Spiking Correlation Matrix Memory

D Kustrin [1] J Austin

Department of Computer Science, University of York, York, UK

1 Introduction

Correlation Matrix Memories are feed-forward static binary associative neural networks equivalent to a single layer fully connected feed–forward network with binary weights. Their simplicity has allowed extensive theoretical analysis as well as numerous industrial applications in areas as diverse as symbolic reasoning (Kustrin and Austin 1999), molecular structure matching (Turner and Austin 1997) and financial forecasting (Kustrin and Austin 1998). Originally, CMMs were inspired by biological neuronal constructions but as many other biologically-inspired models they were modelled and constructed as mathematically continuous abstractions; the natural coding strategy of using spikes was abandoned for more understandable and amenable to analysis continuous models. CMMs, unlike most other models, remained in the binary input binary output domain but the internal workings were very much non-spiking. This paper looks at construction of CMMs as purely spiking entities.

2 Correlation Matrix Memory

Correlation Matrix Memories are simple binary sum-and-threshold memories (Austin 1995). Each output from a CMM is generated by computing a (binary) weighed sum of inputs which is then thresholded at some value to produce a binary output. Since thresholding is used to select which outputs will be present in the final output vector, a number of techniques have been developed to suit various applications. Fixed-weight thresholding is the simplest in which an output is selected if its sum is greater then some fixed value; maximum–activation technique thresholds all the outputs at the level of the maximal activation of all output neurons; k-point thresholding sets the threshold level so that at least k outputs are present in the output vector; Willshaw thresholding sets the cut-off point at the same level as the number of bits set in the input vector (Casasent and Telfer 1987). Current implementations of SCMM exist for both fixed-weight and Willshaw techniques and maximum–activation is currently being tested.

The basic CMM operations are very simple. Given a set of binary input–output pairs represented in form $I_1 \to O_1, \quad I_2 \to O_2 \quad \ldots \quad I_n \to O_n$ training can be expressed as a simple binary disjunction of inner products of the input–output column vectors (a form of Hebbian learning): $M^0 = 0$, $M^i = M^{i-1} \vee O_i I_i^{\mathrm{T}}$. Training can be viewed as construction of a mapping from the input to the output vector space and hence the CMM can be viewed as a simple mapping operator. In this framework, recall becomes simple projection of the input pattern from the input vector space to the output space:

$$O(I) = \Theta\left(MI\right) \tag{1}$$

where Θ is some threshold (transfer) function and I is some input. The need for the threshold function is intuitive: unless the input is orthogonal and of unit length, the output vector would not be a binary vector, but would contain output multiplied with the square of the input vector length.

3 Spiking Correlation Matrix Memory

The spiking correlation matrix memory is a direct implementation of the CMM recall equation (eq. 1): firstly a weighed sum of the inputs is computed which is then thresholded at some set value. These operations are best explained using the spike–response model (Gerstner 1999). Each neuron i is described by a state variable u_i which is a linear superposition of it's own output history negative contributions and the contribution from it's pre-synaptic neurons:

$$u_i(t) = \sum_{t_i^{(f)} \in \mathcal{F}_i} \eta_i(t - t_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in \mathcal{F}_j} w_{ij} \epsilon_{ij}(t - t_j^{(f)}) \tag{2}$$

[1] Supported by EPSRC ROPA grant GR/L75559

where $t_i^{(f)}$ it the time when the neuron i fires an output spike, Γ_i is the set of all pre-synaptic neurons for neuron i, w_{ij} is the weight between neurons i and j, η and ϵ are kernels describing self and pre-synaptic neuron contributions respectively and $\mathcal{F}_i = \{t | u_i(t) = \vartheta \wedge \dot{u}_i(t) > 0\}$ is the set of all firing times for neuron i. The self-contribution refractory kernel, η, is usually modelled by the following equation:

$$\eta_i(s) = -\alpha \exp\left(-\frac{s}{\tau}\right) \mathcal{H}(s) \tag{3}$$

with α being the value the state will have at the moment of firing (ie $u_i(t) = -\alpha$), τ a time constant and $\mathcal{H}(s)$ the Heaviside function. If a neuron has a maximum of N inputs then the maximal EPSP it receives will be $N \times \epsilon(s)$. This suggests that the refractory kernel should be similarly scaled to ensure that spikes are not produced during the excitatory potential. This is achieved by modifying the refractory kernel in eq. 3 to be

$$\eta_i(s) = -N \exp\left(-\frac{s - S_{max}}{\tau_m}\right) \mathcal{H}(s) \tag{4}$$

where S_{max} is the time the sum of all EPSEs would reach $N\left((\tau_s/\tau_m)^{-\tau_s/\tau_s - \tau_m} - (\tau_s/\tau_m)^{-\tau_m/\tau_s - \tau_m}\right)$ which is its maximum.

The response to inputs in form of excitory post-synaptic potentials (EPSPs) is usually modelled with the following equation:

$$\epsilon_{ij}(s) = \left[\exp\left(-\frac{s - \Delta^{AX}}{\tau_m}\right) - \exp\left(-\frac{s - \Delta^{AX}}{\tau_s}\right)\right] \mathcal{H}\left(s - \Delta^{AX}\right) \tag{5}$$

where τ_s and τ_m are time constants and Δ^{AX} is the axonal transmission delay. Spike–response model formulation indicates that all pre-synaptic spike inputs are summed (albeit in a nonlinear fashion) and that the timing of the incoming spikes will have a large impact on the nonlinearity of the sum.

Although relatively simplistic, spike–response model is a good approximation to the actual neuron behaviour; it has also been shown that it is in fact a more general model then integrate-and-fire model which is it's special case (Gerstner 1999).

Summing

If n spikes are received simultaneously by a neuron and the threshold level was set at ϑ, then the time from receiving the input spike to the time of output spike was generated encodes weighed sum of input spikes. In a slightly simplified model when $\tau_m = 2\tau_s$, threshold value will be reached at

$$S_n(s) = \Delta^{AX} + 2\tau_s \ln(2) \; - \; 2\tau_s \ln\left(1 \pm \frac{\sqrt{n^2 - 4n\vartheta}}{n}\right) \tag{6}$$

with the first occurrence being at the positive sum case. It is clear that this is a nonlinear equation in n and that temporal distances nonlinearly encode summed values.

Thresholding

Equation 6 suggests that timing of the incoming spikes has a great impact on a neuron's state. This is manifested by the fact that summing is nonlinear which results in greatest state increase when incoming spikes are coincidental and lesser impact by spikes that are further apart in time. This effect can be used to detect (relative) equality of spike encoded data. Consider eq. 2. From the equation it is clear that for some preset values of the timing constants τ_s and τ_m, it is possible to find value for the threshold ϑ which will allow detection of simultaneity of arrival of two or more spikes with arbitrary error (temporal distance). For example, if values are set at $\tau_s = 1.0$, $\tau_2 = 10.0$ and $\vartheta = 0.3$, the temporal distance between two spikes would have to be below (approx) 4.585 time units for their occurrence to be detected as being coincidental, for $\vartheta = 0.4$ the distance is reduced to approx. 2.197 and for $\vartheta = 0.5$ the distance is zero.

It is clear that coincidence detection can be used for thresholding. In the Willshaw threshold model the threshold is set to be the number of synchronised spikes on input which is simply the sum of all inputs with all weights set to one. An auxiliary neuron performing this sum will produce a time coded value which can then be compared to outputs of other neurons with pre-tuned precision which would result in allowing only the time-encoded sums with the same value as the auxiliary neuron. Fixed weight thresholding is produced without the need of coincidence detection by utilisation of inhibitory synapses.

SCMM Architecture

Different thresholding techniques require slightly different architectures with the summing layer being the invariant. The Willshaw thresholding is implemented by using coincidence detection layer of neurons and an auxiliary neuron in addition to the input neurons, with the total of $2n+1$ neurons. The fixed-weight threshold architecture uses inhibitory synapses for thresholding and thus requires only $n+1$ neurons.

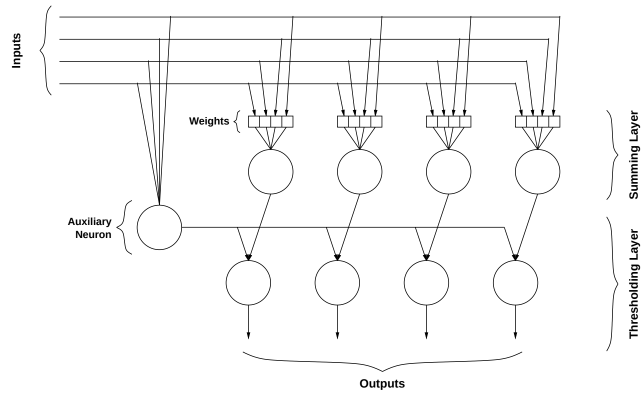

Figure 1: Willshaw threshold SCMM.

4 Example

As a toy problem that will illustrate SCMM operation the following set of relations, $\{A \to B,\ B \to C,\ C \to A\}$, and their binary codings $A = [1001]$, $B = [0110]$, $C = [0011]$, $D = [0001]$ were converted into a CMM weights matrix

$$w = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{7}$$

This weight matrix was then used in construction of a Willshaw threshold SCMMs. All SCMM neurons had time constants fixed at $\tau_m = 2\tau_s$ and $\tau_s = 5.0$ with threshold at $\vartheta = 0.24$. The threshold layer neurons had their timing constants set at $\tau_m = 10.0$ and $\tau_s = 1.0$, to facilitate more accurate discrimination of summed outputs. The Willshaw SCMM architecture is presented in fig. 2. The architecture is recurrent and by presenting input vector corresponding to symbol A it should produce the sequence $A \to B \to C$ after which it will enter a cycle of producing vectors [1011] and [0010] due to the nature of the Willshaw method. The output of the simulation is presented in fig. 3.

References

Austin, J. (1995). Distributed associative memories for high speed symbolic reasoning. *International Journal on Fuzzy Sets and Systems 82*(2), 223–233. Invited paper to the special issue on Connectionist and Hybrid Connectionist Systems for Approximate Reasoning.

Casasent, D. and B. Telfer (1987). Associative memory synthesis, performance, storage capacity and updating: new heteroassociative memory results. In *Proceedings of the SPIE conference on Intelligent Robotics and Computer Vision (IRCV '87)*.

Gerstner, W. (1999). Spiking neurons. In W. Maass and C. M. Bishop (Eds.), *Pulsed Neural Networks*, pp. 3–54. MIT Press.

Kustrin, D. and J. Austin (1998). *Forecasting Financial Time Series with Correlation Matrix Memories for Tactical Asset Allocation.* Ph. D. thesis, Department of Computer Science, University of York.

Kustrin, D. and J. Austin (1999). Connectionist graph traversal implementing a simple propositional logic. In D. Kustrin (Ed.), *Third Weightless Neural Networks Workshop*, York.

Turner, M. and J. Austin (1997). A neural network technique for chemical graph matching. In M. Niranjan (Ed.), *Proceedings of the Fifth International Conference on Artificial Neural Networks.* IEE.

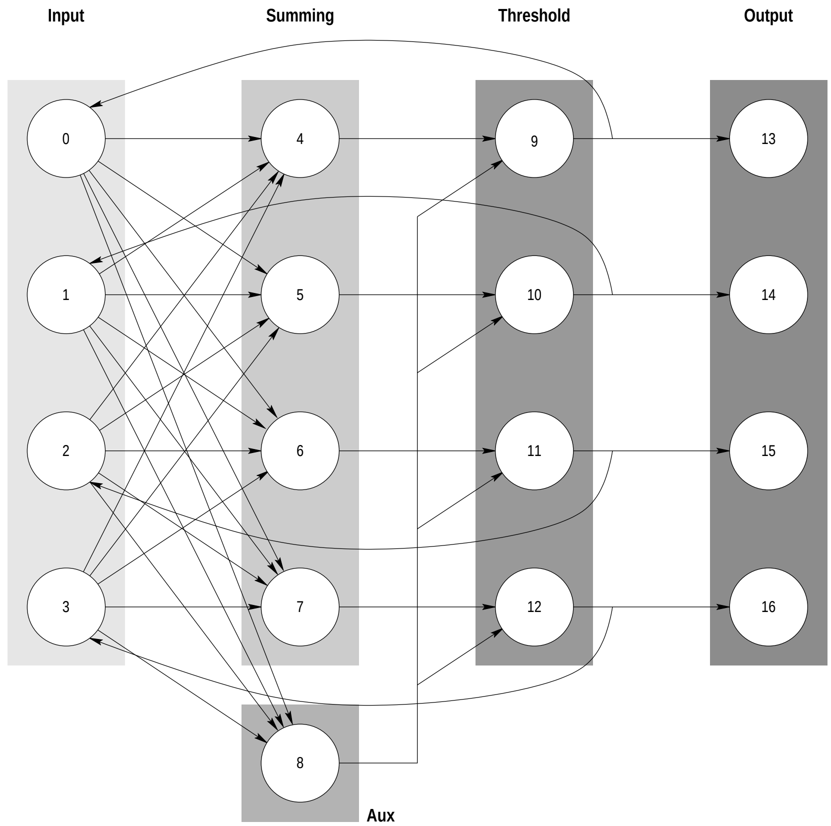

Figure 2: The Willshaw-threshold SCMM architecture used in a the example CMM. All non-thresholding had threshold at 0.25 and timing constants set at $\tau_m = 2\tau_s$ and $\tau_s = 5.0$. Thresholding neurons were set at $\vartheta = 0.5$, $\tau_m = 10.0$ and $\tau_s = 1.0$. Axonal delay for all neurons was set at 80 time units.

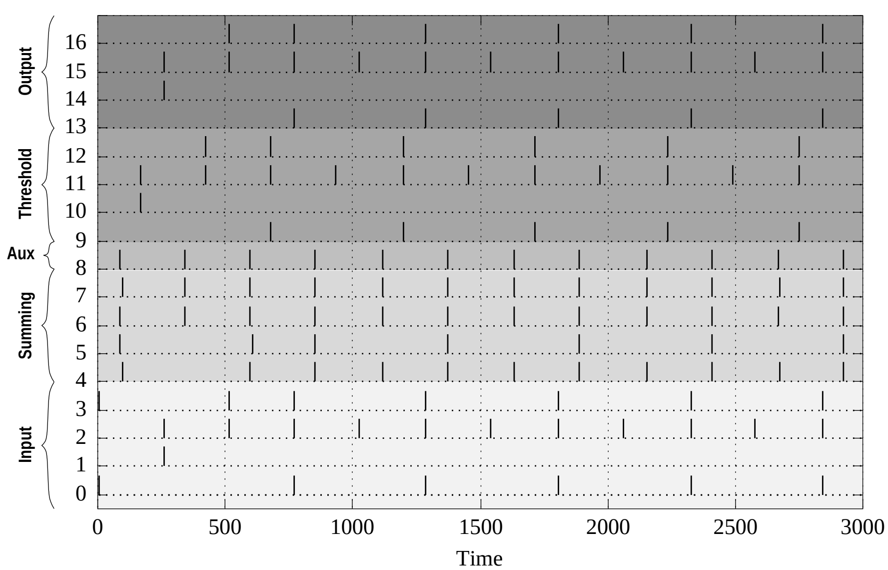

Figure 3: Spike trains for all neurons in the Willshaw-threshold SCMM used in a the example CMM.