Tau Net: The way to do is to be

Garrison W. Cottrell Mai Nguyen Fu-Sheng Tsung

Department of Computer Science and Engineering
and
Institute for Neural Computation
University of California, San Diego

Abstract

We describe a technique for automatically adapting to the rate of an incoming signal. We first build a model of the signal using a recurrent network trained to predict the input at some delay, for a "typical" rate of the signal. Then, fixing the weights of this network, we adapt the time constant τ of the network using gradient descent, adapting the delay appropriately as well. We have found that on simple signals, the network adapts rapidly to new inputs varying in rate from twice as fast as the original signal, down to ten times as slow. So far our results are based on linear rate changes. We discuss the possibilities of application to speech.

Introduction

Most approaches to processing temporal signals using connectionist networks assume a fixed rate of the signal. This is true of most recurrent sequence processing networks, such as Elman's Simple Recurrent Networks [Elman, 1990] or Jordan's output recurrent networks [Jordan, 1986]. In these systems, the network and the input are in lock-step. Thus, even though these networks nominally deal with temporal processing, in fact, they simply deal with *sequences*, while ignoring time. In many real-world situations, the assumption of a fixed rate of input is violated. This is especially true in the case of speech [Miller, 1984].

Rate detection is a notoriously difficult problem. It seems that in order to detect the rate of a signal, you first have to know what the signal is. But in order to recognize the signal, you have to know its rate. This is a classic chicken-and-egg problem. The Interactive Activation model of [McClelland & Rumelhart, 1981] solved a similar part/whole problem in the spatial domain using feedback from a representation of the whole to a representation of its parts. This allowed partial information to "bootstrap" recognition of wholes from ambiguous features, recognizing "RED" from noisy versions of its letters, each of which was ambiguous.

In this paper we report on a technique for rate adaptation that makes use of a similar notion. The basic idea of our approach is to build a *model* of the signal using recurrent networks. This model is trained on the signal occurring at a "normal" rate. The network learns to predict the signal at this rate and a fixed delay (see Figure 1). Thus, if it is a good predictor, the network dynamics should match the dynamics of the signal. Now, given a new instance of the signal at a different rate, the network uses its own prediction error to adapt its processing rate. For classification, multiple models can be compared, choosing the model with the lowest error. We describe initial results with simple signals, and discuss the possible application of the technique to speech.

We call our system *Tau Net* for two reasons. First, because the model uses a variable τ to adapt its rate. Second, there is a felicitous pun on Tao, since the claim is that the way to recognize the rate of the signal is to *be* an adaptive model of the signal: "The way to do is to be" [Bynner, 1944].

The Method

The idea of the technique is quite simple: We use the temporal auto-association idea of [Elman, 1990] to train a network to be a model of the signal. There are two differences between what we do and Elman's original formulation. First, we use real time recurrent learning [Williams & Zipser, 1989] instead of truncated back propagation through time. Second, in order to have a way of adjusting the speed of the network, we use the Delta-Net technique of [Tsung, 1991]. The Delta-net is a finite-difference approximation of a continuous time network. As such, it has a parameter, the *time constant* of the network, that changes how fast the units integrate their input. Basically, this is a knob that adjusts the speed of the network. The network architecture is shown in Figure 1.

We train this network for a particular delay and a particular value of τ, the time constant of the network. [1] We then *fix* the weights of this network, and present it with an abrupt new signal. Activation is propagated through the network, and the error signal is used to compute $\frac{\partial E}{\partial \tau}$.

The equations describing a continuous neural network are:

$$\tau_k \frac{dy_k(t)}{dt} = -y_k(t) + f(s_k(t)) \qquad (1)$$

[1] Since we are only dealing with linear rate changes in this paper, a single τ for the network is sufficient.

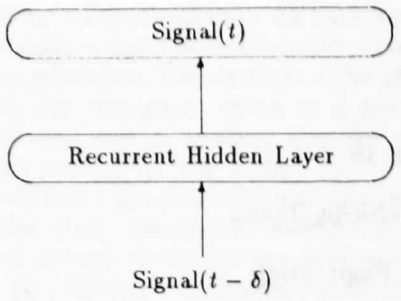

Figure 1: Tau Network

$$s_k(t) = \sum_j w_{kj} y_j(t),$$

where $s_k(t)$ is the net input to unit k at time t, $y_k(t)$ is the output of unit k at time t, w_{kj} are the weights from unit j to unit k, $f(x)$ is the transfer function, and τ_k is the time constant of unit k. Time constants determine the time scale of the system. This can be easily seen by dividing both sides of Equation (1) by τ_k. As τ_k increases, the right hand side of the equation decreases, which means that the network is changing more slowly. Analogously, as τ_k decreases, the opposite effect is achieved.

The equations for the Delta net are derived from a discretized version of the continuous network. One advantage of this approach is that the learning algorithm is simpler than the continuous versions, but the network still retains some essential characteristics of the continuous network [Tsung, 1991]. For a finite-difference approximation to the continuous equations, we use

$$\frac{dy_k(t)}{dt} \approx \frac{y_k(t+\Delta t) - y_k(t)}{\Delta t}$$

to get the following activation rule for the Delta net:

$$y_k(t+\Delta t) = (1 - \frac{\Delta t}{\tau_k}) y_k(t) + \frac{\Delta t}{\tau_k} f(s_k(t)). \quad (2)$$

The rule for updating the weights in the Delta net is:

$$\Delta w_{ij}(t) = -\eta \frac{\partial E}{\partial w_{ij}}(t) \quad (3)$$
$$= -\eta \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial w_{ij}}(t),$$

where, by defining $p_i^k(t+\Delta t) \equiv \frac{\partial y_k}{\partial w_{ij}}(t+\Delta t)$, we have

$$p_i^k(t+\Delta t) = (1 - \frac{\Delta t}{\tau_i}) p_i^k(t) + \frac{\Delta t}{\tau_i} f'(s_k(t)) \left[\sum_j w_{kj} p_i^j(t) + \delta_{ik} y_j(t) \right], \quad (4)$$

and where δ_{ik} is the Kronecker delta:

$$\delta_{ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise.} \end{cases}$$

Updating the time constants is similar to updating the weights. The learning rule for time constants is

$$\Delta \tau_i(t) = -\eta \frac{\partial E}{\partial \tau_i}(t) \quad (5)$$
$$= -\eta \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \tau_i}(t),$$

where, by defining $q_i^k(t+\Delta t) \equiv \frac{\partial y_k}{\partial \tau_i}(t+\Delta t)$, we have

$$q_i^k(t+\Delta t) = (1 - \frac{\Delta t}{\tau_i}) q_i^k(t) + \frac{\Delta t}{\tau_i} \left[f'(s_k(t)) \sum_j w_{kj} q_i^j(t) \right] + \delta_{ik} \frac{\Delta t}{\tau_i^2} [y_k(t) - f(s_k(t))]. \quad (6)$$

The motivation for using the Delta net instead of a discrete network is that the notion of time scale is incorporated into the Delta learning algorithm, providing a natural way to modify the processing speed of a network.

To have the network estimate the rate of the input signal, we first train the network to predict $S(t)$ from $S(t-\delta)$ for a particular δ and with $\tau = 1$ initially. After the network has learned to predict $S(t)$, the weights are fixed. The network is then presented the same signal at a *different* rate. The time constant τ of the network is then adjusted using $\frac{\partial E}{\partial \tau}$. In this way, the network adapts its processing speed to the rate of the input signal, and the final value of τ can be used as a measure of the rate of the input. In order to scale things properly, the delay δ must be adjusted as well. Although a more sophisticated approach is possible, in what follows, we simply adjusted δ by the same amount as τ. There is no guarantee that this will converge, but it did for our test cases.

Experimental results

Sine waves

We trained a 4-3-4 network [2] to predict four sine waves of 4 different phases, with a delay of 8 time steps (there were 80 time steps per period) which corresponds to a phase lag of 36 degrees. We set τ initially to one. The results are shown in Figure 2 (a).

Note that the system adjusts within a single cycle to the rate of the input. The problem posed to the

[2] An i-h-o network refers to a network with i inputs, h hiddens, and o outputs.

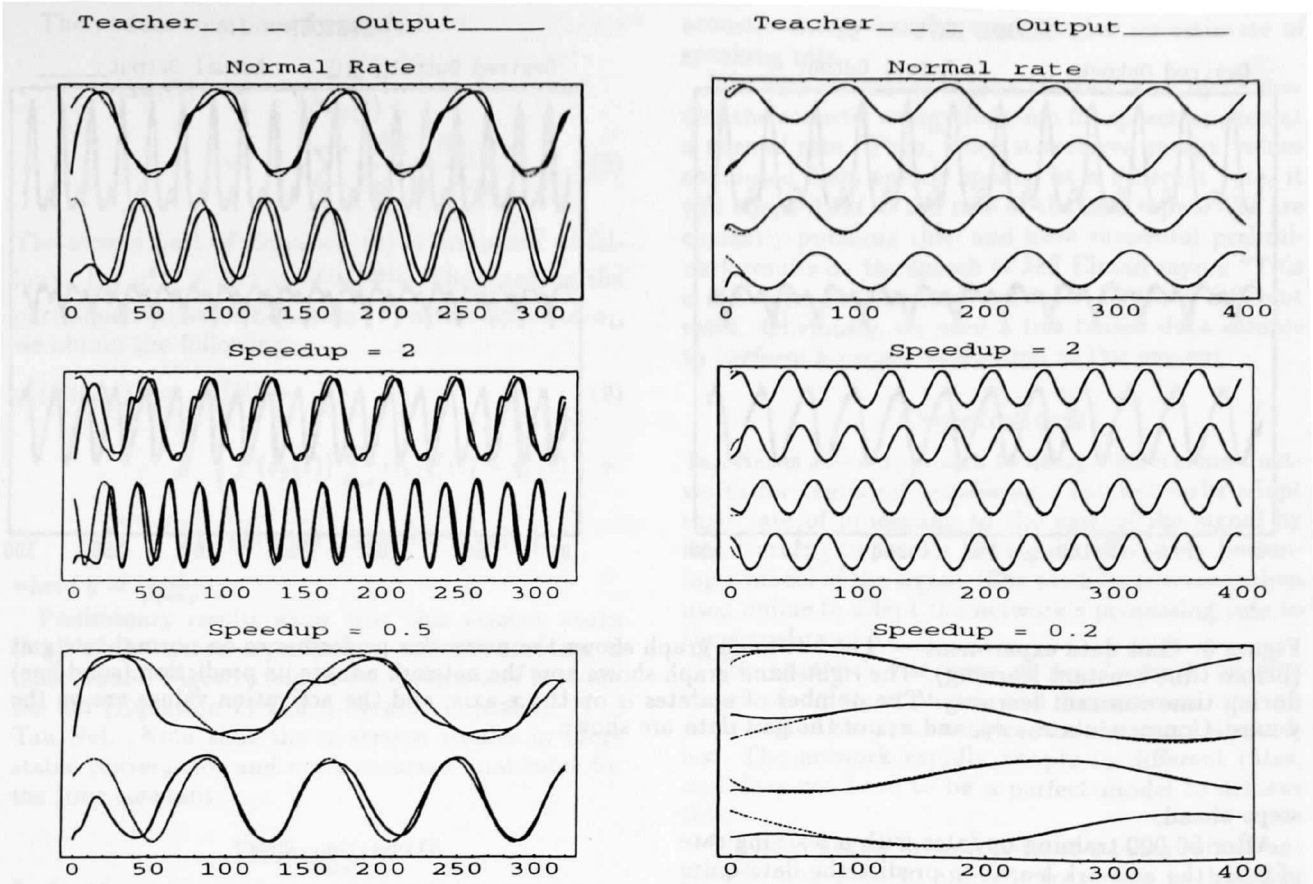

Figure 2: Sine wave experiments — The left-hand graphs show the output with respect to the teacher for the two-frequency problem. The right-hand graphs show results for the four-phase problem. Top Row: Trained network on normal-rate input. Middle Row: Network adapting τ and δ to fast-rate input. Bottom Row: Network adapting τ and δ to slow-rate input. The number of updates is on the x-axis, and the activation values are on the y-axis.

system is exacerbated by the fact that we did not start the system at the original rate and slowly adjust it; we simply presented it with the fast signal abruptly. Note that even though the system is predicting the signal quite well, it only approximately gives the correct time constant (0.54 instead of 0.5 for an input speed of 2x, 4.55 instead of 5 for an input slowed by 0.2x).

The results for a two-frequency problem are shown in Figure 2 (b). For the 2-frequency net, the adapted τ's came quite close to the correct values (0.45 vs. 0.50, and 1.7 vs. 2.0). Thus, even though the system does not fit the signal perfectly during initial training, it still is able to converge to an acceptable time constant.

Gait data

The final problem is a set of motion variables generated by children walking. We used data extracted at the Motion Analysis Laboratory at the Children's Hospital, San Diego, from free-speed, level walking subjects. The data are based on film recordings by fixed-position cameras of the subjects as they traverse a walkway. Each subject has 12 "markers" and 4 "sticks" precisely placed on their waist, legs and feet as reference points for the analysis. A complete description of the data-gathering process and preprocessing is given in [Sutherland et al., 1988]. The results of this calculation are 12 different joint rotation parameters from each side of the body. The data is arranged in a *gait cycle*, defined as (for example) the time from heel-strike to heel-strike, and thus is temporally normalized between subjects. For this problem, we used averaged data from normal seven-year olds.

Each vector consists of 24 floating point numbers to represent the joint angles in both legs at each time step. We do not have data for children walking at different rates, which would be a more interesting problem; instead, we simulated different rates by sampling the signal at different intervals. "Normal rate" refers to every other time step; "fast rate" refers to every third time step for a speedup of 1.5; and "slow rate" refers to every time step for a speedup of 0.5 relative to the normal rate. We use a 24-10-24 network, and ask it to predict the 24-dimensional signal two time

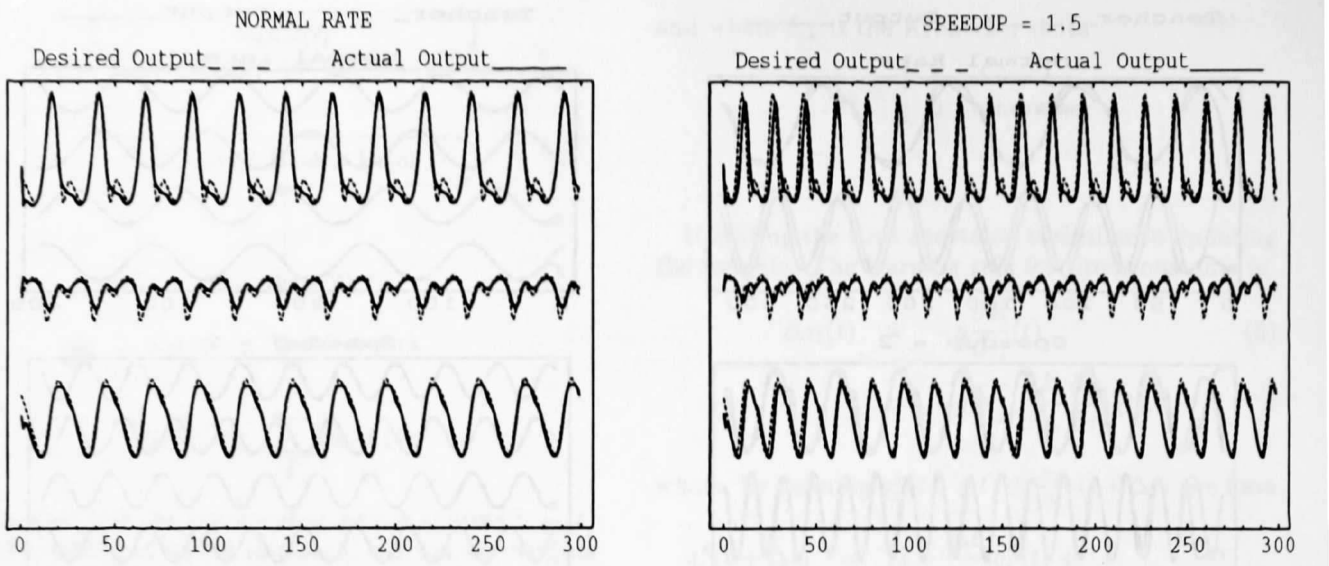

Figure 3: Gait data experiment — The left-hand graph shows the network's performance on normal-rate gait (before time constant learning). The right-hand graph shows how the network adapts its prediction (solid line) during time constant learning. The number of updates is on the x-axis, and the activation values are on the y-axis. Components x_0, x_1, and x_{14} of the gait data are shown.

steps ahead.

After 50,000 training updates with a learning rate of 0.01, the network learns to predict the data quite well, though not perfectly. This is shown in Figure 3. Only three of the components are shown here. Similar results were obtained for all twenty-four components in the gait data. With a time constant learning rate of 0.05, the network learns to predict the fast input within approximately 250 updates. The network's prediction of the fast-rate input as τ and δ are adjusted is shown with respect to the actual values of the delayed input in Figure 3 (right). Similar results were obtained for slowed input. The prediction error during time constant learning and the time constant values for the fast input evolve in a cyclic manner. In the next section, we describe a variation on this approach that ameliorates this problem.

Current Work

Recently, we have been experimenting with a somewhat different set of equations. A potential problem with our system is that $\frac{\partial E}{\partial \tau}$ results in a τ^2 term in the denominator of equation 6, which can lead to instability when τ is small. Also, there are range constraints on the ratio of the step size of the integration (Δt) to τ, i.e., equation 2 makes sense only if $0 < \frac{\Delta t}{\tau_k} < 1$. The learning algorithm can lead to values for τ that make this ratio greater than one, which requires a check to set the ratio back to less than one. To avoid these two problems, following a suggestion by David Rumelhart, we decided to use a change of variables to replace the ratio $\frac{\Delta t}{\tau_k}$ with $g(\alpha) = \frac{1}{1+\exp^{-\alpha_k}}$. Now we adjust α instead of τ, and τ can be recovered as

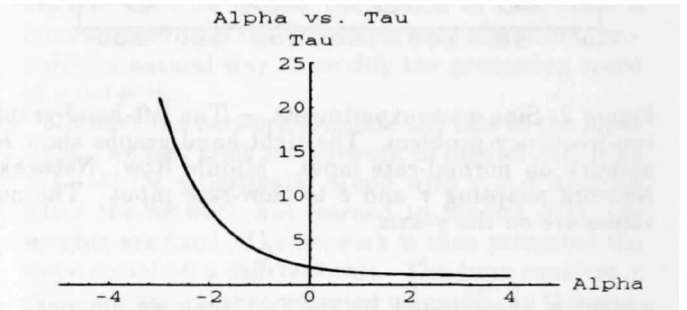

Figure 4: Tau as a function of alpha.

$1+\exp^{-\alpha}$ (note that Δt is now assumed to be 1). This gives a new set of update equations for Tau Net:

$$y_k(t+\Delta t) = (1-g(\alpha))y_k(t) + g(\alpha)f(s_k(t)) \quad (7)$$

Besides being a familiar friend, using this sigmoidal expression for the ratio scales the changes to τ properly. When τ is very big, large changes are needed to make a difference in equation 2. When τ is very small, small changes make a big difference. Figure 4 shows that changing α instead of τ scales these changes properly. That is, when α is negative, and thus the time constant is big (and the system sluggish), a small change in α makes a large change in the effective time constant. When α is positive, and the effective time constant is small, a large change in α makes a small change in the effective time constant.

The update equations for α are:

$$\begin{aligned}\Delta\alpha_i(t) &= -\eta\frac{\partial E}{\partial\alpha_i(t)} \\ &= -\eta\sum_i \frac{\partial E}{\partial y_k(t)}\frac{\partial y_k(t)}{\partial\alpha_i}. \qquad (8)\end{aligned}$$

The second part of Equation (8) is evaluated as follows. Let $q_i^k(t+\Delta t) \equiv \frac{\partial y_k(t+\Delta t)}{\partial\alpha_i}$. Then, taking the partial derivative of Equation (7) with respect to α_i, we obtain the following:

$$\begin{aligned}q_i^k(t+\Delta t) &= q_i^k(t) + \qquad (9)\\ & g\cdot\left(f'(s_k(t))\sum_j w_{kj}q_i^j(t) - q_i^k(t)\right) + \\ & \delta_{ik}\cdot g(1-g)\cdot[f(s_k(t)) - y_k(t)]\,.\end{aligned}$$

where $g = \frac{1}{1+\exp^{-\alpha}}$.

Preliminary results show that this version works in all of the above experiments, and the learning is more stable. Figure 5 shows the comparison between the old (Equation 2) and α-version (Equation 7) of Tau Net. Note that the α-version results in more stable convergence and more accurate final value for the time constant.

Discussion

So far the technique has worked well for simple signals. It turns out that it is not necessary for the system to either predict the signal perfectly to begin with or to converge to exactly the correct time constant to do a good job of prediction. Tau net rapidly adapts to sudden-onset signals at different rates when started "cold", with an incorrect internal state. It works for simple sine waves and for more complicated gait data.

It remains to be seen if this can be applied to signals with variable internal rates. For this, we can use a Tau Net with a time constant for every unit adapting independently. Such a system, if it is a good model, would need to adapt to different "gaits" of the signal. We have not had experience with problems of this type.

How could this system be applied to speech? It is characteristic of speech that rate changes are *nonlinear*, in that vowels are shortened in rapid speech more than consonants. However, it seems reasonable to assume that if we knew the global rate change, we could adapt to the differences. A Tau Net can be used as a rate-estimation module. Then the rate of the recognizers can be set by this global "rate box". The input it receives is the acoustic energy computed from the speech signal. Vowels are the speech sounds with the strongest intensities [Denes & Pinson, 1963]; thus, acoustic energy peaks can be used to indicate vocalic segments in the speech signal. Since the rate at which vocalic segments occur is comparable to the syllable rate in speech production [Rabiner & Schafer, 1978], acoustic energy can be used to give an estimate of speaking rate.

The rate estimator will be trained initially to predict the acoustic energy function for speech spoken at a normal rate. Then, when it receives energy values computed from speech spoken at a different rate, it will adapt itself to the rate of the new input. We are currently pursuing this, and have successful preliminary results on the speech of Jeff Elman saying "This is the voice of the neural network" at three different rates. Obviously, we need a less biased data sample to perform a proper evaluation of this system.

Conclusions

Tau Net is a new approach to using connectionist networks for temporal processing. Tau networks adapt their rate of processing to the rate of the signal by first learning to predict the signal, effectively becoming a model of the signal. The prediction error is then used online to adapt the network's processing rate to the signal rate.

The approach has been shown to work on sine waves of different phases and frequencies, and on complicated motion variables from human gait studies. The network rapidly adapts to different rates, and does not have to be a perfect model to achieve this.

We have begun investigation of a more robust approach, and plan to apply it to speech by building phoneme models and then adapting their rates to the signal online. Rate variation abounds in speech data, and we hope to provide recognizers that are robust with respect to rate using this approach.

References

[Bynner, 1944] Bynner, W. 1942 The Way of Life according to Lao Tzu. New York: Putnam.

[Denes & Pinson, 1963] Denes, P.B. &Pinson, E.N. 1963 The speech chain: the physics and biology of spoken language. Bell Telephone Laboratories.

[Elman, 1990] Elman, J. 1990 Finding structure in time. *Cognitive Science* 14:179–211.

[Jordan, 1986] Jordan, M. 1986 Serial order: A parallel distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego.

[McClelland & Rumelhart, 1981] McClelland, J.L. & Rumelhart, D.E. 1981 An interactive activation model of context effects in letter perception: Part I, An account of basic findings. *Psychological Review* 88:375-407.

[Miller, 1984] Miller, J.L., Grosjean, F. & C. Lomanto 1984 Articulation rate and its variability in spontaneous speech: a reanalysis and some implications. *Phon* 41:215–225.

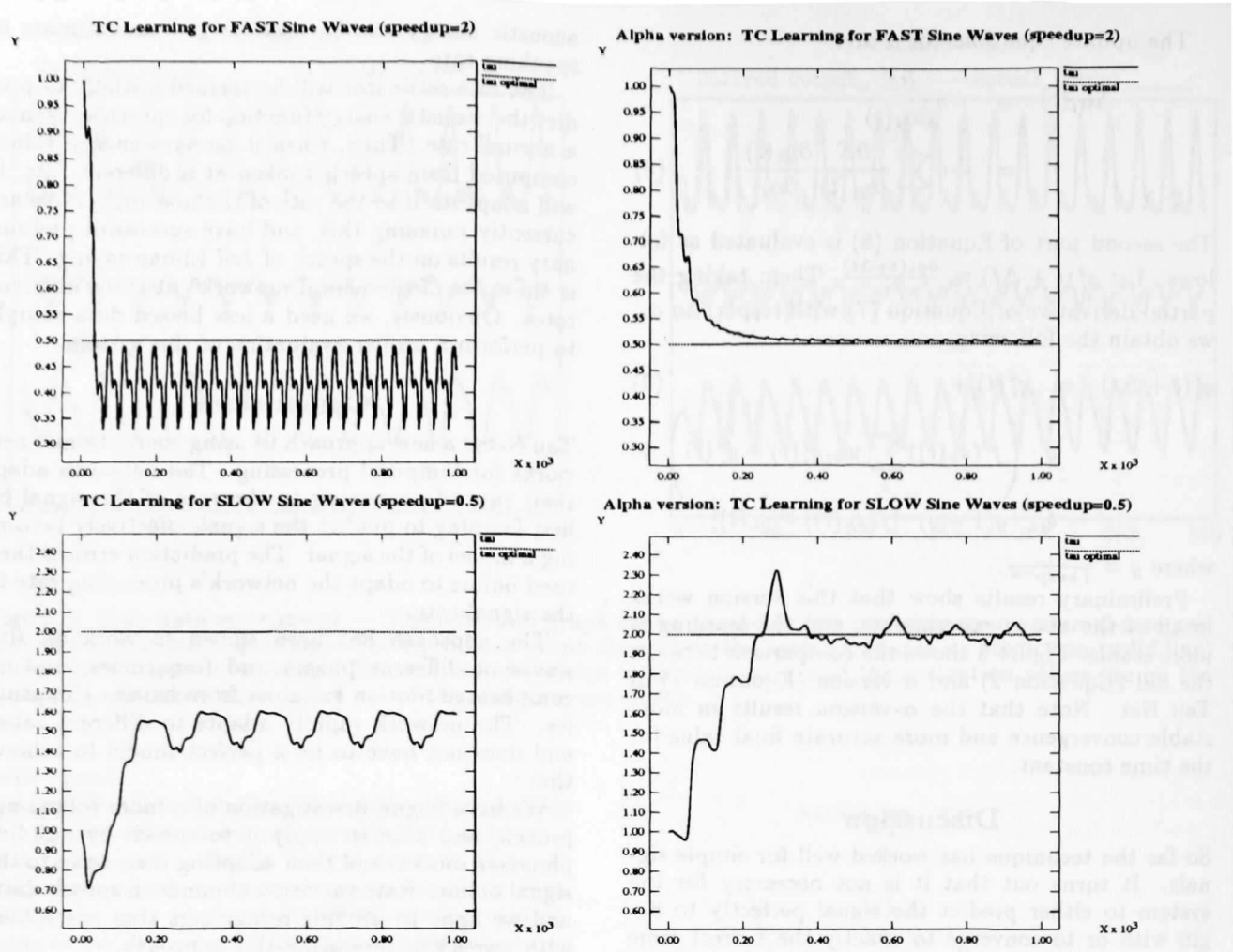

Figure 5: The benefit of Rumelhart's advice. These graphs show how τ evolves during time constant learning on the two-frequency problem. The original version of Tau Net is shown in the left hand graphs, and α-version on the right. The upper graphs are for a speeded signal, the lower graphs for a slowed signal. The dotted line indicates the optimal value of τ.

[Rabiner & Schafer, 1978] Rabiner, L.R. & Schafer, R.W. 1978 *Digital processing of speech signals.* Englewood Cliffs, NJ: Prentice-Hall.

[Sutherland et al., 1988] Sutherland, D. H., Olshen, R. A., Biden, E. N. & Wyatt, M. P. 1988 *The development of mature walking.* Oxford: Mac Keith.

[Tsung, 1991] Tsung, Fu-Sheng 1991 Learning in recurrent finite difference networks. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski & G. E. Hinton (Eds.) *Proceedings of the 1990 Connectionist Models Summer School.* San Mateo: Morgan Kaufmann.

[Williams & Zipser, 1989] Williams, R. and Zipser, D. 1989 A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 1:270–280.