Portland State University

PDXScholar

Dissertations and Theses Dissertations and Theses

1992

A gaze-addressing communication system using artificial neural networks

Gabriel Baud-Bovy
Portland State University

Recommended Citation
Baud-Bovy, Gabriel, "A gaze-addressing communication system using artificial neural networks" (1992). *Dissertations and Theses.* Paper 4258.
https://doi.org/10.15760/etd.6142

AN ABSTRACT OF THE THESIS OF Gabriel Baud-Bovy for the Master of Science in Electrical Engineering presented October 14, 1992.

Title: A Gaze-Addressing Communication System using Artificial Neural Networks

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

[redacted]

George G. Lendaris, Chair

[redacted]

Andrew M. Fraser

[redacted]

Sergio Antoy

Severe motor disabilities can render a person almost completely incapable of communication. Nevertheless, in many cases, the sensory systems are intact and the eye movements are still under good control. In these cases, one can use a device such as the Brain Response Interface (BRI) to command a remote control (e.g. room temperature, bed position), a word-processor, a speech synthesizer, and so on.

The BRI is a gaze-addressing communication system that was developed by Dr. Erik E. Sutter at the Smith-Kettlewell Eye Research Institute, San-Francisco: a menu of communication objects (e.g. letter, word, command) is displayed on a TV screen; the disabled person selects an object by gazing at it; an electrode implanted

in the visual cortex transmits the brain-wave response evoked by the visual stimulus associated with the object; a special device receives and analyzes the brain-wave signal to identify which object is being gazed at; and, finally, the action corresponding to the object is undertaken (e.g. change bed position). Currently, the brain-wave signal is analyzed by computing the correlation between the current brain-wave signal and previously recorded template signals. Although it was not part of the initial objective of this thesis, we found that normalizing the brain-wave signal before computing the correlation yields a significant improvement on the performance. This observation is of immediate utility since, to our understanding, the current implementation of the BRI does not normalize the brain-wave response.

This thesis explores using Artificial Neural Network (ANN) technology instead of the correlation technique to identify the brain-wave signals, with the important objective of shortening the response time. The method pursued was to effect correct classification on short portions of the brain-wave signals (short brain-wave signals lead to short response times).

In the present thesis, we conducted several experiments with two very different ANN architectures. One of the ANN architectures was found to consistently perform at least as well as the correlation technique. None of our experiments, however, performed significantly better than the cross-correlation technique. In the course of the experiments, we repeatedly observed two phenomena that may have had an impact on the performance. We believe that these phenomena may happen in different problem contexts. Thus, the study and a general solution of these problems could have an impact beyond this particular problem context.

The first phenomenon, here called contrast-enhancement, refers to the fact that only the small deviations from the desired output values disappear during the training process whereas the larger ones do not. Associated with this phenomenon, we observed that the average output values of the ANN continue to get closer to the desired output values long after the (correct) identification decisions reached their maximum during the training process. The second phenomenon, here called nearby misclassification, refers to the fact that a misclassification is more likely to happen between two brain-wave signals evoked by "similar" visual stimuli. In this thesis, we describe a modification of the backpropagation paradigm aimed at improving the discrimination among arbitrarily defined classes of inputs (the error computed by the criterion function is made dependent on the class of the input).

Altogether, we believe that the results reported in this thesis are promising since this research was conducted under the severe handicap of having only 99 brain-wave responses. This is insufficient because the brain-wave responses are extremely difficult to categorize (the problem context cannot be adequately represented by only 99 responses), and because it is a well known fact that the size and quality of the training set is crucial to achieving good performance with the ANN technology. In addition, in spite of this handicap, these results provided a better understanding of the problem context and yielded several interesting observations that may be helpful in a future of work.

A GAZE-ADDRESSING COMMUNICATION SYSTEM USING ARTIFICIAL NEURAL NETWORKS

by

GABRIEL BAUD-BOVY

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Portland State University
1992

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of Gabriel Baud-Bovy presented October 14, 1992.

[redacted]

George G. Lendaris, Chair

[redacted]

Andrew M. Fraser

[redacted]

Sergio Antoy

APPROVED:

[redacted]

Rolf Schaumann, Chair, Department of Electrical Engineering

[redacted]

Roy W. Koch, Vice Provost for Graduate Studies and Research

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

A motivation for this research has been to make a contribution toward the satisfactory solution of the communication problems faced by severely disabled persons. The idea is to construct a system that permits a severely disabled person to select a particular object intended for communication just by gazing at it.

> Severe motor disabilities can render a person almost completely incapable of communication (locked-in syndrome). The most frequent causes are amyotrophic lateral sclerose (ALS), severe cerebral palsy, and head trauma. Often, the person's gaze is the most efficient means to transmit information, since in most cases the sensory systems are intact and eye movements are still under good control. (Sutter, 1992, pg. 1)

Any system that uses a person's gaze for communication purposes is called a gaze-addressing communication system. A common approach involves the use of eye tracking devices that observe the eye position. Up-to-date eye tracking systems use video cameras to detect the orientation of the pupils (Eye Typer[1], Cedric[2]). These systems, however, have several drawbacks. For example, they do not tolerate head movements and do not work well with persons wearing eye glasses (Sutter & Tran, 1992a).

[1] "Eye Typer" is a trademark of Sentient System Technology, Inc.

[2] The Cedric communicator is manufactured by Santech Pty Ltd., Australia

1. THE BRAIN RESPONSE INTERFACE

The system developed by Dr. Erich E. Sutter (1992) is based on a different approach. The system, called the Brain Response Interface (BRI), does not rely on the orientation of the pupil; instead, it analyses the brain-wave signals evoked by the visual stimuli associated with the communication objects being gazed at.

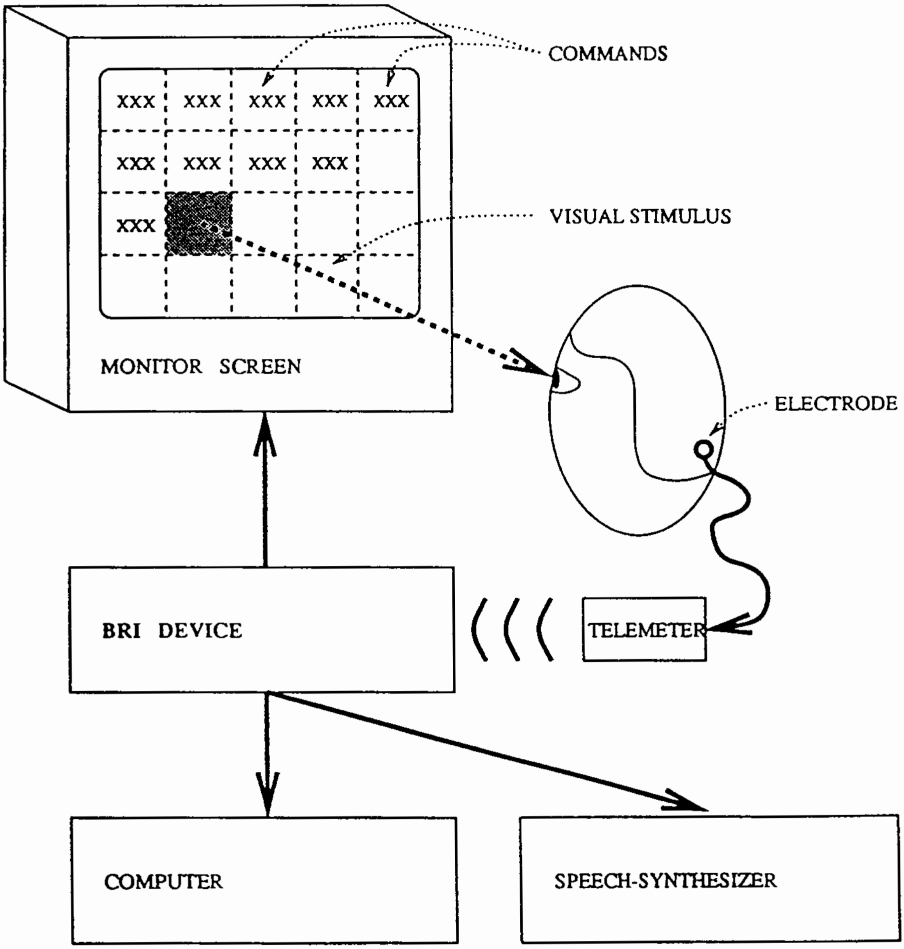

Figure 1. The Brain Response Interface.

The BRI makes use of computer technology: a menu of (64) communication objects (e.g. letter, word, command) is displayed on a monitor screen (a different visual stimulus is associated with each object); the disabled person selects an object by gazing at it; an electrode records the brain-wave signals induced by the visual stimulus; a telemeter[3] transmits the signals to the BRI device; and the BRI device analyzes the brain-wave patterns and determines which object is being gazed at. The disabled person must hold his or her gaze during the entire process (usually 1.5 second). Once the object as been identified, it is finally transmitted to one of the devices connected to the system (e.g. a word-processor, a speech-synthesizer).

In the system designed by Dr. Erich E. Sutter, the analysis of the brain-wave patterns is carried out by an algorithm that computes the correlation between the current brain-wave signal and previously recorded template signals. The present thesis explores using the artificial neural network technology to replace thecross-correlation technique.

2. A COMMUNICATION SYSTEM MODEL

The intention of a gaze-addressing system is to enable a disabled person to communicate with his or her environment. An abstract model of a communication system is illustrated in Figure 2, along with how this model is applied to the present

[3] A telemeter is used to prevent a potential risk of infection that would be present if there was a direct wire connection from the electrode to the BRI device (Sutter, Pevehouse, Barbaro, 1992).

problem context. The components of the model are described in the following paragraphs and, in more detail, in the subsequent chapters.

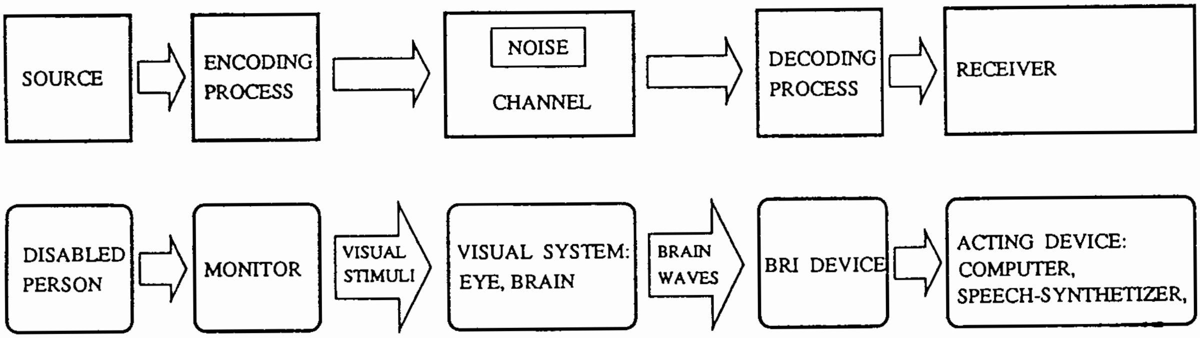

Figure 2. Application of a communication system model to the present problem context. Each part of the communication system can be mapped to a corresponding element of the present problem context.

2.1. Information Source

Information theory defines the information source to be a random generator of events. An information source is characterized by a probability distribution underlying the generation of the events. The particular sequence of events that is generated by the information source constitutes the information that has to be transmitted. In the present problem context, we define an event to be the act of gazing at one of the 64 communication objects displayed on the monitor screen. Thus, we associate the user of the BRI with the information source since it is he or she who selects the command. In the present thesis, we assume that all objects have the same likelihood of being chosen.

2.2. Encoding Process

The function of the encoding process is to associate a codeword with every event generated by the information source. The characteristics of the codewords depend on the problem context. In the present case, the codewords are specially designed visual stimuli generated at the monitor screen.

An encoding process is usually divided into two parts: source encoding and channel encoding. The role of source encoding is to increase the speed of transmission. To do so, source encoding takes advantage of any structure in the information being encoded and, usually, assigns short codewords to frequent symbols. The role of channel encoding is to maximize the likelihood that the information will be recovered after it has been transmitted by the channel. For that purpose, channel encoding takes advantage of whatever properties are known about noise contaminating the signal during the transmission. To combat the noise, channel encoding typically introduces redundancy in the codewords. Both issues are addressed in Chapter II.

2.3. Channel

The role of the channel is to transmit a signal and, thus, to carry the information from one point to another. In the BRI, the channel is the visual system of the user (i.e. the eye and the visual cortex). Note that the visual system changes the representation of the information (i.e. the codewords) as the visual stimulus becomes a brain-wave signal. In the BRI, the brain-wave signal is next detected by an electrode and transmitted to the BRI device by a telemeter. Our model does not include the electrode and the telemeter, because we assume that their effect on the transmission of the information can be neglected.

2.4. Noise

Noise corresponds to any interference happening during the transmission of the signal that causes a loss of information. The noise is often modeled as a random process. In the BRI, since the visual system is a very noisy channel, it is important to carefully design the visual stimuli to prevent significant loss of information.

2.5. Decoding Process

The function of the decoding process is to retrieve the information at the receiving end of the channel. In the system originally designed by Dr. E. Sutter (1992), such analysis is carried out by an algorithm that computes the correlation between the brain-wave signal being detected and previously recorded template signals. In the present thesis, artificial neural network (ANN) technology replaces the cross-correlation algorithm.

2.6. Receiver

The receiver represents any entity that makes use of the information. In the present problem context, the receiver is any type of machine that may be useful for communicating to, or controling the environment. For example, it can be a wordprocessor, a speech-synthesizer, or an universal remote control (stereo, television, bed position, room temperature).

3. OBJECTIVES

Identification of temporal patterns (events) in a noisy background is a classic problem of real-time signal analysis. Our main objective is to design an artificial neural network that can identify certain brain-wave signals evoked by visual stimuli. The ability of artificial neural networks to perform well on noisy data sets is important in this context because the signal-to-noise ratio of brain-wave signals is low.

> Evoked Potentials (EP) can be conceived as deterministic low amplitude signals embedded in colored noise, the electroencephalogram (EEG) background activity, which has temporal and spectral characteristics similar to the EP waveforms. This fact increases the difficulty of detecting and estimating the parameters of the EP themselves. Typical values of the signal-to-noise ratio are less than 0dB. (Uncini, 1990, pg. 145)

One particular problem has attracted our attention: in order to obtain the important objective of shortening the response time of the BRI, the artificial neural network must be able to correctly identify the visual stimuli on the basis of just a portion (e.g. the first half) of the brain-wave responses.

In the present thesis, even though the results obtained with the ANN technology are very similar to those obtained with the cross-correlation technique, we identified two problems that, if solved,could lead to a significant improvement of the classification performance. The first problem is the contrast-enhancement phenomenon that characterized the training process of most of our experiments. The second problem is the nearby misclassification distribution phenomenon that was repeatedly observed when the performance of our experiments were evaluated. Both phenomena are described in detail in Chapter V.

4. SIMILAR WORK

ANN technology has been used in many different problem contexts that are related to the work of the present thesis: detection, identification, reconstruction, or temporal correlation of individual spike trains (Jansen, 1990; Wong, Banik & Bauer, 1988; Hassoun & Spiker, 1988; Tam & Perkel, 1989), electroencephalogram (EEG) topography recognition (Hiraiwa, Shimihara & Tokunaga, 1990), electrocardiogram (ECG) analysis (Caroll & Ved, 1990; Strand & Jones, 1990), classification of myoelectric signals (Kelly, Parker & Scott, 1990), somatosensory-evoked potentials (Holdaway, White & Marmarou, 1990), dimensionality of EEG signals (Lo & Principe, 1989).

Appendix E contains a long list of (additional) references.

CHAPTER II

THE ENCODING PROCESS

The function of the encoding process is to associate a codeword with every event generated by the information source.

1. THE MONITOR SCREEN

The monitor screen in the BRI is divided into 64 command fields arranged in an 8x8 array (see Figure 3). Each command field contains the name of a command[4] displayed in small characters. The name of the command, however, is not taken into account in the definition of the visual stimuli because the characters specifying it cover only a small area of each field: the definition of the visual stimuli is only based on the background state of the field. The names of a command can therefore be changed without having to redefine the visual stimulus associated with the corresponding field. This opens the way to the realization of a system of menus of commands which may potentially offer an unlimited number of commands (the actual implementation of the BRI has 8 levels of menus). The division of the screen into 64 fields limits the size of the menu (i.e. the number of commands that can be simultaneously accessed).

[4] In this chapter as well as in the rest of this thesis, the term "command" is favored instead of the term "communication object" used previously.

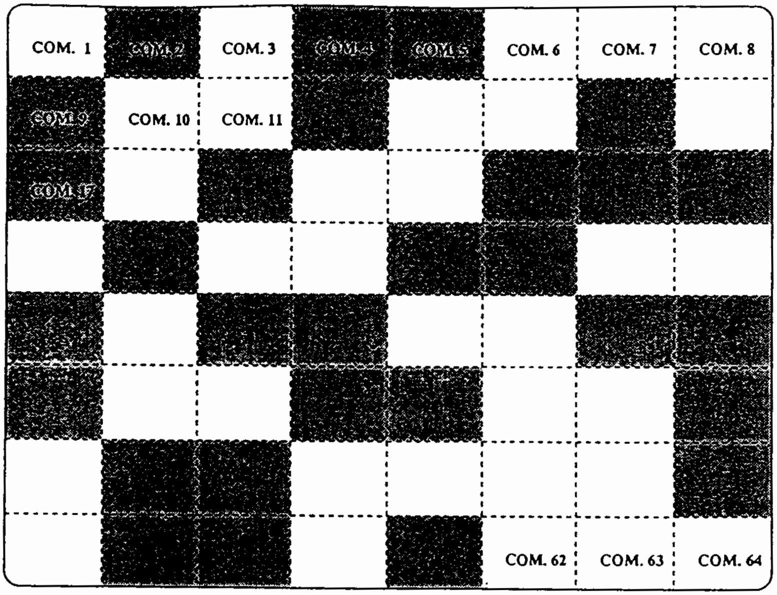

Figure 3. The monitor screen. The names of the commands (communication objects) are displayed in the command fields.

2. THE VISUAL SYSTEM

2.1. Cortical Magnification

In order to give a precise form to the concept of visual stimulus, we make the distinction in this chapter between the 64 visual stimulus elements that correspond to the command fields, and the visual stimulus that corresponds to the entire monitor screen. We need this distinction to explain how the BRI is able to identify which command is being gazed at, despite the fact that the surrounding commands are still included in the visual field.

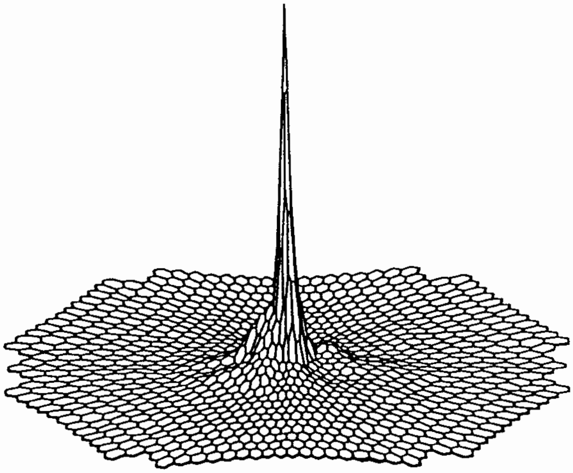

Figure 4. Amplitude of the brain-wave response to stimuli in different locations of the visual field. The figure is copied from Sutter & Tran (1992a, pg. 4).

Figure 4 shows the amplitude of a set of brain-wave responses induced by a set of identical visual stimuli in different locations in the visual field. Figure 4 reveals the increased sensitivity of the visual system at the center of the visual field: the visual stimuli at the center of the visual field generate by far the largest brain-wave responses. This is justified by the disproportionately large amount of the visual system allocated to the processing of the center of our vision, where the acuity is the greatest (Sutter & Tran, 1992b). This property of the visual system is called cortical magnification, and can sometimes reach an accuracy of 1 degree (Sutter & Tran, 1992b, pg. 20). Without this property, it would be very difficult to identify which command is being gazed at, because the visual stimulus at each of the command fields would evoke equal brain-wave activity.

2.2. Constraints On The Visual Stimuli

The visual stimuli must always be modulated-in-time, since significant alterations of brain-wave patterns are induced only by stimuli that change rapidly (e.g. a blinking light). The form of the brain-wave responses depends on the modulation patterns of the visual stimuli, not on the state of the command fields at specific times. It has been determined empirically that the modulation rate of the visual stimuli is limited; visual stimuli with excessively high modulation rates do not appear to effect changes in the brain-wave responses.

Two issues must be considered in order to choose the appropriate modulation patterns for the visual stimuli: noise contamination and response time of the BRI. It is interesting to notice that these issues correspond to the parameters of a fundamental characteristic of a communication system called channel capacity.

Channel capacity defines what is the maximum amount of information that a channel can transmit. This limit stands independently of the encoding and decoding processes. The definition of channel capacity is

$$C = 2WT \log (1 + \frac{S}{N}) \tag{1}$$

where C is the channel capacity, W is the bandwidth of the system, S/N is the signal-to-noise ratio, and T is the signal duration. In the present problem context, the bandwidth (modulation rate) of the visual system is limited. Thus, the only way to increase the channel capacity (i.e. increase the quantity of information that may be transmitted) is to modify either the signal-to-noise ratio of the signal (noise contamination issue) or the signal duration (response time issue). We do not attempt to compute the channel capacity in the present thesis because we do not have reliable means to estimate the signal-to-noise ratio of the brain-wave responses.

2.3. Noise Contamination Issue

Although the brain wave responses are mostly regulated by the visual stimulus elements at the center of the command field (cortical magnification), the surrounding visual stimuli do contaminate the main signal. The surrounding visual stimuli can be viewed as noise sources. Three approaches are possible to combat noise contamination. The first approach takes advantage of the fact that the visual system is overall additive (Sutter, 1992a). In practical terms, this means that the signal-to-noise ratio does not decrease if all brain-waves are uncorrelated. Hence, to insure a good signal-to-noise ratio of the main brain-wave response, the surrounding visual stimuli must generate a set of orthogonal brain-waves responses. The second approach is to repeat the signal until it has been identified. It assumes that the visual system is stationary and relies, for example, on averaging methods to increase the signal-to-noise ratio. The last approach is to increase the length (redundancy) of the signals. Various combinations are possible. For example, if the decoding process has not identified the visual stimulus at the end of the stimulation cycle (i.e. after the visual stimulus has been entirely transmitted), then the visual stimulus could be repeated and the brain-wave responses for both stimulation cycles averaged.

2.4. Response Time Issue

The response time is the time necessary for the decoding process to identify from the brain-wave response which visual stimulus is present. Ideally, the decoding process of the BRI would be able to identify which command is being looked at on the basis of the brain-wave response to either short visual stimuli, or initial segments of visual stimuli. Once a visual stimulus has been identified, the BRI waits for a short interval of time (300ms) to allow the user to select a new command before restarting the process.

The response time issue is closely related to the noise contamination issue; for example, noise-proof signals could help to obtain short response times. However, each issue conveys its own twist and must be considered independently.

2.5. Example

These issues are best described with an example. It is easy to construct a set of orthogonal visual stimuli

$x_1(t)$: *1000 0000 0000 000*
$x_2(t)$: *0000 1000 0000 000*
$x_3(t)$: *0000 0000 1000 000*
$x_4(t)$: *0000 0000 0000 100*

Obviously, the brain-wave responses induced by these visual stimuli are uncorrelated if the lag between the impulses belonging to different visual stimuli is greater than the duration of the brain-wave response to an impulse (80-100ms). This set of visual stimuli, however, is not satisfactory because it is impossible to distinguish between the third and fourth visual stimulus before almost the end of the stimulation cycle. Increasing the number of visual stimuli aggravates the problem; if similar visual stimuli were to be used in the BRI to encode the 64 commands, the response time could exceed 6.4 seconds. In contrast, consider the following set of visual stimuli

$x_1(t)$: *0001 0011 0101 111*
$x_2(t)$: *0011 0101 1110 001*
$x_3(t)$: *0101 1110 0010 011*
$x_4(t)$: *1110 0010 0110 101*

If the visual system were linear, it could be shown that the brain-wave responses induced by the initial segments of the visual stimuli would be nearly uncorrelated (and, a fortiori, different). Thus, the decoding process could determine almost immediately which command is being gazed at. Since however the visual system is highly nonlinear (and very noisy), we cannot demonstrate directly that the brain-wave responses induced by these signals are uncorrelated.

3. THE VISUAL STIMULI

3.1. Field Signals

In the present thesis, the modulation patterns of the visual stimuli are specified by a set of functions $x_1(t), x_2(t), ..., x_{64}(t)$, called field signals, where $x_i(t) \epsilon B$ and $B=\{0, ..., n\}$ is the set of all possible states in which the background of a field can be (e.g. *0* could mean a blue colored field, *1* a red colored field, and so forth).

In the BRI, the field signals have special properties; for example, they are deterministic, binary ($B=\{0,1\}$), periodic, orthogonal, and have desirable correlation properties (e.g. any segment of some specified length within a period is approximately uncorrelated with any other segment of the same length). The definition of the field signals is

$$x_i(t\ ) \ = \ \sum_{k=0}^{\infty} c_i(t\ -kT\ ) \tag{2}$$

where the 64 functions $c_i(\tau)$ are short (discontinuous) signals, called binary m-sequences, defined over $0 \leq \tau \leq T$. The binary m-sequences are defined later in this chapter. Equation 2 essentially expresses the fact that the field signals are <u>periodic</u>: the field signal $x_i(t)$ always repeats the binary m-sequence $c_i(t)$.

A property of the binary m-sequences is that $c_{i+1}(t)$ can be derived from $c_i(t)$ by a single wrap-around shift. Because of the above property, and because the field signals are periodic (as defined in equation 2), the field signals are identical except for a shift of time L as expressed in equation 3

$$x_i(t) \ = \ x_{i+1}(t-L) \tag{3}$$

For example, let *000100110101111* be the binary m-sequence $c_1(t)$. Then, the field signal $x_1(t)$ is (equation 2)

$x_1(t)$: *000100110101111 000100110101111 000100110101111* ...

and, by shifting this signal three times by four bits (equation 3), we obtain the following set of field signals

$x_1(t)$: *000100110101111 000100110101111 000100110101111* ...

$x_2(t)$: *001101011110001 001101011110001 001101011110001* ...

$x_3(t)$: *010111100010011 010111100010011 010111100010011* ...

$x_4(t)$: *111000100110101 111000100110101 111000100110101* ...

3.2. Binary M-sequence

Any binary m-sequence of length 2^m*-1* may be generated by using a special m-digit register. For example, the binary m-sequence (m=4) *000100110101111* is generated by the 4-digit register illustrated in Figure 5.

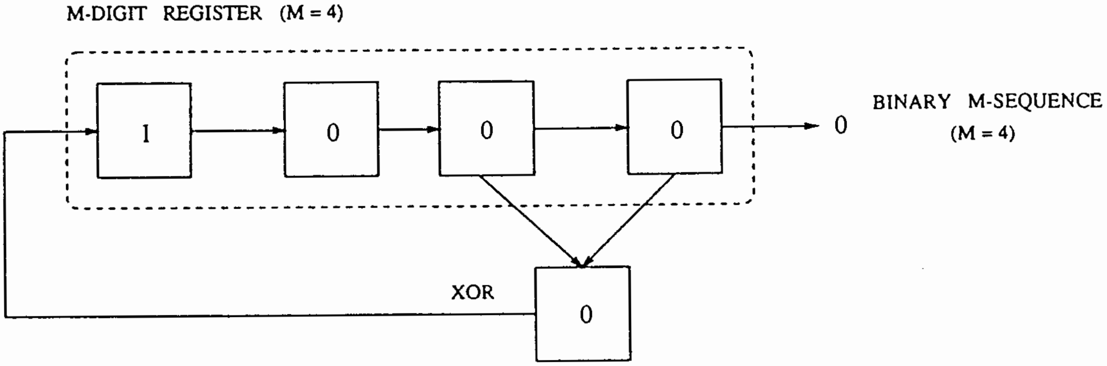

Figure 5. Binary m-sequence register (m=4).

The successive states of the register are described in Table I (the binary m-sequence is defined by the last digit of the register).

TABLE I

SUCCESSIVE STATES OF A BINARY M-SEQUENCE REGISTER
(M=4)

register state	XOR	binary m-sequence
1000	0	0
0100	0	0
0010	1	0
1001	1	1
1100	0	0
0110	1	0
1011	0	1
0101	1	1
1010	0	0
1101	1	1
1110	1	0
1111	0	1
0111	0	1
0011	0	1
0001	1	1

Binary M-Sequence Correlation Properties. Binary m-sequences share many characteristics with a white noise signal. When a binary m-sequence is transformed into a sequence of *+1's* and *-1's* by means of the substitution

$$\begin{aligned} 0 &\rightarrow 1 \\ 1 &\rightarrow -1 \end{aligned}$$

its auto-correlation function becomes

$$cor(k) = \sum_{i=1}^{2^m-1} x_i x_{i-k} = \begin{cases} 2^m-1, & \text{if } k = 0 \\ -1, & \text{if } k \neq 0 \end{cases} \tag{4}$$

where x_i is the i^{th} element of the m-sequence, k is the lag. The sequence is, thus, almost perfectly white over a complete cycle.

We can also compute the correlation between segments of a binary m-sequence

$$cor(k) = \sum_{i=1}^{N} x_i x_{i+k} \tag{5}$$

where N is the length of the segment of the binary m-sequence, k is the lag. For example, the correlation between the initial segment *111-1* of the binary m-sequence *111-111-1-11-11-1-1-1-1* and the sequence itself is indicated in Table II.

TABLE II

COMPUTATION OF THE CORRELATION BETWEEN SEGMENTS (N=4)
OF A BINARY M-SEQUENCE OF LENGTH 15

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$cor(k)$	4	0	0	2	2	-2	0	-2	2	0	0	-2	-4	-2	0

The results reported in Table II indicate that all segments of the binary m-sequence are different except when there is no lag (k=0). More generally,

$$\begin{aligned} cor(k) &= N, && \text{if } k = 0, \quad and \\ cor(k) &\leq N, && \text{if } k > 0 \end{aligned} \tag{6}$$

where N is the length of the segment. In other words, this means that the field signal never repeats itself.

In the present problem context, it also means that the initial segments of two different field signals are always different, and become less and less correlated as the length increases (this can be verified by computing the correlation for segments of various lengths). Thus we can hope that the corresponding brain-waves are also uncorrelated although we cannot demonstrate it analytically since the visual system is so nonlinear.

Other Properties. The m-digit register goes through all possible states except *0000* while generating the binary m-sequence (the *0000* state produces the sequence *0000000000...*). The states of the register are repeated in the same order indefinitely. This property of the binary m-sequence leads to several observations. First, the same binary m-sequence is also repeated indefinitely. Second, the length of a binary m-sequence generated by a m-digit register is equal to 2^m-1 (number of states possible minus one). Third, the initial state is irrelevant; the only effect of changing the initial state of the register is to change the starting point of the binary m-sequence. Fourth, there is a maximum of 2^m-1 binary m-sequences (of length 2^m-1) since there is a maximum of 2^m-1 initial states. These observations may help to understand why binary m-sequences of the same length can be derived one from another and, thereby, why all field signals are identical except for a shift of time. Since the initial state of the register is the only parameter that distinguishes two binary m-sequences, and since the register goes through all possible states while generating a single binary m-sequence, all binary m-sequences are identical but for a different starting point.

Note that a single register can be used to generate several binary m-sequences at once. For example, in Table III, the field signals x_1, x_2, x_3, and x_4 are obtained by repeatedly arranging the binary m-sequence *000100110101111* into four columns.

TABLE III

GENERATION OF MULTIPLE BINARY M-SEQUENCES
BY A SINGLE REGISTER

$x_1(t)$	$x_2(t)$	$x_3(t)$	$x_4(t)$
0	0	0	1
0	0	1	1
0	1	0	1
1	1	1	0
0	0	1	0
0	1	1	0
1	0	1	1
1	1	0	0
0	1	0	0
1	1	.	.
.	.	.	.

This property is used to generate the 64 field signals in the BRI.

4. ADDITIONAL REMARKS

4.1. Source Encoding

In this chapter, we have focused our interest on designing "noise-proof" signals (channel encoding). We have not explicitly discussed the issues related with source encoding although the speed of the transmission is a fundamental characteristic of the problem context.

Source encoding is based on prior knowledge of the probability distribution of the events. A classic method for speeding up the transmission is, for example, to assign short codewords to frequent events. However, this approach does not make sense in the BRI, because the transmission of the codeword is stopped as soon as it is identified. In a sense, the length of the codewords is automatically adjusted.

Nevertheless, it could be possible to improve the performance of the BRI by assigning visual stimuli whose initial segments are particularly "noise-proof" to frequent events. But discussing source encoding in detail in the present context is pointless since we do not have any information about the probability distribution of the events.

4.2. System Analysis

A Volterra series is a mathematical formalism that may be used to model the input/output relationship of linear and/or nonlinear systems. To avoid a layer of mathematical complexity, we assume that the input signals are discrete, and all integrals are replaced by sums. The following expression corresponds to a single-input system

$$\begin{aligned} y(t_n) = & \sum_k h(k)x(t_{n-k}) \\ & + \sum_{kl} h(k,l)x(t_{n-k})x(t_{n-l}) \\ & + \sum_{klm} h(k,l,m)x(t_{n-k})x(t_{n-l})x(t_{n-m}) \\ & + \;..... \end{aligned} \tag{7}$$

The kernels *h*, represent the system's response to the "cross-talk" among different segments of the input history. For example, *h(k,l)* expresses the cross-talk among two segments, *h(k,l,m)* expresses the cross-talk among three segments, and so forth.

In a multiple-input system, it is necessary to take into account all input signals

$$\begin{aligned} y(t_n) \;=\; & \sum_i \sum_k h_i(k) x_i(t_{n-k}) \\ & + \sum_{ij} \sum_{kl} h_{ij}(k,l) x_i(t_{n-k}) x_j(t_{n-l}) \\ & + \;\ldots.. \end{aligned} \tag{8}$$

The kernels of the form h_{ij} , h_{ijk} , *etc.*, correspond to the cross-talk between segments of different input signals.

In a system analysis context, the objective is to determine the values of the parameters of the chosen system model (e.g. the kernels for the Volterra series). For that purpose, it is necessary to get the maximum possible information out of the input/output relationship in order to be able to fully characterize the system. In the linear case, the system response to a single impulse is sufficient to characterize the system. In the nonlinear case, it is necessary to use white noise or deterministic signals with similar properties to extensively "test" the nonlinear system and to get an informative system response. For example, in (Sutter, 1987; Sutter, 1991), binary m-sequences are used as input signals to determine the (kernel) parameters of the visual system.

In the present problem context however, the objective is to identify the input signal (not the system). But, since binary m-sequences are also used in the BRI, we wondered if there was a common reason for using the same signals in the two different contexts. We do not think so because both objectives are different, and "testing" the system does not seem to be a relevant issue.

Marmarelis (1978) provides a good introduction to the general subject of mathematical analysis of nonlinear systems along with a general presentation and examples of biological modeling.

CHAPTER III

THE DECODING PROCESS

In a communication system, the function of the channel is essentially to transmit a signal from one point to another, and the role of the decoding process is to retrieve the information from the encoded signal at the receiving end of the channel.

In this chapter, we specify the desired input/output relationship that the decoding process should realize, and how it can be implemented. For clarity purposes, we divide the decoding process into three successive stages: the preprocessing, classification, and decision stage. The cross-correlation technique and the two possible usages of ANN technology described in this chapter correspond to the classification stage.

1. THE DESIRED INPUT/OUTPUT RELATIONSHIP

1.1. Inputs

In the present problem context, the input to the decoding process is the brain-wave signal evoked by the visual stimulus associated with the command being gazed at (see Figure 6). In the present thesis, these brain-wave signals are often called brain-wave responses.

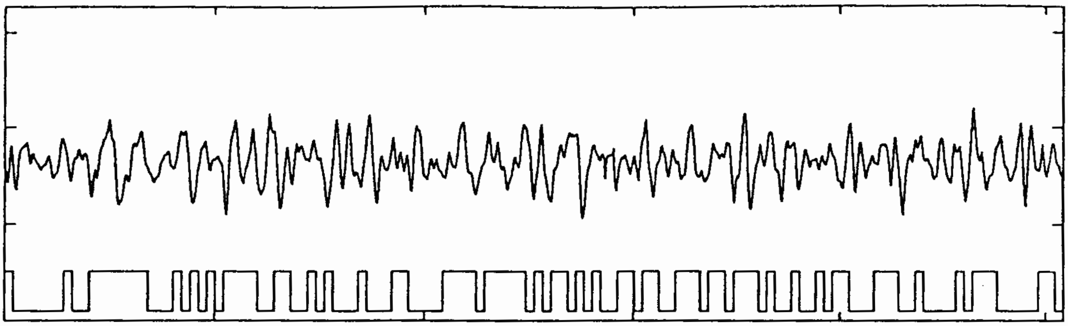

Figure 6. Example of a brain-wave response. The lower line corresponds to the binary m-sequence used as visual stimulus.

1.2. Desired Outputs

The output of the decoding process is a binary vector of dimension 64 that indicates the class of the brain-wave response, and, thereby, which command is being gazed at (by definition, all brain-wave responses that are evoked by the same visual stimulus belong to the same class). The code for the desired outputs is simple: the value of the nth output element is set to one, and the values of all other output elements to zero, if the input (brain-wave response) belongs to the nth class. Thus, the desired outputs are

class 1: 1000000000000000000000000000000000000000000000000000000000000000
class 2: 0100000000000000000000000000000000000000000000000000000000000000
. . . .
class 64: 0000000000000000000000000000000000000000000000000000000000000001

2. PREPROCESSING STAGE

Data preprocessing can be thought of as a preliminary stage of the decoding process. Data preprocessing is required when the format of the signal being detected does not fit the characteristics of the decoder, or when the performance of the decoding process are not satisfactory. It is an important issue that may involve many different operations: sampling, filtering, feature extraction, etc. The application of these techniques depends on the problem context.

2.1. Sampling

In the BRI, the brain-wave signal being detected is analog. Thus, it is necessary to sample and quantify the brain-wave signal in the first place because the techniques used to implement the decoding process require n-dimensional vectors (patterns) as inputs. Sampling is an important issue. Unfortunately, we did not have the information necessary to address it seriously in the present thesis. In our experiments, we used the data that were available in their original format (see Chapter V).

2.2. Filtering

Filtering techniques seek to improve the performance of the decoding process by isolating the information and the noise in signals being detected, and removing the noise.

Low-pass or high-pass filters, for example, are common filtering techniques that increase the signal-to-noise ratio by removing the frequencies that contain most of the noise. These filters, however, are difficult to use in the present problem context because the brain-wave responses are "embedded in colored noise which has temporal and spectral characteristics similar to the evoked potentials" (Uncini, 1990, 145).

Averaging is another filtering technique. This technique, however, is not compatible with the avowed objective of minimizing the response time, since its usage requires several brain-wave responses. In addition, numerous studies have shown that brain-wave responses are not perfectly stationary and may require the use of sophisticated, computationally expensive averaging techniques (Aunon, 1981).

2.3. Adaptive Filtering

There are various forms of adaptive filters reported in the literature. The following description of an adaptive filter is simple.

> The input signals generally originate from sampling continuous signals by analog-to-digital conversion. [...] The output is a linear combination of the current and past input signal samples. By varying the [coefficients of the linear combination], the [relationship] from the input to the output is directly controllable. The output is compared to a special input signal called the "desired response" [e.g. the next input signal sample]. The [coefficients] are usually adjusted to cause the output signal to be a best least square match over time to the desired response. (Widrow, 1988, pg. 250)

Adaptive filters can easily be implemented with the Artificial Neural Network (ANN) technology (Widrow, 1988). Adaptive filtering has generated a lot of interest recently because it does not require a priori knowledge of the signal and the noise function (Iyer, 1990, pg. 536). Besides, it avoids the ensemble averaging and time averaging when used in nonstationnary mode (in which case the coefficients of the linear combination are continuously and automatically updated). Note, however, that we can expect a problem in filtering a nonstationary signal in a very high noise environment (Thakor, 1987).

2.4. Feature Extraction

According to the problem context, it may be necessary to extract special "features" of the data. For example, in a similar problem context, Holdaway (1990) used a peak detection program to preprocess the brain-wave signals.

3. CLASSIFICATION STAGE

In this section, we describe the cross-correlation technique and introduce two types of Artificial Neural Networks (ANNs) used in the research reported in this thesis. In this chapter, we do not detail the ANN architectures since they may vary from one experiment to another (see Chapter V), nor do we address any specific ANN issue (see Chapter IV).

3.1. The Cross-Correlation Technique

The cross-correlation technique is based on 64 templates that represent the archetypal brain-wave responses to the 64 different visual stimuli (field signals) generated by the BRI monitor screen. We assume that the templates are all identical except for a shift of time because the visual stimuli are themselves identical except for a shift of time. Then, the brain-wave response R being detected is compared to each template by computing the correlation

$$cor(R,T_i,N) = \sum_{k=1}^{N} R(k)T_i(k) \qquad i = 1,..,64 \tag{9}$$

where N is the length of the brain-wave response, and T_i is the ith template. This computation supplies us with 64 values. Ideally, the output element with the highest value should indicate the class of the brain-wave response, and, thereby, which command is being gazed at.

Normalization. In the current implementation of the BRI, the templates T_1, ..., T_{64} are normalized

$$\boldsymbol{T}_i = \frac{T_i - \mu_{T_i}}{\sigma_{T_i}} \tag{10}$$

thus, the computation of the correlation becomes

$$\begin{aligned} cor(R,\boldsymbol{T}_i,N) &= \sum_{k=1}^{N} R(k)\boldsymbol{T}_i(k) \\ &= \frac{1}{\sigma_{T_i}}\{\sum_{k=1}^{N} R(k)T_i(k) - \mu_{T_i}\sum_{k=1}^{N} R(k)\} \end{aligned} \tag{11}$$

In the present thesis, we also consider the case where only the brain-wave response is normalized,

$$\begin{aligned} cor(\boldsymbol{R},T_i,N) &= \sum_{k=1}^{N} \boldsymbol{R}(k)T_i(k) \\ &= \frac{1}{\sigma_R}\{\sum_{k=1}^{N} R(k)T_i(k) - \mu_R\sum_{k=1}^{N} T_i(k)\} \end{aligned} \tag{12}$$

and the case where the templates and the brain-wave response are both normalized

$$
\begin{aligned}
cor(\boldsymbol{R},\boldsymbol{T}_i,N) &= \sum_{k=1}^{N} \boldsymbol{R}(k)\boldsymbol{T}_i(k) \\
&= \frac{1}{\sigma_R \sigma_{T_i}} \{ \sum_{k=1}^{N} R(k)T_i(k) - \mu_R \sum_{k=1}^{N} T_i(k) \\
&\qquad - \mu_{T_i} \sum_{k=1}^{N} R(k) + \mu_R \mu_{T_i} \}
\end{aligned}
\qquad (13)
$$

The four above cases are summarized in the following table

TABLE IV

CORRELATION EQUATIONS

Brain-wave response	Templates	
	Not normalized	Normalized
Not normalized	equation 9	equation 11
Normalized	equation 12	equation 13

It is obvious from the previous equations that the length of the brain-wave response and the length of the templates used in the calculation of the correlation are always the same. In order to understand the following remarks, we note that the addition of a constant to the 64 values computed by the cross-correlation technique, and/or the multiplication of these values by a coefficient, does not affect the decision being made about the class of the response, because this decision depends only on which is the highest value produced by the computation of the correlation.

Template Normalization. When the templates are normalized, we can always assume that the average and the variance of the templates are the same for every template because all templates are identical except for a shift in time (thus, they can be replaced by a constant in the previous equations). This is true because it is always possible to normalize the entire templates since they are known in advance. Thus, normalizing the templates does not affect the decision being made about the class of the brain-wave response.

Response Normalization. When the calculation of the correlation is based on the entire brain-wave response, the term

$$\sum_{k=1}^{N} T_i(k) \tag{14}$$

is the same for every template (and can be replaced by a constant in the previous equation). Thus, in this case, normalizing the brain-wave response does not impact its classification. This observation together with the previous one lets us affirm that entire brain-wave responses are always similarly classified whether or not the responses and/or the templates are normalized.

However, when the calculation of the correlation is based on only a segment of the brain-wave response, the classification of the response may be different depending on whether it has been normalized or not[5]. This not surprising since, in this case, the term in equation 14 is different for every template (this term was

[5] Note that normalizing the brain-wave response on the basis of its full length, or on the basis of a segment of it can change the decision being made about the class of the input because the average of the response varies as the length of the brain-wave response increases. The fact that the average of the brain-wave response is independent of i (independent of the templates) does not necessarily imply that it can be neglected because, in equation 12 and 13, the average of the brain-wave response is part of a term that depends on the templates. However, we have determined empirically (on the basis of the available data, see Chapter IV) that we can assume that the average of the brain-wave response is independent of the templates except when very short segments of the brain-wave responses are considered ($N<100$).

introduced when the brain-wave response was normalized, see equation 12 and 13), In Chapter V, we report better results when the brain-wave response was normalized. Logically, the difference is particularly striking when short segments are considered.

Memory And Computational Requirements. In principle, the 64 templates need to be stored in memory. However, since the templates are identical except for a shift of time, one can use 64 different pointers directed to a single template in memory to represent the 64 templates.

The computation of the correlation can easily be carried out on-line. For example, consider equation 13. Normalizing the templates does not require any real-time computation since it can be done in advance. For the brain-wave response however, the normalization must be recaclulated as its length increases. Fortunately, it is possible to reduce equation 12 and 13 to the following

$$\sum_{k=1}^{N} R(k)T_i(k) - \mu_R \sum_{k=1}^{N} T_i(k) \tag{15}$$

without affecting the decision being made about the class of the input. Thus, by keeping (and updating) the following values in memory

$$\begin{aligned} M_{1i}(N) &= \sum_{k=1}^{N} R(k)T_i(k) \\ M_2(N) &= \sum_{k=1}^{N} R(k) \\ M_{3i}(N) &= \sum_{k=1}^{N} T_i(k) \end{aligned} \tag{16}$$

the computation of equation 13 is reduced to

$$M_{1i}(N) - \frac{1}{N}M_2(N)M_{3i}(N) \tag{17}$$

3.2. First ANN Architecture Studied

The first ANN architecture used for the present research is a straightforward implementation of the desired input/output relationship. It consists of a single (backpropagation) ANN with 1016 input elements (maximum length of the brain-wave response) and 64 output elements (size of the desired outputs) as shown in Figure 7.

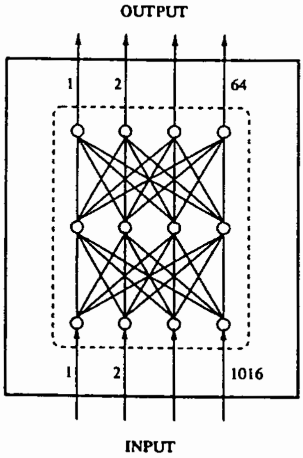

Figure 7. First ANN architecture. One single ANN with 1016 input elements and 64 output elements, and one "hidden"layer.

3.3. Second ANN Architecture Studied

The second ANN architecture used for the present research consists of 64 sub-ANNs, each sub-ANN having to identify one of the classes. Each of the sub-ANNs has only one output element: the desired output value is *1* if the sub-ANN identifies the input as an element of its own class, and *0* otherwise. An important observation is that by assuming that a shift in time of the visual stimulus effects a similar shift in time of the brain-wave response and by using a proper reencoding of the input pattern, it is possible to use the same sub-ANN 64 times as shown in Figure 8.

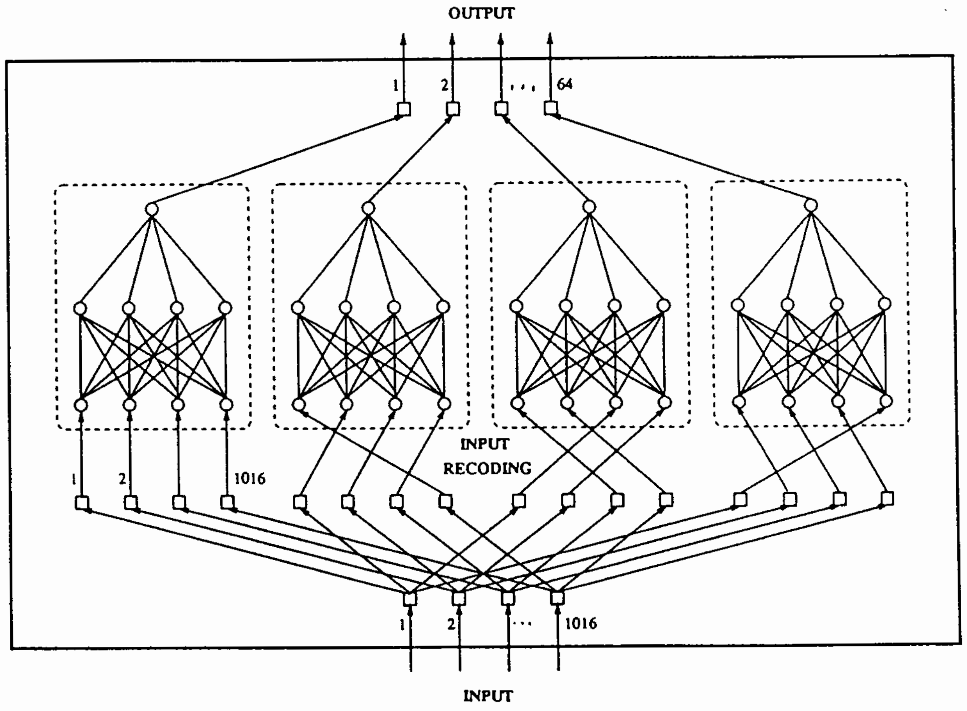

Figure 8. Second ANN architecture. 64 identical sub-ANNs, each having 1016 input elements and 1 output element. The input for each sub-ANN is modified by a wrap-around shift.

4. DECISION STAGE

The desired outputs are binary vectors. In contrast, the usual techniques used to categorize the brain-wave-responses produce real-valued output vectors. It is therefore necessary to complete these techniques with an additional stage, called the decision stage, that converts the actual real-valued output vector into a binary vector. In a sense, the role of the decision stage is to provide an interpretation of the output values produced by the decoder. It may be useful to design the decoder so that the actual outputs approximate the desired outputs. Several different methods can be used at the decision stage. All methods are subject to errors since there is no warranty that the real-valued output contains the necessary information to choose the correct desired output. In the next paragraphs, we define a typology of the possible errors and propose several methods that may be used at the decision stage.

4.1. Error Types

If one considers the output elements individually, two types of error are possible. The decoding process (after the decision stage) may have produced a one instead of a zero, or a zero instead of a one. It is however more interesting to consider the 64 output elements together to define a typology of the possible errors. An output is said to be formally valid if only one output element has a one as output value. All desired outputs are formally valid. A type 1 error is said (here) to arise when the decoding process produces a formally valid but incorrect output. A type 2 error arises when the decoding process produces only zeros. A type 3 error arises when the decoding process produces several ones including the correct one. A type 4 error arises when the decoding process produces several ones, all of them incorrect.

For example, if the desired output is class 1

class 1: *100000000000000000000000000000000000000000000000000000000000000000*

the four possible error types are illustrated as follows

error type 1: *010000000000000000000000000000000000000000000000000000000000000000*
error type 2: *000000000000000000000000000000000000000000000000000000000000000000*
error type 3: *100000000000000000000000100000000000000000000000000000000001000000*
error type 4: *000100000001000000000000000000000000000010000000001000000000000001*

Notice that if the correct answer is not known, it is impossible to distinguish between the correct output and a type 1 error (both look like desired outputs), or between a type 3 error and a type 4 error. Nevertheless, this distinction is useful because one can introduce a bias in the decoding process to decrease the number of errors of one type at the expense of another when designing it. This makes sense if the cost associated with a particular error type is very high.

4.2. Decide via Maximum Method

The maximum method assigns a *1* to the highest (analog) value and *0* to the rest. This method always generates a formally valid (but possibly incorrect) output. Only type 1 errors are possible.

4.3. Decide via Threshold Method

The threshold method assigns a *1* to all values higher than some threshold value and *0* to the rest. In other words, a minimum "level of confidence" is required. This method allows all four previously defined error types to arise. It has the advantage that many errors are signaled by invalid output; for example if all real valued outputs are low (type 2 error) or if more than one output is higher than the threshold (type 3 error and type 4 error), this information can be used to discard the output.

4.4. Decide via Threshold-Maximum Method

The threshold-maximum method is an attempt to overcome the cons of each method by combining them. The idea is to select all real valued outputs higher than some threshold and assign a *1* to the highest of these. The rest of the output values are set to *0*. Only type 1 and type 2 errors are possible with this method.

CHAPTER IV

ARTIFICIAL NEURAL NETWORKS

1. FUNDAMENTALS

1.1. Recall Process

Processing elements (PEs) and connections are the basic components of an ANN architecture: ANNs are "parallel, distributed information structures consisting of processing elements interconnected together with unidirectional signals called connections" (Hecht-Nielsen, 1988). Each connection has a weight that represents the strength of the connection. The processing elements carry out only local information processing operations which depend upon the current values x_i of the PE's input signals and upon the current values w_{ji} of the weights

$$y_j = f(\sum_{i=1}^{n} w_{ji} \, x_i) \tag{18}$$

where f can be an operation of any mathematical type (e.g. threshold, sigmoid). The function f is usually called the transfer function.

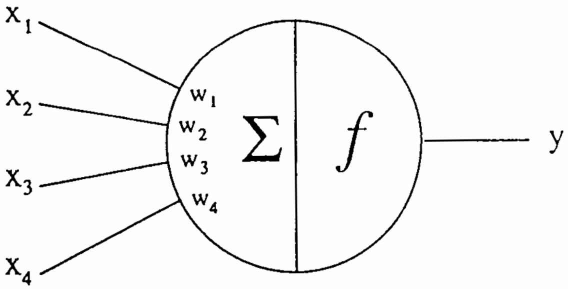

Figure 9. A processing element.

The information processing operations that lead to the computation of the output of the ANN for a given input are collectively referred as the recall process. These operations are not entirely parallel: ANNs process the input signals sequentially, in a few steps, each step involving a large number of PEs. This observation leads to the characterization of ANNs in terms of groups of processing elements, called layers. Typically, all processing elements in a layer share a common interconnection scheme and process the information simultaneously. Feedforward ANN architectures allow information to flow in only one direction, from the input layer to the output layer, while feedback or recurrent ANN architectures allow the information to flow in both directions.

The input layer consists of all processing elements that receive a signal from the environment[6]. The output layer consists of all processing elements that emit a signal to the environment. Any layer that lies between the input and output layer is called a hidden layer.

[6] The input layer is essentially a buffer (the PEs simply store the current value of the signals).

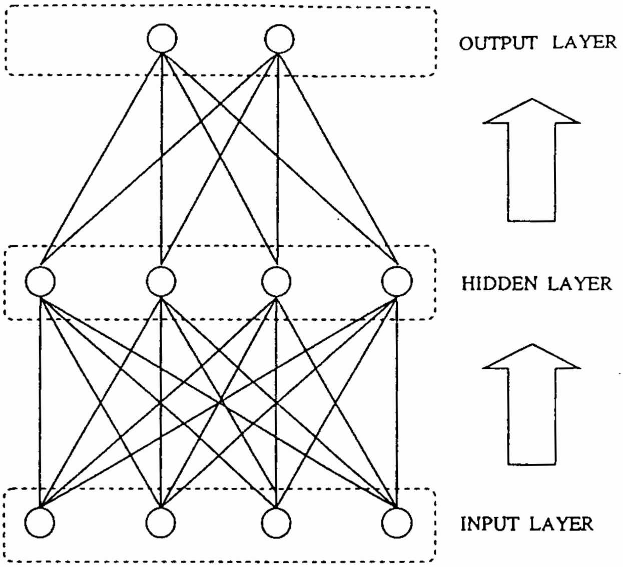

Figure 10. A feedforward fully connected ANN with one hidden layer.

1.2. Training Process

Once the architecture is fixed, the actual input/output relationship of an ANN depends entirely on the values given to the weights. Since an ANN can contain several thousand weights (more than 30,000 in our experiments), discovering a weight configuration that corresponds to the desired input/output relationship is not a trivial task. ANNs solve this problem by learning the desired input/output relationship during a training process. Learning is defined to be any change in the weight configuration: the process can be either supervised or unsupervised. In the case of supervised learning, the objective is to teach the ANN a predetermined input/output relationship. Global information about the problem context (e.g. the desired output) can be incorporated in the learning process. In the case of unsupervised learning, the input/output relationship is not known at the beginning of the training process, rather it emerges as the result of the process. In a sense, the resulting input/output relationships reflects the properties of data that the ANN has discovered during the learning process. In the present thesis, we consider only the (supervised) learning algorithm called backpropagation.

1.3. Backpropagation

Typically, backpropagation is applied to feedforward architectures. Backpropagation ANNs are characterized by a particular training process that consists of the following elements: a training set, a criterion function, and a learning rule (these elements are not exclusive to the backpropagation paradigm).

The training set contains data representative of the problem context (i.e. representative of the desired input/output relationship).

The criterion function provides a global measure of the error associated with the current weight configuration. For example, the criterion function typically computes the difference between the actual output and the desired output value for each PE in the output layer. In theory, the criterion function should take into account the error associated with all the input patterns that are relevant to the current problem context (i.e. all inputs that are specified by the desired input/output relationship). It is convenient to think of the criterion function as a surface in the space of all possible weight configurations. This surface is sometimes called the error surface.

The learning rule interprets the data supplied by the criterion function and determines the next weight configuration to use. The objective of the training process is to find a weight configuration that minimizes the criterion function. In the backpropagation paradigm, the error e_i associated with the PEs in the output layer is backpropagated in successively lower layers

$$e_j = f'(\sum_i w_{ji} e_i) \tag{19}$$

where e_j is the error associated with the PE in a lower layer, w_{ji} are the weights associated with the connections between the upper and lower layer, f' is the derivative of the transfer function at the point corresponding to the value of the PE's output, x_j. Finally, the error associated with each processing element is used to update the weight as follows

$$w_{ji}^{new} = w_{ji}^{old} + w_{ji}^{old} e_i \tag{20}$$

From an abstract viewpoint, the backpropagation training process is functionally equivalent to a gradient descent algorithm operating on the error surface in weight space.

2. ARTIFICIAL NEURAL NETWORK ISSUES

2.1. Credit Assignment Problem

As previously mentioned, the criterion function provides only a global assessment of the performance of an ANN. Hence, the learning rule must include a mechanism that interprets the data provided by the criterion function and assigns the "credit" (or the "blame") to each weight individually. The backpropagation algorithm offers a general solution to this problem for multi-layered, feedforward ANNs. This explains the popularity of the backpropagation paradigm today.

2.2. Local Minimum Problem

As previoulsy mentioned, the backpropagation learning rule is functionally equivalent to a gradient descent algorithm operating on the surface determined by the criterion function. In this case, the local minimum problem originates from the fact that the criterion function is evaluated only locally. Representing the criterion function as a surface is convenient but does not reflect the fact that the points of the surface are known only one at a time (for the current weight configuration). Thus, depending on the initial weight configuration, such a learning algorithm can get stuck in a local minimum (unsatisfactory weight configuration).

In fact, the local minimum problem is essentially a computational problem since any systematic approach to solve it would require the computation of more points of the surface; in other words, more information about the problem context or more guidance. In a sense, this problem is shared by every type of learning machine.

2.3. Generalization Problem

The term 'generalization' relates to the act of generating an output for an input not seen during the training process. The generalization problem is thus connected with the fact that (in most cases) the training set is only a partial definition of the input/output relationship. One side effect of this is that a bias is introduced in the computation of the criterion function because, in principle, all inputs relevant in the problem context should be considered. In practice, the performance of the ANN depends on the size and quality of the training set. Hence, a good understanding of the problem context is necessary to be able to select appropriate data for the training set.

Overfitting Problem. The overfitting problem is a special case of the generalization problem. It arises when the ANN performs very well on the training set but poorly on any other data set. In a sense, the ANN has memorized the training set without learning the underlying features that characterize the problem context. The ANN cannot generalize (well). A small training set, or a noisy set of data can cause the problem. In the case of a noisy data set, the problem arises because the ANN learns the features of the noise along with the features of the data; it then performs poorly on data that have been contaminated by a different noise.

2.4. Representation problem

From an abstract viewpoint, an ANN is a box with an input and an output. However, the set of possible input/output mappings that the ANN can actually perform is limited by its architecture: ANNs are not meant to be universal Turing machines, they do not process the information symbolically. In addition, the number of achievable input/output relationships is further reduced if one takes into consideration the training process, because in effect the training process creates "attractors" as it brings the weights to a particular configuration associated with a local minimum (only input/output mappings with large "attractors" are likely to be learned). In practice, solving the representation problem means figuring out an appropriate input/output mapping for a particular ANN architecture given a specific problem context. The representation problem is possibly the most complex issue; it has not yet received a satisfactory theoretical solution.

2.5. Memory and Computational Requirements

ANNs typically have many connections. Each connection is associated with a weight that must be stored in memory. This creates a serious problem as the size of the ANN scales up. Parallel ANN implementations solve memory-related problems by storing locally the value of the weight into specialized processors associated with the PEs.

We distinguish the computational costs associated with the recall and the training process. The recall process requires the computations of a sum and several products for each PE in the ANN. Although an ANN can contain many PEs and connections, it is possible to design fast parallel implementations by allotting a specialized processor to each PE, since all information processing operations are local. The training process is much more computationally expensive. In a backpropagation ANN, each presentation of a datum during the training process requires twice as many computations (the error needs to be backpropagated). More significant is the fact that backpropagation ANNs usually require many presentations to learn an appropriate weight configuration (15,000 to 500,000 in our experiments). On one side, computational requirements associated with the training process may not seem relevant since the process does not have to be repeated once the ANN has learned the desired input/output relationship. But, on the other side, the length of the training process can become a very significant obstacle while designing an ANN. Indeed, the design stage is still an empirical process and usually requires many experiments before an adequate ANN architecture is found.

3. THE ART OF ARTIFICIAL NEURAL NETWORK DESIGN

Designing an ANN that performs well on a given problem context raises many problems. First, an ANN paradigm must be chosen (e.g. backpropagation). This choice depends on the problem context.

Then, in many cases, the basic ANN paradigms must be modified in order to obtain satisfactory performance (e.g. decrease the number of errors, speed up the training process). Some of these modifications have become usual (e.g. the addition of a momentum term to the learning rule), others must be specifically designed to fit special characteristics of the problem context (e.g. the modification of the backpropagation algorithm made by the author, and detailed in Appendix A).

Finally, once the general characteristics of the ANN have been fixed, it is still necessary to determine the value of many parameters (e.g. the size of the hidden layer, the epoch size). Note that it is possible for some of these parameters to change during the training process (e.g. the learning coefficients). Note also that the distinction between a modification of an ANN paradigm and a parameter, is somewhat arbitrary.

3.1. The Design Issue

This sub-section gives a quick (and partial) account of the current state of development of ANN technology theoretical knowledge.

It has been proved that a three layer ANN can approximate a wide range of input/output relationship to any degree of accuracy (White, 1988). Unfortunately, this result states only that a mapping exists; it does not state how to find the mapping, nor the required number of PEs in the hidden layer. There is not yet any theory answering these questions.

As a matter of fact, full fledged theories are just beginning to be developed. To give an indication about the direction of the current state of the art, many of these theories are based on ideas borrowed from Information Theory. Their objective essentially is to match the "complexity" of the problem context with the "complexity" of the ANN (e.g. in simple models, the complexity of an ANN is sometimes related to the number of weights).

For example, there is an obvious relationship (and experimental confirmations) between, say, the overfitting problem and the "complexity" of the ANN. Previously, we partially described the overfitting problem as primarily depending on the size and characteristics of the training set. But, in fact, if the ANN is too "simple" relatively to the "complexity" of the problem context, then it will never learn how to generalize, since it will never be able to learn in the first place (generalization problem). And if the ANN is too "complex", then it will not be constrained enough by the training process (overfitting problem).

The slow progress toward a full-fledged theory is easy to understand, since it should probably include an evaluation of the "complexity" of the desired input/output relationship (i.e. the problem context), a description of the available data sets (e.g. the sizes of the training and testing sets), a description of the architecture of the ANN (e.g. the number of weights), and a description of the training process (e.g. the learning schedule). In addition, such a theory should also be backed up by a deep understanding of the problem context because such a knowledge is necessary to estimate the "complexity" of the problem context, and to define an appropriate input/output relationship, and to select the data sets.

In the previous paragraphs, we basically discussed the design issue as if it were a theoretical problem that should be addressed before doing any experiments. Another approach is to let the ANN adapt its architecture during the training process. For example, pruning is a technique that minimizes the number of connections and of PEs during the training process. Other techniques start from a simple architecture and increase the number of connections and PEs as long as it is necessary. Although these techniques can (sometimes) automate the design process, all have the drawback that the training time tends to be greatly increased.

3.2. Epoch Size

In this sub-section, we discuss a parameter, the epoch size, whose importance is too often neglected. Of course, the epoch size is not the only important parameter.

The epoch size refers to the number of presentations before the weight configuration is updated. For example, in the original definition of the backpropagation algorithm, the epoch size should theoretically be the size of the entire desired input/output relationship. In practice however, the epoch size is limited by the size of the training set. Empirically, this turns out not being a major problem since, in many cases, the training process succeeds with a small epoch size.

The impacts of the epoch size on the training process are just starting to be studied theoretically. As a matter of fact, one must be aware that computing the criterion function on the basis of only a subset of the required data changes its value. In addition, the order of presentation of data becomes a factor. For example, let x_1 and x_2 be two elements of the training set, let W_1, W_2, W_3 and W_4 be different weight configurations, suppose that the learning rule changes the weight configuration each time a data is presented as specified below

$$\begin{array}{c}(W_1, x_1) \text{ -> } W_2\\ (W_1, x_2) \text{ -> } W_3\\ (W_2, x_2) \text{ -> } W_3\\ (W_3, x_1) \text{ -> } W_4\end{array}$$

then a different order of presentation will produce a different result

$$\begin{array}{c}(W_1, x_1, x_2) \text{ -> } W_3\\ (W_1, x_2, x_1) \text{ -> } W_4\end{array}$$

CHAPTER V

EXPERIMENTS AND RESULTS

From an abstract viewpoint, the decoding process is simply a box with 1016 input elements and 64 output elements. In this chapter, we evaluate the performance of the cross-correlation technique and the ANN technology on the basis of the available data. We demonstrate that the ANN technology can perform at least as well as the cross-correlation technique currently used in the BRI. In addition we describe two particular phenomena that, if solved, could lead to a significant improvement of the performance.

1. METHODOLOGICAL ISSUES

1.1. Data

The data used in the present research were collected at the Smith-Kettelwell Eye Research Institute by the team of Dr. Erich E. Sutter. A set of 99 brain-wave responses was recorded by the means of an external electrode applied on the scalp of a healthy subject. An unique binary m-sequence was used as visual stimulus. The analog signal recorded by the electrode was then sampled and quantified. After having examined the data, we suppressed the eight initial anomalous values that were always equal to 32768, and, thereby, reduced the length of the data to 1016 samples. In addition, after a discussion with Dr. Sutter, we set the value of the 527th sample to the value of the 526th sample in the data (the value of the 527th sample was equal to zero in all data for an unknown reason).

For methodological reasons, the data are divided in two sets. The first data set contains 66 of the 99 brain-wave responses, and these are used to define templates required by the cross-correlation technique or to construct the training sets required by the ANN methodology. The second data set comprises the 33 remaining responses and is used to evaluate the performance of the techniques used in the present thesis to implement the decoding process.

1.2. Training Set

Defining the training set is a very important step of the ANN methodology. In the present problem context, it is obviously necessary to include data representative of the brain-wave responses induced by each of the 64 visual stimuli into the training set. In the present research, we constructed the training set by assuming that a shift of time of the visual stimulus produces a similar shift of the brain-wave response. Thus, each brain-wave response included in the first data set (66 responses) was successively shifted 63 times to simulate the brain-wave responses associated with the other visual stimuli. In other words, the new training set was constructed by augmenting the original data set with 63 shifted copies of it as shown in Figure 11. In this manner, we defined 64 classes: class 1 designates the responses that have not been shifted, class 2 designates the responses that have been shifted only once, class 3 designates the responses that have been shifted two times, and so on.

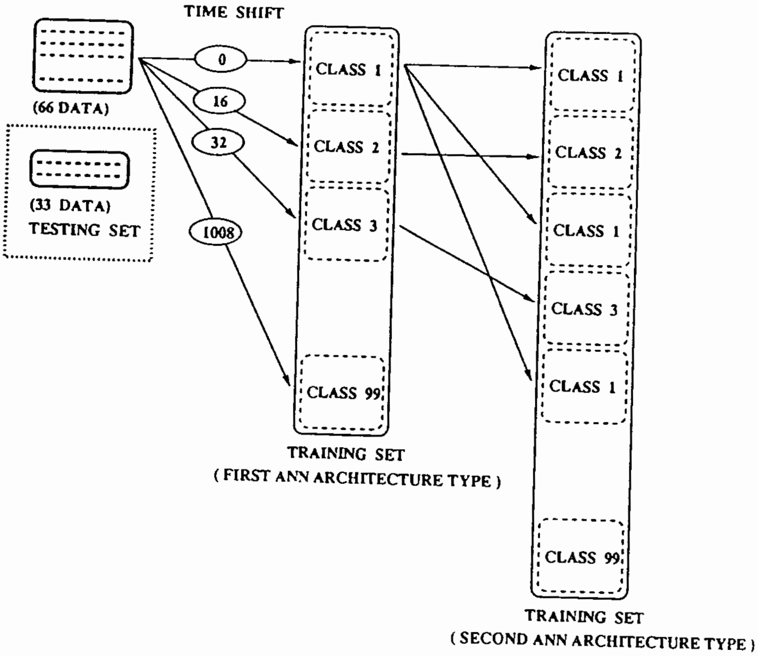

Figure 11. The data sets. This figure illustrates how the training sets used in the ANN experiments are derived from the original data set of 99 brain-wave responses.

In experiments involving the second ANN type (see Figure 8), we further augment the training set by duplicating 63 times every response in class 1 (unshifted brain-wave response). The objective of this last modification of the training set is to equalize the number of responses whose desired output is 1 (responses in class 1) with the number of responses whose desired output is 0 (responses in class 2, ..., and class 64). This is done in an effort to avoid the typical results wherein an ANN learns to always output a 0, when the number of responses whose desired output is 0 exceeds by far the number of responses whose desired output is 1. In experiments involving the first ANN type (see Figure 7), there is not such a motivation to duplicate the responses in class 1, since the desired outputs are always 64-dimensional vectors with one 1 and 63 0s (this does not mean that the encoding of the desired outputs is appropriate).

1.3. Evaluation of the Performance

The 33 responses in the testing set represent different brain-wave responses that have all been evoked by the same visual stimulus. Let us first remark that 33 responses constitute a very small testing set. However, it did not make sense to augment artificially the size of the testing set by shifting the responses to simulate the brain-wave responses to other visual stimuli. As a matter of fact, this would not provide any additional information about the performance of the techniques used in the present research, because, in the case of the cross-correlation technique, the templates are themselves identical except for a shift of time, and because, in the case of the ANN technology (type 2 ANN), the 64 sub-ANNs are identical. In both cases, the only effect of shifting the input is to shift the output.

Thus, ideally, the results should indicate that all responses in the testing set belong to class 1 as indicated in Figure 12[7]. We note that in some cases, when ANN technology was used, some experiments were run more than once in order to render the experiment results less sensitive to the random elements of the experimental process (e.g. initial value of the weights). In these cases, the results that are reported

[7] In the present thesis, we used the graphic capabilities of MATLAB to enhance the presentation of the results (MATLAB is a trademark of The MathWorks, Inc).

correspond to the average performance between the different realizations of the same experiment.

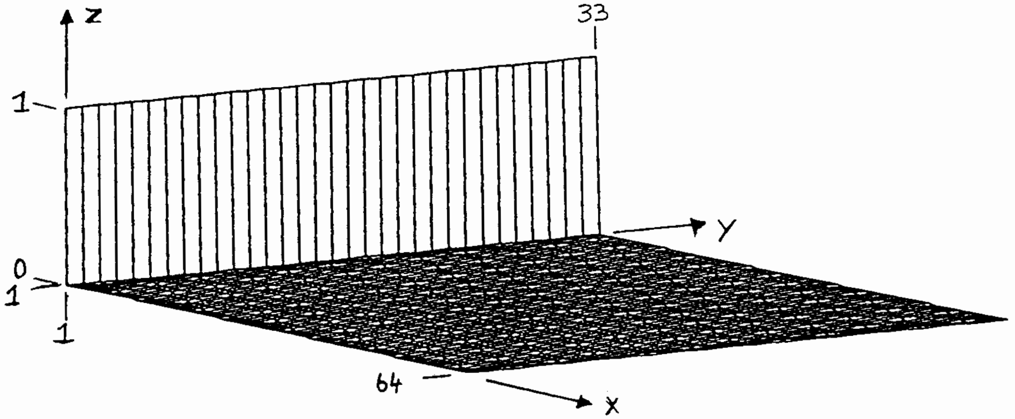

Figure 12. Three dimensional representation of the ideal case when all the responses in the testing set are classified correctly. The x-axis corresponds to the 64 output elements, the y-axis corresponds to the 33 responses in the testing set, and the z-axis indicates the value associated with the output elements.

Figure 13 illustrates an alternative format to communicate the results that is adequate when the output values (z-axis) are binary.

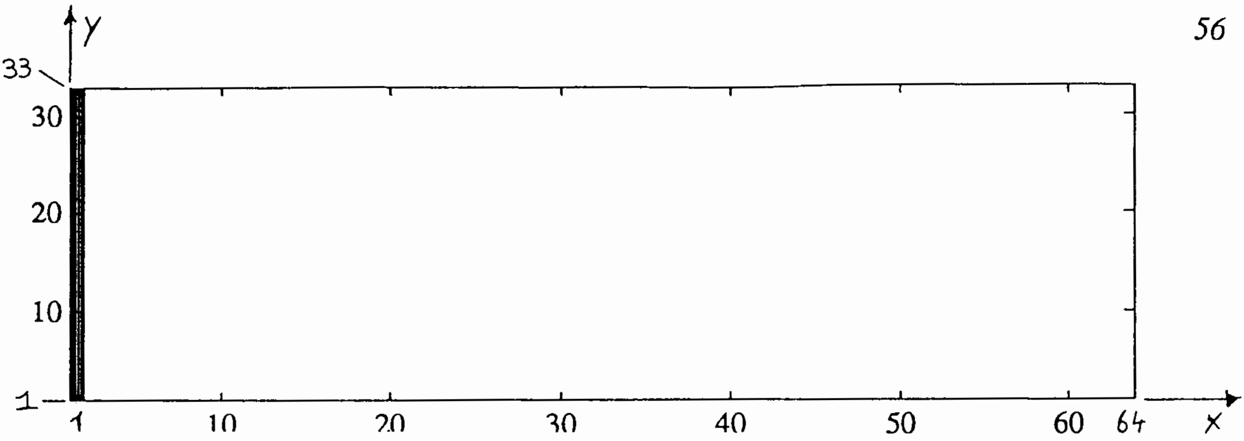

Figure 13. Two dimensional representation of the ideal case when all the responses in the testing set are classified correctly. The x-axis corresponds the 64 output elements, the y-axis corresponds to the 33 responses in the testing set.

2. CROSS-CORRELATION TECHNIQUE

The first step of the cross-correlation technique corresponds to the definition of the templates. In the present thesis, the first template, T_1, is obtained by averaging 66 brain-wave responses. The 63 other templates are derived from the first template by successive wrap-around shifts.

The second step of the cross-correlation technique is the computation of the correlation. The responses in the testing set and/or the templates are eventually normalized. The four possible cases are indicated in Table V. This issue was discussed in detail in Chapter II.

TABLE V

CROSS-CORRELATION TECHNIQUE:
THE FOUR POSSIBLE CASES

Brain-wave response	Templates	
	Not normalized	Normalized
Not normalized	case A	case B
Normalized	case C	case D

The time necessary to (correctly) classify the brain-wave response being detected is the most important criterion by which the BRI is evaluated. Figure 14 shows the percentage of correct classifications as a function of the size of the segment of the brain-wave response being used in the calculation.

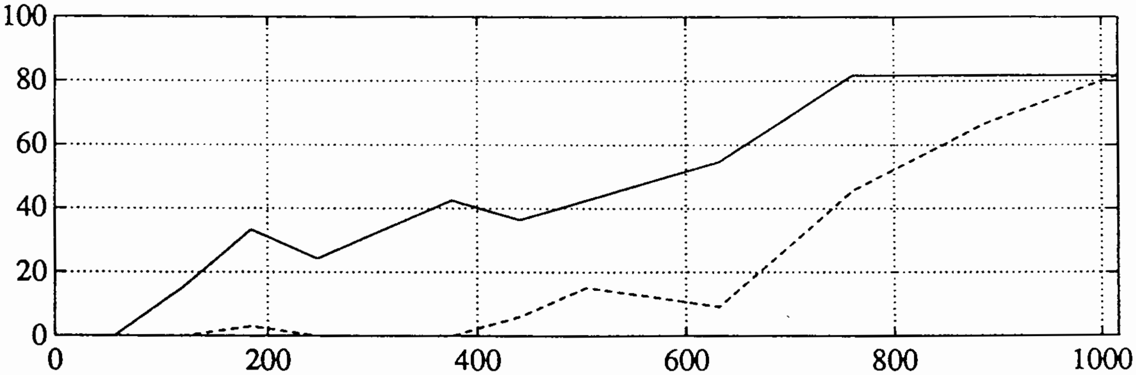

Figure 14. Percentage of correct classifications by the cross-correlation technique as a function of the length of the response. The dashed line corresponds to cases A and B (the brain-wave response has not been normalized), the solid line to cases C and D (the brain-wave response has been normalized).

As one can see, better classification performance is achieved when the brain-wave responses are normalized.

Figure 14 provides an experimental confirmation of two observations that were made in Chapter III. First, normalizing the templates does not impact the classification of the brain-wave responses in the testing set (therefore only two lines are used in Figure 14). But, normalizing the brain-wave response impacts the classification of the responses except when the entire brain-wave response is taken into account in the computation of the correlation (both lines indicate the same number of misclassifications when the length of the response is 1016).

Figure 15 shows how the brain-wave responses in the testing set are classified when entire responses are used in the calculation of the correlation (Figure 15 is the same for all four cases because, as previously mentioned, the classification performance is the same in all four cases when entire responses are used).

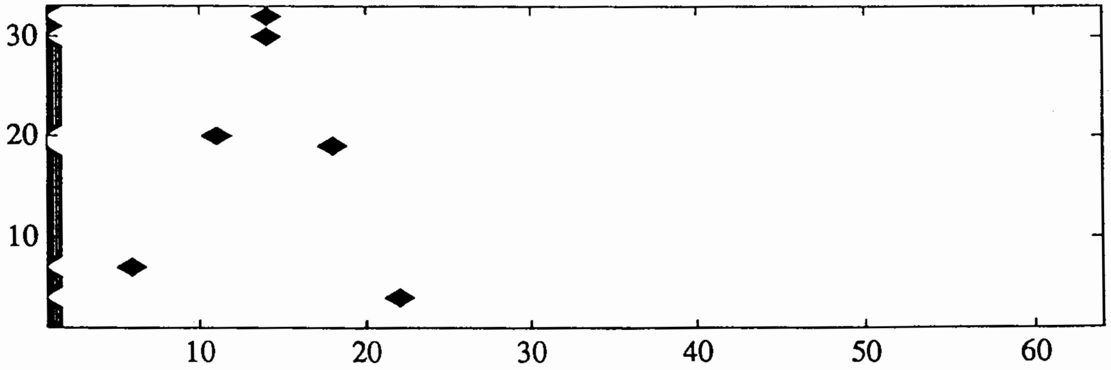

Figure 15. Classification of the brain-wave responses by the cross-correlation technique. Entire responses are used in the calculation of the correlation, and the maximum method is used at the decision stage. The x-axis corresponds to the 64 templates, the y-axis corresponds to the 33 responses in the testing set (see Figure 13 for perfect results).

3. FIRST ARTIFICIAL NEURAL ARCHITECTURE

In this research, experiments were carried out with two different ANN types. The general form of each ANN type is described in Chapter III. The first ANN type yielded poor results and is described here for completeness. A summary of the experimental conditions is given in Table VI. In every case, a single ANN with 64 PEs in the output layer is used (see Figure 7).

TABLE VI

FIRST ANN ARCHITECTURE EXPERIMENTS

X4	
PEs	1016, 30, 64
Transfer function	sigmoid
Learning rule	delta
Learning schedules[8]	default, hidx4, outx4
Epoch size	16
Input range	0, 1
Output range	0.2, 0.8
X7	
PEs	1016, 30, 64
Transfer function	tanh
Learning rule	cum-delta
Learning schedules	default, ncmhid, ncmout
Epoch size	66
Input range	-1, 1
Output range	-0.6, 0.6
X8	
Learning schedule	default, ncmhid2, ncmout2

[8] The learning schedules are different for the input, hidden, and output layers. The learning schedules are described in Appendix C.

The first ANN type failed to learn to correctly classify the responses. In every case (after 500,000 presentations), the ANN did not exhibit any sign of learning: the root mean square (RMS) error did not change at all. The problem may be related to the representation issue, as during most of the training process, the output elements are learning to produce a 0. Another possible reason for the failure of this approach is that we may have stopped the training process to early. However, 500,000 presentations seemed large when compared to the 15,000 (or even fewer) presentations required by the second architecture type.

If more experiments were to be run with the first ANN type, two suggestions are to change the representation of the desired outputs, and to include more than one hidden layer in the ANN.

4. SECOND ARTIFICIAL NEURAL NETWORK ARCHITECTURE

The second ANN type uses a combination of 64 identical sub-ANNs, each having only 1 PE in the output layer (see Figure 8).

4.1. Evaluation of the Performance

Figure 16 shows the percentage of correct classifications as a function of the length of the brain-wave response. The few results reported in Figure 16 are representative of all experiments with the second ANN type.

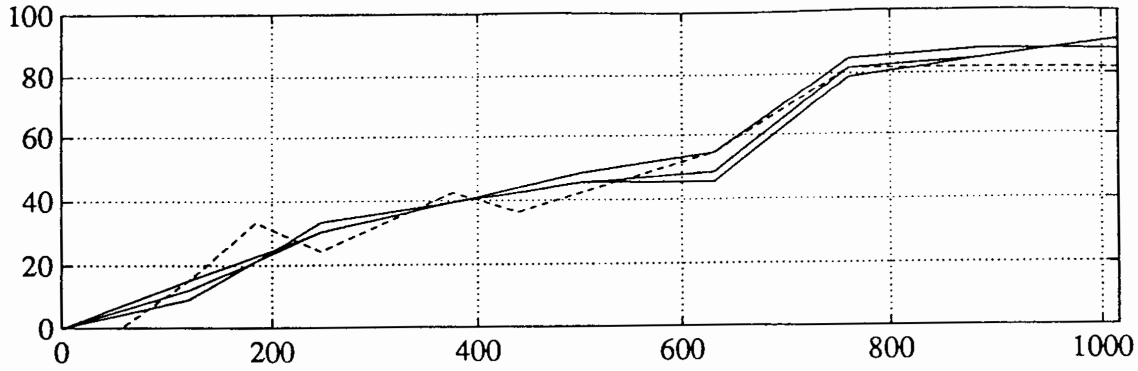

Figure 16. Percentage of correct classifications by the ANN technology as a function of the length of the response. Maximum method used at the decision stage. ANN experiments (solid line), cross-correlation technique (dashed line).

As one can see, the results reported in Figure 16 show the ANNs did as well as the cross-correlation technique, but did not achieve the hoped improvement of classfication rate on the shorter response segments.

In the subsequent paragraphs, we discuss the results that we obtained.

4.2. Comparison with the Cross-Correlation Technique

Even though the classification performance obtained with the cross-correlation technique (case C or D), and those obtained with the ANN technology are similar, there are major differences between the two techniques. For example, Figure 17 shows that the values associated with the 64 output elements are very different in each case.

The relative difference between the value of the first output element (whose desired output value is 1), and the value of the other output elements (whose desired output value is 0) is usually larger for the ANN technology than for the cross-correlation technique.

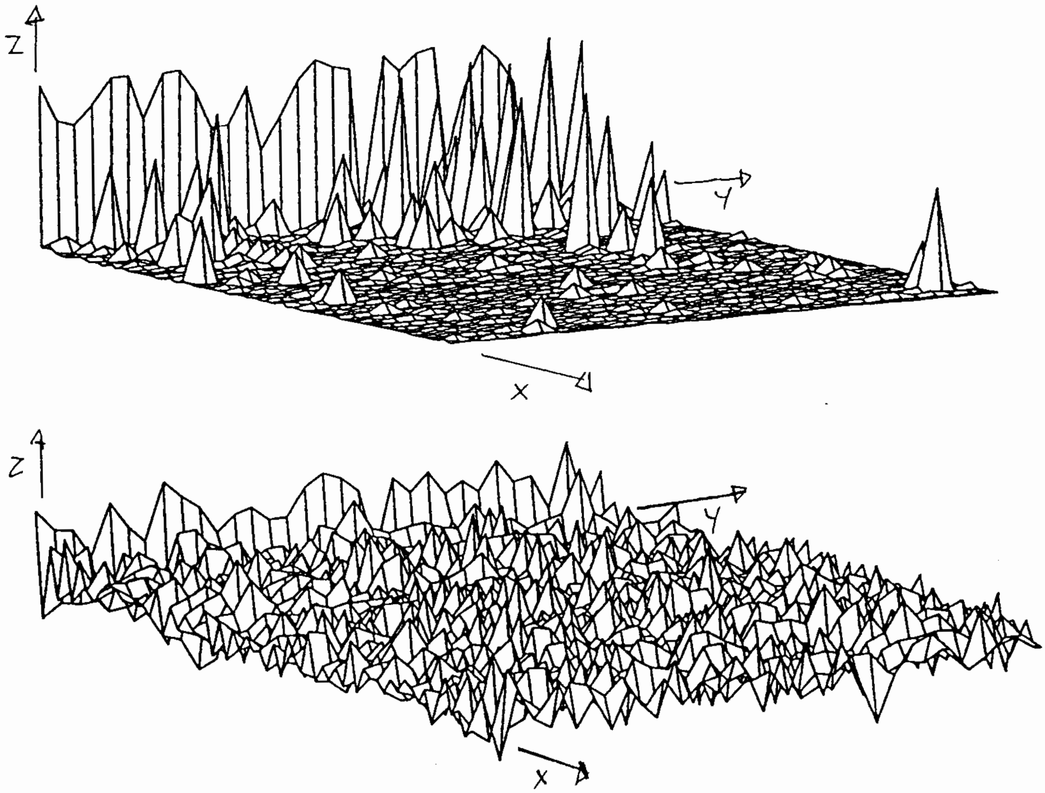

Figure 17. Output values: ANN technology versus cross-correlation technique. Experiment **x96** after 15,000 presentations (top figure), cross-correlation technology (bottom figure). The x-axis corresponds to 64 templates, the y-axis corresponds to the 33 responses in the testing set, the z-axis corresponds to the value of the output element.

This observation is confirmed by averaging, for each one of the 64 output elements, the 33 values that are associated with the responses in the testing set. A plot of the (64) average values associated with each one of the output elements is represented in Figure 18. The term "average output values" designates this type of results.

As one can see, the average output values of experiment **x96** is closer to the desired output than the average output values of the cross correlation technique[9].

[9] In Figure 18, the original results obtained with cross-correlation technique are rescaled in order to facilitate the comparison.

This observation is promising. The ANN technology, however, does not perform significantly better than the cross-correlation technique (as indicated by the classification performance reported in Figure 16). As a matter of fact, the ANNs sometimes generate undesirable high values (the "pics" that may be observed in Figure 17) that will cause the corresponding response to be misclassified (the maximum method is used at the decision stage). We shall describe, in section 4.5, a how these "pics" materialize during the training process (contrast-enhancement phenomenon).

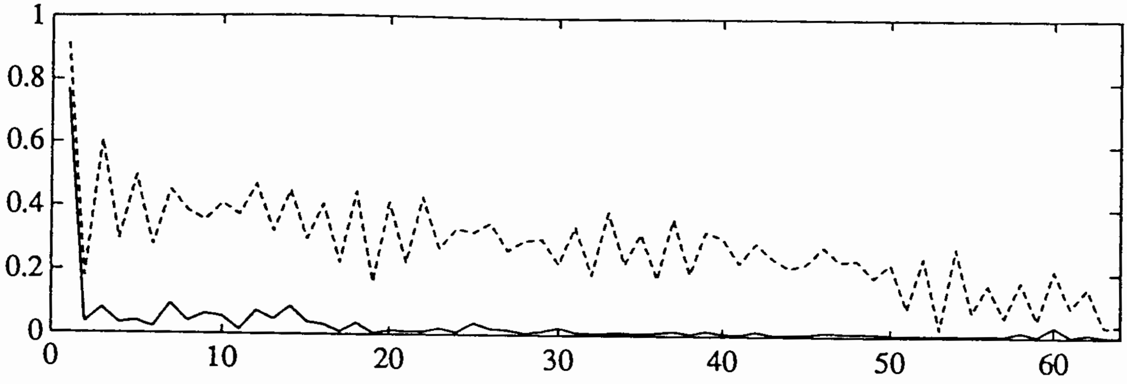

Figure 18. Average output values: ANN technology versus cross-correlation technique. Experiment **x96** after 15,000 presentations (solid line), cross-correlation technique (dashed line).

4.3. Training Process with Short Responses

In the present research, some ANN experiments were designed to evaluate the impacts of particular ANN design choices (e.g. error processing, hidden layer size). In this case, the ANNs were trained with entire brain-wave response (1016 samples).

Other ANN experiments were designed to produce good classification results as soon as possible (short response time). In those experiments, we trained the

ANNs with short brain-wave responses since the time necessary to identify (correctly) the brain-wave response being detected obviously depends on the length of the responses. Note that the ANNs trained with shorter responses nevertheless had 1016 PEs in the input layer. In the present research, the shorter responses were simulated by setting to zero the values of the PEs (in the input layer) exceeding the size of the response[10].

In the present research, the ANNs trained with shorter brain-wave responses can be divided in two sets.

One set of experiments involved training with responses of only one size (e.g. responses 888 samples long for experiment **x60**). In a sense, these experiments are intended to provide information on how well specially dedicated ANNs can perform on short responses. The classification results after 15,000 presentations are reported in Table VII (the ANNs are trained and tested with responses of the same length).

TABLE VII

TRAINING PROCESS WITH PARTIAL RESPONSES

experiments	response length	performance	
		training set	testing set
x90, x91, x92	1016	98.5%	87.9%
x60, x61	888	97.0%	86.4%
x64, x65	760	95.5%	77.3%
x66, x67	632	89.4%	53.0%
x68, x69	504	78.8%	47.0%

[10] For that purpose we used the "minmax table" facility of the ANN simulator NWORKS.

The other set of experiments was trained with a mix of responses of different lengths (the corresponding training processes are described in Appendix B). The results associated with these experiments are reported in Figure 19.

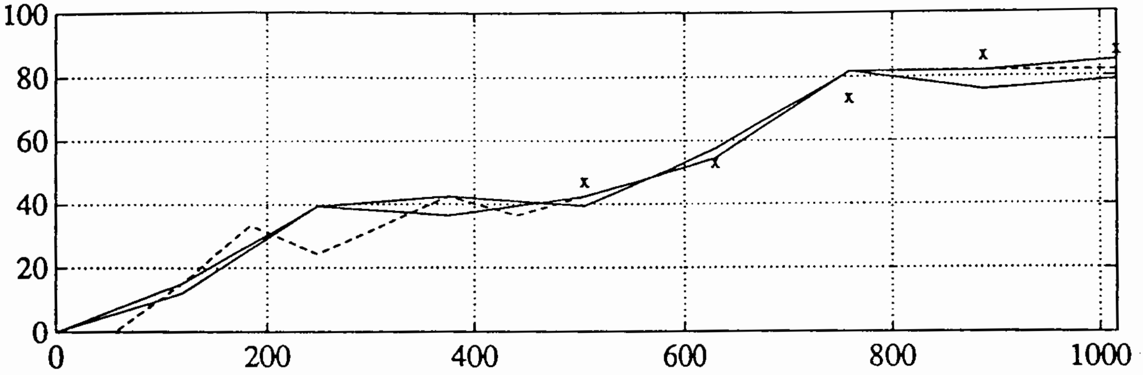

Figure 19. Percentage of correct classifications by ANN experiments trained on a mix of responses of different lengths as a function of the length of the response. ANN technology (solid line), cross-correlation technique (dashed line). The crosses correspond to the classification performance of the experiments reported in Table VII.

There are only minor differences between the results corresponding to the experiments trained solely on entire responses (Figure 16) and those corresponding to experiments that included shorter responses in their training process (Figure 19).

4.4. Generalization And Overfitting Problems

Comparing the results for the testing and training set are useful in determining if the ANNs are overtrained (overfitting problem). Figure 20 and 21 show the percentage of responses in the training and testing set that are classified correctly for experiments **x96** and **X92**. Experiment **x96** is representative of the experiments that were trained with entire brain-wave responses. Experiment **X92** is representative of the experiment that were trained with a mix of responses of different lengths.

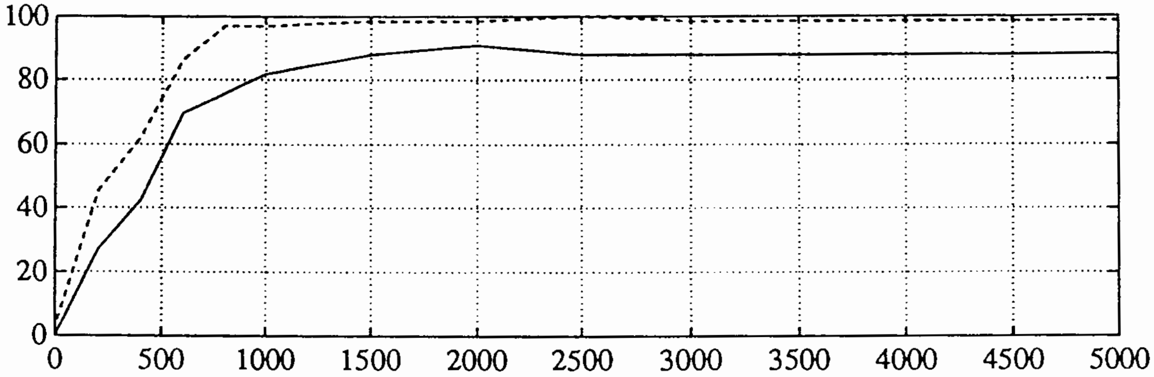

Figure 20. Percentage of correct classification as a function of the length of the training process (experiment **x96**). Entire brain-wave responses were used during the training process. Testing set (solid line), training set (dashed line).

In experiment **x96**, the percentage of correct classifications after 15,000 and 25,000 presentations remained stable (87.9% on the testing set, 98.5% on the training set).

Note that, in Figure 20, the classification performance on the testing set continue to improve between 800 and 2,000 presentations even though the classification performance associated with the training set remain at the same level. We address this issue again in the next section (in fact, we will show that, in a sense, the ANNs are still learning long after the classification performance associated with the training and/or the testing set have stopped to improve).

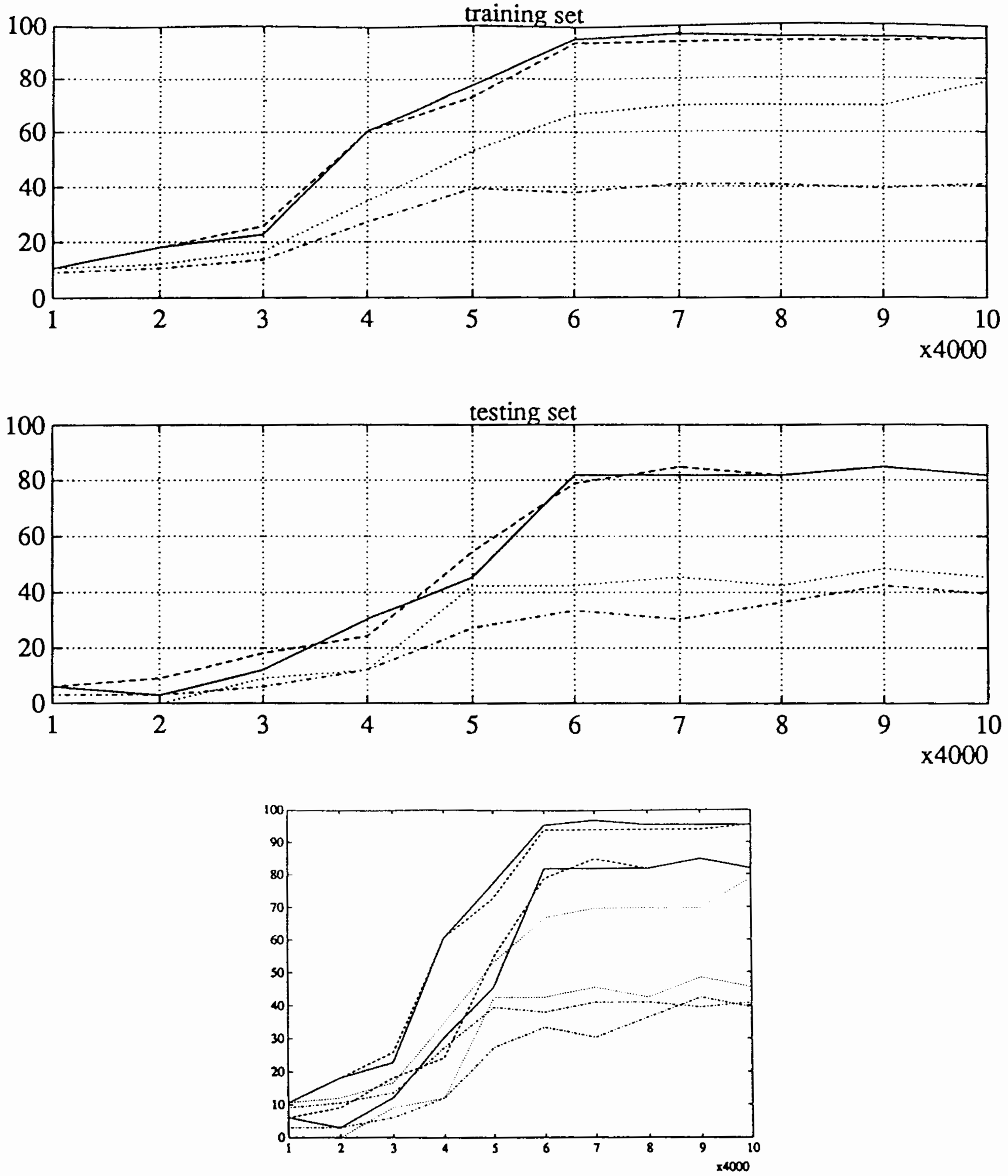

Figure 21 Percentage of correct classification as a function of the length of the training process (experiment X92). A mix of responses of various length was used during the training process. Performance on the training set (top figure), performance on the testing set (center figure). The bottom figure is a superposition of the two previous figures. The results are plotted for responses of four different lengths: 1016 samples (solid line), 760 samples (dashed line), 504 samples (dotted line), 248 samples (dot-dashed line).

The classification performance reported in Figures 20 and 21 do not exhibit any obvious sign of overfitting: the percentage of correctly classified responses in the training and testing set is close. In one particular case (experiment **X92**, 504 samples long responses), there is a significant difference between the classification performance obtained on the training and testing set. This observation, however, is limited to responses of this particular length, there is no such difference when the same ANN is tested with responses of different lengths. Thus, one cannot, on the basis of these results, conclude that experiment **X92** is overtrained.

4.5. Contrast-Enhancement

Figure 22 shows the evolution of the average output values at different stages of the training process.

As one can see, Figure 22 shows that the average outputs values continue to get closer to the ideal case during the training process during as many as 15,000 presentations. This observation apparently contradicts the results reported in Figure 20 showing that the classification performance of the ANNs reaches its maximum very quickly (90% after 2,000 presentations).

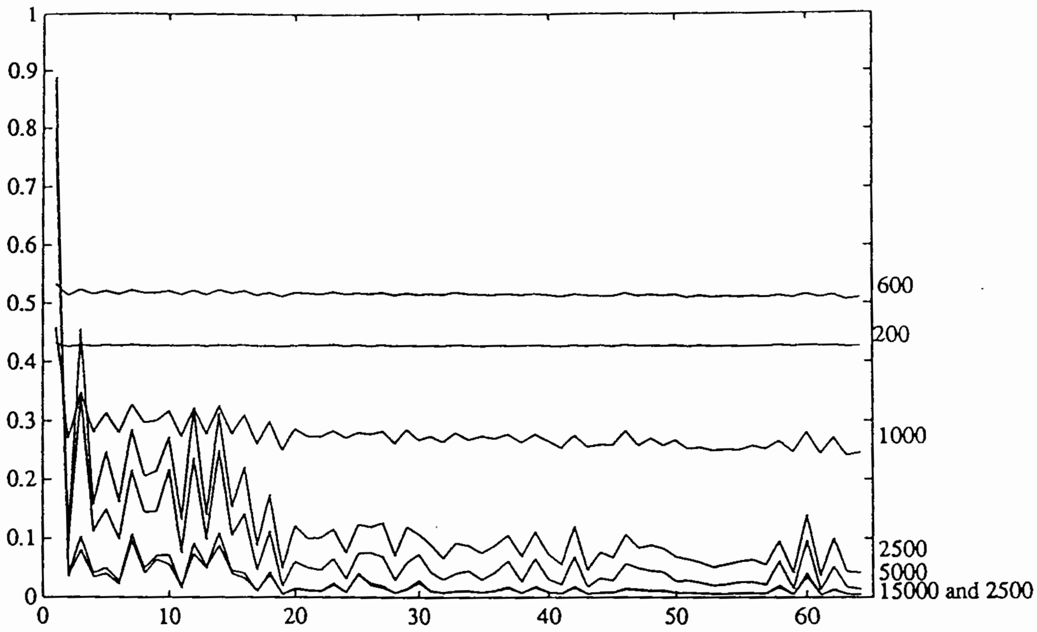

Figure 22. Evolution of the average output values at different stages of the training process (experiment **x96**). Each line correspond to the average output values at a different stage of the training process, the corresponding number of presentation is inscribed in the right margin.

We believe that this contradiction is essentially due to a phenomenon, here called contrast-enhancement, that characterizes the training process of all the experiments in this research. In order to observe this phenomenon, it is necessary to look at the values associated with the 64 output elements for each response in the testing set at different stages of the training process (Figure 23).

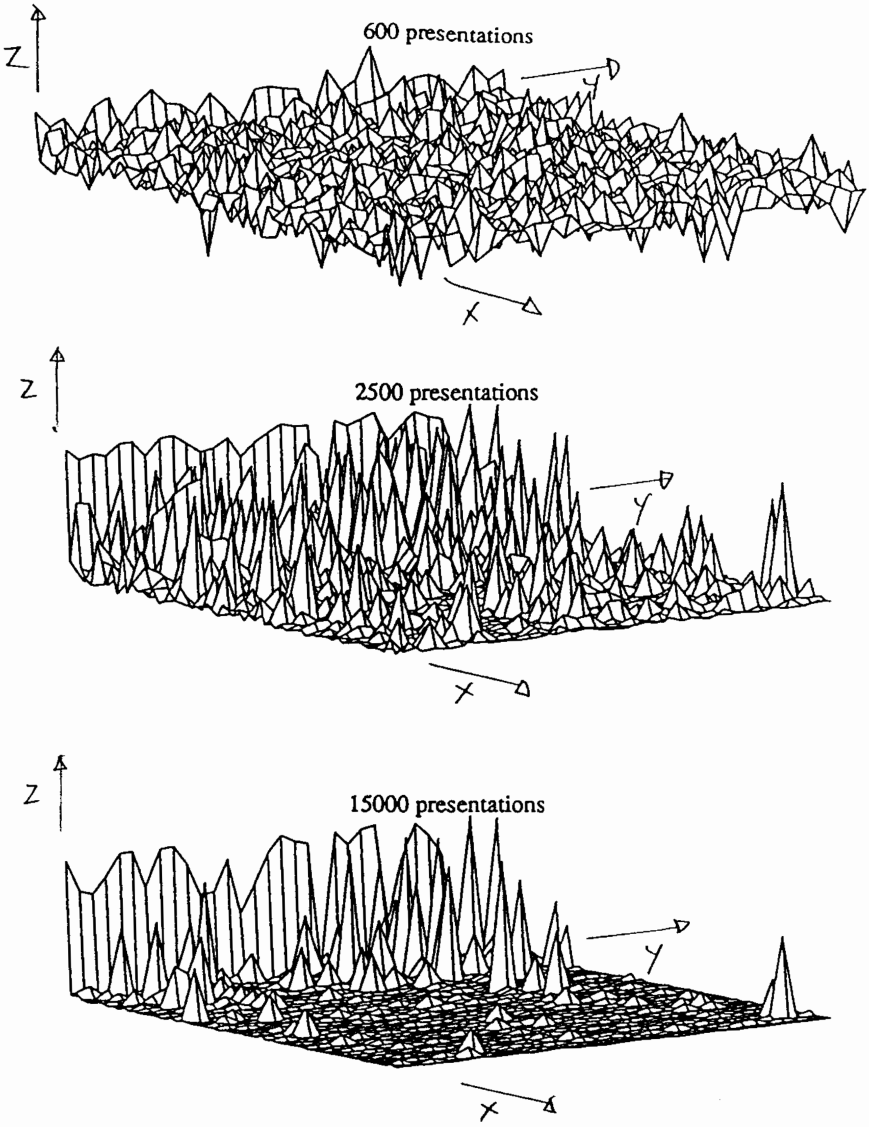

Figure 23. Output values at different stages of the training process. Experiment **x96** after 600 (top figure), 2500 (center figure), and 15,000 presentations (bottom figure). The x-axis corresponds to 64 output elements, the y-axis corresponds to the 33 responses in the testing set, and the z-axis indicates the value associated with the output elements.

Figures 23 suggests that the output values that are already close to the desired outputs stay close (or get closer) at the expense of other values that get worse. In other words, the contrast among the output values is enhanced because intermediary values (the "gray" values) disappear during the training process. At the end of the training process, only the values close to 0 (the "white" values) or close to 1 (the "black" values) remain. Thus, the improvement in the average output value during the training process is due to the fact that most of the small deviations from the desired output values (the "gray" value) disappear. However, the classification performance does not change since the misclassifications are mainly due to the "black" or "white" values.

A number of our experiments were designed with the objective of lowering the error associated with the "black" and "white" values as well as the error associated with the "gray" values. To achieve this objective, we experimented with changing several parameters of the training process, but did not succeed. For example, we increased the size of the epoch from 16 to 132 (experiment **X5**); we reduced the output range (experiment **X6**); and so on.

Our conclusion is that increasing the length of the training process will not improve performance of these experiments. We believe that the contrast-enhancement phenomenon described in this section is a problem that may have appeared in other problem contexts as well. It would be interesting to see if there is a general solution to this phenomenon. We believe that, in the present problem context, it will most likely be necessary to increase the size of the training set to obtain better results.

Figure 24 shows the evolution of the output values associated with two particular brain-wave responses in the testing set that exemplifies particularly well the contrast-enhancement phenomenon.

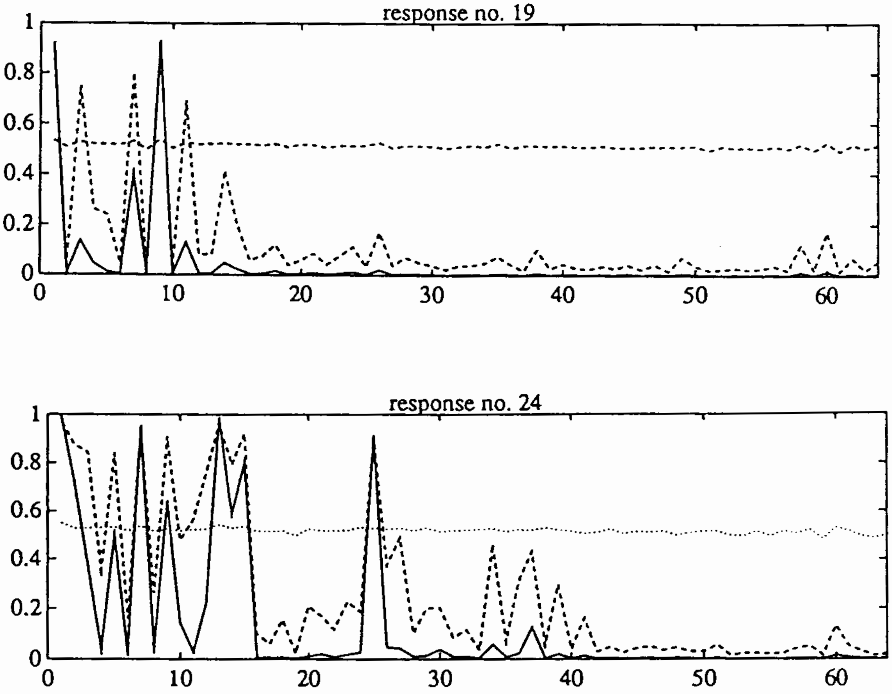

Figure 24. Output values associated with two specific brain-wave responses at different stages of the training process (experiment **x96**). The top figure corresponds to the 19th response in the testing set, the bottom figure corresponds to the 24th response in the testing set. 600 presentation (dotted line), 2,000 presentations (dashed line), 15,000 presentations (solid line). The x-axis corresponds to 33 responses in the testing set, the y-axis indicates the actual value of the output element.

4.6. Nearby Misclassification

The term ‘nearby misclassification’ refers to the fact that a misclassification is more likely to occur among the "early" classes, e.g. between classes 2 and 10 (see

Figure 25). This observation was repeated many times during this research[11]. Note that the impact of nearby misclassification distribution can be observed in the average output values (Figure 24). Because most misclassifications happen when the value of the second and following output elements is significantly bigger than 0, and because most misclassifications occur in the early classes, it is therefore normal for the average output values reported in Figure 24 to be bigger in the early classes than in the later ones.

Note also that if the responses in the testing set belonged to another class (all responses in the testing actually belong to class 1), then the correct classifications as well as the misclassifications would be accordingly shifted (see section 1.3).

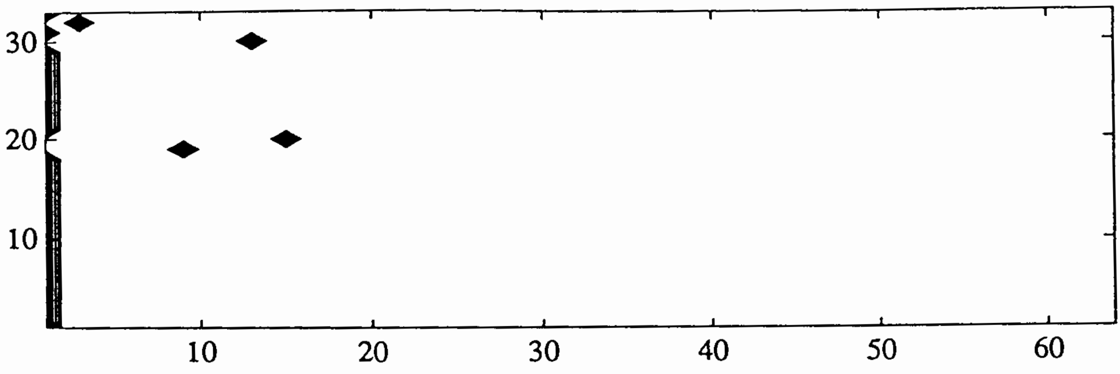

Figure 25. Classification of the responses in the testing set in experiment **x96**. The maximum method is used at the decision stage.

[11] Figure 25 shows also something else. The comparison between the classification of the responses by the ANN technology (Figure 25) and by the cross-correlation technique (Figure 15) shows that the misclassifications, albeit different, are caused by the same responses. This suggests that the misclassifications are really due to attributes of the responses rather than to some peculiar characteristic of the methods used. We cannot, however, discard these responses because we do not have enough information about the characteristics of typical responses.

Observing that the visual stimuli inducing the brain-wave responses belonging to these classes are "closer in time" to the visual stimulus corresponding to class 1 (i.e. there is smaller lag between $x_1(t)$ and $x_2(t)$ than between $x_1(t)$ and $x_3(t)$, etc.) can certainly be part of an explanation of the nearby misclassification distribution phenomenon. But this reason is insufficient because the misclassifications do not follow a symmetric pattern, there are fewer misclassifications between classes 56 and 64 than between classes 2 and 10. Thus, there must be an additional reason to explain why the misclassifications occur essentially between class 2 and class 10.

We designed several experiments that specifically address this issue (the ANNs used in these experiment require a modification of the backpropagation algorithm). The results associated with these experiments are reported in Tables VIII and IX. As one can see, these experiments have similar classification performance.

TABLE VIII

ERROR PROCESSING I

experiment	response's class	error factor	performance (testing set)
x90, x91, x92	1-64	1	87.9%
x10, x11	1-2 3-64	2 1	87.9%
x20, x21, x22	1-10 11-64	2 1	88.8%
x25, x26, x25	1 2-10 11-64	1 2 1	88.8%

TABLE IX

ERROR PROCESSING II

experiments	response's class	error factor	performance for various response sizes (testing set)			
			1016	760	504	248
X1	1 2-7 8-11 12-64	2 4 2 1	87.9%	78.8%	42.4%	33.3%
X2	1 2-11 12-21 22-64	2 4 2 1	84.8%	78.8%	45.5%	27.3%
X3	1 2-11 12-64	2 0.5 1	90.9%	78.8%	45.5%	15.2%

In a typical backpropagation ANN (e.g. experiments **x90**, **x91**, and **x90**), the error computed by the criterion function is simply the difference between the actual and the desired output (i.e. the error e_i associated with the PE x_i is $o_i - d_i$, where o_i is the actual output value, d_i is the corresponding desired output value). In our version of the backpropagation paradigm, the idea is to modify the error evaluated by the criterion function as a function of the class of the response. For example, in experiments **x10** and **x11**, the computation of the error in the output layer is

$$e_i = \begin{cases} 2(o_i - d_i) & \textit{if the input belongs to class } 1, \textit{ or class } 2 \\ o_i - d_i & \textit{otherwise} \end{cases} \tag{21}$$

In equation 21, the coefficient that eventually multiplies the error is called the error factor. A detailed description of the implementation of this modified version of the backpropagation paradigm is given in Appendix A, and a detailed description of these experiment is given in Appendix B.

As previously mentioned, the results of these experiments are not conclusive. Nevertheless, we believe that this approach has potential. A more simple problem context would certainly be desirable to understand it better both empirically, and theoretically.

4.7. Hidden Layer Size

The size of the hidden layer is one of the most important characteristics of an ANN architecture. The results associated with several experiments addressing this issue are reported in Table X.

TABLE X

SIZE OF THE HIDDEN LAYER

experiments	size (PEs)	performance	
		training set	testing set
x84, x85	10	98.5%	90.0%
x90, x91, x92	20	98.5%	87.9%
x81, x82	30	98.5%	89.4%
x87, x88	50	98.5%	87.9%

As previously mentioned, these results are not conclusive. On the positive side, it was noted that the experiments with fewer PEs in the hidden layer required a shorter training process (the differences were on the order of a few thousand presentations).

CHAPTER VI

CONCLUSION

In the present thesis, we tried to improve the speed of classification in an existing gaze-addressing communication system, called the Brain Response Interface, that was developed at the Smith-Kettelwell Institute at San-Francisco by Dr. Erich Sutter. In particular we explored the potential of a new approach based on ANN technology to replace the cross-correlation technique currently used in the BRI.

1. DISCUSSION OF THE RESULTS

1.1. Artificial Neural Network Technology

The initial objective was only partially achieved. Even though we definitely established that the ANN technology can perform at least as well as the cross-correlation technique, none of our experiments performed substantially better than the cross-correlation technique.

One must notice however that this research was conducted under a severe handicap of having only 99 brain-wave responses. Sixty six of these were used as the training set, and the remaining 33 as the testing set. This is insufficient for several reasons.

First, brain-wave response are signals that are extremely difficult to categorize, and this set is not fully representative of the problem context. The characteristics of the brain-wave response depends strongly on the individual, and on his or her general medical status. In addition, temporary bursts of noise can transform them significantly. It is felt that substantially more data will be necessary to insure that results obtained reflect adequately the performance potential of the ANN technology.

Second, it is a well known fact that the size and quality of the training set is crucial in achieving good performance with ANN technology. Any future research should include more responses. This, however, may cause several practical problems (e.g. the training file in our experiments exceeded 35MBytes[12]). Thus, it may be necessary to change the sampling rate, and/or the representation of the responses. In any case, these experiments will require expertise in neuro-physiology as well as in ANN technology.

Despite this handicap, the results reported in this thesis brought a better understanding of the problem context, yielded several interesting observations, and helped to define lines of future research.

First, we found a fairly robust way to use ANNs in this problem context. Our experiments show that relatively minor changes in the architecture of the ANNs did not impact the performance. This same observation, however, prevented us from precisely relating a specific set of parameters of the ANN to the characteristics of the

[12] The size of the training file could be significantly reduced by using the user I/O facility if NWORKS. This would require writing an I/O program that automatically shift the responses on the fly when they are presented to the ANN (see section 1.2, Chapter V).

present problem context. The design of future experiments could certainly be facilitated if more conclusive results had been obtained.

Second, we identified two phenomena that, if properly addressed, could lead to a significant improvement of the classification performance. The first phenomenon, called contrast-enhancement, relates to the fact that only the small deviations from the desired output values disappear during the training process. Therefore, the percentage of classification remains stable even though the ANN continues to learn (the average output values get closer to the desired output values). What is necessary is a method that positively impacts the large deviations from the desired output values.

The second phenomenon, called nearby distribution, relates to the observed situation that a misclassification is more likely to happen in a class close to the actual (correct) class of the input. Any method providing better discrimination among nearby classes would solve this problem. In this thesis, we tried to modify the backpropagation paradigm so that the criterion function depends on the class of the brain-wave response, but no improvements were discovered.

We believe that the ANN technology has the flexibility to address both problems. We also believe that these phenomena are worth being studied for themselves because they may happen in different problem contexts. Thus, a solution to these problems could have a larger applicability than the present problem context.

Finally, we should mention a point that may give an edge to using this technology in the present problem context.

In this thesis, we assumed that the visual system was stationary[13]. But, factors exterior to the visual system like fatigue, stress, or environmental changes, affect the visual system in the long run. This problem is handled differently according to the technology used. In the case of the cross-correlation technique, it possible to use averaging techniques to adjust templates. In the case of the ANN technology, it is possible to automatically adjust the weights by continuing the training process while it is performing its classification task. The use of ANN technology in nonstationnary problem contexts is increasingly popular as indicated by the amount of recently published literature about adaptive filtering.

Unfortunately, the data available for this research do not afford the possibility of addressing this issue.

1.2. The Cross-Correlation Technique

Although this was not part of our initial objective, we found that normalizing the brain-wave response before computing its correlation to the templates, has a positive and significant impact on the performance. This observation is of immediate utility since, to our understanding, the current implementation of the BRI does not normalize the brain-wave response. In Chapter III, we showed that normalization of the brain-wave response may be realized at a low computational cost.

[13] The input/output relationship of a stationary system doesn't change with time. Stationary systems are also called time-invariant systems.

REFERENCES CITED

Aunon JI., Mc Gillem CD., Childers DG. (1981) "Signal Processing in Evoked Potential Research: Averaging and Modeling", *CRC Critical Reviews in Bioengineering*, 5, 4, 323

Carroll TO., Ved H. (1989) "A Neural Network for ECG Analysis" (Abstract), *International Joint Conference on Neural Network*, Washington, 2, 575

Hassoun MH., Spitzer AR. (1988) "Neural Network Identification and Extraction of Repetitive Superimposed Pulses in Noisy 1-D Signals" (Abstract), *NN Supplement: International Neural Network Society (INNS) Abstracts*, 1, 443

Hecht-Nielsen R. (1988) "Applications of counterpropagation neural networks", *Neural Network*, 1, 131-140

Holdaway RM., White MW., Marmarou A. (1990) "Classification of Somatosensory-Evoked Potentials Recorded from Patients with Severe Head Injuries", *IEEE Engineering Medecine and Biology Magazine*, 9, 3, 43

Iyer VK., Ploysongsang Y., Ramamoorthy PA. (1990) "Adaptive Filtering in Biological Signal Processing", *Critical Reviews in Biomedical Engineering*, 17, 6, 531

Kelly MF., Parker PA., Scott RN. (1990) "The Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study", *IEEE Transactions on Biomedical Engineering*, 37, 3, 221

Marmarelis PZ., Marmarelis VZ (1978) *Analysis of Physiological Systems: The White Noise Approach*, Plenum Press, New-York

Strand EM., Jones WT. (1990) "A Neural Network for tracking the prevailing heart rate of the ECG", *Proceedings Third Annual IEEE Symposium on Computer-Based Medical System*, 358

Sutter E.E. (1992) "The Brain Response Interface: Communication through Visually Induced Electrical Brain Responses", personal communication

Sutter E.E., Tran D. (1992a) "Communication through Visually Induced Electrical Brain Responses", personal communication

Sutter E.E., Tran D. (1992b) "The Field Topography of ERG Components in Man: 1. The Photopic Luminance Response", personal communication

Sutter E.E. Pevehouse B., Barbaro N. (1992) "Intra-Cranial Electrodes for Communication and Environmental Control", personal communication

Sutter E.E. (1991) "A Deterministic Approach to Non-Linear Systems", in *Nonlinear Vision*, ed. Robert B. Pinter and Bahram Nabet, CRC Press

Sutter E.E. (1987) "A practical non-stochastic approach to nonlinear time-domain systems", *Proceedings Biomedical Simulations Resource*, University of Southern California: Advanced Methods of Physiological System Modelling, 1, 303

Tam DC., Perkel DH. (1989) "A Model for Temporal Correlation of Biological Neuronal Spike Trains", *International Joint Conference on Neural Network*, Washington, 1, 781

Thakor NV. (1991) "Applications of Adaptive Filtering to ECG Analysis: Noise Cancellation and Arrhuthmia Detection", *IEEE Transactions on Biomedical Engineering*, 38, 8, 785

Uncini A, Marchesi M., Orlandi G., Piazza F. (1990) "Improved Evoked Potential Estimation Using Neural Network", *International Joint Conference on Neural Network*, San-Diego, 2, 143

White H. (1988) "Some asymptotic results for learning in single hidden layer feedforward network model", University of California at San-Diego, Departement of Economics, Discussion Paper

Widrow B., Winter R. (1990) "Neural Networks for Adaptive Filtering and Adaptive Pattern Recognition" in *An Introduction to Neural and Electronics Networks*, ed. by Zornetzer SF., Davis JL., Clifford L., Accademic Press, 249

Wong Y., Banik J., Bower JM. (1988) "Neural Network for Template Matching: Application to Real-Time Classification of the Action of Real Neurons" in *Neural Information Processing System*, ed. by Anderson DZ., American Institute of Physics, 103

NOTE: Appendix E contains an extended bibliography.

APPENDIX A

ERROR PROCESSING

DESCRIPTION

The ANN presented in this appendix is a modification of the backpropagation paradigm. It suggests a new way of using knowledge of the problem context (expert knowledge) in ANN technology.

The first step is to attach a class to each datum in the training set. In principle, the class could be arbitrarily defined. In practice however, it makes sense to base the definition of the data on the properties of the data, or on other properties of the problem context. A practical way to define a class is, for example, to add some bits to each datum in the training set.

The second step is to make a component of the ANN (say, the criterion function) dependent on the datum's class k

$$e = f(d,a,k)$$

where e is the error returned by the criterion function f, d is the desired output, and a is the actual output. For example, in the present thesis, the difference between the desired output and actual output, d-a, is multiplied by a coefficient, called the error factor, that depends on the datum's class k

$$e = c_k(d-a)$$

where c_k is the error factor associated with (input) datum's class k, and e the error that will be backpropagated.

The hope here is that the ANN will learn to discriminate the data whose class is associated with the largest error factor c_k (we did not provide a theoretical basis for this "hope").

NWORKS IMPLEMENTATION

The data format in a **nna** file is (one datum per line)

[input][class][desired_output]

where *[class]* is a sequence of bits $c_1\ c_2 \ldots\ c_n$. In the present thesis, the first class was encoded by the sequence *100...00*, the second class was encoded by the sequence *010...00*, and so on. We note that, in this case, the class and the desired output are similar. This is not necessary. One could imagine two inputs having the same desired output, but belonging to two different classes.

The typical backpropagation architecture (i.e multilayered feedforward ANN) must be modified: four layers are added, and a special control stategy **csbkp4** must be used (see Figure A.1).

The **input** layer, as usual, is a buffer. **Net1** and **net2** layers correspond to a 2-layer backpropagation network (as many layers as desired can be defined). **Buffer**, **error1**, **error2**, and **output** layers have special functions, and they (as well as the connection schemes) must be defined according to what the control stategy **csbkp4** expects.

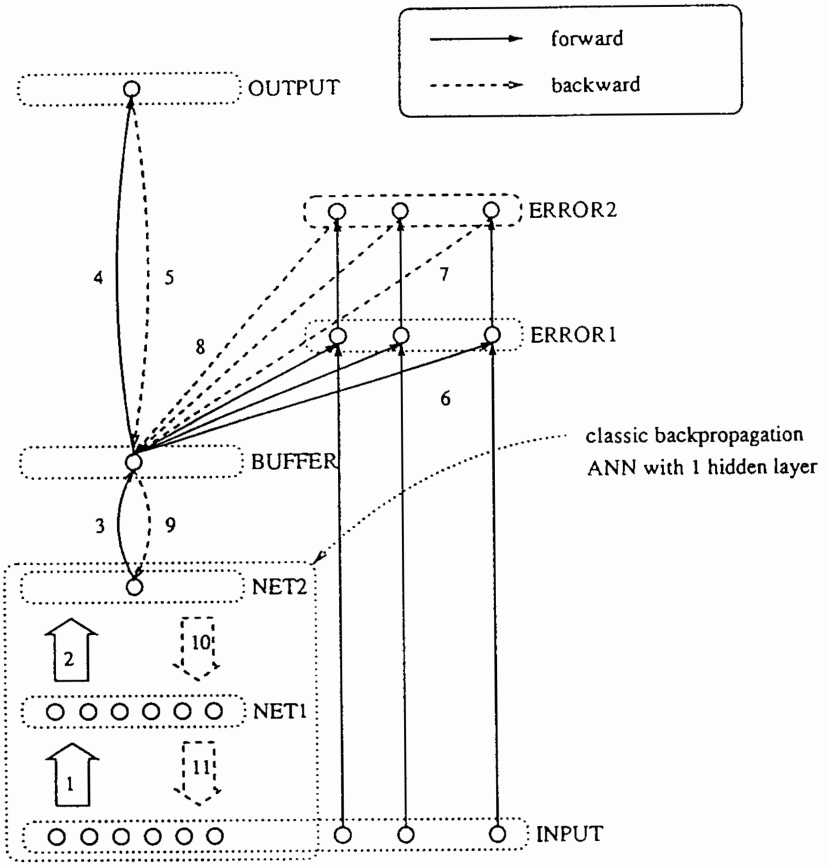

Figure A.1

The control strategy **csbkp4** works as follows: the input and the class are loaded in the **input** layer; **csbkp4** goes through a classic recall process through **net1** and **net2** layer (steps 1 and 2); the output of **net2** is transmitted to the **output** layer through the **buffer** layer (steps 3 and 4); and the difference between the actual output and the desired output is determined in the **output** layer and backpropagated to the **buffer** layer (step 5).

From that point, what happens during the training process is unusual: the error stored in the **buffer** layer is transmitted to **error1** layer and to **error2** layer (step 6 and 7); and, the error is backpropagated again to the **buffer** layer (step 8).

The **error1** layer is a particular layer whose "sum" function returns the product between all incoming inputs. **Error1** layer is very important since the bits defining the class of the input will determine wich processing elements in **error1** layer will contain the error. This makes possible *to differentiate the processing of the error according to the class of the input datum* (the pathways for the error are different according to the datum's class). The error can be modified, for example, by the error factor of each processing element in the **error2** layer, by the transfer function of **errorl** layer, by the weights between **buffer** and **errorl**, by the weights between **errorl** and **error2**, and/or by the weights between **error2** and **buffer**. In each case, the modification of the error depends on the class of the datum.

Once the error is back in the **buffer** layer, it is simply backpropagted to the **input** layer (steps 9, 10, and 11), and the weights are updated as in a classic backpropagation training process.

Of course, more error layers could be added between **errorl** and **error2** layers to permit a more sophisticated processing of the error. In addition, one could imagine an ANN where the error processing would itself be subject to some training. The error layers could constitute an independent backpropagation ANN with zero as desired output.

DISCUSSION

The ANN paradigm described in this appendix and changing the representation of the data, are two different methods of using expert knowledge in ANN technology. The challenge, in both cases, is to clearly understand the impact of these modifications on the performance of the ANN, on the duration of the training process, and so on.

Modeling the impact of changing the representation of the data is difficult. As a matter of fact, despite many studies, the representation problem is not yet solved. Changing the data's representation has only an indirect (implicit) effect on the ANN's dynamic. Its impact is difficult to model (the information flow in an ANN is distributed over many processing elements through complex connection patterns).

In contrast, the modification of the ANN backpropagation algorithm that we propose, is explicitely stated. Thus, its impact could be studied independently in specially designed problem contexts. Then, this knowledge could be applied in real-world applications. One could imagine a catalog of possible modifications depending on the characterisitcs of the problem context (e.g. the properties of the data).

APPENDIX B

ANN EXPERIMENTS

FIRST ANN ARCHITECTURE TYPE EXPERIMENTS

X4	
Layer sizes	1016, 30, 64
Transfer function	sigmoid
Learning rule	delta
Learning schedules	default, hidx4, outx4
Epoch size	16
Input range	0, 1
Output range	0.2, 0.8
X7	
Layer sizes	1016, 30, 64
Transfer function	tanh
Learning rule	cum-delta
Learning schedules	default, ncmhid, ncmout
Epoch size	66
Input range	-1, 1
Output range	-0.6, 0.6
X8	
Layer sizes	1016, 30, 64
Transfer function	tanh
Learning rule	cum-delta
Learning schedules	default, ncmhid2, ncmout2
Epoch size	66
Input range	-1, 1
Output range	-0.6, 0.6

EXPERIMENT **X90, X91, X92**

x90, x91, x92	
Layers sizes	1016, 20, 1 (see Hidden Layer Size Table)
Transfer function	sigmoid
Learning rule	delta
Learning schedules	default, hidden1, out3
Epoch size	16
Input range	0, 1
Output range	0, 1
Error factor	1-64:1 (see Error Factor I Table)

The characteristics of the experiments described in the two next tables are similar to those of experiments **X90**, **X91**, and **X90** (except for the differences that are explicitely stated).

SIZE OF THE HIDDEN LAYER

experiments	size (PEs)
x84, x85	10
x90, x91, x92	20
x81, x82	30
x87, x88	50

ERROR FACTOR I

experiments	response's class	error factor
x90, x91, x92	1-64	1
x10, x11	1-2 3-64	2 1
x20, x21, x22	1-10 11-64	2 1
x25, x26, x25	1 2-10 11-64	1 2 1

EXPERIMENTS X10, X20, X30

X10, X20, X30	
Layer sizes	1016, 30, 1 (45 PEs in the hidden layer of X2)
Transfer function	sigmoid
Learning rule	delta
Learning schedule	default, hidden, output
Epoch size	16
Input range	0.2, 0.8
Output range	0, 1
Error factor	see Table Error Factor II

ERROR FACTOR II

experiments	response's class	error factor
X1	1	2
	2-7	4
	8-11	2
	12-64	1
X2	1	2
	2-11	4
	12-21	2
	22-64	1
X3	1	2
	2-11	0.5
	12-64	1

MISCALLENEOUS EXPERIMENTS

X50	
Layer sizes	1016, 30, 1
Transfer function	sigmoid
Learning rule	norm-cum-delta
Learning schedule	default, hidden, output
Epoch size	132
Input range	0, 1
Output range	0, 1
Error factor	1;2, 2-11:4, 12-21:2, 22-64:1
X60	
Layer sizes	1016, 30, 1
Transfer function	sigmoid
Learning rule	norm-cum-delta
Learning schedule	default, hidden, output
Epoch size	16
Input range	0, 1
Output range	0.3, 0.7
Error factor	1;2, 2-11:4, 12-21:2, 22-64:1

TRAINING PROCESS WITH PARTIAL INPUTS I

experiments	input size	number of presentation
X83	1016	10,000
	760	10,000
	1016	5,000
	504	10,000
	1016	5,000
	248	10,000
	1016	5,000
	504	5,000
	1016	2,500
	760	5,000
	1016	2,500 (70,000)
	760	10,000
	120	10,000
	248	10,000
	504	10,000
	760	10,000
	376	10,000
	632	10,000
	888	10,000
	120	10,000
	760	10,000
	1016	10,000
	632	10,000
	376	10,000
	888	10,000 (210,000)

The classification performance of experiment **X83** was evaluated after 70,000, and 210,000 presentations.

TRAINING PROCESS WITH PARTIAL INPUTS II

experiments	input size	number of presentation
x96	120	1000
	248	1000
	376	1000
	504	1000
	632	1000
	760	1000
	888	1000
	1016	1000 (8,000)
	The previous training was repreated 4 more times (32,000 presentations).	
	120	2500
	248	2500
	376	2500
	504	2500
	632	2500
	760	2500
	888	2500
	1016	2500 (60,000)
	120	5000
	248	5000
	376	5000
	504	5000
	632	5000
	760	5000
	888	5000
	1016	5000 (100,000)

APPENDIX C

LEARNING/RECALL SCHEDULES

L/R SCHEDULE I

Schedules	Parameters	2'000	5'000	10'000	17'500	30'000
hidden	learning	.225	.125	.05	.02	.01
	momentum	.4	.2	.1	.05	.05
output	learning	.15	.1	.07	.05	.04
	momentum	.4	.2	.1	.05	.05

L/R SCHEDULE II

Schedules	Parameters	2,000	5,000	15,000	30,000	75,000
hidden1	learning	.2	.1	.05	.01	.002
	momentum	.2	.1	.05	.01	.001
		5,000	10,000	20,000	50,000	100,000
out3	learning	.15	.075	.01875	.0017	0
	momentum	.4	.2	.05	.00313	0

L/R SCHEDULE III

Schedules	Parameters	5000	15,000	30,000	75,000	100,000
ncmhid	learning	.6	.2	.1	.05	.03
	momentum	0	0	0	0	0
ncmout	learning	.5	.2	.1	.02	.03
	momentum	0	0	0	0	0

L/R SCHEDULE IV

Schedules	Parameters	25,000	50,000	10,000 (sic)	200,000	400,000
ncmhid2	learning	.9	.7	.45	.175	.275
	momentum	.2	.2	.2	.2	.2
ncmout2	learning	.8	.6	.45	.325	.25
	momentum	.2	.2	.2	.2	.2

L/R SCHEDULE V

Schedules	Parameters	5,000	10,000	20,000	50,000	100,000
hidx4	learning	.4	.2	.1	.05	.02
	momentum	.4	.2	.1	.05	.02
outx4	learning	.15	.1	.07	.05	.02
	momentum	.4	.2	.1	.05	.02

APPENDIX D

RECORDS

The names of directories and files are indicated in bold typeface. All records are recorded on two tapes (2x150 Mbyte).

The data (brain-wave responses) are located in several directories: **data0**, **data2**, **data3**, **data4**, **class0**, **class1**, **class2**. The content of the directories is described in **README** files.

The C-programs used in this thesis are available in the directory **src**. This directory contains the programs used to convert the data from a binary to ASCII the format, and to the format required by ANN simulator NWORKS (**.nna** files), to construct the training sets required by the ANN methodology. The programs, **nmax**, **nconf** and **nerr**, were used to analyzes the test files (**.nnr** files). **nconf** and **nerr** can be used in different problem contexts. A detailed listing of the results obtained in this research is contained in the file **res/nmax.res**.

The ANN experiments are divided into four directories: **exp1**, **exp2**, **exp3**, **exp3**, **exp4**. **exp1** and **exp2** contain preliminary experiments. **exp3** contains the experiments whose initial letter is a lowercase 'x' (e.g. experiment **x90** corresponds to the file **bkpx90???.nnd**, the last digits usually indicate the stage in the training process, and/or the minmax table used during the testing). **exp4** contain the experiments whose intial letter is an uppercase 'X'. The directories **nna** and **nnb** contain the training sets that were used to train the ANNs. The directories **nnc**, **nnt**, and **nnd**, contain respectively the minmax tables, learning/recall schedules, and the control strategies used in the ANN experiments.

The directory **cor** contains computations and results associated with the cross-correlation technique.

APPENDIX E

EXTENDED BIBLIOGRAPHY

The following references are grouped by subjects (this appendix contain references that have not been mentioned in the text of this thesis).

THE BRAIN-RESPONSE INTERFACE

Sutter E.E. (1992) "The Brain Response Interface: Communication trough Visually Induced Electrical Brain Responses", personal communication

Sutter E.E., Tran D. (1992a) "Communication through Visually Induced Electrical Brain Responses", personal communication

Sutter E.E., Tran D. (1992b) "The Field Topography of ERG Components in Man: 1. The Photopic Luminance Response", personal communication

Sutter E.E. Pevehouse B., Barbaro N. (1992) "Intra-Cranial Electrodes for Communication and Environmental Control", personal communication

Sutter E.E. (1991) "A Deterministic Approach to Non-Linear Systems", in *Nonlinear Vision*, ed. Robert B. Pinter and Bahram Nabet, CRC Press

Sutter E.E. (1987) "A practical non-stochastic approach to nonlinear time-domain systems", *Proceedings Biomedical Simulations Resource*, University of Southern California: Advanced Methods of Physiological System Modelling, 1, 303

EVOKED POTENTIALS

General References

Celesia GG. (1985) "Visual Evoked Responses" in *Evoked Potential Testing*, ed. Grune & Stratton, 1

De Weerd JPC., Stegeman (1990) "Technichal and Methodological considerations on the measurement of evoked potentials" in *Evoked Potential Manual*, ed. by Colon EJ., Visser SL., Kluwer Academic Publishers, 3

Hudnell HK., Boyes WK., Otto DA. (1990) "Stationary Pattern Adaptation and the Early Components in Human Visual Evoked Potentials", *Electroencephalography and Clinical Neurophysiology*, 77, 190-198

Riemslag FCC., Spekreijse H. (1990) "Electrodiagnosis by Luminance and Pattern Stimulation" in *Evoked Potential Manual*, ed. by Colon EJ., Visser SL., Kluwer Academic Publishers, 117

Processing and Modeling

Arpaia JP., Isenhart R., Sandman CA. (1989) "A Characterization of a Single-Trial Adaptive Filter and Its Implementation in the Frequency Domain", *Electroencephalography and Clinical Neurophysiology*, 73, 362-368

Aunon JI., Mc Gillem CD., Childers DG. (1981) "Signal Processing in Evoked Potential Research: Averaging and Modeling", *CRC Critical Reviews in Bioengineering*, 5, 4, 323

Marmarelis PZ., Marmarelis VZ (1978) *Analysis of Physiological Systems: The White Noise Approach*, Plenum Press, New-York

Spekreijse H., Reits D. (1982) "Sequential Analysis of the Visual Evoked Potential System In Man; Nonlinear Analysis of a Sandwich System", *Annals New York Academy of Sciences*, 72

Spreckelsen von M., Bromm B. (1988) "Estimation of Single-Evoked Cerebral Potentials by Means of Parametric Modeling and Kalman Filtering", *IEEE Transactions on Biomedical Engineering*, 35, 9, 691

Yu K-B., Mc Gillem CD. (1983) "Optimum Filters for Estimating Evoked Potential Waveforms", *IEEE Transactions on Biomedical Engineering*, BME-30, 11, 730

ARTIFICIAL NEURAL NETWORKS

General Reference

Hecht-Nielsen R. (1989) *Neurocomputing*, Addison-Wesley Publishing Company

Hecht-Nielsen R. (1988). "Applications of counterpropagation neural networks", *Neural Network*, 1, 131-140

Simpson PK. (1990) *Artificial Neural Systems - Foundations, Paradigms, Applications, and Implementations*, Pergamon Press

Wang J., Malakooti B. (1989) "On Training of Artificial Neural Networks", *International Joint Conference on Neural Network*, Washington, 2, 387

White H. (1988) "Some asymptotic results for learning in single hidden layer feedforward network model", University of California at San-Diego, Departement of Economics, Discussion Paper

Williams RJ. (1991) "Adaptive State Representation and Estimation Using Recurrent Connectionist Networks" in *Neural Networks for Control*, ed. by Miller WT., Sutton RS., Werbos PJ., The MIT Press, 97

Applications (EEG, Evokek Potentials, Neural Spikes)

Atiya AF., Bower JM. (1988) "Optimal Neural Spike Classification" in *Neural Information Processing System*, ed. by Anderson DZ., American Institute of Physics, 95

Eberhart RC., Dobbins RW., Webber WRS. (1989) "EEG Spike detection using Back-Propagation Neural Networks" (Abstract), *International Joint Conference on Neural Network*, Washington, 2, 637

Cheung JY., Hull Jr. SS. (1989) "Detection abnormal ECG using Neural Network approach", *Proceedings of International Conference of IEEE Eng. in Medecine and Biology Soc.*, Seattle, 6, 2015

Hiraiwa A., Shimohara K., Tokunaga Y. (1990) "EEG topography recognition by Neural Networks", *IEEE Engineering Medecine and Biology Magazine*, 9, 3, 39

Holdaway RM., White MW., Marmarou A. (1990) "Classification of Somatosensory-Evoked Potentiials Recorded from PAtients with Severe Head Injuries", *IEEE Engineering Medecine and Biology Magazine*, 9, 3, 43

Hassoun MH., Spitzer AR. (1988) "Neural Network Identification and Extraction of Repetitive Superimposed Pulses in Noisy 1-D Signals" (Abstract), *NN Supplement: International Neural Network Society (INNS) Abstracts*, 1, 443

Jansen RF. (1990) "The Reconstruction of an Individual Spike trains from Extracellular Multineuron recordings using a Neural Network Emulation Program", *Journal of Neuroscience Methods*, 35, 203

Lo PC., Principe JC. (1989) "Dimensionality Analysis of EEG Segments: Experimental Considerations", *International Joint Conference on Neural Network*, Washington, 1, 693

Pietka E. (1989) "Neural Network for ECG Classification", *Proceedings of International Conference of IEEE Eng. in Medecine and Biology Soc.*, Seattle, 6, 2021, 89 [5880]

Tam DC., Perkel DH. (1989) "A Model for Temporal Correlation of Biological Neuronal Spike Trains", *International Joint Conference on Neural Network*, Washington, 1, 781

Tam DC., Perkel DH., Tucker WS. (1988) "Correlation of Multiple Neuronal Spike Trains Using the Back-Propagation Error Correction Algorithm" (Abstract), *NN Supplement: International Neural Network Society (INNS) Abstracts*, 1, 277

Tam DC., Perkel DH., Tucker WS. (1988) "Temporal Correlation of Multiple Spike Trains Using the Back-Propagation Correction Algorithm" (Abstract), *NN Supplement: International Neural Network Society (INNS) Abstracts*, 1, 278

Thakor NV (1991) "Adaptive Fourier Serie Modeling of Time Variable Evoked Signals: Study of Human Somatosensory Evoked Response to Etomidate Anesthesic", *Electroencephalography and Clinical Neurophysiology*, 80, 108

Uncini A, Marchesi M., Orlandi G., Piazza F. (1990) "Improved Evoked Potential Estimation Using Neural Network", *International Joint Conference on Neural Network*, San-Diego, 2, 143

Wong Y., Banik J., Bower JM. (1988) "Neural Network for Template Matching: Application to Real-Time Classification of the Action of Real Neurons" in *Neural Information Processing System*, ed. by Anderson DZ., American Institute of Physics, 103

Applications (Myoelectric Signals)

Kelly MF., Parker PA., Scott RN. (1990) "The Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study", *IEEE Transactions on Biomedical Engineering*, 37, 3, 221

Kelly MF., Parker PA., Scott RN. (1990) "Myoelectric Signal Analysis Using Neural Networks", *IEEE Engineering Medecine and Biology Magazine*, 61, 90

Applications (ECG)

Carroll TO., Ved H. (1989) "A Neural Network for ECG Analysis" (Abstract), *International Joint Conference on Neural Network*, Washington, 2, 575

Xue Q., Hu y., Tompkins WJ. (1989) "A Neural Network weight pattern study with ECG pattern recognition", *Proceedings of International Conference of IEEE Eng. in Medecine and Biology Soc.*, Seattle, 6, 2023

Strand EM., Jones WT. (1990) "A Neural Network for tracking the prevailing heart rate of the ECG", *Proceedings Third Annual IEEE Symposium on Computer-Based Medical System*, 358

Sandham WH., Deehan C. (1989) "Digital signal processing of the VECP - survey and future directions", *IEEE Colloquium on Biomedical Applications of Digital Signal Processing*, London, 144, 5

Zhu K., Noakes PD., Green ADP. (1990) "Training Neural Network for ECG feature detection", *International Neural Network Conference*, Paris, 1, 137

Applications (Speech Recognition)

Danielson S., Telephon J. (1990) "Recognition of Danish Phonemes Using an Artificial Neural Network", *International Joint Conference on Neural Network*, San-Diego, 3, 677

Murdock JY. (1988) "Improvement on Speech Recognition and Synthesis for Disabled Individuals Using Fuzzy Neural Network Retrofits", *Proceedings IEEE International Conference on Neural Network*, San-Diego, 2, 251

Wan EA. (1990) "Temporal Back-Propagation for FIR Neural Networks", *International Joint Conference on Neural Network*, San-Diego, 1, 575

Waibel AH., Hanazawa T., Hinton G., Shikano K., Lang KJ. (1989) "Phonem Recognition Using Time-Delay Neural Network", IEEE Transactions on Acoustic Speech and Signal Processing, March, 37

Waibel AH., Lang KJ. (1990) "A Time-Delay Neural Network Architecture for Isolated Word Recognition", *Neural Networks*, 3, 23

Applications (Miscellaneous)

Eckhorn R., Reitbock HJ., Arndt M., Dicke P. (1989) "Feature Linking via Stimulus - Evoken Oscillations: Experimental Results from Cat Visual Corttex and Functional Implication from a Network Model, *International Joint Conference on Neural Network*, Washington, 1, 723

Gindi GR., Darken CJ., O'Brien KM., Stetz ML., Deckelbaum LI. (1991) "Neural Network and Conventional Classifiers for Fluorescence-Guided Laser Angioplasty", *IEEE Transactions on Biomedical Engineering*, 38, 3, 246

McCullogh N. (1990) "Recurrent Networks for Learning Stochastic Sequences", *International Joint Conference on Neural Network*, San-Diego, 3, 561

ADAPTIVE FILTERING

Levin E., Gewirtzman R., Inbar GF. (1991) "Neural Network Architecture for Adaptive System Modeling and Control", *Neural Networks*, 4, 185

Mathews MB., (1989) "Nonlinear Adaptive Filtering using Neural Network", *Mitt. AGEN.*, 51, spec. issue, 13

Narendra KS. (1991) "Adaptive Control Using Neural Networks" in *Neural Networks for Control*, ed. by Miller WT., Sutton RS., Werbos PJ., The MIT Press, 115

Saratchandran P. (1991) "Dynamic Programming Approach to Optimal Weight Selection in Multilayer Neural Networks", *IEEE Transactions on Neural Networks*, 2, 465

Thakor NV. (1990) "Adaptive Filtering of Evoked Potentials", *IEEE Transactions on Biomedical Engineering*, BME-34, 1, 6

Thakor NV. (1991) "Applications of Adaptive Filtering to ECG Analysis: Noise Cancellation and Arrhuthmia Detection", *IEEE Transactions on Biomedical Engineering*, 38, 8, 785

Uncini A., Marchesi M., (1990) "Nonlinear adaptive filter using Neural Network", *International Neural Network Conference*, Paris, 1, 158

Uncini A., Marchesi M. (1990) "An adaptive Neural Network filter for evoked potentials", IEEE International Symposium on Circuits and Systems, 2, 1086

Widrow B., Winter R. (1990) "Neural Networks for Adaptive Filtering and Adaptive Pattern Recognition" in *An Introduction to Neural and Electronics Networks*, ed. by Zornetzer SF., Davis JL., Clifford L., Accademic Press, 249

Adaptive Filtering (no mention of ANN technology)

Furst M., Blau A. (1991) "Optimal a Posteriori Time Domain Filter for Average Evoked Potentials", *IEEE Transactions on Biomedical Engineering*, 38, 9, 827

Iyer VK., Ploysongsang Y., Ramamoorthy PA. (1990) "Adaptive Filtering in Biological Signal Processing", *Critical Reviews in Biomedical Engineering*, 17, 6, 531