A Habituation Based Neural Network for Spatio-Temporal Classification

Bryan W. Stiles and Joydeep Ghosh *
Department of Electrical and Computer Engineering
The University of Texas at Austin

Contents

*This work was supported in part by an NSF grant ECS 9307632 and ONR contract N00014-92C-0232. Bryan Stiles was also supported by the Du Pont Graduate Fellowship in Electrical Engineering. We thank Prof. I. Sandberg for several fruitful discussions.

List of Figures

Abstract

A novel neural network is proposed for the dynamic classification of spatio-temporal signals. The network is designed to classify signals of different durations, taking into account correlations among different signal segments. Such a network is applicable to SONAR and speech signal classification problems, among others. Network parameters are adapted based on the biologically observed habituation mechanism. This allows the storage of contextual information, without a substantial increase in network complexity. Experiments on classification of high dimensional feature vectors obtained from Banzhaf sonograms, demonstrate that the proposed network performs better than time delay neural networks while using a substantially simpler structure. A mathematical proof is provided which demonstrates that a certain set of generalized neural network structures, including the habituation based design, are capable of approximating arbitrarily well any continuous, causal, time-invariant, mapping from one discrete time sequence to another.

Keywords: dynamic neural networks, habituation, classification, spatio-temporal signals, recurrent networks

1 INTRODUCTION

Many tasks performed by humans and animals involve decision-making and behavioral responses to spatio-temporally patterned stimuli. Thus the recognition and processing of time-varying signals is fundamental to a wide range of cognitive processes. Classification of such signals is also basic to many engineering applications such as speech recognition, seismic event detection, sonar classification and real-time control [19],[20].

A central issue in the processing of time-varying signals is how past inputs or "history" is represented or stored, and how this history affects the response to the current inputs. Past information can be used explicitly by creating a spatial, or static, representation of a temporal pattern. This is achieved by storing inputs in the recent past and presenting them for processing along with the current input. Alternatively, the past events can be indirectly represented by a suitable memory device such as a series of possibly dispersive time-delays, feedback or recurrent connections, or changes in the internal states of the processing *cells* or "neurons" [12], [20].

The past few years have witnessed an explosion of research on neural networks for temporal processing, and surveys can be found in [13], [20], and [21] among others. Most of this research has centered on artificial neural network models such as time delayed neural networks, the recurrent structures of Elman and Jordon, etc., that can utilize some variant of the backpropagation algorithm. Another large class of networks, inspired by physics, are based on transitions between attractors in asymmetric variants of the Hopfield type networks [3],[16].

Some researchers have also studied spatio-temporal sequence recognition mechanisms based on neurophysiological evidence, especially from the olfactory, auditory and visual cortex. Representative of this body of work is the use of non-Hebbian learning proposed by Granger and Lynch, which, when used in networks with inhibitory as well as excitatory connections, can be used to learn very large temporal sequences [14]. Similar networks have been used to act as adaptive filters for speech recognition [17], or provide competitive-cooperative mechanisms for sequence selection [7]. At the neuronal level, irregularly firing cells have been proposed as a basic computational mechanism for transmitting temporal information through variable pulse rates [6]. All these efforts are concentrated on neurophysiological plausibility rather than being geared toward algorithmic classification of man-made signals. Issues such as computational efficiency and ease of implementation on computers, are secondary. In contrast, we introduce a new mechanism in this paper which has its roots in biological systems, but is adapted for efficient and practical applications involving temporal processing.

Among biological mechanisms that can encode temporal information, is a particularly simple and well understood phenomenon known as *habituation* [1], [4], [23],[27]. Primarily, habituation is a means by which biological neural systems vary their synaptic strengths in order to ignore repetitive, irrelevant stimuli. Habituation serves as a novelty filter. If the presynaptic neuron is active for a short period of time, habituation tends to decrease the synaptic strength which then recovers only after the period of activity is over. The longer the presynaptic neuron is active the slower it recovers. It is important to note that habituation does not act in a vacuum. Other learning mechanisms, such as sensitization and Hebbian learning, may also be operating concurrently to alter synaptic strengths based on the utility of the information from presynaptic neurons.

Several researchers in neurophysiology have developed mathematical models of habituation [1], [4], [27]. A discrete time version of the Wang Arbib [27] habituation model for varying the strength, $W(t)$, of a single synapse is summarized by:

$$W(t+1) = W(t) + \tau(\alpha z(t)(W(0) - W(t)) - W(t)I(t)); \tag{1}$$

$$z(t+1) = z(t) + \gamma z(t)(z(t) - 1)I(t). \tag{2}$$

In this model, I(t) is the activation of the presynaptic neuron at time t, τ is a constant used to vary the habituation rate and α is a constant used to vary the ratio between the rate of habituation and the rate of recovery from habituation. The function $z(t)$ monotonically decreases with each activation of the presynaptic neuron. This function is used to model long term habituation. Due to the effect of $z(t)$, after a large number of activations of the presynaptic neuron, the synapse recovers from habituation more slowly.

Aside from its primary function, habituation has also been suggested to be a means of encoding short term temporal information [23]. In this paper, we introduce a mechanism for using habituation to encode temporal information in an artificial neural network. In Section 2 we describe our mechanism, and provide a mathematical proof of its capabilities. We demonstrate that our mechanism is a special case of a general neural network structure which is capable of approximating arbitrarily well any continuous, causal, time-invariant, mapping from one discrete time sequence to another. Finally we explain how it fundamentally differs from the gamma network model of de Vries and Principe [9]. In Section 3, we discuss the general issues involved in spatio-temporal processing with artificial neural networks, motivate the data sets used, and give experimental results for our network on the classification of artificial Banzhaf sonograms. We demonstrate that our network is more efficient than TDNNs for a number of classification problems involving long term temporal information. Finally in Section 4, we draw conclusions based upon our theoretical and experimental results.

2 A HABITUATION MECHANISM FOR ENCODING TEMPORAL INFORMATION

2.1 General Structure

We have designed short term habituation units based upon the Wang and Arbib model of habituation and used them in a spatio-temporal classification network. A set of habituated weights is first obtained from the input $I(t)$. If the input is multi-dimensional, one set is extracted for each component. These weights are affected by the past values of the input, and implicitily encode temporal information. Spatio-temporal classification can thus be achieved by using such habituated weights as inputs to a static classifier. For example, if a multilayered perceptron (MLP) (alt. radial basis function network) is used, the overall network is a habituated MLP (alt. habituated RBF) that can be applied for spatio-temporal classification. The model equation is shown as follows.

$$W_k(t+1) = W_k(t) + \tau_k(\alpha_k(1 - W_k(t)) - W_k(t)I(t)) \tag{3}$$

This equation is derived from Equation 1 by setting $z(t) = 1$ to eliminate long term habituation effects, and letting $W_k(t)$ rebound to 1 instead of $W_k(0)$. Long term habituation is eliminated so that the ability of $W_k(t)$ to recover from habituation does not vary over time. Otherwise the $W_k(t)$ values would eventually decrease to zero for all but the most infrequent of inputs. The k index is used to indicate that multiple values $W_k(t+1)$ are determined for an input signal $I(t)$. It was found mathematically, that multiple habituation values are better able to encode temporal information. This fact may also have biological context, because it is known that a given pair of neurons often have multiple synapses between them.

In this paper, dynamic classification is achieved by training a suitable nonlinear feedforward network, whose inputs are a set of m habituated values, $W_k(t+1)$, $1 \leq k \leq m$, that are extracted from the raw input $I(t)$. Figure 1 shows the generic structure of such a classifier. In [27] $W_k(t)$ represents a synaptic strength, and $I(t)$ the activity of the presynaptic neuron, but because our designs use habituated values as network inputs rather than weights, the variables are redefined accordingly. We do not mean to imply that this network construction is either the most biologically feasible or the only method in which habituation might be used. A more biologically inspired approach would be to reflect $W_k(t)$ as modulating weights of the inputs. We found by experiment, however, that this approach, although more biologically feasible, does not encode temporal information as well for the classification problems which we studied. Moreover, the structure of Figure 1 can be shown mathematically to be very powerful.

The parameters, τ_k and α_k affect the rate at which habituation occurs, thereby determining the temporal resolution and range of the information obtained. The issues and tradeoffs involved are akin to memory depth versus resolution in dispersive delay line based models [9], [21]. We set $W_k(0)$ to zero for all k, employ positive values of α_k and τ_k such that $\alpha_k\tau_k + \tau_k < 1$ and normalize the input such that $I(t) \in [0,1]$. With these specifications, we can guarantee that the habituation process is stable. In fact we can guarantee that $W_k(t) \in [0,1]$ for all values of k and t.

2.2 Mathematical Properties

The following is a mathematical proof of the ability of a general category of neural networks, including habituation based networks, to approximate arbitrarily well any continuous, causal, time-invariant mapping f from one discrete sequence to another. Since all functions realized by TDNNs with arbitrarily large but finite input window size are continuous, causal, and time-invariant, the proof given also implies that habituation based networks can realize any function which can be realized by a TDNN [24]. The key to the proof is to show that the memory structure realized by the habituated weights is a complete memory. Then so long as the feedforward stage is capable of uniformly approximating continuous functions, the overall network will be capable of mapping one sequence to another.

First, the terms causal and time-invariant are defined. Let X be the set of mappings from the set of integers, $\mathcal{Z}$, to the set of real numbers, $\mathcal{R}$, with the following constraints. For all $I \in X$, $I(t) = 0$ if $t \leq 0$; and $\forall t \in \mathcal{Z}$, $I(t) \in [0,1]$. Let R be the set of all mappings from the positive integers, $\mathcal{Z}^+$, to $\mathcal{R}$. Let the operator T_β be a mapping from X to X defined as follows. For all $x \in X$, and t and $\beta \in \mathcal{Z}^+$,

$$(T_\beta x)(t) = \begin{cases} 0 & \text{if } t < \beta \\ x(t-\beta) & \text{otherwise} \end{cases}$$

We say a mapping M from X to R is *time-invariant* if for each $\beta \in \mathcal{Z}^+$ and each $x \in X$ the following is true.

$$(MT_\beta x)(t) = \begin{cases} 0 & \text{if } t < \beta \\ (Mx)(t-\beta) & \text{otherwise} \end{cases}$$

Let the mapping P_β from X to X be defined as follows.

$$(P_\beta x)(t) = \begin{cases} 0 & \text{if } t > \beta \\ x(t) & \text{otherwise} \end{cases}$$

Let C_a denote the intersection of the two sets, $[0, a]$ and $\mathcal{Z}^+$. We say a mapping M from X to R is *causal* if $\forall a \in \mathcal{Z}^+$, the statement $P_a x = P_a y$ implies $(Mx)(t) = (My)(t)$ for all $t \in C_a$. For a causal M, the value of the sequence Mx at any point t is independent of the future values of x.

We are attempting to approximate a continuous, time-invariant, causal function, f, from X to R. We know additionally that any TDNN can be represented by such a function. As illustrated by Figure 1, we suggest transforming the inputs with m habituation units. After habituating, it is our conjecture that f can be approximated arbitrarily well by an MLP or RBF which takes the habituated values, $W_k(t+1)$, as inputs. If our conjecture is true, then a habituation based network is able to realize any function realizable by a TDNN.

First we shall show that a two layer neural network with an exponential activation function and a general structure for processing the inputs can universally approximate f. Then we will show that habituation based networks are a specific case of the generalized structure.

The results achieved in this manner can be readily extended to include habituated MLP and RBF networks and to include multiple ($d > 1$) spatial input dimensions, $I_h(t)$, $1 \leq h \leq d$. Rigorous proofs of these extensions are not given for the sake of conciseness. In order to show that habituated MLPs and RBFs can perform the same approximations it is sufficient to show that the exponential function can be approximated arbitrarily well by a summation of sigmoids or gaussian functions. This is a special case of theorems which have already been proven for sigmoids by Cybenko [5] among others and for gaussian functions by Park and Sandberg [22]. The expansion of the result to multiple spatial dimensions follows directly from the proof of Theorem 1.

Theorem 1 *Let f be a continuous, causal, time-invariant function from X to R, and let B be a set of mappings with the following four properties: First, there exist real numbers a and c such that $(bx)(t) \in (a, c)$ for all $t \in \mathcal{Z}^+$, $x \in X$, and $b \in B$. Second, for any $t \in \mathcal{Z}^+$ and any t_0 such that $0 < t_0 \leq t$, the following is true. If x and y are elements of X and $x(t_0) \neq y(t_0)$, then there exists some $b \in B$ such that $bx(t) \neq by(t)$. Third, if $b \in B$ then $(bT_\beta x)(t) = (bx)(t-\beta)$ for all $t \in \mathcal{Z}^+$, all $x \in X$ and any β such that $0 \leq \beta \leq t$. Fourth, every $b \in B$ is causal. If all these properties are met by B, then we say B is a* complete memory. *If B is a complete memory then the following is true. Given any $\epsilon > 0$ and any arbitrarily large positive integer, t_0, there exist real numbers, a_j and c_{jk}, elements of B, b_{jk}, and natural numbers, p and m, such that the following inequality holds for all $x \in X$ and all $t < t_0$.*

$$\left| (fx)(t) - \sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk} (b_{jk} x)(t) \right) \right| < \epsilon \qquad (4)$$

A proof of this theorem can be found in the appendix.

By proving Theorem 1 we demonstrate that a two layer static neural network with an exponential activation function and inputs operated on by elements of a complete memory, B, can perform the same function as any TDNN. In order to show that a habituation based network is a special case of this type of generalized structure we state the following theorem which is also proven in the Appendix.

Theorem 2 *Let $b_0 = W(0) = 1$. A prototypical habituation function is defined recursively as follows.*

$$bx(0) = W(1) = b_0 + \alpha\tau(1 - b_0) - \tau b_0 x(0) \tag{5}$$

$$bx(t) = W(t+1) = bx(t-1) + \alpha\tau(1 - bx(t-1)) - \tau bx(t-1)x(t) \tag{6}$$

Let B be the set of all such functions for all α and $\tau \in \mathcal{R}$ such that $\alpha > 0$, $\tau > 0$, $\tau < 1$, and $\alpha\tau + \tau < 1$. B is a complete memory.

It is important to notice that the input processing functions, b_{jk} used in Theorem 1 depend on j and thus the habituation parameters used also depend on j. This means that different hidden units in the feedforward network may have different input values. This dependency is not present in the structure illustrated in Figure 1. However, we can show that for any approximation g of the form discussed in Theorem 1, there is an equivalent network without this dependency. Let g be an approximation function of the form $\sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk} b_{jk} x\right)$. It is easy to see that given any such g one can find an h of the following form such that $g(x) = h(x)$ for all $x \in X$.

$$h(x) = \sum_{j=1}^{p} a_j \exp\left(\sum_{i=1}^{M} w_{ji} s_i(x)\right) \tag{7}$$

Here w_{ji} are real numbers which serve as weights to the hidden units and s_i are elements of a complete memory B.

Simply choose M to be the number of distinguishable functions b_{jk} used in g and let the sequence $\{s_i\}$ be the list of these distinguishable functions. For a particular s_i and a particular hidden node j, set w_{ji} to zero if the original b_{jk} corresponding to s_i was not present at hidden node j, otherwise set w_{ji} to the appropriate c_{jk}. An approximation of the form given by h has the same structure given in Figure 1, so the structure illustrated in Figure 1 is adequate.

Now we have demonstrated that habituated MLPs and RBFs are satisfactory substitutes for TDNNs. The question that remains is which are more efficient. The answer depends on the nature of the function that is being realized. The complexity of the TDNNs depends on n, the input window size. The number of weighted inputs to each hidden unit in a TDNN is nd. For functions which only depend on recent values of the inputs, TDNNs can be quite efficient; but for functions which depend on long term temporal information or variable amounts of temporal information, TDNNs are not efficient solutions. For habituated networks, the required memory depth and resolution affects the choice of α and τ in Equation 3, and the number of habituated weights. Differents weights can have different values of α and τ, and the number of weights used can vary in different dimensions. Thus the memory structure can be optimized for a given mapping. The number of inputs to each hidden unit is $\sum_{i=1}^{d} m_i$, where m_i is the number of habituation values used to encode the ith component of $I(t)$.

A parallel can be drawn between finding a suitable m and finding a suitable number of hidden units in an MLP or RBF. In both cases there is no guarantee that the number required will not be inordinately large. However, in the case of the MLP, a large set of problems have been found for which a small number of hidden units is suitable. The same is true for finding m. There may be many simple problems which are unsuitable for TDNNs because they require long term temporal information, but which can be solved with habituated networks with small values of m.

Since the output of the short-term memory stage is different for TDNNs and habituated networks, the complexity (number of hidden units) of the feedforward network needed at the output stage may also differ. For certain problems, habituated networks require a smaller feedforward output stage as compared to TDNNs for a given level of approximation. We have previously performed experiments using habituated MLPs to classify real SONAR data and have found that small habituated networks outperformed larger TDNNs. In fact we found that even $m = 1$ networks dramatically outperformed TDNNs with time window length of 5 or more [25], [26]. Unfortunately due to the proprietary nature of the real SONAR data sets, they cannot be made public. Therefore, in the next section, we discuss experimental results on artificial Banzhaf sonograms, which can be easily generated and verified by other researchers.

An interesting side issue, is that habituation is not the only input transformation mechanism that might be used to reduce input dimensionality. For example, consider the set of mappings B such that any $b \in B$ is given by $bx(t) = W^*(t+1)$ with W^* defined as follows.

$$W^*(t+1) = \alpha(W^*(t) + x(t)). \tag{8}$$

In this case B can also be shown to be a complete memory for $\alpha \in (0, .5)$. However, we have not yet performed any experimental work using W^* or any complete memory other than habituation. The Gamma network proposed by Principe [9] also has a similar structure in that raw inputs are transformed and then fed into a feedforward neural network, but there are significant differences. For one, unlike habituation, the input transformation used in the Gamma network is linear. Secondly, in the Gamma network each transformed input, W_i, depends on the previous transformed input, W_{i-1}, but in a habituated network each W_i is generated independently from the raw inputs.

3 EXPERIMENTAL RESULTS

Before describing the data sets used and providing comparative experimental results, some fundamental issues and terminology pertaining to classification of spatio-temporal signals are summarized in Section 3.1.

3.1 Basic Issues in Spatio-Temporal Classification

The signals of interest in speech or sonar processing exhibit time-varying spectral characteristics, and can also be of varying lengths or time durations. Before classification can be attempted, a signal has to be separated from background noise and clutter, and represented in a suitable form. In order to use a *static* neural network such as the MLP or RBF, the entire signal must first be described by a single *feature vector*, i.e., a vector of numerical attributes, such as signal energy in different frequency ranges or bands [11]. The selection of an appropriate set of attributes is indeed critical to the overall system performance, since it fundamentally limits the performance of any classification technique that uses those attributes [10].

The use of static classifiers is not ideal because representing each signal by a single feature vector results in a blurring of the spatio-temporal pattern characteristics and may lead to a loss of information that would be useful for discrimination. Another, more detailed way to view a time-varying nonstationary signal is to treat it as a concatenation of quasi-stationary segments [15]. Each segment can be represented by a feature vector obtained from the corresponding time interval. Thus the entire signal is represented by a sequence of feature vectors that constitute a spatio-temporal pattern. A popular way of representing such signals is by a two-dimensional feature-time plot called a spectrogram which shows the sequence of feature vectors obtained by extracting attributes from consecutive segments. The number of segments representing a signal is not necessarily pre-determined. Segmentation of nonstationary signals is a difficult problem and has been extensively studied [8].

Any neural network applied to the signal detection task must also address the following issues.

(i) *Cueing*: Generally, any network that performs a recognition task must be informed of the start and end of the pattern to be recognized within the input stream. If this is so, the classifier is called a cued classifier. An uncued classifier, on the other hand, is fed a continuous stream of data from which it must itself figure out when a pattern of interest begins and when it ends.

(ii) *Spatio-temporal Warping*: Often the original pattern gets distorted in temporal and spatial domains as it travels through the environment. A robust classifier must be insensitive to these distortions. If the classifier is not sophisticated in this aspect, then the distorted signal received needs to be preprocessed to annul the effects of the environment. This is called equalization and is widely used in communication systems. The characteristics of the communication or signal transmission medium are "equalized" by the preprocessor, so that the output of this processor is similar to the original signal. Alternatively, mechanisms such as the dynamic time warping algorithm [18] can be built into the classifier to make it less susceptible to the distortions.

Since a general purpose cued classifier must detect the presence of a signal as well as classify it, measures of quality include not only classification accuracy, but false alarm rate, the number of times the system indicates presence of a signal when there is none, and missed detection rate, the number of real signals that are present but not detected. Also important are the computational power required and confidence in classification decisions when they are made [11].

Another problem inherent in spatio-temporal classification, that is not present in static classifiers, is when to decide to classify a signal. With a static classifier, after each pattern is presented some decision must be made, but in spatio-temporal classification the network is presented sequentially with some number of subpatterns. The network must determine at what point it has enough information to classify a signal. One method of making this decision is to classify a signal only when a particular class membership is indicated over several consecutive subpatterns.

In this paper, a signal or pattern refers to a sequence of feature vectors. Each component feature vector is referred to as a sample or subpattern. Signals are interspersed by arbitrarily long noise segments, to mimic real-world problems.

Banzhaf signals, consisting of 30-dimensional feature vectors, with sequence length typically between 30-45, and including the effects of spatio-temporal warping, are used to evaluate the habituation networks. These signals are described in the next section. Also note that the proposed classifier is uncued and has to tackle both detection and classification. Therefore, it involves selection of thresholds to yield a range of missed detection rate vs. confidence in classification tradeoffs, as detailed in Section 3.3. False alarm rate is not reported, because the simplicity of our background noise model tends to produce artificially few false alarms.

3.2 Banzhaf Sonograms

This section describes Banzhaf sonograms [2], which were selected to obtain comparative experimental results on high dimensional spatio temporal classification. Banzhaf sonograms were used for several reasons. First, they are easily reproduced by other researchers. Secondly, they can be used to vary the difficulty of the data sets, the dependence of performance on temporal information, and the amount of warping in both time and space. Finally, it is known that superpositions of gaussian functions can be used to model any continuous spectrogram arbitrarily well and several interesting spectrograms (sonar, radar) are well modeled using small numbers of gaussian components.

A Banzhaf mother signal is generated by superposing two dimensional gaussian functions. The pth component gaussian, $G_p(x,t)$, is completely described by the constant parameters, $x_0[p]$, $t_0[p]$, $\theta[p]$, $h_x[p]$, $h_t[p]$, $\lambda_x[p]$, and $\lambda_t[p]$. The pair $(x_0[p], t_0[p])$ is the coordinates in time and space of the center of the gaussian, $\theta[p]$ is the angle of axial tilt, $h_x[p]$ and $h_t[p]$ determine the height of the peak, and $\lambda_x[p]$ and $\lambda_t[p]$ determine the width of the gaussian in space and time respectively. Values for each component gaussian, $G_p(x,t)$, are determined for integer values of x and t such that $0 \leq x \leq 29$ and $0 \leq t \leq 40$. For each value of x and t, the computation is performed in the following manner. First the following permutation is applied to rotate the gaussian by an angle of $\theta[p]$ about its center point, $(x_0[p], t_0[p])$.

$$\begin{aligned} x^* &= (x - x_0[p])\cos\theta[p] + (t - t_0[p])\sin\theta[p] + x_0[p] \\ t^* &= (x_0[p] - x)\sin\theta[p] + (t - t_0[p])\cos\theta[p] + t_0[p] \end{aligned} \tag{9}$$

Next the value, $G_p(x,t)$ is calculated as follows.

$$G_p(x,t) = h_x[p]e^{-\left(\frac{x^* - x_0[p]}{\lambda_x[p]}\right)^2} h_t[p]e^{-\left(\frac{t^* - t_0[p]}{\lambda_t[p]}\right)^2} \tag{10}$$

The Banzhaf mother signal, $G(x,t)$, is then computed by summing the component gaussians. In our experiment we generated six different mother signals using six different sets of parameter values. Each mother signal serves as the prototypical member of a class. Figure 2 shows the set of mother signals for each class. Table 1 lists the parameter values for the mother signals. The values for $\theta[p]$ are given in radians. Classes B and E have two component gaussians. All other classes have three. Training and test examples of each class were generated by adding uniformly distributed noise in the range $[-0.1, 0.1]$ to $G(x,t)$ and also perturbing the parameters in order to rotate, scale, and warp the mother signals to some extent. When the signals were time warped, the number of time samples calculated was varied accordingly, in order to create signals with a variable length in time. The number of spatial samples, however, was held constant at 30. A seventh "noise only" class was also constructed. The training and test sets were made up of 10 examples of each signal class and 9 examples from the "noise only" class. Once the data sets were compiled, they were normalized so that all data values were in the range [0,1]. Figure 3 gives typical examples of each signal class after rotating, scaling, and warping. The data set illustrated in Figures 2 and 3 is designated as data set 1 (DS1).

Clearly classification of DS1 is a problem which requires relatively long term temporal information. It is impossible to uniquely classify any signal based on only a short temporal window of inputs. For example, consider the mother signals of classes A, B, and C as illustrated in Figure 2. The signals in classes A and B are identical for the first twenty time samples, while classes A and C are identical for the last twenty time samples. Additionally, there is no time window of less than ten samples in any of the three signals that is not identical to a time window in one of the other two signals. This classification problem is obviously difficult for short window TDNNs. In order to demonstrate the effectiveness of habituation

Table 1: Parameters of Mother Signals of Data Set 1

	Class A	Class B	Class C	Class D	Class E	Class F
$x_0[1]$	7.5	7.5	15.0	7.5	15.0	15.0
$t_0[1]$	7.5	7.5	7.5	7.5	7.5	7.5
$\theta[1]$	-0.785	-0.785	0.0	-0.785	0.0	0.0
$h_x[1]$	1.0	1.0	1.0	1.0	1.0	1.0
$h_t[1]$	1.0	1.0	1.0	1.0	1.0	1.0
$\lambda_x[1]$	2.0	2.0	2.0	2.0	2.0	2.0
$\lambda_t[1]$	7.6	7.6	6.6	7.6	6.6	6.6
$x_0[2]$	15.0	15.0	15.0	15.0	15.0	15.0
$t_0[2]$	27.5	27.5	27.5	27.5	27.5	27.5
$\theta[2]$	0.0	0.0	0.0	0.0	0.0	0.0
$h_x[2]$	1.0	1.0	1.0	1.0	1.0	1.0
$h_t[2]$	1.0	1.0	1.0	1.0	1.0	1.0
$\lambda_x[2]$	2.0	2.0	2.0	2.0	2.0	2.0
$\lambda_t[2]$	9.2	9.2	9.2	9.2	9.2	9.2
$x_0[3]$	7.5		7.5	22.5		22.5
$t_0[3]$	27.5		27.5	27.5		27.5
$\theta[3]$	0.0		0.0	0.0		0.0
$h_x[3]$	1.0		1.0	1.0		1.0
$h_t[3]$	1.0		1.0	1.0		1.0
$\lambda_x[3]$	5.94		5.94	5.94		5.94
$\lambda_t[3]$	2.0		2.0	2.0		2.0

for problems with a range of difficulties we have constructed two other data sets which do not depend as severely on long term temporal information. Data set 2 (DS2) and data set 3 (DS3) were generated using the same parameters as DS1 except that the centers of the component gaussians were shifted to reduce the overlap among the classes. In DS2, the component gaussians of samples of a particular class were all shifted uniformly, whereas in DS3, individual component gaussians were shifted so that a particular gaussian component might act as a tag for identifying the class membership of the signal. For this reason DS3, is the most local temporal information rich of the three data sets, and one would expect TDNNs to perform relatively better on DS3 than on either of the other data sets.

3.3 Classification of Uncorrelated Sequences

For our first experiment, we trained habituated MLPs and TDNNs on DS1 and DS2. The patterns in both data sets were randomly shuffled so that the classification of each pattern was uncorrelated with the classification of nearby patterns in the sequence. When the habituated MLP was tested, the habituation values, $W_{hk}(t+1)$, were calculated at each instance in time and then fed into the feedforward portion of the network. At each instance in time an output vector was computed and then used in classification. As an optimization, only the habituation values computed for the last ten samples in each signal were used to train the habituated MLP. This reduced training set method was used because habituation gradually builds up information about a signal as the signal is presented. During the first few samples of a signal, a habituated MLP does not have enough information to classify a signal. By the end of the signal, however, the network should have accumulated enough information to perform the classification. The reduced training set method was not used for TDNNs, however, because they exhibit no similar dichotomy in the way they store information over time.

For the first set of experiments we used habituated MLPs with random values of α_{hk} and τ_{hk} in the range $[0, 0.5]$. The α_{hk} and τ_{hk} parameters were not modified during training. The number of habituation units per input, m, was set to one. We found that for DS1 the habituated MLP, (HMLP), greatly outperformed a 5 sample time window TDNN and an MLP. All three networks utilized 10 hidden units. Increasing the number of hidden units was not found to greatly effect the performance.

Classification and detection of signals is accomplished using two thresholds, H and L. Detection occurs whenever a single output node has an output value, O_{max}, larger than all other output nodes, $O_{max} > H$, and all other output values are less than $1 - H$, for L consecutive input presentations. Classification is considered to be correct for a given signal if the only class detected within the length of the signal is the desired class. The best values of H and L may vary from network to network. For a fair comparison, for each network one should select the L for which the network achieved its highest classification rate for some H.

Figures 4, 5, and 6 show the results for data set 1. The results given in Figures 4 and 5 are the percentage of signals detected, and among those signals the percentage classified correctly. Figure 6 illustrates performance in terms of the classification rate, i. e. the percentage of signals detected as well as correctly classified. Labels such as "L10" are included in the figures to denote the particular value of L used. As mentioned earlier, the best value of L was chosen for each classifier in order to make a fair comparison.

Results for DS2 similar to those found for DS1 are illustrated in Figure 7. For conciseness, this time the results are given in terms of the classification rate only. One notices in Figure 7, that although the HMLP has achieved a greater maximum result than the TDNN, it does not have a better result at every value of H. This is an artifact of the particular L values used. The greater the value of L the steeper the decrease in performance for increasing H. Figure 8 illustrates the effect of varying H and L on performance for both the TDNN and HMLP.

The HMLP was able to outperform the TDNN, because unlike the TDNN, the HMLP is capable of encoding long term temporal information. On DS1 this is particularly important, because classification is impossible without information from a large portion of the entire signal length.

3.4 Classification of Correlated Sequences

The improvement in performance of the HMLP as compared to the TDNN is even more noticeable on data sets where there is a temporal correlation between the the classifications of adjoining signals. In such data sets there is important information contained in the order in which signals of various classes occur. Speech processing is an example of data sets where order of occurrence information is important. Any TDNN of reasonable depth is unable to fully encode this type of information.

For our second experiment we examined the capability of the HMLP to learn information about the order in which signals are encountered. The network was again trained and tested on DS1 with one modification. In order to highlight the HMLP's ability to encode order of occurrence information, we examined the extreme case in which the patterns in both the training and test sets were arranged so that all patterns in class A appeared first, followed by all patterns in class B and so forth. The HMLP was able to use this additional ordering information to improve its performance on DS1 as shown in Figure 9. The percentage of signals which were both detected and classified correctly increased from 55.1 to 76.8 when the ordering information was used. Both of these results are much better than the best TDNN result of 39.3 percent. While the ability to encode such information is an advantage for HMLPs, it does add a complication in training. Since an HMLP can learn information about the order of occurrence of patterns in a training set, it may use such information inappropriately. When a training set is presented repeatedly to the network in the same order, the HMLP can use order of occurrence information to improve its performance on the training set. However, the order of occurrence of the patterns in a training set is often only an artifact of the manner it which the training set was assembled. In that case the use of such information will result in poor generalization to the test set. The obvious way to overcome this problem is to produce a larger training set which consists of multiple randomly shuffled copies of the original training set. Ideally, the expanded training set would be large enough so that only a single epoch would be needed for training. Unfortunately, such an expanded training set is impractical for large data sets, such as those used in this paper. For our experiments, we have used several more practical methods. One method is to use the unexpanded training set but to randomly shuffle the patterns after every 5 epochs. We used this method for all of our experiments on DS1 and DS2. Another method is to expand the training set but only by a factor of 5, so that 40 epochs of training is still required. We used this method on DS3.

3.5 Comparison of Different Training Methods

As previously mentioned, one method for improving HMLP performance is to train only on the last few habituated feature vectors in each signal. In the results discussed so far the last ten samples of each signal were used to train the HMLP. We shall henceforth refer to this method as the reduced training set (RTS) approach. Figure 10 demonstrates how the number of training samples per signal affects the performance on data set 1 as measured by the mean square error at each subpattern. Because, signals have variable length some care must be taken to make the numbering of the subpatterns uniform. Subpattern 45 of any signal refers to the last subpattern. Subpattern k for $k < 45$ refers to the subpattern $45 - k$ subpatterns from the end of the signal. It is apparent from the figure that the reduction of the number of training samples is only useful up to a point. For too few training samples, generalization may suffer. Also, for few training samples, the performance peaks sharply at the end of the signals, and the performance on the rest of the signal is unpredictable. On the other hand, training on the entire signal is suboptimal, because too much effort is wasted trying to classify the first few samples of each signal, at which time there is not enough information to make a correct classification. It may be that by training on the first few samples of the data, the network is forced to inappropriately use pattern ordering information as the only available means of reducing MSE on those samples. As previously discussed, this may inhibit generalization to the test set.

An alternative training method to RTS is to apply ramping in the desired outputs as depicted in Figure 11. With ramping, the network is trained to gradually change from one signal classification to another. At the beginning of each pattern all the desired output values ramp toward a quiescent value, β, in R time steps. After S time steps, the desired output, O_{new}, associated with the new pattern begins to increase toward one, while all other desired outputs decrease toward zero. After R additional time steps O_{new} reaches one. The β value is the reciprical of the number of classes, including the noise class. Because of this choice of β and since all the rampings are linear, the sum of the desired outputs is always one. This is required so that the outputs of a well-trained network can estimate the aposteriori probabilities of each class with respect to the input. In practice we set $\beta = 0$ and found that the performance of the networks was unaffected. The ramping method has an advantage over RTS in that the behavior of the network on the first few samples of a new pattern is more predictable. This is true because with ramping the network is trained to produce low output values for the first few training samples, while if RTS is used the network is not trained on these samples at all. On the other hand, ramping requires two parameters R and S whereas RTS requires only the number of training samples per signal. Additionally, training on the first few samples may require the inappropriate use of pattern order information even if the desired output is zero. It may be undesirable for the network to try to determine whether or not it is at the beginning of a pattern.

Figure 12 illustrates the effects of the ramping parameters R and S on the HMLP's performance. Ramping worked best for intermediate values of S and R. The results of comparisons between RTS and ramping were mixed. RTS seemed to do slightly better on DS2, while ramping performed slightly better on DS3. Apparently, habituation does not do as well on DS3 as on the other data sets. This result is reasonable because of the three data sets DS3 is the one which is least dependent on long term temporal information. DS3 can be fairly well classified with only local time information. For such a data set, TDNNs perform adequately well. The results, however, are still encouraging for habituation, because the HMLP is much simpler in structure than a TDNN of depth 5. The HMLP has much fewer trainable parameters. Even if the α and τ parameters are considered, the HMLP has less than one fourth as many parameters. Additionally, the habituation parameters were randomly assigned and never trained in all the experiments illustrated so far. Despite the simplicity of the HMLP, it outperformed much more complex TDNNs on DS1 and DS2, and performed similarly to the TDNN on DS3.

3.6 Using Multiple Habituation Units for Each Input

So far all the experiments discussed have focused on HMLPs with m=1. The mathematical proof which we demonstrated in Section 3, required that m be variable. Values of $m > 1$ make the network more powerful and may allow improvements in performance. In order to examine this possibility, we now consider the effect of increasing m to construct multiply habituated MLPs, (MHMLPs). This experiment was performed using data set 1. It was found that increasing m did lead to improved performance. HMLPs with $m = 2$, $m = 3$, and $m = 5$ achieved classification rates of 59.4, 56.5, and 58.0 respectively as compared to 55.1

for the m=1 case. On DS3 an MHMLP with m=2 performed as well as the 5 time sample TDNN. Both networks achieved a classification rate of 76.8.

3.7 Hybrid Habituated TDNN Network

Another type of network which bears investigation, is the hybrid habituated TDNN, (HTDNN), in which the habituated values, $W_k(t+1)$, are taken as inputs to a TDNN. This network combines the local temporal information of a TDNN with the long term temporal information available from an HMLP. We found that a 5 time sample HTDNN with $m = 1$ did outperform an HMLP with $m = 1$ on artificial data set 2. The classification rate was 84.1 for the HTDNN as compared to 81.2 for the HMLP. We also tried HTDNNs with 2 and 3 sample time windows, but the results were not as good as the results obtained with the HMLP.

3.8 Effect of Varying Habituation Parameters

To determine the dependence of HMLP performance on the α_k and τ_k parameters, HMLPs with $m = 1$ were trained and tested on DS2 with all α_k and τ_k parameters set to constant values A and T respectively. Figure 13 illustrates the effect of varying T with $A = 0.1$. Figure 14 demonstrates what happens when A is varied for $T = 0.3$. It is important to note that the networks illustrated in this experiment are all suboptimal. For one thing, they are all trained on the entire signals. Figure 10 illustrates that this is not the best method, but it better demonstrates the temporal effects of varying habituation parameters.

From Equation 3 we observe that T affects the rate at which habituation values change with changes in the input. For large values of T, the $W_k(t)$ values change quickly with time. The A parameter, on the other hand, controls the rate at which information is forgotten. The larger the value of A the faster the $W_k(t)$ parameters decay to a constant unity value. From Figures 13 and 14, one notices that large values of T or A result in performance peaks closer to the beginning of the signal, whereas small values of either parameter leads to peaks toward the end of the signals. For large values of A information is forgotten quickly and the latter parts of a signal are therefore classified more poorly. For small values of T, information is accumulated slowly and the earlier parts of the signal are therefore classified poorly. As both figures illustrate, intermediate values of A and T are required to optimize performance. For A too large, information about the beginning of a signal is forgotten before the end of the signal; while for A too small, unnecessary information prior to the current signal is maintained at the expense of more useful recent information. A similar tradeoff exists for values of T. Large T values lead to information which is too localized. Small values of T lead to information which is too long term.

4 CONCLUSIONS

From the proofs in section 2.2, it is apparent that a multiply habituated MLP (MHMLP) can realize any function which a TDNN with an arbitrarily large but finite input window can. The relevant issues for comparing the two thus are which generates a simpler structure for a given accuracy level, and which network is easier and quicker to train. From the empirical results determined so far, MHMLPs are consistently more efficient than TDNNs. In fact for DS1 and DS2, a single habituation unit per input HMLP outperforms a TDNN with 5 sample time windows. Even on DS3, an MHMLP with m=2 was able to perform as well as the more complex 5 time sample TDNN. Note that because of the difficult nature of the data sets, the classification rates are low for both networks. Of course they perform much better than static classifiers (MLP, RBF) which fail because of substantial overlap among different classes at any time instant.

Simple MHMLPs seem to do particularly well on data sets which require long term temporal information for classification. In such cases, TDNNs need long time windows in order to perform well. Such TDNNs tend to be overly complex, leading to slow training and poor generalization.

The performance of MHMLPs are quite robust with respect to parameter values. The networks do well even when the habituation parameters are randomly assigned and never optimized. The next step in this research is to develop means of optimizing the habituation parameters, α and τ. Another topic for further study is to investigate other input transformation mechanisms, since a result similar to Theorem 1 can be proven for any set B which is a complete memory.

References

[1] Bailey, C. and Kandel, E. "Molecular Approaches to the Study of Short-Term and Long-Term Memory," *Functions of the Brain.* pp 98-129, Clarendon Press, Oxford, 1985.

[2] Banzhaf, W., and Kyuma, K. "The Time-into-Intensity-Mapping- Network" *Biological Cybernetics,* Vol. 66, pp 115-121, 1991.

[3] Bell, T. Sequential processing using attractor transitions. In *Proceedings of the 1988 Connectionist Models Summer School*, pages 93–102, 1988.

[4] Byrne, J. H. and Gingrich, K. J. "Mathematical Model of Cellular and Molecular Processes Contributing to Associative and Nonassociative Learning in *Aplysia*," *Neural Models of Plasticity*, pp 58-70, 1989.

[5] Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems,* Vol. 2, pp 303-314, 1989.

[6] Dayhoff, J. Regularity properties in pulse transmission networks. In *Proceedings of the Third International Joint Conference on Neural Networks*, pages III–621:626, 1990

[7] S. Dehaene, J.-P. C. and Nadal, J.-P. Neural networks that learn temporal sequences by selection. *Proc. National Academy of Sciences, USA*, 84:2727–2731, 1987

[8] Djuric, P. M., Kay, S. M., and Boudreaux-Bartels, G. F. Segmentation of nonstationary signals. In *Proc. ICASSP*, pages V:161–164, 1992.

[9] de Vries, B. and Principe, J. C. "The Gamma Model - A New Neural Model for Temporal Processing," *Neural Networks,* Vol. 5, pp565-576, 1992.

[10] Ghosh, J., Deuser, L., and Beck, S. Impact of feature vector selection on static classification of acoustic transient signals. In *Government Neural Network Applications Workshop*, 1990

[11] Ghosh, J., Deuser, L., and Beck, S. A neural network based hybrid system for detection, characterization and classification of short-duration oceanic signals. *IEEE Jl. of Ocean Engineering*, 17(4):351–363, 1992.

[12] Ghosh, J. and Wang, S., "A Temporal Memory Network with State-dependent Thresholds,", *Proc. Intl. Conf. on Neural Networks*, pp. 359-364,San Francisco, March 1993.

[13] Ghosh, J. and Deuser, L., "Classification of Spatio-temporal Patterns with Applications to Recognition of Sonar Sequences", in *Neural Representation of Temporal Patterns,* E. Covey, H. Hawkins, T. McMullen and R. Port (Eds.), 1995

[14] Granger, R., Ambros-Ingerson, J., and Lynch, G. Derivation of encoding characteristics of layer II cerebral cortex. *Jl. of Cognitive Neuroscience*, pages 61–78, 1991.

[15] Hermand, J.-P. and Nicolas, P. Adaptive classification of underwater transients. In *Proc. ICASSP*, pages 2712–2715, 1989.

[16] Hertz, J., Krogh, A., and Palmer, R. G. *Introduction to the Theory of Neural Computation.* Addison-Wesley, 1991.

[17] Kurogi, S. A model of neural network for spatiotemporal pattern recognition. *Biological Cybernetics*, 57:103–114, 1987.

[18] Kung, S. *Digital Neural Networks.* Prentice Hall, Englewood Cliffs, NJ, 1993.

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

[20] Maren, A. Neural networks for spatio-temporal recognition. In Maren, A., Harston, C., and Pap, R., editors, *Handbook of Neural Computing Applications*, pages 295–308. Academic Press, 1990.

[21] Mozer, M. C. Neural network architectures for temporal sequence processing. In Weigend, A. S. and Gershenfeld, N., editors, *Time Series Prediction*, pages 243–264. Addison Wesley, 1993.

[22] Park, J., and Sandberg, I. W. "Universal Approximation Using Radial-Basis-Function Networks," *Neural Computation,* Vol. 3, pp 246-257, 1991.

[23] Robin, D., Abbas, P., and Hug, L. "Neural Responses to Auditory Patterns," *J. Acoust. Soc. Am.* Vol. 87 part 4, pp 1673-1682, 1990.

[24] Sandberg, I. Structure theorems for nonlinear systems. *Multidimensional Systems and Signal Processing*, 2:267–286, 1991.

[25] Stiles, B. W., "Dynamic Neural Networks for the Classification of Oceanographic Data," Master's thesis, Department of Elec. Eng., Univ. of Texas at Austin, 1994.

[26] Stiles, B. W., and Ghosh, J. "Habituation Based Neural Classifiers for Spatio-temporal Signals," *Proceedings of ICASSP-95 Conference,* 1995.

[27] Wang, D. and Arbib, M. "Modeling the Dishabituation Hierarchy: the Role of the Primordial Hippocampus," *Biol. Cybern.* Vol. 67, pp 535-544, 1992.

5 APPENDIX

5.1 Proof of Theorem 1

In order to prove the theorem we first prove the following Lemma.

Lemma 1 *Let t_0 be an arbitrarily large natural number, then there exists natural numbers, p and m, real numbers, a_j and c_{jk}, and elements of B, b_{jk}, such that the following inequality holds.*

$$\left|(fx)(t_0) - \sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk}(b_{jk}x)(t_0)\right)\right| < \epsilon \tag{11}$$

Proof of Lemma

Let g be a mapping from $[0,1]^t$ to $\mathcal{R}$ defined as follows.

$$g(x) = fI_x(t_0) \tag{12}$$

Here I_x is an element of X such that $I_x(i) = x_i$, the ith component of x for all $i \in \{1, 2, ..t_0\}$. Observe that g is a continuous function on a compact metric space. It is continuous because f is continuous, and it is a function because f is causal and $I_x(t)$ is zero for all $t \leq 0$. Similarly we define S a set of all functions from $[0,1]^{t_0}$ to $\mathcal{R}$, of the form $b^*(x) = bI_x(t_0)$ for each $b \in B$. Now we define K to be the set of all functions from $[0,1]^t$ to $\mathcal{R}$ of the following form.

$$s(x) = \sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk} b^*_{jk}(x)\right) \tag{13}$$

Since g is a continuous real-valued function on a compact metric space, by the Stone-Weierstrass Theorem we can state the following. If K is an algebra, separates the points of $[0,1]^{t_0}$, and does not vanish on $[0,1]^{t_0}$, then there exists an $s \in K$ such that $|g(x) - s(x)| < \epsilon$ for all $x \in [0,1]^{t_0}$. We now show that K has the three required properties. First, clearly K does not vanish because $exp(y)$ is nonzero for any real value y. Secondly, it can be readily shown that if a and b are elements of K then the pointwise product $ab \in K$, $(a+b) \in K$, and $\gamma a \in K$ for any $\gamma \in \mathcal{R}$. Therefore K is an algebra. All that remains to complete the requirements of the Stone-Weierstrass Theorem is that K separates the points of $[0,1]^{t_0}$. Let x and y be elements of $[0,1]^{t_0}$ such that $x \neq y$. If for any such x and y, there exists some $s \in K$ such that $s(x) \neq s(y)$ then K separates the points of $[0,1]^{t_0}$. The fact that x is not equal to y implies that there exists some integer k such that $1 \leq k \leq t_0$ and $I_x(k) \neq I_y(k)$. Therefore by the second property of B given in Theorem 1, there exists some $b \in B$ such that $bI_x(t_0) \neq bI_y(t_0)$. By the definition of S, therefore, there exists a $b^* \in S$ such that $b^*(x) \neq b^*(y)$. Since the exponential function is strictly monotonic, $\exp(b^*(x)) \neq \exp(b^*(y))$. Since $\exp(b^*) \in K$, K separates the points of $[0,1]^{t_0}$. By the Stone-Weierstrass Theorem, there exist real numbers, a_j and c_{jk}, natural numbers, p and m, and elements of S, b^*_{jk}, such that the following inequality is true for all $x \in [0,1]^{t_0}$.

$$\left|(g(x) - \sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk} b^*_{jk}(x)\right)\right| < \epsilon \tag{14}$$

We now make a couple final observations to complete the proof. First recall that $g(x) = fI_x(t_0)$ and $b^*_{jk}(x) = b_{jk}I_x(t_0)$. Finally observe that because of the causality of f and b_{jk}, for each $I \in X$ there is an $x \in [0,1]^{t_0}$ such that $fI_x(t_0) = fI(t_0)$ and $b_{jk}I_x(t_0) = b_{jk}I(t_0)$. This completes the proof of the lemma.

From Lemma 1 and the fact that $T_\beta I \in X$ for all positive values of β, the following inequality is true for all $\beta > 0$ and for all $I \in X$.

$$\left|(fT_\beta I)(t_0) - \sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk}(b_{jk}T_\beta I)(t_0)\right)\right| < \epsilon \tag{15}$$

Due the time-invariance of f, $fT_\beta I(t_0) = fI(t_0 - \beta)$ for $\beta < t_0$. By the third property of B in Theorem 1, $b_{jk}T_\beta I(t_0) = b_{jk}I(t_0 - \beta)$ for $\beta < t_0$. From these two observations it is apparent that the following inequality is true for all β such that $0 \leq \beta < t_0$ and for all $I \in X$.

$$\left| (fI)(t_0 - \beta) - \sum_{j=1}^{p} a_j \exp\left(\sum_{k=1}^{m} c_{jk}(b_{jk}I)(t_0 - \beta) \right) \right| < \epsilon \tag{16}$$

Let $t = t_0 - \beta$ and the proof of the theorem is complete.

5.2 Proof of Theorem 2

In order to show that B, the set of all habituation functions, is a complete memory, it is necessary to show that it meets the four properties detailed in Theorem 1. First we will show the first property, that there exists real numbers a and c such that $bx(t) \in (a, c)$ for all $b \in B$, $x \in X$, and $t \in \mathcal{Z}^+$. It is sufficient to show that $bx(t) \in [0, 1]$. This can be readily proven by using mathematical induction and recalling the range of values α and τ can take. Since $b_0 = 1$, $bx(0) = 1 - \tau x(0)$. Because τ and $x(0)$ are elements of $[0, 1]$, $bx(0) \in [0, 1]$. All that remains to be shown is that $bx(k) \in [0, 1]$ implies $bx(k + 1) \in [0, 1]$. Because of the range of τ and α values, the following inequalities hold.

$$0 \leq (1 - \tau x(k + 1))bx(k) \leq bx(k) \tag{17}$$

$$0 \leq \alpha\tau(1 - bx(k)) \leq 1 - bx(k) \tag{18}$$

Since $bx(k + 1) = (1 - \tau x(k + 1))bx(k) + \alpha\tau(1 - bx(k))$, the following inequality also holds.

$$0 \leq bx(k + 1) \leq bx(k) + (1 - bx(k)) = 1 \tag{19}$$

So B satisfies the first property of a complete memory.

Next we show that B satisfies the second property, for any $t \in \mathcal{Z}^+$ and any t_0 such that $0 < t_0 \leq t$ the following is true. If x and $y \in X$ and $x(t_0) \neq y(t_0)$, then there exists $b \in B$ such that $bx(t) \neq by(t)$.

First it is necessary to show the following lemma.

Lemma 2 *If $b \in B$ is a habituation function with habituation parameters α and τ as defined in Theorem 2, an equivalent definition for b is the following.*

$$bx(t) = \alpha\tau + \alpha\tau \sum_{j=1}^{t} \prod_{h=j}^{t} (1 - \alpha\tau - \tau x(h)) + b_0 \prod_{i=0}^{t} (1 - \alpha\tau - \tau x(i)) \tag{20}$$

The lemma is readily proven using mathematical induction.

Let x and y be elements of X such that $x(t_0) \neq y(t_0)$ for some $t_0 \leq t$. This implies that there exists a natural number i with the following properties, $0 \leq i \leq t$, there exists a $\delta > 0$ such that $|x(i) - y(i)| > \delta$, and $i < j \leq t$ implies x(j)=y(j). The number i represents the latest time prior to t at which x and y differ. Now we use i to define a value β. If $i < t$ β is defined as follows.

$$\beta = \alpha\tau \prod_{h=i+1}^{t} (1 - \alpha\tau - \tau x(h)) = \alpha\tau \prod_{h=i+1}^{t} (1 - \alpha\tau - \tau y(h)) \tag{21}$$

If $i = t$, then $\beta = \alpha\tau$. Observe $\beta > 0$ because $\alpha\tau + \tau < 1$. Using Lemma 2 and some algebraic manipulation we can derive the following equation.

$$\begin{aligned} b(x) - b(y) &= \beta \sum_{j=0}^{i} \left[\prod_{h=j}^{i} (1 - \alpha\tau - \tau x(h)) - \prod_{h=j}^{i} (1 - \alpha\tau - \tau y(h)) \right] \\ &+ \beta \left(\frac{1-\alpha\tau}{\alpha\tau} \right) \left[\prod_{j=0}^{i} (1 - \alpha\tau - \tau x(j)) - \prod_{j=0}^{i} (1 - \alpha\tau - \tau y(j)) \right] \end{aligned} \tag{22}$$

Since we have restricted τ and α to have positive values such that $\alpha\tau + \tau < 1$ we can make an important observation. The following inequalities hold for all $x \in X$ and $t \in \mathcal{Z}^+$.

$$0 < (1 - \alpha\tau - \tau) \leq (1 - \alpha\tau - \tau x(t)) \leq (1 - \alpha\tau) < 1 \tag{23}$$

Now we can exhibit a lower bound on $|b(x) - b(y)|$ in terms of i, δ, and β.

$$\begin{aligned} |b(x) - b(y)| &\geq \beta\tau\delta - \beta \sum_{j=2}^{i+1} (1-\alpha\tau)^j - (1 - \alpha\tau - \tau)^j \\ &- \beta \left(\frac{1-\alpha\tau}{\alpha\tau} \right) \left((1-\alpha\tau)^{i+1} - (1 - \alpha\tau - \tau)^{i+1} \right) \end{aligned} \tag{24}$$

Because $(1 - \alpha\tau) > (1 - \alpha\tau - \tau)$ and $i > 0$ the following set of inequalities hold. Let $\phi = \frac{1}{(\alpha\tau)(\alpha\tau+\tau)}$ -1. Let $\gamma = \frac{(1-\alpha\tau)^3(1+\alpha)}{(\alpha\tau)(\alpha\tau+\tau)}$.

$$\sum_{j=2}^{i+1} (1-\alpha\tau)^j - (1 - \alpha\tau - \tau)^j \leq \sum_{j=2}^{\infty} (1-\alpha\tau)^j - (1 - \alpha\tau - \tau)^j = \tau\phi \tag{25}$$

$$\left(\frac{1-\alpha\tau}{\alpha\tau} \right) \left((1-\alpha\tau)^{i+1} - (1 - \alpha\tau - \tau)^{i+1} \right) \leq \tau\gamma \tag{26}$$

$$|bx(t) - by(t)| \geq \beta\tau(\delta - \phi - \gamma) \tag{27}$$

Since β, τ, and δ are positive values, it is sufficient to show that the quantity $\phi + \gamma$ can be made arbitrarily close to zero by selecting appropriate values for α and τ. The upper bound on the range of α is given by the inequality $\alpha\tau + \tau < 1$. For any c such that $(1 - \tau) > c > 0$ the following is an acceptable value for α.

$$\alpha = \frac{1 - (\tau + c)}{\tau} \tag{28}$$

Since τ can take values arbitrarily close to zero, we can complete the proof of the second property by demonstrating that the limit as τ and c approach zero of $\phi + \gamma$ is zero.

$$\lim_{\tau \to 0} \lim_{c \to 0} \phi + \gamma = \lim_{\tau \to 0} \left(\frac{1}{1-\tau} - 1 + \frac{\tau^2}{1-\tau} \right) = 0 \tag{29}$$

Thus B satisfies the second property of a complete algebra.

Next we show the third property, if $b \in B$ then $(bT_\beta x)(t) = (bx)(t - \beta)$ for all $t \in \mathcal{Z}^+$, all $x \in X$ and any β such that $0 \leq \beta \leq t$. By using mathematical induction it can be readily shown that

$(bT_\beta x)(t) = 1$ for all $t < \beta$. Using this fact, it is then easy to show that the recursive definition of habituation given in Theorem 2 satisfies the third property. Once again we use mathematical induction. Because $b_0 = (bT_\beta x)(\beta - 1) = 1$ and $x(0) = T_\beta x(\beta)$, $(bT_\beta x)(\beta) = (bx)(0)$; and since $T_\beta x(t) = x(t-\beta)$ for all $t > \beta$, we can readily show that the assumption, $bT_\beta x(\beta + k) = bx(k)$ implies $bT_\beta x(\beta + k + 1) = bx(k+1)$ for any integer $k > 0$. Therefore B satisfies the third property of a complete memory. The fourth requirement for B to be a complete memory is that the elements of B are causal. Causality is readily apparent from the definition of B given in Theorem 2, so B is a complete memory and the proof of Theorem 2 is complete.

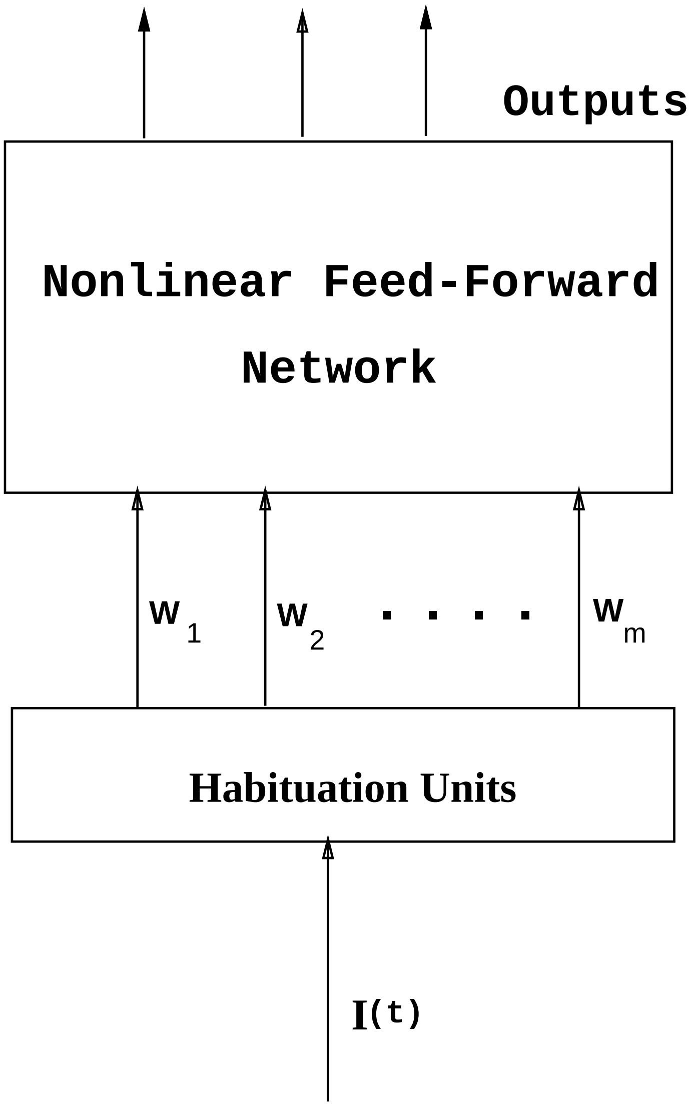

Figure 1: Structure of Habituated Neural Networks

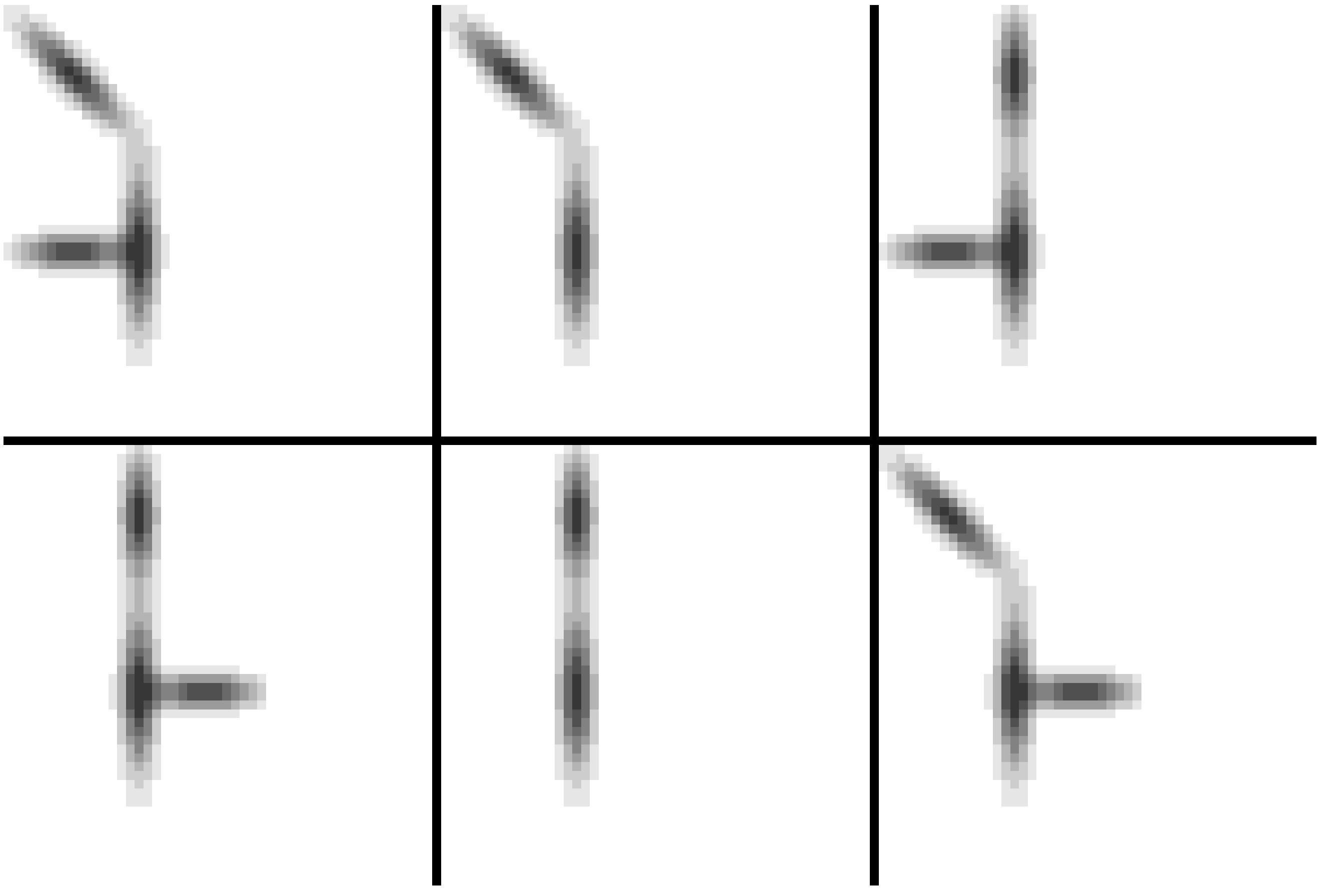

Figure 2: Banzhaf Mother Signals for classes A-F (clockwise from the top left) of Data Set 1. The vertical and horizontal axes depict time(increasing top to bottom) and frequency(increasing left to right) respectively.

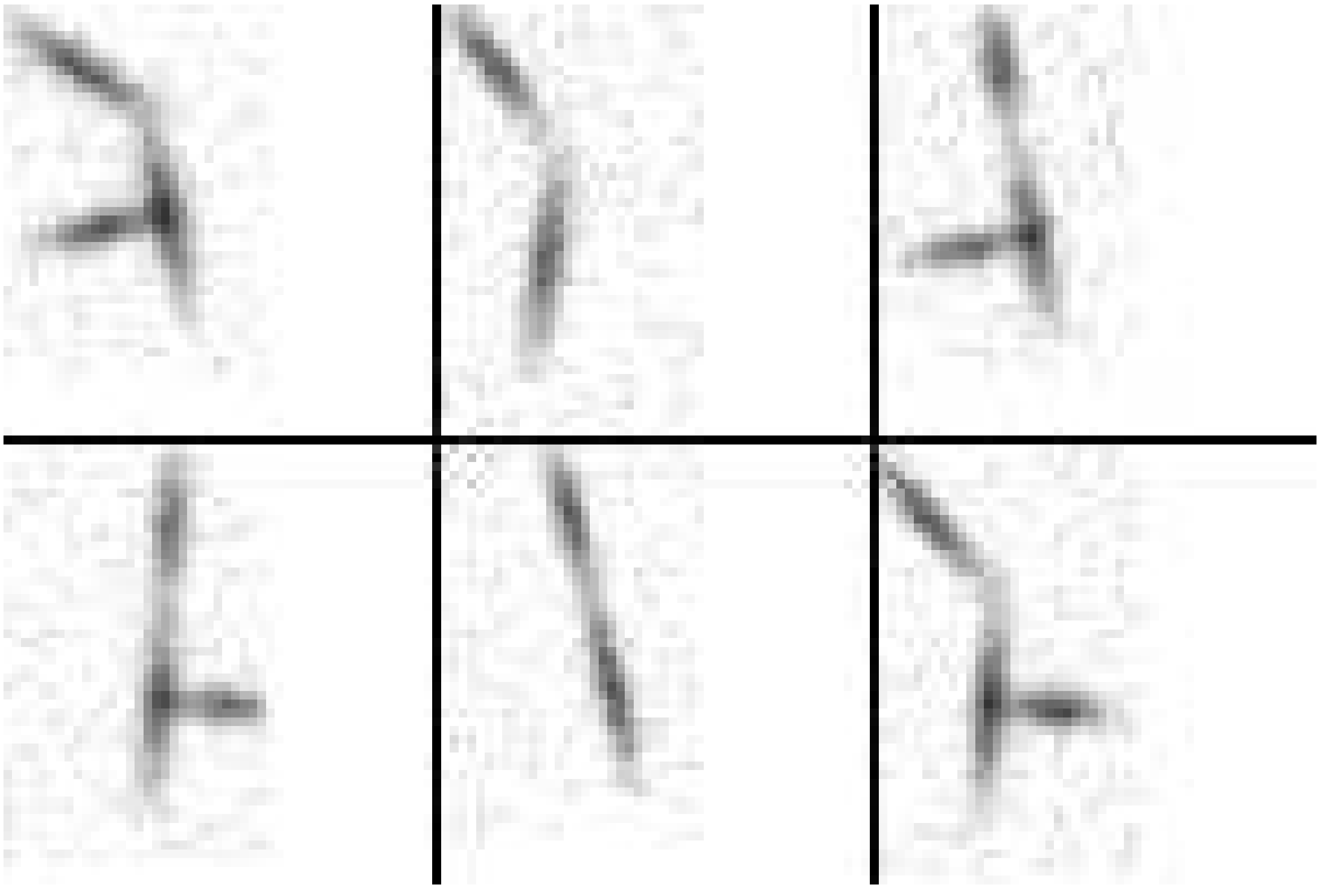

Figure 3: Sample Signals from Data Set 1.

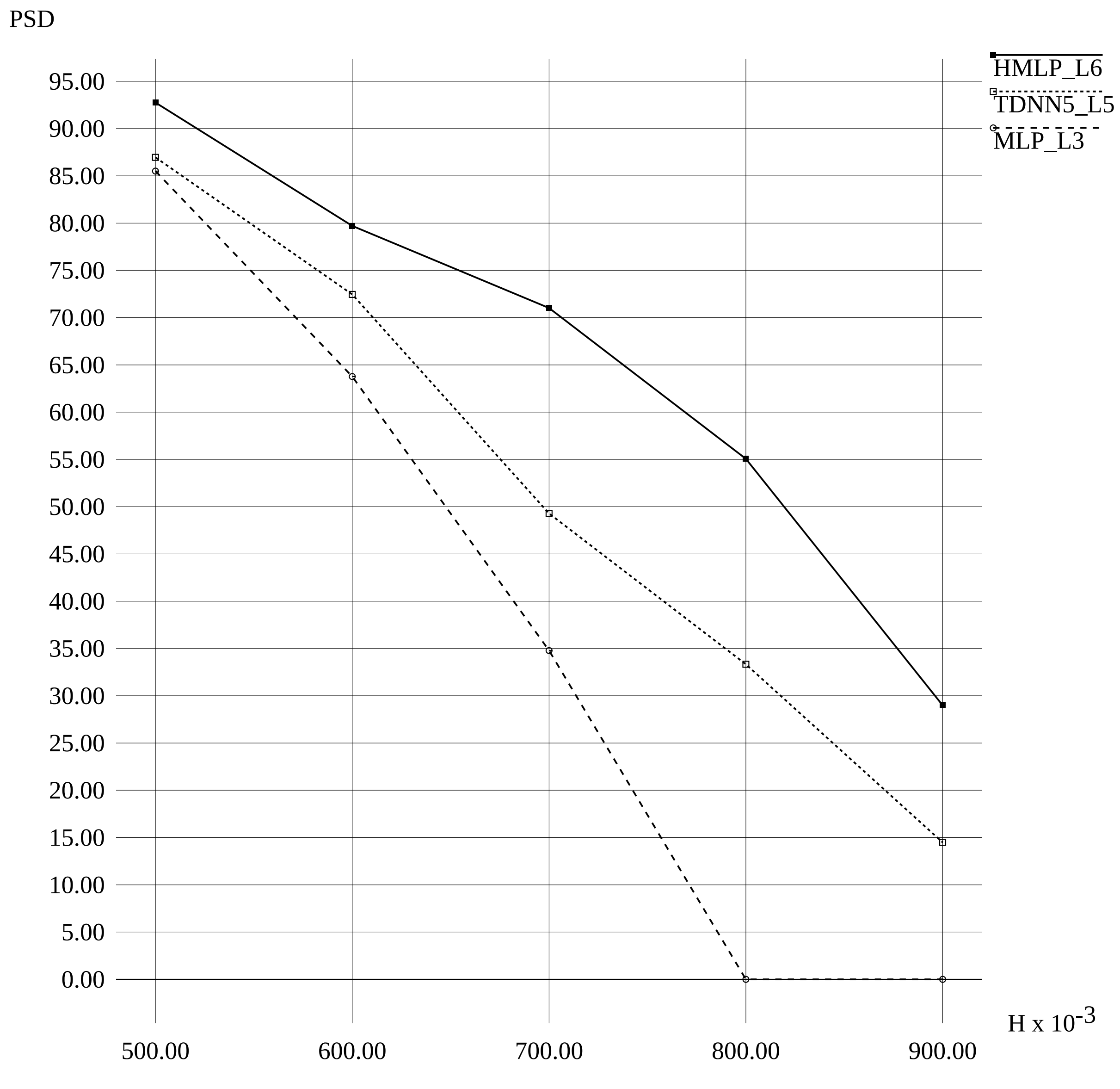

Figure 4: Percentage of Signals Detected (PSD) on Data Set 1, as a Function of Detection Threshold, H.

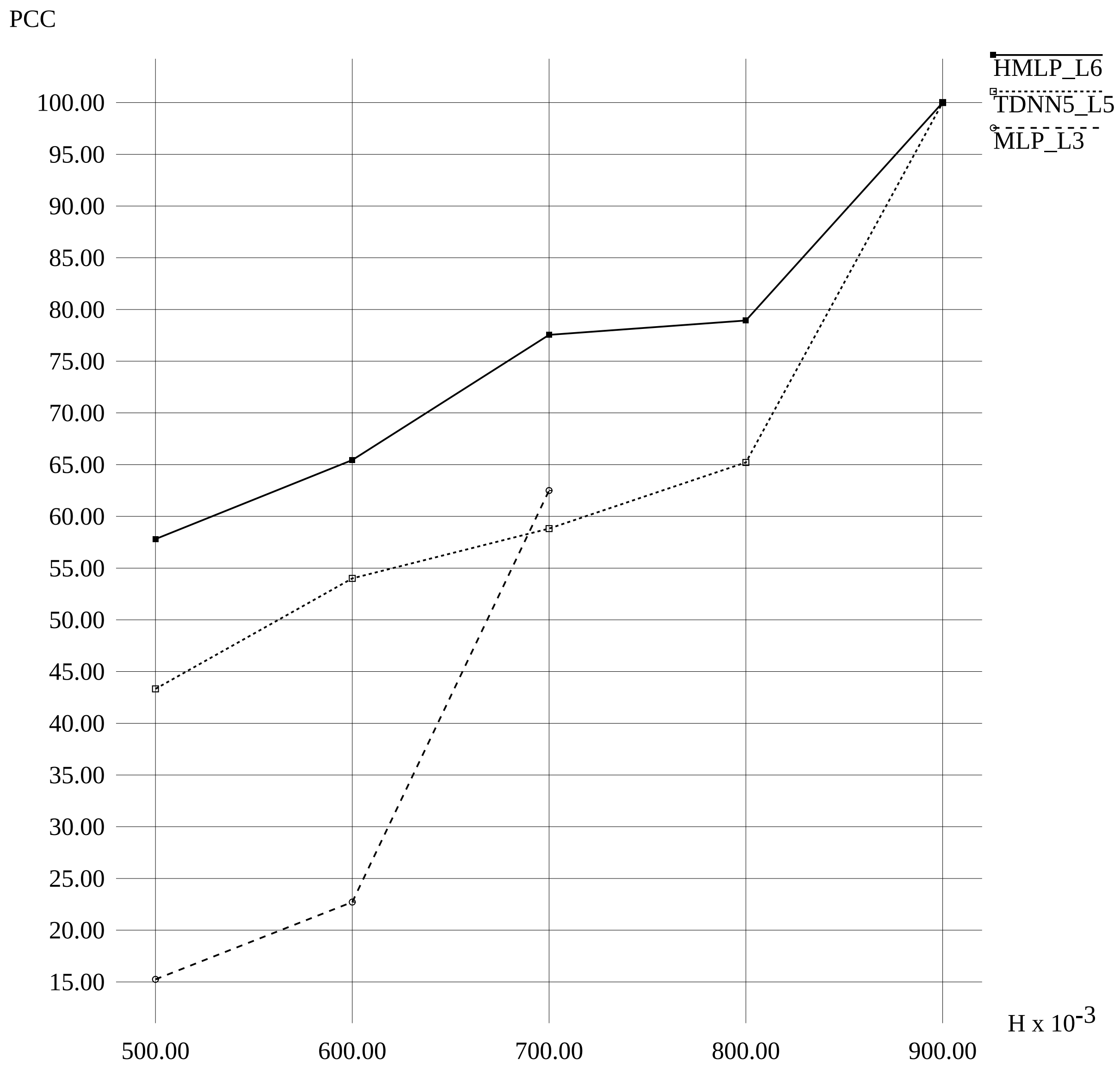

Figure 5: Percentage Correctly Classified (PCC) on Data Set 1, as a Function of Detection Threshold, H.

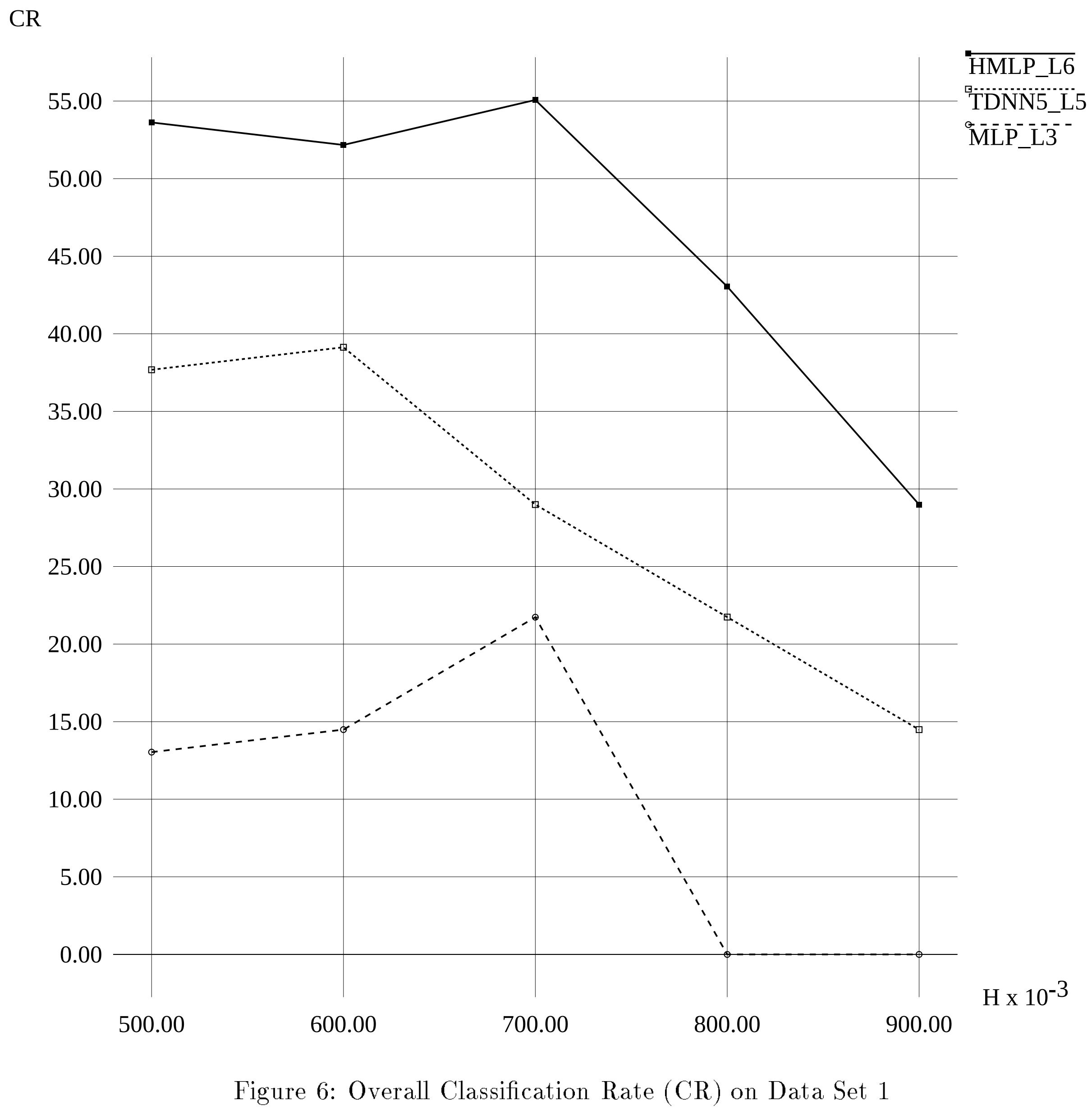

Figure 6: Overall Classification Rate (CR) on Data Set 1

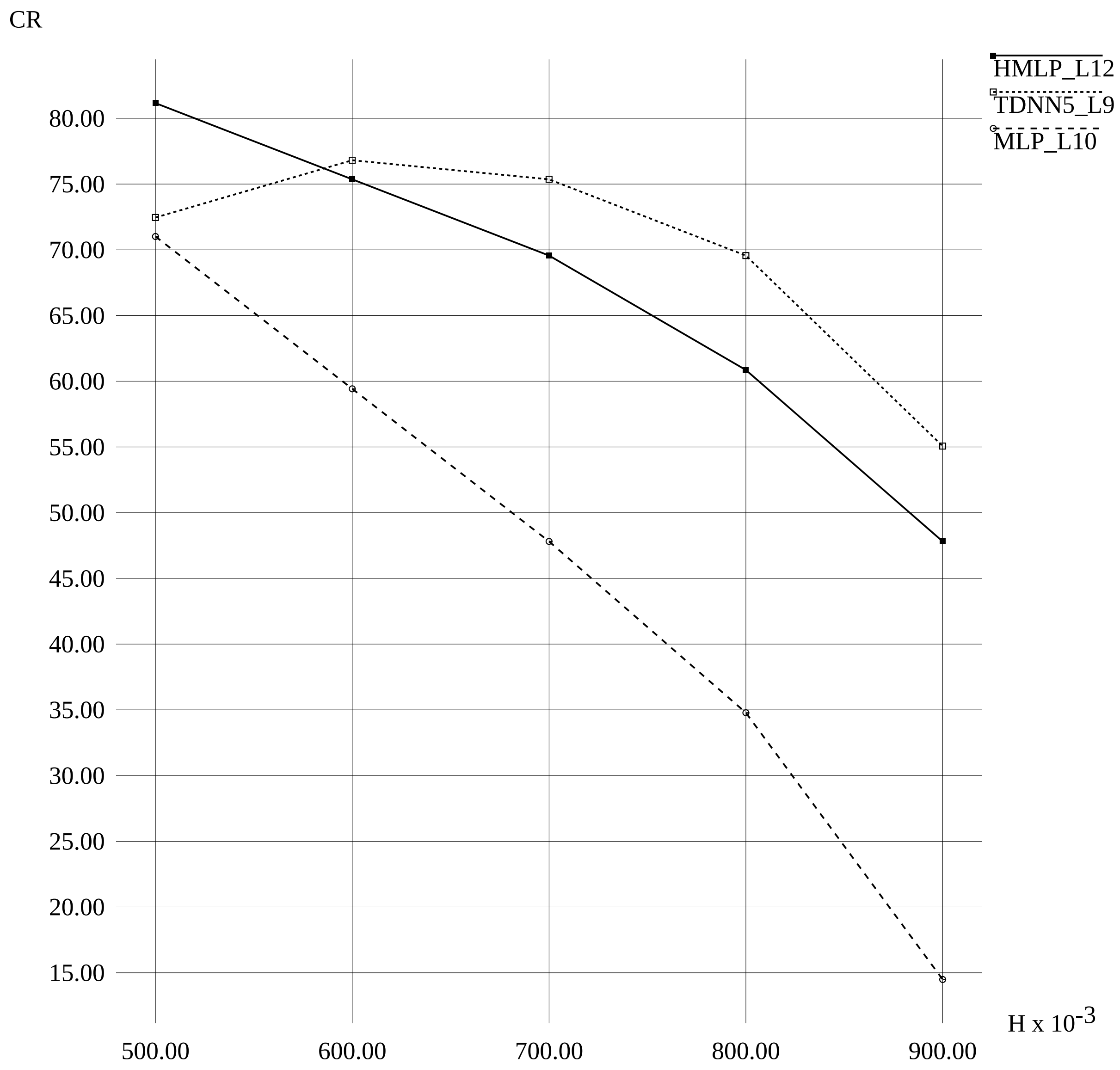

Figure 7: Overall Classification Rate on Data Set 2

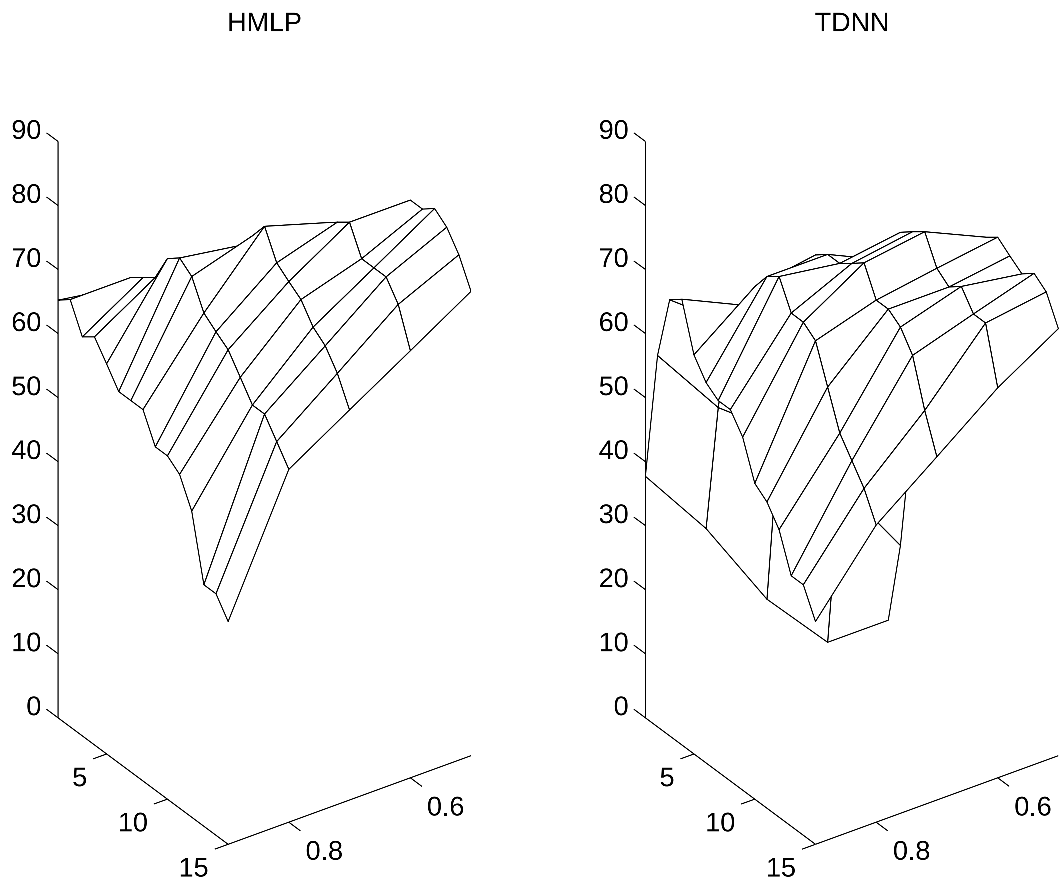

Figure 8: Effect of varying H and L on Classification Rate

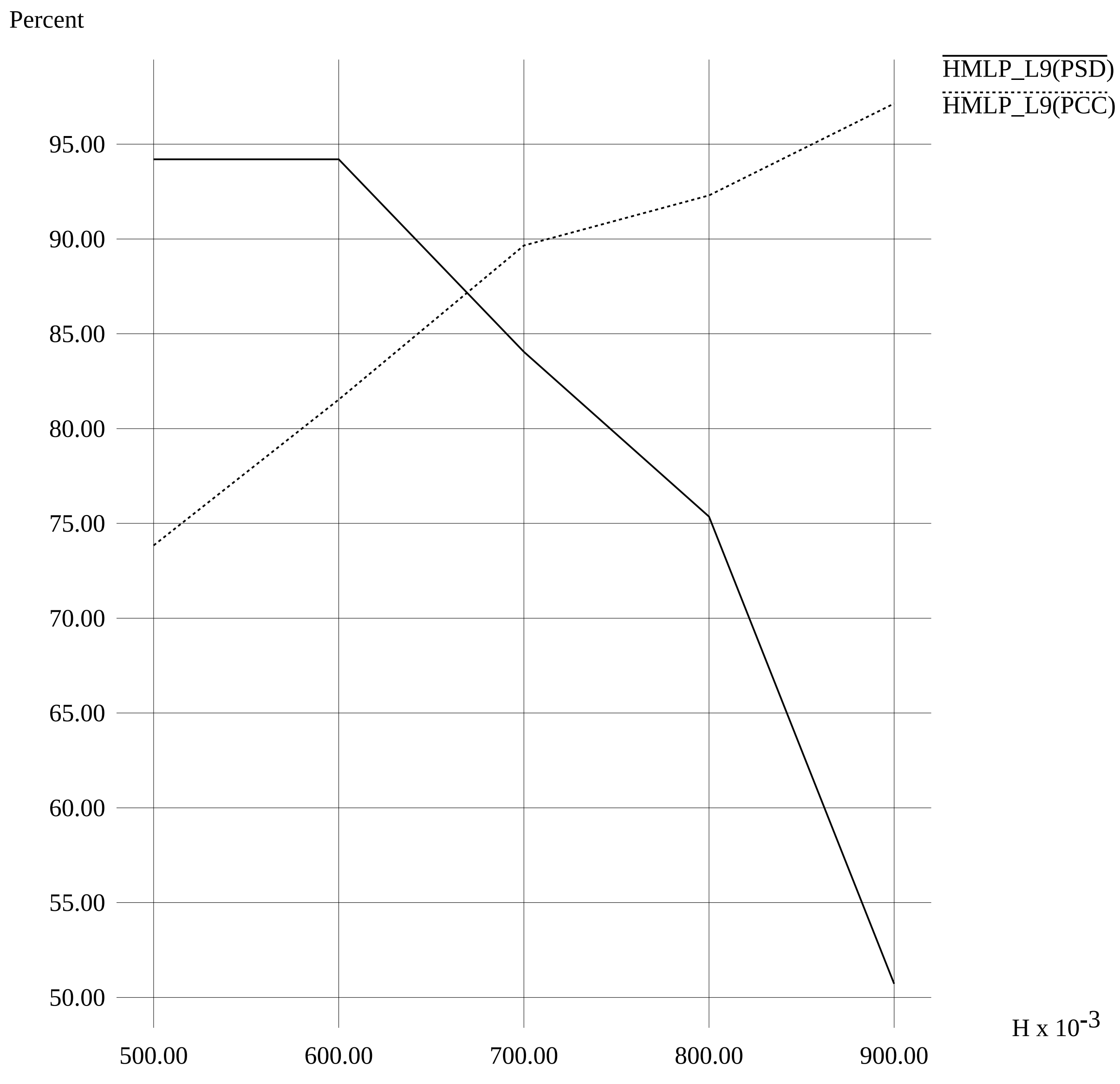

Figure 9: Performance of HMLP Using Pattern Ordering Information

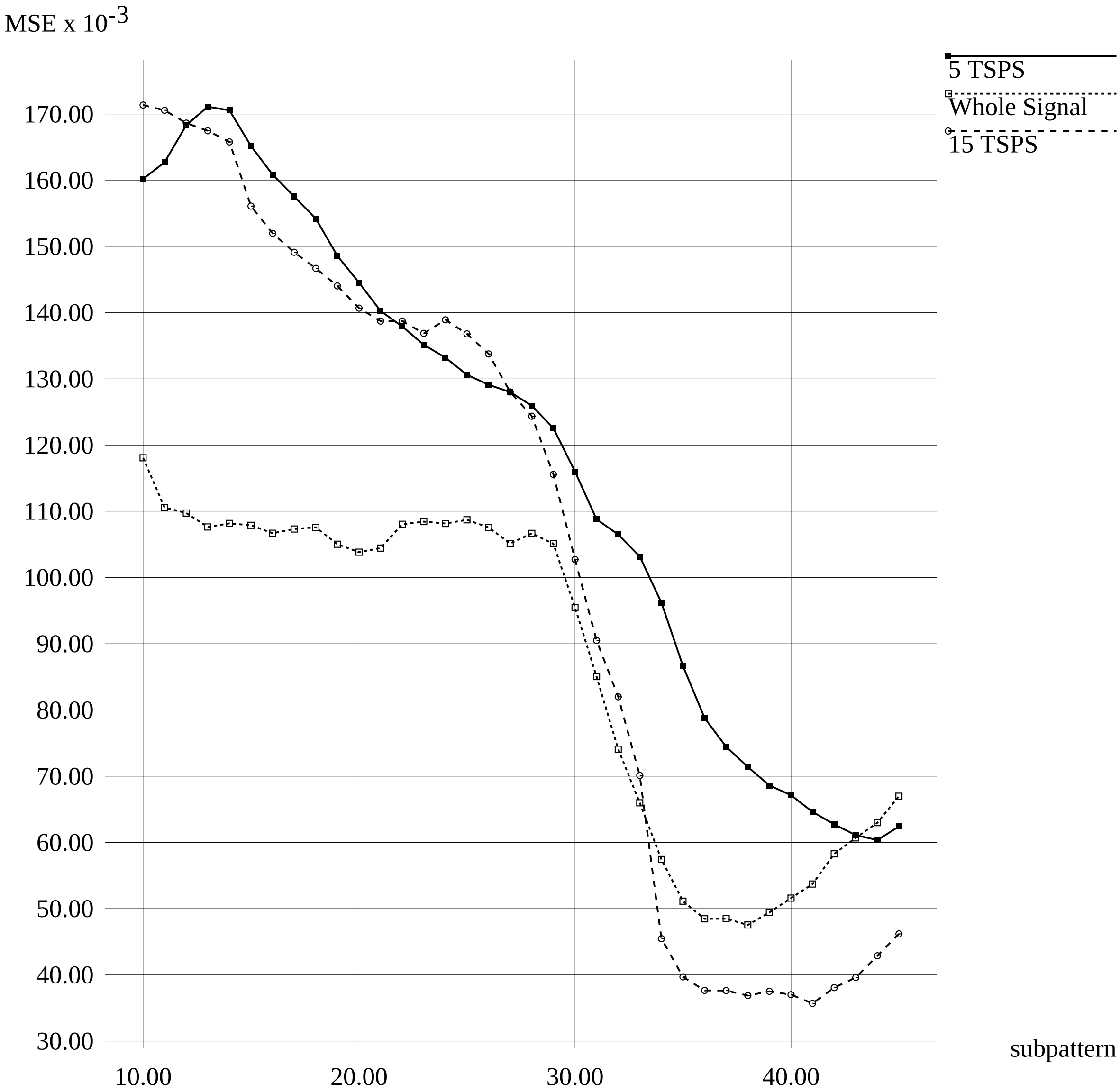

Figure 10: Effect of the Number of Training Samples per Signal (TSPS) on Performance of HMLP

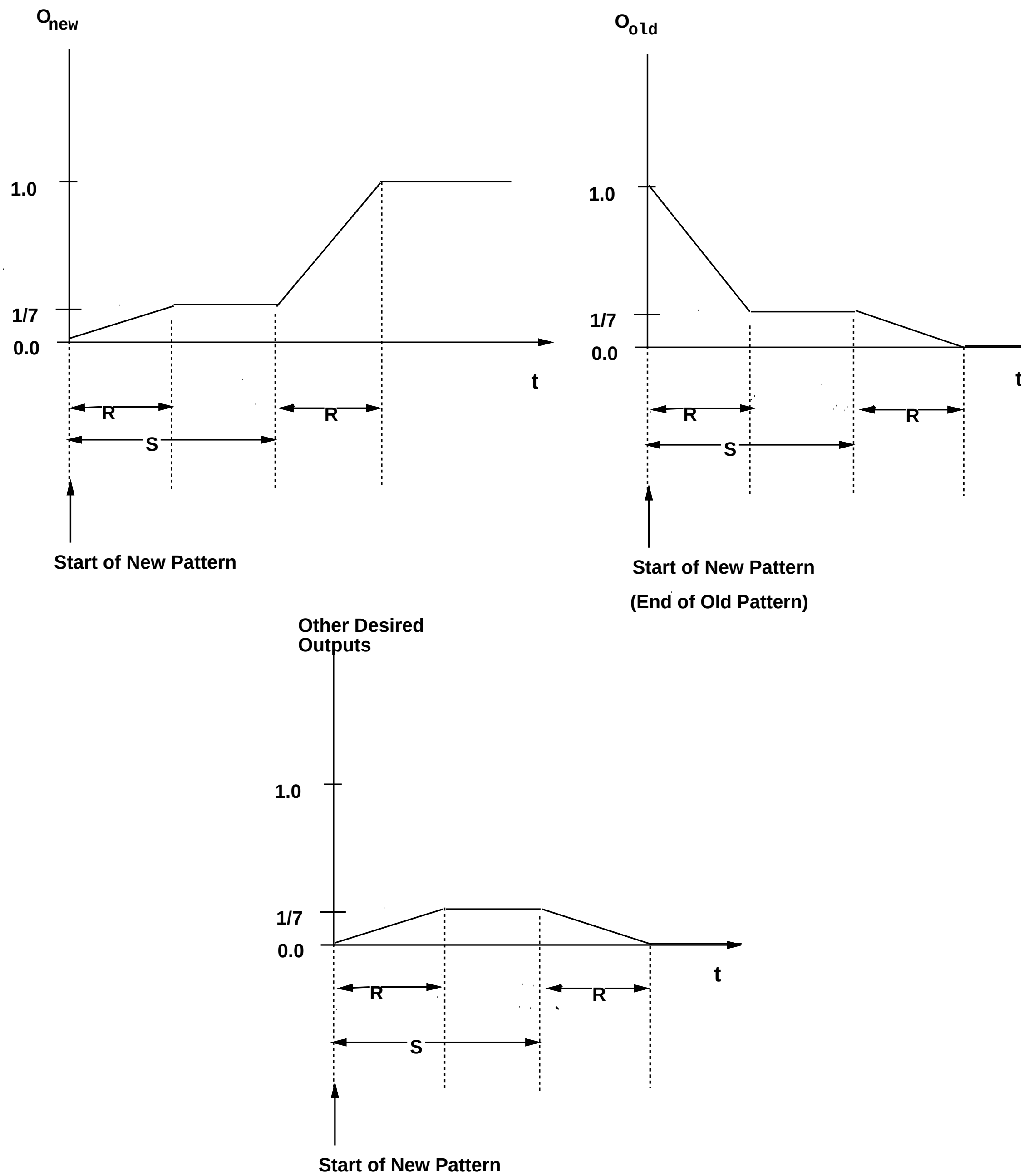

Figure 11: Illustration of Ramping Method

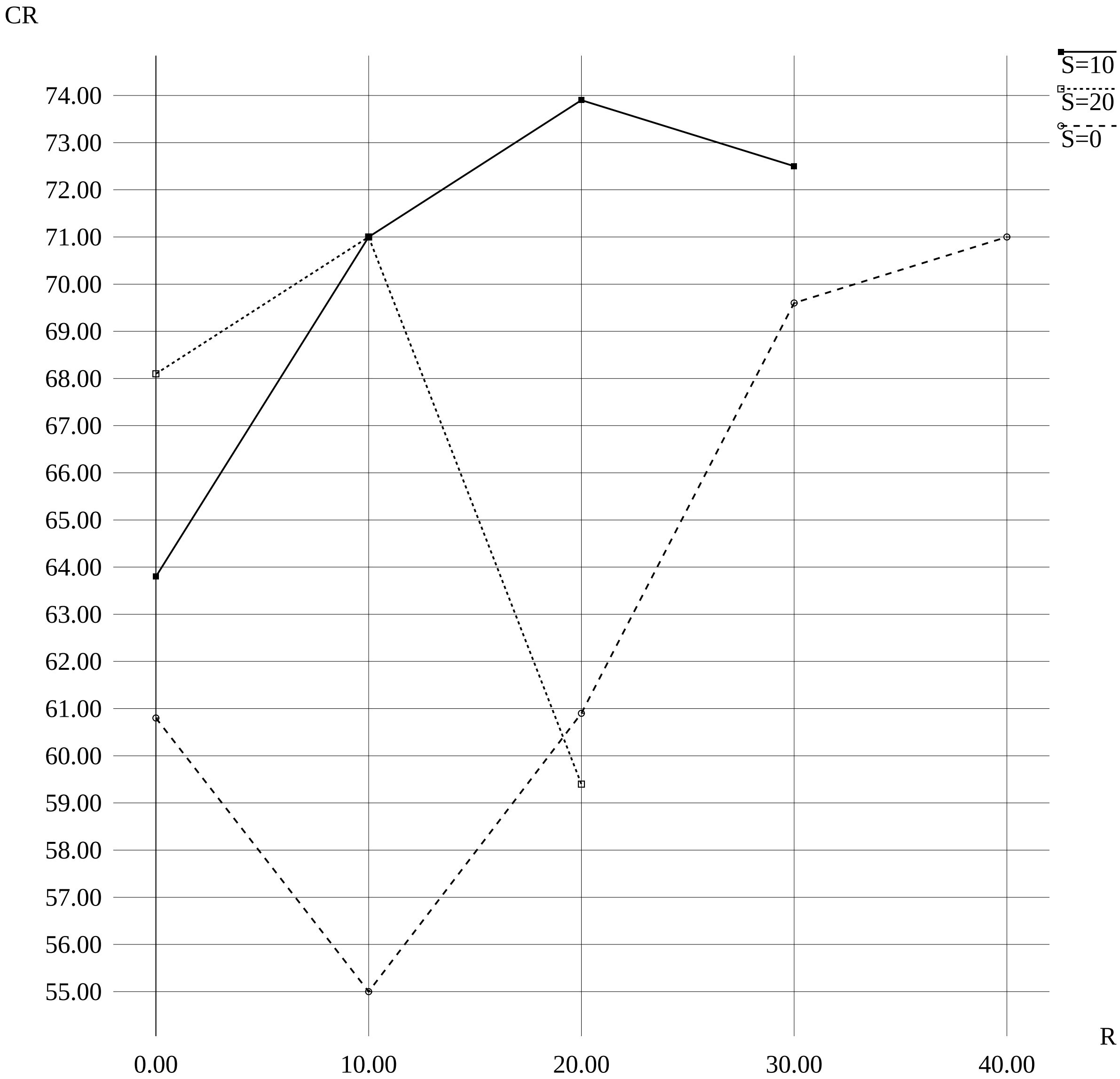

Figure 12: Effect of Ramping on Performance of HMLP on Data Set 3

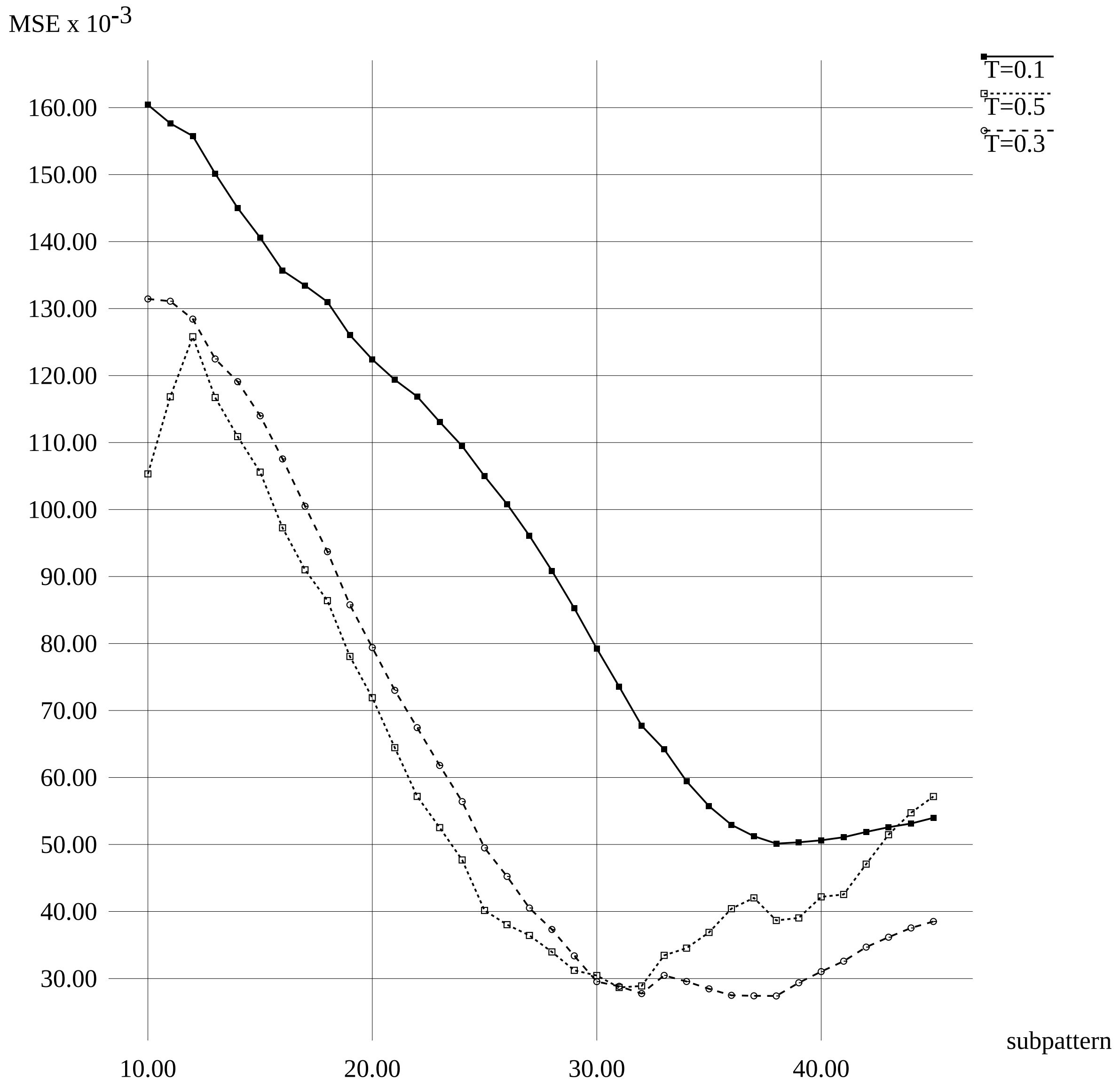

Figure 13: Effect of Varying T on HMLP Performance

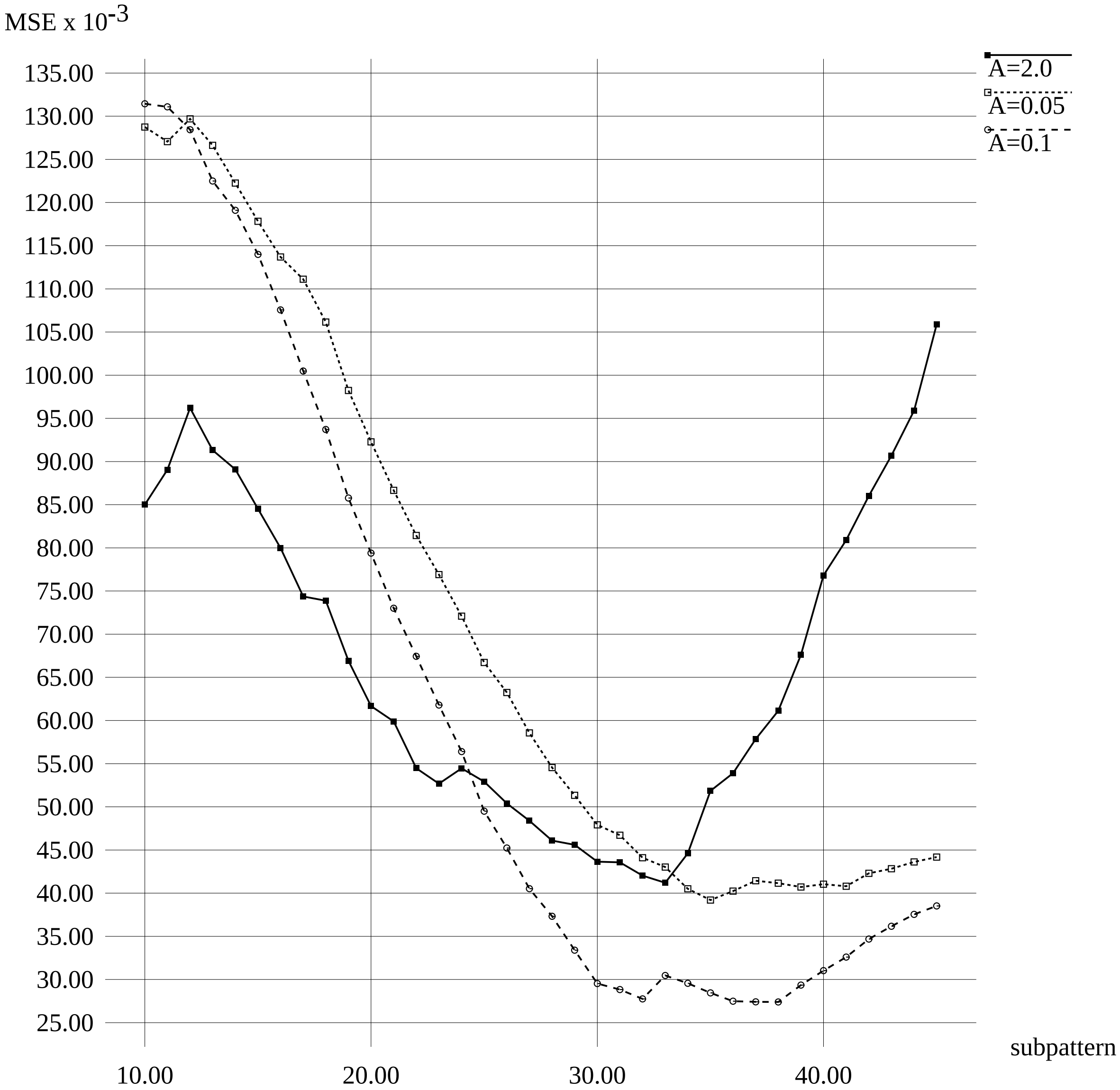

Figure 14: Effect of Varying A on HMLP Performance