Recognition of Melody Fragments in Continuously Performed Music

Robert Port and Svĕn Anderson
Department of Linguistics, Department of Computer Science
Indiana University, Bloomington, Indiana 47405

Abstract

The processing of continuous acoustic signals is a challenging problem for perceptual and cognitive models. Sound patterns are usually handled by dividing the signal into long fixed-length windows—long enough for the longest patterns to be recognized. We demonstrate a technique for recognizing acoustic patterns with a network that runs continuously in time and is fed a single spectral frame of input per network cycle. Behavior of the network in time is controlled by temporal regularities in the input patterns that allow the network to predict future events.

INTRODUCTION

An important step in the application of neural networks to perceptual problems is performance that runs continuously in time. Relevant perceptual tasks include (1) labelling of sequential patterns and (2) the prediction of stimulus events. Only recently have attempts been made to experiment with networks that handle sequential inputs [Elman, 1988,Gallant and King, 1988]. Achievement of these goals must be attempted in the context of several additional constraints. First, the network should receive only a single frame of input at a time. This implies that the system must construct a temporary memory of past inputs or reach a distinctive state that allows recognition of patterns that extend over a number of input frames. The employment of a long static input window (cf., [Elman and Zipser, 1988, Prager et al., 1986]) is an unsatisfactory solution for biological perceptual systems since response time must then be delayed to the end of the window, preventing optimal reaction time [Port et al., 1988]. A second constraint is that the system must be prepared to deal with patterns distributed in time that partially resemble each other yet have varying durations. This constraint prevents use of a built-in restart signal to begin analysis from a common initial network state, such as at the beginning of each perceptual trial. External reset, as found in unfolded networks ([Elman and McClelland, 1986]), fixed window networks ([Elman and Zipser, 1988]) or fixed length dynamic windows greatly reduces the biological plausibility of a system. Some means for resetting the system that can be partly controlled from the bottom up is required (cf., [Grossberg, 1980]).

We have been designing network architectures with a limited number of recurrent connections (inspired by [Jordan, 1986]) for processing continuous input signals, with the intention of eventually handling speech. We demonstrate in this paper that a modification of Real-Time Backpropagation [Williams and Zipser, 1988] permits generalization of the sequential network in interesting ways. In the simulations described below, we employ recurrent loops of nodes to store information about the history of the signal by training the system to predict the next input. This contrasts with memory that is either a record of previous inputs (as in delay-line systems [Waibel et al., 1988]) or a record of previous node activations ([Jordan, 1986,Elman, 1988,Anderson et al., 1988]). To support reset of the network when a partially recognized pattern is to be rejected, we use sigma-pi nodes [Rumelhart et al., 1986].[1] These allow the activation of nodes in one clique to gate the activity of another clique. In experiments with this system, we have attempted recognition of melodic patterns produced by hand on a keyboard instrument. In this way, we deal with some forms of natural variation and demonstrate the robustness of our system without tackling the magnitude of variability observed in speech.

DYNAMIC GRADIENT DESCENT FOR SIGMA-PI UNITS

Willaims & Zipser [1988] derive a gradient descent alogorithm for computing weight changes in continuously running recurrent networks. The technique requires maintaining a matrix of activation partials that is updated at each time step. In this section we extend the gradient descent algorithm to networks that contain modules composed of sigma-pi units. These are units that receive weighted activation products from 2 or more other units.

Suppose that the network contains m inputs and n nodes. Let I index the set of all external inputs, and N index each node in the network. We can then label input activations using the variable x and node activations with the variable y. It is straightforward to write the input to the node indexed k at a time t.

The input to a sigma-pi unit k can be written:

$$a_k(t) = \sum_{l \in I} x_l(t) + 1/2 \sum_{s \in N} \sum_{r \in N} w_{ksr} y_s(t) y_r(t)$$

Note that sigma-pi nodes simplify to the case of sigma nodes in the case where $y_r(t) \equiv 1 \ \forall t$, so that this derivation applies as well to combined modules of sigma and sigma-pi nodes. For clarity we have assumed that the external inputs k are weighted by a constant 1. The activation of node k at the next time step is

$$y_k(t+1) = \phi_k(a_k(t))$$

where ϕ_k is a C^1 function. If we know what the output values should be for some group of nodes at time τ, then we can define an error function for the network's performance using the sum-squares metric.

$$E(\tau) = 1/2 \sum_{k \in N} (e_k(\tau))^2$$

[1]Williams [1986] has shown that sigma-pi connections can compute any monotonic function of their inputs. They have been used to generate and maintain simple rhythms by Dehaene et al. [Dehaene et al., 1987].

where the error for output node k (e_k) is the difference between the desired value and the actual output for node k.

$$e_k(t) = \begin{cases} d_k(t) - y_k(t) & k \in T(t) \\ 0 & \text{else} \end{cases}$$

where the set $T(t)$ is the (possibly time varying) set of indices for nodes for which there exist desired values at time t.

Because the node activations vary at each time step, we wish to minimize the total error over a time frame beginning at t_i and ending at t_f by changing the weights. This implies changing the weights in the direction of the negative gradient of the total error with respect to the weights. The total error over this time frame is the sum of the network error at each time frame.

$$E_{total}(t_i, t_f) = \sum_{\tau=t_i+1}^{t_f} E(\tau)$$

Since

$$E_{total}(t_0, t_2) = E_{total}(t_0, t_1) + E_{total}(t_1 + t_2)$$

induction yields

$$\nabla_W E_{total}(t_i, t_f) = \sum_{\tau=t_i+1}^{t_f} \nabla_W E_{total}(\tau)$$

This means that the error at each time point in the time frame can be calculated and these values summed to yield the total gradient. At the end of the time frame, the ijth weight should be adjusted by

$$\Delta w_{hij}(t) = -\alpha \frac{\partial E(t)}{\partial w_{hij}} \quad h, i, j \in N$$

But

$$-\frac{\partial E(t)}{\partial w_{hij}} = \frac{\partial}{\partial w_{hij}} \left[1/2 \sum_{k \in N} (e_k(\tau))^2 \right] = \sum_{k \in N} \frac{\partial y_k(t)}{\partial w_{hij}} e_k(t)$$

Thus, computing the changes to be made to the weights boils down to calculating $\frac{\partial y_k}{\partial w_{hij}}$.

$$\frac{\partial y_k(t+1)}{\partial w_{hij}} = \frac{\partial}{\partial w_{hij}} \phi_k(a_k(t)) = \phi_k\prime(a_k(t)) \frac{\partial a_k(t)}{\partial w_{hij}} \tag{1}$$

and then

$$\frac{\partial a_k(t)}{\partial w_{hij}} = \sum_{s \in N} \sum_{r \in N} \left[1/2 \delta_{kh} y_i(t) y_j(t) + w_{ksr} y_r \frac{\partial y_s(t)}{\partial w_{hij}} \right]$$

where δ is 1 when $k = h$ and otherwise 0. At this point we have an iterative solution to the gradient that allows computation of future $\frac{\partial y_k}{\partial w_{hij}}$ if we assume that the initial partials are zero. At each time step this matrix of partials must be updated according to

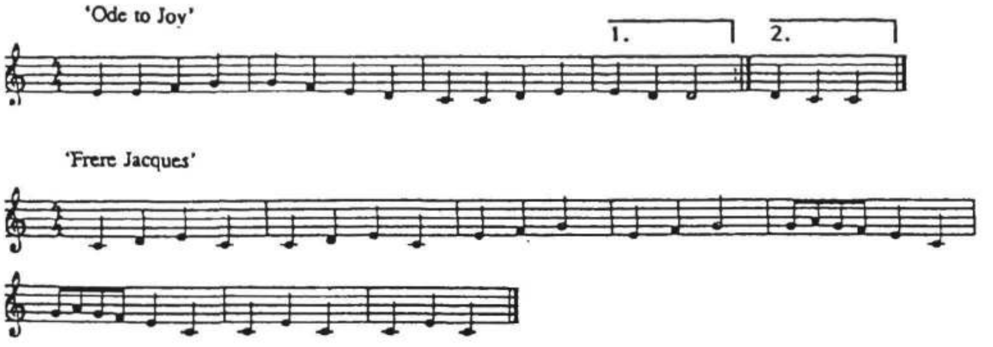

Figure 1: The two melodies used in the experiment are slightly edited familiar tunes. Note that measures 1 and 5 of 'Ode to Joy' are similar to measures 3 and 4 of 'Frere Jacques'

equation 1. Because the computation of the partials necessary for the learning algorithm scales as n^4, in actual implementations it is important to severely limit the number of sigma-pi units used.

In the simulations reported here weight updates were carried out at each time step. Output activations for supervised nodes were set to their desired values after each time step and the corresponding elements of the matrix of partials was set to zero (cf. [Williams & Zipser, 1988]).

MELODY RECOGNITION EXPERIMENT

We applied this architecture and algorithm to a problem in auditory sequence recognition. We recorded productions of two melodies on an electronic keyboard and trained a network to recognize two measures selected from the melodies. The measures have the same note sequence, differing only in the rhythm of the note patterns. This task requires robustness across natural temporal variability due to human performance while still restricting the difficulty of frequency discrimination required through use of the keyboard.

Stimuli

One of the experimenters produced 2 repetitions of the two familiar 8-measure melodies (slightly edited) shown in Figure 1. The melodies were played in the key of C, using a 6-note scale from C_4 (261 Hz) to A_4 (440 Hz) at a tempo of 132 beats/minute on a Casio Model C-140 keyboard. The tempo of performance was stabilized by use of a visual metronome and the performance was done in a staccato style to minimize note overlap. Statistical measures on the productions showed that the mean duration of each measure was 452 ms (SD=11 ms). There was a total of 10 different measure types. The training corpus was restricted to all of the measures from 2 of the repetitions; target measures in the 3 different repetitions of the melodies were used as testing data. Based on the duration of the shortest measure, a 1024 point fast-Fourier transform (FFT) was performed with Hamming window spacing that provides 8 equally spaced FFT frames for the shortest measure (that is, approximately one per 1/8th note). The windows were 64 ms wide with their centers fixed 48 ms apart. Figure 4 shows 8 successive FFT frames for

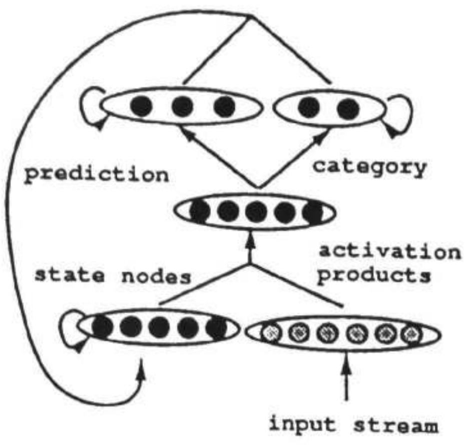

Figure 2: The architecture of the network employed.

two sample measures. Six frequency bins were centered approximately over each note in the 6-note scale that we employed. The highest 40 dB of dynamic range was retained for each bin; all values more than 40 dB down were replaced with 0. The dB values were then mapped linearly onto the range [0,1] which served as input to the network.

Procedure

For our experiments, the first measure of tune 'Ode to Joy' and the third measure of 'Frere Jacques' were selected as targets A and B respectively. This provides a challenging temporal discrimination task for the network in contrast to the easier task of differentiating each target from the other measures. The network was presented with a continuous random sequence of measures. As can be seen in Figure 4, the productions exhibited many variations that reduced the quality of the inputs. In order to be able to test on different productions of the target measure than were employed in the training, only half of the 10 instances of the target measure were used. The other 5 were employed only in the testing phase.

The network was as shown in Figure 2 with 6 input nodes, 5 hidden nodes, 5 output nodes (2 of which were trained to the category labels, and 3 trained to predict the subsequent input) and 4 state nodes that received connections from the prediction and category nodes. Connections to nodes on the hidden layer paired the input and state node cliques. These sigma-pi nodes weighted the product of the activation pairs as input. All weights were learned using real-time back propagation. The target function for the category node was a monotonic ramp from 0 to 1 during the presentation of the target measure. The prediction node targets rose to 1 and fell to 0.1 at appropriate times during each target measure.

Results

After training, the network was able to correctly identify occurrences of the target syllables in the input stream. Percent correct identification by itself, however, is not a suitable statistic for performance since both missed targets and 'false alarms' are possible

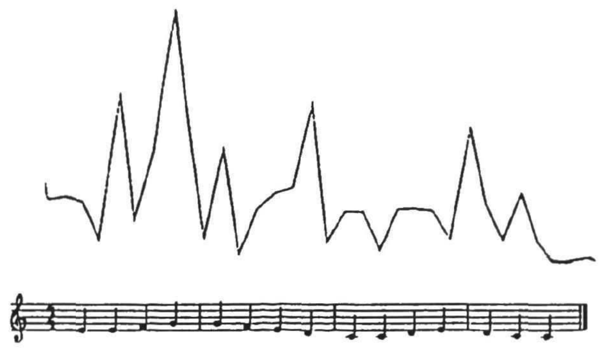

Figure 3: The output activation of the category node trained to recognize measure 1 during presentation of measures 1-4 of 'Ode to Joy'. The activation begins to climb every time an E (the first note in the target measure) occurs in the input, but it quickly falls for false positives.

errors. Reducing the frequency of targets missed invariably increases the number of false alarms. The statistic d' from signal detection theory is designed to provide a performance score that is independent of response criterion and which is thus a robust discriminability measure [Swets, 1961]. The units of d' represent z-scores along a hypothetical discriminant continuum. Thus a d' of 2 means that the mean values of the likelihood of 'target present' and 'target not present' are 2 standard deviations apart.

We apply d' here to evaluate two properties of performance: first, the ability of the network to differentiate each target measure from the measures in the data that do not include either target, and, second, to differentiate the two targets from each other. The second task should be more difficult. Figure 3 shows the output of the category node for Target A during presentation of 4 measures of the first tune. A threshold activation level was defined on the activity of the category nodes and each measure from a sample was classified as either a hit (correct labelling of the pattern), a false alarm (where the activation exceeded the threshhold during the measure but there was no target), a miss (target occurred but response was below threshhold) or a correct rejection (no target, no response).

When the test tokens of Target A were differentiated from nontarget measures ($n = 116$ measures), $d' = 2.03$. This value means, for example, that, if the ratio of misses to false alarms is 1, there will be about 85% correct identification of the target against about 15% false alarms. For test tokens of Target B versus all nontargets, d' was somewhat better, 2.75 (with $n = 96$). This implies about 91% correct identification and 91% correct rejection.

The most difficult discrimination, however, is between the two similar measures, Targets A and B. As mentioned above, the system was trained on a set of 10 tokens of each measure. When it is tested on two new sets of tokens, $d' = 1.64$ ($n = 87$),illustrating poorer discriminability for the two similar target measures. This implies about 80% correct with 20% false alarms. By testing the ability of the network to identify the same

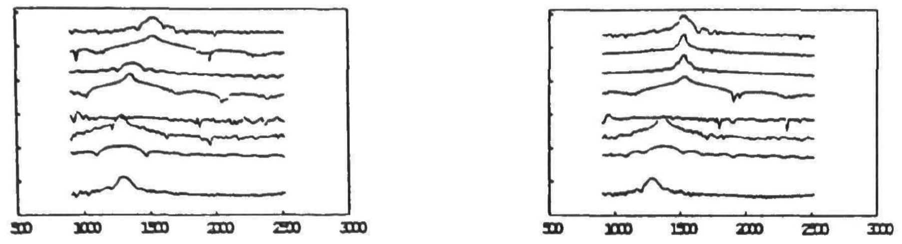

Figure 4: The FFT sequence for one token of each of the two targets for Experiment 2. Frequency increases from left to right, and time-ordering from bottom to top. The left-hand panel is measure 1 of 'Ode to Joy' while the right-hand panel is measure 3 of 'Frere Jacques'.

tokens it was trained on, we can evaluate the robustness of the system to the kinds of natural noise that existed in the data. Here $d' = 1.84$ ($n = 69$), not too much better than on the novel tokens. This suggests that our system did not greatly overlearn the training data and was able to generalize effectively.

CONCLUSIONS

This network is able to recognize real auditory patterns that were produced with characteristic human inaccuracy. The system 'listens' continuously, without an *a priori* input window other than the minimum time frame for the spectral analysis. This input sampling rate was also the rate of internode communication within the network itself, but this 'clock' does not control the point of reset of the network. Instead, we allowed periodicity in the input pattern to entrain the output behavior of the network. Then the targets themselves control the resetting function.

These perception experiments using acoustic input demonstrate that it is possible to use sparsely connected recurrent networks to continuously monitor inputs for particular patterns. This was achieved by use of a training technique that emphasized prediction of the next input signal and sigma-pi nodes to support rapid reset when a partial pattern is violated. This distributes control of the system toward the lowest levels—near the input—rather than having it externally specified by the design of the system. Apparently this system is robust enough to handle considerable natural variation in inputs. This is a step toward a continuously functioning dynamically controlled perceptual system for auditory inputs.

Acknowledements

We are grateful to Chan-Do Lee and William Mills for assistance with this project. This research was supported in part by the National Science Foundation, Grants DCR-8505635 and DCR-8518725.

References

[Anderson et al., 1988] Anderson, S., Merrill, J., and Port, R. (1988). Dynamic speech categorization with recurrent networks. In *Proceedings of the 1988 Connectionist Summer School*, pages 398–406. Morgan-Kauffmann, San Mateo, California.

[Dehaene et al., 1987] Dehaene, S., Changeux, J.-P., and Nadal, J.-P. (1987). Neural networks that learn temporal sequences by selection. In *Proceedings of the National Academy of Science*, volume 84, page 2727. National Academy of Science.

[Elman, 1988] Elman, J. (1988). Finding structure in time. Technical Report 8801, Center for Research in Language, University of California at San Diego, La Jolla, CA.

[Elman and Zipser, 1988] Elman, J. and Zipser, D. (1988). Learning the hidden structure of speech. *Journal of the Acoustical Society of America*, 83:1615–26.

[Elman and McClelland, 1986] Elman, J. L. and McClelland, J. L. (1986). Exploiting the lawful variability in the speech wave. In Perkell, J. S. and Klatt, D. H., editors, *Invariance and Variability of Speech Processes*, chapter 17, pages 360–381. Erlbaum, Hillsdale, NJ.

[Gallant and King, 1988] Gallant, S. I. and King, D. J. (1988). Experiments with sequential associative memories. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. L. Erlbaum, Hillsdale, NJ.

[Grossberg, 1980] Grossberg, S. (1980). How does a breain build a cognitive code? *Psychological Review*, 87(1):1–51.

[Jordan, 1986] Jordan, M. (1986). Serial order. Technical Report 8604, Institute for Cognitive Science, U. of California at SanDiego, La Jolla, CA.

[Port et al., 1988] Port, R., Anderson, S., and Merrill, J. (1988). Temporal information and memory in connectionist networks. Technical Report 265, Indiana University Computer Science Department.

[Prager et al., 1986] Prager, I., Harrison, T. D., and Fallside, F. (1986). Boltzmann machines for speech recognition. *Computer Speech and Language*, 1:1–20.

[Rumelhart et al., 1986] Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In Rumelhart, D. and McClelland, J., editors, *Parallel Distributed Processing*, volume 1. MIT Press, Cambridge, Massachusetts.

[Swets, 1961] Swets, J. A. (1961). Is there a sensory threshold? *Science*, 34:168–177.

[Waibel et al., 1988] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1988). Phoneme recognition: Neural networks vs. hidden markov models. In *Proceedings of the ICASSP*, pages 107–110. IEEE.

[Williams and Zipser, 1988] Williams, R. and Zipser, D. (1988). A learning algorithm for continually running fully recurrent neural networks. Technical Report 8805, ICS, UCSD, La Jolla, CA.

[Williams, 1986] Williams, R. J. (1986). The logic of activation functions. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing, Vol. 1*, chapter 4, pages 423–443. Mit Press, Cambridge, Massachusetts.