2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

The Generative Adaptive Subspace Self-Organizing Map

Thusitha N. Chandrapala[1] and Bertram E. Shi[2]
Department of Electronic and Computer Engineering[1,2] and Division of Biomedical Engineering[2]
Hong Kong University of Science and Technology, Kowloon, Hong Kong SAR
tnc@ust.hk[1], eebert@ust.hk[2]

***Abstract*— The Adaptive Subspace Self Organized Map (ASSOM) is a model that incorporates sparsity, nonlinear pooling, topological organization and temporal continuity to learn invariant feature detectors, each corresponding to one node of the network. Temporal continuity is implemented by grouping inputs into "training episodes". Each episode contains samples from one invariance class and is mapped to a particular node during training. However, this explicit grouping makes application of this algorithm for natural image sequences difficult, since the grouping is generally not known *a priori*. This work proposes a probabilistic generative model of the ASSOM that addresses this problem. Each node of the ASSOM generates input vectors from one invariance class. Training sequences are generated by nodes that are chosen according to a Markov process. We demonstrate that this model can learn invariant feature detectors similar to those found in the primary visual cortex from an unlabeled sequence of input images generated by a realistic model of eye movements. Performance is comparable to the original ASSOM algorithm, but without the need for explicit grouping into training episodes**.

Keywords— generative model; hidden Markov model; invariance; self-organization

I. INTRODUCTION

The initial stages of the visual cortex perform a feature extraction task. Visual cortical cells are often loosely categorized into two classes: simple and complex [1]. In the primary visual cortex, both simple and complex cells exhibit orientation and spatial frequency selectivity. However, the responses of simple cells are highly dependent upon the phase (position) of the stimulus, whereas complex cell responses exhibit phase invariance. These complex cells are often assumed to be inputs to higher cortical stages, and the development of invariance along certain feature dimensions is thought to be an important process in the learning of more complex visual features.

The concept of invariant feature detectors is closely related to the concept of subspaces or manifolds of the input space [2, 3]. Natural inputs, such as auditory or visual signals, are usually high dimensional, but distributed along many lower dimensional manifolds. For example, although objects in the visual field may be quite complex, neurons in the initial stages of visual processing respond to relatively small spatial regions. At this scale, the typical inputs may be relatively simple and regular (e.g. oriented edges), but vary according to a class of transformations, such as translation. The inputs corresponding to one common pattern subject to different transformations within this class are referred to as an invariance class, and lie along a manifold of the high dimensional input space. If the transformation is a linear operation, then the manifold is a linear subspace. Invariant features correspond to subspaces, with the strength of each feature depending upon the length of the projection of the input onto the subspace.

Kohonen modeled the development of invariant feature detectors via the Adaptive Subspace Self Organizing Map (ASSOM) [2, 4]. The map consists of a set of nodes arranged in a two dimensional topology. Each node corresponds to one invariant feature detector, which is defined by a set of orthogonal basis vectors spanning a linear subspace. The strength of each invariant feature is computed summing by the squared length of the projection onto each basis vector. Learning is competitive. Input vectors in the training set are assigned to particular nodes, whose basis vectors are adapted towards the input vector. Neighboring nodes, as defined by the two dimensional topology, are updated in the same direction, but by a smaller amount. This ensures that the collection of nodes represents the variety of invariance classes found in the data, with neighboring nodes representing similar invariance classes. Later work has captured a similar idea using the concept of independence [5]. Topographical organization can also be introduced by having each node have only one basis vector, but by defining invariant features by summing the square lengths of the projections onto the basis vectors at that node and its neighbors [6][7]. However, this may reduce selectivity along other feature dimensions. For example, for image features, basis vectors at neighboring nodes may vary in orientation, reducing the orientation selectivity of the resulting feature detectors.

The concept of a training episode is quite significant in the ASSOM. An episode consists of a set of samples that are transformed versions of each other, i.e. they belong to the same invariance class. Depending upon the desired invariance, the transformations applied to the samples may differ, e.g. translations, rotations or scaling. The ASSOM learns invariance by assigning the same node to represent all samples in the episode. The training episode captures the idea that temporally neighboring inputs are similar, and that the resulting representation should vary slowly, and thus is related to ideas of slow feature analysis [8].

This work was supported by the Hong Kong Research Grants Council through the Hong Kong PhD Fellowship Scheme and grant 618713.

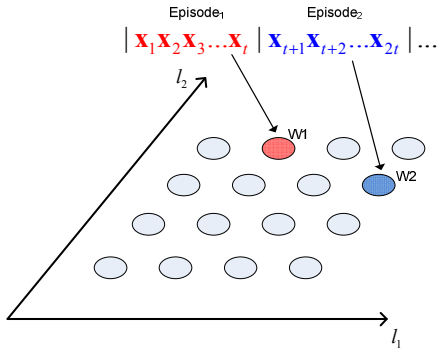


Fig. 1. For the ASSOM, the nodes are arranged in a two dimensional latent space. Each epsiode of training vectors is assigned to the node whose subspace has minimum projection error with the vectors in the episode.

However, this requirement that episode boundaries have to be specified explicitly makes application of the ASSOM to natural image sequences problematic, since information about episode boundaries may not be available. In addition, given the topological organization, the requirement that all input nodes in the episode be assigned to the same node may be too restrictive. Since neighboring nodes are similar, they may provide a better representation of a particular input.

This work addresses this limitation by proposing a generative model that captures the key ideas in the ASSOM, including sparsity, nonlinear pooling, topological organization and temporal continuity (slowness). Using the generative model, we can infer the node assignment of each input vector, while maintaining the idea that the representation should change slowly in time. This eliminates the need for explicit segmentation of the data into training episodes. The generative model also enables us to obtain an online training algorithm, where each incoming input vector is assigned to a node based only on the current and past vectors, and the basis vectors of the node and its neighbors are updated sample by sample. This generative model has some similarity with the Generative Topographic Mapping (GTM) [9], but the GTM does not include any concept of temporal continuity.

We demonstrate that the generative ASSOM can learn invariant features similar to those found in the primary visual cortex, based on unlabeled input sequences generated by a realistic model of eye movements that includes both saccades and drift. We also show that performance of the generative model is similar to that of the original ASSOM, where saccades are used to mark training episode boundaries.

I. Methods

A. The Adaptive Subspace Self Organizing Map

The ASSOM [9] is an extension of the Self Organized Map [10]. A set of nodes organized in lattice architecture seeks to find a set of low dimensional subspaces that reflect the statistics of a collection of N dimensional input vectors. The input to the network is a sequence of vectors $\mathbf{x}_t \in \mathbb{R}^N$ where t represents time.

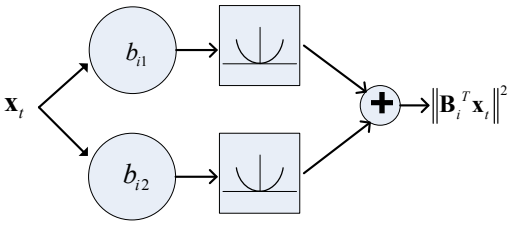


Fig. 2. The calculation of the squared length of the projection of input vector $\mathbf{x}_t$ onto each subspace.

An ASSOM consists of a set of S nodes indexed by $i \in \{1...S\}$. As illustrated in Figure 1, the nodes are organized in a 2-D latent space with locations $l_i = [l_{i1} \quad l_{i2}]^T$.

Associated with each node is an H-dimensional subspace of $\mathbb{R}^N$ defined by a set of H orthonormal basis vectors $\mathbf{B}_i = [b_{i1} ... b_{ih} ... b_{iH}]$. As illustrated in Figure 2, we define the response of each node as the squared length of the projection of an input vector $\mathbf{x}_t$ onto the subspace,

$$|| \mathbf{B}_i^T \mathbf{x}_t ||^2 = \sum_{h=1}^{H} (b_{ih}^T \mathbf{x}_t)^2 \quad (1)$$

This calculation is similar to that used in energy models of the visual cortical complex cell responses. Basis vectors are analogous to simple cell linear receptive fields.

The ASSOM learns features that are invariant to transformations of the input patterns. During training, the algorithm is presented with data in episodes consisting of training vectors that are similar but vary along the dimension for which the invariance is desired. For example, in order to learn translation invariant features, each episode consists of patterns generated by translating an image patch in different directions.

For each episode, the node whose subspace minimizes the total squared projection error over all input patterns in the episode is selected as the "winner". If we denote the index of the winning node by c, then

$$c = \arg\min_j \sum_{\tau \in \mathcal{E}} \left\| \tilde{\mathbf{x}}_\tau^j \right\|^2 \quad (2)$$

where $\mathcal{E}$ is the set of time indices within the episode and

$$\tilde{\mathbf{x}}_\tau^j = \mathbf{x}_\tau - \hat{\mathbf{x}}_\tau^j \quad (3)$$

is the difference between the input pattern at time τ, and its projection onto subspace j,

$$\hat{\mathbf{x}}_t^i = \mathbf{B}_i \mathbf{B}_i^T \mathbf{x}_t \quad (4)$$

The subspaces of the winning node and its neighbors are updated towards the input vectors in the episode according to

$$\Delta \mathbf{B}_i = \lambda h_{i,c} \sum_{\tau \in \mathcal{E}} \frac{\tilde{\mathbf{x}}_\tau^i (\mathbf{x}_\tau^T \mathbf{B}_i)}{\left\| \hat{\mathbf{x}}_\tau^i \right\| \left\| \mathbf{x}_\tau \right\|} \quad (5)$$

where λ is a positive learning rate and

$$h_{i,c} = \mathcal{N}(\mathbf{l}_i \mid \mathbf{l}_c, \sigma^2 \mathbf{I}) \tag{6}$$

where $\mathcal{N}(\cdot \mid \mathbf{m}, \mathbf{C})$ is a Gaussian density function with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$. The Gaussian neighborhood function $h_{i,c}$ ensures the winning node, *c*, and its neighbors are updated the most. The width of the Gaussian is controlled by σ. This update creates a topological map of smoothly varying subspaces. This learning rule is slightly modified from the original to improve computational efficiency [11].

Using all the samples in an episode to compute the update ensures that the subspaces capture invariance by accounting for transformed versions of the same input. However, it has the disadvantages that the episodes must be explicitly labeled, and that online updates are difficult, since all inputs in the episode must be seen before a decision about the winning node is made. In the following subsection, we describe an algorithm that overcomes these difficulties through a generative model of the inputs.

B. The Generative ASSOM Model

We formulate the generative ASSOM model as a Hidden Markov Model (HMM) [12, 13]. As in the standard ASSOM, we assume a set of S nodes arranged in a 2D latent space. In the generative model, these nodes are latent variables. At each time, *t*, the input $\mathbf{x}_t$ is generated by one of the nodes, which is identified by the indicator vector $\mathbf{z}_t \in \{0,1\}^S$ according to 1 of *S* coding,

$$\mathbf{z}_t = [z_{t1} \ldots z_{ti} \ldots z_{tS}] \tag{7}$$

where

$$z_{ti} = \begin{cases} 1 & \text{if } \mathbf{x}_t \text{ is generated by subspace } i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The sequence of nodes is generated according to a Markov random process. We assume that initially all nodes are equally probable for generating the input

$$\pi_i = P(z_{0i} = 1) = S^{-1}. \tag{9}$$

For subsequent times, the nodes change according to a transition probability

$$a_{ij} = P(z_{tj} = 1 \mid z_{t-1,i} = 1). \tag{10}$$

We will consider two possibilities assigning these transition probabilities: either by learning as described in Section II.E or by a fixed assignment,

$$a_{ij} = \rho S^{-1} + (1-\rho) \frac{\mathcal{N}(l_j \mid l_i, \sigma_{Tr}^2)}{\sum_k \mathcal{N}(l_k \mid l_i, \sigma_{Tr}^2)}. \tag{11}$$

The fixed assignment is a mixture of a uniform distribution and a discretized version of a Gaussian with standard deviation σ_{Tr}^2. The mixture weights are determined by $\rho \in [0,1]$ The Gaussian component provides for temporal continuity of the winning node (the node responsible for $\mathbf{x}_{t-1}$ or those close to it are more likely to be responsible for $\mathbf{x}_t$). The uniform component allows for large instantaneous changes in the winning node as at episode boundaries in the original ASSOM (all nodes are equally likely to be responsible independently of the past).

Given the latent state $\mathbf{z}_t$, we assume that the input vector $\mathbf{x}_t$ is the sum of two components: one lying in the subspace and one orthogonal to the subspace:

$$\mathbf{x}_t = \mathbf{B}_i \mathbf{w}_t + \mathbf{B}_i^{\perp} \mathbf{n}_t \tag{12}$$

where $\mathbf{w}_t \in \mathbb{R}^H$, $\mathbf{B}_i^{\perp} \in \mathbb{R}^{N \times (N-H)}$ is a matrix with orthogonal columns spanning the orthogonal complement of $\mathbf{B}_i$, and $\mathbf{n}_t \in \mathbb{R}^{N-H}$. We assume that $\mathbf{w}_t$ and $\mathbf{n}_t$ are independent and Gaussian distributed with diagonal covariance matrices. By orthogonality, we have $\mathbf{w}_t = \mathbf{B}_i^T \mathbf{x}_t$ and $\mathbf{n}_t = (\mathbf{B}_i^{\perp})^T \mathbf{x}_t$. Thus,

$$P(\mathbf{x}_t \mid z_{ti} = 1) = \mathcal{N}\left(\mathbf{w}_t \mid 0, \sigma_W^2 \mathbf{I}\right) \cdot \mathcal{N}\left(\mathbf{n}_t \mid 0, \sigma_N^2 \mathbf{I}\right) \tag{13}$$

We assume that $\sigma_W^2 \gg \sigma_N^2$, so that input vectors lying in the subspace have large probability, and input vectors off the subspace have low probability. This ensures the probability is largest when the input vector lies in the subspace of the generating node, and decays to zero when the input vector is orthogonal to the subspace. As discussed below, this assumption makes the generative model similar to the original model.

Using this model, we propose two algorithms for winner selection. The first algorithm, which we refer to as batch selection, selects the winner at time *t* based on all of the training data from time 0 to *T*. The second algorithm, which we refer to as online selection, selects the winner at time *t* based in the data from time 0 to time *t*.

C. Batch Winner Selection

This method is suitable is situations where offline training is possible. Given the entire input sequence $\mathcal{X} = [\mathbf{x}_1 \ldots \mathbf{x}_t \ldots \mathbf{x}_T]$, batch selection seeks to determine a sequence of latent states $\mathcal{Z} = [\mathbf{z}_1 \ldots \mathbf{z}_t \ldots \mathbf{z}_T]$ to account for the data, taking into account both the probability that each node generates the output, as well as the slow spatial variation in the winner implied by the transition probability matrix. Based on the HMM formulation presented here, there are two natural choices: the Viterbi algorithm or the forward-backwards algorithm [13]. In our experiments, we have observed little difference in the results of the two approaches. In the following, we present only the second approach, as it is more similar to the online selection algorithm presented later.

Batch winner selection computes

$$\gamma_t(i) = P(z_{ti} = 1 \mid \mathcal{X}) \tag{14}$$

It then selects winners for each time *t* by

$$c(t) = \text{argmax}_j [\gamma_t(j)] \tag{15}$$

We use the forward-backward algorithm to compute $\gamma_t(i)$ by

$$\gamma_t(i) = \tfrac{1}{K_t}\alpha_t(i)\beta_t(i) \tag{16}$$

where $K_t = \sum_{j=1}^{S}\alpha_t(j)\beta_t(j)$, and

$$\begin{aligned}\alpha_t(i) &= P(\mathbf{x}_1,\ldots,\mathbf{x}_t, z_{ti}=1)\\ \beta_t(i) &= P(\mathbf{x}_{t+1},\ldots,\mathbf{x}_T \mid z_{ti}=1)\end{aligned} \tag{17}$$

The quantities $\alpha_t(i)$ and $\beta_t(i)$ are computed via recursions that proceed forward and backwards in time. For example,

$$\alpha_t(i) = P(\mathbf{x}_t \mid \mathrm{z}_{ti}=1)\sum_{j=1}^{S}\alpha_{t-1}(j)a_{ji} \tag{18}$$

A similar backwards equation holds for $\beta_t(i)$.

Once the winners are selected, the subspaces are updated according to

$$\Delta\mathbf{B}_i = \lambda\sum_{t=1}^{T} h_{i,c(t)}\frac{\tilde{\mathbf{x}}_\tau^i(\mathbf{x}_\tau{}^T\mathbf{B}_i)}{\left\|\hat{\mathbf{x}}_\tau^i\right\|\left\|\mathbf{x}_\tau\right\|} \tag{19}$$

This equation is similar to (4), except the winning node $c(t)$ changes over time. After each update, the basis vectors are orthonormalized.

The generative model can be made equivalent to the original ASSOM with a single training episode by choosing $\rho = 0$, $\sigma_{Tr}^2 = 0$, and allowing $\sigma_W^2 \to \infty$. Under the first two assumptions, the transition probability matrix reduces to the identity matrix, which implies that the responsible node is the same for all time instants, similar to the assumption of the standard ASSOM with a single episode. In this case, (18) simplifies to $\alpha_t(i) = P(\mathbf{x}_t \mid z_{ti}=1)\alpha_{t-1}(i)$, which implies that

$$\alpha_t(i) = \prod_{\tau=1}^{t} P(\mathbf{x}_\tau \mid z_{\tau i}=1) \tag{20}$$

Combining with a similar analysis for $\beta_t(t)$, we obtain

$$\alpha_t(i)\beta_t(i) = \prod_{\tau=1}^{T} P(\mathbf{x}_\tau \mid z_{\tau i}=1) \tag{21}$$

for all t. Under the assumption $\sigma_W^2 \to \infty$,

$$P(\mathbf{x}_t \mid z_{ti}=1) \propto \exp\left(-\frac{\left\|\tilde{\mathbf{x}}_t^i\right\|^2}{2\sigma_N^2}\right) \tag{22}$$

In this case, the product in (21) is the exponential of the negative total squared projection error, and the winner selection criteria in (2) and (15) are identical.

D. Online Winner Selection

The online selection algorithm seeks to a winner for each input vector $\mathbf{x}_t$ using only the input sequence up to time t, $\mathcal{X}_t = [\mathbf{x}_1 \ldots \mathbf{x}_t]$. Because basis vectors can be updated at each time step, this algorithm is better suited to modeling learning in biological systems. Similarly to (14) and (15), we compute

$$\hat{\alpha}_t(i) = P(z_{ti}=1 \mid \mathcal{X}_t) \tag{23}$$

and then select winners for each time t by

$$c(t) = \mathrm{argmax}_j[\hat{\alpha}_t(j)] \tag{24}$$

The values of $\hat{\alpha}_t(i)$ can be computed recursively using the forward algorithm (18), since

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{S}\alpha_t(j)} \tag{25}$$

The online selection algorithm enables us to update basis vectors at every presentation of the input pattern,

$$\Delta\mathbf{B}_i = \lambda h_{i,c(t)}\frac{\tilde{\mathbf{x}}_\tau^i(\mathbf{x}_\tau{}^T\mathbf{B}_i)}{\left\|\hat{\mathbf{x}}_\tau^i\right\|\left\|\mathbf{x}_\tau\right\|} \tag{26}$$

Fig. 3 presents the pseudo-code for the online GASSOM algorithm.

```
Given input vectors x_t and parameters a_ij , σ_N² , σ_W²
∀i, initialize α_0(i) = S^-1
For each time t,
   ∀i, compute α_t(i) from { α_t-1(j) } using (18)
   ∀i, compute α̂_t(i) using (23)
   Select a winner c(t) using (24)
   ∀i, update basis vectors using ΔB_i in (26) and
      orthonormalize.
```

Fig. 3. Pseudocode for the online GASSOM algorithm. Here we assume that the model parameters are fixed, but they could be learned by re-estimating them using online versions of (27) - (30)

E. Learning the model parameters

For batch selection, the model parameters a_{ij}, σ_N and σ_W can be estimated using the Baum-Welch algorithm [13]. We calculate

$$\begin{aligned}\xi_t(i,j) &= P(z_{ti}=1, z_{t+1,j}=1 \mid \mathcal{X})\\ &= \frac{\alpha_t(i)a_{ij}P(\mathbf{x}_{t+1}\big|\, z_{t+1,j}=1)\beta_{t+1}(j)}{P(\mathcal{X})}\end{aligned} \tag{27}$$

and estimate new transition probabilities a_{ij} and variances σ_N and σ_W by

$$a_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{28}$$

$$\sigma_N^2 = \frac{1}{S(N-H)}\cdot\sum_{i\in\{1\ldots S\}}\left[\frac{\sum_{\tau=1}^{T}\gamma_\tau(i)\left\|\tilde{\mathbf{x}}_\tau^i\right\|}{\sum_{\tau=1}^{T}\gamma_\tau(i)}\right] \tag{29}$$

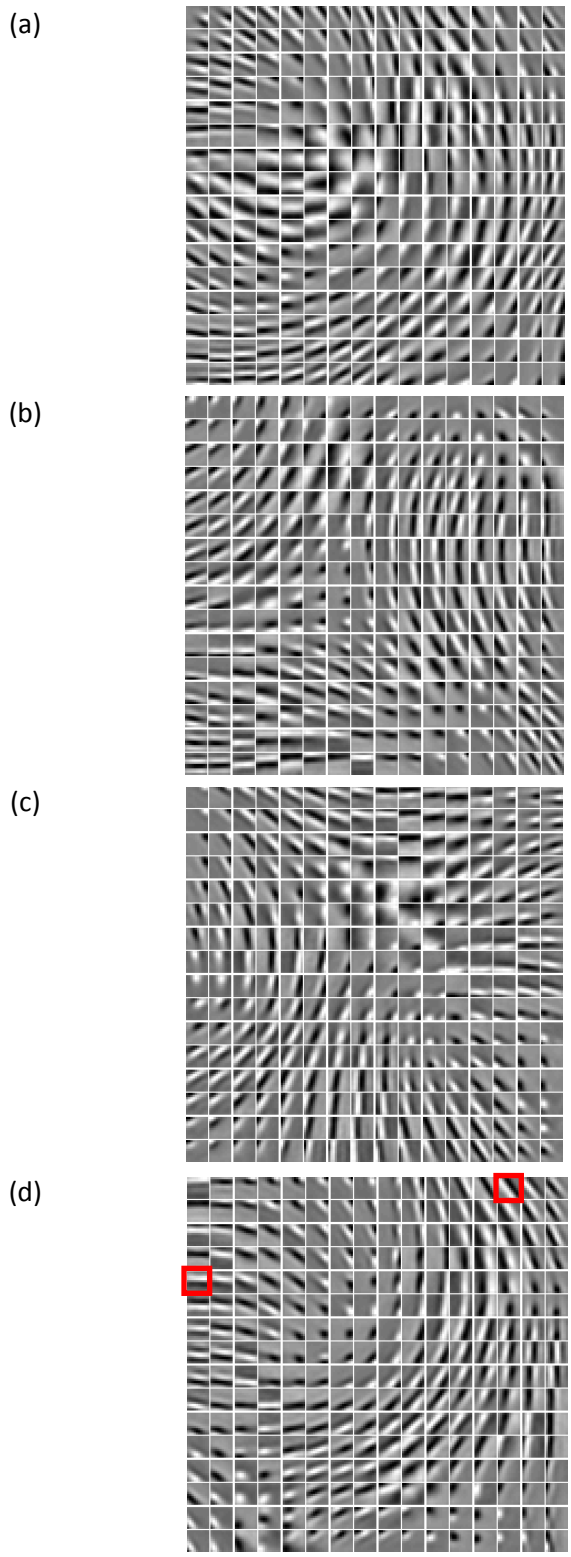


Fig. 4. One set of basis vectors obtained from: (a) The original ASSOM, (b) Batch selection with learned transition probabilities, (c) Batch selection with fixed transition probabilities, (d) Online selection with fixed transition probabilities. Basis vectors are arranged in the 16x16 ASSOM topology. The highlighted bases in (d) are analyzed in Fig. 6.

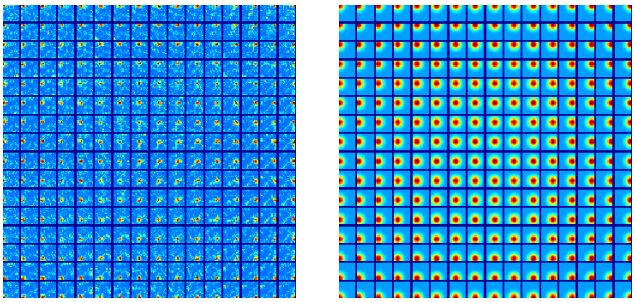


Fig. 5. (a) The transition probabilities learned by batch GASSOM using the Baum-Welsh algorithm. Each sub-image shows the transition probabilities a_{ij} for the node j at time t-1. The sub-images are arranged according to the topology of the nodes j. Each pixel shows the probability of transitioning to node i at time t, where the pixels are arranged according to the topology of the nodes. Sub-images are normalized so that blue corresponds to zero probability and red to the maximum probability. (b) The transition probabilities when the model parameters are fixed.

$$\sigma_W^2 = \frac{1}{SH} \cdot \sum_{i \in \{1 \ldots S\}} \left[\frac{\sum_{\tau=1}^{T} \gamma_\tau(i) \left\| \hat{\mathbf{x}}_\tau^i \right\|}{\sum_{\tau=1}^{T} \gamma_\tau(i)} \right] \tag{30}$$

For online selection, we can estimate parameters using the online estimation algorithm presented in [14][15].

II. Experiments and Results

We present the results from the original ASSOM, the batch ASSOM with learned model parameters, the batch ASSOM with fixed model parameters, and the online ASSOM with fixed model parameters.

All maps consist of 16×16 arrays of latent nodes. Input vectors are 10×10 pixel image patches (N = 100) generated according to the eye movement model described below. Each subspace is two-dimensional (H = 2).

A. Fixational eye movement model

In the human visual system, drift during fixation produces a set of visual samples with a linear displacement around the fixation point [16]. Saccades introduce large shifts in the fixation point. Thus, fixations are analogous to the training episodes, with successive fixations being delineated by saccades.

The input stimuli in our experiments are generated based on a model based on saccades and ocular drifts described in [17]. Saccade amplitudes are modeled using an exponential distribution with mean 2 degrees. The saccade direction is a uniform random variable over $[0, 2\pi]$. The intersaccadic interval is modeled by an exponential distribution with mean 300ms. Ocular drift is modeled as a diffusion process with a diffusion constant of 40 arcmin^2/sec.

The sequence of image patches presented to the model are generated from the natural images in Van Hateren database [18], which are first pre-whitened [19]. Time in the eye movement model is discretized at 25ms per frame. Visual space is quantized at 1 arcmin per pixel. At the beginning, an

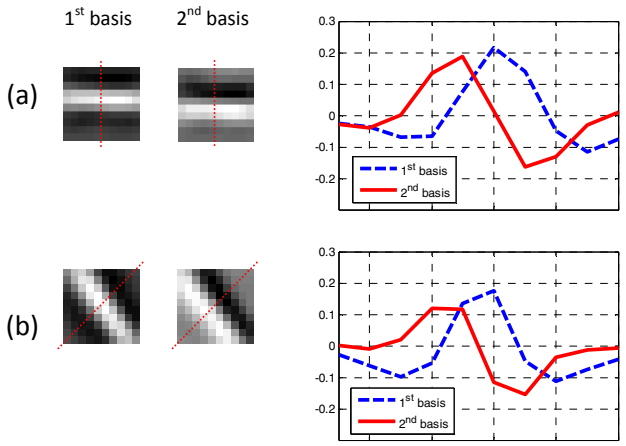


Fig. 6. Selected basis vectors from Fig. 4(d). The plots on the right show the cross sections along the dotted lines.

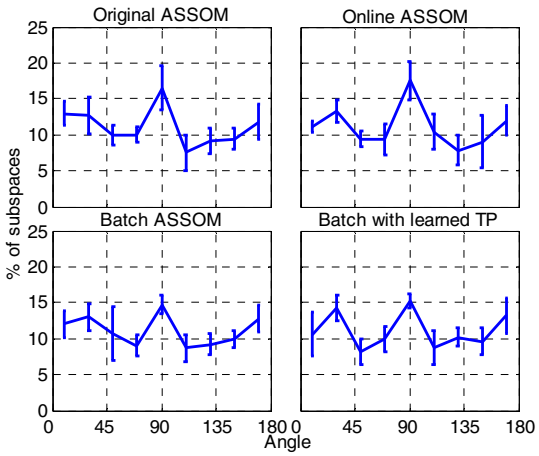


Fig. 7. The distribution of orientation angles of the learned basis vectors

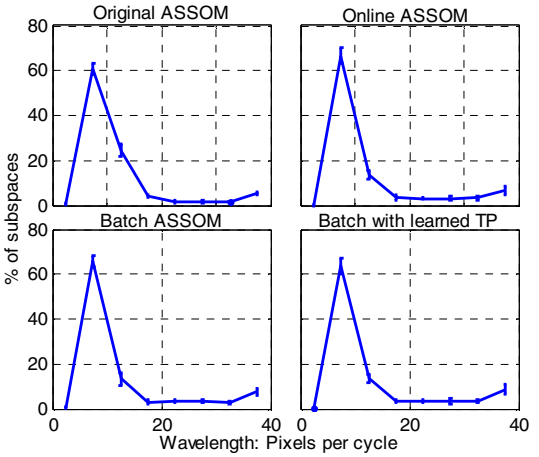


Fig. 8. The distributions of wavelengths of the learned basis vectors

image from the database and a fixation point in the image are chosen randomly. The gaze point is first chosen at the fixation point. A 10 x 10 patch input to the model is obtained by interpolating the image around the fixation point. Patches are normalized to have a unit norm. At each frame, a new gaze point is chosen according to the ocular drift diffusion process until the next saccade. After every 20 saccades, a new image is chosen from the database and a new fixation point chosen.

The total length of the training sequence is 1×10^7 frames. For the original ASSOM simulations, training episodes are defined by the period between two successive saccades. For the batch ASSOM, one batch consists of an unlabeled sequence of image patches collected over 20 saccades. The online ASSOM is presented with the stream of patches generated by the model. Neither the batch nor the online ASSOM have any information regarding the episode boundaries, apart from that contained implicitly in the image content of the patches.

B. *Parameter settings*

The learning rate starts at 1×10^{-2} and decays exponentially to 1×10^{-4} with a time constant 4×10^4. The standard deviation of the neighborhood function $h_{i,c}$ starts at a value of 4 and decays exponentially to 0.5 with a time constant of 4×10^4.

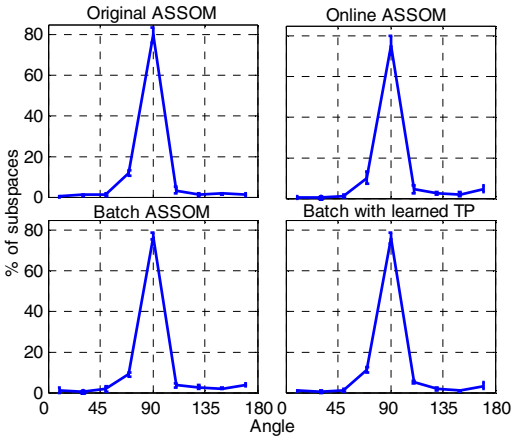
Basis vectors are initialized as independent and identically distributed (i.i.d.) vectors whose components are chosen from a uniform distribution. They are then orthonormalized. When learned, the elements of the transition probability matrix are chosen independently from a uniform distribution between 0 and 1, and then normalized so that each row sums to one. The standard deviations of the Gaussian emissions are initialized σ_N = 0.1 and σ_W =1. When fixed, we set the transition probabilities according to (11) with $\sigma_{Tr}=2$ and $\rho=0.3$. The emission probabilities are given by (13) with σ_N = 0.08 and σ_W =0.4. These parameters were set based on the results from the model parameter learning.

C. *Basis vectors obtained*

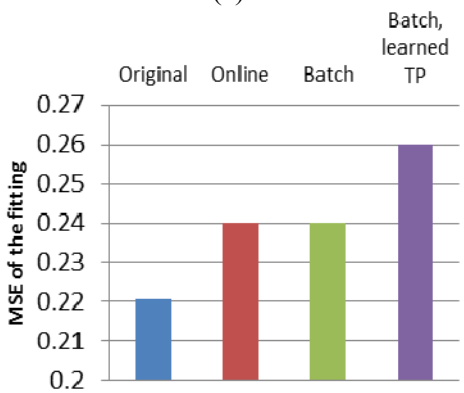
Fig. 4 shows one of the basis vector from each subspace of the maps obtained by the original ASSOM, the batch ASSOM with learned model parameters, and the batch and online ASSOM with fixed model parameters. We show only one basis vector since the other basis vector in the subspace is usually similar, as discussed in more detail below. Comparing the basis vectors which emerge from the different models, we observe that they are qualitatively quite similar in appearance and in their topological arrangement. We provide more quantitative measures in the following.

As a result of the topological updates of the basis vectors, the map of feature detectors exhibits many properties found in the primary visual cortex, where selectivity to orientation and spatial frequency vary smoothly across the cortical surface. We observe a similar smooth variation in the orientation and spatial frequency of the basis vectors in the maps generated by our algorithms. The topographical maps of basis vectors exhibit orientation pinwheels, similar to pinwheel-like structures that have been identified in the mammalian visual cortex [20, 21].

Fig. 5(a) shows transition probabilities learned by the batch ASSOM starting from random initial conditions. We observe



(a)



(b)

Fig. 9. (a) The distributions of the phase difference between the two basis vectors in each subspace. (b) The mean squared error (MSE) of the Gabor fits to the basis vectors. Since the basis vectors are normalized, the maximum fitting error is 1.

that the learned transition probabilities display a very similar trend as assumed in equation (11), with a Gaussian-like peak centered at the latent node location at time *t*-1. A least squares fit of (11) to the learned transition probabilities gives parameters $\sigma_{Tr} = 1.25$ and $\rho = 0.5$. The parameter assumed when the model parameters were fixed were $\sigma_{Tr} = 2$ and $\rho = 0.3$. The similarity between the basis vectors which emerge illustrates that the model is quite robust to the exact choice of the transition probabilities, as long as they ensure a balance between temporal slowness and minimizing projection error. The emission probabilities from learning were $\sigma_N = 0.08$ and $\sigma_W = 0.4$, the same as assumed when the model parameters were fixed.

Fig. 6 show the pairs of basis vectors from two of the subspaces learned by the online ASSOM. We observe that the two basis vectors have the very similar spatial frequencies and orientations. Their cross sections in the directions perpendicular to the level sets show a Gabor-like profile: a sinusoidal variation modulated by a spatially decaying envelope. The two basis vectors are in approximate phase quadrature, as the sinusoidal variations differ in phase by about 90 degrees.

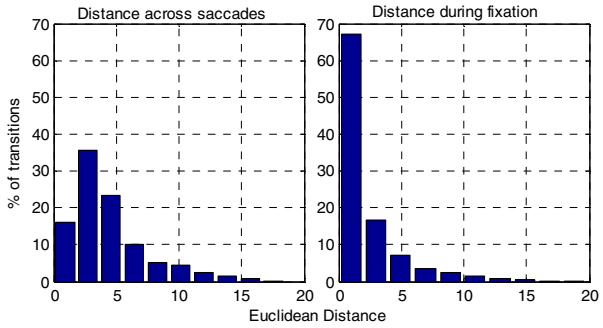


Fig. 10. Histograms of the distances between temporally adjacent winners during fixation and across saccades.

In order to analyze the structure of the basis vectors learned, we fit individual basis vectors with two-dimensional Gabor functions and extract the wavelength and orientation from the fits. Fig. 7 shows the distributions of orientations in the models. We find that all models provide a fairly uniform sampling of orientations. Fig. 8 shows the distributions of spatial wavelengths, which are also similar across all models.

To measure the phase relationship between the basis vectors, we fit the basis vectors in each subspace with a common Gabor fit: the two basis vectors share the same spatial frequency, orientation and bandwidth parameters, but may differ in phase. Fig. 9(a) plots the distributions of phase parameters. We observe a clustering around 90 degrees, indicating that the two basis vectors are usually in approximate phase quadrature. Fig. 9(b) shows that quality of the Gabor fits is quite consistent across all models.

To evaluate the ability of the online method to identify the episode boundaries, we recorded the locations of the winners during training. Fig. 10 shows the histograms of the Euclidian distances between temporally adjacent winning nodes both across saccades (analogous to boundaries between episodes) and within each fixation (analogous to the period within a training episode). Across saccades, winning nodes are more widely separated. During fixations when the gaze location drifts slowly, transitions tend to occur either to the same node or nearby nodes. This demonstrates that the model captures the

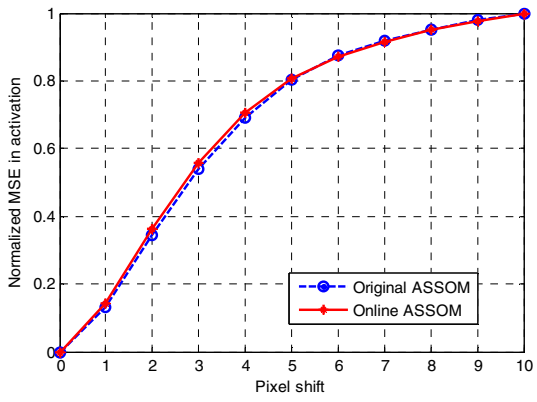


Fig. 11. Mean Square Error of the activation when the input is shifted**.** The online algorithm has the same invariance properties as the original.

temporal slowness which was required to be explicitly specified in the original ASSOM. The generative ASSOM models are provided with no indication about the location of saccades, aside from changes in the image patches themselves.

Invariance of the receptive fields was tested by measuring the difference in activation for a shifted input sequence. Activation is calculated as the squared projection length of the input on a subspace as defined in Fig. 2. Fig. 11 shows the average mean squared error between the activations for the shifted and unshifted patches. Averaging was done over activations to 5×10^3 patches. The MSE is normalized to one at a shift of 10 pixels. Both the original and online algorithms showed very similar invariance for translated inputs.

III. Discussion

The ASSOM learns invariance by looking at explicitly defined training episodes in the input, and by adjusting the subspaces to minimize error between the winning subspace and its neighbors and the input patterns in the episodes. Maximizing invariance leads to the emergence of basis vectors very similar to the phase quadrature receptive fields of simple cells in the primary visual cortex, which are hypothesized to provide input to the same complex cell.

In this paper, we have presented a generative model for the ASSOM, which can learn similar invariant feature detectors as the original ASSOM, but without the need for on an external agent to provide explicit information about episode boundaries. The algorithms use temporal continuity of the latent node accounting for the data to uncover slowly varying features from quickly varying inputs. Thus, this approach shares a similar motivation as Slow Feature Analysis [22, 23], which has been suggested as a mechanism used by the brain to account for neuronal development.

This scheme is particularly useful in sensory coding where inputs are generated from active sensors, such as the eyes, which are in constant motion. By uncovering slowly varying features, these algorithms may be able to provide a basis for more robust estimation about the changes in the state of the environment. We have demonstrated the applicability of our approach to this situation using input generated by a realistic model of eye movements.

The generative ASSOM model also enables us to develop batch and online estimation algorithms. The results presented here demonstrate that both algorithms can achieve similar performance to the original ASSOM. This generative extension of the ASSOM should increase the applicability of the ASSOM model to a wider variety of situations, especially those involving unlabeled temporal sequences as might be generated during natural behavior. In particular, the online algorithm is particularly applicable to modeling neuronal development during interaction with the environment.

References

[1] D. H. Hubel, *Eye, Brain, and Vision.* Scientific American Library/Scientific American Books, 1995.

[2] T. Kohonen, "Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map," *Biological Cybernetics,* vol. 75, pp. 281-291, 1996.

[3] A. Hyvèarinen, J. Hurri and P. O. Hoyer, *Natural Image Statistics.* Springer, 2009.

[4] T. Kohonen, S. Kaski, H. Lappalainen and J. Saljärvi, "The Adaptive Subspace Self Organizing Map (ASSOM)," in *International Workshop on Self Organizing Maps,* pp. 191-196, 1997.

[5] A. Hyvärinen and P. Hoyer, "Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation,* vol. 12, pp. 1705-1720, 2000.

[6] A. Hyvärinen and P. O. Hoyer, "A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images," *Vision Research,* vol. 41, pp. 2413-2423, 2001.

[7] K. Kavukcuoglu, M. A. Ranzato, R. Fergus and Y. Le-Cun, "Learning invariant features through topographic filter maps," in *IEEE Conference On Computer Vision and Pattern Recognition,* pp. 1605-1612, 2009.

[8] P. Földiák, "Learning invariance from transformation sequences," *Neural Computation,* vol. 3, pp. 194-200, 1991.

[9] C. M. Bishop, M. Svensén and C. K. Williams, "GTM: The generative topographic mapping," *Neural Computation,* vol. 10, pp. 215-234, 1998.

[10] T. Kohonen, S. Kaski and H. Lappalainen, "Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM," *Neural Computation,* vol. 9, pp. 1321-1344, 1997.

[11] T. Kohonen, "The self-organizing map," *Proceedings of IEEE,* vol. 78, pp. 1464-1480, 1990.

[12] H. Zheng, G. Lefebvre and C. Laurent, "Fast-learning adaptive-subspace self-organizing map: An application to saliency-based invariant image feature construction," *IEEE Transactions on Neural Networks,* vol. 19, pp. 746-757, 2008.

[13] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *ASSP Magazine,* vol. 3, pp. 4-16, 1986.

[14] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of IEEE,* vol. 77, pp. 257-286, 1989.

[15] O. Cappé and E. Moulines, "Online EM algorithm for latent data models," *Journal of the Royal Statistical Society,* 2008.

[16] O. Cappé, "Online EM algorithm for hidden Markov models," *Journal of Computational and Graphical Statistics,* vol. 20, pp. 728-749, 2011.

[17] M. Rolfs, "Microsaccades: Small steps on a long way," *Vision Research,* vol. 49, pp. 2415-2441, 2009.

[18] X. Kuang, M. Poletti, J. D. Victor and M. Rucci, "Temporal encoding of spatial information during active visual fixation," *Current Biology,* 2012.

[19] J. H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society of London: Biological Sciences,* vol. 265, pp. 359-366, 1998.

[20] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by VI?" *Vision Research,* vol. 37, pp. 3311-3326, 1997.

[21] T. Bonhoeffer and A. Grinvald, "The layout of iso-orientation domains in area 18 of cat visual cortex: optical imaging reveals a pinwheel-like organization," *The Journal of Neuroscience,* vol. 13, pp. 4157-4180, 1993.

[22] T. Bonhoeffer and A. Grinvald, "Iso-orientation domains in cat visual cortex are arranged in pinwheel-like patterns," *Nature,* vol. 353, pp. 429-431, 1991.

[23] P. Berkes and L. Wiskott, "Slow feature analysis yields a rich repertoire of complex cell properties," *Journal of Vision,* 2005.

[24] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation,* vol. 14, pp. 715-770., 2002.