AN ANTICIPATORY SELF-ORGANIZED MAP FOR ROBUST RECOGNITION

Samarth Swarup and Kiran Lakkaraju and Alexandre Klementiev
and Sylvian R. Ray
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, USA
email: swarup@uiuc.edu

Ernst Sassen
Department of Applied Software Engineering
University of Munich
München, Germany

ABSTRACT
When performing any real-time detection task, such as face detection, speech recognition, etc., we can take advantage of the temporal correlations within the data stream. This can help us make detection more robust by using anticipations about the target to overcome the variance due to noise. We present an extended self-organized map that uses lateral weights between the nodes to learn temporal relations between clusters. These weights are then used during recognition to bias certain nodes to win the competition. This converts the self-organized map from a maximum likelihood to a maximum *a posteriori* estimator. We present an experiment using artificial data to demonstrate the benefit of the anticipatory self-organized map.

KEY WORDS
Anticipation, Self-Organized Maps, Neural Networks, Temporal Prediction.

1 Introduction

Real-time object detection is a very desirable capability for robotic agents, surveillance systems, etc. However, it has proven to be very challenging to implement because small changes in lighting, object orientation, position etc. can cause large changes in the observed signals. (See [1] for a good overview.) Most approaches to robust detection have concentrated on extracting features that are robust to such changes ([2],[3],[4],[5]). However, the temporal aspect of the problem is often overlooked. We can take advantage of the fact that targets tend to persist in the same location over short periods of time to make existing recognition systems more robust to noise. The anticipatory self-organized map (SOM) we present here clusters feature vectors in the usual manner. These may be feature vectors obtained from another detection algorithm, such as [3]. It then does a second stage of learning, where it learns lateral weights between the SOM nodes. These lateral weights are used to anticipate the most likely winning node for the next time step. Consequently, the anticipatory SOM can cluster the input not only on the basis of the current input, but also on the basis of the recent history of inputs. We will show that this leads to more robust detection because the SOM now performs maximum *a posteriori* classification.

The rest of this paper is organized as follows. The next section reviews other extensions of the self-organized map into the temporal domain, and compares these with our anticipatory SOM. Sections 3 and 4 provide a description of our model, and a Bayesian perspective on the computation it is performing. Section 5 presents an experiment with artificial data, designed to show the improved noise robustness of the anticipatory SOM. Finally we present some conclusions and suggestions for future work.

2 Related Work

There have been several extensions to the basic self-organized map to take time into account, such as the temporal Kohonen map (TKM) [6], the recurrent self-organized map (RSOM) [7], [8], the contextual self-organized map (CSOM) [9], and others [10], [11]. However most extensions have focused on retrieval of the temporal sequence, whereas our focus is on using the temporal correlations to enable robust recognition.

The TKM and the RSOM both use leaky integrators, though they use slightly different formulations. A leaky integrator node decays its activation slowly once it fires. Therefore the winning node will tend to win again in the next time step.

Similarly, in [11], Földiák relies upon the stability of a pattern in space and time. This allows a grouping of cells to respond to a particular stimulus. In addition, groups of cells that respond to similar stimuli are dynamically connected to the same complex cell, which combines elementary features. Here also, the temporal aspect is that a complex cell, once activated, has a higher chance of being activated in the subsequent time steps.

However, this only works well if the input vector corresponding to the stable target falls into the same cluster every time. Small changes in target orientation, position and lighting, though, can cause significant changes in the input vector.

The CSOM actually augments the input vector with a copy of the activations from the previous time step, thus providing true recurrence. The goal is to predict the temporal sequence by encoding the sequence to the required depth in the receptive fields of the CSOM nodes.

Similarly, in [10], Barreto and Araújo looked at the problem of learning multiple temporal sequences with a self-organized map. They proposed learning lateral weights between neurons, using a Hebbian rule, which are used to encode the temporal sequence. They are also interested in explicitly recalling the temporal sequence, whereas we merely use the lateral weights to provide a prior probability for the current time step based on the activations of the nodes in the previous time step.

3 Model Description

The model consists of a two-dimensional self-organized map (SOM) [12], which combines the current input with the previous activations to determine current activations. Training proceeds in two stages. The first stage consists of a traditional SOM, where the activation $y_{ij}(t)$, of node (i,j) in the SOM, at time t, is calculated as follows:

$$y_{ij}(t) = \frac{\alpha_{ij}(t)}{1 + e^{-\frac{1}{\beta_{ij}(t)} \cdot u_{ij}(t)}} \tag{1}$$

This is basically a sigmoid function with some modifications. It is parameterized using β_{ij} and b_{ij} (see equation 2 so that we can fit it to the distribution of points around each cluster center. This enables us to produce better estimates of the likelihood of a point having been generated by this cluster.

The net input, $u_{ij}(t)$, is given by

$$u_{ij}(t) = \mathbf{w}_{ij}(t) \cdot \mathbf{x}(t) - b_{ij}(t) \tag{2}$$

where $\mathbf{x}(t)$ is the input vector at time t, $\mathbf{w}_{ij}(t)$ is the weight vector at node (i,j) at time t, and $b_{ij}(t)$ is the bias input for node (i,j) at time t. The dot product provides a measure of the distance between the weight vector and the input vector. However, it is larger when the two vectors are close to each other. Thus, the node with the maximum activation is called the winning node.

The value $\alpha_{ij}(t)$ in equation (1) is set to 1 at this stage. The weights are updated as follows.

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_{ij}(t) + \eta \cdot h_{ij} \cdot \mathbf{x}(t), \quad \forall \boldsymbol{i,j} \tag{3}$$

where η is the learning rate, and h_{ij} is a neighborhood function, given by

$$h_{ij} = exp(-\frac{d^2_{win}(i,j)}{2\sigma^2}) \tag{4}$$

where $d_{win}(i,j)$ is the Euclidean distance of node (i,j) from the winning node.

After the ascending weights to a node have been updated, they are normalized using the L2 norm

$$w_{ijk} = \frac{w_{ijk}}{\|\mathbf{w}_{ij}\|}, \quad \forall k \tag{5}$$

The bias, b_{ij} is updated as follows:

$$b_{ij}(t+1) = b_{ij}(t) + \eta \cdot h_{ij} \cdot u_{ij}(t) \tag{6}$$

β_{ij} is updated as follows:

$$\beta^2_{ij}(t+1) = \beta^2_{ij}(t) + \eta \cdot h_{ij} \cdot (u^2_{ij}(t) - \beta^2_{ij}(t)) \tag{7}$$

β_{ij} and b_{ij} are used to fit the logistic function at each node to the distribution of points around it. β_{ij} estimates the standard deviation, and b_{ij} estimates the mean of the distances of the points from the cluster center.

As usual, η, the learning rate, starts out fairly large and is decayed over time. After stage 1 training, the weights are frozen and stage 2 training commences.

In stage 2, we augment the SOM with a set of lateral weights, $v_{ij,kl}$, between all the nodes. The lateral weight from node (i,j) to node (k,l) keeps count of the number of times node (k,l) is the winning node in the time step after node (i,j) is the winning node. After stage 2 training, we normalize the outgoing lateral weights from each node to sum to 1.

During inference, the activation $y_{ij}(t)$ is calculated using equation 1 as before. However, the value of $\alpha_{ij}(t)$ is now calculated by multiplying the activation of each node at the previous time step with its corresponding lateral weight to the current node, and summing.

$$\alpha_{ij}(t) = \sum_{k,l} v_{ij,kl}(t) \cdot y_{kl}(t-1) \tag{8}$$

where $y_{kl}(t-1)$ refers to the activation of node (k,l) at the previous time step, and $v_{ij,kl}(t)$ is the lateral weight from node (k,l) to node (i,j) in the SOM at time step t. Now we normalize the $y_{ij}(t)$ to sum to 1.

Next we provide a Bayesian perspective on the computation being performed by the SOM.

4 A Bayesian Perspective

Let $P(N_i \mid \mathbf{x}_j)$ be the probability of node N_i winning the competition when $\mathbf{x}_j$ is presented as the input. Then, by Bayes' theorem,

$$P(N_i \mid \mathbf{x}_j) = \frac{P(\mathbf{x}_j \mid N_i) \cdot P(N_i)}{P(\mathbf{x}_j)} \tag{9}$$

Thus,

$$argmax_i \, P(N_i \mid \mathbf{x}_j) = argmax_i \, P(\mathbf{x}_j \mid N_i) \cdot P(N_i) \tag{10}$$

If we consider the SOM as a generative model, the probability of input vector $\mathbf{x}_j$ being generated by node N_i decreases monotonically with the distance between $\mathbf{x}_j$ and the weight vector at node N_i. Thus, finding the winning node in a traditional SOM corresponds to finding the maximum likelihood winner (where the prior probability of each node is the same).

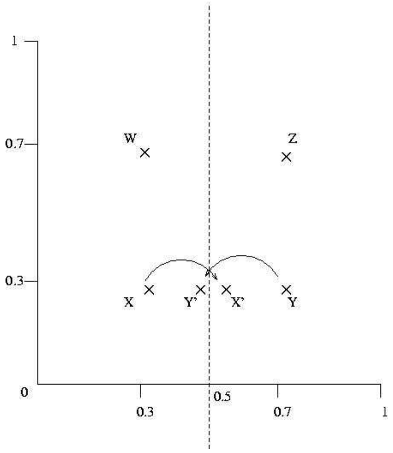

Figure 1. The training vectors.

In the anticipatory SOM, we take advantage of the temporal correlations between the inputs to include an estimate of the prior term in the above equation, and thus compute the maximum *a posteriori* winner. The prior probability of any node is computed recursively as follows:

$$P(N_i(t)) = \sum_k P(N_i(t) \mid N_k(t-1)) \cdot P(N_k(t-1)) \quad (11)$$

The lateral weights estimate the conditional probabilities. Therefore, calculating the prior corresponds to computing the value of $\alpha_{ij}(t)$ in equation 8, using the activations of the nodes from the previous time step and the lateral weights. Thus the lateral weights, $v_{ij,kl}(t)$, correspond to the conditional probabilities, and the activation of the node from the previous time step corresponds to the prior probability.

5 Results

We designed an experiment to demonstrate the improved noise tolerance of the anticipatory SOM, compared to a traditional SOM.

The training set consisted of the four points or vectors in two dimensions shown in figure 1. The four vectors were divided into two groups, A and B. Group A consisted of vectors X and W, and group B consisted of vectors Y and Z. At each time step, one of the vectors was chosen randomly to be presented to the SOM. Temporal correlations were introduced into the data stream by increasing the probability of a vector from a group being chosen if a vector from the same group had been chosen in the previous time step. A noisy version of the vector was generated by picking the actual input vector from a circular Gaussian centered at the chosen vector.

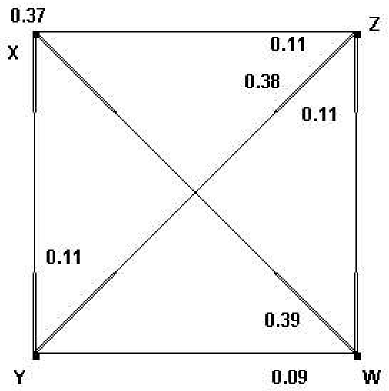

Figure 2. Some of the lateral weights between the nodes. Since the lateral weight matrix turns out to be quite symmetric, not all the weights are shown. Note that the nodes corresponding to group A (vectors X and W) are strongly linked to each other, and the nodes corresponding to group B (vectors Y and Z) are strongly linked to each other.

An anticipatory SOM and a traditional SOM were trained with this data stream for 50000 time steps. After training, the SOM nodes were labeled as either A or B, depending on to which group's vectors they responded maximally. Figure 2 labels the nodes of the anticipatory SOM, including the vectors to which they respond maximally in parentheses. The links between the nodes show the lateral weights. We see that the nodes corresponding to vectors from the same group have strong lateral weights connecting them.

For testing, we create two new vectors $X^{'}$ and $Y^{'}$, which are noisy versions of X and Y. By design, they are noisy enough to fall into the receptive field of a node belonging to the other group (Y and X respectively). Now, the data stream is generated using the same statistics as before, except that X is substituted with $X^{'}$ and Y is substituted with $Y^{'}$. We expect the regular SOM to misclassify each occurrence of these vectors, whereas the anticipatory SOM should classify them correctly if it has the right context. The context is provided by the Z and W vectors, i.e. if $X^{'}$ appears in the time step after W, it should be classified correctly by the anticipatory SOM, but not by the regular SOM. The same goes for $Y^{'}$ and Z also. Figure 3 shows the results of testing. The results are the average of 10 runs. We see that the anticipatory SOM significantly outperforms the regular SOM. Actually, the regular SOM never classifies a vector correctly that the anticipatory SOM does not classify correctly.

6 Conclusions and Discussion

By transforming the self-organized map from a maximum likelihood estimator to a maximum *a posteriori* estimator, we show that we can make it much more robust to noise. By exploiting the temporal correlations in the data stream we can include an estimate of the prior probabilities. It is important to note that the estimation of prior probabilities

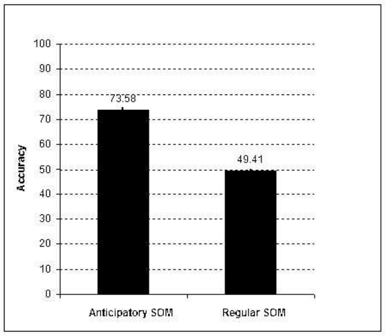

Figure 3. Performance on artificial data. We see that the anticipatory SOM significantly outperforms the traditional SOM

does not have to be done in this way. For example, in 11 we make the Markovian assumption. We could, instead, consider an arbitrarily long history. Many more sophisticated approaches are possible, such as Kalman filters, which have been used to model the visual system [13]. Another interesting effect of including the priors is that a high prior on a node corresponds to a temporarily expanded receptive field, a phenomenon which has been reported in the neuroscience literature [14].

Many interesting applications are possible. Any application which involves sequential processing could use the anticipatory SOM. An obvious example is face detection in a video stream. There are several good face detection systems available already, such as [3]. However, they only work well if the person is looking directly at the camera. Turning the head or tilting the head slightly causes the face detection system to lose the face. In this situation, the idea of temporal persistence of faces in the environment could easily be used to provide anticipations. Our anticipatory SOM could be used in conjunction with such a face detection system to improve it's performance. We intend to do just this using the Illinois Self-Aiming Camera [15], in order to provide it with robust face detection capabilities. The Illinois Self-Aiming Camera is a device being developed by us to model the superior colliculus and to test our ideas about multisensory integration and sensor fusion. We intend to make it more sophisticated and brain-like by including semantic modules such as face detection, voice detection etc. Many of these could use the anticipatory SOM to do robust recognition.

7 Acknowledgements

We would like to thank Professor Thomas Anastasio, Dav Zimak, Rodrigo Braz, Apu Kapadia, and everyone at the CSN group for their comments which helped improve this paper.

8 References

1. Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, Detecting faces in images: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), January 2002.
2. Dorin Comaniciu and Visvanathan Ramesh, Robust detection and tracking of human faces with an active camera, *Proceedings of the 3rd IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 2000, 11-18.
3. Paul Viola and Michael Jones, Robust real-time object detection, *Proceedings of the ICCV workshop on statistical and computational theories of vision*, Vancouver, Canada, July 2001.
4. Andreas Koschan, Sang Kyu Kang, Joonki Paik, Besma Abidi, and Mongi Abidi, Video object tracking based on extended active shape models with color information, *Proceedings of the 1st European Conference on Color in Graphics, Imaging and Vision*, Poitiers, France, 2002, pp. 126131.
5. G. J. Edwards, C. J. Taylor, and T. F. Cootes, Learning to identify and track faces in image sequences, *Proceedings of the 6th IEEE Internation Conference on Computer Vision*, 1998, 317322.
6. G. Chappell and J. Taylor, The temporal Kohonen map, *Neural Networks*, 6, 1993, 441-445.
7. Markus Varsta and Jukka Heikkonen, Analytical comparison of the temporal Kohonen map and the recurrent self-organizing map, *Proceedings of the European symposium on artificial neural networks (ESANN00)*, Bruges, Belgium, April 2000, 273280.
8. Markus Varsta, Jukka Heikkonen, and Jose del R. Millan, Context learning with the self-organized map, *Proceeding of the Workshop on Self-Organizing Maps*, Helsinki University of Technology, Espoo, Finland, 1997, 197202.
9. Thomas Voegtlin, Context quantization and the contextual self-organizing maps, *Proceedings of the International Joint Conference on Neural Networks (IJCNN00)*, Como, Italy, July 24-27 2000.
10. Guilherme de A. Barreto and Aluizio A. R. Araujo, Unsupervised context-based learning of multiple temporal sequences, *Proceedings of the International Joint Conference on Neural Networks (IJCNN99)*, Washington DC, USA, 1999, 11021106.
11. Peter Foldiak, Learning invariance from transformation sequences, *Neural Computation*, 3(2), 1991, 194200.
12. Teuvo Kohonen, *Self-Organizing Maps* (Springer, second edn., 1997).
13. Rajesh P. N. Rao, An optimal estimation approach to visual perception and learning, *Vision Research*, 39(11), 1999, 19631989.
14. A. Das and C. D. Gilbert, Receptive field expansion in adult visual cortex is linked to dynamic changes in strength of cortical connections, *Journal of*

Neurophysiology, 74(2), August 1995, 779792.

15. Samarth Swarup, Tuna Oezer, Sylvian R. Ray, and Thomas J. Anastasio, A self-aiming camera based on neurophysical principles, *Proceedings of the International Joint Conference on Neural Networks (IJCNN03)*, Portland, OR, July 2003, volume 4, pp. 32013206.