Pergamon

Neural Networks, Vol. 9, No. 7, pp. 1099–1118, 1996

Printed in Great Britain
0893–6080/96 $15.00+.00

PII: S0893-6080(96)00034-2

CONTRIBUTED ARTICLE

On Temporal Generalization of Simple Recurrent Networks

DeLiang Wang, Xiaomei Liu and Stanley C. Ahalt

The Ohio State University

(*Received* 20 *April* 1995; *revised and accepted* 4 *March* 1996)

Abstract—*Simple recurrent networks (Elman networks) have been widely used in temporal processing applications. In this study we investigate temporal generalization of simple recurrent networks, drawing comparisons between network capabilities and human performance. Elman networks are trained to generate temporal trajectories sampled at different rates. The networks are then tested with trajectories at the trained rates and other sampling rates, including trajectories representing mixtures of different sampling rates. It is found that for simple trajectories the networks show interval invariance, but not rate invariance. However, for complex trajectories which require greater contextual information, these networks do not seem to show any temporal generalization. Similar results are also obtained using measured speech data. These results suggest that this class of recurrent networks exhibits severe limitations in temporal generalization. Discussions are provided regarding rate invariance and possible ways to achieve it in neural networks. Copyright © 1996 Elsevier Science Ltd*

Keywords—Neural networks, Temporal generalization, Recurrent networks, Elman networks, Speech processing, Rate invariance, Interval invariance, Simple sequence, Complex sequence.

1. INTRODUCTION

Time plays a fundamental role in almost all kinds of cognitive behavior, including perception, memory, and motor pattern generation. A considerable amount of work has been done in neural networks to address various aspects of temporal processing, ranging from recognition to production of temporal patterns, such as abstract sequences, speech, time series, and grammatical structures (see Wang, 1995). There are two ways that time is embedded in a temporal pattern: temporal order and time duration. Temporal order refers to the ordering of components in a temporal sequence, and time duration describes how long each component lasts.

1.1. Psychological Background on Temporal Generalization

Time duration plays a critical role in temporal processing. In speech recognition, for example, the relative durations of vowels (e.g., /i:/ in "sheep" and /i/ in "ship") are important. In motor control, the timing of limb motions often characterizes different gaits. More interestingly, humans seem to pay more attention to the relative timing among components of a temporal pattern than the absolute durations of components. For example, when listening to a song or a piece of music we can readily adjust to rate changes and can recognize the piece when it is played at a different speed (tempo). It is well known that subjects can reproduce an acquired temporal pattern with different rates, provided that relative timing is not changed. This holds for both speech production and music generation (Klatt, 1976; Sloboda, 1985; Port et al., 1987, 1996). A substantial body of psychological evidence demonstrates that, to a certain extent, human subjects exhibit rate invariance in recognizing temporal patterns (Klatt, 1976; Bartlett & Dowling, 1980; Watson & Foyle, 1985; Espinoza-Varas & Watson, 1986; Port et al., 1987; Kidd & Watson, 1988, 1992). Bartlett and Dowling (1980) found little effect of presentation rate in learning and recognizing transposed melodies. In

Acknowledgements: We wish to thank Jeff Elman, Osamu Fujimura, Mari Jones, John Kolen, and Ashok Krishnamurthy for useful discussions. We also wish to thank two referees whose extensive comments have significantly improved the presentation of the paper. The work described was supported in part by the NSF grants IRI-9211419, IRI-9423312, and CDA-9413962, the ONR grant N00014-93-1-0335, and by WPAFB.

Requests for reprints should be sent to DeLiang Wang, Department of Computer and Information Science, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210-1277, USA.

speech perception, listeners appear to be capable of adjusting their perception according to perceived speaking rates without affecting perceptual outcome (Klatt, 1976). Port and Dalby (1982) found overall speaking tempo does not have a large influence on the voicing of a consonant, provided that the ratio of the consonant to the preceding vowel is kept constant. Port et al. (1987) examined mora timing in Japanese, which asserts that Japanese speech is composed of timing units—moras—with roughly equal intervals, and they found little effect of speaking rate on such mora timing. Watson and his colleagues have conducted extensive studies on human temporal generalization capabilities in the perception of tonal sequences. In the experiments of Watson and Foyle (1985), subjects were asked to discriminate variations in serial positions and frequencies of the components. For instance, subjects listened to a sequence with different durations (corresponding to different presentation rates). Their results show that subjects' performance in discrimination tasks is affected by the number of freely varying components in the sequences, but not by the presentation rate—at least over a limited range of rates. When human subjects listen to acoustic patterns, adjustment for rate variation is obtained naturally (Kidd & Watson, 1988).

On the other hand, human subjects seem to be sensitive to relative durational variations. Relative durations characterize music scores. The experimental results of Bartlett and Dowling (1980) suggest that both experienced and inexperienced subjects store relative durations of a melody in their long-term memory after they have learned the melody, and use the relative durations in recognition tasks. Varying the relative rate of musical notes makes a piece of music much harder to recognize. The threshold of the just noticeable change of the duration of a component was found to be about 10–20% of the duration of that component (Espinoza-Varas & Watson, 1986). Jones and Ralson (1991) have studied human performance on melody recognition with respect to rhythmic changes, which corresponds to relative durational changes of a temporal sequence. After being trained to recognize melodies, the subjects were tested on durational variations of the learned melodies. It was found that changes in relative durations significantly lower a listener's ability to distinguish target melodies from decoys. This was further confirmed by the study of Kidd and Watson (1992) that emphasized the critical importance of relative duration of a component with respect to the total duration of a temporal pattern (they termed their observation the proportion-of-the-total-duration rule).

In speech perception, relative durations also play an important role. As pointed out earlier,

durations sometimes play critical roles in recognition—it is more important in some languages than others. Klatt (1976) reviewed at different levels many effects of relative durational structures of an English sentence, such as semantic emphasis, word-final lengthening, and linguistic stress. It has been argued that a rhythmic structure (durational structure) is important for speech perception in general (Jones & Boltz, 1989; Levelt, 1989). Port and Dalby (1982) proposed that the durational ratio of consonant/vowel is a primary acoustic cue for English voicing. The principle of isochrony asserts the existence of the regular structure of relative onset-to-onset durations of stressed syllables in a continuous speech, and this principle has been argued to exist in some languages [see Port et al. (1996), for a review on the dispute surrounding the isochrony principle]. In French and Spanish, each syllable in a continuous speech seems to have roughly the same duration (Levelt, 1989, p. 392). Particularly in Japanese, mora timing has been established as a governing rule of speech (Port et al., 1987; Han, 1994). On the other hand, we realize that speech perception is a highly complex process involving many dimensions, and that it is sometimes difficult to put temporal boundaries on continuous speech. Variations to the rate of speech are not equally distributed across all phonemes and spaces between syllables. Some components, such as vowels with steady-state formants (primary resonant frequencies of the vocal track), are varied in proportion to the rate of an overall speech, while other components, such as consonant phonemes preceding or following vowels, maintain relatively constant durations (Handel, 1989). In sum, although probably not as definitively important as for the perception of music and nonspeech signals, relative durations are undeniably very important aspects of speech perception.

The following subsection reviews neural network research that addresses the problem of temporal learning.

1.2. Neural Network Studies

Multilayer perceptrons have been extended to include recurrent connections in order to perform temporal processing (Jordan, 1986; Pineda, 1987; Williams & Zipser, 1989; Elman, 1990). An advantage of these recurrent networks is that time is represented implicitly in the architecture by incorporating a form of *short-term memory* (STM) that is implemented through feedback connections. This type of recurrent network was first studied by Jordan (1986), whereby part of the input layer receives external input, and the rest of the input layer consists of state or context units that retain contextual information one-to-one projections from the output layer and feedback from themselves. Elman (1990) later proposed another architecture in which part of the input layer (context layer) simply holds a copy of the activation of the hidden layer from the previous time step. In these two recurrent architectures, the context units function as STM. Following Cleeremans et al. (1989), Elman networks are also referred to as simple recurrent networks in this paper, and such recurrent networks are among the most popular of recurrent architectures. Simple recurrent networks can be extended to fully recurrent networks in which each unit has connections with all the other units of the network (Williams & Zipser, 1989).

This type of recurrent network has been widely used for temporal processing tasks. Sequence recognition tasks have been carried out by these recurrent networks. For example, Watrous et al. (1990) described a recurrent network for speech processing. The network was trained to discriminate /b/, /d/, and /g/ in the consonant–vowel syllables. Port (1990) used a form of the Elman network for recognizing melodies, and described that the system exhibits a chain of associations between various stable states. Pollack (1991) demonstrated that a similar recurrent network can recognize high-level temporal structures. Cleeremans and collaborators have used simple recurrent networks extensively for learning abstract sequential structures of temporal patterns, and reported successful comparisons with some psychological data of human sequential learning (Cleeremans, 1993). Hanes et al. (1994) used simple recurrent networks to map acoustic data to phonetic representations. In this experiment, which we shall refer to later, networks were trained to recognize phonemes in consonant–vowel–consonant syllables. They used formant data as input, and each output unit represented the probability of the presence of a particular phoneme.

Recurrent networks have also been used for temporal sequence generation. For example, Pearlmutter (1989) trained recurrent networks to learn state space trajectories using continuous recurrent networks which evolve through time according to a set of differential equations. Massone and Bizzi (1989) applied Jordan's network to generate artificial limb trajectories for robot control. The network was trained to translate a sensory stimulus directly into a temporal sequence of muscular activation which corresponds to minimum jerk trajectories.

In addition to recurrent networks which are based on multilayer perceptrons, several other recurrent architectures have been proposed for temporal sequence processing. One such architecture is based on the dynamics of the Hopfield model of associative memory (Tank & Hopfield, 1987), where various time delays were used as STM. This network recognizes sequences through attractor dynamics, and an input sequence controls the process of dynamic evolution so that the most similar temporal template will be recalled when the equilibrium is reached. This model was later used for spoken digit recognition (Unnikrishnan et al., 1992). Another architecture uses temporal template matching for recognition and recurrent connections from the recognition layer to an STM layer to generate complex temporal sequences (Wang & Arbib, 1990, 1993). STM can be encoded either as activity decay, or interference among components of a sequence. With the help of self-organization which is implemented by recurrent connections within the recognition layer, Wang and Yuwono (1995) proposed a network that learns temporal patterns based on anticipation and match between the network's anticipation and the actual input in the next time step. The network was shown to be capable of generating any complex sequence and learning temporal patterns incrementally.

While the importance of temporal generalization is well established in human performance, only limited research has been conducted to assess temporal generalization capabilities of recurrent networks. Tank and Hopfield (1987) incorporated limited distortions in component intervals. Wang and Arbib (1993) showed that their network can recognize learned sequences when the time courses of these sequences are varied, and produce learned sequences with different rates. For simple recurrent networks, Port (1990) reported that their network did not generalize well when the tempo of a learned sequence was slowed by a factor of two. He further reported that a fully recurrent network was able to generalize appropriately in this situation. Abu-Bakar and Chater (1993) investigated rate-dependent factors in sequence recognition, and reported that the recognition of their network is not affected by the durations of sequence components. They considered only simple sequences (definition given below). The recurrent networks used by Hanes et al. (1994) were able to recognize syllables spoken at both normal and slow speeds after the networks were trained with syllables at both rates. Although all of the above studies show one way or another that neural networks can perform successfully recognition at different rates, these studies have not addressed the critical question of how sensitive the networks are with respect to variations of relative durations.

Because of their widespread use, research investigating temporal generalization of simple recurrent networks is of particular importance. In this paper, we report our results of studying temporal generalization capabilities of simple recurrent networks. We train this type of network to generate temporal trajectories which are sampled at different rates, and investigate the generalization ability of these networks to rate and interval changes of acquired temporal patterns.

The remainder of this paper is organized into the following sections. In Section 2, some basic definitions and the network architecture are specified. In Section 3, we describe our experiments performed on simple sequences. In Section 4, similar experiments with complex sequences are reported. In Section 5, we use measured speech data to verify our basic findings. Finally in Section 6, we conclude the paper with further discussion and suggestions.

2. CONCEPTS AND NETWORK ARCHITECTURE

We represent a sampled trajectory as a temporal sequence. Our definitions concerning temporal sequences follow those of Wang and Arbib (1990). A sequence consists of ordered components. The *context* of each component is the shortest subsequence prior to the component that associates with the component unambiguously. The *degree* of a component is the length of its context, and the degree of a sequence is the maximum degree of all of its components. A *simple sequence* is a sequence with degree one, i.e., in a simple sequence each component is different from every other component (see, for example, *A-B-C-D*). A sequence is *complex* if its degree is greater than one (see, for example, *A-B-C-C-B-A*).

We use the term *rate invariance* to mean that sequence recognition is *not* affected by varying presentation speeds of a sequence, but *is* affected by varying the relative durations of components. In contrast to rate invariance, we use the term *interval invariance* to mean that sequence recognition is not affected by varying presentation durations for individual components of a sequence (Wang & Arbib, 1993; see also Wang, 1995). It is clear from the description of Section 1.1 that human subjects exhibit rate invariance, but *not* interval invariance. It is also clear that rate invariance is more specific than interval invariance, because the latter encompasses all durational variations while the former includes only a specific subset of all possible durational variations.

To illustrate this point, three sequences are shown in Figure 1. Each sequence is composed of three components *A*, *B*, and *C*, and the duration of each component is different in each sequence. Suppose a network is successfully trained to recognize sequence I. The network is subsequently tested on sequences II and III. Notice that sequence II, but not sequence III, has the same relative durations as sequence I. If both sequence II and sequence III are recognized without further training, the network exhibits interval invariance because the network cannot discriminate

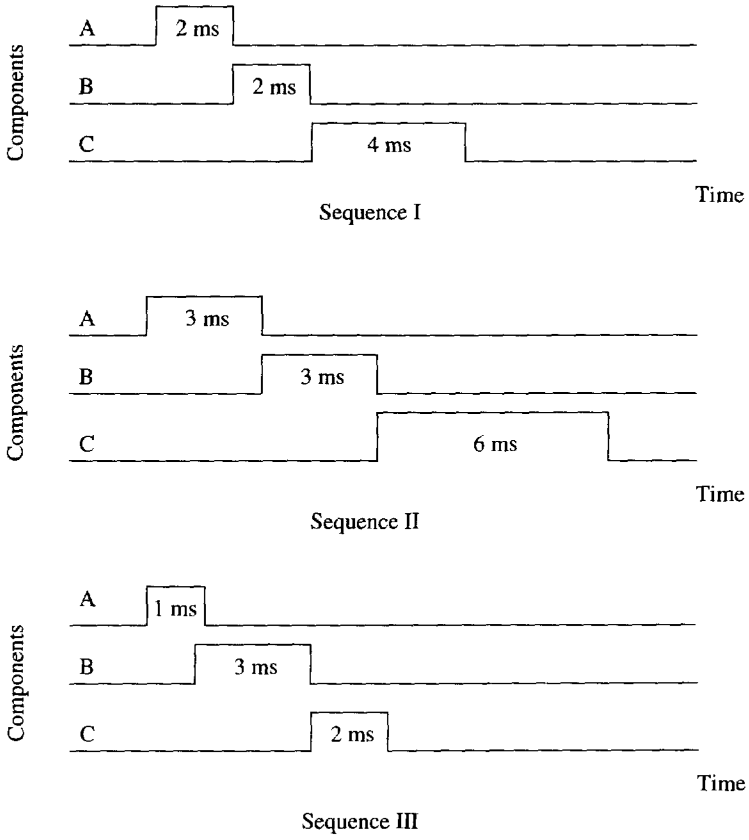


FIGURE 1. Rate invariance vs interval invariance. All three sequences consist of the same components arranged in the same temporal order: *A-B-C*, but with different variations of component durations.

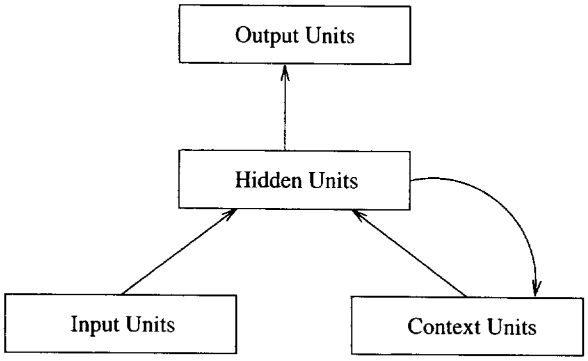


FIGURE 2. The architecture of the Elman network. An arrow represents weight connections in a certain direction.

on the basis of relative durational differences. On the other hand, if *only* sequence II is recognized and sequence III is *not* recognized, the network exhibits rate invariance because the network has distinguished between sequences with the same ordering, but differing relative component durations. Thus, interval invariance differs from rate invariance in that only event order is considered in an interval invariant system, and time durations of the events are totally ignored. But rate invariant networks are sensitive to both event order *and* relative event durations.

In light of the distinction made above, we investigate in this paper both interval invariance and rate invariance of recurrent networks. Our following study will focus on simple recurrent networks (Elman, 1990), because they are well defined and the target of many studies and applications. Figure 2 shows the structure of the simple recurrent network we use for the present investigation. The network has one input layer, one output layer, and one hidden layer, as well as one context layer "appended" to the input layer. There are one-to-one, fixed connections from the hidden layer to the context layer. The context layer feeds forward to the hidden layer in all-to-all correspondence by adjustable weights, thereby providing contextural information (STM) to the network. During both training and testing, input units receive samples of the input sequence, while output units are compared with the desired response of the network sampled at a specific rate. This type of recurrent network can be generally described in vector format as follows. Let the output, $\mathbf{y}(t)$, be generated by $\mathbf{f}_o$, and let the activation of the hidden units, $\mathbf{h}(t)$, be generated by $\mathbf{f}_h$. Then with the input $\mathbf{x}(t)$,

$$\mathbf{h}(t) = \mathbf{f}_h(\mathbf{W}_i\mathbf{x}(t) + \mathbf{W}_c\mathbf{h}(t-1))$$

and

$$\mathbf{y}(t) = \mathbf{f}_o(\mathbf{W}_h\mathbf{h}(t))$$

where $\mathbf{W}_i$, $\mathbf{W}_c$, and $\mathbf{W}_h$ are the matrices of adjustable weights from the input, the context, and the hidden units, respectively, $\mathbf{f}_o$ and $\mathbf{f}_h$ are nonlinear but differentiable, and thus permit training via gradient-descent methods such as backpropagation. Here, $\mathbf{f}_o$ and $\mathbf{f}_h$ are taken to be the sigmoid function. The architecture of the Elman network will be extended in Section 4 where the context layer is expanded to a number of sublayers, each of which holds a copy of the hidden layer.

The network is trained using the standard backpropagation algorithm for recurrent networks (Elman, 1990). This learning algorithm is derived in a similar way as for multilayer feedforward networks. That is, gradient descent is used to minimize an error criterion that is computed as the summation of the squared difference between the actual and the desired output trajectories. Weights are updated at each time step. The training procedure used here can be viewed as a special case of the training algorithm given by Williams and Zipser (1989) for fully recurrent networks.

To investigate temporal generalization, we train the network to produce temporal trajectories of various complexities. In order to correctly produce a sampling point on a trajectory, however, the network must be able to recognize the context which consists of the current input and the activity held in the context units (Figure 2). We vary the presentation rate of a trajectory by sampling the trajectory with different densities, which we call *sampling rates*. A high sampling rate corresponds to a high-density sampling, and a low sampling rate to a low-density sampling. Since an input signal produces an output signal at each step during sequence generation, different sampling rates also correspond to different rates of presenting the same sequence. Figure 3 illustrates this point with the trajectory of a sine wave with one period. Figure 3a shows the original trajectory. Figures 3b–3d show the same trajectory sampled at the base rate, the double rate, and the triple rate, respectively. In this figure, the step size is held constant, and different rates of sampling are directly converted to different rates of presenting the original trajectory. Thus, to examine temporal generalization of a network, we can train the network with a sampling rate, and test the performance of the network with a different sampling rate.

3. SIMPLE SEQUENCES

We begin our experiments with simple sequences. Since temporal dependence spans only one time step in a simple sequence, very limited contextual information is needed to recognize/generate the sequence. Since an STM model is used to realize temporal dependencies, only modest STM require-

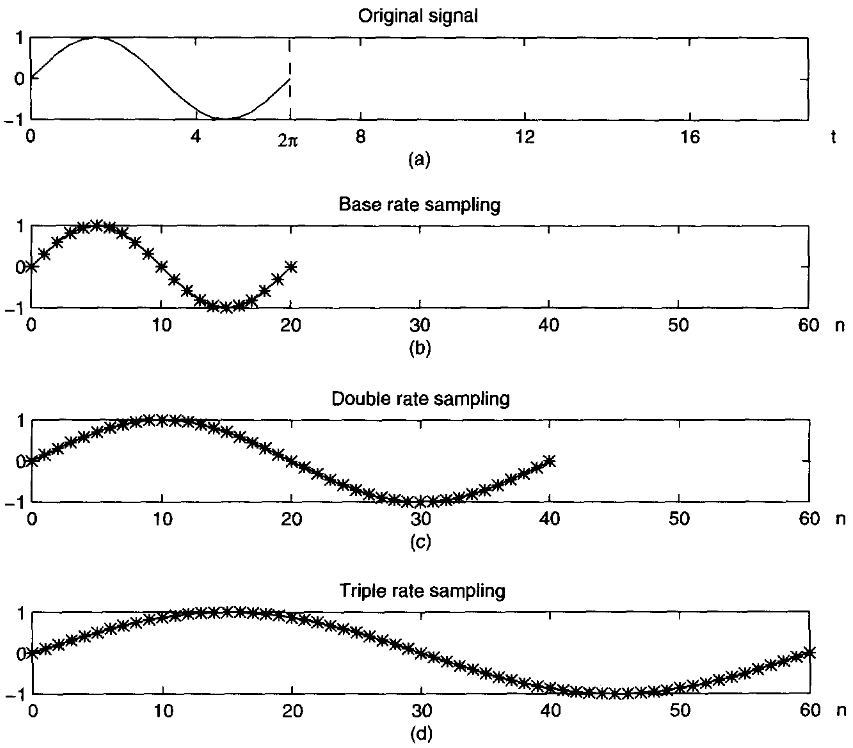


FIGURE 3. Sine wave trajectory with three rates of sampling. (a) Original continuous trajectory. (b) Base rate sampling with the sampling interval $T = 2\pi/20$ and sampling points $t_n = nT, n = 0, \ldots, 20$. (c) Double rate sampling with the sampling interval $T = 2\pi/$ 40 and sampling points $t_n = nT, n = 0, \ldots, 40$. (d) Triple rate sampling with the sampling interval $T = 2\pi/60$ and sampling points $t_n = nT, n = 0, \ldots, 60$.

ments are imposed on the networks by simple sequences.

3.1. Input/Output Representation

The network is trained to produce the contour trajectory of a "folded" double-period sine wave, which is shown in Figure 4. As described earlier, the continuous trajectory is sampled into a set of discrete sample points, forming a temporal sequence. Each sample point has a pair of values: an input and an output, and the output is used as the desired output during learning. The order of the input samples is indicated in the figure by arrowheads. Therefore, the

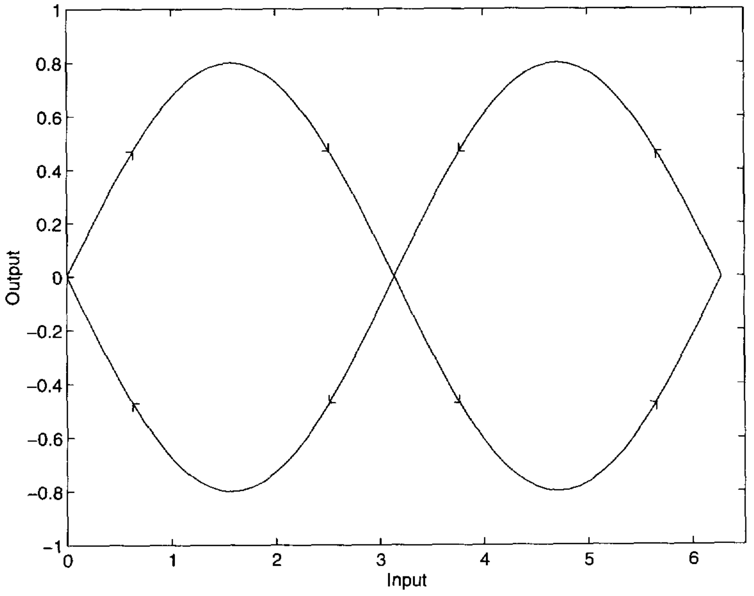


FIGURE 4. Trajectory of a folded sine wave. Arrows indicate time flow directions. Simple sequences are obtained by sampling this trajectory at different rates.

training process for a sequence consists of sequentially presenting to the network the input signal and the (desired) output signal. The main purpose of using the folded sine wave is to create repetitive input sequences so that the context of an input signal is important to determine the correct output signal.

Three sampling rates are used in this experiment. For the base rate, 40 sample points are used to represent the input and output sequences. The input is a sequence of values going from 0 to 2π and then from 2π to 0, corresponding to the axis of the sine function. In order to form a closed output contour, the desired output sequence is the sine of the input when the input goes from 0 to 2π, and the negative sine of the input when the input goes from 2π to 0. At the double and triple sampling rates, the same input trajectory is represented by 80 and 120 sample points, and the lengths of the two resulting sequences are 80 and 120, respectively. We note that (1) this sequence is simple since the output is determined uniquely by the state of the network at the last time step, and (2) the base rate satisfies the Nyquist sampling theorem.

3.2. Experiments and Results

The network was first trained with the base sampling rate. After training, the network was able to produce the sequence of the sampled sine trajectory at the base sampling rate correctly (Figure 5a). Only four hidden units were needed for the Elman network to learn this sequence. The network was then tested at the double sampling rate. From the results shown in Figure 5b, it can be seen that the output trajectory for the double rate differs dramatically from the desired trajectory. This demonstrates that the network with the training method cannot automatically generalize to a sampling rate at which it has not been trained, even for simple sequence production. This result is consistent with that obtained by Port (1990), who reported experiments on simple recurrent networks too, and noted that a change in the presentation rate results in poor network performance. These experimental results suggest that temporal generalization capability is not an intrinsic property of simple recurrent networks.

To investigate the Elman network further, another network was trained alternately at the base rate and the triple rate so that it could perform production successfully at both sampling rates. After training, the network was tested on the double sampling rate which had not been used in training. The results are shown in Figure 6. From Figure 6, it can be seen that

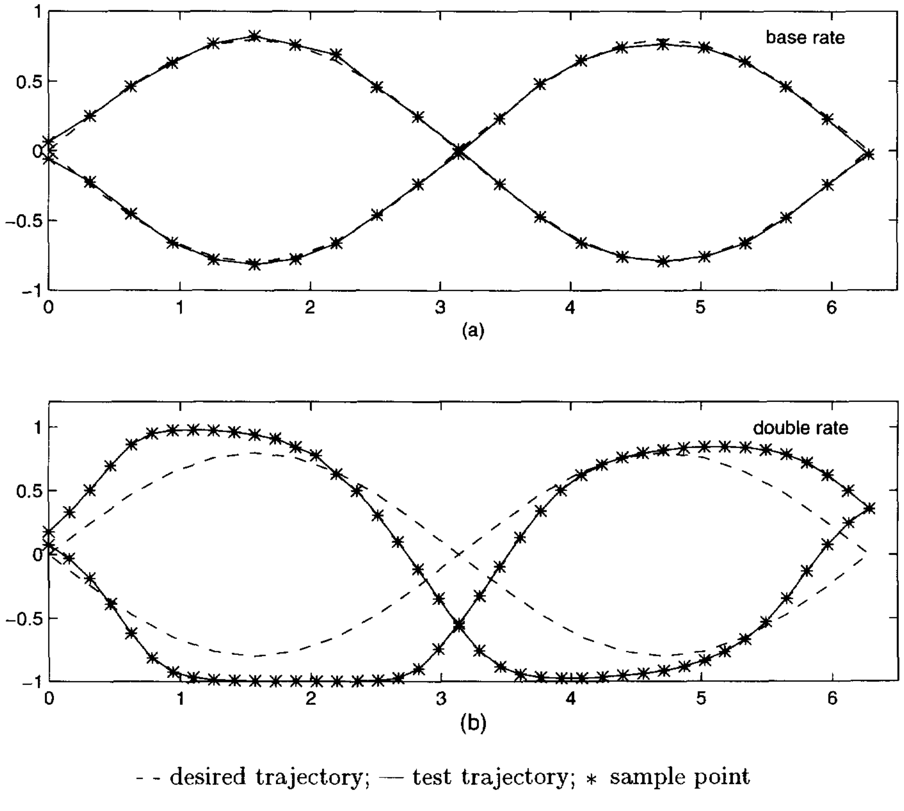
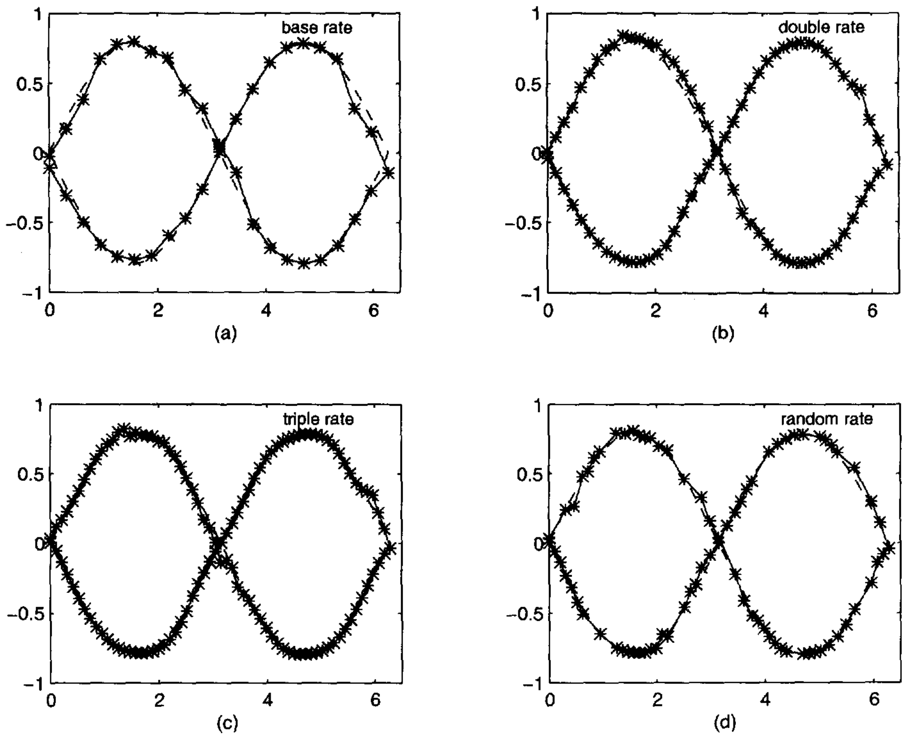


FIGURE 5. Training and test results with a simple sequence sampled at the base and the double rates. The abscissa indicates input which goes from 0 to 2π and then from 2π to 0. (a) Test trajectory at the base rate is used during training. (b) Test trajectory at the double rate which was not used during training. The network size is $1 \times 4 \times 1$, i.e., one input unit, four hidden units, and one output unit. Training took 20 000 iterations. The sigmoid function used has a slope of 1 and a magnitude of 2 (from −1 to 1). The learning rate was 0.1. Weights were randomly initialized between −2.5 and 2.5.



- - desired trajectory; — test trajectory; * sample point

FIGURE 6. Training and test results with a simple sequence sampled at the base, double, triple, and random rates. (a) Test trajectory at the base rate as used during training. (b) Test trajectory at the double rate which was not used during training. (c) Test trajectory at the triple rate as used during training. (d) Test trajectory at the random rate which was not used during training. The random rate is composed of a random mixture of the base, double, and triple sampling rates. The network size is 1 × 10 × 1. Training took 120 000 iterations. The learning rate was 0.1 for the first 60 000 iterations, 0.05 for the next 40 000 iterations, and 0.01 for the last 20 000 iterations. To improve training performance, we started by training a network with six hidden units (100 000 iterations) and later expanded the trained network to include four more hidden units for further training (20 000 iterations). The weights involving the first six hidden units were randomly initialized between −2.5 and 2.5, and those involving the last four hidden units were initialized to 0. Other parameters are the same as used in Figure 5.

the network was able to produce the sequences at both the base and the triple rates on which it was trained (Figures 6a and 6c). Note that training the network to produce two sequences with different sampling rates is more difficult than the training required to produce either one of the two separately. Thus, in our experiments, the network was first trained with six hidden units, but later tests indicated that six hidden units were insufficient to yield successful training. The network was subsequently expanded to 10 hidden units with the newly added connections initialized to zero and the old connections (from the six hidden units previously trained) remaining unchanged. After further training, the expanded network successfully learned to produce both sequences. This way of increasing from six hidden units to 10 is merely a technique for speeding up the training process. The test results displayed in Figure 6b show that not only did the network correctly produce the base and triple rate trajectories, it also correctly produced a double rate trajectory on which it had not been trained. This indicates that the network can generalize to a new sampling rate that is between the two rates on which it was trained.

To address the question of whether the temporal generalization shown in Figure 6b is the desired rate invariance, the same network was then tested on a sequence generated from a random mixture of normal, double, and triple rates. As explained in Section 2, this is equivalent to randomly varying the interval durations of the components of the sequence. To our surprise (and dismay!), as shown in Figure 6d, the output trajectory of the network was not significantly degraded by randomly varying the intervals of the input sequence. Thus we were led to the conclusion that temporal generalization as exhibited in Figure 6 is interval invariance, and not rate invariance. From a different perspective, we may say that the network over-generalized in time because it generalizes to treat all of the durational changes in the same way (or to ignore the relative durational information which is embedded in the training sequences). The network's ability to generalize to a new sampling rate, in this case the double sampling rate, is therefore an example of over-generalization,

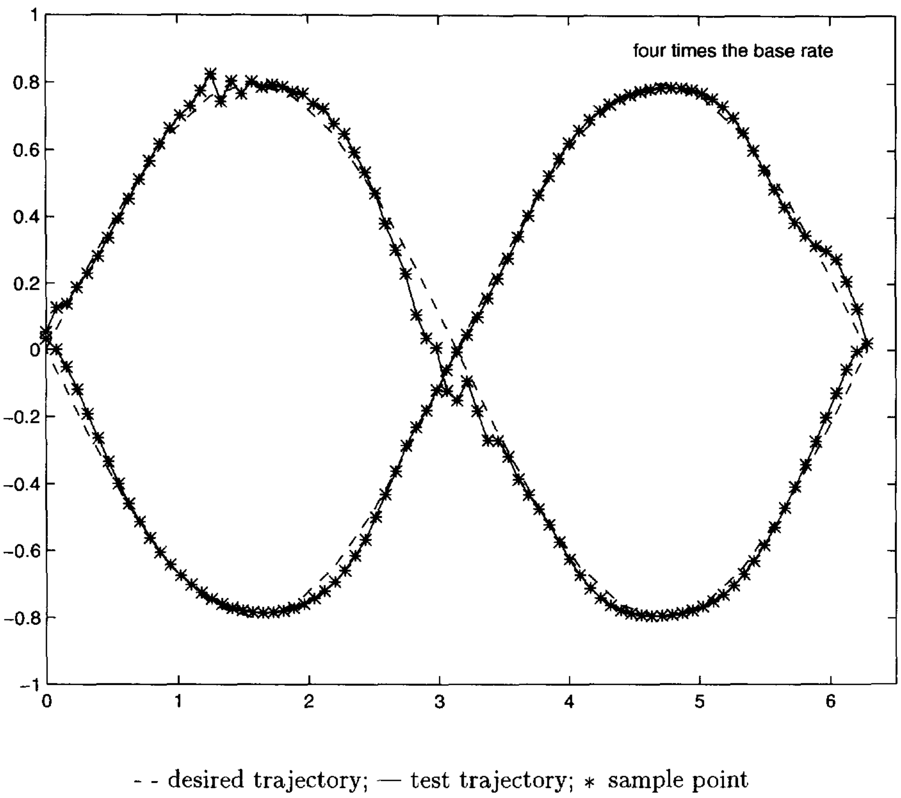


FIGURE 7. Generalization to sampling rates outside the range of training samples. Test trajectory at four times the base rate. The network used for testing is the same as in Figure 6.

or interval invariance. To further confirm our observation, we tested the same network on a sampling rate which was outside the scope of the two sampling rates used in training, that is, four times the base sampling rate. The result is shown in Figure 7. Consistent with the results in Figure 6, the network performed well on the sampling rate that is four times the base rate.

4. COMPLEX SEQUENCES

The results discussed so far were for simple sequences. In the real world, however, simple sequences are rare and most natural sequences are complex sequences. To uniquely specify a component of such complex sequences, more information is needed than just the state of the network at the last time step. To process complex sequences correctly, a recurrent network must develop extended STM to keep track of the network states earlier than just the previous time step. We conducted the following experiments to see how the difference in STM requirements affects the temporal generalization capability of simple recurrent networks.

4.1. Network Structure and Training Method

When training sequences are complex sequences, more powerful STM is needed to successfully recognize/produce the input sequences. A straightforward way to increase the STM capability of an Elman network is to increase the number of hidden units. When the number of hidden units becomes large, there are many connection weights and training the network takes a very long time and becomes more sensitive to initial conditions. When training simple recurrent networks on complex sequences, we find that training becomes more difficult as temporal dependence among sequence components increases. To learn to generate the complex sequence to be described later in this section, we needed 40 hidden layer units. In a typical experiment, it took an HP 715 workstation about 7 h to complete 10 000 iterations. Because of the complexity of the problem, tens of thousands of iterations are usually needed for each trial. In short, the training becomes very time consuming. This difficulty has also been discussed by other researchers. For example, Bengio et al. (1994) proved that gradient-descent based learning algorithms face increasing difficulties as the length of the dependencies to be captured increases. Elman (1993) suggested an incremental input method to improve training. The basic idea of his method is to gradually increase the complexity of the training samples. This method usually involves training on a large number of different training samples. In our experiment, however, we trained with at most two sequences (of equivalent complexity) at the same

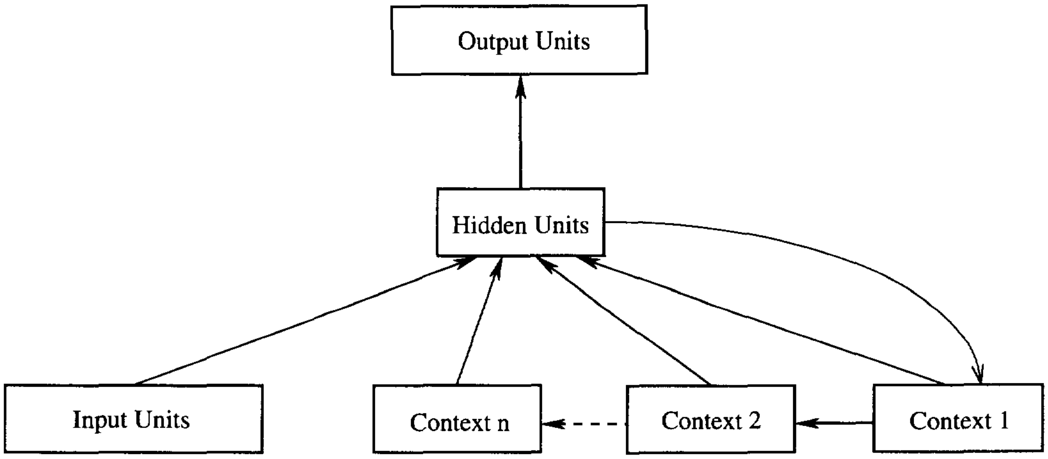


FIGURE 8. The architecture of the modified Elman network. See the caption of Figure 2 for explanations. The dotted arrow indicates omitted context sublayers from 3 to $n - 1$.

time, therefore the complexity of the sequences could not be increased gradually and incremental training could not offer much help.

To facilitate training, a new network structure, a modified Elman network, is used in our experiment. This new structure is shown in Figure 8. The context layer is expanded to a number of sublayers, and each context sublayers holds one copy of the hidden layer. Thus, several copies of the hidden layer are maintained in the network. At each time step, the contents of a context sublayer are shifted to its right sublayer. The contents of the rightmost sublayer are discarded, and the leftmost sublayer directly receives input from the hidden layer. In the original Elman network, the trace of entire STM is blended into a single context layer. As pointed out by Wang and Arbib (1993), because of this blending and compressing, the content of each component in a complex sequence becomes difficult to represent in STM. In the modified architecture, short-term memory is spread over multiple context sublayers so that the necessary information for sequence processing can be held for a longer time. Since more information about the past is available at the time of sequence production, these networks should acquire complex sequences more easily than that of the original Elman network. This is confirmed by the following experiments.

To compare training effects of the modified Elman network with those of the original Elman network, both networks were trained to perform the same task. The total number of connections required in the modified architecture was found to be much less than that required for the original Elman network and training time was decreased dramatically in our modified network. For example, in order to learn the complex sequences used in this experiment, the Elman network required 40 hidden units, and thus about 1600 weights. In contrast, only about 200 weights were needed using the modified Elman network with three context sublayers and with each sublayer having eight units. Instead of 7 h to complete 10 000 iterations for successful training, less than an hour was needed with the modified architecture. As before, the training algorithm was based on back propagation.

4.2. Input/Output Representation

The complex trajectory used for the following experiments is shown in Figure 9. This trajectory is composed of two passes. The input trace for the second pass is exactly the same as for the first pass. For each pass, the input is a sequence of values that varies gradually from 0 to 2 and then from 2 to 0 (see Figure 9). The output trace of the first pass is a diamond-shaped curve, that of the second pass is a hexagon-shaped curve. The two passes of the entire trajectory have substantial overlaps. As in Section 3, this complex trajectory was sampled with three rates, i.e., the base rate, the double rate, and the triple rate. The three sequences thus obtained contain 17, 33, and 49 samples, respectively. It is the overlaps of the trajectory that make these sequences complex. For the base sampling rate, each overlapping part of the trajectory has three sample points. Therefore, to separate the two passes of the trajectory, the network must keep track of its previous states for at least four time steps in order to generate next components at branching points correctly. The number of prior sample points which needs to be remembered, i.e., the degree of the sequences, for the double and triple sampling rates are six and eight, respectively. Hence, the sequences at higher sampling rates demand more extensive STM.

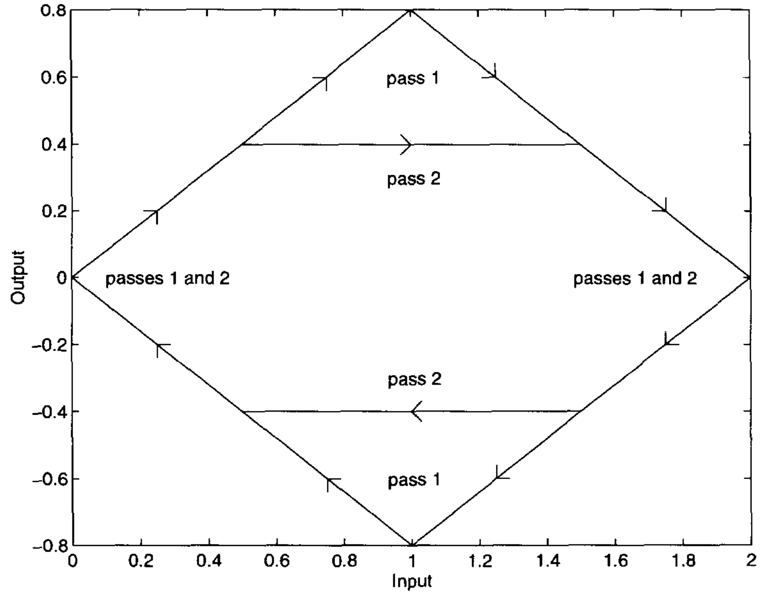


FIGURE 9. A complex trajectory. Arrows represent time flow directions. Complex sequences are obtained by sampling this trajectory at different rates.

4.3. Experiments and Results

The following training procedure is similar to that described in Section 3.2. An Elman network with 40 hidden units was trained alternately with the two sequences that were generated by sampling the trajectory of Figure 9 at the base rate and triple rate. After the network learned both training sequences, it was tested on the sequence sampled at the double rate that had not been used during previous training. The results are shown in Figure 10. As displayed in Figures 10a and 10c, the network learned to generate the base and the triple rate sequences correctly. However, in contrast to the simple sequence case, the network did not correctly generalize to produce the sequence at the double sampling rate. Although the sequence generation was not good, a closer inspection reveals some interesting aspects of the generated trajectory. A reminiscence of the target trajectory was exhibited. We further tested the network with a random mixture of sequences obtained at all of the three sampling rates. As shown in Figure 10d, the network did not generalize well to these durational variations either. Similar to Figure 10b, the network seems to have obtained some elements of the target trajectory. All the above results indicate that the temporal generalization capability is impaired when the temporal sequence being processed becomes more complex. The fact that generalization in Figures 10b and 10d matches some elements of the target sequences suggests that the network exhibits some interval invariance. On the other hand, because the network performance in Figure 10d is not worse than that in Figure 10b, we can clearly rule out any rate invariance in these experiments with complex sequences.

We find by probing the activities of hidden units that, although 40 hidden units are needed to successfully learn the sequences at both the base and the triple rates, only a handful of these units actually exhibit significant activities during test trials. In addition, the same set of units is involved in generating sequences despite all durational variations. During training with different rates of sampling, the same set of hidden units has developed significant representations, and this same set is involved in all subsequent tests. The same phenomena occurred when we conducted the training with simple sequences (see Figure 6). This observation suggests that small differences of hidden layer activation may be responsible for yielding the responses to different durational variations.

As described in Section 4.1, a modified Elman network with eight hidden units and three context sublayers was trained on the same task. The trained network was tested on the complex trajectory of Figure 9 sampled at the base, the double, and the triple rates, as well as a random mixture of the three sampling rates. The results are displayed in Figure 11. It can be seen from the figure that the results with the modified architecture are quite similar to those obtained using the original Elman network. As shown in Figures 11a and 11c, the network successfully learned the two training sequences, but failed to generalize properly to interval variations. Thus, the modification to the network structure did not significantly change the temporal generalization capability of the network. The performances in

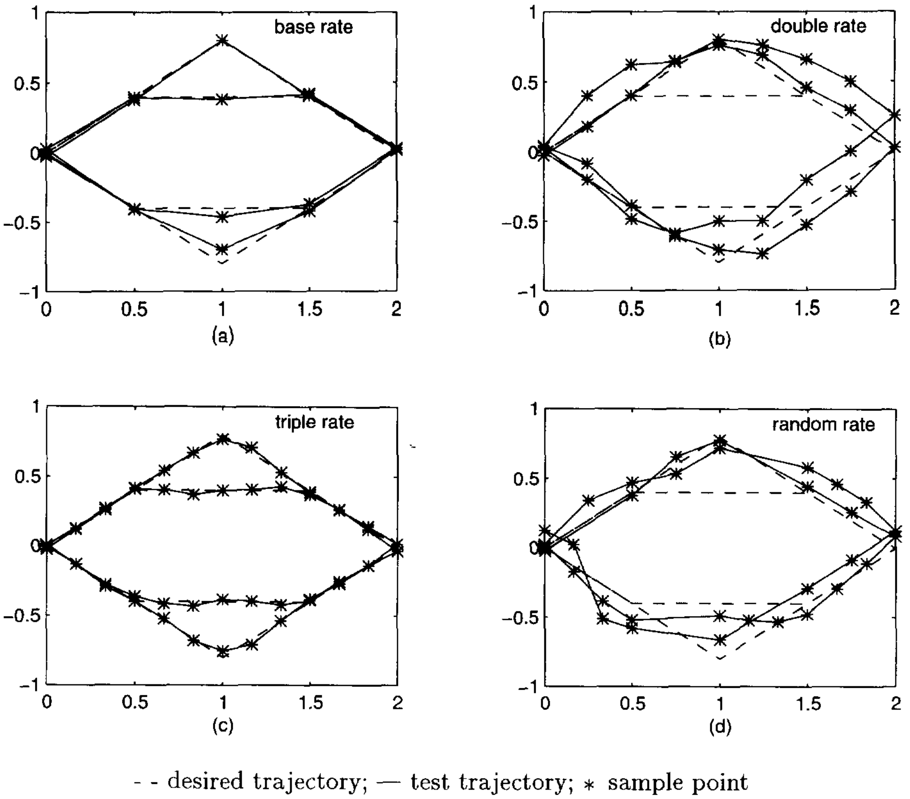


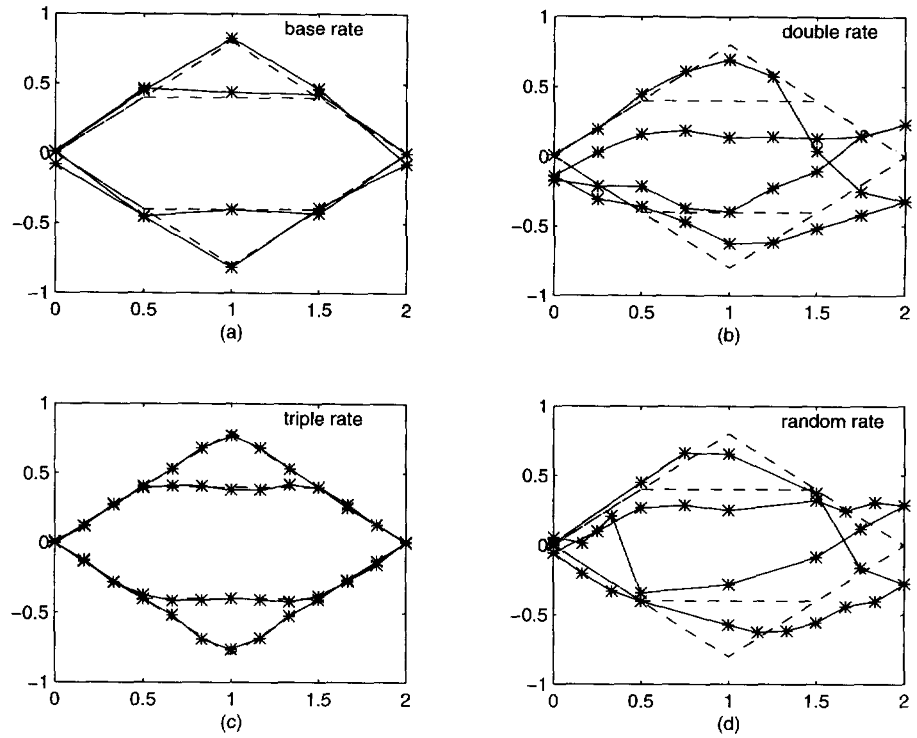
FIGURE 10. Training and test results using the Elman network with a complex sequence at the base, double, triple, and random rates. (a) Test trajectory at the base rate as used during training. (b) Test trajectory at the double rate which was not used during training. (c) Test trajectory at the triple rate as used during training. (d) Test trajectory at the random rate which was not used during training. The random rate is composed of a random mixture of the base, double, and triple sampling rates. The network size is 1 × 40 × 1. Training took 60 000 iterations. The learning rate was 0.1 for the first 10 000 iterations, 0.05 for the next 35 000 iterations, and 0.02 for the last 15 000 iterations. The weights were randomly initialized between −0.2 and 0.2. Other parameters are the same as used in Figure 6.

Figures 11b and 11d are comparable to those of Figures 10b and 10d. Though the latter exhibits a better production of the diamond, the former seems to have picked up the horizontal trajectories better. Although three context sublayers are used instead of a single context layer, temporal generalization is not improved. In other words, an expanded STM does not seem to help temporal generalization of simple recurrent networks. To further verify our observations, we conducted another experiment where the number of hidden units was expanded to 16, double that shown in Figure 11. The network again was able to learn to generate the trajectories both at the base rate and at the triple rate. But the generalization to the double rate and the random rate yielded very similar results as shown in Figures 11b and 11d. This suggests that the failure to generalize to different rates is not caused by insufficient hidden units. It appears that the failure has to do with the way this class of recurrent networks generalizes from input data. As discussed in Section 2, rate invariance imposes a global constraint on temporal generalization, which does not seem to be captured by simple recurrent networks.

We note that a number of studies have reported improved generalization when the number of parameters was reduced rather than increased. This is a way to reduce overfitting the data, which limits generalization. We did not observe this phenomenon in our study. Indeed, only by increasing the number of hidden units were we able to train the networks on more than one sampling rate, as noted in Section 3.2 and also observed in the experiments reported in this section.

5. ACOUSTIC TO PHONETIC MAPPING

After investigating the temporal generalization of simple recurrent networks on synthetic trajectories, we performed an experiment on a more realistic set of data. We selected speech data for our following study because speech processing is one of the primary areas of potential applications of recurrent networks, and earlier research on speech processing suggests that the Elman network can be effectively used in such applications. Neural networks have been used in a number of speech processing tasks (see Lippmann, 1989; Bourlard & Morgan, 1994, for reviews). More specifically, the problem of mapping acoustic waveforms to pho-



- - desired trajectory; — test trajectory; $*$ sample point

FIGURE 11. Training and test results using the modified Elman network with a complex sequence at the base, double, triple, and random rates. (a) Test trajectory at the base rate as used during training. (b) Test trajectory at the double rate which was not used during training. (c) Test trajectory at the triple rate as used during training. (d) Test trajectory at the random rate which was not used during training. The random rate is composed of a random mixture of the base, double, and triple sampling rates. The network size is 1 × 8 × 1. Training took 34 000 iterations. The learning rate was 0.05. The weights were randomly initialized between −0.5 and 0.5. Other parameters are the same as used in Figure 6.

netic representations has been studied recently by Watrous et al. (1990) and Hanes et al. (1994) (see Section 1.2). Their work has focused on mapping the contours of the first three formants to output values that indicate the presence of consonants and vowels in CV or CVC (consonant–vowel–consonant) syllables.

Our experiment is based on an earlier study by Hanes et al. (1994), who successfully used the Elman network to recognize CVC syllables spoken at different rates. In their experiment, a speaker was asked to utter CVC syllables at both normal and slow rates. The initial and final consonants were from the phoneme set {/b/, /d/, /g/} and the vowels were from the phoneme set {/a/, /e/, /i/}. They reported that once trained with one set of syllables spoken at different rates, the network can readily identify phonemes in another set of syllables spoken at different rates. The main purpose of our experiment is to examine generalization of simple recurrent networks on more realistic temporal trajectories; in particular, we want to examine the type of temporal generalization reported by Hanes et al. (1994), since their study seems to imply that the Elman network can perform proper rate generalization.

5.1. Input/Output Representation

In the work reported here, samples of the first three formant trajectories were used as the inputs to the Elman networks, and these samples were estimated from the closed glottal portion of the pitch period. The formant frequencies were estimated using a high-order linear prediction analysis followed by a singular value decomposition. Since the formants are estimated pitch-synchronously, the formant contours are not sampled uniformly. Cubic spline functions were fit to each formant contour, which was then sampled every 10 ms to get a uniformly sampled formant contour. The formant contour was laid out on a wideband spectrogram of the utterance and inspected visually for accuracy.

We adopt an input/output representation similar to that used in Hanes et al. (1994). Three separate networks were used to recognize the initial consonant, the vowel, and the final consonant, respectively. The network for recognizing the initial consonant is $3 \times 9 \times 3$, where three input units represent three formants in the utterance. Formant frequencies are normalized to the range between 0 and 1, with 1 corresponding to 3 kHz. Three output units represent three possible consonants from the set {/b/, /d/, /g/}. Each output unit has a desired trajectory formed on the basis of input formant trajectories. As suggested by Watrous et al. (1990), the desired output trajectories were chosen so that the outputs could be interpreted as the *a posteriori* probabilities of the presence of the individual phonemes. The details of creating output trajectories were given in Hanes et al. (1994), including sample target trajectories for a complete CVC syllable. As argued by Watrous et al. (1990) and Hanes et al. (1994), this method of forming output trajectories produces reasonable outcomes while avoiding entirely arbitrary solutions. During testing, the output unit whose desired output trajectory matches the actual output trajectory most closely is selected as the recognized consonant. The mapping for the vowel and for the final consonant is done in the same manner, except that the network for vowel recognition is 3 × 3 × 3 and for final consonant recognition is 3 × 5 × 3. For vowel recognition, the three output units represent the three vowels {/a/, /e/, /i/}, respectively. For final consonant recognition, the three output units represent three possible consonants from the set {/b/, /d/, /g/}, as for initial consonant recognition.

5.2. Training and Results

The training data consist of 50 utterances of 25 CVC syllables spoken at both normal and slow rates. These 25 syllables are shown in Table 1. There are 27 possible combinations of syllables from the three consonants and the three vowels. Two of them were not used because the corresponding utterance data are not available. In Hanes et al. (1994), the networks were trained with a set of utterances that contain both slow and normal rates, and tested with another set of utterances not used during training. Since we are primarily interested in investigating temporal generalization of these networks to different interval variations, we use all of the 50 utterances as training samples. After training, the networks were able to recognize the initial consonants with a correct rate of 96%, the vowels with 100%, and the final consonants with 98%. Notice that recognition rates are computed using both normal and slow utterances. These results are consistent with those reported in Hanes et al. (1994). During testing, the desired (output) trajectory for an utterance of a different rate is formed by linearly shortening or lengthening the desired trajectory used during training.

After training, the networks are tested on input trajectories which were synthetically generated from the original speech data. During testing, an input trajectory consists of two parts: the first half is

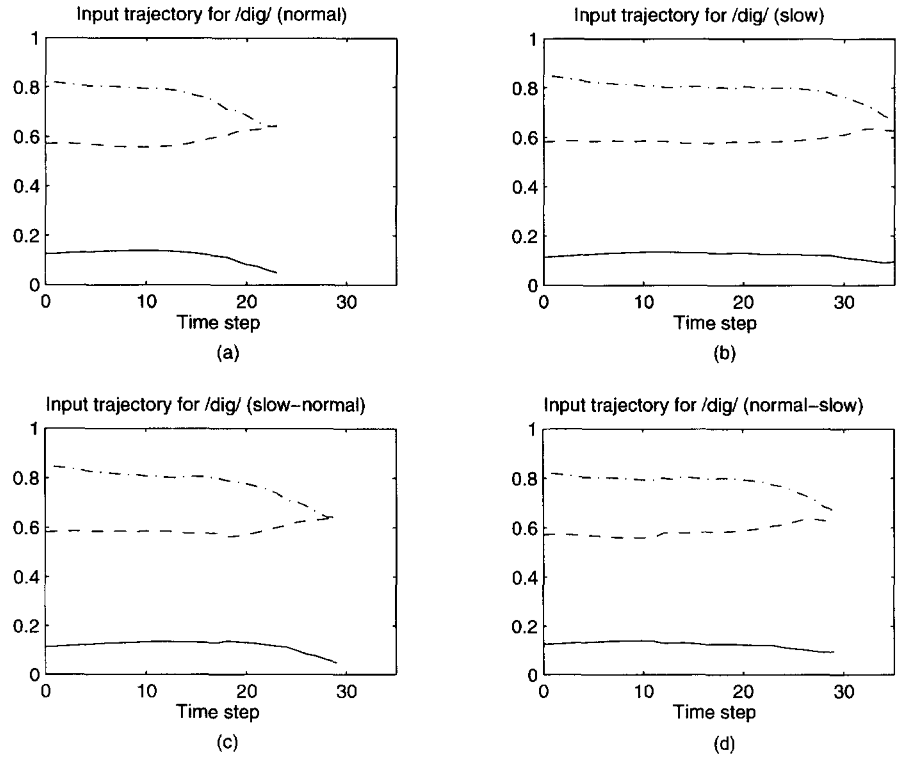


FIGURE 12. Input trajectories with different spoken rates for the utterance /dig/. Only the first three formants are shown. (a) The normal rate utterance. (b) The slow rate utterance. (c) The mixed trajectory with the first half at the slow rate and the second half at the normal rate (slow–normal). (d) The mixed trajectory with the first half at the normal rate and the second half at the slow rate (normal–slow).

TABLE 1
Phonemes Used in the Experiment

bab	dab	gab
bad	dad	gad
bag	dag	gag
beb	ded	geb
bed	deb	ged
beg	deg	geg
bib	dib	gib
bid	dig	gid
big		

composed of the formant data of a syllable spoken at one rate (normal or slow) and the second half is composed of the formant data of the same syllable spoken at a different rate. Figure 12 shows an example with /dig/, where all of the four possible input trajectories are illustrated. The desired (target) output trajectories are constructed similarly. The recognition is achieved by selecting a consonant or vowel whose desired output trajectory is most similar [see Hanes et al. (1994), for detailed description] to

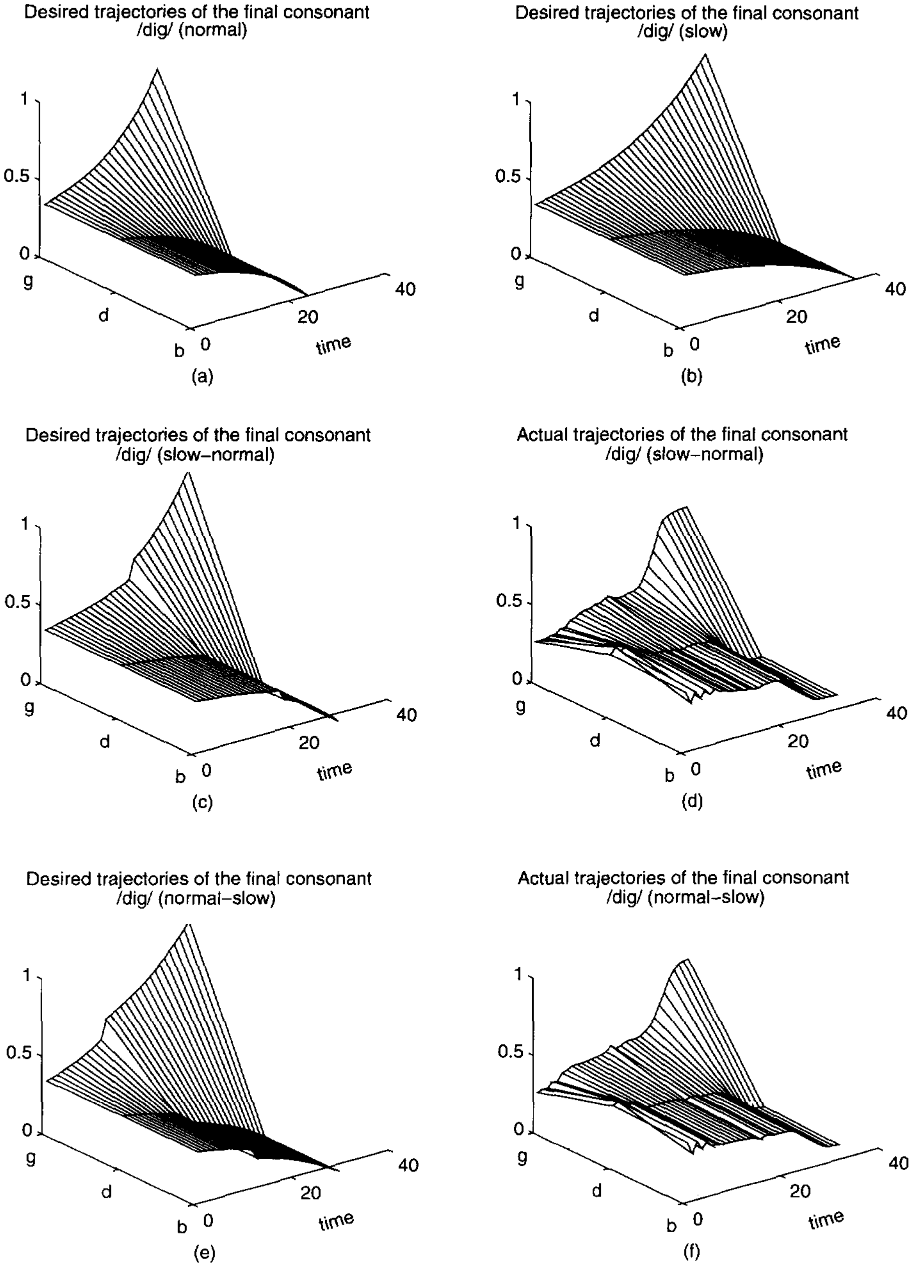


FIGURE 13. Desired and actual output trajectories for the output layer of the final consonant recognition network. These trajectories are for the utterance /dig/. (a) The desired output trajectory for the normal rate utterance. (b) The desired output trajectory for the slow rate utterance. (c) The desired trajectory for the mixture of the slow–normal rate. (d) The actual trajectory for the mixture of the slow–normal rate. (e) The desired trajectory for the mixture of the normal–slow rate. (f) The actual trajectory for the mixture of the normal–slow rate.

TABLE 2
Summary of Network Classification Accuracy

Phoneme	Network Size	Training Accuracy (%)	Test Accuracy (%) (Normal–Slow)	Test Accuracy (%) (Slow–Normal)
Initial	3 × 9 × 3	96	100	92
Vowel	3 × 3 × 3	100	100	100
Final	3 × 5 × 3	98	96	100

the actual output trajectory of the corresponding output unit. The test results of the three networks, corresponding to the initial consonant, the vowel, and the final consonant, are summarized in Table 2. Figure 13 shows the four desired output trajectories of different durational variations for the final consonant of the utterance /dig/. Also shown are the two actual output trajectories of the output layer (with three units) of the final consonant recognition network. In both cases, consonant identification is correct.

As is evident from Table 2, the classification accuracy of the phonemes is not degraded by disproportional interval variations in the test sequences. Note that, in Table 2, the improvement in classification accuracy observed when testing with initial and final consonants occurred predominantly for testing tasks with normal presentation rates in the corresponding phoneme positions (e.g., the case involving initial consonant and normal–slow presentation). The slight degradation/improvement during testing is not statistically significant, given that only 50 utterances were used for testing. What is significant from Table 2 is that the networks not only generalize to recognize the trajectories collected at different rates (Hanes et al., 1994), but also generalize to local interval variations. In other words, the temporal generalization demonstrated by these networks is not rate invariance, but interval invariance. We do not claim that our method of varying the input and desired trajectories of an utterance necessarily corresponds to that of well-formed speech, but only that it is a reasonable way of producing durational changes. Still, it suffices to demonstrate that the recurrent networks used in these experiments do not warrant the temporal generalization of rate invariance, as seemingly suggested in the experimental results of Hanes et al. (1994).

The above task of mapping acoustic waveforms to phonemes is similar to the tasks of producing temporal trajectories investigated in Sections 3 and 4. Both involve transforming from an input trajectory (or trajectories) to an output trajectory, and both involve variations to rate and relative rate. One may argue that speech utterances should have longer temporal dependencies, and thus should show behavior similar to complex temporal sequences (Section 4), not to simple sequences (Section 3). But it is not clear whether the task dealt with here, i.e., recognizing phonemes from the first three formant trajectories, necessarily involves long temporal dependencies. Also, it is not difficult to see that recognition tests both in the experiments of Hanes et al. (1994) and in our experiments have limitations. One major limitation is that there are only three possible outputs for all of the recognition networks, as opposed to many possible outputs studied in Sections 3 and 4. Because of this, no accurate generalization is needed for performing reasonably well in this context of speech recognition.

6. CONCLUDING REMARKS

As described in Section 1.1, many psychological studies conclude that human subjects show rate invariance in recognizing various kinds of temporal patterns. Our study on temporal generalization was motivated by the psychological observation, and we chose simple recurrent networks because they have been widely used and have been shown to exhibit the ability to process temporal information. To examine temporal generalization capabilities of simple recurrent networks, we have conducted a set of computer experiments. From the results of the experiments, several conclusions can be drawn.

First, it is clear that temporal generalization capabilities are not inherent in simple recurrent networks. Simple recurrent networks do not generalize to produce a trajectory at another sampling rate if they are trained with only one sampling rate. Because no explicit mechanism for rate adaptation is present, this should be expected from the learning scheme of these networks, which in essence follows examples (or example-based learning). Simple recurrent networks are different from the networks proposed by Tank and Hopfield (1987) and Wang and Arbib (1993) where some degree of temporal generalization is incorporated into the network design. This conclusion does not imply a limitation on the recurrent networks.

Second, temporal generalization to rate changes is limited for simple recurrent networks after they are trained with example trajectories with rate variations. It appears that these networks show different qualitative behavior when dealing with simple sequences and with complex sequences. With simple sequences, these networks exhibit interval invariance in recognition. For complex sequences, however, the networks do not even seem to exhibit interval invariance (at least not conclusively). What is most conclusive is that simple recurrent networks do not exhibit any form of rate invariance. In other words, they are incapable of recognizing relative durational variations. This conclusion is consistent with all of our experiments, including our investigation using real speech data.

We have conducted our experiments using both the standard architecture of simple recurrent networks (Elman, 1990) with various sizes, and extensions to the standard architecture (Figure 8) to incorporate more extended short-term memory. Thus we believe that our conclusions about temporal generalization of simple recurrent networks are not specific to the particular networks we have used in this paper, but reflect the general characteristics of simple recurrent networks. Our findings are consistent with earlier studies of simple recurrent networks with respect to interval invariance, such as those obtained by Port (1990) and Hanes et al. (1994). On the other hand, we note that our experiments, no matter how extensive they are, cannot decisively conclude that simple recurrent networks are incapable of proper temporal generalization. After all, our conclusions are reached through experiments, not theoretical analyses (for this matter, all positive conclusions about these networks are also not decisive). What we have shown is that some fundamental changes, either with respect to the architecture, the learning algorithm, or the way STM is maintained, must be introduced in order to make this class of recurrent networks perform properly with respect to rate changes.

Rate invariance requires invariance to overall rate changes, but maintains sensitivity to relative durational changes among sequence components. Unlike interval invariance which can be dealt with locally (paying no attention to durations is one way), rate invariance is a global property, thus posing a significant challenge to neural networks as a whole. To our knowledge, no proposed network architecture is able to achieve rate invariance (see also Wang, 1995). The network of Tank and Hopfield (1987) can deal with limited interval variations, but does not seem capable of adjusting to overall rate changes. While the model described by Port (1990) successfully generalizes to different rates, it appears that the network is insensitive to changes in relative component durations. Wang and Arbib (1993) demonstrated that their networks are capable of only interval invariance. In a recent study, Bersini et al. (1994) show that a fully recurrent network based on continuous dynamics of the Hopfield net can recognize temporal trajectories, including a trajectory that is very similar to the folded sine wave of Figure 4. Their results show that the network is capable of recognizing scaled trajectories. Unfortunately, it is unclear whether their network is sensitive to relative durations, a critical question from our perspective. From their design, it does not seem that their network should show sensitivity to relative timing.

Another way of looking at rate invariance is to make an analog between duration in time and size in space. From this analog, rate variations to a temporal pattern correspond to size variations to a spatial pattern, and rate invariance in temporal pattern recognition corresponds to size invariance, albeit one-dimensional, in spatial pattern recognition. Existing techniques to achieve size invariance in pattern recognition may provide useful insights to the problem of rate invariance. One idea to achieve size (scale) invariance is based on a log-polar transformation from a preprocessing layer to the input layer. Under log-polar mapping, scale transformation is converted into translation (Reitboeck & Altmann, 1984), and the same input with different scales activates the same graph at different positions. Translation invariance is relatively easier to handle. A more straightforward idea, as used in Lades et al. (1993), is based on scaling: before an input object is applied to a recognition system, it is scaled by a factor within a certain range, keeping its center fixed. This is the idea behind elastic nets (Durbin & Willshaw, 1987), which can be used to achieve other forms of invariance. By modifying the scale factor within a reasonable range, the stored pattern in the memory layer is identified that shows the highest matching among all possible scales. A similar idea is proposed by Simard et al. (1993) which incorporates invariance directly in the distance measure between a template and an input pattern. Perantonis and Lisboa (1992) used high-order networks to achieve size invariance. In their networks, scale invariance is encoded by forming equivalence classes, each of which consists of similar triangles. Similar triangles are then encoded by a third-order network, where each weight involves three input units and one output unit. It is possible that some of these techniques can be adapted to address rate invariance in temporal processing. For example, temporal patterns can be stored as templates, and in recognition, these templates are allowed to stretch in the time axis within a reasonable range. The template that exhibits the best matching with the input pattern is chosen as the recognition. Whether and to what extent these techniques for size invariant object recognition can be adapted to rate invariance remain to be a topic of future research.

The above techniques for handling size invariance handle scale invariance regardless of the range of scale variations. Of course, an explicit range can always be included *a priori*, say a factor of 10 of the stored template, but the techniques in essence are insensitive to the range of scale variations. This insensitivity to scale range may suit size invariance, since the range of legitimate size variations in spatial patterns is very large. However, natural rate variations occur only in a limited range, mostly within a factor of two (Klatt, 1976; Watson & Foyle, 1985; Port, 1990). So a blind treatment of rate invariance would entail undesired over-generalization of rate. For auditory patterns, there appear to be natural markers that can be used to detect the rate of the input flow. In speech, as described in Section 1.1, languages have been argued to exhibit a rhythmic structure, whether it be mora timing in Japanese or stress rhythms in English and other languages. In addition, models have recently been devised to detect rate changes in an input flow. The Tau net (Cottrell et al., 1993; see also Nguyen, 1995) uses a recurrent network to learn a template pattern, and then adapts the time constant of the network to match the template pattern with rate variations in the input flow. This is essentially similar to the size scaling method in Lades et al. (1993). This approach is computationally intensive when dealing with a large number of templates. Large and Kolen (1994) have proposed the idea of entraining oscillators with input rhythms of music meter (see also Large, 1994). A similar model of adaptive oscillators has been proposed by McAuley (1995) to account for human rate discrimination of a tone series. These adaptive oscillator models can quickly estimate the rate of a rhythmic flow, which is allowed to vary in real time.

Based on the above observations, we suggest the following approach for rate invariant recognition: explicitly estimate the rate of a temporal flow and then adapt recognition to the estimated rate. This approach does not exhibit the problem of rate over-generalization—the accepted range of variation is derived from the input itself, not imposed *a priori*. More specifically, the current rate estimated by an adaptive oscillator can be fed to a recognition model in real time to cue the recognition process to the current rate. One way to incorporate rate information is to adapt the time constant of a recurrent network to the period of an adaptive oscillator. This proposal builds on both adaptive oscillator models and the Tau net, and its main advantage lies in eliminating the need to search through all possible rate variations, which is computationally intensive. Note that this approach does not have to couple with simple recurrent networks, and may apply to other models of temporal pattern recognition.

It is interesting to look at rate invariance from the perspective of complexity of formal languages. If we discretize a component with a certain duration into multiple occurrences of that component with each component having a constant interval, then the duration corresponds to the number of the consecutive and repeated components. Thus, the sequence *S*: *A-A-A-B-B-C-C-C-C* corresponds to the sequence that is composed of three components, i.e., *A*, *B*, and *C*; the duration of *A* is three time steps, *B* two time steps, and *C* four time steps. We may use a superscript to indicate the number of repetitions of the same component so that the sequence *S* is represented as $A^3B^2C^4$. In this representation, an arbitrary sequence with arbitrary durations can be represented as $p_1^{i_1} - p_2^{i_2} - \ldots - p_n^{i_n}$, where $n, i_1, i_2, \ldots, i_n$ are positive numbers. Coming back to our notation, all interval variations to a sequence that contains the sequential components of $p_1, p_2, \ldots, p_n$ constitute the language $L_1 : \{p_1^{i_1} - p_2^{i_2} - \ldots -p_n^{i_n} | i_1 > 0, i_2 > 0, \ldots, i_n > 0\}$. It can be easily shown that L_1 is a regular language (Hopcroft & Ullman, 1979). On the other hand, all rate variations to the sequence $p_1, p_2 - \ldots - p_n$ constitute the language $L_2 : \{p_1^i - p_2^i - \ldots - p_n^i | i > 0\}$. It is well known that L_2 is a context-free language if $n = 2$, and is a context-sensitive language if $n \geqslant 3$ (Hopcroft & Ullman, 1979). Theoretical computer science has shown that to recognize context-sensitive languages demands computing devices with higher magnitudes of complexity than those required to recognize regular languages. The above interpretation shows from a different perspective the difficulty of recognizing temporal patterns with rate invariance as opposed to interval invariance. Our interpretation is consistent with a recent study of recurrent networks by Kolen (1994). On the other hand, our previous discussion regarding the range of rate variations suggests that it is not to compute the complete language of L_2 that we should pursue, but to compute a subset of L_2 which fits the plausible range of rate invariance.

To conclude, simple recurrent networks in the present form do not show proper temporal generalization. In particular, they are incapable of rate invariance in temporal pattern recognition. Our conclusion is derived by a careful investigation involving standard and extended architectures and using both abstract sequences of various complexities and real speech data. We point out that the inability to handle rate invariance is not limited to just simple recurrent networks, but common to all neural networks that have been proposed to process temporal patterns (see also Wang, 1995). Theoretical computer science suggests why this is a difficult problem: it is difficult to embed linear-bounded automata in neural networks. A proposal to tackle rate invariance is provided that combines rate estimation and recognition at the estimated rate. Given the fundamental importance of rate invariance both for explaining psychological data and for engineering applications, future neural network research must address this issue.

REFERENCES

Abu-Bakar, M., & Chater, N. (1993). Processing time warped sequences using recurrent neural networks: modeling rate dependent factors in speech perception. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* (pp. 191–196).

Bartlett, J. C., & Dowling, W. J. (1980). Recognition of transposed melodies: a key distance effect in developmental perspective. *Experimental Psychology: Human Perception and Performance*, **6**, 501–515.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, **5**(2), 157–166.

Bersini, H., Saerens, M., & Sotelino, L. G. (1994). Hopfield net generation, encoding and classification of temporal trajectories. *IEEE Transactions on Neural Networks*, **5**(6), 945–953.

Bourlard, H. A., & Morgan, N. (1994). *Connectionist speech recognition: A hybrid approach*. Norwell, MA: Kluwer Academic.

Cleeremans, A. (1993). *Mechanisms of implicit learning*. Cambridge, MA: MIT Press.

Cleeremans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, **1**, 372–381.

Cottrell, G. W., Nguyen, M., & Tsung, F.-S. (1993). Tau net: The way to do is to be. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* (pp. 365–370).

Durbin, R., & Willshaw, D. (1987). An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, **326**, 689–691.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, **14**, 179–211.

Elman, J. L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, **48**, 71–99.

Espinoza-Varas, B., & Watson, C. (1986). Temporal discrimination for single components of nonspeech auditory patterns. *Journal of the Acoustical Society of America*, **80**(6), 1685–1694.

Han, M. S. (1994). Acoustic manifestations of mora timing in Japanese. *Journal of the Acoustical Society of America*, **96**(1), 73–82.

Handel, S. (1989). *Listening: An introduction to the perception of auditory events*. Cambridge, MA: MIT Press.

Hanes, M. D., Ahalt, S. C., & Krishnamurthy, A. K. (1994). Acoustic to phonetic mapping using recurrent neural networks. *IEEE Transactions on Neural Networks*, **5**(4), 659–662.

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation*. Reading, MA: Addison-Wesley.

Jones, M. R., & Boltz, M . (1989). Dynamic attending and responses to time. *Psychological Review*, **96**, 459–491.

Jones, M. R., & Ralston, J. T. (1991). Some influences of accent structure on melody recognition. *Memory & Cognition*, **19**, 8–20.

Jordan, M. J. (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).

Kidd, G. R., & Watson, C. S. (1988). Detection of changes in frequency- and time-transposed auditory patterns. *Journal of the Acoustical Society of America*, **84**, S141–S142.

Kidd, G. R., & Watson, C. S. (1992). The "proportion-of-total-duration-rule" for the discrimination of auditory patterns. *Journal of the Acoustical Society of America*, **92**(6), 3109–3118.

Klatt, D. H. (1976). Linguistic uses of segmental duration in English: Acoustic and perceptual evidence. *Journal of the Acoustical Society of America*, **59**, 1208–1221.

Kolen, J. F. (1994). *Exploring the computational capabilities of recurrent neural networks*. PhD dissertation, Department of Computer and Information Science, The Ohio State University, Columbus, OH.

Lades, M. et al. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, **42**(3), 300–311.

Large, E. (1994). *Dynamic representation of musical structure*. PhD dissertation, Department of Computer and Information Science, The Ohio State University, Columbus, OH.

Large, E., & Kolen, J. F. (1994). Resonance and the perception of musical meter. *Connection Science*, **6**, 177–208.

Levelt, W. J. M. (1989). *Speaking: from intention to articulation*. Cambridge, MA: MIT Press.

Lippmann, R. P. (1989). Review of neural networks for speech recognition. *Neural Computation*, **1**, 1–38.

Massone, L., & Bizzi, E. (1989). Generation of limb trajectories with a sequential network. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 2, pp. 345–349).

McAuley, J. D. (1995). *Perception of time as phase: Toward an adaptive-oscillator model of rhythmic pattern processing*. PhD dissertation, Department of Computer Science, Indiana University, Bloomington, IN.

Nguyen, M. H. T. (1995). *Modeling dynamic signals with neural networks*. PhD dissertation, Department of Computer Science and Engineering, University of California, San Diego, CA.

Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, **1**(2), 263–299.

Perantonis, S. J., & Lisboa, P. J. G. (1992). Translation, rotation, and scale invariant pattern recognition by high-order neural networks and moment classifiers. *IEEE Transactions on Neural Networks*, **3**(2), 241–251.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, **59**(19), 2229–2232.

Pollack, J. B. (1991). The induction of dynamic recognizers. *Machine Learning*, **7**, 227–252.

Port, R. F. (1990). Representation and recognition of temporal patterns. *Connection Science*, **2**, 151–176.

Port, R. F., & Dalby, J. (1982). Consonant/vowel ratio as a cue for voicing in English. *Perception & Psychophysics*, **32**, 141–152.

Port, R. F., Dalby, J., & O'Dell, M. (1987). Evidence of mora timing in Japanese. *Journal of the Acoustical Society of America*, **81**, 1574–1585.

Port, R. F., Cummins, F., & Gasser, M. (1996). A dynamic approach to rhythm in language: Toward a temporal phonology. In *Proceedings of the Chicago Linguistic Society*, to appear, Department of Linguistics, University of Chicago.

Reitboeck, H. J., & Altmann, J. (1984). A model for size- and rotation-invariant pattern processing in the visual system. *Biological Cybernetics*, **51**, 113–121.

Simard, P. Y., LeCun, Y., & Denker, J. (1993). Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems* (NIPS-93), 50–58.

Sloboda, J. A. (1985). *The musical mind*. Oxford: Clarendon.

Tank, D. W., & Hopfield, J. J. (1987). Neural computation by concentrating information in time. *Proceedings of the National Academy of Sciences of USA*, **84**, 1896–1900.

Unnikrishnan, K. P., Hopfield, J. J., & Tank, D. W. (1992). Speaker-independent digit recognition using a neural network with time-delayed connections. *Neural Computation*, **4**, 108–119.

Wang, D. L. (1995). Temporal pattern processing. In M. A. Arbib (Ed.), *Handbook of brain theory and neural networks* (pp. 967–971). Cambridge, MA: MIT Press.

Wang, D. L., & Arbib, M. A. (1990). Complex temporal sequence learning based on short-term memory. *Proceedings of IEEE*, **78**, 1536–1543.

Wang, D. L., & Arbib, M. A. (1993). Timing and chunking in processing temporal order. *IEEE Transactions on Systems, Man, and Cybernetics*, **23**(4), 993–1009.

Wang, D. L., & Yuwono, B. (1995). Anticipation-based temporal pattern generation. *IEEE Transactions on Systems, Man, and Cybernetics*, **25**, 615–628.

Watrous, R. L., Ladendorf, B., & Kuhn, G. (1990). Complete gradient optimization of a recurrent network applied to /b/, /d/, /g/ discrimination. *Journal of the Acoustical Society of America*, **87**(3), 1301–1309.

Watson, C. S., & Foyle, D. C. (1985). Central factors in the discrimination and identification of complex sounds. *Journal of the Acoustical Society of America*, **78**(1), 375–380.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, **1**, 270–280.