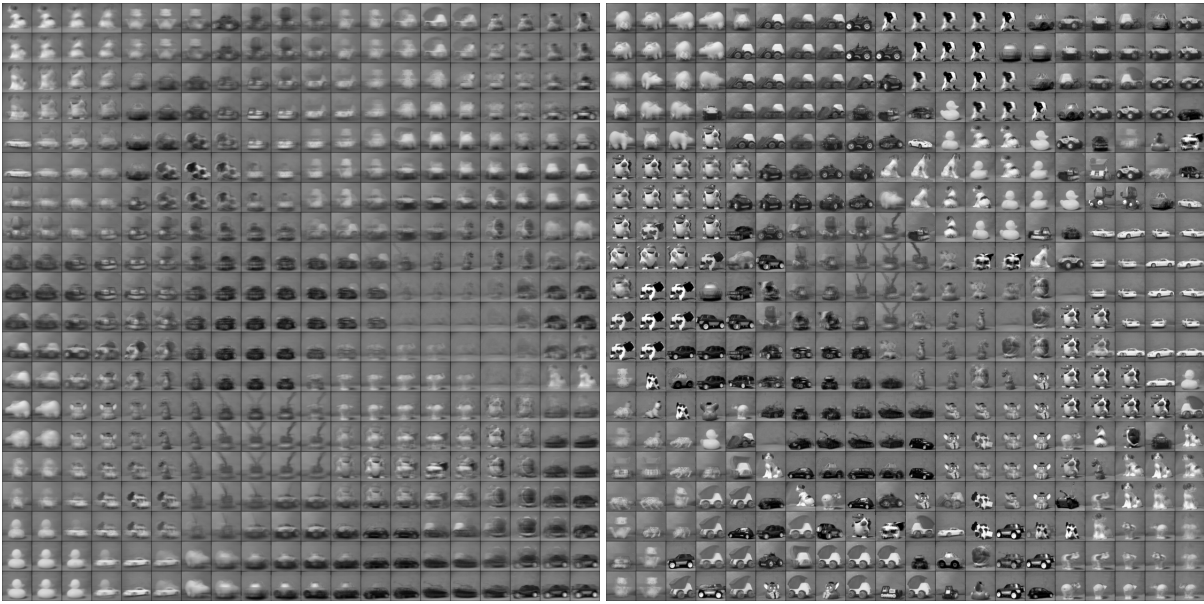


Fig. 5. On the left is the bottom-up weights after development for a neural map (each weight can be viewed as an image) that utilized 3×3 updating without top-down. On the right are the weights for a neural map that developed using explicit adaptive lateral connections (also without top-down). Note the smearing of the features on the left and the relatively higher precision of representation on the right, while still being somewhat topographically organized.

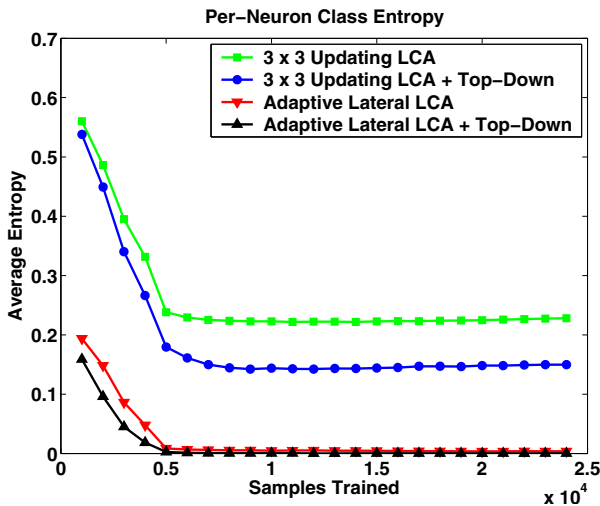


Fig. 4. Per-neuron class entropy for the four variants of LCA in the tests with the 25-Objects data.

but instead using the “dot-product” SOM updating equation [8] instead of the LCA update (step 3 in the algorithm)

$$\mathbf{v}_i(t) = \frac{\mathbf{v}_i(t-1) + \eta(t)\mathbf{x}(t-1)}{\|\mathbf{v}_i(t-1) + \eta(t)\mathbf{x}(t-1)\|_2} \quad (10)$$

where \mathbf{v}_i is the winning component vector at time t .

We also compared by instead using the LISSOM updating equation [13]⁵

$$\mathbf{v}_i(t) = \frac{\mathbf{v}_i(t-1) + \eta(t)\mathbf{x}(t-1)y_i(t)}{\|\mathbf{v}_i(t-1) + \eta(t)\mathbf{x}(t-1)y_i(t)\|_1} \quad (11)$$

⁵Note that we are *not* comparing with SOM and LISSOM – we simply replaced LCA’s updating equation.

The tuning of $\eta(t)$ for both of these is in general not simple⁶. Since the two above updating methods use only a single learning rate parameter to adapt the neuron weights to each new updating input, and a method to bound the strengths of synaptic efficacies (e.g., vector normalization), while CCI LCA uses the time-varying retention rate $w_1(t)$ and learning rate $w_2(t)$, where $w_1(t) + w_2(t) = 1$, in order to *optimally* maintain the energy estimate (as formalized in [18], and in order to achieve optimal representation.

The result in Fig. 6 supports this, as the non-LCA updating methods led to much worse performance. Interestingly, the entropy and organization of the feature layers was not significantly different between the three methods. The problem arose for the motor neuron’s bottom-up weights, which were not optimal and unstable for the SOM and LISSOM updating methods.

VI. CONCLUSION

Efficient (not wasting the available resource) and effective (leading to good performance) emergent internal representation is crucial for development. In published computational cortical maps, self-organization – topographic “smoothness” – is often achieved at the cost of high precision. The work reported here showed adaptive lateral excitatory connections used developmental scheduling to both self-organize a cortical map and to develop feature subgroups without cross-group interference that traditional (e.g., 3×3 updating) methods exhibit. The drastic performance improvements are shown as composite effects of several cortex inspired new mechanisms

⁶We used the “power” equation with initial learning rate of 0.1 for the SOM method [16], and based our tuning of the LISSOM equation from the appendix in [13]

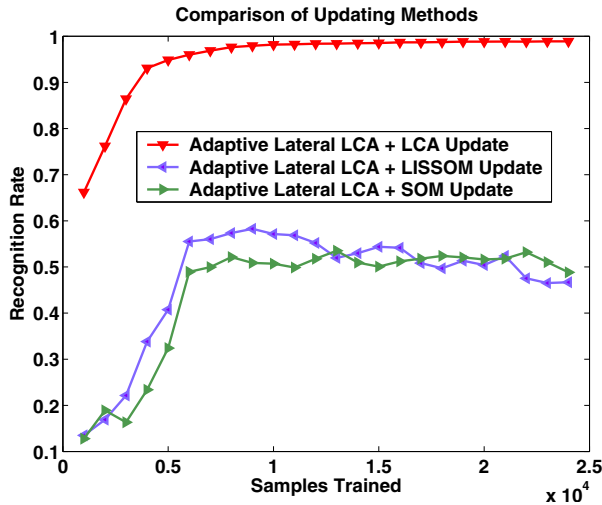


Fig. 6. Performance comparison for different updating methods, using lateral connections without top-down.

including top-down connections (attention), dually optimal LCA updating, and local sorting that simulated lateral inhibition without requiring many iterations. Under these mechanisms, we showed that global-to-local scope scheduling and adaptive lateral connections leads to effective and efficient self-organization. For future work, we will apply to a multilayered, locally connected (neurons receptive fields are less than the entire input), instead of globally connected, network to make it scalable for more complex real-world problems, such as developmental visual attention.

APPENDIX PER-NEURON CLASS ENTROPY

Each neurons “purity” of representation can be measured by using entropy. Neuron i can be considered class-pure if its entropy of firing for disjoint data is small. A neuron with zero entropy fires only for samples from the same class. A neuron with max (one) entropy fires with equal probability for all classes. To measure the purity of each neuron, we define the empirical “probability” the neuron fires for class j as:

$$p_{i,j} = \frac{\sum_d y_i(d)}{\sum_{t=1}^T y_i(t)} \quad \text{sample at time } d \in \text{class } j \quad (12)$$

over T firing events. To quantify the entropy of the probability distribution for the i -th neuron, we have

$$\varepsilon_i = - \sum_{d=1}^c p_{i,d} \log_c(p_{i,d}) \quad (13)$$

The upper bound of one (uniformly distributed) is due to the base c logarithm.

REFERENCES

- [1] J. Bullier. Hierarchies of cortical areas. In J.H. Kaas and C.E. Collins, editors, *The Primate Visual System*, pages 181–204. CRC Press, New York, 2004.
- [2] P. Buzas, K. Kovacs, A.S. Ferecsko, J.M.L. Budd, U.T. Eysel, and Z.F. Kisvarday. Model-based analysis of excitatory lateral connections in the visual cortex. *Journal of Comparative Neurology*, 499:861–881, 2006.

- [3] E. M. Callaway. Local circuits in primary visual cortex of the macaque monkey. *Annu. Rev Neurosci*, 21:47–74, 1998.
- [4] J.-F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4:112–114, 1997.
- [5] Y-M Cheung and L. Law. Rival-model penalized self-organizing map. *IEEE Transactions on Neural Networks*, 18:289–295, 2007.
- [6] E. G. Jones. Microcolumns in the cerebral cortex. *Proc Natl Acad Sci USA*, 97:5019–5021, 2000.
- [7] C. Koch. *The Quest for Consciousness: A Neurobiological Approach*. Roberts and Company Publishers, Englewood, Colorado, 2004.
- [8] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3rd edition, 2001.
- [9] M. D. Luciw and J. Weng. Topographic class grouping with applications to 3D object recognition. In *Proc. International Joint Conference on Neural Networks*, Hong Kong, June 1-6 2008.
- [10] M. D. Luciw, J. Weng, and S. Zeng. Motor initiated expectation through top-down connections as abstract context in a physical world. In *Proc 7th Int'l Conf on Development and Learning (ICDL)*, Monterey, CA, August 9-12 2008.
- [11] P.E. Maldonado, I. Godecke, C.M. Gray, and T. Bonhoeffer. Orientation selectivity in pinwheel centers in cat striate cortex. *Science*, 276:1551–1555, 1997.
- [12] B.A. McGuire, C.D. Gilbert, P.K. Rivlin, and T.N. Wiesel. Targets of horizontal connections in macaque primary visual cortex. *J Comp Neurol*, 305:370–392, 1991.
- [13] R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer, Berlin, 2005.
- [14] R.B.H. Tootell, K.J. Devaney, J.C. Young, R. Rajimehr G. Postelnicu, and L.G. Ungerleider. fmri mapping of a mophed continuum of 3d shapes within inferior temporal cortex. *Proc Natl Acad Sci USA*, 105:3605–3609, 2008.
- [15] D.Y. Ts’o, C.D. Gilbert, and T.N. Wiesel. Relationships between horizontal interactions and functional architecture in cat striate cortex as revealed by cross-correlation analysis. *J Neurosci*, 6:1160–1170, 1986.
- [16] Alhoniemi E Vesanto J, Himberg J and Parhankangas J. Som toolbox for matlab 5. Technical Report A57, Helsinki University of Technology: Finland, 2000.
- [17] J. Weng, H. Lu, T. Luwang, and X. Xue. A multilayer in-place learning network for development of general invariances. *International Journal of Humanoid Robotics*, 4(2), 2007.
- [18] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. on Autonomous Mental Development*, 1(1), 2009. accepted and to appear.
- [19] Y. Zhang and J. Weng. Convergence analysis of complementary candid incremental principal component analysis. *Michigan State University*, 2001.