Learning Temporal Clusters with Synaptic Facilitation and Lateral Inhibition

Chris L. Baker and Aaron P. Shon and Rajesh P.N. Rao

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350

Abstract

Short-term synaptic plasticity has been proposed as a way for cortical neurons to process temporal information. We present a model network that uses short-term plasticity to implement a temporal clustering algorithm. The model's facilitory synapses learn temporal signals drawn from mixtures of nonlinear processes. Units in the model correspond to populations of cortical pyramidal cells arranged in columns; each column consists of neurons with similar spatio-temporal receptive fields. Clustering is based on mutual inhibition similar to Kohonen's SOM's. A generalized expectation maximization (GEM) algorithm, guaranteed to increase model likelihood with each iteration, learns the synaptic parameters.

Key words: dynamic synapses, facilitation, self organizing maps, competitive learning, mixture models, temporal filtering

Email address: {clbaker,aaron,rao}@cs.washington.edu (Chris L. Baker and Aaron P. Shon and Rajesh P.N. Rao).

1 Introduction

Dynamic synapses in the cortex have been linked to processing of temporal information because of their sensitivity to frequency changes in input [1]. In [2], mean field equations for populations of neurons with facilitory and depressive synaptic inputs are derived, which allow for efficient and accurate approximation of populations with large numbers of inputs. We use a model of populations of neurons with facilitory synapses based on a mean field approximation of networks of spiking neurons with dynamic synapses given in [2,3]. For feedforward neural networks with only one hidden layer, this model was shown in [3] to have the capacity to approximate the class of nonlinear filters that can be implemented with Volterra series.

However, it is unclear how such supervised learning algorithms could be implemented in the brain. A possible mechanism for generating training signals for individual neuronal populations comes from the self-organizing map (SOM) model [4]. This model uses a neighborhood function that determines weights on the error of a population for a given input. Lateral inhibition limits activation to one spatially contiguous neighborhood on each time step. Together, these features determine how the population parameters get updated. One advantage of the SOM method is its biological plausibility. In our model, we link the SOM's biologically plausible rules for local, competitive learning with a commonly used statistical algorithm called generalized expectation maximization (GEM). Under mild assumptions, the algorithm provably increases the likelihood of the model with each iteration.

Section 2 introduces the dynamic synapse model. Section 3 describes the mixture model, and the unsupervised training algorithm that results from estimating the maximum likelihood mixture parameters using GEM. Section 4 presents simulation results on learning mixtures of artificial and natural time series.

2 Dynamic synapse model

The dynamic synapse model we use is equivalent to the model of dynamic synapses given in [3]. This model is a mean-field approximation to populations of spiking neurons based on the model given in [2].

In our model, dynamic synapses receive real valued inputs and update their synaptic weights at discrete time steps. Let $x(t)$ be the real-valued presynaptic input at time t. Let Θ be a set of hyperparameters that determine the temporal behavior of a synapse, in our model consisting of scalars α, β, and μ. The dynamic update of a synaptic weight w is defined by the equation:

$$w(t+1) = w(t) + \alpha\Big(\mu - w(t)\Big) + \beta x(t). \tag{1}$$

The β parameter is constrained to be positive to simulate synaptic facilitation. As argued in [3], facilitory synapses can approximate any nonlinear filter whose response can be described using a Volterra series.

2.1 Synaptic dynamics

Expressing the dynamic update of a synaptic weight as a differential equation yields:

$$\frac{dw(t)}{dt} = \alpha\Big(\mu - w(t)\Big) + \beta x(t). \tag{2}$$

The solution to equation (2) is:

$$w(t) = \mu + \beta \int_0^\infty e^{-\alpha\tau}\, x(t-\tau) d\tau. \tag{3}$$

Equation (3) shows that our model of a dynamic synapse implements a temporal filter with fading memory over previous inputs. The hyperparameter α determines the decay rate of the temporal filter. The parameter β determines the amount of facilitation of the filter.

2.2 Supervised spatio-temporal learning with dynamic synapses

Let $\mathbf{w}(t)$ be a vector of dynamic synaptic weights at time t. Let $\mathbf{x}(t)$ be an input vector at time t. The output $y(t)$ of a neuronal population with dynamic synapses for input $\mathbf{x}(t)$ is:

$$y(t) = \sigma\Big(\mathbf{w}(t)^{\mathrm{T}}\mathbf{x}(t)\Big) \tag{4}$$

where σ is the sigmoid function, defined as $\sigma(x) = 1/\Big(1 + \exp(-ax)\Big)$. Below, we describe our conjugate algorithm for optimizing model parameters.

Let M be the dimension of the input to a single unit. Let L be the number of paired input/output matrices given as supervised training for the neuron. Let $\mathbf{X}_l$ be the lth matrix of inputs, of dimension $M \times T$, for $l = 1 \ldots L$. Let $\mathbf{y}_l$ be a vector of real valued outputs of dimension $1 \times T$ for input $\mathbf{X}_l$, and let $\mathbf{d}_l$ be a vector of supervised training values for input $\mathbf{X}_l$. Let $\mathbf{e}_l = \mathbf{d}_l - \mathbf{y}_l$ be a vector of error values for $t = 1 \ldots T$. Then:

$$SSE = \sum_{l=1}^{L} \mathbf{e}_l^{\mathrm{T}} \mathbf{e}_l \tag{5}$$

is the sum squared error of the output of a neuron on input l.

Minimizing SSE with conjugate gradient requires computing first partial derivatives and Hessians of SSE with respect to each of the hyperparameters α, β, and μ. By the chain rule, we can express the partial derivatives of SSE with respect to the hyperparameters as products of two terms. The value of this partial derivative for synapse n of a unit, for hyperparameter $\theta^n \in \Theta^n$ is:

$$\frac{\partial SSE}{\partial \theta^n} = \sum_{l=1}^{L} \sum_{t=1}^{T} \frac{\partial SSE}{\partial w_l^n(t)} \frac{\partial w_l^n(t)}{\partial \theta^n}. \tag{6}$$

The term $\frac{\partial SSE}{\partial w_l^n(t)}$ is given by:

$$\frac{\partial SSE}{\partial w_l^n(t)} = -ae_l(t)y_l(t)\Big(1 - y_l(t)\Big)x_l^n(t). \tag{7}$$

The partial derivatives corresponding to each hyperparameter are given by:

$$\frac{\partial w_l^n(t)}{\partial \alpha^n} = (1-\alpha^n)\frac{\partial w_l^n(t-1)}{\partial \alpha^n} - w_l^n(t-1) + \mu^n \tag{8}$$
$$\frac{\partial w_l^n(t)}{\partial \beta^n} = (1-\alpha^n)\frac{\partial w_l^n(t-1)}{\partial \beta^n} + x_l^n(t-1) \tag{9}$$
$$\frac{\partial w_l^n(t)}{\partial \mu^n} = (1-\alpha^n)\frac{\partial w_l^n(t-1)}{\partial \mu^n} + \alpha^n. \tag{10}$$

These equations are recursive, and depend on the entire sequence of values for $w_l^n(t)$ and $x_l^n(t)$. For the full recursive expressions for the Hessians, see [5].

3 Learning spatiotemporal clusters with dynamic synapses

Cortical columns contain excitatory, recurrent, local connections, and mutually inhibitory long-range connections. This physiology motivates Kohonen's SOM model [4]. The SOM model spatially clusters its input data using lateral inhibition and the notion of spatially contiguous neighborhoods in which synaptic modification occurs.

3.1 Mean-field computational model

In our model, input patterns are spatio-temporal; that is, each input sample is independently generated by one of several nonlinear stochastic processes. Our mean-field model of association cortex therefore performs *spatio-temporal clustering*, where nearby cortical columns are assumed to be enervated by afferents from a single, spatially contiguous region of sensory cortex. Within each spatial region of nearby columns, individual columns respond to sensory firing patterns assumed to be generated by particular stochastic processes. When a spatio-temporal input pattern arrives, nearby columns compete for activation. The columns with the highest activations are considered "winners" of the competition. The higher a column's activation, the greater the probability that the stochastic process it encodes produced the input pattern.

Our model is similar to the model in [6]. However, unlike the model in [6], our temporal filtering units correspond to cortical columns. Additionally, our expectation maximization algorithm is guaranteed to increase the likelihood of the generative model with each iteration. As such, our model also admits a convenient probabilistic interpretation: the firing rates of nearby columns comprise a distribution over spatio-temporal processes being observed by sensory cortices. In accordance with our probabilistic interpretation, the model assumes that error likelihoods follow a Gaussian distribution.

As a practical example, consider the task of identifying the velocities of rapidly-moving objects in a monkey's visual field. For a single spatial neighborhood in the monkey's retinotopic map, an object might be perceived as a single moving dark point. Movement of the point (and thus the object) at different speeds is generated by a different temporal sequence (or process). We propose that a different cortical column classifies each temporal sequence. During learning, lateral inhibition encourages competition for synaptic modification across columns. After learning, lateral inhibition causes competitive normalization of activity levels, ensuring that activities of columns within a single spatial neighborhood encode a probability distribution (Fig. 2(a)). Similar clustering could take place in, for example, auditory cortex, for classifying speech signals.

3.2 Unsupervised spatio-temporal clustering using expectation maximization

Expectation maximization (EM) [7] is an iterative algorithm where each iteration consists of two steps. Let C be the number of cortical columns. The E step computes the likelihood that column $c \in \{1 \ldots C\}$ selects for a particular spatio-temporal input pattern $\mathbf{x}_l$, $l \in \{1 \ldots L\}$. During the M step, each column tunes its hyperparameters to maximize activation in response to patterns likely generated by the stochastic process the column encodes. Weight vectors $\mathbf{w}$ are tuned to maximize the normalized dot product between the weights and the input patterns. This maximizes the activation of a column when the response of its dynamic synapses most closely resembles the input pattern. Let Θ_c^i denote the hyperparameters of column c on iteration i. Each input pattern is a vector $\mathbf{x}_l$, assumed independent of the other patterns and corrupted by zero-mean Gaussian noise. With a Gaussian error likelihood Q takes the form:

$$Q = \prod_{l=1}^{L}\prod_{c=1}^{C}\left[p\left(\mathbf{w}_l^c|\mathbf{x}_l, \Theta^c\right)\right] \tag{11}$$

$$Q = \prod_{l=1}^{L}\prod_{c=1}^{C}\left[\exp\left((\mathbf{w}_l^c - \mathbf{x}_l)^{\mathrm{T}}(\mathbf{w}_l^c - \mathbf{x}_l)/k\right)\right] \tag{12}$$

$$\log Q = \frac{1}{k}\sum_{l=1}^{L}\sum_{c=1}^{C} SSE_l \tag{13}$$

The M step uses the gradient of the expectation to maximize the likelihood of model parameters on the next iteration, given the expectations over state values. We denote the set of parameters for the next iteration as $\{\Theta^{1'}, \Theta^{2'}, \ldots, \Theta^{C'}\}$. Since the parameters for each column are chosen independently, we have:

$$\{\Theta^{1'}, \Theta^{2'}, \ldots, \Theta^{C'}\} = {}^{\arg\max}_{\;a} \sum_{c=1}^{C} \log p\left(\Theta^{c'}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L, \Theta^{1^a}, \Theta^{2^a}, \ldots, \Theta^{C^a}\right) \tag{14}$$

Note that a prior probability on each mixture component on the activation of each column c could be added; here we assume a uniform prior over all activation values for all columns. A nonuniform prior distribution might correspond, e.g., to a "top-down" activation signal arriving from distant association cortices or prefrontal cortex.

4 Results

Fig. 1 demonstrates how a single dynamic synapse, trained in a supervised manner using our EM algorithm, learns to reproduce a temporal sequence of desired outputs. Further simulation results demonstrate how our unsupervised

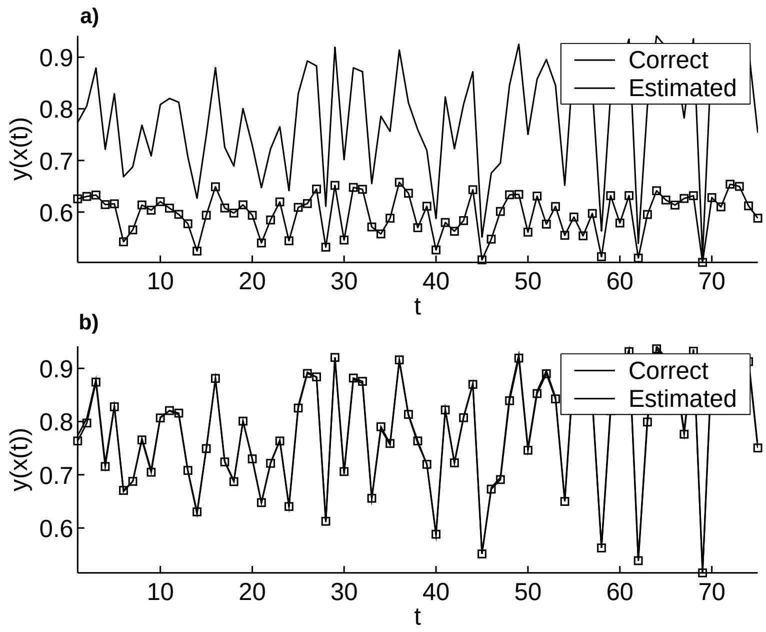

Fig. 1: Supervised spatio-temporal learning with dynamic synapses. (a) Output of a column compared with training signal before learning. (b) Output of a column compared with training signal after learning.

clustering algorithm can differentiate between 3-dimensional visual stimuli moving at different speeds. The simulations use two model cortical columns; after learning, each column encodes a different velocity of motion. Fig. 2 shows that the algorithm correctly identifies the temporal clusters, and that weights of the column accurately learn the temporal sequence.

5 Conclusion

The brain undoubtedly employs plasticity on several timescales to process temporal signals. Dynamic synapses are promising candidates for classifying and predicting quickly fluctuating time series. We have demonstrated a population-level model for how dynamic synapses can cluster temporal input sequences. Learning in the model can be performed in a supervised or in an unsupervised manner. Future work will involve expanding our framework to handle long time courses with many clusters, with data drawn from several different sensory modalities such as audition or vision.

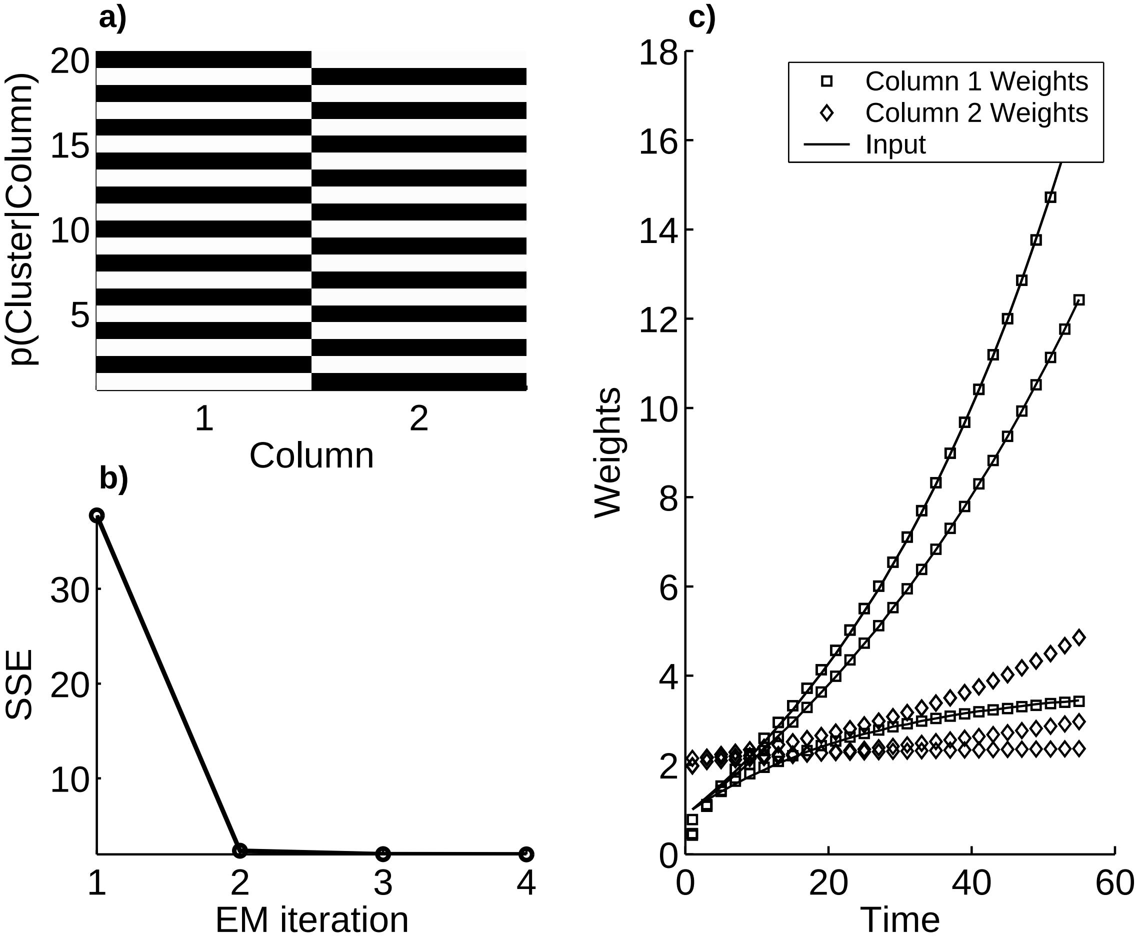

Fig. 2: Unsupervised spatio-temporal clustering with dynamic synapses. (a) Probability that a column selects for an input sequence. The lighter the square, the higher the probability. 20 inputs were presented from two stochastic processes, and the algorithm correctly identified them. (b) Sum squared error versus EM iteration. The algorithm converges after 1 iteration in this case. (c) Weights vectors of two columns compared with input vectors over one sequence. The weights of the column with the highest activation accurately track the input , while the weights of the column with the lowest activation do not.

References

[1] J. A. Varela, K. Sen, J. Gibson, J. Fost, L. Abbott, S. B. Nelson, A quantitative description of short-term plasticity at excitatory synapses in layer 2/3 of rat

primary visual cortex, The Journal of Neuroscience 17 (20).

[2] M. Tsodyks, K. Pawelzik, H. Markram, Neural networks with dynamic synapses, Neural Compuatation 10 (821-835).

[3] W. Maass, E. D. Sontag, Neural systems as nonlinear filters, Neural Computation 12 (1743-1772).

[4] T. Kohonen, Self-Organizing Maps, Springer, 1997.

[5] C. L. Baker, A. P. Shon, R. P. N. Rao, Temporal clustering with dynamic synapses, Tech. rep., University of Washington Dept. of Computer Science and Engineering (2004).

[6] J. Storck, F. Jäkel, G. Deco, Temporal clustering with spiking neurons and dynamic synapses: towards technological applications, Neural Networks 14 (2001) 275–285.

[7] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Royal Statistical Soc., Series B 39 (1) (1977) 1–38.

[8] L. Abbott, J. Varela, K. Sen, S. Nelson, Synaptic depression and cortical gain control, Science 17 (20).

[9] J.-S. Liaw, T. W. Berger, Dynamic synapse: A new concept of neural representation and computation, Hippocampus 6 (591-600).

[10] A. D. Back, A. C. Tsoi, A simplified gradient algorithm for IIR synapse multilayer perceptrons, Neural Computation 5 (456-462).

[11] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1994.