A Method for Temporal Association in Bayesian Networks

Roland Orre, Anders Lansner

Studies of Artificial Neural Systems
Dept. of Numerical Analysis and Computer Science
Royal Institute of Technology
SANS, NADA, KTH, S-100 44 Stockholm, Sweden

Abstract

A method for sequential pattern recognition and prediction in Bayesian networks is investigated. The basic approach in this method is to add stimulus delay lines to an associative network, thus converting temporal structure to a spatial one. Some methods to avoid very large connection matrices are studied. Results show that it is possible to efficiently store sequences in a network where the connection matrix is strongly reduced.

1 Background

Considering certain aspects of information processing performed by biological neuronal networks such as recognition and motor control, it is quite reasonable to assume that the treatment of temporal patterns is a fundamental property of the nervous system. In the case of auditory recognition the cochlea performs something quite similar to an electro-mechanical discrete Fourier transform of the sound input. The mechanical pressure variations are converted to a time varying pattern of intensities for the decoded frequencies. Certain sounds are only discriminable due to their temporal properties. For instance, when a recorded piano is played backwards it will certainly not be recognized as a piano. The same effect is seen on higher level audio perception, such as listening to melodies or speech understanding. As another example we could consider motoric efference. When generating motor actions such as speech and locomotion, both a precise timing and precise patterns of muscle activation sequences are essential for an accurate result. Yet most of the research in the field of artificial neural networks has dealt with static pattern recognition and classification problems. Of course it is fundamental to the science of neural networks to have a good knowledge and understanding of how to design static classifiers and associative memories. On the other hand, for studies of higher level behavioral aspects of biological systems and for application of neural network solutions to real world problems, it is important to have network models that processes temporal patterns with reasonable performance.

2 Purpose

The goal of the present work was to develop a useful model for temporal pattern recognition and generation using a neural network based on Bayesian learning. The temporal network model will further serve as a part of an ANS (Artificial Neural System). It provides a way to do simulation studies of problems that need to be treated in a temporal manner.

3 Definition of the temporal problem

Is there such a thing as an ideal temporal associative memory? If so, then how could that be defined? A temporal memory could be imagined to remember spatio-temporal patterns. Experience from static associative networks tells us, however, that patterns stored in a network should not be dependent on intermediate values for single neurons. Such a net will be too sensitive to disturbances as noise and faulty neurons. If we want continuous values we use groups of neurons instead. A reasonable limit then would be to treat only sequences of patterns, whose prototypes are binary, and to use graded output from units as belief propagation. In any case, graded values and spatio-temporal patterns can be approximately achieved by different coding techniques such as interval coding and population coding.

One approach to define a sequential memory could be to extend a definition of a static memory. Consider the following definition of an ideal static autoassociative memory for binary patterns (Kohonen 1988):

(i) *An ideal autoassociative memory is a system which holds copies of distinct input signal sets* $x^{(p)}, p = 1, 2, \ldots, k$ *in its internal state, and produces the copy of a particular set* $x^{(r)} = (\xi_1^{(r)}, \xi_2^{(r)}, \ldots, \xi_n^{(r)})$, $r \in 1, 2, \ldots, k$ *to the outputs, whenever (in the recall mode) the inputs are excited by a set of signals* $x = (\xi_1, \xi_2, \ldots, \xi_n)$ *in which a specified subset of the values* ξ_i *matches with the corresponding subset of* $\xi_i^{(r)}$.

This could serve as the basis for a definition of an "ideal" sequential memory. The following definition has been the basis for the present work:

(ii) *An ideal sequential associative memory holds copies of sequences of instantaneous patterns, defined as in (i),*

$$x^{(p,s)}, p = 1, 2, ..., k; s = s_0, s_1, ..., s_n$$

in its internal state, where s *is an implicit state number, and produces a copy*

$$X^{(r,s)} = x^{(r0,s0)}, ..., x^{(ri,si)}, ..., x^{(rj,sj)}, ..., x^{(rm,sm)}$$

of a particular stored sequence of instantaneous patterns, whenever, in recall mode, the net is stimulated with a sequence of instantaneous patterns

$$Y^{(u,s)} = y^{(u_0,s_0)}...y^{(u_i,s_i)}...y_{(u_j,s_j)}$$

where, in a specified subset of the sequence $Y_{(u,s)}$ *each member* $y_{(ui,si)}$ *matches a specified subset of each member* $x_{(ri,si)}$ *according to (i), in the sequence* $X_{(r,s)}$. *An ideal temporal associative memory could then be defined based on (ii) where the relative time between different instantaneous patterns and the absolute recall speed is also considered.*

(iii) *An ideal temporal associative memory holds copies of sequences of instantaneous patterns, where each instantaneous pattern is a set of signals,*

$$x^{(p,t)}, p = 1, 2, ..., k; t = t_0, t_1, ..., t_n$$

in its internal state, where t *is an implicit time stamp, and produces a copy*

$$X^{(r,\tau)} = x^{(r_0,\tau_0)}, ..., x^{(r_i,\tau_i)}, ..., x^{(r_j,\tau_j)}, ..., x^{(r_m,\tau_m)}$$

of a particular stored temporal sequence of instantaneous patterns

$$X^{(r,t)} = x^{(r_0,t_0)}, ..., x^{(r_i,t_i)}, ..., x^{(r_j,t_j)}, ..., x^{(r_m,t_m)}$$

whenever, in recall mode, the net is stimulated with a temporal sequence of instantaneous patterns

$$Y^{(u,t)} = y^{(u_0,\tau_0)}...y^{(u_i,\tau_i)}...y^{(u_j,\tau_j)}$$

where, in a specified subset of the sequence $Y^{(u,t)}$ *each member* $y^{(ui,ti)}$ *matches a specified subset of each member* $x^{(r_i,t_i)}$, *according to (i), in the sequence* $X^{(r,t)}$ *in monotonic order, and*

$$t_0 - t_i = T \cdot (\tau_0 - \tau_i), t_0 - t_j = T \cdot (\tau_0 - \tau_j), ...; T \in R.$$

In most circumstances, however, we are not interested in reversed recalls. Thus we may limit the produced sequence of sets to those where T ¿ 0, i.e. a sequence would only be recalled in the same order as it was stored. As a further restriction, in the current study we have focused on sequential memories according to definition (ii). Thus it is only the order of instantaneous patterns in a sequence that is essential. In the following examples some sequences; S1, S2 and S3; are given. Each instantaneous pattern in these is a character, or, as in examples e2 and e3 a pair of characters. Assuming that we have an ideal associative memory available, a sequential memory may then be implemented under one of two trivial constraints. Constraint 1, each instantaneous pattern is unique. Two sequences, S1 and S2, are stored in the associative memory, such as:

```
(e1) S1 = 142857142857...
     S2 = 093093...
```

Here any of the sequences S1 or S2, both infinitely long, may be uniquely produced if a stimulus pattern y(u,t) matches any element of the pattern sets 1,2,4,5,7,8 or 0,3,9. Constraint 2, we have the possibility to store time or equivalent contextual information together with each instantaneous pattern x(p,t),(Rosenblatt 1962):

```
(es) S1 = (a,0),(b,1),(a,2),(c,3),(a,4),(d,5)
     S2 = (e,0),(c,1),(e,2),(b,3),(e,4),(b,5)
```

Each instantaneous pattern, in this case, is made unique by giving it an explicit time tag, but any pattern may occur just once in a specific context or at a specific time. If the sequences S1 and S2 in (e2) above are learned and a third sequence, S3, such as:

```
(e3) S3 = (c,0),(d,1),(a,2),(d,3),(c,4),(a,5)
```

is added, the sequences S1 and S3 could not be unambiguously recalled because the instantaneous pattern (a,2) is no longer unique among the sequences S1, S2 and S3 because it occurs both in S1 and S3. Another drawback of making each instantaneous pattern unique is that it becomes hard for the network to make generalizations. Let, for example, a temporal network learn the following sequences S1 and S2:

```
(e4) S1 = ababcde
     S2 = bcabcab
```

When storing such sequences as S1 and S2, in which an instantaneous pattern "a" is always followed by a "b", it is possible to generalize about this. A constraint that makes each instantaneous pattern to be unique, i.e. where a network may only use pattern information from the previous timestep, will be hard to fulfill. The basic problem in temporal networks is how to deal with the history. A prediction must normally use data older than t minus 1. The property of a set of sequences telling how old the data must be for one to deal with it, in order to specify each continuation uniquely, is here called context length.
Definition:

> *The context length of a sequence is the maximum age of the data required to specify each continuation uniquely.*

For example: The sequence "142857142857..." (fraction part of 1/7) has context length 1 but the word "mathematics" has context length 4. Now, consider the differences between the properties needed for sequential pattern classifiers and sequential pattern generators. For classification purposes, e.g. phoneme recognition, the problem consists mainly of detection of specific features in the input stream, features that are often invariant to some properties of the input signal. One property to pay attention to in recognition circumstances is whether automatic segmentation of the input stream has to be managed. Such as:

```
"CANYOUREADTHIS" vs "CAN YOU READ THIS"
"SEGMENTSMAYBEAPROBLEM" vs "SEGMENTS MAY BE A PROBLEM"
```

This is a well known problem in, for example, recognition of continuous speech. There are certain languages, like Finnish, where it may be less accentuated due to consequent stress of the first syllable of each word. The stress of a word is, however, not yet a property to be treated in an efficient manner in automatic speech recognition. (Elenius K., personal communication). For pattern generation purposes we may pay attention to how the sequence is activated and how tolerant the generated sequence has to be against errors in the triggering stimulus. For low level pattern generators in biological motor systems, it has been shown (Grillner et al 1987) that basic spinal pattern generators may be driven by a tonic stimulus. This has also been shown in simulated models of spinal pattern generators (Lansner et al 1989). Figure 1 shows a model of the Lamprey swim generator that has been simulated.

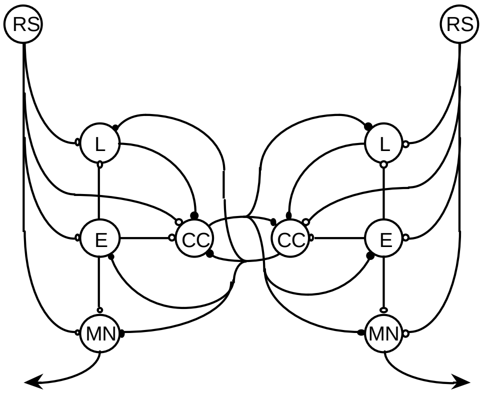


Figure 1: The spinal swimming rhythm generating network of a Lamprey. "E" are excitatory interneurons that drive the motorneurons. "CC" are interneurons that inhibit the opposite side. "L" are lateral interneurons that terminate activity on the active side. "MN" are motorneurons and "RS" are reticulospinal neurons. The reticulospinal neurons are driving the spinal network. Filled circles designate inhibitory synapses and unfilled circles excitatory synapses.

For temporal motor control mechanisms we do not yet know what kind of activating patterns are used. It is however reasonable to believe that the activating pattern sequences are quite short. In fact, the start condition could be just a single pattern, like a goal coordinate in a reaching movement. At a certain level in the system this may reflect how the movement is initialized. In the following we focus on the problem of generating long sequences from short activating sequences.

4 Different methods for temporal sequential networks

We introduce by giving a short review of some previously studied methods for sequential pattern recognition and completion. This is in no way a complete review. It is an illustration of some methods which differ from the approach in the present work. A state machine built round an associative memory is studied by Kohonen. Outputs from the associative memory are fed back through delay lines to the network. Delayed replicas of the outputs will be associated with incoming stimuli (Kohonen 1988). Methods and theories from adaptive signal processing that are used for prediction of stochastic processes are also applicable to neural network models. The weights may be calculated using the least-mean-square (LMS) algorithm or, e.g., Kalman-filter algorithms (Trvn 1988). In another model, that is called short term memory (STM), each unit remembers a small history of its input signal by letting the signal pass a convolute giving an exponential decay (Trvn 1988). Each signal passes a STM-loop. There are several STM-loops in parallel and the outputs from each STM-loop are weighted into a decision net that selects output patterns that are most close to the valid patterns. The Hidden Markov Model is a kind of state machine where the weights are interpreted as transition probabilities. A HMM may be trained using the forward-backward algorithm (Waibel et al 1987). A Jordan network is a way of implementing sequential association in back propagation networks by adding feedback and a set of recurrent state units. (Jordan 1986), (Massone, Bizzi 1989). Another way to use back-propagation for temporal processing is to use "Time Delayed Neural Networks (TDNN's)". TDNN is often used to designate multilayer backpropagation networks where each unit has multiple weights with different delays. These units may however be used with other learning rules (Waibel et al 1987), (Lang, Hinton 1988). The sequential or temporal networks mentioned above may be used either for pattern recognition or pattern generation. Most of them have in common that they may be seen as implementing a type of predictor

$$\hat{p}(t) = F(s(t), \hat{p}(t-1), \ldots, \hat{p}(t-k)) \tag{1}$$

where the estimated pattern at time t is solely a function of the stimulus at time t and k steps of predicted pattern history. This is, of course, not true for models where units with different input delays, as the TDNN units, are used to sample the stimuli. The other extreme in that case is

$$\hat{p}(t-c) = F(s(t), s(t-1), \ldots, s(t-k)) \tag{2}$$

where the estimated pattern at time t-c is solely a function of the input stimuli.The constant c is normally chosen between 0 and k. For classification purposes the estimated pattern may be a decision, like hyphenate vs not hyphenate. A variant of the latter is

$$\hat{p}(t, t-1, \ldots, t-k) = F(s(t), s(t-1), \ldots, s(t-k)) \tag{3}$$

where a sequence may be recognized as a whole. A network model for recognition of temporal patterns that corresponds rather well to this predictor principle has been studied (Tank, Hopfield 1986). Stimuli are projected on a network through continuous delay functions that also make a compression of information in time .

5 Methods and simulation results

In this section we present the models developed and some simulation results.

5.1 Networks with delayed inputs

The basic method for implementation of sequential memories studied in this work is the addition of delayed stimulus connections to an autoassociative network, i.e.

temporal structure is transformed to a spatial one (figure 2). With this architecture we will get the predictor

$$\hat{p}(t,t-1,\ldots,t-k) = F(s(t),s(t-1),\ldots,s(t-k),\hat{p}(t),\hat{p}(t-1),\ldots,\hat{p}(t-k)) \quad (4)$$

where the sequence within the whole context length k is predicted from both k steps of predicted pattern history and from k+1 steps of stimulus. Assumptions: the rate of stimulus change is slow compared with the relaxation time of the network. The delay lines has equal delay characteristics.

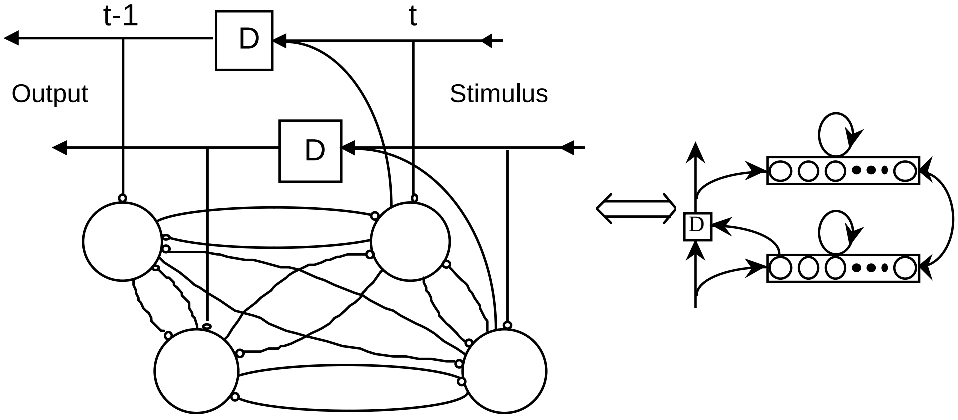


Figure 2: Picture illustrating the principle for temporal to spatial conversion. The left figure shows a fully connected associative net, here represented by 4 neuronal units. A part of the net will see a delayed replica of the input signal (t-1). Outputs from some of the units will be mixed with the input signal. The right figure shows the same in a more formalized way. A network population is represented here by a rectangular box. An oval with an arrow shows that the population is recurrent. An arc binding two network populations together means that all units in one population are projected on all units in the other population in the direction of the arrows.

The associative network model chosen is of Bayesian type (Lansner,Ekeberg 1989). There are several reasons for choosing a Bayesian network. The learning rule is fairly simple and biologically reasonable. The Bayesian criterion is also considered to be the best in comparison with other common classifiers as Perceptron (Linear), Least Mean Square and Sigmoid (Barnard, Casasent 1989). The learning problem in these types of Bayesian networks is mainly a question of collecting statistics. The weights are computed from mutually conditional probabilities (assuming independent patterns), such that:

$$W_{ij} = \ln \frac{P(j|i)}{P(j)} = \ln \frac{P(j\&i)}{P(i)P(j)} \quad (5)$$

The method used in this work for collecting relevant statistics is an incremental learning rule (Ekeberg ., personal communication). The probabilities are estimated without prior knowledge of the patterns by using exponential convolutes and thus will be good estimates for both stationary and non-stationary processes. In the notation used here Sj(n) is sample value for unit j when the n'th pattern is presented. Pij(n) is the compound probability that unit i and j are simultaneously active. t is a time constant that is chosen large enough to smooth out short term variations but short enough to follow nonstationary processes.

$$\tilde{P}_j^{(n+1)} = \tilde{P}_j^{(n)} + \frac{S_j^{(n)} - \tilde{P}_j^{(n)}}{\tau} \qquad \tilde{P}_{ij}^{(n+1)} = \tilde{P}_{ij}^{(n)} + \frac{S_i^{(n)} \cdot S_j^{(n)} - \tilde{P}_{ij}^{(n)}}{\tau} \quad (6)$$

A problem with these fast and simple learning rules in one layer nets is that probabilities are assumed to be pairwise independent. This means that certain patterns are not distinguishable. It is possible to deal with dependent patterns by using multilevel nets. E.g. a one level Bayesian network could be enhanced by complex nodes (Lansner, Ekeberg 1987), i.e. "interneurons", working as feature detectors, that are using a different and more complicated learning rule. In this work there has been no attention paid to this possibility. Here, only one layer nets with incremental Bayesian learning are considered. We assume that the problem of independence has been solved before the input is given to the Temporal Associative Network (TAN). In a network where just one delay step is used,as shown in figure 2, it is possible to store sequences with a context length of one. To recall sequences in this network, output from the part of the network that is stimulated without delay is fed back and mixed with the delayed stimulus. To manage longer context lengths the most obvious thing would be to add more stimulus delay lines thus spreading the temporal information over a larger network, figure 3. The outputs from the subnets is coupled to the delay lines and from there propagated. This is refered to here as output feedback. The degree of output feedback is not critical but some tests showed rather good results when the outputs and the inputs influenced the propagated data with one half each. If the output feedback is too large the network will have hard to change a faulty decision. In the corresponding way the net will be sensitive to noise when the influence of the inputs is too high.

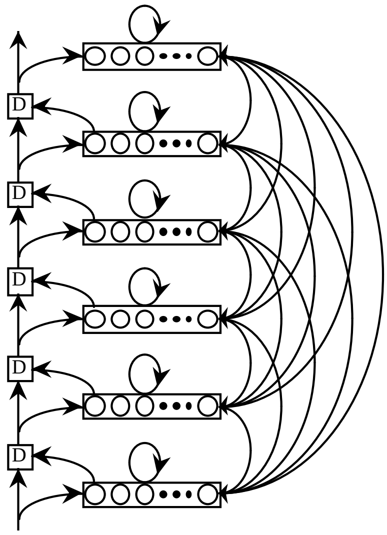


Figure 3: A network with 5 delay steps that could manage context lengths of 5. The stimuli is propagated in the direction of the vertical arrows. After the network has reached a state of relaxation it produces an output from each neuronal population (the rectangles). Outputs from the neuronal populations are fed back to the stimulus propagation line as the arrows going from top of a rectangle and leftwards show. These connections are here called output feedback .

The storage capacity (Zmax) in an associative network relates to the number of units (N), where "ln" is the natural logarithm, as (Lansner, Ekeberg 1985):

$$Z_{max} = \mathcal{O}\left((\frac{N}{\ln(N)})^2 \right) \tag{7}$$

When storing sequences of patterns in a network, configured as in figure 2, that could manage a context length of one, it would be expected that the maximum number of sequences of length ”l” possible to store would be

$$Seq_{max} = \mathcal{O}\left(\frac{(\frac{N}{\ln(N)})^2}{l-1}\right) \tag{8}$$

The assumption behind this is that each instantaneous pattern in the sequence is coded as a single unit. If this is the case, the first and last step of each sequence will not generate any weight change, i.e. each pattern in a sequence is associated with its follower, except, of course, for the last one. When we add more delay lines to manage contexts of length ”c”, as in figure 2, we would, with the same assumption as in the previous example, expect the maximum capacity to be decreased to

$$Seq_{max} = \mathcal{O}\left(\frac{(\frac{N}{\ln(N)})^2}{l+c-2}\right) \tag{9}$$

When we need to treat long context lengths, with this model, very large amounts of weights are required. The number of weights will increase with the square of the context length ”c” ,where ”n” is number of units in a single pattern:

$$W_{tot} = \left((c+1)\cdot n^2\right) - (c+1)\cdot n\Big) \tag{10}$$

6 Matrix reduction due to time invariant relations

When looking at the connection matrix (figure 4) one could expect some symmetries to be found due to dependencies between patterns at different timesteps. A matrix element like [t-3,t-2] that connects output from timestep (t-3) with inputs at (t-2) should be the same as the element [t-4,t-3] etc.

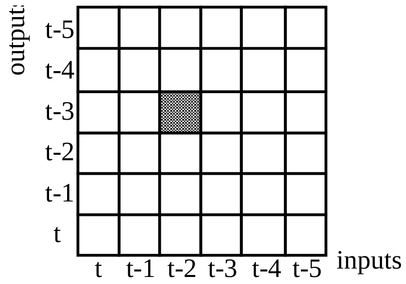


Figure 4: The connection matrix that connects different timesteps with each other. Observe that the elements of this matrix are also matrices which make the network at each timestep recurrent. The interpretation is that the outputs from timestep t-3 connects to inputs at timestep t-2 and so on.

If the symmetry principle is correct then we would get a matrix like the one in figure 5. The connection matrix will have a diagonal structure where each element, representing the set of weights projecting one population on another, is constant along a diagonal. Due to this symmetric structure the number of unique weights is reduced to increase linearly with the context length instead of growing with the square.

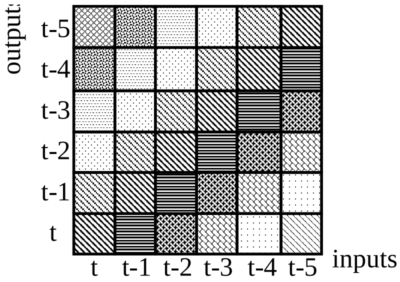


Figure 5: Due to invariance between dependencies at different timesteps the matrix will show a diagonal structure.

By utilizing the diagonal structure of the connection matrix it would be possible, in a simulated network model, to use a smart lookup of weight values. If this is possible to realize in a simulated model it is still, however, not very attractive because it is still computationally expensive and totally unplausible from a biological point of view. It would also be rather unpractical to implement this model in hardware. Perhaps it could be a creative approach to reason in the following way. The multiple sets of equal weights along the diagonals are redundant in the sense that, once a set of weights has been used in the relaxation it is possible to ignore that set at further timesteps. As a consequence, we could then try to simply remove the redundant part of the matrix as shown by figure 6. This operation would also make the number of weights linearly proportional to the context length, such that:

$$W'_{tot} = (n^2 - n) \cdot (2 \cdot c + 1) \tag{11}$$

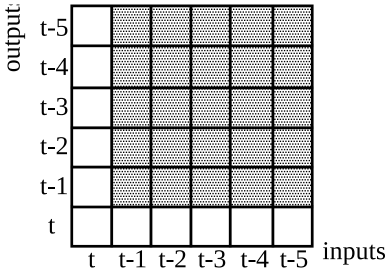


Figure 6: Assuming that the multiple occurrences of equal weight sets over the diagonals are redundant, we could simply remove them (grey). Thus the total matrix will be L-shaped and the amount of weights will grow linearly with the context length.

A network with connectivity reduced according to the hypothesis that multiple sets of time invariant weights are redundant is shown in figure 7.

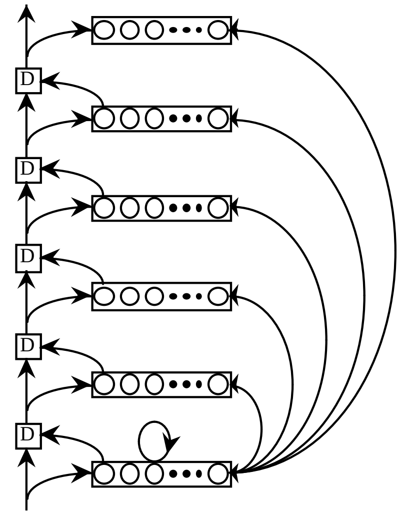


Figure 7: A TAN (Temporal Associative Network) with 5 delay steps and "redundant" connectivity removed.

To characterize the performance of the present model and to investigate the effect of removing the "redundant" part of the matrix some tests have been performed. figure 8 shows some results from tests that have been run using random sequences of characters as input. Each character has been coded as a unique active unit. By coding each character uniquely we are sure that the characters themselves are pairwise independent. Tests where distributed character coding is used have also been performed with reasonable results on the larger nets but they are not shown here. The same set of training and test data has been used for all the following tests.

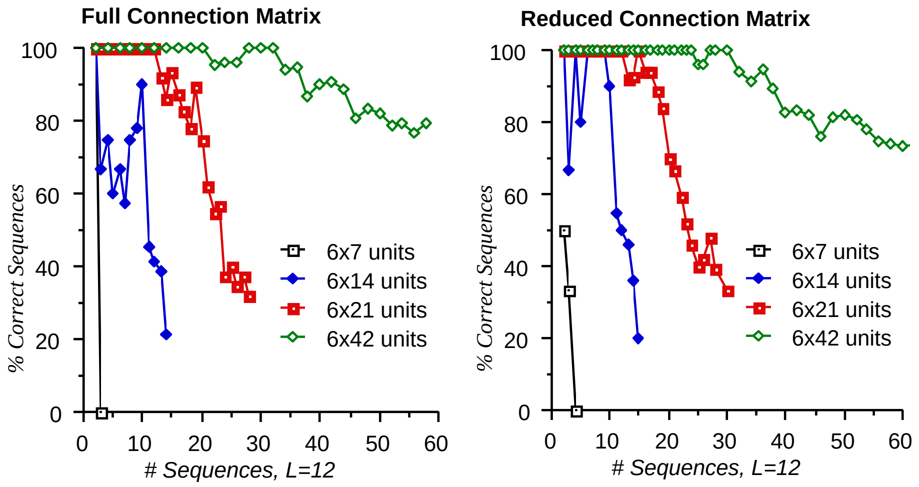


Figure 8: Comparison of behavior between a TAN with full connection matrix (left) and a TAN with reduced connection matrix (right). The same random character sequences has been used in both the tests. The vertical axis shows the percentage of sequences that has been 100 been stimulated with the first three steps of the sequences learned.

When completion tests are run on the net with full and reduced connection matrix we see, as figure 8 shows, that the nets behave almost equivalently. As a matter of fact, the nets with reduced matrices perform slightly better except for the small net with 42 units. The results above indicate that the principle for temporal to spatial conversion, that is used here, could be a useful method for implementation of sequential associative memories.

7 Left-Right context

One reason for using a network that is recurrent in the time domain, as we did in the previous section, is that this makes it possible to do pattern completions within the whole manageable context length for the net, i.e. we not only may predict the future from old data, but also tell what the past should have been, based on recent data. Earlier we defined the context length for a sequence as the maximum number of timesteps required to specify each continuation uniquely. The context in his definition could be designated ”left-context”. In the same way, ”right-context” could be defined as the number of steps required to specify a unique history. It is, of course, totally irrelevant to speak about left and right in the time domain but, if we associate time with, for instance, sequential reading of text, which in most languages is performed from left to right, we get a useful interpretation. A more general expression would be preand postcontext. Seen from a statistical point of view pre- and post-context may be compared with pre- and post-dictors that are used to determine the correct value of a signal based upon passed history and future (Parsons 1987). Pre- and post-contexts have the same size for any sequence but this does not imply that this amount of timesteps is necessary for determination of the sequence uniquely. The context length defines the minimal number of delay steps we need to generate the sequence. There may be many parts of a sequence shorter than the context length that uniquely specifies it. Assume that a network similar to the one in figure 7, but with 10 delay steps, has learnt the following sequences whose context length is 4 :

```
S1 = MATHEMATICS
S2 = MATERIALLY
S3 = MATRICULATE
```

If the net is stimulated with e.g. ”MAT” it will not be able to unambiguously choose a continuation. The net will choose one of the sequences anyway. Consider the following stimulus-recall process (”.” means empty input) :

```
Stim = MAT.             Recall = MATH
Stim = MAT..            Recall = MATHE
Stim = MAT...           Recall = MATHEM
Stim = MAT...L          Recall = MATERIAL
Stim = MAT...L.         Recall = MATERIALL
Stim = MAT...L.T        Recall = MATRICULAT
Stim = MAT...L.T.       Recall = MATRICULATE
```

By stimulating the same net with "........ICS" it will recall "MATHEMATICS", i.e., by utilizing the right-context (or post-context), a temporal network will be able to recall a whole history or ”cause” when it is stimulated with a part of a sequence or change its decision when additional input is available. For a sequence generating network this is a valuable property to prevent noise in input to give errors in output. Experiments like these has been performed on human beings for the English language (Shannon 1951). It was found that the predictor and the postdictor have the same characteristics but the predictor is slightly better. It means that it is somewhat easier to guess a word when given the beginning of it than given the end. Actually, a network with temporal recurrency, like the one here described, could equally well be used to recall a sequence backwards if the delay lines were reversible. Biological evidence for reversible temporal networks are currently unknown, but for certain problem domains this could be a useful property. Take for instance a labyrinth learning network that has learnt a sequence of turns to find its way across the labyrinth. By reversing the direction of recall and reversing the direction of turn the network would find the reverse way across.

8 Feedforward classification of old data

To predict time sequences is somewhat like walking in a rather well-known landscape. Often we may know where we are and where to go by just looking in the closest environment. But, sometimes the closest environment is not enough and we have to look around for distant landmarks. When we look in the close environment we sometimes take a faulty decision, but soon we may recognize a distant environmental feature that makes us change our mind and choose a new direction. This is easy, but if we have misinterpreted a distant landmark we may go several kilometers in the wrong direction. In a sequential associative network it could, referring to the analogy above, be reasonable to assume that we may use the feedforward principle for old data, i.e. "distant landmarks" and the recurrent principle for recent data, i.e. "close environment". The network will then be robust in the recurrent part where data are highly correlated and an error in stimuli may cause great errors in the decided output. When the stimulioutputs mix is propagated to the feedforward part of the network the output decided is no longer possible to change. To investigate if the feedforward principle is feasible with the Bayesian learning rule for predictions of sequences some tests has been done on feedforward connected networks (figure 9, Left).

As can be seen from tests on a feedforward network (figure 9, Right) the capacity is about half of the capacity for a recurrent network (figure 8), with the same set of data. This is expected since there is half the number of connections and the stimulus is not noisy. A network of this type has to make correct decisions at each timestep, otherwise the faulty patterns will be propagated and be the basis for new decisions. An attempt to decrease the number of weights without loosing too much capacity or the left-right context principle is to use recurrency for recent data and feedforward connectivity for old data as figure 10 shows. The capacity is slightly less than for the fully recurrent net. The reason for the capacity to be that high here may be explained when one considers how autocorrelation functions usually behave as a function of time difference. Since most sequences probably have a rather short context length, the feedforward weights are only needed to resolve just a few ambiguities.

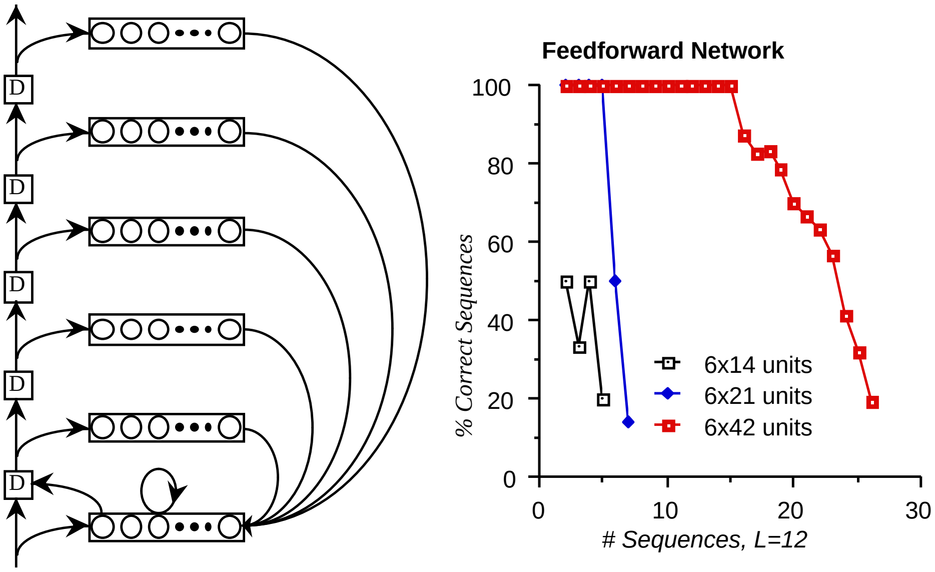


Figure 9: (Left). A temporal network with feedforward temporal connections. (Right). Results from completion tests on a feedforward temporal network.

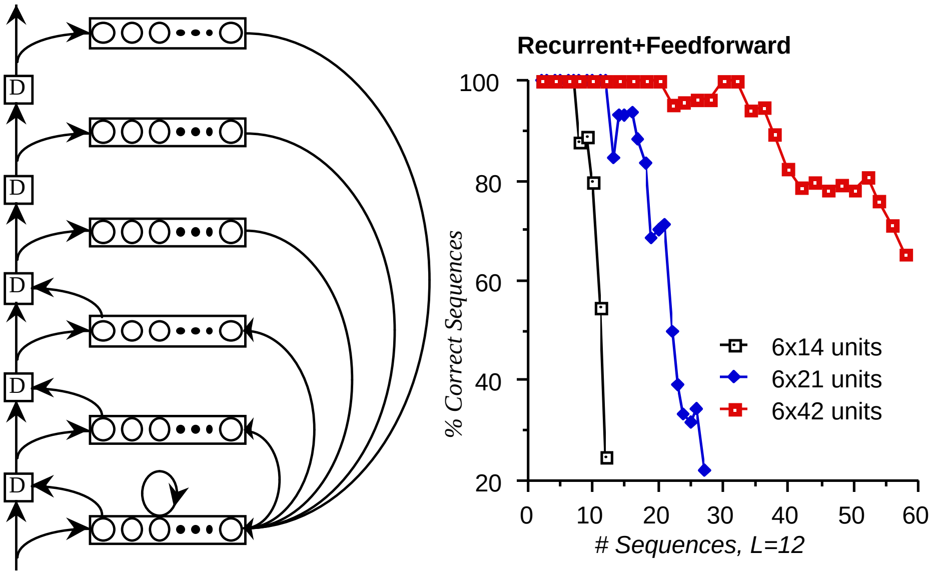


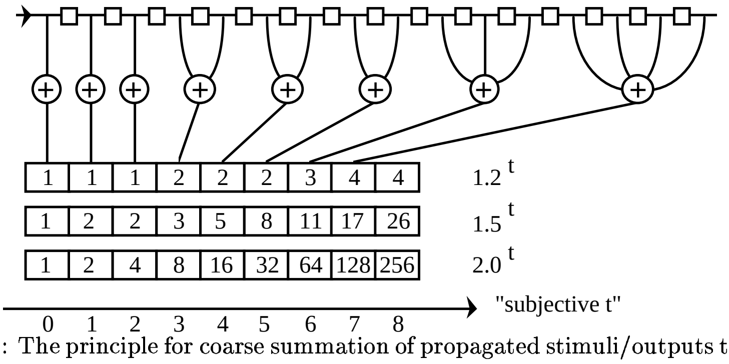
Figure 10: (Left):A temporal network where recent timesteps are recurrent and older timesteps are feedforward only. (Right): A few results obtained with the combined network. These results are only slightly worse than for the fully recurrent network.

9 Coarse summation of old data

If we consider the consequences of being able to manage very long sequences in temporal neural networks we will probably come to the conclusion that it is rather unrealistic to have a temporal resolution that is linear with time. We would get very large connection matrices even when using the symmetry and feedforward principles investigated above. Many of the weights would be unused due to small correlations between data with large time differences. As an example we can look at the following expression ”the correlation is low”. If we consider the temporal connections between ”the” and ”low” it should be quite clear that the exact time when ”the” occurred compared with ”low” is less important compared with ”t” and ”e” in ”the”. When looking at perception in biological systems we know that the threshold for a difference in sensation relates to the change in stimulus as (Weber’s law) (Kandel 1985): k=DS/S, i.e. the threshold for experience of a change in stimulus is proportional to the size of the stimulus. In this way our sensations are proportional to the logarithm of the magnitude of the absolute stimulus. A hypothesis about temporal resolution could then be stated in the same way

$$\Delta Resolution = k \cdot \frac{\Delta T}{T} \tag{12}$$

which would give a resolution that is a logarithmic function of time distance. One way of implementing a logarithmic-like resolution in a network like the one we havedescribed here, would be to sum the history over a number of time steps that grows exponentially with the ”subjective” time distance. Figure 11 shows some examples of this summation of history that we will now refer to as coarse summation.



1	1	1	2	2	2	3	4	4	1.2^t
1	2	2	3	5	8	11	17	26	1.5^t
1	2	4	8	16	32	64	128	256	2.0^t
0	1	2	3	4	5	6	7	8	"subjective t"

Figure 11: The principle for coarse summation of propagated stimuli/outputs to obtain a logarithmic-like time resolution. The number of sum steps is an exponential function of the ”subjective” time, here with bases 1.2, 1.5 and 2.0.

In an implementation of a temporal network that uses coarse summation it is necessary to choose the base for the exponential function in accordance with the statistical properties of the data. To see if the hypothesis about logarithmic resolution could be reasonable anyway, some tests were run on a network configured as in figure 12, with 2 as a base for the exponential summation.

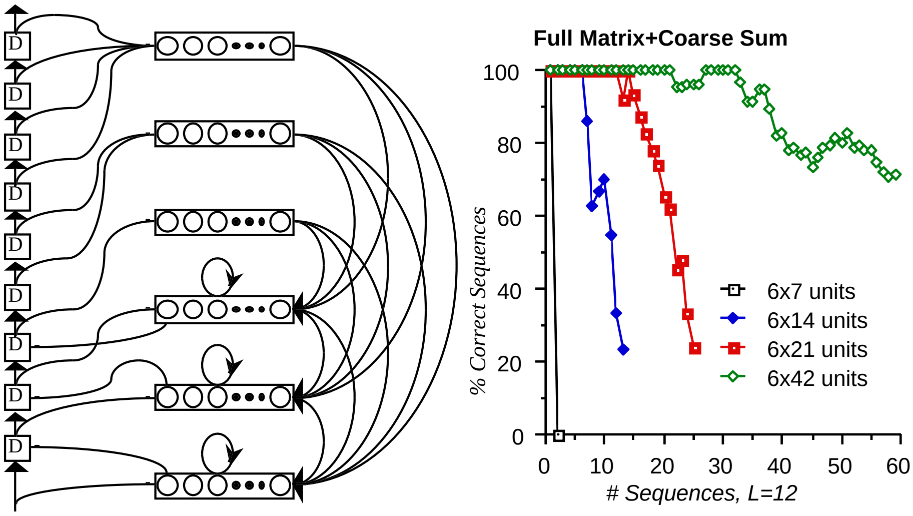


Figure 12: (Left) A temporal network that achieves a logarithmic-like resolution versus age of data. The varying resolution is achieved by coarse summation of old data. The coarse summed data is feed forward weighted into a fully recurrent net that spans over a few timesteps. (Right) Results from some tests with random sequences on some nets with coarse summation.

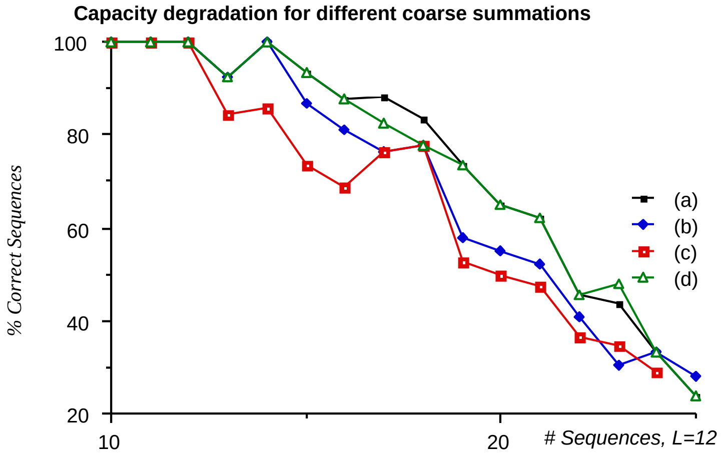


Figure 13: This diagram shows how the capacity degrades when different type of coarse summation is used. The capacity for perfect recall is, in this test, the same for all, but they show different degradation when this maximum capacity is exceeded. (a): Unit on if any input is on, i.e. like a logical OR-function. (b): Average, i.e. the sum has been divided with the number of coarse summed units. (c): A unit is on if most of the units are on, i.e. an average with a threshold. (d): The summation unit's output value is just the sum of its inputs.

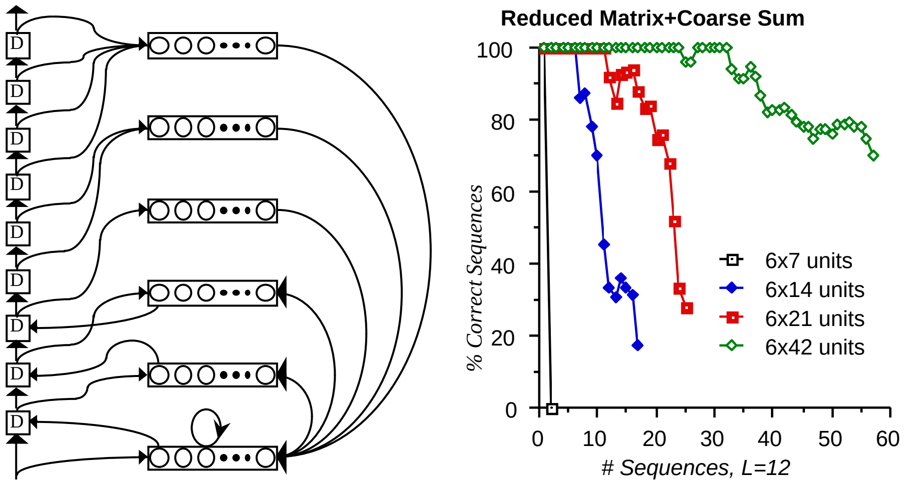


Figure 14: (Left) A temporal network with logarithmic-like time resolution where the coarse summed stimuli is only feedforward weighted into the recurrent part of a network with reduced connectivity. (Right) Results from tests with coarse summation and reduced connectivity.

A question that arises is how the stimuli has to be summed to give the best performance. If we assume that the coarse summed network works according to the same principles as the feedforward network described above, then the best summation principle would probably be one were the new weight values match the values in the feedforward network as much as possible. To achieve this, the output from a coarse summation unit could be just a simple sum of its inputs. A strange thing with this method is that the outputs of the summation units would exceed one. The output from a unit in the Bayesian network model that is used here reflects the probability for this unit to be active. We may however, see these summation units as help units that makes us collects statistics for several units at the same time. Results from a test run with some different types of coarse summation is shown in figure 13. This test indicates that the best type of summation to use is either a simple OR-function or a simple sum.

Completion tests on networks with coarse summation of old data showed quite good results when the sums where coupled to a fully connected network, figure 12 (Right). Corresponding tests on network where the recurrent part had a reduced matrix show, as can be seen in figure 14, even a slightly better result.

10 Sequence capacity for different architectures

The goal with this work was to find a useful model for temporal association in Bayesian networks to be used in simulation studies of temporal phenomena. Thus it is important to consider the model characteristics such as, capacity versus the number of units and number of weights.

To make possible such a comparison all tests have been run with the same random training and test sequences of characters with unique unit coding of characters. Each sequence in the test had the length 12. The number of neuronal populations were 6, i.e. the manageable context length would be 5, except for the coarse summed network that could possibly manage a context length of 9.

As the basic principle for temporal association investigated here is to use delayed inputs for conversion of temporal information to spatial, there may be different performances expected depending on how the spatial matrix is connected due to, for instance, time invariance effects.

The number of units in these models have been kept the same. A "unit" here designates everything which has its outputs connected to other units via weights. Some of these units are, for the feed forward nets, just used as input units, or for the coarse summed networks as stimulus summation units.To vary the number of units in a network model, the width of each pattern and the repertoire of the random sequences were varied.

Figure 15 shows the capacity plotted as number of sequences versus number of units for the different models investigated in this work. It should be observed that the theoretical capacity is based on experiences from static content addressable memories (CAM). It may be a surprise that the theoretical capacity is lower than most of the capacities measured. There is, however, nothing strange in this. What really matters is that we have the same proportionality. The theoretical capacity is based on empirical results for independent patterns with a sparse activity rate of about 1temporal examples we have both dependencies between the patterns and a varying rate of activity.

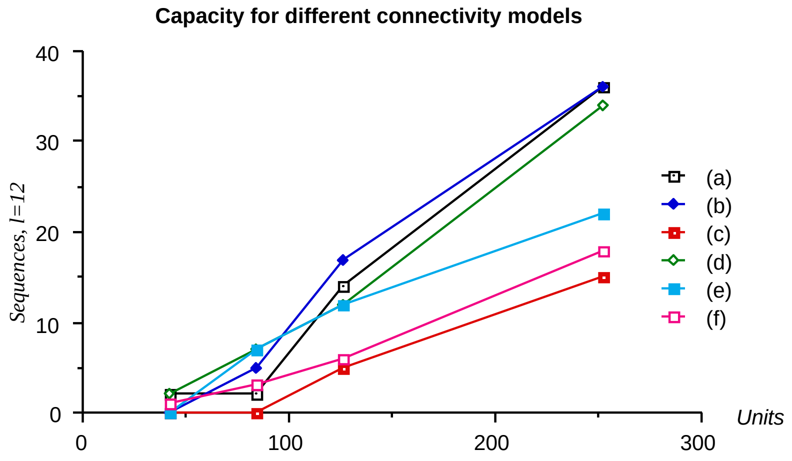


Figure 15: Capacity versus number of units for some temporal network models with different connectivity principles. (a): Fully connected recurrent network. (b): Recurrent network with reduced connectivity due to symmetry. (c): Network with feedforward temporal information only. (d): Mixed recurrent , fully connected with coarse feedforward summation (e): Mixed recurrent, reduced connectivity, with coarse feedforward summation (f): Theoretical capacity. based on static CAM results.

11 Sequence capacity versus number of weights

In the previous plot in figure 15 we looked at capacity versus number of units in each of the tested network models, but, what really makes the cost, at least, in artificial neural networks, is the number of weights. To check which model uses its weights in the most efficient manner the same data as above is used, but instead the number of recalled information bits versus the number of weights for the different models is investigated. The information content of a recalled sequence is designated as the ratio between the number of possible full length sequences and the number of possible sequences that are used as input for recall. The number of information bits in a recalled sequence is just the two-logarithm of this ratio. We get the total number of information bits (I) as the difference between the two-logarithms for the number of possible sequences and the number of possible input sequences (R) times the number of stored sequences (S).

$$I = S \cdot (\log_2(S_{possible}) - \log_2(R_{possible})) \quad (13)$$

This is based upon corresponding calculations for static CAM capacity (Lansner, Ekeberg 1985). Figure 16 shows how the recall capacity per weight differs for the different architectures. As these test indicate the weights are much better utilized when connectivity is reduced than with fully connected networks. The criterion for correct completion in this figure is that at least 95 tests above show a dip in the capacity to e.g. 90 100 figure 16 were plotted, the first dip in capacity was construed as the capacity limit.

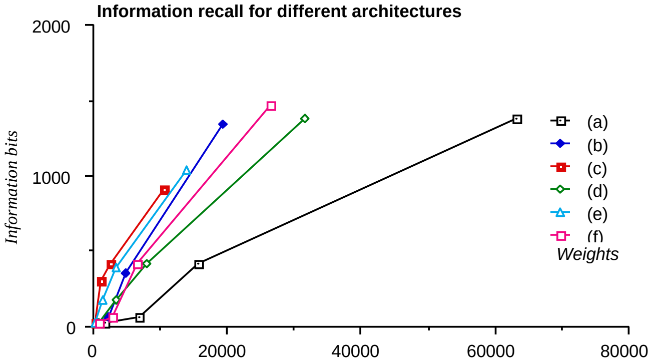


Figure 16: Capacity as recalled information per weights for some temporal network models with different connectivity principles. (a): Fully connected recurrent network. (b): Recurrent network with reduced connectivity due to symmetry. (c): Network with feedforward temporal information only. (d): Mixed recurrent, fully connected with coarse feedforward summation (e): Mixed recurrent, reduced connectivity, with coarse feedforward summation. (f): Feedforward only, as in (c), but with 10 delay steps.

12 Discussion

We may ask: Is there any special reason to choose a temporal model that spreads the temporal information across a spatial network? Yes, this method not only gives the system the possibility to draw conclusions out of the input data from the history but also to "change its mind" when new inputs show that an earlier decision was wrong. Thus, the network will be robust against noise in the input and generate correct sequences also when faulty decisions sometimes are taken. Using the feedforward principle for old data means that the network will be unable to do this.

There are some aspects of temporal association that have not been dealt with in the present work: speed of recall, recall in reverse order, time independent sequential, logical reasoning and semi-sequential processing that may run independently in parallel except for certain rendezvous. It has also been assumed that the sequences do not overlap to much for a one-layer network.

Speed of recall and recall in reverse order are both represented by the constant T given in the paragraph "Definition of the temporal problem" above. When T is less than one, the recall speed is greater than the learning speed and when T is greater than one, the recall speed is lower. Further, when T is negative the order of recall is reversed. Considering the varying speed of recall, it seems to have relevance in biological systems. When we have learned for example a sequence of movements, the speed may be varied within certain limits. A reversed order of recall may not be equally relevant in biological systems. It is common that people have problems when trying to do certain tasks in reverse order. Try for instance to rattle off the alphabet backwards!

When looking at motor systems it is rather unrealistic to imagine that the performance of an action could be reversed by just reversing the order of muscle stimulation. To reverse a movement a completely different strategy has to be taken with different muscles involved etc. This does not imply that this should be the fact in all levels of the system. It is possible that an action stored at a high level in the system, where just model coordinates about the outer world is treated, may be recalled in a reversed order.

The actual movements are then planned from these coordinates and different learned strategies would then be used in different directions. When one consider coarse summation as a method to manage long context lengths there are a few questions that may arise. (a) Is this a relevant method? (b) Is this in some sense biologically relevant?

(a) Further tests of this method on sequences with long contexts has to be done in combination with a more thorough statistical analysis. If looking at implementation of such a method it may also seem impractical to propagate and sum stimuli/outputs in the way we do here. Some function that approximates this behavior is probably to be preferred.

(b) There is currently no indication that coarse summation, as it is described here, may take place in biological systems. If such a mechanism exists it is more probable that it is based upon, for instance, concentration changes etc. in biochemical reaction systems.

All the tests in this work have been run on random sequences of characters. Further investigations have to be done on structured sequences of data, i.e. sequences where it is possible to define a grammar; as well as on sequences with distributed activity in the instantaneous patterns. Investigations have to be done with noisy start sequences and the capacity convergence for large networks has to be checked as well.

13 Conclusion

The studies done in this work indicate that efficient sequential associative memories may be built using one layer Bayesian networks, where temporal information is transformed to spatial information using stimulus delay lines. The reason for this principle to be affordable is that, due to time invariant relations between patterns, the weight matrix has a structure with multiple symmetries where "redundant" connections may be removed. The number of connections will then grow linearly with the maximum context length managed by the network.

By using recurrency in the time domain, a temporal network will be able to tell the next continuation of a sequence and at the same time change its previously taken decision when new contradicting stimuli arrives. The results indicates that the best information capacity per weight is achieved when reduced connectivity is used. It can be used either for a network that is fully recurrent in the whole context length or in combination with feedforward and coarse summation to manage longer context lengths.

As a result of this work a Temporal Associative Network (TAN) software package is available. The package is written in ANSI-C and is instantiable with the following parameters: numbers of units in a pattern, number of neuronal populations, number of fully connected steps, number of steps with reduced connectivity, number of steps with feedforward connections, number of feedback steps, number of normal input steps and number of coarse summed stimuli steps.

14 Acknowledgements

Many thanks to Mikael Djurfeldt and Tom Wadden for critical reading and valuable comments on the manuscript. Funding provided by the National Swedish Board for Technical Development (STU), under grant 89-01826P is hereby gratefully acknowledged. Support from Digital Equipment Corporation's European External Research Program (EERP) is also gratefully acknowledged.

References

[Barnard and Casasent, 1989] Barnard E. and Casasent D. (1989). A comparison between criterion functions for linear classifiers, with an application to neural nets. *IEEE Trans on Systems, Man and Cybernetics* **19**: 1030–1041.

[Grillner *et al.*, 1987] Grillner S., Wallén P., Dale N., Brodin L., Buchanan J., and Hill R. (1987). Transmitters, membrane properties and network circuity in the control of locomotion in lamprey. *Trends in Neuroscience* **10**: 34–41.

[Hopfield and Tank, 1987] Hopfield J. and Tank D. W. (1987). Neural computation by concentrating information in time. *Proc. Natl. Acad. Sci. USA,* **84**: 1869–1900.

[Jordan, 1986] Jordan M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 531–546. Lawrence Erlbaum, Hillsdale, Amherst 1986.

[Kandel and Schwartz, 1985] Kandel E. R. and Schwartz J. H. (1985). *Principles of Neural Science.* Elsevier Science Publishing Co, Inc, New York. 2 edition.

[Kohonen, 1988] Kohonen T. (1988). *Self-Organization and Associative Memory.* Springer-Verlag, Berlin. 2 edition.

[Lang J.K., 1988] Lang J.K. H. G. (1988). A time-delay neural network architecture for speech recognition. Tech. Rep. CMU-CS-88-152, Dept. of Computing Science, Carnegie-Mellon University, PA.

[Lansner and Ekeberg, 1985] Lansner A. and Ekeberg Ö. (1985). Reliability and speed of recall in an associative network. *IEEE Trans on Pattern Analysis and Machine Intelligence* **7**: 490–498.

[Lansner and Ekeberg, 1987] Lansner A. and Ekeberg Ö. (1987). An associative network solving the "4-bit adder problem". In *Proceedings of the IEEE First Annual International Conference on Neural Networks*, pp. II–549. San Diego, USA.

[Lansner and Ekeberg, 1989] Lansner A. and Ekeberg Ö. (1989). A one-layer feedback, artificial neural network with a Bayesian learning rule. *International Journal of Neural Systems* **1**: 77–87, Also extended abstract in Proceedings from the Nordic symposium on Neural Computing, April 17–18, Hanasaari Culture Center, Espoo, Finland.

[Lansner *et al.*, 1989] Lansner A., Ekeberg O., Tråvén H., Brodin L., Wallén P., Stensmo M., and Grillner S. (1989). Simulation of the experimentally established segmental, supraspinal and sensory circuitry underlying locomotion in lamprey. *Soc. Neurosciene Abstr* **15**: 1049.

[Massone L, 1989] Massone L B. E. (1989). A neural network model for limb trajectory formation. may be submitted, currently don't know.

[Parsons, 1987] Parsons T. W. (1987). *Voice and Speech Processing.* McGraw-Hill Book Company, New York.

[Rosenblatt, 1962] Rosenblatt F. (1962). *Principles of Neurodynamics.* Spartan, New York.

[Shannon, 1951] Shannon C. (1951). Prediction and entropy of printed english. *Bell System Technical Journal* **30**: 50–64.

[Tråvén, 1988] Tråvén H. (1988). Temporal associative memory. Tech. Rep. TRITA-NA-P8802, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.

[Waibel *et al.*, 1989] Waibel A., Hanazawa T., Hinton G., Shikano K., and Lang K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **37**: 328–339.