

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Theses, Dissertations, and Student Research from
Electrical & Computer Engineering

Electrical & Computer Engineering, Department of

Fall 11-5-2010

IDENTIFICATION OF PSYCHOLOGICAL PATTERNS USING NEURAL NETWORKS APPROACH

Panpan Hu

University of Nebraska – Lincoln, panpan.hu@huskers.unl.edu

Follow this and additional works at: <http://digitalcommons.unl.edu/elecengtheses>



Part of the [Electrical and Computer Engineering Commons](#)

Hu, Panpan, "IDENTIFICATION OF PSYCHOLOGICAL PATTERNS USING NEURAL NETWORKS APPROACH" (2010).
Theses, Dissertations, and Student Research from Electrical & Computer Engineering. 11.
<http://digitalcommons.unl.edu/elecengtheses/11>

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Theses, Dissertations, and Student Research from Electrical & Computer Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

IDENTIFICATION OF PSYCHOLOGICAL PATTERNS USING NEURAL NETWORKS APPROACH

by

Panpan Hu

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Hamid Vakilzadian

Lincoln, Nebraska

November, 2010



IDENTIFICATION OF PSYCHOLOGICAL PATTERNS USING NEURAL NETWORKS APPROACH

Panpan Hu, M.S.

University of Nebraska, 2010

Advisor: Hamid Vakilzadian

Recent studies have shown that Artificial Neural Networks (ANNs) are suitable for recognizing patterns in the medical area. However, no study has been done to show whether or not they are also effective in the psychological area. In this study, ANNs are developed for six psychological cases related to sociobehavioral functioning. The cases are independent living skill deficits, disorder management deficits, occupational skill deficits, social skill deficits, dysregulation of anger/aggression, and substance abuse. Two models, one based on a backpropagation algorithm and the other based on a posteriori probability approach, were developed. The models were tested using data from 118 patients in a Community Transition Program (CTP). For each case, a certain percentage of data was randomly selected for training the network, and the remaining data were used for testing the network. DESIRE was used to test the developed models. The networks using DESIRE correctly identified 61.0%, 56.8%, and 56.1% of the test cases in the dysregulation of anger/aggression, substance abuse, and social skill deficits models, respectively. The results were also compared with those obtained from the MATLAB Neural Network toolbox. While MATLAB builds the model internally without identifying the type of the model it is building, the performance results were

close to those obtained by DESIRE. The neural networks for dysregulation of anger/aggression and social skill deficits were grouped into a single network, which provided 42.1% accuracy on the test data. However, the combined network of dysregulation of anger/aggression and substance abuse achieved 36.0% accuracy for the test data. Finally, the ANNs developed were used to identify the 6 problem assessment cases for the 37 untitled patients in the CTP database. The results obtained illustrate that the ANN approach can be a valuable method for mining the data for clinical assessment. While there was not enough data in the database to completely train the models, the results obtained from the limited CTP database show that ANN can be a promising method of identifying patterns of psychological problem with a high degree of accuracy. However, more data is needed in order to make a definite conclusion on its increased predictability.



Acknowledgement

The research in this thesis was carried out during my graduate study in the Department of Electrical Engineering at the University of Nebraska - Lincoln. Many people helped me to complete this work. First, I would like to express my gratitude to my thesis adviser, Dr. Hamid Vakilzadian, who has a solid background in electrical engineering and wide research experience in modeling and simulation. I am grateful for his guidance, invaluable advice and support during my graduate program. I highly appreciate Dr. William Spaulding for his help in making me understand the psychological issues and for his guidance in interpreting the results, indentifying the cases that were of importance in this filed, and serving as a members of my graduate committee. I would like also to thank Dr. Mustafa Cenk GURSOY for consenting to serve in my thesis committee and review of my thesis.

Contribution of Dr. Granino Korn, Emeritus Professor of the Department of Electrical Engineering, University of Arizona for making this software available free of charge in this research as well as his help in making me understand some of the terminology used in entering the model is acknowledged, and they are very much appreciated.

I would like to acknowledge the contribution of my colleagues Charles Davidson, Elizabeth Cook, and Amin Fakhrazari for their assistance and support in my laboratory work and writing my thesis. Their discussions and collaboration were very important in this study. I also would like to thank the Department of Electrical Engineering for providing me with the support, friendly, and challenging research

environment to complete my thesis.

I would also like to thank my parents for their love, help, support and understanding during the course of my study at the University of Nebraska-Lincoln.

Contents

Chapter 1 Introduction.....	1
1.1 Background and Motivation	1
1.2 Survey of Related Work	4
1.3 Task of Current Research	6
Chapter 2 Methodology	8
2.1 Description of Neural Network	8
2.2 Backpropagation Algorithm	11
2.2.1 Momentum Term.....	12
2.2.2 Backpropagation with Batch Learning	12
2.3 Softmax Function and Posteriori Probability Approach	13
2.4 DESIRE software.....	15
2.5 Matlab™ Neural Network Toolbox	17
Chapter 3 Data Preparation.....	19
3.1 Data Collection	19
3.2 Treatment of Missing Data in the Database	20
3.3 Data Transformation	22
3.4 Assumptions and Requirements.....	24
Chapter 4 ANN Models for Single Problem Case.....	25
4.1 Single Output Problem Cases	25
4.2 The Neural Network Models and Architectures	27
4.3 Programming Flowcharts	31
Chapter 5 ANN Models for Two Problem Cases	34
5.1 Development of Models for Two Problem Groups.....	34
5.2 Untitled Patient Assessments.....	35
Chapter 6 Results.....	37
6.1 Validations of the One-output Models	37
6.2 Validations of the Two-output Models	43



6.3 Prediction of Other Cases.....	45
Chapter 7 Conclusion and Future Work	47
Bibliography	51
Appendices.....	54

Chapter 1 Introduction

Electronic data processing technologies have been used to aid in clinical diagnosis for more than 20 years [1]. Up to now, these technologies have been applied in various medical fields such as acute myocardial infarction in emergency department and predicting discharge destination from rehabilitation hospitals. However, these techniques have not been utilized in psychiatric or psychological rehabilitation due to the complexity of the cases and nonlinearity of the parameters.

1.1 Background and Motivation

The SOCRATES [2] for a decision support system is a top-down architecture to help clinical decision making in psychiatric rehabilitation. The system helps the decision making process by creating a database for the patient history and using software techniques on the data in order to provide meaningful information for decision making purpose to the treatment team [2]. The data source is used for SOCRATES come from the Community Transition Program (CTP) of the Nebraska Lincoln Regional Center. CTP, which was founded in 1981 and closed in 2009, was a program to help a special population of individuals with severe mental illnesses as well as to examine the process, policy issues, and outcome of an intensive psychosocial rehabilitation [3].

Skill deficits indicate failure to show behaviors that are needed to accomplish a

specifiable task. The failure is not completely attributed to specific problems at molecular levels of functioning, i.e. neurophysiological dysregulation or cognitive impairment. For the purposes of rehabilitation for disabling mental illness, skills are organized into categories of functionally related elements. These categories comprise the specific skill deficit problem titles [4].

Skills represent complex combinations of abilities, spanning all levels of biobehavioral¹ functioning, plus acquired information stored in brain memory, in continuous interaction with complex environmental conditions. The acquired information is often quite extensive and complex, accumulated over the entire course of the treatment. A skill deficit problem is initially hypothesized in order to achieve optimal rehabilitation with skill training interventions related to that type of the problem. That is, the interventions are designed to provide effective treatment by providing information (education), guided rehearsal of key components of the desired skill, in vivo practice, coaching and related techniques [4].

Psychophysiological dysregulations indicate failure to effectively regulate one's state of mental and bodily interaction, resulting in subjective distress, cognitive impairment, and/or disruption of skill performance. Psychophysiological dysregulation has distinct patterns that reflect the relative independent activity of neurophysiological systems, especially patterns of autonomic activation. These patterns comprise the specific

¹ Of or relating to the interrelationships among psychosocial, behavioral, and biological processes, as in the progression or treatment of a disease [26].



problems [4].

As mentioned above, the diagnosis of these problems follows the complex patterns combined with many factors. Human brain sometimes cannot make a correct decision. The increased demand for correct diagnosis necessitates that computer-aided decision support is involved in diagnosis process. The decision support is an interdisciplinary project requiring psychology, statistics, computer and informatics. It can help the treatment team in decision making process. As mentioned earlier, artificial neural networks have been studied extensively in clinical medicine but not in psychiatry. It enlightens us to investigate the ANN's application for psychiatric patterns. Neural network techniques can learn the problem patterns and provide the diagnosis that can help the treatment team.

Neural networks can be successfully used in many applications. Pattern recognition is one of these applications. It finds the unknown patterns by first going through a learning process that will require a training data set over many trials. Test patterns are then presented to the trained network to relate the new pattern with one of the learnt categories of patterns. When a network is well trained, it can identify the category of the test patterns by extracting the information from the trained data set. In addition to the pattern recognition, neural networks can be used to simulate a nonlinear functional relationship as (1):

$$y = f(x) \tag{1}$$

where input x and output y are known, but the function f that relates y to x needs to be



determined. The sin function in $y=\sin(x)$ is a classic example to relate y to x . The third application of ANN is pattern association. In this application the network is first train according with a set of given patterns. This phase is called storage phase. The patterns are retrieved from the memorized patterns (stored patterns) in response to a noisy or distorted version of that pattern [5].

There are two common types of learning methods in ANN. They are:

1. Supervised learning.
2. Unsupervised learning.

In supervised learning, the training data consist of input patterns and desired output patterns. In unsupervised learning, only sample of input patterns are provided to the learning system. It seeks to determine how the data are organized. In this project, the supervised learning has been used.

1.2 Survey of Related Work

Data mining, the extraction of hidden patterns from data, is a powerful technology with great potential for finding important information from a database. Data mining techniques permit predicting future trends and behaviors, allow businesses to make proactive, knowledge-driven decisions [6]. Artificial neural network being one of the data mining techniques can be applied to a data set to extract useful information. This study is inspired by [7]. In order to develop data mining model for decision support system, Fong

et.al proposed seven steps. The first step is to establish mining goals that also includes the expected accuracy and usefulness of the results. The second step is selecting attributes that worth to be considered and decent sample size of useful data. Preprocessing data contains filtering out noisy, erroneous or irrelevant data and handling missing data is the third step. Transforming data and projecting the data onto working spaces is the next step. Storing the input data together under a unified scheme comes later. This is the most critical step is the mining of data. The last step is to evaluate mining results and perform various operations. The user feedback from this step can prompt changes to earlier steps. The authors claim that this process is flexible enough to suit any multidimensional database and work with most popular data mining tools [7].

The preliminary studies for application of ANN in a number of medical cases have shown of potential value. In particular in some cases, neural networks have outperformed physicians in predicting clinical outcomes [8]. G. Baxt [8] used artificial neural network to identify myocardial infarction in patients referring to an emergency department with anterior chest pain. The network was tried on 331 patients suspected of having had a myocardial infarction. The ability of the network to distinguish patients with or without acute myocardial infarction was compared with that of physicians caring for the same patients. The results showed neural network had 20% higher diagnostic sensitivity² and 12% higher diagnostic specificity³.

² Ratio of true positive diagnoses to true positive + false negative

³ Ratio of true negative diagnoses to true negative + false positive

A detailed review of the current neural network methods for early diagnosis of acute myocardial infarction is provided by R.L. Kennedy et al [9]. This study has confirmed that artificial neural networks are adept at recognizing patterns in sets of clinical data from chest pain patients. 39 items of clinical and electrocardiographic data are used in [9] to derive 53 binary inputs for training a back propagation network. On test data, overall accuracy of correct diagnose from ANN was higher than corresponding values using linear discriminate analysis⁴ (LDA). This work showed that the computer-aided analysis of clinical factors and biochemical markers such as myoglobin could be an ideal support for emergency room physicians.

A.A. El-Solh, et. al. in [10] trained an ANN model for clinical pattern set derived from 452 patients. They validated the model on 209 consecutive patients admitted to postacute geriatric rehabilitation units. ANN is used to predict discharge to the community postacute rehabilitation with high degree of accuracy. It was valuable to predict the discharge to the community of older adults with multiple comorbidities after an acute hospitalization.

1.3 Task of Current Research

The work in this research involves developing and training ANN models for

⁴ LDA is closely related to ANOVA (analysis of variance) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements [27].



predicting patterns for psychological patients from The Nebraska Community Transition Program. The specific tasks of this research include:

- Transformation of the raw data.
- Development of neural network models for six psychosocial problem classification cases.
- Evaluation of the developed performance of models.

1.4 Organization of the Thesis

The remainder of this thesis is organized as follows. The methodology is presented in Chapter 2. This includes general neural network models including back propagation algorithm, Softmax and a posteriori probability presented in DESIRE software [18] and MatlabTM Neural Network Toolbox.

Data preparations are explained in Chapter3 and artificial neural networks for six psychological problem cases are described in Chapters 4 and 5.

In Chapter 6, the results of the ANN models are presented and compared. The significance of the results is also analyzed in detail.

Concluding remarks and discussion for future work are presented in Chapter 7. References and appendixes are provided at the end.

Chapter 2 Methodology

2.1 Description of Neural Network

An artificial neural network is a massively parallel distributed processing system made up of simple processing units called neurons. The network has the ability to learn from the experimental data (for gaining knowledge) expressed by inter-unit connection strengths (weights) and can make the knowledge available for later use [5].

Fig. 1 shows the structure of an artificial neural network model. Every neural network consists of three types of layers: input layer, hidden layers and output layer. Furthermore, there are two kinds of output. Actual output refers to the output of the neural network and the target is the desired output which the ANN is trained to recognize. The difference between the actual output and the desired output is the network error.

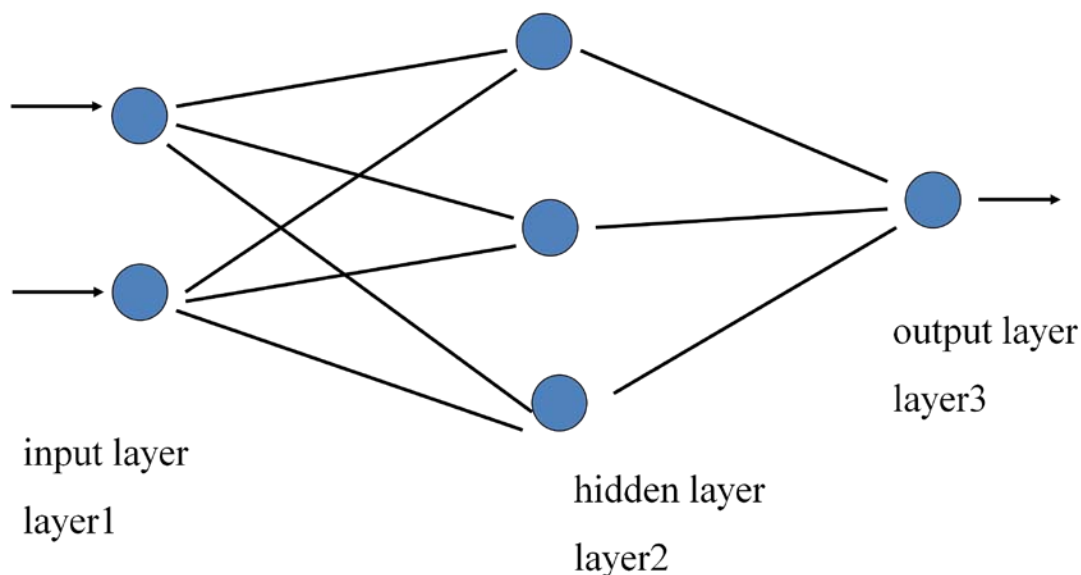


Fig. 1: Basic neural network model

As shown in Fig. 2, each neuron consists of four basic elements:

- 1) A set of links connecting different inputs x_i , each of which is characterized by a weight w_{ki} .
- 2) An external bias b_k to compensate for error
- 3) An adder to sum the input x_i weighted by w_{ki} and bias.
- 4) A nonlinear activation function f to produce the output y_k of a neuron.

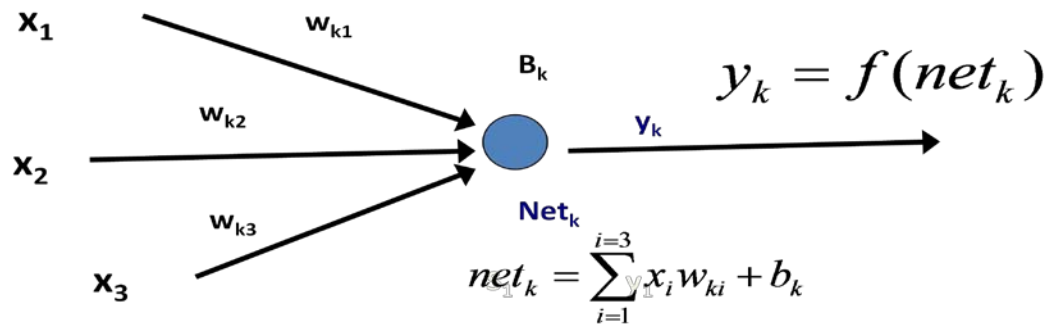


Fig. 2: Model of an artificial neuron

Three typical activation functions are shown in Table 1. Sign activation function is useful for binary classification applications. Sigmoid functions, an S-shaped curve with saturating limits, are helpful for calculating the weight updates in training when derivatives are needed. The derivative of the logistic sigmoid function

$f(net_k) = \frac{1}{1 + e^{-a \times net_k}}$ is given by [11]:

$$\frac{df(net_k)}{dt} = af(net_k)[1 - f(net_k)] \quad (2)$$

where, a specifies the slope of the curve.

Similarly, the derivative of the hyperbolic tangent sigmoid function

$$f(net_k) = \frac{e^{net_k} - e^{-net_k}}{e^{net_k} + e^{-net_k}} \text{ is}$$

$$\frac{df(net_k)}{dt} = 1 - f^2(net_k)$$

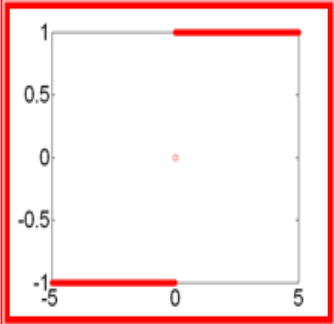
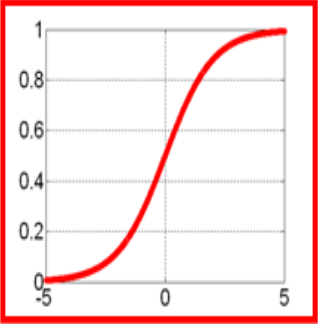
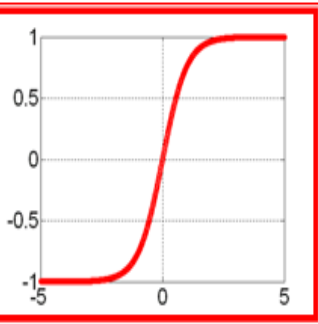
		
Sign Function	Logistic Sigmoid	Hyperbolic Tangent Sigmoid
$f(x) = \begin{cases} 1 : x > 0 \\ 0 : x = 0 \\ -1 : x < 0 \end{cases}$	$f(x) = \frac{1}{1 + e^{-ax}}$	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Table 1: Three typical activation functions

Neural networks can be trained in forward mode or backward mode. In forward training mode, patterns are propagated forwards through the neural network to generate the actual outputs. The error, $e_j(n)$, is calculated as:

$$e_j(n) = y_j(n) - t_j \quad (3)$$

where $y_j(n)$ denotes the actual output and t_j denotes the desired output namely target.

The mean square error function is defined by (4) as:

$$E(n) = \frac{1}{2} \sum_{i=1}^n e_j(n)^2 \quad (4)$$

The error function above influences the weights and bias. In order to optimize the function and reduce the error to zero, a number of algorithms [5] such as delta rule are used. In this study, three approaches are used to train the ANN for recognition of the

psychological patterns.

2.2 Backpropagation Algorithm

Backpropagation Algorithm is a common method for training artificial neural networks to perform a given task. It was first proposed by Arthur E. Bryson and Yu-Chi Ho in 1969 [12]. This method is easy to understand and easy to program. It is usually used for decision making and pattern recognition. In this approach, the error is propagated backwards according to the delta rule [5].

One way to adjust weights in backpropagation algorithm is to use gradient descent learning rule, which is called Delta Rule. In the backward mode, the update of the weight is accomplished by passing error backward from the output layer to the input layer. In this rule, the partial derivative is calculated to adjust the weights to decrease the error. The change in weights is given by (5):

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (5)$$

where $\Delta w_{ji}(n)$ denotes the correction to the weight w_{ji} from neuron i to neuron j at n^{th} trial. η is the learning rate according to the back propagation algorithm. The error function (4) is minimized using the chain rule that can be expressed as:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} = -\eta \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial f_j(n)} \frac{\partial f_j(n)}{\partial net_j(n)} \frac{\partial net_j(n)}{\partial w_{ji}(n)} \quad (6)$$

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n); \quad \frac{\partial e_j(n)}{\partial f_j(n)} = -1; \quad \frac{\partial f_j(n)}{\partial net_j(n)} = f_j'(n); \quad \frac{\partial net_j(n)}{\partial w_{ji}(n)} = x_i(n); \quad (7)$$

$$\text{Therefore, } \Delta w_{ji}(n) = \eta e_j(n) f_j'(n) x_i(n) \quad (8)$$

In $(n+1)^{\text{th}}$ trial, the weight is updated to $w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n)$.

(9)

2.2.1 Momentum Term

The delta rule decreases the error in order the network output to approach the target output. This process can be unstable and may result in oscillation if it falls into the local extremums. A simple way to prevent the instability is to add a momentum term, (α) , to the delta rule [5].

$$\Delta w_{ji}(n) = \eta e_j(n) f_j'(n) x_i(n) + \alpha \Delta w_{ji}(n-1) \quad (10)$$

Where α is a small positive constant less than 1.

2.2.2 Backpropagation with Batch Learning

Backpropagation with batch learning [13] implies correcting the bias and weights over all input patterns rather than updating them one by one. Optimization with batch learning minimizes the error computed in N successive trials for each of the N input patterns. The weight and bias increments are accumulated for N trials and are updated at the end of each N -pattern “epoch”. For this purpose, we compute:

$$e_j(n) = y_i(n) - t_i \quad (11)$$

$$\Delta w_{ji}(n) = \eta e_j(n) f_j'(n) x_i(n) + \alpha \Delta w_{ji}(n-1) \quad (12)$$

Accumulating auxiliary weight W over N trials will yield,



$$W_{ji}(n+1) = W_{ji}(n) + \Delta w_{ji}(n) \quad (13)$$

Following the completion of a batch of N trials, the updates (auxiliary weights W) are assigned to actual weights w . This assignment occurs only once per N trials:

$$w_{ji}(N) = W_{ji}(N) \quad (14)$$

2.3 Softmax Function and Posteriori Probability Approach

In this approach the ANN is regarded as an optimal classifier, and each neuron in the output layer represents one class. The neural network is able to produce not only integer/binary outputs but also decimal values that can be interpreted as probabilities. Based on that, each input pattern can be classified to a certain class with the maximum probability where output neuron is used to select the class. This approach is inspired by statistical pattern recognition which uses Bayes' theorem [15] to evaluate the posteriori probabilities. In Bayesian statistics [15], the posterior probability of a random event or an uncertain proposition is the conditional probability that is assigned after the relevant evidence is taken into account. Thus, we let the neural network produce estimates of a posteriori probabilities unlike to the classical pattern recognition methods.

Softmax activation function assumes the neural network is being trained, for example, to minimize the mean squared error. The neural network has sufficient number of neurons to produce the required mapping from input space to targets (output layer). The produced output probabilities are between 0 and 1 that they all add to 1 for each input pattern. The outputs represent the probability that the input pattern is in the specified

class. In order to guarantee that the output sums to 1, a new output neuron with the following softmax activation function is introduced:

$$y_k = \frac{\exp(\text{net}_k)}{\sum_j \exp(\text{net}_j)} \quad (15)$$

The denominator in (15) sums over all network outputs. The softmax function is different from sigmoid functions because the outputs are scaled by the total activation at the output layer.

The conditional average of the target data from Bishop [16], is derived as the mean squared error (MSE) (16):

$$E = \frac{1}{2} \sum_k \iint [y_k(x, w) - t_k]^2 p(t_k, x) dt_k dx \quad (16)$$

where the joint probability can be factored as the product of the input probability density function $p(x)$ and the conditional probability of the target data given the input $p(t_k|x)$.

The squared term can be written as (17):

$$(y_k(x, w) - t_k)^2 = (y_k(x, w) - \langle t_k | x \rangle + \langle t_k | x \rangle - t_k)^2 \quad (17)$$

where $\langle t_k | x \rangle$ is the conditional average given by $\langle t_k | x \rangle = \int t_k p(t_k | x) dt_k$.

The expression in (17) can be rewritten in the form of (18):

$$(y_k(x, w) - t_k)^2 = (y_k(x, w) - \langle t_k | x \rangle)^2 + 2(y_k(x, w) - \langle t_k | x \rangle)(\langle t_k | x \rangle - t_k) + (\langle t_k | x \rangle - t_k)^2 \quad (18)$$

Now if the expression in (18) is substituted back into the MSE in (16) and the results are

simplified, we obtain:

$$E = \frac{1}{2} \sum_k \int (y_k(x, w) - \langle t_k | x \rangle)^2 p(x) dx + \frac{1}{2} \sum_k \int (\langle t_k^2 | x \rangle - \langle t_k | x \rangle^2) p(x) dx \quad (19)$$

In (19), the second term is independent to the network and it will not change during training. The minimum of the first term is obtained when the weights produced will satisfy

$$y_k(x, w) = \langle t_k | x \rangle \quad (20)$$

This allows us to work with the numerical outputs of the network as a posteriori probabilities for pattern recognition, an important derivation.

2.4 DESIRE software

DESIRE (Direct Executing Simulation in Real Time) is a very fast interactive simulation environment for dynamic systems, which has been used in industry and academia since 1986 [17]. The current version is compatible with Microsoft Windows (including Vista and Windows 7), Linux, and Cygwin. This simulation program supports differential equations and vector/matrix operations. It also includes a capability for simulating neural networks and fuzzy logic systems [18]. This software uses double-precision (64-bit) floating-point arithmetic and accepts command scripts and model descriptions in a readable mathematical notation. Command script language is itself a general-purpose mathematical language and handles vectors, matrices, and even complex numbers (e.g., frequency-response and root-locus plots). Each program begins

with an experiment-protocol script. When the experiment-protocol scripts encounter a "drun" statement, a built-in runtime compiler automatically compiles a DYNAMIC program segment listing model equations. Moreover, very fast compilation simplifies interactive modeling. Experimenters can immediately observe results of a programmed model [18].

When the icon Wdesire.bat is double clicked, a file manager window, an editor window and a command window show up as in Fig. 3. After a program is plugged into the editor window, the red OK button on editor window transfers the selected program to command window. Typing zz or erun in command window makes it run.

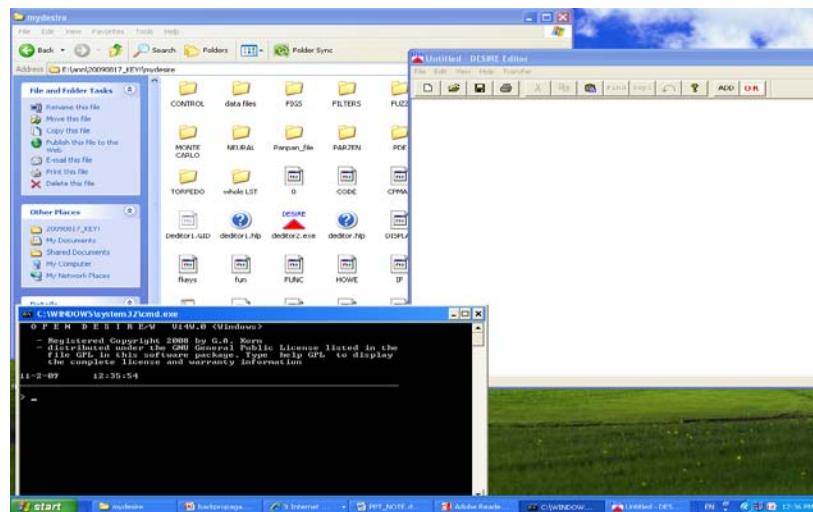


Fig. 3: DESIRE showing a file manager window, an editor window, and a command window

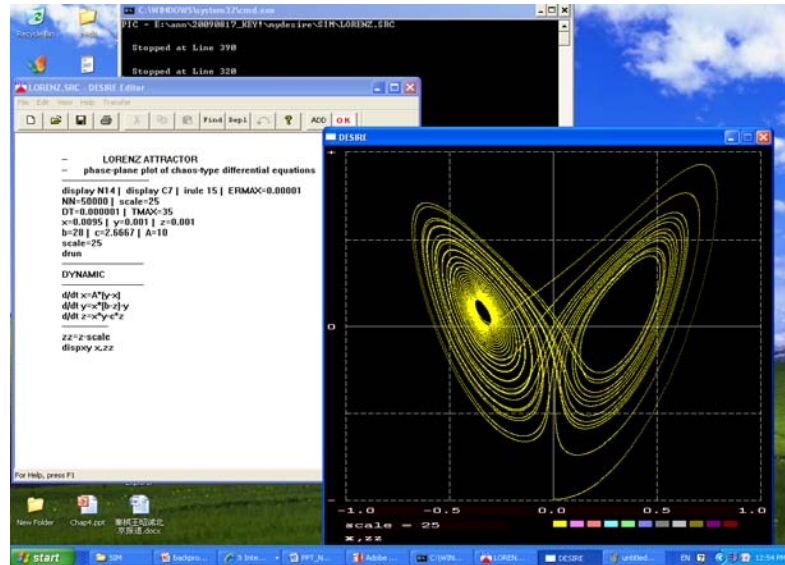


Fig. 4: Running a program in DESIRE

2.5 Matlab™ Neural Network Toolbox

Matlab™ Toolbox has tools for designing, implementing, visualizing, and simulating neural networks. These networks are invaluable for applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control [19]. Neural Network software in this toolbox provides comprehensive support for many network paradigms. Moreover, it provides graphical user interfaces (GUIs) that enable you to specify and manage your networks. It also includes many commonly used algorithms. Typing “nntool” in command window will bring a dialog window to import the input data and target data and assign certain functions and parameters of the network.

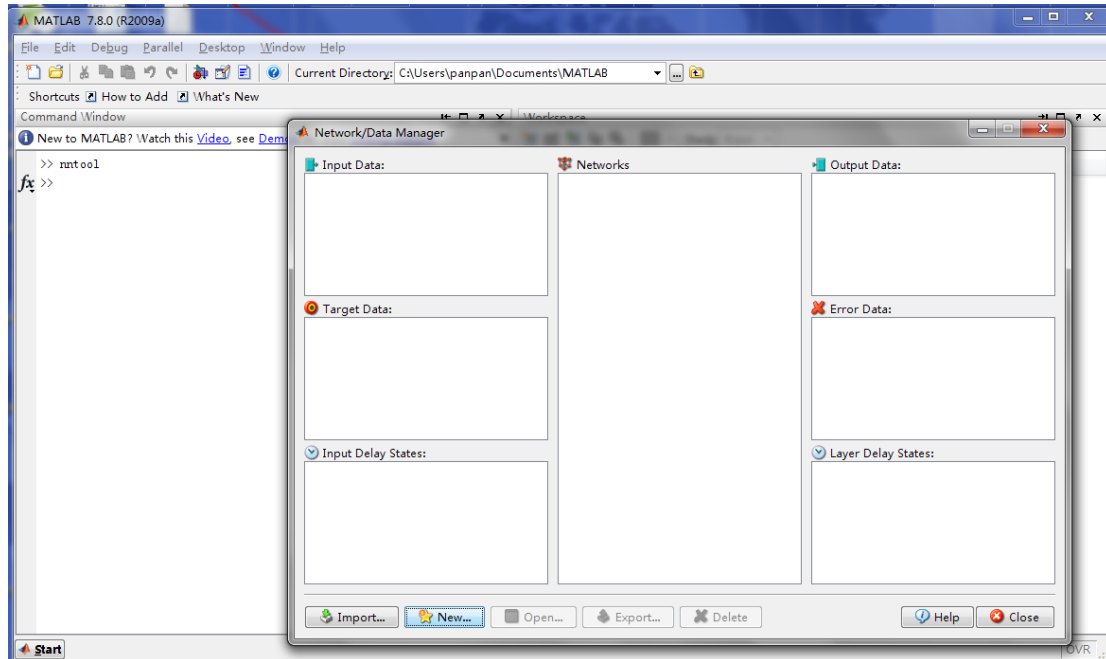


Fig. 5: Matlab™ neural network toolbox interface

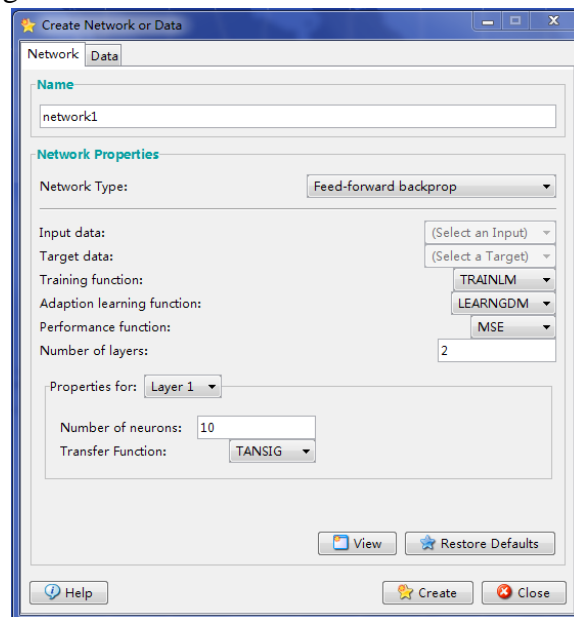


Fig. 6: Neural network specification

The modular, open, and extensible design of the toolbox simplifies the creation of customized functions and networks [19].

Chapter 3 Data Preparation

3.1 Data Collection

The patient data used for training and testing of the neural networks in this study came from Community Transition Program (CTP) in Lincoln, Nebraska. CTP's database includes demographic information (age, gender, etc), clinical information (value obtained from instruments), and problem assessments for 177 admissions.

Problem cases considered in this study all are related to sociobehavioral functioning. They are independent living skill deficits, disorder management deficits, occupational skill deficits, social skill deficits, dysregulation of anger/aggression and substance abuse as are shown in Fig. 7. These problems are heuristically organized as in Fig. 7 and their diagnoses are based on clinical follow-ups.

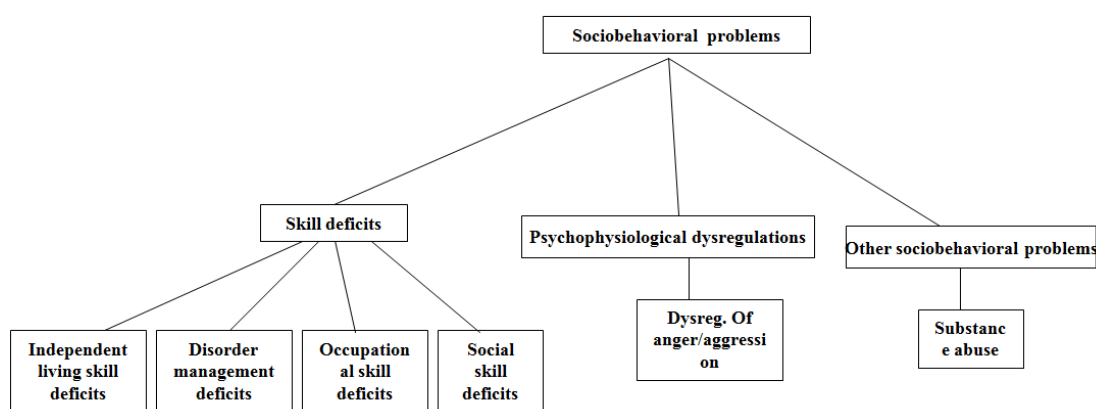


Fig. 7: Problem title hierarchy tree

A set of 34 variables are used to describe these cases for each patient. They were

recorded on admission to CTP or during the treatment process by human clinicians.

These variables are listed in Table 2. The meaning of them is provided in appendix A.

No.	Variable Name	No.	Variable Name	No.	Variable Name
1	psybin3	13	prev_hos	25	ilsige2
2	depbin3	14	age_onse	26	asama2
3	anxbin3	15	occ_func	27	asahit2
4	chobin3	16	ilsipe2	28	asafa2
5	othbin3	17	ilsihy2	29	wcstco2
6	legalMHBbin	18	ilsicl2	30	wcstpe2
7	legalVpGbin	19	ilsiba2	31	wcstre2
8	legalNRRlbin	20	ilsiin2	32	wcste2
9	legalvolbin	21	ilsiho2	33	bdi_tota
10	Gender	22	ilsimo2	34	bhs_tota
11	Age	23	ilsico2		
12	high_ed	24	ilsire2		

Table 2: 34 variables with regards to sociobehavioal functioning

This study was performed using CTP database that was provided by the Department of Psychology, University of Nebraska-Lincoln without patients' identifications. This assured the patient confidentiality.

3.2 Treatment of Missing Data in the Database

A certain number of patients in the CTP database had incomplete data for the 34 variables or the assignment of the problem title was missing. In the database, 22 out of 177 patients (cases) which had too many missing data were excluded from the study. Thirty seven patients didn't have problem title assignment. The remaining 118 patients

had been assigned one or more problem title which indicated the patient had or didn't have that problem title. Table 3 shows the number of cases for each problem title.

Problem title	Total Number of Patients with Recoded One Problem Title	Number of Patients with Recorded Problem Title	Number of Patients Without the Problem Title
Independent Living Skill Deficit	107	99	8
Disorder Management Deficit	114	111	3
Occupational Skill Deficit	110	108	2
Social Skill Deficit	105	83	22
Dysregulation of Anger/Aggression	100	52	48
Substance Abuse	111	49	62

Table 3: Numbers of Cases for Each Problem Title

Since a number of data points for the 34 variables mentioned above were missing in the database, the following procedure [20] was used to estimate the missing values.

This is done by a) finding the mean of that variable from other known cases, and b) using linear regression on the value of the other variables to determine the value of the missing variable. For example, to determine the value of the missing variable VAR1_i for the ith patient, first find the mean of this variable (VAR1) from other known cases. Then, use linear regression expression 21 and the value of the remaining known variables (33 variables in our case) and find VAR1_i.

$$\text{VAR1}_i' = B_1(\text{VAR2}) + B_2(\text{VAR3}) + B_3(\text{VAR4}) \dots + B_{33}(\text{VAR 33}) + b \quad (21)$$

where VAR1_i' is the estimated value of VAR1_i based on the linear combination of all

other variables and constant b . The value of b is selected to compensate for the measurement errors. The coefficients B_1, B_2, \dots are the variance in VAR1. They are calculated using the Statistical Package for the Social Sciences (SPSS) software. A similar procedure was applied to determine the value of the other missing variables [20].

3.3 Data Transformation

For artificial neural networks, binary code was used to indicate presence or absence of a factor, whereas continuous valued variables were assigned to category ranges (design variables) [25]. Discretization of real and integer values to binary was performed by the following steps:

Step1 : Round all the real values to their nearest integer. This step was ignored for the integer variables. For example,

13.76453 becomes 14 and 14.34334 also becomes 14.

Step2: Find the range (maximum and minimum value) of the corresponding variable.

Step3: Based on the range of the value, determine the discrete category. The number of categories and their corresponding increment are determined by an expert in the area.

For instance, age which range from 19 (minimum) to 64 (maximum) was divided into 6 categories named from age_a to age_f.

Category Name	Description
Age_a	less than 20
Age_b	20-29
Age_c	30-39

Age_d	40-49
Age_e	50-59
Age_f	60 or older

Table 4: Age variable division

Step4: Fill in each category with a binary code corresponding with the presence and absence of that variable.

- For presence: 1
- For absence:-1

A 31 years old patient and a 49 years old patient are converted to binary values as shown in Table 5.

Patient ID	Age_a	Age_b	Age_c	Age_d	Age_e	Age_f
1	-1	-1	1	-1	-1	-1
2	-1	-1	-1	1	-1	-1

Table 5: Age variable transformation

In this scheme continuous valued variables were assigned to categorical ranges (design variables). The reason for using negative 1 (-1) for absence for the place of zero (0) is that the product of zero and the network's weight values will produce zero which would impair the influence of the weights.

As discussed above, the 34 real and integer variables with their corresponding categories produced 114 binary inputs. The expansion of variables is described in appendix B. Conversion of the 118 cases (patients) generated no identical input patterns. This guarantees that the inputs for the neural network are unique.

The output of the network was coded as positive when the corresponding patient had

that type of problem title, and it was coded as negative when the patient didn't have it.

3.4 Assumptions and Requirements

Neural networks don't have as many requirements as statistical methods do. It is just assumed that the input items will affect the output and the output is only influenced by the 34 items in Table 2.

In using posteriori probability approach, which is a special neural network, three requirements need to be considered.

- Similar to back propagation algorithm, neural network is being trained to minimize the mean squared error.
- Neural network has a sufficient number of neurons to produce the required mapping from input space to targets.
- Outputs are between 0 and 1, and they all add to 1 for each input pattern. The output represents the probability that the input pattern is in the specified class.

Chapter 4 ANN Models for Single Problem Cases

4.1 Single Output Problem Cases

These networks are built to recognize a single case problem under sociobehavioral functioning (Fig. 7). Four problem cases belong to this skill deficit set. They are:

1) Independent living skill deficit: It indicates insufficient competence and/or performance skills related to the demands of routine adult living such as housekeeping, personal budgeting and banking, utilization of public resources (library, public transportation, etc), maintenance of wardrobe (purchasing clothing, minor repairs, laundry), household shopping and cooking.

2) Disorder management deficit: It is insufficient competence and/or performance of skills related to management of mental illness, including understanding nature and purpose of medication, identification of psychotic symptoms and warning signs of psychotic episodes, management of residual symptoms and deficits and management of stress.

3) Occupational skill deficit: This is insufficient competence and/or performance of skill related to maintaining a work role or similar social role. This problem includes skills generally related to most or all occupational functioning. These are not skills required for particular vocational pursuits. These are skills relevant to job punctuality, self-regulation and pacing, following instructional protocols, appropriately using

problem-solving in unfamiliar situations, and maintaining interpersonal relationships appropriate to the occupational setting.

4) Interpersonal skill deficit: This problem is also known as social skill deficits. It presents insufficient competence and/or performance of skills related to interpersonal interactions, including making conversation, expressing needs, making requests, identifying and resolving ordinary conflicts, establishing and maintaining friendships.

In general, patients who have any of these four problems are defined as the ones who have skill deficits.

The next set of the problems, as shown in Figure 7, is psychophysiological dysregulations. This set indicates failure to effectively regulate one's state of psychophysiological activation, resulting in subjective distress, cognitive impairment, and/or disruption of skill performance. Dysregulation of anger/aggression defines a pattern of psychophysiological dysregulation associated with extreme, explosive and/or socially unacceptable anger and/or aggression. The clinical picture may conform to the diagnostic criteria for explosive personality disorder. This problem is usually not diagnosed separately.

The last problem set in Figure 7 is substance abuse. It refers to a persistent pattern of using alcohol or other drugs to induce an altered state of consciousness, when such use contributes uniquely to other problems or deficits in a person's neurophysiological, cognitive and/or sociobehavioral functioning. Problems consequent to substance abuse may include increased vulnerability to episodic neurophysiological dysregulation and

acute psychosis and socially unacceptable behaviors associated with obtaining substances of abuse.

4.2 The Neural Network Models and Architectures

Six neural network models were developed for the six problems described in the previous section. Since every problem title is relevant to sociobehavioral functioning, all 114 binary inputs derived from 34 variables in Table 2 were used as input to these models.

As shown in Table 3, the first three problems (independent living skill deficits, disorder management deficits, and occupational skill deficits) are very common among the patients who have that type of problem title assessment. The percentage of occurrence for these cases are 99% (99/107), 97% (111/114), and 98% (108/110) respectively. All the available data for each problem in the database were used to train the network for that problem using backpropagation algorithm. They are then used to validate the network. Three methods (backpropagation algorithm, a posteriori probability approach, and Matlab™ Neural Network Toolbox) were also applied for the other three problems cases which were social skill deficits, dysregulation of anger/aggression and substance abuse.

In dysregulation of anger/aggression model, the hundred cases in the database were divided into training sample set and testing data set. Two forty cases of the data, which

were selected randomly from presence or absence dataset of anger/aggression dysregulation respectively, were combined for training sample. The model was tested using the remaining cases in the hundred cases which the network had never been exposed. In the substance abuse model, the training sample consisted of eighty cases. The rest (thirty one cases) were used for evaluation. The social skill deficit model was trained randomly using thirty cases, half of the cases with presence of social skill deficit and half without. The remaining seventy five cases were used as evaluation data during the testing process.

The neural network used in backpropagation algorithm has basic three layers with a single output. Its structure is illustrated in Fig. 8. The input layer has 114 binary inputs that were derived from 34 variables. Each of these inputs is multiplied by its pertaining weight and added by the bias. This value is then passed through a transfer function that activates all the neurons in the hidden layer. Among the activation functions, hyperbolic tangent sigmoid function, *tanh*, is chosen for this analysis because the bounded range from -1 to 1 [21]. The activation of the hidden units is multiplied by the second layer's weights and summed by the bias. The weighted sum is then fed into the same hyperbolic tangent sigmoid function in order to calculate the network output. This value is the current classification which is compared with the desired output. The difference is subsequently propagated backward, and the connection weights and bias are changed to optimize the network. Training is accomplished by repeated sequential presentation of the training pattern set to the network until the error term in the output becomes small. In

training, the learning rate was set to 0.02 and the momentum was adjusted to 0.001 (in hidden layer) and 0.0005 (in output layer) after some trials and error process.

The effect of changing the number of hidden neurons was investigated. This number determines how well a problem can be learned. Too many neurons in hidden layer result in a network that memorizes the patterns, thus resulting in poor generalization. On the other hand, too few neurons make the network ill prepared to learn the pattern well. With respect to the number of neurons in the hidden layer, there are usually four rules of thumb:

- 1) Pick a number between the number of neurons in input layer and the output layer [22].
- 2) Select a number which is not more than one half of the number of neurons in the input layer.
- 3) Choose a number using two thirds of the number of input and output neurons (number of inputs+ outputs)*(2/3) [23].
- 4) Use a number from the following relationship [10].

$$\text{No. of hidden neuron} = \frac{(\text{inputs} + \text{outputs})}{2} + \sqrt{\text{No. of patterns in the training set}}$$

Rule number 4 is selected, as recommend in [10]. Using this rule, 68 neurons are used as shown in Fig. 8. This number of neuron also satisfies most the rules of thumb mentioned above.

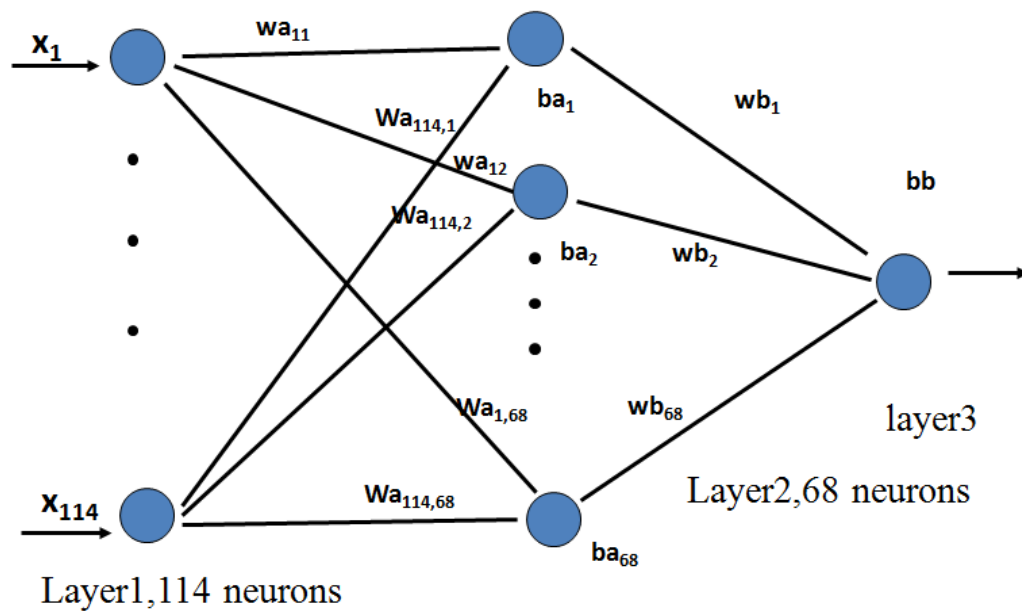


Fig. 8: Single problem neural network structure for using backpropagation algorithm

In posteriori probability approach, the associative memory model is used. In this model, the neural network has 2 layers and 114 neurons for the 114 binary inputs. The product of inputs and weights using the softmax activation function will produce the probability. Two neurons in the second layer refer to two classes (with and without problem title) respectively. If the probability of the first neuron is higher than the second, the input pattern belongs to top class otherwise to the bottom class. The program for determining posterior probability corresponding to Figure 9 is shown in Appendix D.

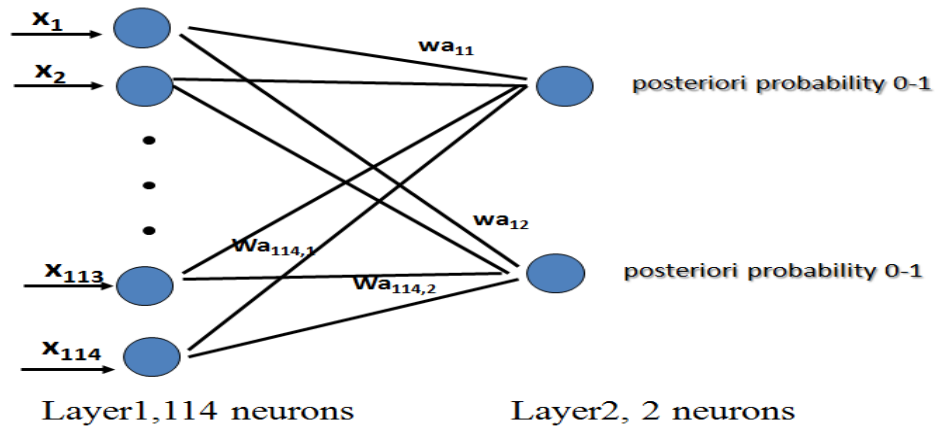


Fig. 9: Single problem neural network structure by posteriori probability approach

The algorithms used by MATLAB™ Neural Network Toolbox are not specified explicitly. A user specifies the settings such as:

Network Type	Feed-forward backpropagation
Training function	TRAINGD(Gradient descent backpropagation)
Adaption learning function	LEARNGD(Gradient descent weight and bias learning function)
Performance function	MSE(mean square error)

Table 6: Matlab™ neural network parameter setting

In using Matlab ANN Toolbox, the number of layers, number of neurons in each layer, and activation function are kept the same as they were used with the back proportion algorithm.

4.3 Programming Flowcharts

The flowcharts below show translation of the ANN models using the backpropagation algorithm and posteriori probability approach. They were used to program the models in DESIRE programming language [18].

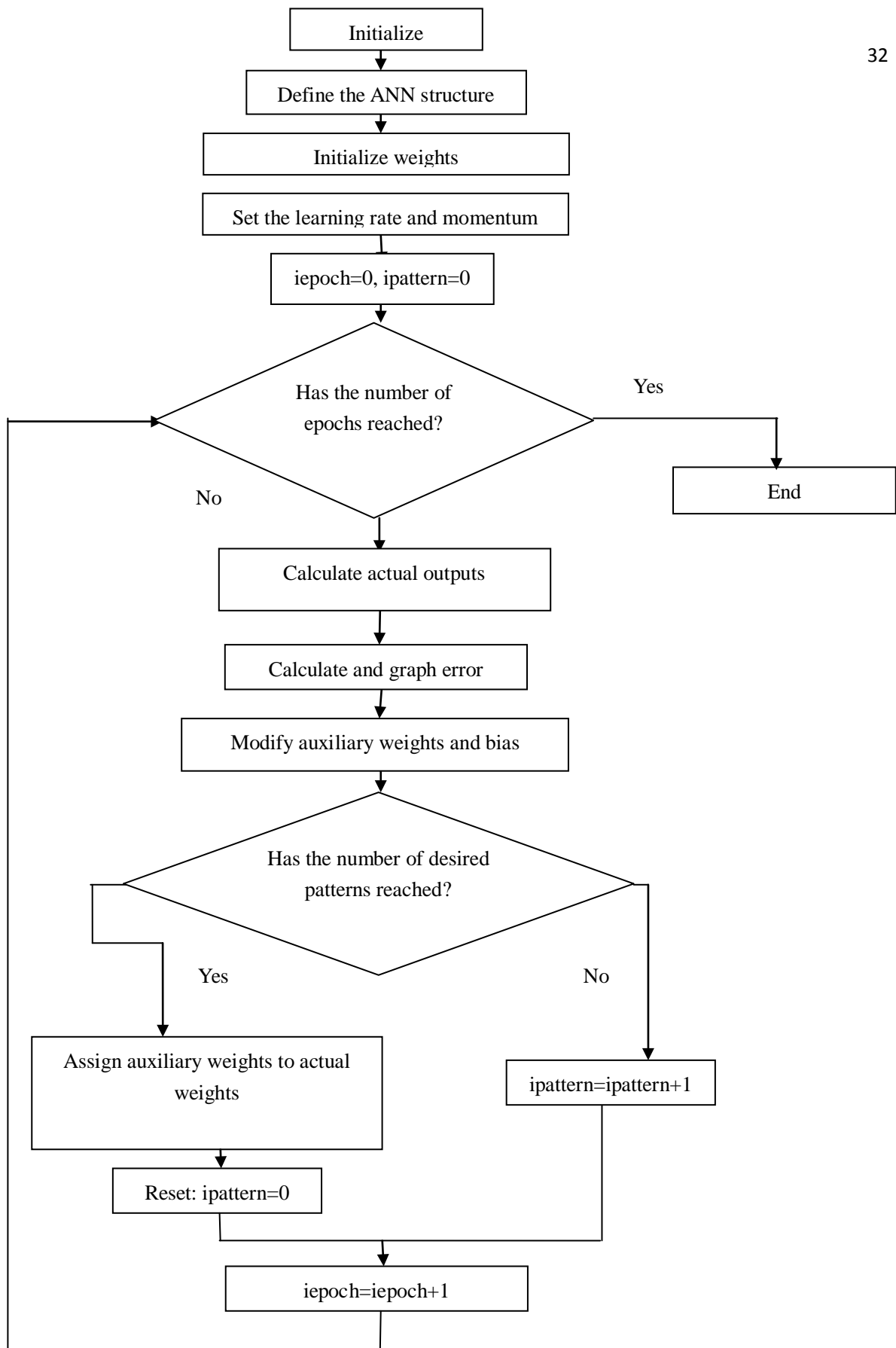


Fig. 10: Backpropagation algorithm flowchart

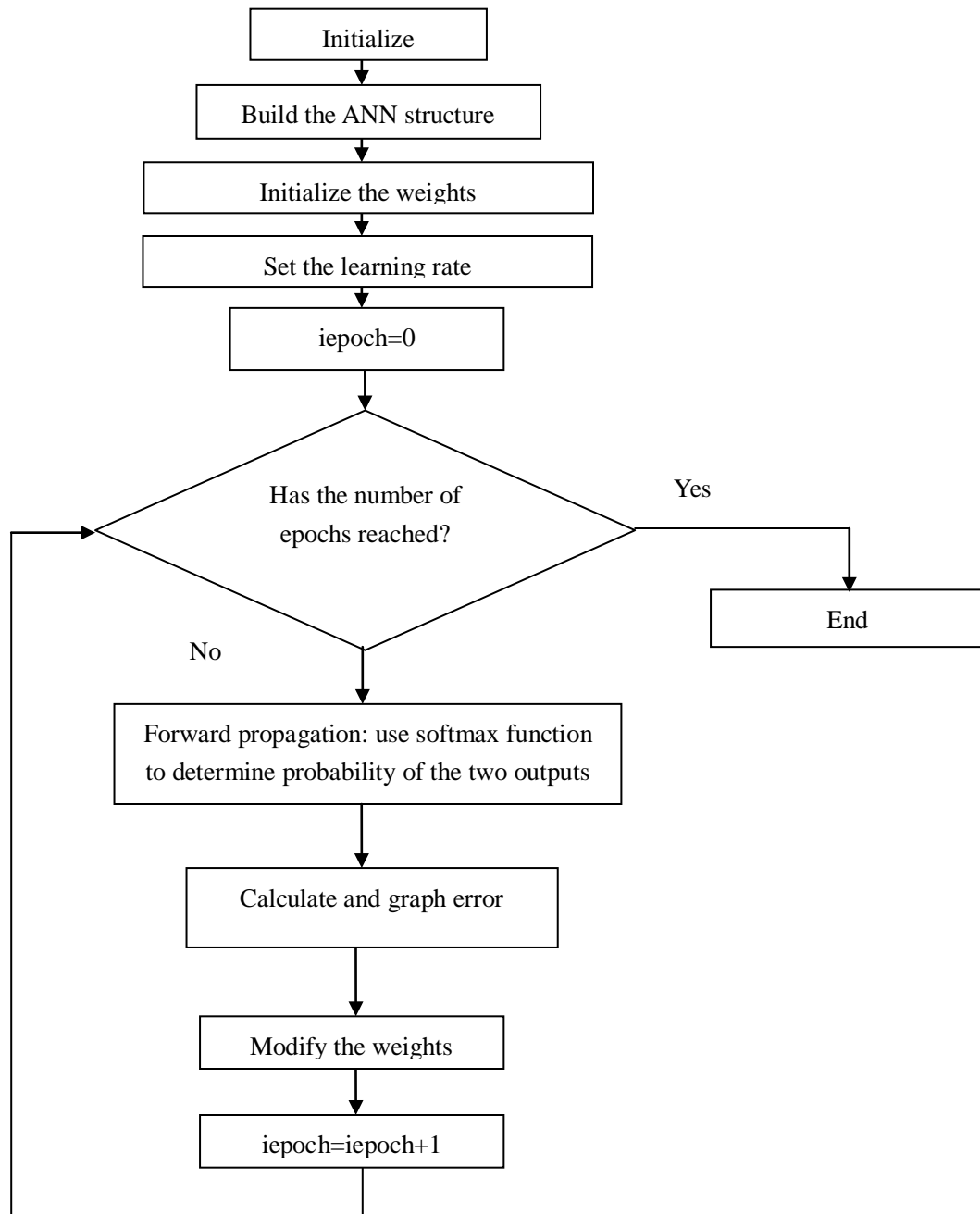


Fig. 11: Posteriori probability approach flowchart

Chapter 5 ANN Models for Two Problem Cases

5.1 Development of the Models for Two Problem Groups

Psychologists usually group social skill deficit and dysregulation of anger/aggression. They also group dysregulation of anger/aggression and substance abuse. This is because these two pairs of problems share some similarities in symptoms and make decision making by clinicians rather difficult.

In CTP database, there are 89 recorded cases that have both social skill deficit and dysregulation of anger/aggression assessments. Eighty percent of these cases (71 cases) were randomly selected for training the group ANN model and the other twenty percents was used for testing. Similarly, there are 97 recorded cases of that have both dysregulation of anger/aggression and substance abuse assessments. The artificial neural network for these two problems was trained on 77 randomly selected patient data and then they were tested on remaining data that the network had not been exposed.

The process for training the group- problem neural network model by backpropagation algorithm is similar to the model in Figure 8 except for having two neurons in the output layer as shown in Fig. 12. Each output neuron denotes one problem.

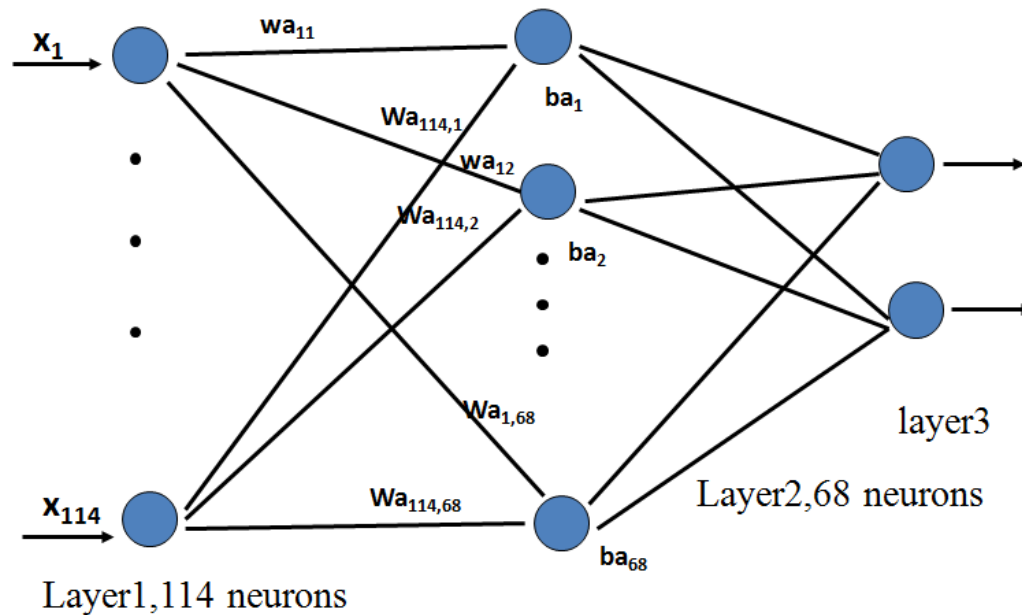


Fig. 12: Group problem neural network structure by backpropagation algorithm

5.2 Untitled Patient Assessments

ANN models will perform better and will provide useful information for unseen patterns if they are trained well with sufficient data. In CTP database, there were thirty seven patients with no recorded problem title assessment. Therefore, it is significant to train neural networks with the cases that have complete data and then testing them with cases that have missing patient assessment data.

For example, using the available 100 cases of dysregulation of anger/aggression assessments, a network was developed and trained to identify dysregulation of anger/aggression for the 37 patient cases. The other five problem titles for these 37 patients were identified the same way. The results are shown in Chapter 6. We hope,

the assessment predictions using these neural network models to assist clinicians in their diagnoses

Chapter 6 Results

6.1 Validations of the One-output Models

In order to validate the results obtained from different ANN models, two performance measure parameters are defined and calculated. These parameters are train accuracy and test accuracy. The high value of these parameters indicates that the model performs better,

$$\text{train accuracy} = \frac{\text{number of correct actual outputs}}{\text{total number of train samples}} \quad (22)$$

$$\text{test accuracy} = \frac{\text{number of correct actual outputs}}{\text{total number of test cases}} \quad (23)$$

Three single problem models for independent living skill deficit, disorder management deficit, and occupational skill deficit were trained and tested on the available data for these cases respectively. Since training samples and testing data were the same, the outputs obtained were also the same. As a result, each problem's train accuracy was equal to its test accuracy. Ten thousands epochs were run during the training process for every model. Fig. 13 shows the error versus the training epoch. The performance of these three models achieved 100% for the trained and test accuracies.

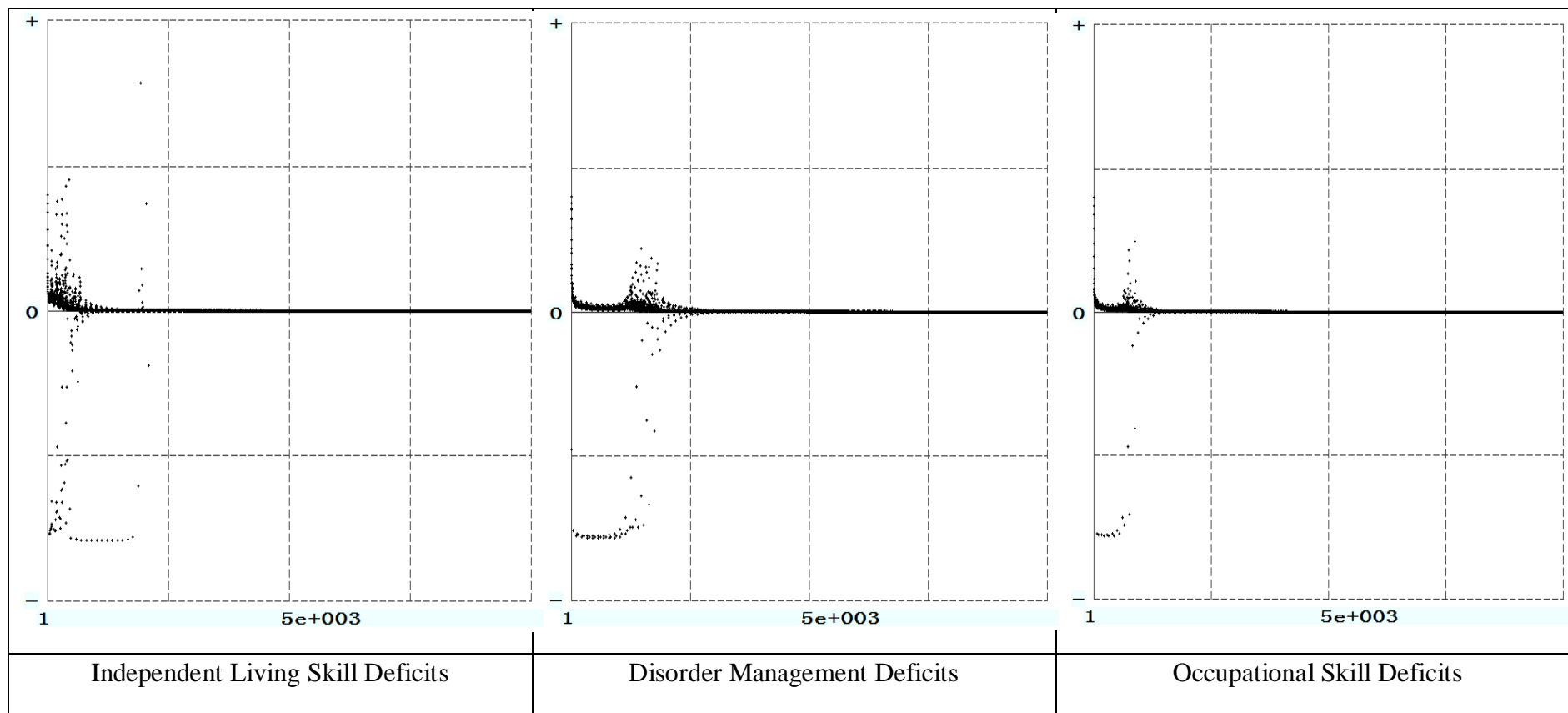


Fig. 13: Training process graph

One hundred patients were analyzed for dysregulation of anger/aggression using neural network models. The models were developed in DESIRE with backpropagation algorithm and posteriori probability approach, and in Matlab™ using Neural Network Toolbox. The samples used for training were selected randomly, as mentioned in Chapter #4. This implies that there were many possibilities for selecting training samples and testing data set. To try these different possibilities, ten replications of training and testing were performed. The train and test accuracies for each of these ten cases (replications) were determined using the three methods is shown in Table 7. Furthermore, the mean, maximum and minimum for each of these ten replications were calculated.



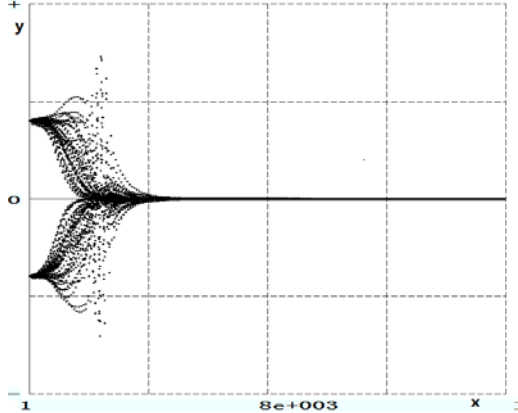
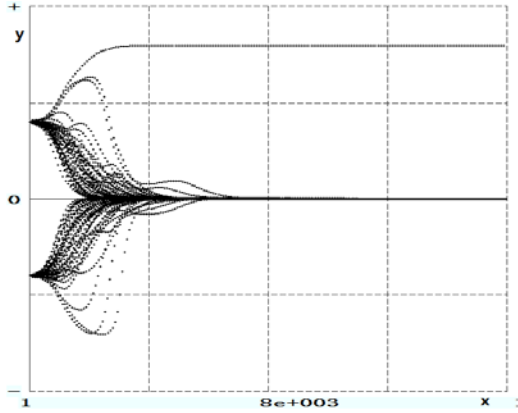
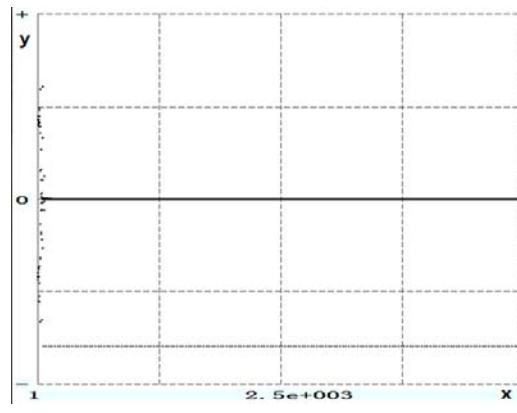
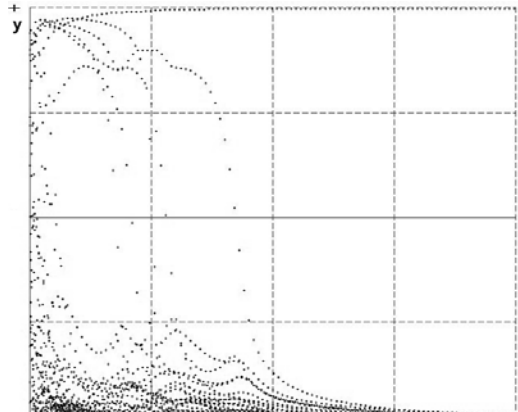
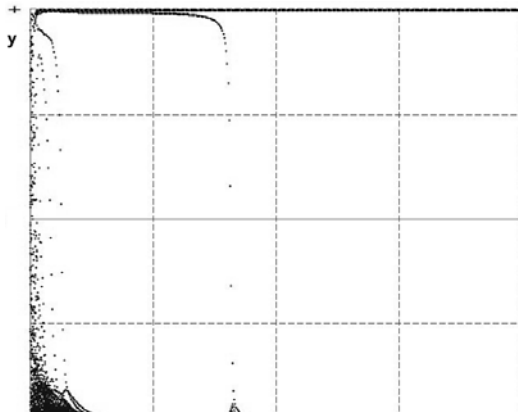
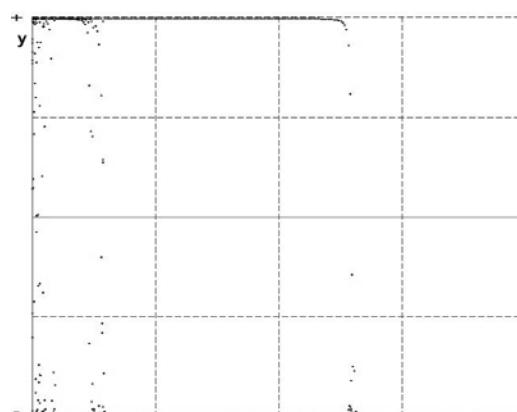
	Dysregulation of Anger/Aggression	Substance Abuse	Social Skill Deficits
Backpropagation Algorithm	 <p>Fig.14.1</p>	 <p>Fig.14.3</p>	 <p>Fig.14.5</p>
Posteriori Probability Approach	 <p>Fig.14.2</p>	 <p>Fig.14.4</p>	 <p>Fig.14.6</p>

Figure 14: Error (Y-axis) versus training epoch (X-axis) graph for Dysregulation of Anger/Aggression, Substance Abuse, and Social Skill Deficits. In the graphs the training patterns have been shown using dotted lines. As seen from the figures in first row (Backpropagation Algorithm), the error converged to zero for the majority of trained patterns using the backpropagation algorithm except for one in Figure 14.3. Thus, the train accuracy was less than 100%. The error for these three cases were similar to the one obtained using the posteriori probability approach.

The output values obtained from Matlab™ Neural Network Toolbox Model were real numbers (approaching to -1 and 1) compared with the desired binomial values of -1 or 1 obtained from the other two cases(backpropagation algorithm and posteriori probability approach) using DESIRE. The real numbers are the difference between actual output and desired output (called error performance). It is evaluated in Matlab™ models' training performance as a predefined parameter integrated in the Toolbox. The lower error performance means the better performance of neural network model in Matlab™, which is also equal to high value of train accuracies using DESIRE. The trained performance achieved using these three methods were up to the test accuracy 96.5%. It was 0.000314 in Matlab Neural Network Toolbox. Figures 14.1 and 14.2 display the convergence of training for backpropagation algorithm and the posteriori probability approach. The trained neural networks were tested on 20 cases. The average test accuracies and their range for the three methods as shown in Table 7 were 61.0% (50.0%-75.0%), 59.5% (45.0%-70.0%) and 53.5% (45.0%-70.0%) respectively.

dysregulation of anger/aggression Replications	backpropagation algorithm		posteriori probability approach		Matlab™ Neural Network Toolbox	
	train accuracies	test accuracies	train accuracies	test accuracies	error performance	test accuracies
1	97.5%	65.0%	97.5%	65.0%	0.00029	55.0%
2	97.5%	55.0%	100.0%	55.0%	0.000314	55.0%
3	100.0%	50.0%	97.5%	55.0%	0.000301	55.0%
4	97.5%	60.0%	97.5%	70.0%	0.000262	50.0%
5	100.0%	55.0%	96.5%	70.0%	0.000301	50.0%
6	100.0%	70.0%	97.5%	45.0%	0.00027	50.0%
7	100.0%	50.0%	98.5%	50.0%	0.000273	50.0%
8	100.0%	75.0%	97.5%	60.0%	0.000289	45.0%
9	96.5%	65.0%	100.0%	65.0%	0.000281	55.0%
10	97.5%	65.0%	98.5%	60.0%	0.000313	70.0%
upper bound(max)	100.0%	75.0%	100.0%	70.0%	0.000314	70.0%
lower bound(min)	96.5%	50.0%	96.5%	45.0%	0.000262	45.0%
mean	98.7%	61.0%	98.1%	59.5%	0.0002894	53.5%

Table 7: Neural network results for Dysregulation of Anger/Aggression

Analysis of substance abuse and social skill deficit models was carried on in the same way. The training graphs for each specific model are shown in Figures 14.3 through 14.6. The substance abuse models yielded mean and range of test accuracies in 10 replication of 56.77% (48.3% - 71.0%), 44.6% (29.0%-64.5%), and 54.1% (48.4%-61.0%) using the three methods, as represented in Table 8. On the average, the social skill deficit model correctly diagnosed 56.1%, 46.5%, and 51.4% of test data set for the three methods, as shown in Table 9. The ranges of these values were 45.3%-74.0%, 24.0%-57.3%, and 38.0%-60.0% respectively.

Substance Abuse	Backpropagation Algorithm		Posteriori Probability Approach		Matlab™ Neural Network Toolbox	
	train accuracies	test accuracies	train accuracies	test accuracies	error performance	test accuracies
1	98.5%	54.8%	98.5%	45.0%	0.000301	48.4%
2	97.5%	54.8%	96.5%	54.8%	0.000333	54.8%
3	100.0%	71.0%	100.0%	41.9%	0.00033	58.0%
4	97.5%	58.1%	96.5%	64.5%	0.000336	51.6%
5	97.5%	58.1%	96.5%	38.7%	0.000327	54.8%
6	98.5%	48.3%	97.5%	48.0%	0.000312	58.0%
7	97.5%	58.1%	96.5%	29.0%	0.000328	51.6%
8	97.5%	58.1%	98.5%	38.0%	0.0003	54.5%
9	98.5%	51.6%	98.5%	38.0%	0.000335	61.0%
10	98.5%	54.8%	96.5%	48.0%	0.000343	48.4%
Upper Bound(max)	100.0%	71.0%	100.0%	64.5%	0.000343	61.0%
Lower Bound(min)	97.5%	48.3%	96.5%	29.0%	0.0003	48.4%
Mean	98.2%	56.8%	97.6%	44.6%	0.0003245	54.1%

Table 8: Neural network results for Substance Abuse

Substance Abuse	Backpropagation Algorithm		Posteriori Probability Approach		Matlab™ Neural Network Toolbox	
	train accuracies	test accuracies	train accuracies	test accuracies	error performance	test accuracies
1	98.5%	54.8%	98.5%	45.0%	0.000301	48.4%
2	97.5%	54.8%	96.5%	54.8%	0.000333	54.8%
3	100.0%	71.0%	100.0%	41.9%	0.00033	58.0%
4	97.5%	58.1%	96.5%	64.5%	0.000336	51.6%
5	97.5%	58.1%	96.5%	38.7%	0.000327	54.8%
6	98.5%	48.3%	97.5%	48.0%	0.000312	58.0%
7	97.5%	58.1%	96.5%	29.0%	0.000328	51.6%
8	97.5%	58.1%	98.5%	38.0%	0.0003	54.5%
9	98.5%	51.6%	98.5%	38.0%	0.000335	61.0%
10	98.5%	54.8%	96.5%	48.0%	0.000343	48.4%
Upper Bound(max)	100.0%	71.0%	100.0%	64.5%	0.000343	61.0%
Lower Bound(min)	97.5%	48.3%	96.5%	29.0%	0.0003	48.4%
Mean	98.2%	56.8%	97.6%	44.6%	0.0003245	54.1%

Table 9: Neural Network Results for Social Skill Deficits

6.2 Validations of the Two-output Models

The train accuracy and test accuracy are calculated for the two outputs of these network models. The number of correct outputs that is used to determine the accuracy

parameters for these models are those that are correct for both outputs.

These models were trained on 80% of randomly selected data using backpropagation algorithm and were tested on the remaining 20% data. This process was replicated ten times. The average and range of train accuracy and test accuracy for combination of social skill deficit and dysregulation of anger/aggression models were 98.0% and (97.2%-98.6%) and 42.1% and (27.8%-55.6%) respectively as shown in Table 10.

Social skill deficit & Dysregulation of anger/aggression	Backpropagation Algorithm	
	Train Accuracies	Test Accuracies
1	98.6%	44.0%
2	98.6%	44.0%
3	98.6%	50.0%
4	98.6%	27.8%
5	98.6%	33.3%
6	97.2%	44.0%
7	97.2%	38.9%
8	97.2%	44.0%
9	98.6%	55.6%
10	97.2%	38.9%
Max	98.6%	55.6%
Min	97.2%	27.8%
Mean	98.0%	42.1%

Table 10: Neural network results for grouping Social Skill Deficits and Dysregulation of Anger/Aggression

The network that grouped dysregulation of anger/aggression and substance abuse learned accurately 92.2%-97.4% of the times with the average of 96%. The trained model, however, identified correctly 30.0%-45.0% of test data set with the average of

36%. These results are shown in Table 11.

Dysregulation of anger/aggression & Substance abuse	Backpropagation Algorithm	
	Train Accuracies	Test Accuracies
1	96.1%	30.0%
2	97.4%	45.0%
3	92.2%	35.0%
4	94.8%	35.0%
5	96.1%	40.0%
6	96.1%	30.0%
7	97.4%	35.0%
8	96.1%	40.0%
9	97.4%	35.0%
10	96.1%	35.0%
Max	97.4%	45.0%
Min	92.2%	30.0%
Mean	96.0%	36.0%

Table 11: Neural network results for grouping of Dysregulation of Anger/Aggression and Substance Abuse

6.3 Prediction of Other Cases

The six ANN models that were developed to predict the six single case problems (Section 4.2) were tested on a new set of 37 patients. The patients were 19 men and 18 women with mean age of 42.5 years and range 22-66 years. The predictions obtained from these models were compared with the opinion of clinical psychologists. The predictions for independent living skill deficits matched 81.1% (30 out of 37 cases) of those of the clinical psychologist's opinion, The prediction of the models for disorder management deficits, occupational skill deficits, dysregulation of anger/aggression, substance abuse, and social skill deficits were 35, 35, 25, 25 and 23 cases out of 37 cases respectively as detailed in Table 12.

Problem Titles	Number of <u>Consistent</u> Cases	Number of <u>Inconsistent</u> Cases
Independent Living Skill Deficits	30	7
Disorder Management Deficits	35	2
Occupational Skill Deficits	35	2
Dysregulation of Anger/Aggression	25	12
Substance Abuse	25	12
Social Skill Deficits	23	14

Table 12: Comparison for 37 untitled patients

Chapter 7 Conclusion and Future Work

The results presented in Chapter 6 confirm that artificial neural network models are suitable for analyzing patterns for psychological patients. These models have a strong learning ability with high train accuracy and a certain degree of prediction ability for testing (prediction). They can help health care clinicians in determining the likelihood of correct problem assessments.

In comparisons with posteriori probability approach and Matlab™ Neural Network Toolbox, the backpropagation algorithm models performed better on test accuracies for problem cases of sociobehavioral functioning. The results were also better than random statistical probability (0.5 for these cases although one or two replications fell below 0.5).

The interpretation of the low test accuracy (correct prediction of the patterns) is attributed to inadequate training of the network because of unavailability of enough training data. In order to make sure that this is the case, we ran the neural network model of dysregulation of anger/aggression with randomly selected 30, 40, 50, 60, 70, and 80 of the available 100 data samples. We used the remaining samples for testing. We repeated each case 10 times. For example, for training with 80 randomly selected samples, we repeated the process with ten different 80 randomly selected samples and recorded the test accuracies for each case. The results are shown in Table 13.

Number of Training Sample	Test Accuracies		
	Low Bound	Upper Bound	Mean
80	50.0%	75.0%	61.0%
70	46.7%	67.7%	56.7%
60	45.0%	67.5%	56.3%
50	44.0%	66.0%	53.4%
40	41.7%	65.0%	52.7%
30	41.0%	60.0%	52.3%

Table 13: Test accuracy tendency when training samples increased

As it is seen from Table 13, with increasing number of training samples, the mean test accuracies also increases (52.3% to 61.0%). Therefore, it is expected that if the networks are trained well, the test accuracies to increase substantially as shown in Table 13.

In group problem models with two outputs, the test accuracies (even their minimum value) were larger than the random statistical probability of 0.25 (two outputs with four possibilities). The mean test accuracy of 42.15% for the combined model of social skill deficit and dysregulation of anger/aggression was larger than that of 36.0% of the combined model of dysregulation of anger/aggression and substance abuse. This is because social skill deficit and dysregulation of anger/aggression are more related with each other than dysregulation of anger/aggression and substance abuse. Clinically, the 34 input variables of CTP database that were used in neural network models (Section 3.1) were direct indicators for social skill deficit and dysregulation of anger/aggression whereas indirect indicators for substance abuse. A psychological analogy may explain this case better. Assume a teacher of a morning class wants to predict whether students drank a lot the night before by their note

takings. The factors to be considered in this case (similar to the neural network inputs) might measure if the students look tired or fall asleep, if they look they are showered, whether they look at the teacher, whether they take notes, etc. These factors are not directly related to drinking. Drinking in this case is analogous to the substance abuse in the ANN model. At the same time, if an independent investigator was in class who was also taking the notes may predict if the students are paying attention. The note takings are direct measure of paying attention. Paying attention is more analogous to the social skill deficit and dysregulation of anger/aggression.

Furthermore, the artificial neural network may achieve higher test accuracy than linear statistical methods [24]. This is probably because they can take into account the complex nonlinear interrelationships among the variables which they are not easily incorporated into linear statistical methods. Nevertheless, it has some disadvantages because the network may end up learning at a “local minima⁵” which is not the optimal model but hard to detect and correct. Sometimes it may lead to over fitting situation of the training sample and poor generalization.

Finally, models presented in this study were developed to recognize the psychological patterns in CTP. The assessments (predictions) from these models could facilitate the accuracy in clinical decision making. Further clinical and laboratory studies are necessary to illustrate the full potential of artificial neural networks for predicting problem cases from multi rehabilitation units.

The problem cases presented here are sociobehavioal functioning. There is four

⁵ A local minima, also called a relative minimum, is a minimum within some neighborhood that need not be (but may be) a global minimum. [28]

other functioning at this level in the main hierarchy tree [4]. They are neurophysiological functioning, neurocognitive functioning, sociocognitive functioning, and socio-environmental problems. Research needs to be carried out for these cases to determine accuracy of neural network models for their identification.

In this research linear regression is used to determine missing data in the database as it was discussed in Chapter 3. However other methods in [29] and [30] may give more accurate values for them. Therefore, it is recommended that these methodologies also be tried.

The developed ANN models in this research can be only used for the CTP's patients using the parameters of the CTP database. Different rehabilitation units will probably have different parameter set and inter-relations. While the structure of the ANN models, developed here will remain the same, number of the inputs, the variables selected as input, and the performance obtained from them might be different. Therefore, it is recommended that similar ANN models for those databases also be tried and their results be compared against the ones obtained from this research for performance comparison and enhancement.

In the future clinical and laboratory studies, more detailed patients' information will be included to further improve or enhance the test accuracies. Not only does this possibility have important implication about artificial neural network assistance in medical diagnose, but it also suggest that such technologies, as reported here, may potentially be used in other new areas.



Bibliography

- [1]. Reggia JA, Tuhim S. “Computer-Assisted Medical Decision Making”.
Computers and Medicine. v.2. New York: Springer-Verlag, 1985
- [2]. The SOCRATES Decision Support System: Organizational structure and domain ontology, Fall 2008
- [3]. “Department of Health and Human Services: Statutory Compliance in Closing the Lincoln Regional Center Community Transition Program”, *Committee Report*, Vol. 16 , No. 1, November, 2009
- [4]. Jeffrey Stephen Poland, Mary E. Sullivan, William D. Spaulding, *Treatment and Rehabilitation of Severe Mental Illness(Hardcover)*, 2003
- [5]. M. Kantardzic. *Data Mining Concepts Models Methods and Algorithms*. Wiley Interscience Press, 2003
- [6]. Thearling.com, “Information about data mining and analytic technologies”, <http://www.thearling.com/text/dmwhite/dmwhite.htm>, 2010.6
- [7]. A.C.M. Fong, S.C.Hui, and G.Jha, “Data Mining for Decision Support”, *IT Professional*, Volume 4, Issue 2, Pages: 9 – 17, March 2002
- [8]. William G.Baxt, “Use of an Artificial Neural Network for the Diagnosis of Myocardial Infarction”, *Annals of Internal Medicine*, Volume115, 1991
- [9]. R.L. Kennedy, et al, “An artificial neural network system for diagnosis of acute myocardial infarction (AMI) in the accident and emergency department:

- evaluation and comparison with serum myoglobin measurement ”, *Computer Methods and Programs in Biomedicine*, 1997
- [10]. A.A. El-Solh, et al, “Validity of an Artificial Neural Network in Predicting Discharge Destination From a Postacute Geriatric Rehabilitation Unit”, *Arch Phys Med Rehabil*, Vol. 81, 2000
- [11]. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Neural_networks, http://en.wikipedia.org/wiki/Sigmoid_function, 2009.11
- [12]. Raúl Rojas, *Neural Networks - A Systematic Introduction*, Springer Verlag, 1996.
- [13]. Granino A.Korn, *Neural Networks and Fuzzy-Logic Control on Personal Computers and Workstations*, The MIT Press, Cambridge, Massachusetts, 1995
- [14]. José C. Príncipe, Neil R. Euliano, W. Curt Lefebvre, *Neural and adaptive systems: fundamentals through simulations*, John Wiley, 2000.
- [15]. Wikipedia, the free encyclopedia, definition of Posterior_probability: http://en.wikipedia.org/wiki/Posterior_probability, 2010.6
- [16]. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Oxford University Press, 1995
- [17]. OPEN DESIRE: <http://sites.google.com/site/gatmkorn/>, 2010.6
- [18]. Granino A.Korn, *Advance Dynamic-System Simulation*, a Johon Wiley & Sons, Inc, 2007
- [19]. Matlab™ Help Documents for Neural Network Toolbox.
- [20]. P. Roth. “Missing data: A conceptual review for applied psychologists”. *Personnel Psychology*, 47(3):537–560, 1994.

- [21]. Sarle WS. "Neural networks and statistical models". *Proceedings of 19th annual SAS User Group International Conference*: 1994 Apr: Cary(NC). Cary(NC).SAS Institute;1996. P1-13
- [22]. Blum, A., *Neural Networks in C++*, NY: Wiley, 1992
- [23]. Faqs.org: <http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-10.html>, 2010
- [24]. H.P. Selker, et.al. "A comparison of mathematical predictive methods for medical diagnosis: identifying acute cardiac ischemia among emergency department patients", *J. Investing. Med.*43 (1995) 468-476
- [25]. R.F. Harrison, R.L. Kennedy, "Artificial Neural Network Models for Prediction of Acute Coronary Syndromes Using Clinical Data From the Time of Presentation", *Annals of Emergency Medicine*. 2005 Nov; 46(5):431-9.
- [26]. Medical Dictionary:<http://medical-dictionary.thefreedictionary.com/biobehavioral>, 2010.8
- [27]. Wikipedia, the free encyclopedia, definition of Linear_discriminant_analysis: http://en.wikipedia.org/wiki/Linear_discriminant_analysis, 2010.8
- [28]. Wolfram Mathworld: <http://mathworld.wolfram.com/LocalMinimum.html>, 2010.9
- [29]. Little, R.J.A., Rubin, D.B. *Statistical analysis with missing data*. Wiley: Hoboken, NJ. 2002
- [30]. G. Molenberghs, M. Kenward, *Missing Data in Clinical Studies*, John Wiley and Sons, 2007

Appendix A

Variable Description

Variable Name	Variable Label	Variable Codes	Entity/Functional Domain	Why collected?
id				
psybin3	Antipsychotic Med 1 Month 1-90		Neurophysiological	Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications
depbin3	Antidepressant Med 1 Month 1-90		Neurophysiological	Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications
anxbin3	Anxiolytic Med 1 Month 1-90		Neurophysiological	Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications
chobin3	Anticholinergic Med 1 Month 1-90		Neurophysiological	Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications
othbin3	Other Med 1 Month 1-90		Neurophysiological	Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications

legalMHBbin	Legal Status	1=MHB (Mental Health Board)		Indicates type of commitment
legalVpGbin		2=VpG (Voluntary per Guardian)		
legalNRRIbin		3=NRRI (Not Responsible by Reason of Insanity)		
legalvolbin		4=Vol. (Voluntary)		
gender	gender	1 male 2female		Demographic grouping variable
age	Age at Admission	#		Demographic variable
high_ed	Years of Education	##		Quick index of intelligence, could indicate premorbid functioning
prev_hos	Number of Previous Hosps (before CTP, not including CTP)	#		Indicates treatment non-response/non-adherence
age_onse	Age of Onset	##		Indicates duration of illness
occ_func	Highest Occupational Functioning	1=Professional, technical, manager, official, proprietor, clerical, sales, student		Could indicate premorbid (occupational) functioning
		2=skilled worker-craftsman, foreman		
		3=semi-skilled-farmer, service worker, operator		
		4=laborer		
		5=unemployed		
ilsipe2	ILSI Personal Management Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability

ilsihy2	ILSI Hygiene & Grooming Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsicl1-17	ILSI Clothing Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsiba1-17	ILSI Basic Skills Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsiin1-17	ILSI Interpersonal Skills Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsiho1-17	ILSI Home Maintenance Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsimo1-17	ILSI Money Management Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsico1-17	ILSI Cooking Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsire1-17	ILSI Resource Utilization Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
ilsige1-17	ILSI General Occupational Functioning Time 1-17		Sociobehavioral; subdomain: skills	Indicates Independent Living Skill Ability
asama1-17	COGLAB Asarnow Masking Time 1-17		Neurocognitive	COGLAB measures attention and executive function
asahit1-17	COGLAB Asarnow Hits Time 1-17		Neurocognitive	COGLAB measures attention and executive function
asafa1-17	COGLAB Asarnow False Alarms Time 1-17		Neurocognitive	COGLAB measures attention and executive function
wcstco1-17	COGLAB WCST Correct Time 1-17		Neurocognitive	COGLAB measures attention and executive function

wcstpe1-17	COGLAB WCST Perseverative Errors Time 1-17		Neurocognitive	COGLAB measures attention and executive function
wcstre1-17	COGLAB WCST Random Errors Time 1-17		Neurocognitive	COGLAB measures attention and executive function
wcstc1-17	COGLAB WCST Consolidation Time 1-17		Neurocognitive	COGLAB measures attention and executive function
bdi_tota	BDI (Beck Depression Inventory) Total Score at CTP Admission	##	Neurophysiological, sociocognitive, sociobehavioral	Indicates severity of depression
bhs_tota	BHS (Beck Hopelessness Scale) Total Score at CTP Admission	##	Neurophysiological, sociocognitive, sociobehavioral	Indicates severity of depression/hopelessnes s

Appendix B

Categorization of variables

Variable Name	#	discrete category
psybin3	1	psybin3
depbin3	2	depbin3
anxbin3	3	anxbin3
chobin3	4	chobin3
othbin3	5	othbin3
legalMHBbin	6	legalMHBbin
legalVpGbin	7	legalVpGbin
legalNRRlbin	8	legalNRRlbin
legalvolbin	9	legalvolbin
gender	10	gender
age	11	agea 19-24
	12	ageb 25-39
	13	agec 40-55

	14	aged 56 or older
high_ed	15	high_eda 12 or less
	16	high_edb 13-16
	17	high_edc 17 or more
prev_hos	18	prev_hosa 0-10
	19	prev_hosb 11-20
	20	pre_hosc 21 or more
age_onse	21	age_onsea less than 16
	22	age_onseb 17-20
	23	age_onsec 21-25
	24	age_onsed 26-30
	25	age_onsee 31-35
	26	age_onsef 36-45
	27	age_onsef 46 or older
occ_func	28	occ_funca 1=Professional, technical, manager, official, proprietor, clerical, sales, student
	29	occ_funcb 2=skilled worker-craftsman, foreman
	30	occ_funcc 3=semi-skilled- farmer, service worker, operator
	31	occ_funcd 4=laborer
	32	occ_funce 5=unemployed
ilsipe2	33	ilsipe2a 0-5
	34	ilsipe2b 6-11
	35	ilsipe2c 12-14
	36	ilsipe2d 15-18
ilsihy2	37	ilsihy2a 0-8
	38	ilsihy2b 9-17
	39	ilsihy2c 18-22
	40	ilsihy2d 23-27
ilsicl2	41	ilsicl2a 0-4
	42	ilsicl2b 5-9
	43	ilsicl2c 10-12
	44	ilsicl2d 13-15
ilsiba2	45	ilsiba2a 0-5
	46	ilsiba2b 6-11
	47	ilsiba2c 12-14
	48	ilsiba2d 15-18
ilsiin2	49	ilsiin2a 0-16
	50	ilsiin2b 17-33
	51	ilsiin2c 34-42
	52	ilsiin2d 43-51
ilsiho2	53	ilsiho2a 0-5
	54	ilsiho2b 6-11

	55	ilsiho2c 12-14
	56	ilsiho2d 15-18
ilsimo2	57	ilsimo2a 0-5
	58	ilsimo2b 6-11
	59	ilsimo2c 12-14
	60	ilsimo2d 15-18
ilsico2	61	ilsico2a 0-7
	62	ilsico2b 8-15
	63	ilsico2c 16-19
	64	ilsico2d 20-24
ilsire2	65	ilsire2a 0-5
	66	ilsire2b 6-11
	67	ilsire2c 12-14
	68	ilsire2d 15-18
ilsige2	69	ilsige2a 0-10
	70	ilsige2b 11-20
	71	ilsige2c 21-27
	72	ilsige2d 28-33
asama2	73	asama2a less than 25
	74	asama2b 25-34
	75	asama2c 35 or more
asahit2	76	asahit2a 0-5
	77	asahit2b 6-10
	78	asahit2c 11-15
	79	asahit2d 16-20
	80	asahit2e 21-25
	81	asahit2f 26-30
asafa2	82	asafa2a 0
	83	asafa2b 1-5
	84	asafa2c 6-10
	85	asafa2d 11 or more
wcstco2	86	wcstco2a less than 20
	87	wcstco2b 20-24
	88	wcstco2c 25 or more
wcstpe2	89	wcstpe2a 5 or less
	90	wcstpe2b 6-15
	91	wcstpe2c 16-25
	92	wcstpe2d 26-34
	93	wcstpe2e 35 or more
wcstre2	94	wcstre2a 0-15
	95	wcstre2b 16-25
	96	wcstre2c 26-35
	97	wcstre2d 36-45

	98	westre2e 46-55
	99	westre2f 56 or more
wstc2	100	wstc2a 0
	101	wstc2b 1
	102	wstc2c 2
	103	wstc2d 3
	104	wstc2e 4
	105	wstc2f 5
	106	wstc2g 6 or more
bdi_tota	107	bdi_totaa 0-13
	108	bdi_totab 14-19
	109	bdi_totac 20-28
	110	bdi_totad 29 or more
bhs_tota	111	bhs_totaa 0-3
	112	bhs_totab 4-8
	113	bhs_totac 9-14
	114	bhs_totad 15 or more

Appendix C

DESIRE program for describing the Dysregulation of Anger/Aggression model using

Backpropagation Algorithm

```

210 --          THREE-LAYER NETWORK, BACKPROPAGATION
220 -----
230 ntrain=80 | ntest=20 --80 patterns for training and 20 patterns for test
240 M=80 | --          patterns per updating batch
250 n2=68 | --          number of hidden-layer neurons
260 --
270 ARRAY layer1[114],layer2[n2],layer3[1] – input layer has 114 neurons
290 ARRAY w12[n2,114],w23[1,n2],bias2[n2],bias3[1] –actual weights and bias
300 ARRAY W12[n2,114],W23[1,n2],BIAS2[n2],BIAS3[1] | -- auxiliary weights
and bias
310 ARRAY INPUT[80,114],TARGET[80,1],INPUTTEST[20,114]
320 ARRAY error[1],delta3[1],delta2[n2],error2[1]
330 ARRAY ori[100,115],tartest[ntest,1],err[ntest] – ori matrix stores data
340 -- read raw data into the array ori[100,115]
350 connect 'e:\anndata\0526\psy_100_114_1-1.txt' as input 3 –import data into
matrix
360 for i=1 to 100
370   for j=1 to 115

```

```

380     input #3,ori[i,j]
390     if eof then go to 303 | else proceed
400     next
410 next
420 disconnect 3
430 -- random selection for training set INPUT[80,114] and test set
INPUTTEST[20,114]
440 -- produce 52 random number for random selection
450 n=52
460 ARRAY a3[n]
470 i=0
480 repeat
490   a1=(ran()+1)/2
500   a2=round(a1*n)
510   if a2<>0 then i=1 | a3[1]=a2 | else proceed
520   until i=1
530 repeat
540   a1=(ran()+1)/2
550   a2=round(a1*n)
560   -- write a1,a2
570   f=0
580   for j=1 to i
590     if a2=a3[j] then f=1 | else proceed
600   next
610   if a2=0 then f=1 | else proceed
620   if f=0 then i=i+1 | a3[i]=a2 | -- write a3
630   else proceed
640   until i=n
650 ----
660 -- produce 48 random number for random selection
670 n=48
680 ARRAY a4[n]
690 i=0
700 repeat
710   a1=(ran()+1)/2
720   a2=round(a1*n)
730   if a2<>0 then i=1 | a4[1]=a2 | else proceed
740   until i=1
750 repeat
760   a1=(ran()+1)/2
770   a2=round(a1*n)
780   f=0
790   for j=1 to i
800     if a2=a4[j] then f=1 | else proceed

```



```

810     next
820     if a2=0 then f=1 | else proceed
830     if f=0 then i=i+1 | a4[i]=a2 | -- write a4
840     else proceed
850     until i=n
860     ----
870     for i=1 to 40
880     tmp1=a3[i]
890     tmp2=a4[i]+52
900     for j=1 to 114
910     INPUT[i*2-1,j]=ori[tmp1,j]
920     INPUT[i*2,j]=ori[tmp2,j]
930     next
940     TARGET[i*2-1,1]=ori[tmp1,115]
950     TARGET[i*2,1]=ori[tmp2,115]
960     next
970     for i=1 to 12
980     tmp2=i+40
990     tmp3=a3[tmp2]
1000     for j=1 to 114
1010     INPUTTEST[i,j]=ori[tmp3,j]
1020     next
1030     tartest[i,1]=ori[tmp3,115]
1040     next
1050     for i=13 to 20
1060     tmp2=i+28
1070     tmp3=a4[tmp2]+52
1080     for j=1 to 114
1090     INPUTTEST[i,j]=ori[tmp3,j]
1100     next
1110     tartest[i,1]=ori[tmp3,115]
1120     next
1130     --
1140     -- random selection ends
1150     -- small random initial weights to "break symmetry"
1160     --
1170     -- Initialize the random weights
1180     for i=1 to n2
1190     for j=1 to 114
1200     W12[i,j]=0.02*ran()
1210     next
1220     W23[1,i]=0.03*ran()
1230     next
1240     -----

```

```

1260 -----
1270 -- set experiment parameters
1280 lrate2=0.002 | lrate3=0.002 --set learning rate
1290 r2=0.001 | r3=0.0005 --set momentum
1310 NN=16000 | -- 16000 trials
1320 -----
1330 display N15 | display C7 | -- set display colors
1340 display R | -- and thick dots
1350 -----
1370 drun | -- make a simulation run (NN trials)
1390 -----
1400 write 'type go for a recall test'
1410 -- STOP
1420 -----
1430 label jjj | -- try al
1440 NNold=NN | NN=20 | -- save old NN for new runs
1450 lrate2=0 | lrate3=0 | r2=0 | r3=0
1460 t=1 | TMAX=NN-1
1470 -----
1480 -----
1490 drun RECALL
1500 cnt=0 -- calculate test accuracy
1510 for i=1 to ntest
1520 if err[i]=0 then cnt=cnt+1 | else proceed
1530 next
1540 rate=cnt/ntest
1550 write rate | -- write test accuracy
1560 NN=NNold | TMAX=NN-1 | -- restore NN
1570 -----
1580 DYNAMIC
1590 -----
1600 -- weight-learning run
1610 iRow=t | Vector layer1=INPUT#
1620 Vector layer2=tanh(w12*layer1+bias2)
1630 Vector layer3=tanh(w23*layer2+bias3)
1640 out1=layer3[1]
1650 -----
1660 Vector error=TARGET#-layer3 | -- backpropagation
1670 ----- Vector delta3=e
1680 Vector delta3=error*(1-layer3*layer3)
1690 Vector delta2=w23%*delta3*(1-layer2^2)
1700 --
1710 -- accumulate temporary weights, biases for M trials
1720 --

```

```

1730 DELTA W12=lr2*delta2*layer1+r2*w12
1740 DELTA W23=lr3*delta3*layer2+r3*w23
1750 Vectr delta BIAS2=lr2*delta2+r2*bias2
1760 Vectr delta BIAS3=lr3*delta3+r3*bias3
1770 --
1800 ERROR=2*error[1] | -- offset error display
1810 --
1820 -----
1830 SAMPLE M | -- update actual weights, biases only ...
1840 ----- ... after M patterns!
1870 -----
1880 MATRIX w12=W12 | MATRIX w23=W23 | -- update
1890 Vector bias2=BIAS2 | Vector bias3=BIAS3
1900 -----
1910 dispt ERROR | -- display error measure
1920 -----
1930 label RECALL
1940 iRow=t | Vector layer1=INPUTTEST# | -- recall run
1950 Vector layer2=sat(w12*layer1+bias2)
1960 Vector layer3=sat(w23*layer2+bias3)
1970 -----
1980 out1=layer3[1]
1990 Vector error2=tartest#-layer3 | out2=error2[1]
2000 type out1,out2 | -- test for correct truth table
2010 store err=error2[1]

```

Appendix D

DESIRE program for describing the Dysregulation of Anger/Aggression model using

Posteriori Probability Approach

```

210 -- estimates a posteriori probabilities
220 -----
230 display N14 | display C7 | -- set display colors
240 display R | -- and thick dots
250 nx=114 | N=2 | -- 114 nodes in input layer and 2 nodes in output layer
260 ntrain=80 | ntest=20 -- --80 patterns for training and 20 patterns for test
270 -- build the ANN model
280 ARRAY x[nx]

```



```

290 ARRAY INPUT[ntrain,nx],TARGET[N,N],INPUTTEST[ntest,nx]
300 ARRAY tar[ntrain,1],tartest[ntest,1] – set training target matrix and test target
matrix
310 ARRAY v[N],q[N],y[nx],yy[nx],W[N,nx]
320 ARRAY error[N],error2[N]
330 ARRAY v1[N],output[ntest],ori[100,115]
340 --
350 for aa=1 to 4
360   for i=1 to N | for k=1 to nx | -- initialize W
370     W[i,k]=ran() | next | next
380   next
390 --
400 -- read raw data into the array ori[100,115]
410 connect 'e:\anndata\0526\psy_100_114_1-1.txt' as input 3
420 for i=1 to 100
430   for j=1 to 115
440     input #3,ori[i,j]
450     if eof then go to 303 | else proceed
460     next
470   next
480 disconnect 3
490 -- random selection for training set INPUT[80,114] and test set
INPUTTEST[20,114]
500 -- random 52
510 n=52
520 ARRAY a3[n]
530 i=0
540 repeat
550   a1=(ran()+1)/2
560   a2=round(a1*n)
570   if a2<>0 then i=1 | a3[1]=a2 | else proceed
580   until i=1
590 repeat
600   a1=(ran()+1)/2
610   a2=round(a1*n)
620   -- write a1,a2
630   f=0
640   for j=1 to i
650     if a2=a3[j] then f=1 | else proceed
660     next
670   if a2=0 then f=1 | else proceed
680   if f=0 then i=i+1 | a3[i]=a2 | -- write a3
690   else proceed
700   until i=n

```

```

710 -----
720 -- produce 52 random number for random selection
730 n=48
740 ARRAY a4[n]
750 i=0
760 repeat
770   a1=(ran()+1)/2
780   a2=round(a1*n)
790   if a2<>0 then i=1 | a4[1]=a2 | else proceed
800   until i=1
810 repeat
820   a1=(ran()+1)/2
830   a2=round(a1*n)
840   -- write a1,a2
850   f=0
860   for j=1 to i
870     if a2=a4[j] then f=1 | else proceed
880   next
890   if a2=0 then f=1 | else proceed
900   if f=0 then i=i+1 | a4[i]=a2 | -- write a4
910   else proceed
920   until i=n
930 -----
940 for i=1 to 40
950   tmp1=a3[i]
960   tmp2=a4[i]+52
970   for j=1 to 114
980     INPUT[i*2-1,j]=ori[tmp1,j]
990     INPUT[i*2,j]=ori[tmp2,j]
1000    next
1010    next
1020   for i=1 to 12
1030     tmp2=i+40
1040     tmp3=a3[tmp2]
1050     for j=1 to 114
1060       INPUTTEST[i,j]=ori[tmp3,j]
1070     next
1080   next
1090   for i=13 to 20
1100     tmp2=i+28
1110     tmp3=a4[tmp2]+52
1120     for j=1 to 114
1130       INPUTTEST[i,j]=ori[tmp3,j]
1140     next

```

```

1150     next
1160  --
1170  -- random selection ends
1180  MATRIX TARGET=1 |  -- (binary-classifier rows)
1190  -----
1200  --                               set experiment parameters
1210  lrate=1.75 |  c=0.2
1220  lrate1=lrate*c
1230  NN=8000 |  -- 8000 trials
1240  -----
1250  drun
1260  write 'type go for successive recall runs' |  -- STOP
1270  display F
1280  -----
1290     label recall
1300  t=1 |  NN=2 |  TMAX=NN-1 |  restore |  -- reset the read point
1310  for i=1 to ntest
1320     drun RECALL
1330     ii=i+1 |  -- test pattern output to the screen
1340     write 'current test vector',ii
1350     for jj=1 to N
1360         if q[jj]=1 then output[i]=jj |  write output[i], 'is estimated _
to be the best match train vector' |  else proceed
1370     next
1380     write '-----'
1390     next
1400  cnt=0 |  -- counter: count the correct outputs
1410  for i=1 to 11
1420     if output[i]=1 then cnt=cnt+1 |  else proceed
1430     next
1440  for i=12 to 19
1450     if output[i]=2 then cnt=cnt+1 |  else proceed
1460     next
1470  if output[20]=1 then cnt=cnt+1 |  else proceed
1480  rate=cnt/ntest
1490  write rate |  -- outprint test accuracy
1500  connect 'e:\anndata\ssd_output_time5.txt' as output 3
1510  for i=1 to ntest
1520     write #3,output[i]
1530     next
1540  disconnect 3
1550  NN=8000 |  TMAX=NN-1 |  restore
1560  -----
1570  DYNAMIC

```

```

1580 -----
1590 iRow=t | Vector x=INPUT#
1600 Vector v=exp(c*W*x) | DOT vsum=v*1 --use softmax function
1610 Vector v=v/vsum | -- probability estimate
1620 Vector error=TARGET#-v | -- calculate error
1630 Vector q^=v | Vector q=swtch(q) | -- enhance the larger to 1 and reduce
the smaller to 0
1640 Vector error2=TARGET#-q |
1650 -----
1660 Vector v1=error*v*(1-v)
1670 DELTA W=lrate1*v1*x | -- update weights
1680 --
1690 DOT enormsq=error*error
1700 ENORMSQ=enormsq |
1710 dispt ENORMSQ
1720 -----
1730 label RECALL
1740 iRow=t | Vector x=INPUTTEST# | -- testing process forwards
propagation
1750 Vector v=exp(c*W*x) | DOT vsum=v*1
1760 Vector v=v/vsum | -- probability estimate
1770 Vector q^=v | Vector q=swtch(q) | -- binary selector

```

Appendix E

DESIRE program for describing combination of social skill deficit and dysregulation

of anger/aggression model

```

210 n2=68 --68 neurons in hidden layer
250 ntrain=71 | ntest=18 -- 71 patterns for training and 18 patterns for test
260 -- build ANN model
270 ARRAY layer1[114],layer2[n2],layer3[2] -- 114 neuron in input layer
290 ARRAY w12[n2,114],w23[2,n2],bias2[n2],bias3[2]
300 ARRAY W12[n2,114],W23[2,n2],BIAS2[n2],BIAS3[2]|-- auxiliary weights W
and bias B
310 ARRAY INPUT[ntrain,114],TARGET[ntrain,2],INPUTTEST[ntest,114]
320 ARRAY tartest[ntest,2],ori[89,116],errssd[ntest],errpsy[ntest]
330 ARRAY error[2],delta3[2],delta2[n2],error2[2]
340 n=89
350 ARRAY a3[n]
360 -- import data into matrix ori[89,116]

```

```

370 connect 'e:\anndata\0505\two_89_114.txt' as input 3
380 for i=1 to 89
390   for j=1 to 116
400     input #3,ori[i,j]
410     if eof then go to 303 | else proceed
420     next
430   next
440 disconnect 3
450 i=0 |-- random selection for training set and test set
460 repeat
470   a1=(ran()+1)/2
480   a2=round(a1*n)
490   if a2<>0 then i=1 | a3[1]=a2 | else proceed
500   until i=1
510 repeat
520   a1=(ran()+1)/2
530   a2=round(a1*n)
540   f=0
550   for j=1 to i
560     if a2=a3[j] then f=1 | else proceed
570     next
580   if a2=0 then f=1 | else proceed
590   if f=0 then i=i+1 | a3[i]=a2 | -- write a3
600   else proceed
610   until i=n
620 -----
630 for i=1 to ntrain
640   tmp1=a3[i]
650   for j=1 to 114
660     INPUT[i,j]=ori[tmp1,j]
670     next
680   TARGET[i,1]=ori[tmp1,115]
690   TARGET[i,2]=ori[tmp1,116]
700   next|-- random selection ends
710 -- small random initial weights to "break symmetry"
720 -- Initialize the random weights
730 for i=1 to n2
740   for j=1 to 114
750     W12[i,j]=0.02*ran()
760     next
770   for j=1 to 2
780     W23[j,i]=0.03*ran()
790     next
800   next

```



```

1300 DYNAMIC
1310 -----
1320 --                                weight-learning run
1330 iRow=t | Vector layer1=INPUT# |-- forwards propagation
1340 Vector layer2=tanh(w12*layer1+bias2)
1350 Vector layer3=tanh(w23*layer2+bias3)
1360 out1=layer3[1]
1370 -----
1380 Vector error=TARGET#-layer3 | --      backpropagation
1400 ----- Vector delta3=e
1410 Vector delta3=error*(1-layer3*layer3)
1420 Vector delta2=w23%*delta3*(1-layer2^2)
1430 --
1440 --
1450 --
1460 DELTA W12=lrata2*delta2*layer1+r2*w12
1470 DELTA W23=lrata3*delta3*layer2+r3*w23
1480 Vectr delta BIAS2=lrata2*delta2+r2*bias2
1490 Vectr delta BIAS3=lrata3*delta3+r3*bias3
1500 --
1530 ERROR=2*(error[1]+error[2]) | --      offset error display
1540 --
1570 -----
1580 MATRIX w12=W12 | MATRIX w23=W23 | --                                update
1590 Vector bias2=BIAS2 | Vector bias3=BIAS3
1600 -----
1620 dispt ERROR | --                                display error measure
1630 -----
1640 label RECALL
1650 iRow=t | Vector layer1=INPUTTEST# | -- test process          recall
run
1660 Vector layer2=sat(w12*layer1+bias2)
1670 Vector layer3=sat(w23*layer2+bias3)
1680 -----
1690 Vector error2=tartest#-layer3
1700 out1=layer3[1]
1710 out2=layer3[2]
1720 store errsd=error2[1] |--      store for result calculation
1730 store errpsy=error2[2]

```