A Neural Model of Temporal Sequence Generation with Interval Maintenance[1]

DeLiang Wang and Michael A. Arbib

Center for Neural Engineering, University of Southern California
Los Angeles, CA 90089-2520, USA

Abstract

Based on an interference theory of forgetting in short-term memory (STM), we model STM by a network of neural units with mutual inhibition. Sequences are acquired by combining a Hebbian learning rule and a normalization rule with sequential system activation. As long as sequences are acquired, they can be recognized without being affected by speeds in presentation. The model of sequence reproduction consists of two reciprocally connected networks, one of which behaves as sequence recognizers. Reproduction of complex sequences is shown to be able to maintain interval lengths of sequence components. A mechanism of degree self-tuning based on a global inhibitor is proposed for the model to optimally learn required context lengths in order to disambiguate associations in complex sequence reproduction.

Introduction

A temporal sequence S is denoted as: p_1-p_2-...-p_N, and the length of a sequence is the number of components in the sequence. Any p_i-p_{i+1}-...-p_j, where $1 \le i \le j \le N$, is called a subsequence of S. If S contains repetitions of the same subsequence, like A-B in C-A-B-D-A-B-E, it is called a complex sequence, otherwise a simple sequence. In complex sequences, the correct successor can be determined only by knowing a subsequence prior to it. We refer to the prior subsequence required to cue unambiguously the current symbol p_i in S as the context of p_i, and the length of the context as the degree of p_i. The degree of a sequence is the maximum degree of its components.

Neural networks to reproduce a temporal sequence of input stimuli have been previously studied by a number of investigators (among others see Grossberg 1969; Dehaene, Changeux, & Nadal 1987; Kühn, van Hemmen, & Riedel 1989). In most of these models, reproduction of complex sequences poses great difficulty. Recently, we have proposed a new mechanism for learning temporal sequences (Wang & Arbib 1990) in which we model STM by units comprising recurrent excitatory connections between two local neuron populations. Each neuron population is represented by a single quantity corresponding to local field potential. The activity induced by an input signal to a unit oscillates with damping. By applying a Hebbian learning rule at each synapse and a normalization rule among all synapses to a unit, we have demonstrated that the neural networks with this model of STM are able to learn and reproduce complex temporal sequences. What distinguishes our model from others are two basic hypotheses embodied in the model: (1) We assume that there is a common mechanism to process both complex sequences and simple sequences; (2) Reproduction of a component in a sequence is based on recognition of the context of the component.

Since STM is modeled by decay, it has a fixed temporal course, which makes the previous model unable to handle the *time-warp* problem. For a solution to the time-warp problem, we wish that a network can recognize a time-warped sequence for sequence recognition, whereas for reproduction we wish that a network can reproduce a sequence with the same temporal course as the learned sequence. This is the central theme of the present paper.

A Computational Model of STM

A model of STM must provide the following four basic functions:

(1) Maintaining a symbol for a short time period. What causes forgetting? An *interference* theory proposes that other materials or tasks interfere with memory and thus cause forgetting. A *decay* theory proposes that forgetting occurs even if the subject had to do nothing over the retention interval, so long as the subject did not rehearse the material. (2) Maintaining a number of symbols. Miller (1956) tells us that the number is about seven. (3) Coding the order of input symbols. (4) Coding the length of the presentation of each symbol. The function of STM provides first level information for solving the time-warp problem. When learning a sequence, one can recognize it even though each component of the sequence is presented at considerably different intervals. This function is called *interval invariance*. Yet, a professional musician can recall a multiple-page score, reproducing almost exactly the memorized length of each note. This function is called *interval maintenance*.

Our previous model cannot code the length of each symbol presentation, and therefore cannot solve the time-warp problem. Furthermore, the model conforms with the decay theory of forgetting, whereas the current majority view seems to be that, although some decay may occur, the amount of forgetting caused by decay is substantially less than the amount caused by interference (Murdock 1987). Our following model is based on the interference theory.

Let us assume that there are n memory units, numbered 1, 2, ..., n, with each unit inhibited by all the other units,

[1] The research described in this paper was suported in part by grant no. 1RO1 NS 24926 from the NIH (M.A.A, PI).

as shown in Fig.1. Each unit receives an external input E_i, which is 1 so long as the external input is on and 0 otherwise. The *internal* state of unit i, s_i, is defined as

$$s_i(t) = \begin{cases} 1 & \text{if } E_i(t)=1, E_i(t-1)=0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

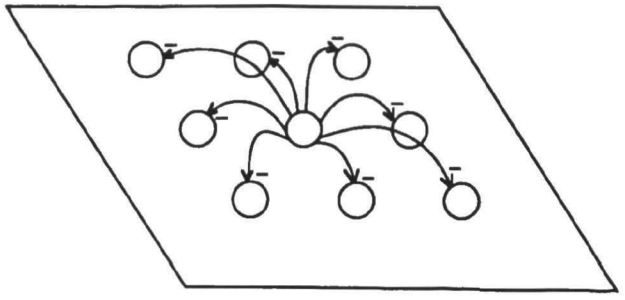

Figure 1. Diagram of the STM model. Each unit projects and inhibits all the other units in the model. Shown in the figure is only outgoing projections from one unit. Minus sign indicates inhibition.

From the definition we can see that the internal state is activated only by the beginning of an external input. The *excitation level* of each unit has value range {0, 1, ..., T}, and is defined as

$$x_i(t) = \begin{cases} T & \text{if } s_i(t)=1 \\ x_i(t-1) - 1 & \text{if } x_i(t-1)>0,\ y_i(t)=1 \\ x_i(t-1) & \text{otherwise} \end{cases} \quad (2)$$

where y_i represents overall inhibition that unit i receives from the other units, formulated as

$$y_i(t) = f(\sum_{j \neq i} s_i(t-1) - 1) \quad (3)^2$$

$$\text{with } f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

From the above definitions we see that whenever $s_i(t) = 1$, $x_i(t)$ is brought to its highest value T and unit i is activated. If any of the units is activated, the inhibition that it exerts on the rest of the network will drive all other active units, i.e. those whose excitation levels are larger than 0, down to the next lower level.

This model satisfies the above four requirements for an adequate STM model. It preserves a symbol on a unit whose excitation level codes the item. Let us assume that external inputs arrive at STM serially (it is easy to serialize simultaneous inputs by a competitive network). Any new item input to STM decrements the excitation levels of all active units in STM. Therefore STM can at most code T items so that T is the capacity of the STM model. A symbol gets lost from the STM model because there are other more recent symbols input into the model, conforming with the interference theory. The order of input symbols is coded since the larger the excitation level of a unit, the more recent is the symbol represented by the unit. Finally, the length of a symbol's presentation is reflected by the time period while the corresponding external input is on, and its coding will be given later.

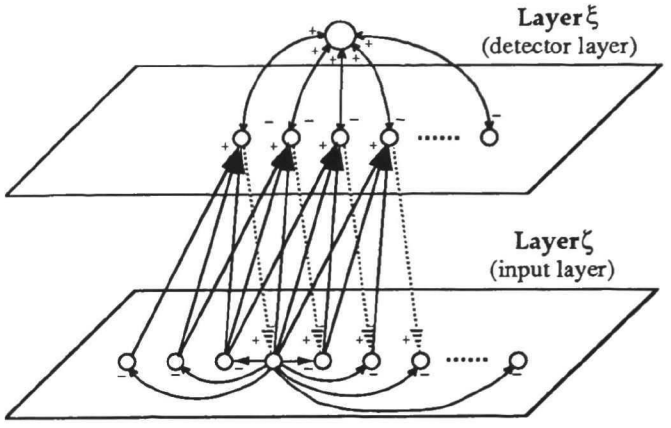

Figure 2. Architecture for complex sequence reproduction. Within layer ζ (the input layer), every unit inhibits every other one to form the STM model shown in Fig.1. Within layer ξ (the detector layer), all units project to a global inhibitor which further projects back to them. Plus sign indicates excitation, and minus sign indicates inhibition.

Network Architecture

The structure of the model for sequence reproduction has two layers, as shown in Figure 2. Layer ζ is called the input layer, which basically serves as a STM model shown in Fig.1. Multiple occurences of a particular symbol in a sequence is represented by one single unit in this layer, so different units represent different spatial patterns in layer ζ. Units in layer ξ function as sequence detectors, and there is a global inhibitor within this layer (see footnote 2). These units recognize the contexts of individual components in a sequence, and anticipate the occurence of these components. Layer ξ connects with layer ζ bidirectionally, and before training connections between them are complete. The projections shown in Fig.2 depict what results from training, such that unit i in layer ξ receives projections only from those units in ζ that represents symbols in the context detected by unit i, and unit j in layer ζ only receives units in ξ that anticipate the occurence of the symbol represented by unit j. This resulted connection pattern is formed through learning. During the training process, a sequence with

[2] Since the weights of inhibitory connections are the same, the mutual inhibitory connections can be replaced by an global inhibitor. An global inhibitor can reduce the number of connections by one order of magnitute, but results in a less reliable system due to information centralization in the inhibitor.

various component intervals is presented to layer ζ. At the end of each component presentation, a unit in layer ξ is randomly selected (but fixed in successive trainings) to fire. The recurrent connections from layer ξ to layer ζ are formed according to a Hebbian rule as following. If unit i in layer ζ (recorded as $<i, \zeta>$) and unit j in layer ξ (recorded as $<j, \xi>$) are firing simultaneously then a connection link from $<j, \xi>$ to $<i, \zeta>$ is established, and its weight will be defined later. All connection weights from units in ξ to ones in ζ are initially zero.

We proposed in the previous paper (Wang & Arbib 1990) that a unit was represented by an expanded network, such that it has multiple terminals to hold different occurences of a symbol. Each terminal directly connects to other units, and thus a unit has multiple channels to connect to another unit. The following description combines this idea for solving the overwriting problem with the new STM model.

Suppose unit $<i, \zeta>$ has m terminals, and the excitation level of its rth terminal is represented by x_{ir}. The STM model (Eq.1 through Eq.4) and the definitions of E_i, s_i, and y_i remain the same except

$$x_{ir}(t) = \begin{cases} T & \text{if } s_i(t)=1, r=1 \\ x_{i,r-1}(t-1) - 1 & \text{if } s_i(t)=1, r>1, x_{i,r-1}(t-1)>0 \\ x_{ir}(t-1) - 1 & \text{if } s_i(t)=0, x_{ir}(t-1)>0, y_i(t)=1 \\ x_{ir}(t-1) & \text{otherwise} \end{cases} \quad (5)$$

The global inhibitor in layer ξ receives input from all units in the layer and projects back to them. A degree parameter d_i is introduced for $<i, \xi>$, and it affects the dynamics of the internal state of $<i, \xi>$ in the following way

$$s_i^{\xi}(t) = f(\sum_{j=1}^{n} \sum_{r=1}^{m} W_{ij}^{r} h(x_{jr}(t-1), d_i) + I_i^{\xi}(t-1) - I_i^{\xi}) \quad (6)$$

$$h(x, y) = \begin{cases} x & \text{if } x > T - y \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where label ξ in (6) indicates layer ξ, x_{jr} is the excitation level of the rth terminal of unit $<j, \zeta>$, and W_{ij}^{r} represents the connection weight from the rth terminal of unit $<j, \zeta>$ to $<i, \xi>$. Symbols n and m stand for the number of units and the number of terminals for each unit in layer ζ respectively. The domain of d_i is $\{1, 2, ..., T\}$. Through function $h(x, y)$ the role of d_i is to gate in certain excitation levels of units in layer ζ. Obviously, the larger is d_i, the more items can $<i, \xi>$ sense from layer ζ. Learning, or modification of connection weights W_{ij}^{r}, follows a Hebbian rule and a later normalization as follows

$$\begin{cases} \widehat{W}_{ij}^{r}(t) = W_{ij}^{r}(t-1) + C_i s_i^{\xi}(t) h(x_{jr}(t), d_i) \\ W_{ij}^{r}(t) = \widehat{W}_{ij}^{r}(t) / \sum_{j'=1}^{n} \sum_{r'=1}^{m} \widehat{W}_{ij'}^{r'}(t) \end{cases} \quad (8)$$

where C_i is a gain factor of learning. The effect of learning on the detector is to change the distribution of all weights to that unit, so it is reasonable to assume that initially all weights are set equal.

Since without a further activation the excitation level of a unit in layer ζ is monotonically decreasing, the formal analysis in our previous paper (Wang & Arbib 1990) applies. In particular, if the threshold of unit $<i, \xi>$ in (6) is set as

$$I_i^{\xi} = \frac{2}{d_i(2T-d_i+1)} \sum_{i=1}^{d_i} (T-d_i+i)^2 \quad (9)$$

then the result of training is to build up activity so as to fire the detector by the presentation of a specific subsequence. Furthermore, after the detector has learned the sequence (simple or complex), only presentation of that sequence induces the maximum activity on the detector unit, regardless of presentation speed of the sequence. The idea behind this interval invariance is that during presentation of a sequence component, only the beginning portion of presentation is captured by the recognition model (cf. Eq.1), and therefore it does not matter how long that presentation lasts. Separation of effective input from external input is an intrinsic property of the STM model, which exhibits differences in computational power resulting from different models of basic brain processes like STM.

Degree Self-tuning

Let the activity of the global inhibitor of layer ξ be represented by z, and q represent the number of units in layer ξ. Variable z is defined as

$$z(t) = f[\sum_{i=1}^{q} s_i^{\xi}(t-1) - 2] \quad (10)$$

and therefore the inhibitor will be activated if there is more than one unit firing simultaneously in layer ξ. According to (6), the internal state $s_i^{\xi}(t)$ can be triggered either by system input (called *attention*) through $I_i^{\xi}(t-1)$ or by input signals from layer ζ. The latter is called *anticipation*. What the inhibitor actually does is to detect conflicts among those detectors in layer ξ. Since system attention is always sequential, the inhibitor can only be activated by conflicting attention and anticipation or just by conflicting anticipation of the detector layer.

Degree d_i (i = 1, ..., q) is initially set to 1. Self tuning of d_i is done according to

$$d_i(t) = d_i(t\text{-}1) + 1 \qquad \text{if } s_i^{\xi}(t\text{-}1)=1,\ z(t)=1,\ d_i(t\text{-}1)<T \quad (11)$$

that is, the degree of $<i, \xi>$ increments if this unit together with other units causes activation of the global inhibitor. If the degree of $<i, \xi>$ increments, there will be one more item from the input layer that can be sensed by $<i, \xi>$. Thus the previously learned weight distribution to the unit (see Eq. 8) will have to change its direction of distribution. In the situation, the model re-initiates the weight distribution to $<i, \xi>$ and threshold Γ_i^{ξ} is also modified according to (9) based on the new value of d_i. From (6), (7) and (8), it is clear that if $d_i(t)$ grew larger than T, the STM capacity of layer ζ, it would be equivalent to that $d_i(t) = T$ in the dynamics of the internal state and weight distribution of $<i,\xi>$. That is why $d_i(t)$ has an upper limit of T.

A computer simulation of the model was conducted for reproducing a complex sequence S_C: *J-B-A-C-D-A-B-A-E-F-A-B-A-G-H-A-B-A-H-I*. Learning a complex sequence is slower than learning a simple sequence, because the complex sequence needs dynamically increasing the degrees of certain detectors, and each time such self organization is done earlier training of those detectors has to be discarded. Roughly speaking, time required for training increases linearly with the degree of a sequence. It took 18 training trials before the model learned to reproduce S_C, whereas 6 trials suffice to reproduce a simple sequence. The degree vector acquired by the degree self-tuning mechanism is {1, 2, 3, 1, 1, 2, 3, 4, 1, 1, 2, 3, 4, 1, 2, 2, 3, 4, 2} for those detectors. The ninth component E, for example, requires to memorize the prior subsequence of 4 components *D-A-B-A* in order to be generated; and the second component B, however, only requires to memorize the previous component J in order to be generated.

The above neural algorithm optimally identifies amount of context required to reproduce any complex temporal sequence unambiguously. The same problem of finding minimum amount of context has been studied by Kohonen (1987) for producing unambiguous inference rules in sequence generation. The proposed solution relies on explicit rules for resolving inference conflicts. A basic difference of our proposal from his is that we do not resort to any external rules. Units representing symbols and detectors in our model are connected in a neuron-like manner, and communication among units is typically neural.

Interval Maintenance

In our model, the interval length of a component presentation is the time period during which the external input of the unit corresponding to that component equals 1. This is equivalent to the period when the excitation level of the unit equals T. In sequence reproduction, a unit in layer ξ detects the onset of the context of a component in order to trigger that component in the reproduction process. In S_C above, for example, there is a detector in layer ξ that is trained to detect the context *D-A-B-A* and to anticipate the onset of symbol E. According to the model, after training this detector is activated just one time step after the second A starts to occur (see Eq.6). But E should not be triggered until the whole interval of the A occurrence has elapsed. The idea for interval maintenance is to code intervals by connection weights from the detector layer to the input layer. Since the backward projections from layer ξ to ζ are many-to-one correspondence, the interval of a symbol presentation can be simply coded as the reciprocal of the corresponding connection weight, so that temporal integration of the entire interval is required to trigger the next component.

In general, one interval series of presentation may be different from another one. In order to cope with this situation, instead of storing one interval directly in a weight, two parameters are stored in the connection terminal, one is an average μ of different training intervals and another is a deviation σ^2. During reproduction of a sequence, a Gaussian number is generated based on μ and σ^2, to control a specific interval. Each generated interval will also modify μ and σ^2 like a presentation interval. Therefore, learning is nothing but formation of μ and σ^2. Let e_i represent the interval of the ith presentation of a symbol. Two factors are taken into consideration for forming μ and σ^2. First, each interval should contribute a certain amount. This is called an *averaging factor*. Second, a recent interval should have more impact than a remote one. This is called a *recency factor*. These two factors are embodied in the following learning rules.

$$\begin{cases} \mu_1 = e_1 \\ \mu_{k+1} = (1-\beta)\,\mu_k + \beta\, e_{k+1} \end{cases} \quad (12)$$

where β is the recency parameter ranging between 0 and 1, which describes that except the first interval the most recent interval has a constant amount of contribution, regardless of the presentation history.

The following recurrence learning rule for the deviation can be derived from (12)

$$\begin{cases} \sigma_1^2 = 0 \\ \sigma_k^2 = \dfrac{k(1-\beta)}{k-1}\left[\dfrac{k-2}{k-1}\,\sigma_{k-1}^2 + \beta(e_k - \mu_{k-1})^2\right] \end{cases} \quad (13)$$

, and it is easy to see that $\sigma_k^2 = 0$, if $e_1 = \ldots = e_k$.

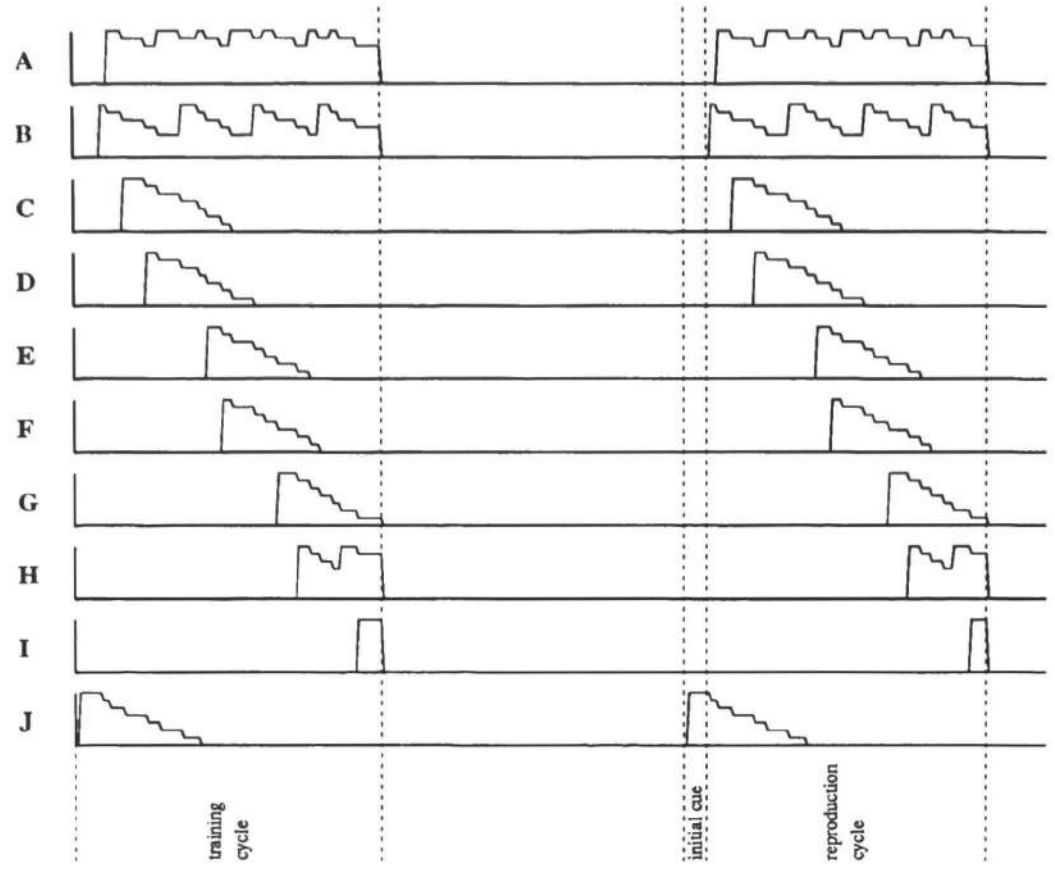

Figure 3. Reproduction of the complex sequence S_C: *J-B-A-C-D-A-B-A-E-F-A-B-A-G-H-A-B-A-H-I*. The interval series {9,3,6,9,5,9,7,3,6,4,9,4,5,8,5,4,5,3,7,8} was first randomly generated, and fixed in subsequent training trials. All units in layer ζ have 3 terminals, and $C_i = 0.3$. The other parameters are $\beta = 0.3$, and $T = 7$.

With the learning rule of (12) and (13), interval maintenance defined above is thus achieved. A computer simulation of the model was conducted to reproduce the complex sequence S_C. As previously stated, the model took 18 training trials to learn the sequence. The number of trials is basically decided by requirement of degree self-tuning. After learning, the entire sequence with various interval lengths was able to be reproduced by the initial context of the sequence, subsequence *J* in this case. Fig.3 presents the simulation result, which contains a temporal course of the last training trial together with the reproduction process. Since in this simulation the speed of presentation is the same from one trial to another, the acquired deviation for every link interval is zero. Therefore the time course of the sequence is faithfully preserved in reproduction.

In summary, this article presents a neural model of temporal sequence reproduction, which is based on an interference model of short-term memory. The model of neural circuit proposed reproduces any complex temporal sequence which may be distorted in time (time-warped). The new abilities demonstrated in this paper, particularly interval maintenance, demonstrate that a dramatic difference in computational power could be lead to by results from basic studies of cognitive science.

References

Dehaene, T.; Changeux, J.P.; and Nadal, J.P. 1987. Neural Networks That Learn Temporal Sequences by Selection. *Proceedings of National Academy of Sciences USA* 84: 2727-2731.

Grossberg, S. 1969. Some Networks That Can Learn, Remember, and Reproduce Any Number of Complicated Space-time Patterns, I. *Journal of Mathematics and Mechanics* 19: 53-91.

Kohonen, T. 1987. Dynamically Expanding Context, with Application to the Correction of Symbol Strings in the Recognition of Continuous Speech. In Proceedings of the International Conference on Neural Networks II, 3-9. San Diego, CA.

Kühn, R., van Hemmen, J.L., and Riedel, U. 1989. Complex Temporal Association in Neural Networks. *Journal of Physics A* 22: 3123-3135.

Miller, G.A. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review* 63: 81-97.

Murdock, B.B., Jr. 1987. Serial-order Effects in a Distributed-memory Model. In Gorfein, D.S. and Hoffman, R.R. eds. *Memory and Learning: The Ebbinghaus Centennial Conference*, 227-310. Hillsdale, NJ: Erlbaum.

Wang, D.L. and Arbib, M.A. 1990. Complex Temporal Sequence Learning Based on Short-term Memory. *Proceedings of the IEEE* 78: 1536-1543.