A TIME WARPING NEURAL NETWORK

Earl Levine

Department of Electrical Engineering, Stanford University
Stanford CA 94305 USA
earl@isl.stanford.edu

ABSTRACT

A method is proposed to improve any temporal pattern recognition system by time warping each pattern before presentation to the recognition system. The time warping function for a pattern is generated by repeated local application of a neural network to sections of the pattern. The output of this neural network is the slope of the warping function, and the internal weight parameters are trained by a gradient descent learning rule which attempts to minimize the recognition system's error. Experimental results show that this method can improve recognition of vowel phonemes.

1. INTRODUCTION

In certain temporal pattern recognition problems such as speech recognition, part of the variability within a class of patterns is due to time-compression or time-expansion of the pattern during production. This compression or expansion is not necessarily uniform across the pattern, that is, some parts of the pattern may be compressed in time—"speeded up"—and some parts may be expanded in time—"slowed down". A recognition method which counteractively compresses and expands parts of the pattern during recognition, a process known as *time warping*, will supposedly improve pattern recognition by reducing pattern variability within each class.

Time warping approaches are often used for speech recognition tasks. Examples are dynamic programming and hidden Markov models[1]. In this paper, I propose a technique which will first time warp the input, and then present it to a recognition system. Unlike other methods, the time warping neural network will learn to optimize its warping to minimize the error incurred at the output of the recognizer. The only requirement of the recognition system is that it be able to provide the error gradient at its pattern inputs.

2. TIME WARPING FOR IMPROVED RECOGNITION

A recognition system has $K+1$ vector inputs $\mathbf{o}_k$, $k = 0, 1, 2, \dots K$, each with M scalar components $\mathbf{o}_{k,1}$, $\mathbf{o}_{k,2}$, $\dots \mathbf{o}_{k,M}$. The data available to those inputs is a vector-valued function of time $\mathbf{i}(t)$ with M continuous component functions $\mathbf{i}_1(t), \mathbf{i}_2(t), \dots \mathbf{i}_M(t)$. Each input $\mathbf{o}_k$ will sample $\mathbf{i}(t)$ at a different moment in time. A uniform sampling method will cause the time between consecutive samples to be a uniform positive value T, that is,

$$\mathbf{o}_k = \mathbf{i}\left(T_{start} + kT\right), k = 0, 1, 2, \dots K \tag{1}$$

where T_{start} is the time of the first sample. Using the samples of $\mathbf{i}(t)$ as input, the recognition system generates an output result, which is in error by an amount ε. The error is a function of the inputs [1] $\mathbf{o}_k$,

$$\varepsilon = F_\varepsilon\left(\mathbf{o}_0, \mathbf{o}_1, \dots \mathbf{o}_K\right) \tag{2}$$

The sampling method could be improved for lower error. Suppose there is a strictly increasing continuous function of time $\Upsilon(t)$, known as the *time warping function*, which is used to determine the sample times by

$$\mathbf{o}_k = \mathbf{i}\left(\Upsilon\left(T_{start} + kT\right)\right), k = 0, 1, 2, \dots K. \tag{3}$$

Note that uniform sampling will occur for the special case $\Upsilon(t) = t$. If, for a particular pattern, a time warping function which minimizes the recognizer's error can be found, then using that warping function (as in Equation 3) is preferable to uniform sampling (as in Equation 1). The goal of the proposed system, after having been trained, is to generate for each pattern a time warping function which is expected to minimize the recognizer error.

3. A TIME WARPING NEURAL NETWORK STRUCTURE

Figure 1 shows a neural network structure which examines the data $\mathbf{i}(t)$ and then time warps it before presentation to the recognition system. Let τ_k be the sample time of $\mathbf{o}_k$, that is,

$$\mathbf{o}_k = \mathbf{i}(\tau_k). \tag{4}$$

Now the values of τ_k represent the time warping function,

$$\tau_k = \Upsilon\left(T_{start} + kT\right). \tag{5}$$

As a starting point, let τ_0 be a predetermined constant. Let τ_{k+1} be determined from τ_k in the following way: a feedforward neural network "placed" at τ_k takes samples of $\mathbf{i}(t)$ at N times relative to τ_k, specifically, at times $\tau_k + t_1, \tau_k + t_2, \dots \tau_k + t_N$. Note that the t_n values, which define

[1] Of course, the error also depends on the desired "target" value of the recognizer output for this $\mathbf{i}(t)$, as well as internal parameters of the recognizer. However this discussion is only concerned with the effect of the $\mathbf{o}_k$ on ε, so these other values are taken as constants.

0-7803-2431-5/95 $4.00 © 1995 IEEE

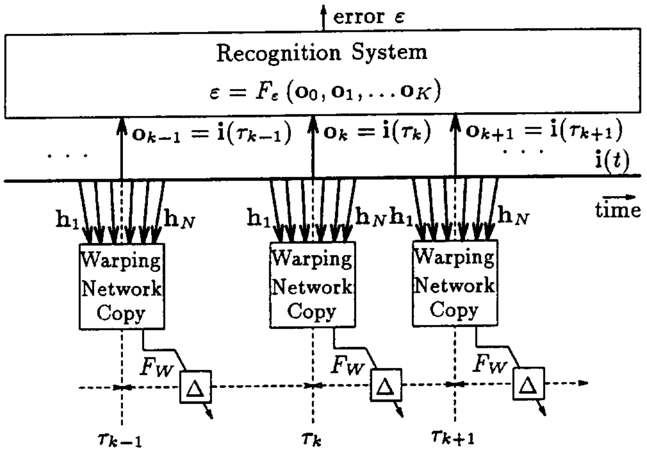
error ε
Recognition System
$\varepsilon = F_\varepsilon(\mathbf{o}_0, \mathbf{o}_1, \ldots \mathbf{o}_K)$
$\mathbf{o}_{k-1} = \mathbf{i}(\tau_{k-1})$
$\mathbf{o}_k = \mathbf{i}(\tau_k)$
$\mathbf{o}_{k+1} = \mathbf{i}(\tau_{k+1})$
$\mathbf{i}(t)$
time
$\mathbf{h}_1$
$\mathbf{h}_N$
Warping Network Copy
F_W
Δ
τ_{k-1}
τ_k
τ_{k+1}

Figure 1: Time Warping Neural Network Structure

an "input window" about τ_k, are fixed but can be negative, zero, or positive. The neural network produces one positive-valued output using its internal weight parameters, represented by the vector $\mathbf{w}$, and the N vector samples of $\mathbf{i}(t)$. This output is added to τ_k to give τ_{k+1}. Then another copy of the network with identical weights $\mathbf{w}$ is placed at τ_{k+1}, and the process is repeated to determine τ_{k+2} from τ_{k+1} using the N samples of $\mathbf{i}(t)$ relative to τ_{k+1}, and so on. Using the function $F_W()$ to represent the output of the neural network, if $0 \le k \le K-1$,

$$\tau_{k+1} = \tau_k + [F_W(\mathbf{w}, \mathbf{h}_1, \mathbf{h}_2, \ldots \mathbf{h}_N)]|_{\mathbf{h}_n = \mathbf{i}(\tau_k + t_n) \text{ for } n=1,2,\ldots N}. \tag{6}$$

By sampling in a time window (determined by the t_n values) about τ_k, it is hoped that the neural network can make a judgment about the speed of production of $\mathbf{i}(t)$ at $t = \tau_k$. From such knowledge, an appropriate interval to the next sample should be selected. It is reasonable to restrict the next sample from being placed farther in the future than $\tau_k + \max_n(t_n)$, since the network should only be able to select a next-sample which it can "see" in its time window of input. This can be easily accomplished by selecting values of t_n and selecting an activation function for the output such that $0 < F_W() < \max_n(t_n)$.

4. A GRADIENT DESCENT LEARNING RULE

The time warping neural network's weight parameters $\mathbf{w}$ need to be established by a training procedure with a goal to reduce the recognition error. An iterative gradient descent algorithm can be used. An initial weight vector is selected[2] (for example, randomly) and then the weights are iteratively adjusted along the error gradient by $\mathbf{w}_{next} = \mathbf{w}_{current} - \mu \frac{\partial \varepsilon}{\partial \mathbf{w}_{current}}$, where μ is a positive constant known as the *learning rate*.

[2]Upon initialization of the training procedure, it is desirable for the system to behave nearly as if no warping is taking place, that is, $F_W \approx T$. This can be achieved by appropriate choice of output activation function and initialization of weights.

The error gradient $\frac{\partial \varepsilon}{\partial \mathbf{w}}$ needs to be found. From Equation 2,

$$\frac{\partial \varepsilon}{\partial \mathbf{w}} = \sum_{k=0}^{K} \sum_{m=1}^{M} \frac{\partial F_\varepsilon(\mathbf{o}_0, \mathbf{o}_1, \ldots \mathbf{o}_K)}{\partial \mathbf{o}_{k,m}} \frac{\partial \mathbf{o}_{k,m}}{\partial \mathbf{w}}. \tag{7}$$

Any recognition system from which $\frac{\partial F_\varepsilon()}{\partial \mathbf{o}_{k,m}}$ can be determined may be used for this training procedure. If the recognition system is a feedforward neural network, for example, those values can be determined by backpropagation[2].

Assuming that the $\frac{\partial F_\varepsilon()}{\partial \mathbf{o}_{k,m}}$ values are available, the next step is to determine the values of $\frac{\partial \mathbf{o}_{k,m}}{\partial \mathbf{w}}$. From Equation 4,

$$\frac{\partial \mathbf{o}_{k,m}}{\partial \mathbf{w}} = \left[\frac{\partial \mathbf{i}_m(t)}{\partial t}\right]\Bigg|_{t=\tau_k} \frac{\partial \tau_k}{\partial \mathbf{w}}. \tag{8}$$

The time derivative $\frac{\partial \mathbf{i}_m(t)}{\partial t}$ is available directly from $\mathbf{i}(t)$. Keeping in mind that τ_0 is constant, $\frac{\partial \tau_0}{\partial \mathbf{w}} = \mathbf{0}$. For nonzero values of k, it can be found from Equation 6 that

$$\frac{\partial \tau_{k+1}}{\partial \mathbf{w}} = \left[\frac{\partial F_W(\mathbf{w}, \mathbf{h}_1, \mathbf{h}_2, \ldots \mathbf{h}_N)}{\partial \mathbf{w}} + \frac{\partial \tau_k}{\partial \mathbf{w}} \left(1 + \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\partial F_W(\mathbf{w}, \mathbf{h}_1, \mathbf{h}_2, \ldots \mathbf{h}_N)}{\partial \mathbf{h}_{n,m}} \left[\frac{\partial \mathbf{i}_m(t)}{\partial t}\right]\Bigg|_{t=\tau_k+t_n}\right)\right]\Bigg|_{\mathbf{h}_n = \mathbf{i}(\tau_k+t_n) \text{ for } n=1,2,\ldots N}. \tag{9}$$

As before, the values of the time derivatives $\frac{\partial \mathbf{i}_m(t)}{\partial t}$ are available directly from $\mathbf{i}(t)$. For each copy k of the time warping network, the values of $\frac{\partial F_W()}{\partial \mathbf{w}}$ and $\frac{\partial F_W()}{\partial \mathbf{h}_{n,m}}$ can be determined by a variant of the backpropagation algorithm. Unlike "usual" backpropagation, the gradient of the output F_W (rather than the error) should be determined, and the gradient should be propagated all the way back to the neural network's inputs ($\mathbf{h}_n$). Once these values are determined, starting with $\frac{\partial \tau_0}{\partial \mathbf{w}} = \mathbf{0}$ and using Equation 9, all the values of $\frac{\partial \tau_k}{\partial \mathbf{w}}$ can be determined. Then the error gradient $\frac{\partial \varepsilon}{\partial \mathbf{w}}$ is found from Equations 7 and 8.

Although in practice, $\mathbf{i}(t)$ may be represented by discrete samples, it was assumed that $\mathbf{i}(t)$ is a continuous function of time, and this algorithm will require values of $\mathbf{i}(t)$ and $\frac{\partial \mathbf{i}(t)}{\partial t}$ at arbitrary times t. Determining these values exactly for each time t from the original samples of $\mathbf{i}(t)$ may be very computationally intensive. One possible compromise is to generate adequately oversampled versions of $\mathbf{i}(t)$ and $\frac{\partial \mathbf{i}(t)}{\partial t}$, and then each time a value is required, to select the oversampled value at the time closest to the actual t.

5. EXPERIMENTAL RESULTS

An experimental comparison of a recognition system with and without the addition of time warping neural networks of various sizes was made. The input data for the network were generated by preprocessing a subset of the DARPA

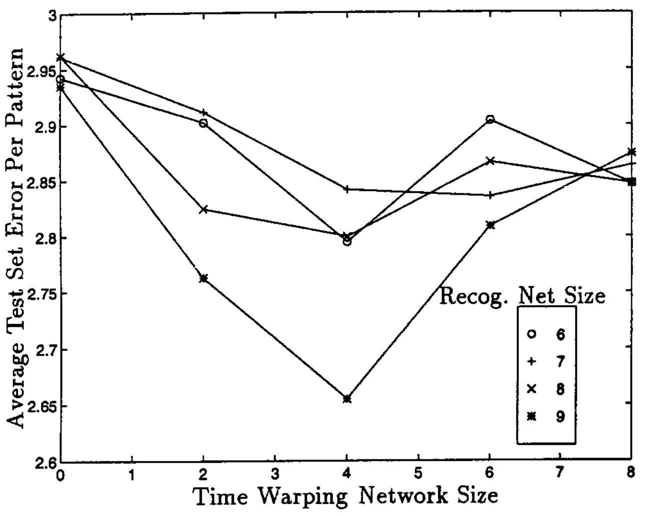

Figure 2: Recognition Error for Various Combinations of Recognition and Time Warping Network Sizes

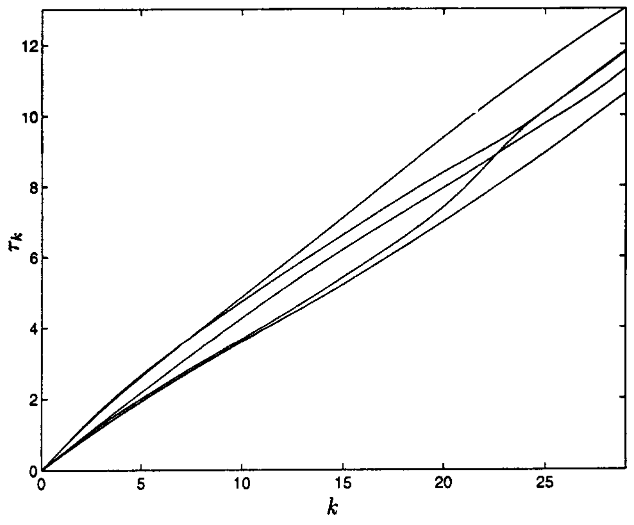

Figure 3: Time Warping Functions for Five Instances of a Phoneme

TIMIT speech database for 6 RASTA-PLP[5] features. The patterns selected for training and testing were members of 20 vowel phoneme classes spoken by multiple male and female speakers. Vowels are expected to benefit more than other phonemes from time warping, since they undergo more warping during production. The recognition system used was an integrated segmentation and recognition (ISR) neural network[3], a variant of the time-delay neural network[4]. The recognition system's error metric was the classification cross-entropy as in [3]. Different sizes of recognition networks were used, with the size varied by changing the number of unique hidden units but keeping all other aspects of the network structure constant. Furthermore, time warping networks of various sizes differed only in the number of hidden units. While each additional hidden unit in the recognition network added 241 unique weight and bias parameters, each additional hidden unit in the time warping network added only 32 parameters.

Figure 2 shows the results of training the recognition and time warping networks simultaneously for several combinations of sizes of the two networks. Note that a "size 0" time warping network corresponds to no time warping network, that is, a recognition network only. Each point represents the minimum test set error, average per pattern, during the course of a single training run; that error is expected to approximate the average over many training runs. Test set error represents the performance on novel data. Figure 2 appears to show a trend toward optimal recognition performance with a time warping network of a certain size.

Figure 3 shows time warping functions for five instances of the same phoneme (*ay*) for the best system of Figure 2. Indeed the network has learned to use diverse time warping functions for different instances within a class, supporting the contention that time warping works by reducing pattern variability within each class.

6. REFERENCES

[1] H. F. Silverman and D. P. Morgan, "The Application of Dynamic Programming to Connected Speech Recognition," *IEEE ASSP Magazine,* July 1990, pp. 7-24.

[2] D. Rumelhart and J. McClelland, *Parallel Distributed Processing,* MIT Press, 1986.

[3] D. Rumelhart, "Theory to Practice: A Case Study — Recognizing Cursive Handwriting," in E. B. Baum, editor, *Computational Learning and Cognition: Proceedings of the Third NEC Research Symposium,* Society for Industrial and Applied Mathematics, 1993, pp. 177-196.

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* March 1989, pp. 328-339.

[5] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP Speech Analysis Technique," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* 1992.