Tau Net: A Neural Network for Modeling Temporal Variability

Mai H. Nguyen* Garrison W. Cottrell
Department of Computer Science & Engineering
Institute for Neural Computation
University of California, San Diego
La Jolla, CA 92037-0114
mnguyen@cs.ucsd.edu

November 1996

Abstract

The ability to handle temporal variation is important when dealing with real-world dynamic signals. In many applications, inputs do not come in as fixed-rate sequences, but rather as signals with time scales that can vary from one instance to the next; thus, modeling dynamic signals requires not only the ability to recognize sequences but also the ability to handle temporal changes in the signal. This paper discusses "Tau Net," a neural network for modeling dynamic signals, and its application to speech. In Tau Net, sequence learning is accomplished using a combination of prediction, recurrence and time-delay connections. Temporal variability is modeled by having adaptable time constants in the network, which are adjusted with respect to the prediction error. Adapting the time constants changes the time scale of the network, and the adapted value of the network's time constant provides a measure of temporal variation in the signal.

Tau Net has been applied to several simple signals: sets of sine waves differing in frequency and in phase [2], a multidimensional signal representing the walking gait of children [3], and the energy contour of a simple speech utterance [11]. Tau Net has also been shown to work on a voicing distinction task using synthetic speech data [12]. In this paper, Tau Net is applied to two speaker-independent tasks; vowel recognition (of /ae/, /iy/, /ux/) and consonant recognition (of /p/, /t/, /k/) using speech data taken from the TIMIT database. It is shown that Tau Nets, trained on medium-rate tokens, achieved about the same performance as networks without time constants trained on tokens at all rates, and performed better than networks without time constants trained on medium-rate tokens. Our results demonstrate Tau Net's ability to identify vowels and consonants at variable speech rates by extrapolating to rates not represented in the training set.

Keywords: recurrent neural networks, predictive neural networks, temporal processing, speech recognition

1 Introduction

In many real-world applications, (e.g., control plant monitoring, financial prediction, EKG tracking, music performance, speech recognition), inputs are dynamic signals. To model dynamic signals, two components are required: the ability to recognize sequences, and the ability to handle temporal

*Current affiliation: TRW Space & Technology Division, One Space Park, Redondo Beach, CA 90278.
The authors wish to thank members of GEURU at UCSD for many useful discussions on Tau Net. We also thank the reviewers for helpful comments on this manuscript.
variability. Sequence recognition is necessary because dynamic signals are not single patterns but sequences of patterns. Modeling temporal variability is important because in many real-world applications, inputs do not come in as fixed-rate sequences, but rather as temporal signals with varying time scales. For example, in speech, signals are frequently warped due to differences in speaking rate and style. In this paper, we apply a neural network designed to model dynamic signals, called "Tau Net," to the speaker-independent recognition of vowels and consonants.

2 Existing Connectionist Approaches to Speech Recognition

Speech production is an inherently temporal process. Capturing the dynamic nature of the speech signal has always been a challenge in speech recognition, especially for connectionist approaches. Following is a brief review of existing connectionist models that explicitly address this challenge.

2.1 Hybrids

One approach to capture speech signal dynamics is to combine neural networks with more conventional methods with properties suitable for addressing timing problems, such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW). Both HMM and DTW use a dynamic programming algorithm for sequence recognition and handling temporal variation. Most current hybrid approaches use neural networks to perform recognition at the frame level and HMMs on top to concatenate sequences of frame-level results to perform phoneme- and word-level recognition. Hybrid neural network-HMM systems have been used for large-vocabulary, speaker-independent, continuous speech recognition tasks (e.g., see [15]).

2.2 Time Delay Neural Networks

Another approach is to use time delay neural networks (TDNNs). A TDNN is a multi-layer feedforward network in which time is modeled as space. It has a large input window that spans several time steps; thus, the temporal sequence is turned into a spatial pattern at the input layer of the network. Temporal relationships between acoustic events are captured via time delay connections and successively larger receptive fields from layer to layer, and shift invariance is achieved by imposing local receptive fields and weight sharing on units within the same layer. TDNNs have been applied to a speaker-dependent phoneme recognition task [21] and a multi-speaker isolated word recognition task of the difficult E-set (BDEV) [7].

2.3 Recurrent Neural Networks

A third approach is to use recurrent connections to enable the network to develop internal states. This allows the network to keep a history of its responses to previous inputs, and to use that information in processing the current input. A record of previous activations is essential for sequence learning. Recurrent connections can be either fixed or adaptable, and a network can be partially recurrent (containing mostly feedforward connections with a carefully chosen set of feedback connections such as recurrence only at the hidden layer or self-recurrent connection on each unit) or fully recurrent (all units are connected to all other units in the network).

Successful approaches using recurrent networks for speech recognition tasks include the temporal flow model and the recurrent error propagation network (REPN). In the temporal flow model, each hidden and output unit has a self-recurrent weight, and connections between successive layers are time-delayed. The temporal flow model has been applied to several single-speaker phonetic
discrimination tasks involving place of articulation, voicing, etc [22]. The REPNet has a fully recurrent hidden layer, and each weight in this network has a separate learning rate. The REPNet can also be considered a hybrid system since frame-level classifications determined by the network are processed by an HMM to produce phone-level classifications. The REPNet has been shown to work quite well on a speaker-independent phoneme recognition task using data from TIMIT, a large standard speech corpus [16].

2.4 Predictive Neural Networks

Yet another way to model time-dependent speech signals with neural networks is to use a predictive approach. A predictive neural network is trained to predict the value of the signal for the current time step given \( p \) previous values; thus, the network is trained to extrapolate the speech signal rather than classify it. The predictive approach forces the network to explicitly learn temporal correlations among successive feature vectors in the speech pattern, thereby enabling the network to model the speech signal dynamics. A separate predictive network is trained to model each subword (phoneme or word, depending on the task). Recognition is done indirectly, by comparing prediction errors of all networks in response to an unknown speech signal. The network with the lowest prediction error most closely models the unknown signal, and therefore classifies it.

Approaches using predictive networks for speech recognition include the neural prediction model (NPM), linked predictive neural networks (LPNN), and the hidden control neural network (HCNN). In the NPM, a word is modeled by a sequence of predictive networks. The NPM has been applied to a speaker-independent isolated Japanese digit word recognition task [4]. A similar approach is taken in the LPNN: A word is modeled by a sequence of predictive networks, and a phoneme, in turn, is modeled by a sequence of three predictive networks. Networks that correspond to the same phoneme (but in different words or in different parts of the same word) have their weights linked together, to allow for modeling phonemes in different contexts. The LPNN has been applied to a single-speaker Japanese isolated word task [17] and a continuous speech recognition task [18]. In the HCNN, sequencing is modeled in a manner similar to HMMs: the current state of the model is explicitly determined, and the output of the model is dependent on the current state. The current state of the HCNN is determined by a control input, which is separate from the acoustic input. The HCNN has been tested on a speaker-independent connected digit recognition task [8].

2.5 Temporal Variability

As discussed earlier, modeling real-world dynamic signals such as speech requires sequence recognition and the ability to handle temporal variability. In all of the approaches described above, except for the hybrid approach, sequence recognition capability is provided within the neural network framework, via time delay connections, a large input window, recurrent connections, or prediction. Temporal variability, on the other hand, is a different story. With the exception of the HCNN, all of the neural network models described above rely on external mechanisms for handling temporal variability. It seems logical that addressing temporal variability within the neural network itself would be more efficient and would provide a better model of the signal dynamics than addressing it at a later processing stage. In the HCNN, time variability is modeled by keeping the network parameters fixed and changing the network's input-output mapping as a function of the current network state, as determined by the control input. An alternative is to allow some network parameter(s) to vary with time. The latter approach is taken in Tau Net, as described next.
3 Tau Net

Tau Net also uses a predictive approach for sequence recognition. As shown in Figure 1, Tau Net receives the feature vector at time \( t - \delta \), and is trained to predict the feature vector at time \( t \). The idea is that, after training, the dynamics of the network should match the dynamics of the signal, and the network essentially becomes a model of the signal. Compensating for temporal variability is then accomplished by adapting the time constant of the network with respect to the prediction error, which adjusts the time scale of the network, as explained below.

Following Tsung [19], the set of equations we use for Tau Net is a finite-difference approximation to a continuous-time network. The equations describing a continuous neural network are:

\[
\frac{dy_k(t)}{dt} = \frac{1}{\tau_k} \left[ -y_k(t) + f(s_k(t)) \right] \\
\]

In the above equations, \( s_k(t) \) is the input to unit \( k \) at time \( t \), \( y_j(t) \) is the output of unit \( j \) at time \( t \), \( w_{kj} \) are the weights from unit \( j \) to unit \( k \), and \( \tau_k \) is the time constant of unit \( k \). The activation of unit \( k \) is computed by passing the net input \( s_k(t) \) through a nonlinear transfer function \( f(\cdot) \). In our experiments, the tanh function was used.

Discretizing Equation 1 using the finite-difference method,

\[
\frac{dy_k(t)}{dt} \approx \frac{y_k(t + \Delta t) - y_k(t)}{\Delta t},
\]
yields

\[ y_k(t + \Delta t) = (1 - \frac{\Delta t}{\tau_k}) \cdot y_k(t) + \frac{\Delta t}{\tau_k} \cdot f(s_k(t)) \] (2)

\[ s_k(t) = \sum_j w_{kj} y_j(t). \]

One advantage of using the discretized version of a continuous network is that the learning algorithm is simpler than the continuous versions, but the network still retains some essential characteristics of the continuous network [19].

Equation 2 may lead to instabilities, however, since the gradient \( \frac{\partial E}{\partial \tau_i} \) contains a \( 1/\tau_i^2 \) term. This is, of course, undesirable since \( \frac{\partial E}{\partial \tau_i} \to -\infty \) as \( \tau_i \to 0 \). In addition, Equation 2 makes sense only if \( 0 \leq \Delta t/\tau_k \leq 1 \). Hard-limiting \( \Delta t/\tau_k \) to 0 and 1 seems a clumsy way to enforce this constraint. Thus, to sidestep these problems and provide a more stable implementation, the ratio \( \Delta t/\tau_k \) is replaced with \( 1/(1 + e^{-\alpha k}) \) in Tau Net.

In the training phase, the time constant is fixed while the weights are adapted. In our experiments, \( \Delta t/\tau = 0.5 \) during training, which is the center value in the allowable range of \((0,1)\) for \( \Delta t/\tau \). The weights are adapted using gradient descent to minimize the prediction error. At the end of training, when the network has learned to predict the signal, the weights are fixed. During recognition, only the time constant is adjusted with respect to the prediction error. Adjusting the time constant changes the time scale of the network. This can be seen most easily by referring to Equation 1: as \( 1/\tau \) decreases, \( dy/dt \) also decreases, and the network operates on a slower time scale, and vice versa. Adjusting the network’s time constant allows it to operate at different time scales, thus enabling the network to model signals with varying time scales. Note that Equation 2 allows for one time constant per unit in Tau Net. However, in our experiments, we have used only one time constant for the entire network. This is so that we have a single measure of temporal variation. In summary, having adaptable time constants adds flexibility to Tau Net: adaptable time constants allow the network to adjust its time scale, enabling it to model variable-rate signals, and the change in the time constant usefully provides a measure of the signal’s temporal variation.

Because Tau Net is trained using the predictive approach, it explicitly learns any temporal correlations that exist between successive feature vectors in the signal. This is essential in modeling time-dependent patterns. Additionally, the predictive approach eliminates the need for an explicit target function, as the delayed input signal itself serves as the target. The choice of target function is especially crucial in modeling a dynamic signal: a time-dependent target function is desirable, but the appropriate selection of such a function is difficult. Note also that there is no explicit teacher for the correct rate or time scale; that is, there is no target value to tell the network how fast the signal is coming in or how fast the network should be processing its input. For many temporal signals, this is an ill-defined concept. (This is especially true of speech: Should the rate of the speech signal be defined as average syllable duration or number of syllables per second or number of words per minute, etc.? Additionally, how should the frequency and duration of pauses be figured into the equation?) Thus, in many cases, it would be difficult to come up with a measure of temporal variation that is accurate enough to be used as a target value for the network. With our approach, the network automatically determines its processing rate in the process of minimizing its prediction error.

Other researchers have used time constants in neural networks (e.g., [10] [13] [14]). However, none of this work investigated the adaptation of time constants to dynamically adjust the processing rate of a neural network.
4 Experimental Results

In previous work, we have tested Tau Net on sine waves of different phases and frequencies [3], a 24-dimensional signal representing gait motion in children [2], the energy contour of a simple speech utterance [11], and on a voicing task using synthetic speech data [12]. In all these experiments, Tau Net is trained on signals at the medium rate and tested on signals at all rates (or time scales). During testing, the weights are fixed, but the time constant is adjusted to allow the network to adjust its time scale to adapt to signals at different rates. In all cases, adapting its time constant allows Tau Net to correctly categorize signals at rates not in the training set, and the adjusted time constant appropriately reflects temporal variation in the test signal; that is, $\Delta t^1/\tau$ increases when the rate of the signal increases, and vice versa.

In the following experiments, Tau Net is applied to phoneme recognition tasks using speech data taken from the TIMIT database. The first experiment is a recognition task using vowels from the set $\{/ae/, /iy/, /ux/\}$. The second is a recognition task using consonants from the set $\{/p/, /t/, /k/\}$. Substantial rate variation exists in speech, across speakers as well as for individual speakers. Consequently, we expect to encounter variable rates in the speech data, thus, providing a good test for Tau Net's ability to model temporal variation.

4.1 Speech Data

The data used for these experiments were taken from TIMIT, a large speech database with over 600 speakers [6]. TIMIT was designed for developing speaker-independent phoneme-based speech recognition systems and is widely used. A Hamming window of width 256 samples was applied to the speech waveform every 10 msec. From each windowed segment, or frame, the following features were extracted: log power, and 14 normalized melscale spectral coefficients computed from the power spectrum. The range of these features is $[0,1]$. For the consonant data, the log power was also extracted for each frame, resulting in 15 speech parameters per input vector.

4.2 Training and Testing Methodology

The Xerion simulator [20] was used to implement the networks used in these experiments. The hyperbolic tangent was used for the unit activation function, the networks were trained with the real-time recurrent learning (RTRL) algorithm [23]. Optimization was performed using the conjugate gradient method; for the following experiments, the default settings for conjugate gradient in Xerion were used.

Since TIMIT was designed for speaker-independent recognition (as opposed to multi-speaker recognition), speech data for training and testing were taken from two separate sets of speakers. Thus, there was no overlap between training and test data used in our experiments. Approximately 15 to 20% of the training data was set aside to be used for validation. Training was stopped at the point at which prediction error on the validation set was lowest, and the corresponding set of weights for the network was used for testing.

\footnote{In our experiments, we use $\Delta t = 1$.}
4.3 Vowel Recognition

4.3.1 Vowel Data

In the vowel experiment, the task was to distinguish between three vowels: /æ/, /iy/, and /ux/. These vowels were chosen because they are among the most elastic phonemes in the TIMIT database; their durations have standard deviations of 0.0442, 0.0357, and 0.0528 seconds, respectively, compared to the average standard deviation of 0.0316 seconds for all phonemes in TIMIT [1]. These vowels would therefore make a good test for temporal variability. Training and test vowel tokens were extracted from the following contexts: had, greedy, suit for /æ/, /iy/, /ux/, respectively. Contexts were constrained to reduce variations in the speech signal due to coarticulation. The TIMIT database is divided into 8 different dialect regions. For the vowel recognition experiment, data of male speakers from dialect region 7 (Western region) and dialect region 8 (Army brat, or people who have moved around a lot) were used.

From the above training sets, subsets of training data containing only medium-rate 2 tokens were extracted. For /iy/, medium-rate tokens were defined to be tokens of average duration, as computed from the training data. For /æ/ and /ux/, medium-rate tokens also included tokens that are one frame longer than the average duration. Tokens with shorter durations can be considered fast tokens, and those with longer durations can be considered slow tokens. Training sets (not including those used for validation) consisted of 40, 88, and 60 tokens for /æ/, /iy/, /ux/, respectively, of which 14, 26, and 14 tokens were medium-rate tokens. Test sets contained 11, 15, and 14 tokens. Table 1 shows the distribution of fast, medium, and slow tokens for training and test sets for the three vowels.

<table>
<thead>
<tr>
<th>VOWEL</th>
<th>DATA SET</th>
<th>FAST</th>
<th>MEDIUM</th>
<th>SLOW</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>/æ/</td>
<td>Train</td>
<td>17</td>
<td>14</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>/iy/</td>
<td>Train</td>
<td>42</td>
<td>26</td>
<td>20</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>/ux/</td>
<td>Train</td>
<td>28</td>
<td>14</td>
<td>18</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 1: Vowel Data — Distribution of fast-, medium-, and slow-rate tokens for training and test sets for the three vowel classes. NO TC-ALL was trained on all training tokens, while TAU-NET and NO TC-MED were trained only on medium-rate tokens.

4.3.2 Vowel Experiment Setup

Three different Tau Nets were trained, one for each of the three vowels. The network architecture for each Tau Net had 14 input units, 10 hidden units, and 14 output units (see Figure 1). The hidden layer was fully recurrent and fully connected to the output layer. The task was to predict the feature vector two time steps ahead. (The prediction delay d was set to three, but since there is an inherent delay between the input and output layers due to the recurrent hidden layer, the actual delay between the input and target signals was two time steps.) Since the spectral characteristics

---

2In speech, temporal variation is caused by changes in speaking rate and results in lengthening or shortening of the token duration; tokens spoken at a fast speaking rate will have shorter durations, and vice versa. Thus, terms such as “tokens of average duration” and “medium-rate tokens” will be used interchangeably.
for vowels are relatively stable over a short time interval, a prediction step size that is too small would produce input and target frames that are very similar and may prevent the network from learning the dynamics of the signal. A prediction step size that is too large, on the other hand, would make the prediction task unnecessarily difficult. A prediction step size of two time steps seemed a reasonable choice for vowel tokens to circumvent both problems. No other prediction step sizes were tried for the vowel experiments.

Only medium-rate tokens were used for training the Tau Nets. The goal was to see if the networks could extrapolate to tokens of durations not encountered during training. As described in Section 3, during training, the time constant of each network was fixed while the weights were adapted to minimize prediction error. The final set of weights used for testing for each network was determined by the lowest validation set error.

During recognition, each test token was run through all three Tau Nets, and classification was determined by the Tau Net with the smallest prediction error. The time constant was adapted for each test token to compensate for temporal variability. Time constants of different Tau Nets were adapted independently of one another. The questions of interest are: (1) How does time constant adaptation affect classification? and (2) How does the change in the time constant correlate with the temporal variation in the signal?

In the experiments described in this paper, time constant adaptation was performed over the length of a token, i.e., the time constant was adjusted to minimize the prediction error accumulated for all frames of the token. This one-dimensional minimization was performed using a line search method. This procedure made use of the segmentation information (endpoints of a token), which was readily available for TIMIT data. It is possible to adapt the time constant online as well. In previous work ([3] [2] [11]), we have used online adaptation of the time constant using backpropagation with momentum.

To quantify the benefits of adaptable time constants, a comparison was made with networks with the same architecture but without time constants. The main difference is that, for these networks, \(\Delta t/\tau = 1\) during training and testing, so in effect, these networks had no time constants and were just the normal discrete networks. The same training and testing procedures were used with these networks, namely, each network was trained using the predictive approach to recognize a different vowel, and classification of a test token was determined by the network with the smallest prediction error. One set of these networks was trained on the entire training set, and another set was trained on the subset of training data containing only medium-rate tokens (just as with Tau Nets). For simplicity, networks without time constants that were trained on all tokens will be referred to as NoTC-ALL, and networks without time constants that were trained on medium-rate tokens will be referred to as NoTC-MED. The system consisting of Tau Nets will be referred to as Tau-Net. Note that all systems used the same sets of initial random weights, and that all systems were evaluated on the same test sets.

### 4.3.3 Vowel Recognition Results

Recognition performance is the metric by which the three systems of networks, NoTC-ALL, NoTC-MED, Tau-Net, were evaluated. Classification of a test token is determined by the network with the smallest prediction error. Thus, recognition performance of a network is determined by its prediction performance relative to the other networks on a given token. Figure 2 shows prediction errors expressed as unit mean squared error (MSE) for all three Tau Nets in response to /ae/ test tokens. Figure 2a shows errors before time constant adaptation (with time constants fixed at \(\Delta t/\tau = 0.5\)), and Figure 2b shows errors after time constants have been adapted to minimize prediction errors. The solid line in these graphs indicates errors for the Tau Net trained on /ae/ tokens, and the other lines indicate errors for the other two Tau Nets. As such, the solid line should
be below the other two lines for all test tokens; a misclassification occurs when the solid line is not below the other two lines. As seen in Figure 2a, there were 5 misclassifications before time constant adaptation (test tokens 2, 8, 9, 10, and 11 were misclassified). Figure 2b shows that this number is reduced to 0 after the time constants were adapted. In these graphs, tokens are ordered from short to long, or equivalently, fast to slow. Note that in Figure 2a, error curves are more or less U-shaped, indicating that short and long tokens have high errors and are more likely to be misclassified than tokens of average duration before time constant adaptation. This is because the training set contained only average-length tokens, and without time constant adaptation, the networks are not able to recognize the long/short tokens, which are essentially the same tokens but at different time scales. In contrast, after time constant adaptation, error curves flatten out, indicating similar errors on all tokens, regardless of duration. These results demonstrate that time constant adaptation enabled the networks to compensate for temporal variation, thus allowing the networks to recognize the same signal at different time scales. Furthermore, better prediction performance due to time constant adaptation also translated to improved recognition performance, as seen in Figure 2b.

Figure 3 shows how the three systems of networks compare on recognition performance. Recognition errors for Tau-Net and NoTC-All were equivalent (5% for all three vowels, 0% for /æ/, 6.7% for /i/ and /ax/), and both were lower than for NoTC-Med (12.5% for all three vowels, 0% for /æ/, 13.33% for /i/ and 21.43% for /ax/). These recognition error rates become much more significant when one considers the difference in training set sizes for Tau-Net and NoTC-All. NoTC-All was trained using 100% of the training data, while Tau-Net and NoTC-Med were trained only on medium-rate tokens, which comprised of approximately 30% of the training data, meaning that Tau-Net only had to perform approximately 30% as many computations as NoTC-All for every training iteration. Additional computations were required for adapting the time constant. However, since there is only one time constant per Tau Net, this is a one-dimensional minimization problem, and therefore, is much less computationally expensive than minimizing the error in multi-dimensional weight space. More important than the difference in training set sizes, however, is the fact that Tau-Net was only trained on medium-rate tokens, yet was able to extrapolate to fast- and slow-rate tokens that were not represented in its training data.
Figure 2: Prediction errors for /ae/ test tokens (a) before time constant adaptation (time constants fixed at 0.5), and (b) after time constant adaptation. Recognition performance of a network is determined by its prediction performance relative to the other networks. As indicated here, both prediction and recognition performances for the network trained on /ae/ improved after time constant adaptation.
Two other sets of random initial weights were used. Results obtained with all three sets of random initial weights are shown in Figure 4. Tau-Net achieved the same or nearly the same level of performance as NoTC-All, and consistently got fewer errors than NoTC-Med. The number of training iterations, averaged over all initial weights and all vowel classes, were 253, 517, and 214 for Tau-Net, NoTC-All, and NoTC-Med, respectively. For this task, not only did Tau-Net require fewer computations per training iteration, but also required fewer training iterations overall. Thus, these results demonstrate that Tau-Net achieved nearly the same level of classification performance on the vowel recognition task with significantly fewer computations than NoTC-All.

Table 2 shows the effect of time constant adaptation on recognition error rates for all vowel classes and for all three sets of initial random weights. These numbers provide evidence of the benefit of having adaptable time constants in Tau-Net, as they show that recognition error rates decreased significantly after time constant adaptation.

Also of interest was how well the change in the time constant reflects temporal variation in the test tokens. Figure 5 shows the correlation between the change in the time constant for the correct network and the duration of each test token. Results are shown for all three sets of initial weights. Duration is normalized with respect to the average duration for that vowel class. Correlation between \( \Delta t/\tau \) and token duration is -0.9232, -0.9298, and -0.9294 for /ae/, /iy/, and /ux/, respectively. This means that \( \Delta t/\tau \) is negatively correlated with token duration, and therefore, positively correlated with speech rate. Thus, the change in the time constant appropriately reflects temporal variation for all three vowel classes.

\footnote{The correct network is the network that was trained on tokens of the same class as the test token. For example, for /ae/ test tokens, the correct network refers to the network that was trained on /ae/ tokens.}
Figure 4: Recognition performances for NoTC-All, Tau-Net, and NoTC-Med, using three different sets of random initial weights.

| EFFECT OF TIME CONSTANT ADAPTATION ON VOWEL RECOGNITION ERROR RATES (%) |
|-----------------|-----------------|-----------------|-----------------|
| WEIGHTS  | VOWEL  | TC FIXED | TC ADAPTED |
| wts.0     | /ae/    | 27.27     | 9.09         |
|           | /iy/    | 6.67      | 6.67         |
|           | /ux/    | 28.57     | 0.00         |
|           | /ae+iy+ux/ | 20.00   | 5.00         |
| wts.1     | /ae/    | 45.45     | 0.00         |
|           | /iy/    | 6.67      | 6.67         |
|           | /ux/    | 14.29     | 7.14         |
|           | /ae+iy+ux/ | 17.50   | 5.00         |
| wts.2     | /ae/    | 18.18     | 0.00         |
|           | /iy/    | 6.67      | 0.00         |
|           | /ux/    | 42.86     | 14.29        |
|           | /ae+iy+ux/ | 22.50   | 5.00         |

Table 2: Benefit of Time Constant Adaptation for Vowels – Comparing recognition error rates for Tau-Net before and after time constant adaptation for three different sets of random initial weights.
Figure 5: Correlation between adapted time constants and test token durations for the correct networks. Results are shown for all sets of initial weights.
4.4 Consonant Recognition

4.4.1 Consonant Data

In the consonant experiment, the task was to distinguish between the three consonants /p, t, k/. Both closure and release parts were included for each consonant token. For this experiment, data of male speakers from four different dialect regions in TIMIT were used: regions 2 (Northern), 3 (North Midland), 7 (Western), and 8 (Army Brat). Contexts for training and test consonant tokens were constrained by extracting only /p, t, k/ tokens that precede vowels in the set /æ, i, ɯ, u/. As with the vowel experiment, subsets containing only medium-rate tokens were extracted from the training data. Here, medium-rate tokens are defined to be tokens of average duration (for each particular consonant class), tokens that are one frame longer, and tokens that are one frame shorter than the average duration. Training sets (not including tokens used for validation) consisted of 63, 88, and 86 tokens for /p, t, k/, respectively, of which 25, 30, and 27 tokens were of average duration. Test sets contained 17, 34, and 34 tokens. Table 3 shows the distribution of fast, medium, and slow tokens for training and test sets for the three consonants. For the three consonant classes combined, training data consisted of 237 tokens from 169 speakers, and test data consisted of 85 tokens from 36 speakers.

<table>
<thead>
<tr>
<th>CONSONANT</th>
<th>DATA SET</th>
<th>FAST</th>
<th>MEDIUM</th>
<th>SLOW</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>/p/</td>
<td>Train</td>
<td>20</td>
<td>25</td>
<td>18</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>/t/</td>
<td>Train</td>
<td>26</td>
<td>30</td>
<td>32</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>14</td>
<td>7</td>
<td>13</td>
<td>34</td>
</tr>
<tr>
<td>/k/</td>
<td>Train</td>
<td>28</td>
<td>27</td>
<td>31</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>8</td>
<td>9</td>
<td>17</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 3: Consonant Data — Distribution of fast-, medium-, and slow-rate tokens for training and test sets for the three consonant classes. NoTC-All was trained on all training tokens. Tau-NET and NoTC-MED were trained on medium-rate tokens.

4.4.2 Consonant Experiment Setup

The same training and testing procedures employed in the vowel experiment were used here: Each network was trained using the predictive approach to recognize a different consonant, and classification of a test token was determined by the network with the smallest prediction error. The final set of weights used for testing for each network was determined by the lowest prediction error on the validation set. Again, performance of the system of Tau Nets (Tau-NET) was compared to networks without time constants trained on all tokens (NoTC-All) and to networks without time constants trained on medium-rate tokens (NoTC-MED). As with the vowel experiment, all systems used the same sets of initial random weights, and all systems were evaluated on the same test sets.

The network architecture used for the consonant recognition task is shown in Figure 6. The network structure consisted of a three-frame input and a fully recurrent hidden layer of 10 units; the network thus had 45 input units, 10 hidden units, and 15 output units. A multiple-frame input was important for making the transient nature of consonants more explicit to the network. Also, since consonants can contain abrupt acoustic changes, the networks were trained to predict only one time step ahead instead of two as with the vowel data.
4.4.3 Consonant Recognition Results

As before, recognition performance was the criteria used to evaluate the networks. The recognition performance for a network was determined by its prediction performance on a test token relative to the prediction performance of the other networks. A network that was trained on tokens of the same class as the test token should have the lowest prediction error on that test token; otherwise, a recognition error occurs.

Three sets of random initial weights were used for the consonant experiments. The number of training iterations, averaged over all initial weights and all consonant classes, were 216, 229, and 296 for Tau-Net, NoTC-ALL, and NoTC-MED, respectively. Recognition results for all three sets of random initial weights are shown in Figure 7. On the consonants, Tau-Net achieved nearly the same recognition error rates as NoTC-ALL for two initial conditions (wts.0 and wts.2), and higher recognition error rates for the third. For all three initial conditions, Tau-Net produced fewer errors than NoTC-MED. As before, Tau-Net and NoTC-MED were trained only on medium-rate tokens, which made up approximately 35% of the available training data. Even so, recognition results show that Tau-Net achieved nearly the same recognition performance as NoTC-ALL, indicating that Tau-Net was able to recognize tokens at various rates, even rates that were not in the training data. These results are consistent with the vowel results.

The effect of adapting the time constants on error rates for the consonants are shown in Table 4. Results for all three sets of initial random weights are shown. As in the vowel experiment, the number of misclassifications was reduced after time constant adaptation. However, the reduction in error was not as significant as with the vowel data.
Figure 7: Recognition error rates for NoTC-All, Tau-Net, and NoTC-Med on consonants, using three different sets of random initial weights.

| EFFECT OF TIME CONSTANT ADAPTATION ON CONSONANT RECOGNITION ERROR RATES (%) |
|-------------------------------|----------------|----------------|----------------|
| WEIGHTS | CONSONANT | TC FIXED | TC ADAPTED |
| wts.0   | /p/ | 29.41 | 35.20 |
|         | /t/ | 58.82 | 58.82 |
|         | /k/ | 44.12 | 38.24 |
|         | /p+t+k/ | 47.06 | 45.88 |
| wts.1   | /p/ | 23.53 | 41.48 |
|         | /t/ | 67.65 | 61.76 |
|         | /k/ | 52.94 | 38.24 |
|         | /p+t+k/ | 52.94 | 48.24 |
| wts.2   | /p/ | 29.41 | 17.65 |
|         | /t/ | 64.71 | 58.82 |
|         | /k/ | 44.12 | 41.18 |
|         | /p+t+k/ | 49.41 | 43.53 |

Table 4: Benefit of Time Constant Adaptation for Consonants – Comparing recognition error rates for Tau-Net before and after time constant adaptation for three different sets of random initial weights.
There was again strong correlation between the change in the time constant and temporal variability in test token durations, as illustrated in Figure 8. Results are shown for all three sets of initial weights. Correlation between \( \text{deltat}/\tau \) and normalized token duration is -0.9393, -0.6869, and -0.8451 for /p/, /t/, and /k/ respectively. If outliers were excluded in the computation of the correlation coefficients, correlation would be -0.9393, -0.9005, and -0.9336 for /p/, /t/, and /k/, respectively. An outlier is defined to be a sample point that is more than two standard deviations away from the regression line, \( y = 1 - 0.5x \), which describes perfect negative linear correlation between \( \Delta t/\tau \) and token duration. There were no outliers for /p/, one for /k/, and five outliers (resulting from two tokens) for /t/. Later inspection revealed that these outliers were quite possibly simply bad tokens; their prediction errors were unusually high for all three systems and across all three sets of initial weights.

Consonant recognition results are not as strong as with the vowel experiment in that recognition error rates were higher and did not decrease as significantly for consonants as for vowels. This is expected, however, as temporal variability in speech is captured mostly in vocalic segments, not consonants, so compensating for temporal variability should improve recognition more for the vowel data than for the consonant data. Also, acoustic features of consonants are transient and are highly influenced by the following vowel; consequently, there may be context effects other than rate variation contributing to the high error rates for the consonant data. Furthermore, the consonant data was much less constrained then data used in the vowel experiment. First, contexts for vowel tokens were more constrained (specific left and right contexts) than for consonant tokens (no left context and multiple right contexts). Second, the consonant data was taken from four different dialect regions instead of two regions for the vowels; this introduces variation due to not only speaker differences but also to dialect differences. Finally, acoustic features of consonants are much less distinctive than for vowels, resulting in more confusability among the different classes. Figure 9 illustrates the difference in acoustic features for vowels versus consonants. Spectrograms in the figure show that distinguishing features among vowel classes are strong and distinctive, whereas distinguishing feature among consonant classes are weak and subtle.

In addition, since acoustic features of consonants are very subtle, and because of their brief and transient nature, consonants are often characterized by their effects on the adjacent vowel rather than by their acoustic features [5]. The fact that token segments contained only the consonant and excludes the following vowel context (since phone endpoints provided in TIMIT were used to extract tokens) makes the identification of these consonant tokens extremely difficult. All these factors contributed to the increased difficulty of the consonant recognition task, resulting in the high error rates seen in Figure 7. The small reduction in recognition errors after time constant adaptation, as shown in Table 4, can also be attributed to the high degree of confusability among consonants. The distinguishing cues among consonant classes can be sufficiently subtle that a misclassification can still occur even if the network can compute the correct rate and can track the acoustic vectors perfectly. Note that even though time constant adaptation helped recognition performance overall, for /p/ tokens with initial weights WTS.0 and WTS.1, the classification error rates actually increased after the time constants were adapted. A comparison of validation errors indicated that, for all sets of initial weights, prediction errors on validation sets for the networks trained on /p/ tokens were higher than for networks trained on /t/ and /k/ tokens; that is, networks trained on /p/ tokens did not generalize as well. This suggests that Tan Net must be trained to model a signal sufficiently well in order for time constant adaptation to be effective in compensating for temporal variation during recognition.
Figure 8: Correlation between adapted time constants and test token durations for consonant tokens. Results are shown for all sets of initial weights.
Figure 9: Vowels vs. Consonants — Spectrograms showing differences in acoustic features for vowels and consonants. Distinguishing features for vowels are strong and distinctive. In contrast, distinguishing features for consonants are weak and subtle.
5 Conclusion & Future Work

We have presented results of applying Tau Net to a speaker-independent vowel recognition task and a speaker-independent consonant recognition task. On both discrimination tasks, Tau Nets, trained on medium-rate tokens, achieved the same performance as networks without time constants trained on all tokens for two out of three sets of initial weights. Results for Tau Nets were slightly worse on a third set of initial weights for both vowel and consonant tasks. However, this illustrates that Tau Nets can perform comparably as networks trained on much more data. On both tasks, Tau Nets performed better than networks without time constants trained on medium-rate tokens. These results demonstrate Tau Net’s ability to identify vowels and consonants at variable speech rates by extrapolating to rates not represented in the training set.

Tau Net employs a predictive approach to explicitly model temporal correlations in the signal, and adaptable time constants to address temporal variability. The adaptable time constants give Tau Net the ability to recognize the same signal at different time scales. This ability can be exploited in two ways, as was seen with the experiments using speech data: (1) Tau Net can be trained using significantly less data than a normal network without time constants, and (2) Tau Net can extrapolate to signals with durations (rates) not represented in the training set. The adaptation of the time constant during recognition is inexpensive computationally, involving only a one-dimensional minimization problem for each network. Also, an additional benefit of the time constant is that its adaptation provides a measure of the temporal variation of the signal. Although the work presented here applies to speech, Tau Net is a general approach for modeling variable-rate dynamic signals and is applicable to other domains as well. Furthermore, Tau Net can also be used as a simple way to achieve rate adaptation in a larger system.

Many further developments with Tau Net are possible. One option to explore is the use of discriminant training. As with many other approaches using one model per class, the setup as described suffers from weak discrimination power since each model is trained only on positive examples. Discriminant training allows for negative examples to be used, thus providing better region boundaries between the classes and should yield better classification, especially for classes with few clearly distinct features such as consonants. Mellouk & Gallinari [9] have investigated the use of discriminant training in the predictive framework. In this approach, the output of the network is converted into a scalar that represents the prediction error. Each network is then trained on the following cost function, which is related to the prediction error:

\[
P_k = \frac{e^{-D_k}}{\sum_j e^{-D_j}},
\]

where \(D_k\) is the prediction error, defined as the Euclidean distance between the desired and actual outputs of network \(k\). Discriminant training can then be accomplished by training the correct network with the target of 1, and all incorrect networks with the target of 0. Computational costs can be decreased by performing discriminant training on only those networks whose classes are the most confusable, i.e., networks that produce similar predictions for the same class of tokens.

Another interesting option to investigate is to adapt the time constant during training as well, not just during recognition. This would allow the time constant to settle to an optimal value for the training data, and adaptation during recognition would start from this value instead of the somewhat arbitrary value of \(\Delta t/\tau = 0.5\). This should strengthen the correlation between the time constant and temporal variation in the signal and should lead to better performance during recognition. However, preliminary experiments along this line suggest that care must be taken in the manner in which the time constant is adapted with respect to the weights. For example, it was discovered that it is generally better to keep the time constant fixed at least during the initial part.
of training to allow the weights to encode the general structure of the training exemplars. Only after the training error has leveled off should the time constant be adapted along with the weights. The idea is to let the weights encode the temporal and spectral structure of the training exemplars, and force the time constant to encode and accommodate for temporal variation only. It may also be beneficial to tip the balance of relative importance to weigh the error gradient with respect to the network weights more heavily in minimizing the error during the initial part of training, and gradually change this balance towards the error derivative with respect to the time constant as training progresses. When both the time constant and weights are adapted during training, a principled way is needed to ensure that changes to the time constant are attributed to errors due to temporal variation and not to other factors.
References


