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

http://digitalcommons.unl.edu/elecengtheses/11
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
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 biobehavioral \(^1\) 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].
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, statists, computer and informatics. It can help the treatment team in decision making process. As mentioned earlier, artificial neural network 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):

$$y = f(x)$$  \hspace{1cm} (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 sensitivity\(^2\) and 12\% higher diagnostic specificity\(^3\).

\(^2\) 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 analysis\(^4\) (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 Matlab™ Neural Network Toolbox.

Data preparations are explained in Chapter 3 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.

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.

$$net_k = \sum_{i=1}^{i=3} x_i w_{ki} + b_k$$

$$y_k = f(net_k)$$

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 $f(net_k)$ is given by [11]:

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

where, $a$ specifies the slope of the curve.

Similarly, the derivative of the hyperbolic tangent sigmoid function

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

\[
e_j(n) = y_j(n) - t_j
\]

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:

\[
E(n) = \frac{1}{2} \sum_{i=1}^{n} e_j(n)^2
\]

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
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):

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

where $\Delta w_{ji}(n)$ denotes the correction to the weight $w_{ji}$ from neuron i to neuron j at $n^{th}$ trial. $\eta$ 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:

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

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n); \quad \frac{\partial e_j(n)}{\partial f_j(n)} = -1; \quad \frac{\partial f_j(n)}{\partial net_j(n)} = f_j'(n); \quad \frac{\partial net_j(n)}{\partial w_{ji}(n)} = x_i(n); \quad (7)$$
Therefore, \[ \Delta w_{ji}(n) = \eta e_j(n)f'_j(n)x_i(n) \] (8)

In \((n+1)^{th}\) 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, \( \alpha \), to the delta rule [5].

\[ \Delta w_{ji}(n) = \eta e_j(n)f'_j(n)x_i(n) + \alpha \Delta w_{ji}(n-1) \] (10)

Where \( \alpha \) 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:

\[ e_j(n) = y_j(n) - t_i \] (11)

\[ \Delta w_{ji}(n) = \eta e_j(n)f'_j(n)x_i(n) + \alpha \Delta w_{ji}(n-1) \] (12)

Accumulating auxiliary weight \( W \) over N trials will yield,
\[ W_{\mu}(n+1) = W_{\mu}(n) + \Delta w_{\mu}(n) \]  

(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:

\[ w_{\mu}(N) = W_{\mu}(N) \]  

(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:

$$y_k = \frac{\exp(net_k)}{\sum_j \exp(net_j)}$$

(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):

$$E = \frac{1}{2} \sum_k \int \int (y_k(x,w) - t_k)^2 p(t_k, x) dt_k dx$$

(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):

$$(y_k(x,w) - t_k)^2 = (y_k(x,w) - <t_k|x>) + <t_k|x> - t_k)^2$$

(17)

where $<t_k|x>$ is the conditional average given by $<t|x> = \int t_k p(t_k|x) dt_k$.

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

$$(y_k(x,w) - t_k)^2 = (y_k(x,w) - <t_k|x>)^2 + 2(y_k(x,w) - <t_k|x>)(<t_k|x> - t_k) + (<t_k|x> - t_k)^2$$

(18)

Now if the expression in (18) is substituted back into the MSE in (16) and the results are
simplified, we obtain:

\[ E = \frac{1}{2} \sum \int (y_k(x,w) - <t_k | x>)^2 p(x)dx + \frac{1}{2} \sum \int (\langle t_k^2 | x > - \langle t_k | x >^2 \rangle p(x)dx \]  \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

\[ y_k(x,w) = <t_k | x > \]  \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
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.
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](image)

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.

<table>
<thead>
<tr>
<th>No.</th>
<th>Variable Name</th>
<th>No.</th>
<th>Variable Name</th>
<th>No.</th>
<th>Variable Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>psybin3</td>
<td>13</td>
<td>prev_hos</td>
<td>25</td>
<td>ilsige2</td>
</tr>
<tr>
<td>2</td>
<td>depbin3</td>
<td>14</td>
<td>age_onse</td>
<td>26</td>
<td>asama2</td>
</tr>
<tr>
<td>3</td>
<td>anxbin3</td>
<td>15</td>
<td>occ_func</td>
<td>27</td>
<td>asahit2</td>
</tr>
<tr>
<td>4</td>
<td>chobin3</td>
<td>16</td>
<td>ilsipe2</td>
<td>28</td>
<td>asafa2</td>
</tr>
<tr>
<td>5</td>
<td>othbin3</td>
<td>17</td>
<td>ilsihy2</td>
<td>29</td>
<td>westco2</td>
</tr>
<tr>
<td>6</td>
<td>legalMHBbin</td>
<td>18</td>
<td>ilsicl2</td>
<td>30</td>
<td>westpe2</td>
</tr>
<tr>
<td>7</td>
<td>legalVpGbin</td>
<td>19</td>
<td>ilsiba2</td>
<td>31</td>
<td>westre2</td>
</tr>
<tr>
<td>8</td>
<td>legalNRRIbin</td>
<td>20</td>
<td>ilsin2</td>
<td>32</td>
<td>westc2</td>
</tr>
<tr>
<td>9</td>
<td>legalvolbin</td>
<td>21</td>
<td>ilsiho2</td>
<td>33</td>
<td>bdi_tota</td>
</tr>
<tr>
<td>10</td>
<td>Gender</td>
<td>22</td>
<td>ilsimo2</td>
<td>34</td>
<td>bhs_tota</td>
</tr>
<tr>
<td>11</td>
<td>Age</td>
<td>23</td>
<td>ilsico2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>high_ed</td>
<td>24</td>
<td>ilsire2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: 34 variables with regards to sociobehavioral 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.

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

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 \( \text{VAR1}_i \) for the \( i \)th 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 \( \text{VAR1}_i \).

\[
\text{VAR1}_i' = B_1(\text{VAR2}) + B_2(\text{VAR3}) + B_3(\text{VAR4}) + \ldots + B_{33}(\text{VAR33}) + b
\]  

(21)

where \( \text{VAR1}_i' \) is the estimated value of \( \text{VAR1}_i \) 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 B1, B2, … 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.

<table>
<thead>
<tr>
<th>Category Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age_a</td>
<td>less than 20</td>
</tr>
<tr>
<td>Age_b</td>
<td>20-29</td>
</tr>
<tr>
<td>Age_c</td>
<td>30-39</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Age_a</th>
<th>Age_b</th>
<th>Age_c</th>
<th>Age_d</th>
<th>Age_e</th>
<th>Age_f</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

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, \( \text{tanh} \), 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.
The algorithms used by MATLAB™ Neural Network Toolbox are not specified explicitly. A user specifies the settings such as:

<table>
<thead>
<tr>
<th>Network Type</th>
<th>Feed-forward backpropagation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training function</td>
<td>TRAINGD(Gradient descent backpropagation)</td>
</tr>
<tr>
<td>Adaption learning function</td>
<td>LEARNGD(Gradient descent weight and bias learning function)</td>
</tr>
<tr>
<td>Performance function</td>
<td>MSE(mean square error)</td>
</tr>
</tbody>
</table>

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. 9: Single problem neural network structure by posteriori probability approach
Fig. 10: Backpropagation algorithm flowchart

1. Initialize
2. Define the ANN structure
3. Initialize weights
4. Set the learning rate and momentum
5. \( i_{epoch} = 0, \ i_{pattern} = 0 \)
6. Has the number of epochs reached?
   - No: Calculate actual outputs
   - Yes: End
7. Calculate actual outputs
8. Calculate and graph error
9. Modify auxiliary weights and bias
10. Has the number of desired patterns reached?
    - No: Assign auxiliary weights to actual weights
    - Yes: \( i_{pattern} = i_{pattern} + 1 \)
11. Assign auxiliary weights to actual weights
12. \( i_{epoch} = i_{epoch} + 1 \)
13. Reset: \( i_{pattern} = 0 \)
Fig. 11: Posteriori probability approach flowchart

- Initialize
- Build the ANN structure
- Initialize the weights
- Set the learning rate
- \( \text{iepoch} = 0 \)

- Has the number of epochs reached?
  - Yes → End
  - No → Forward propagation: use softmax function to determine probability of the two outputs
    - Calculate and graph error
    - Modify the weights
    - \( \text{iepoch} = \text{iepoch} + 1 \)
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.
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.
<table>
<thead>
<tr>
<th>Independent Living Skill Deficits</th>
<th>Disorder Management Deficits</th>
<th>Occupational Skill Deficits</th>
</tr>
</thead>
</table>

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.
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.
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.
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.

<table>
<thead>
<tr>
<th>Social skill deficit &amp; Dysregulation of anger/aggression</th>
<th>Backpropagation Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train Accuracies</td>
</tr>
<tr>
<td>1</td>
<td>98.6%</td>
</tr>
<tr>
<td>2</td>
<td>98.6%</td>
</tr>
<tr>
<td>3</td>
<td>98.6%</td>
</tr>
<tr>
<td>4</td>
<td>98.6%</td>
</tr>
<tr>
<td>5</td>
<td>98.6%</td>
</tr>
<tr>
<td>6</td>
<td>97.2%</td>
</tr>
<tr>
<td>7</td>
<td>97.2%</td>
</tr>
<tr>
<td>8</td>
<td>97.2%</td>
</tr>
<tr>
<td>9</td>
<td>98.6%</td>
</tr>
<tr>
<td>10</td>
<td>97.2%</td>
</tr>
<tr>
<td>Max</td>
<td>98.6%</td>
</tr>
<tr>
<td>Min</td>
<td>97.2%</td>
</tr>
<tr>
<td>Mean</td>
<td>98.0%</td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>Dysregulation of anger/aggression &amp; Substance abuse</th>
<th>Backpropagation Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train Accuracies</td>
</tr>
<tr>
<td>1</td>
<td>96.1%</td>
</tr>
<tr>
<td>2</td>
<td>97.4%</td>
</tr>
<tr>
<td>3</td>
<td>92.2%</td>
</tr>
<tr>
<td>4</td>
<td>94.8%</td>
</tr>
<tr>
<td>5</td>
<td>96.1%</td>
</tr>
<tr>
<td>6</td>
<td>96.1%</td>
</tr>
<tr>
<td>7</td>
<td>97.4%</td>
</tr>
<tr>
<td>8</td>
<td>96.1%</td>
</tr>
<tr>
<td>9</td>
<td>97.4%</td>
</tr>
<tr>
<td>10</td>
<td>96.1%</td>
</tr>
<tr>
<td>Max</td>
<td>97.4%</td>
</tr>
<tr>
<td>Min</td>
<td>92.2%</td>
</tr>
<tr>
<td>Mean</td>
<td>96.0%</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Problem Titles</th>
<th>Number of Consistent Cases</th>
<th>Number of Inconsistent Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent Living Skill Deficits</td>
<td>30</td>
<td>7</td>
</tr>
<tr>
<td>Disorder Management Deficits</td>
<td>35</td>
<td>2</td>
</tr>
<tr>
<td>Occupational Skill Deficits</td>
<td>35</td>
<td>2</td>
</tr>
<tr>
<td>Dysregulation of Anger/Aggression</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>Substance Abuse</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>Social Skill Deficits</td>
<td>23</td>
<td>14</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Number of Training Sample</th>
<th>Test Accuracies</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low Bound</td>
<td>Upper Bound</td>
<td>Mean</td>
</tr>
<tr>
<td>80</td>
<td>50.0%</td>
<td>75.0%</td>
<td>61.0%</td>
</tr>
<tr>
<td>70</td>
<td>46.7%</td>
<td>67.7%</td>
<td>56.7%</td>
</tr>
<tr>
<td>60</td>
<td>45.0%</td>
<td>67.5%</td>
<td>56.3%</td>
</tr>
<tr>
<td>50</td>
<td>44.0%</td>
<td>66.0%</td>
<td>53.4%</td>
</tr>
<tr>
<td>40</td>
<td>41.7%</td>
<td>65.0%</td>
<td>52.7%</td>
</tr>
<tr>
<td>30</td>
<td>41.0%</td>
<td>60.0%</td>
<td>52.3%</td>
</tr>
</tbody>
</table>

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 like 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 minima” 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 sociobehavioral 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


[2]. The SOCRATES Decision Support System: Organizational structure and domain ontology, Fall 2008


[4]. Jeffrey Stephen Poland, Mary E. Sullivan, William D. Spaulding, *Treatment and Rehabilitation of Severe Mental Illness*(Hardcover), 2003


[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


[17]. OPEN DESIRE: [http://sites.google.com/site/gatmkorn/](http://sites.google.com/site/gatmkorn/), 2010.6


[19]. Matlab™ Help Documents for Neural Network Toolbox.


[27]. Wikipedia, the free encyclopedia, definition of Linear_discriminant_analysis: [http://en.wikipedia.org/wiki/Linear_discriminant_analysis](http://en.wikipedia.org/wiki/Linear_discriminant_analysis), 2010.8


## Appendix A

### Variable Description

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Variable Label</th>
<th>Variable Codes</th>
<th>Entity/Functional Domain</th>
<th>Why collected?</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>psybin3</td>
<td>Antipsychotic Med 1 Month 1-90</td>
<td></td>
<td>Neurophysiological</td>
<td>Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications</td>
</tr>
<tr>
<td>depbin3</td>
<td>Antidepressant Med 1 Month 1-90</td>
<td></td>
<td>Neurophysiological</td>
<td>Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications</td>
</tr>
<tr>
<td>anxbin3</td>
<td>Anxiolytic Med 1 Month 1-90</td>
<td></td>
<td>Neurophysiological</td>
<td>Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications</td>
</tr>
<tr>
<td>chobin3</td>
<td>Anticholinergic Med 1 Month 1-90</td>
<td></td>
<td>Neurophysiological</td>
<td>Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications</td>
</tr>
<tr>
<td>othbin3</td>
<td>Other Med 1 Month 1-90</td>
<td></td>
<td>Neurophysiological</td>
<td>Type of treatment; Can use symptom/behavioral assessments to draw conclusions about effectiveness of medications</td>
</tr>
<tr>
<td></td>
<td>Legal Status</td>
<td>Indicates type of commitment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
<td>-------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalMHBbin</td>
<td>1=MHB (Mental Health Board)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalVpGbin</td>
<td>2=VpG (Voluntary per Guardian)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalNRRIbin</td>
<td>3=NRRI (Not Responsible by Reason of Insanity)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalvolbin</td>
<td>4=Vol. (Voluntary)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td>gender</td>
<td>1 male 2 female</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>Age at Admission</td>
<td>#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>high_ed</td>
<td>Years of Education</td>
<td>##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prev_hos</td>
<td>Number of Previous Hosps (before CTP, not including CTP)</td>
<td>#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age_onse</td>
<td>Age of Onset</td>
<td>##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>occ_func</td>
<td>Highest Occupational Functioning</td>
<td>1=Professional, technical, manager, official, proprietor, clerical, sales, student 2=skilled worker-craftsman, foreman 3=semi-skilled-farmer, service worker, operator 4=laborer 5=unemployed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ilsipe2</td>
<td>ILSI Personal Management Time 1-17</td>
<td>Sociobehavorial; subdomain: skills Indicates Independent Living Skill Ability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Domain</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
<td>-----------------</td>
<td>--------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ilsihy2</td>
<td>ILSI Hygiene &amp; Grooming Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsicl1-17</td>
<td>ILSI Clothing Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsiba1-17</td>
<td>ILSI Basic Skills Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsiin1-17</td>
<td>ILSI Interpersonal Skills Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsiho1-17</td>
<td>ILSI Home Maintenance Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsimo1-17</td>
<td>ILSI Money Management Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsico1-17</td>
<td>ILSI Cooking Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsire1-17</td>
<td>ILSI Resource Utilization Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>ilsige1-17</td>
<td>ILSI General Occupational Functioning Time 1-17</td>
<td>Sociobehavioral; subdomain: skills</td>
<td>Indicates Independent Living Skill Ability</td>
<td></td>
</tr>
<tr>
<td>asama1-17</td>
<td>COGLAB Asarnow Masking Time 1-17</td>
<td>Neurocognitive</td>
<td>COGLAB measures attention and executive function</td>
<td></td>
</tr>
<tr>
<td>asahit1-17</td>
<td>COGLAB Asarnow Hits Time 1-17</td>
<td>Neurocognitive</td>
<td>COGLAB measures attention and executive function</td>
<td></td>
</tr>
<tr>
<td>asafa1-17</td>
<td>COGLAB Asarnow False Alarms Time 1-17</td>
<td>Neurocognitive</td>
<td>COGLAB measures attention and executive function</td>
<td></td>
</tr>
<tr>
<td>westcol1-17</td>
<td>COGLAB WCST Correct Time 1-17</td>
<td>Neurocognitive</td>
<td>COGLAB measures attention and executive function</td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>#</td>
<td>discrete category</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>---</td>
<td>-------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>psybin3</td>
<td>1</td>
<td>psybin3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>depbin3</td>
<td>2</td>
<td>depbin3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>anxbin3</td>
<td>3</td>
<td>anxbin3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chobin3</td>
<td>4</td>
<td>chobin3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>othbin3</td>
<td>5</td>
<td>othbin3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalMHBbin</td>
<td>6</td>
<td>legalMHBbin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalVpGbin</td>
<td>7</td>
<td>legalVpGbin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalNRREin</td>
<td>8</td>
<td>legalNRREin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>legalvolbin</td>
<td>9</td>
<td>legalvolbin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td>10</td>
<td>gender</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>11</td>
<td>agea 19-24</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>ageb 25-39</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>agec 40-55</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Appendix B

Categorization of variables
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>aged 56 or older</td>
</tr>
<tr>
<td>high_ed</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>high_ed_a 12 or less</td>
</tr>
<tr>
<td>16</td>
<td>high_ed_b 13-16</td>
</tr>
<tr>
<td>17</td>
<td>high_ed_c 17 or more</td>
</tr>
<tr>
<td>prev_hos</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>prev_hos_a 0-10</td>
</tr>
<tr>
<td>19</td>
<td>prev_hos_b 11-20</td>
</tr>
<tr>
<td>20</td>
<td>prev_hos_c 21 or more</td>
</tr>
<tr>
<td>age_onse</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>age_onse_a less than 16</td>
</tr>
<tr>
<td>22</td>
<td>age_onse_b 17-20</td>
</tr>
<tr>
<td>23</td>
<td>age_onse_c 21-25</td>
</tr>
<tr>
<td>24</td>
<td>age_onse_d 26-30</td>
</tr>
<tr>
<td>25</td>
<td>age_onse_e 31-35</td>
</tr>
<tr>
<td>26</td>
<td>age_onse_f 36-45</td>
</tr>
<tr>
<td>27</td>
<td>age_onse_g 46 or older</td>
</tr>
<tr>
<td>occ_func</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>occ_func_a 1=Professional, technical, manager, official, proprietor, clerical, sales, student</td>
</tr>
<tr>
<td>29</td>
<td>occ_func_b 2=skilled worker-craftsman, foreman</td>
</tr>
<tr>
<td>30</td>
<td>occ_func_c 3=semi-skilled farmer, service worker, operator</td>
</tr>
<tr>
<td>31</td>
<td>occ_func_d 4=laborer</td>
</tr>
<tr>
<td>32</td>
<td>occ_func_e 5=unemployed</td>
</tr>
<tr>
<td>ilsipe2</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>ilsipe2_a 0-5</td>
</tr>
<tr>
<td>34</td>
<td>ilsipe2_b 6-11</td>
</tr>
<tr>
<td>35</td>
<td>ilsipe2_c 12-14</td>
</tr>
<tr>
<td>36</td>
<td>ilsipe2_d 15-18</td>
</tr>
<tr>
<td>ilsihy2</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>ilsihy2_a 0-8</td>
</tr>
<tr>
<td>38</td>
<td>ilsihy2_b 9-17</td>
</tr>
<tr>
<td>39</td>
<td>ilsihy2_c 18-22</td>
</tr>
<tr>
<td>40</td>
<td>ilsihy2_d 23-27</td>
</tr>
<tr>
<td>ilsicl2</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ilsicl2_a 0-4</td>
</tr>
<tr>
<td>42</td>
<td>ilsicl2_b 5-9</td>
</tr>
<tr>
<td>43</td>
<td>ilsicl2_c 10-12</td>
</tr>
<tr>
<td>44</td>
<td>ilsicl2_d 13-15</td>
</tr>
<tr>
<td>ilsiba2</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>ilsiba2_a 0-5</td>
</tr>
<tr>
<td>46</td>
<td>ilsiba2_b 6-11</td>
</tr>
<tr>
<td>47</td>
<td>ilsiba2_c 12-14</td>
</tr>
<tr>
<td>48</td>
<td>ilsiba2_d 15-18</td>
</tr>
<tr>
<td>ilsiin2</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>ilsiin2_a 0-16</td>
</tr>
<tr>
<td>50</td>
<td>ilsiin2_b 17-33</td>
</tr>
<tr>
<td>51</td>
<td>ilsiin2_c 34-42</td>
</tr>
<tr>
<td>52</td>
<td>ilsiin2_d 43-51</td>
</tr>
<tr>
<td>ilsiho2</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>ilsiho2_a 0-5</td>
</tr>
<tr>
<td>54</td>
<td>ilsiho2_b 6-11</td>
</tr>
<tr>
<td></td>
<td>ilsino2</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td>55</td>
<td>ilsino2c 12-14</td>
</tr>
<tr>
<td>56</td>
<td>ilsino2d 15-18</td>
</tr>
<tr>
<td>57</td>
<td>ilsino2a 0-5</td>
</tr>
<tr>
<td>58</td>
<td>ilsino2a 0-5</td>
</tr>
<tr>
<td>59</td>
<td>ilsino2b 6-11</td>
</tr>
<tr>
<td>60</td>
<td>ilsino2d 15-18</td>
</tr>
<tr>
<td>ilsico2</td>
<td>61</td>
</tr>
<tr>
<td>62</td>
<td>ilsico2b 8-15</td>
</tr>
<tr>
<td>63</td>
<td>ilsico2c 16-19</td>
</tr>
<tr>
<td>64</td>
<td>ilsico2d 20-24</td>
</tr>
<tr>
<td>ilsire2</td>
<td>65</td>
</tr>
<tr>
<td>66</td>
<td>ilsire2b 6-11</td>
</tr>
<tr>
<td>67</td>
<td>ilsire2c 12-14</td>
</tr>
<tr>
<td>68</td>
<td>ilsire2d 15-18</td>
</tr>
<tr>
<td>ilsige2</td>
<td>69</td>
</tr>
<tr>
<td>70</td>
<td>ilsige2b 11-20</td>
</tr>
<tr>
<td>71</td>
<td>ilsige2c 21-27</td>
</tr>
<tr>
<td>72</td>
<td>ilsige2d 28-33</td>
</tr>
<tr>
<td>asama2</td>
<td>73</td>
</tr>
<tr>
<td>74</td>
<td>asama2b 25-34</td>
</tr>
<tr>
<td>75</td>
<td>asama2c 35 or more</td>
</tr>
<tr>
<td>asahit2</td>
<td>76</td>
</tr>
<tr>
<td>77</td>
<td>asahit2b 6-10</td>
</tr>
<tr>
<td>78</td>
<td>asahit2c 11-15</td>
</tr>
<tr>
<td>79</td>
<td>asahit2d 16-20</td>
</tr>
<tr>
<td>80</td>
<td>asahit2e 21-25</td>
</tr>
<tr>
<td>81</td>
<td>asahit2f 26-30</td>
</tr>
<tr>
<td>asafa2</td>
<td>82</td>
</tr>
<tr>
<td>83</td>
<td>asafa2b 1-5</td>
</tr>
<tr>
<td>84</td>
<td>asafa2c 6-10</td>
</tr>
<tr>
<td>85</td>
<td>asafa2d 11 or more</td>
</tr>
<tr>
<td>wcstco2</td>
<td>86</td>
</tr>
<tr>
<td>87</td>
<td>wcstco2b 20-24</td>
</tr>
<tr>
<td>88</td>
<td>wcstco2c 25 or more</td>
</tr>
<tr>
<td>wcstpe2</td>
<td>89</td>
</tr>
<tr>
<td>90</td>
<td>wcstpe2b 6-15</td>
</tr>
<tr>
<td>91</td>
<td>wcstpe2c 16-25</td>
</tr>
<tr>
<td>92</td>
<td>wcstpe2d 26-34</td>
</tr>
<tr>
<td>93</td>
<td>wcstpe2e 35 or more</td>
</tr>
<tr>
<td>wcstre2</td>
<td>94</td>
</tr>
<tr>
<td>95</td>
<td>wcstre2b 16-25</td>
</tr>
<tr>
<td>96</td>
<td>wcstre2c 26-35</td>
</tr>
<tr>
<td>97</td>
<td>wcstre2d 36-45</td>
</tr>
</tbody>
</table>
### Appendix C

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

<table>
<thead>
<tr>
<th>98</th>
<th>wcstre 46-55</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>wcstre 56 or more</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>wcstc2</th>
<th>100</th>
<th>wcstc2a 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>101</td>
<td>wcstc2b 1</td>
</tr>
<tr>
<td></td>
<td>102</td>
<td>wcstc2c 2</td>
</tr>
<tr>
<td></td>
<td>103</td>
<td>wcstc2d 3</td>
</tr>
<tr>
<td></td>
<td>104</td>
<td>wcstc2e 4</td>
</tr>
<tr>
<td></td>
<td>105</td>
<td>wcstc2f 5</td>
</tr>
<tr>
<td></td>
<td>106</td>
<td>wcstc2g 6 or more</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bdi_tota</th>
<th>107</th>
<th>bdi_totaa 0-13</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>108</td>
<td>bdi_totab 14-19</td>
</tr>
<tr>
<td></td>
<td>109</td>
<td>bdi_totac 20-28</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>bdi_totad 29 or more</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bhs_tota</th>
<th>111</th>
<th>bhs_totaa 0-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>112</td>
<td>bhs_totab 4-8</td>
</tr>
<tr>
<td></td>
<td>113</td>
<td>bhs_totac 9-14</td>
</tr>
<tr>
<td></td>
<td>114</td>
<td>bhs_totad 15 or more</td>
</tr>
</tbody>
</table>

---

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  
360  for i=1 to 100  
370  for j=1 to 115
input #3,ori[i,j]
if eof then go to 303 | else proceed
next
next
disconnect 3
-- random selection for training set INPUT[80,114] and test set INPUTTEST[20,114]
-- produce 52 random number for random selection
n=52
ARRAY a3[n]
i=0
repeat
a1=(ran()+1)/2
a2=round(a1*n)
if a2<>0 then i=1 | a3[1]=a2 | else proceed
until i=1
repeat
a1=(ran()+1)/2
a2=round(a1*n)
-- write a1,a2
f=0
for j=1 to i
if a2=a3[j] then f=1 | else proceed
next
if a2=0 then f=1 | else proceed
if f=0 then i=i+1 | a3[i]=a2 | -- write a3
else proceed
until i=n
-----
-- produce 48 random number for random selection
n=48
ARRAY a4[n]
i=0
repeat
a1=(ran()+1)/2
a2=round(a1*n)
if a2<>0 then i=1 | a4[1]=a2 | else proceed
until i=1
repeat
a1=(ran()+1)/2
a2=round(a1*n)
f=0
for j=1 to i
if a2=a4[j] then f=1 | else proceed
if \( a_2 = 0 \) then \( f = 1 \) | else proceed

if \( f = 0 \) then \( i = i + 1 \) | \( a_4[i] = a_2 \) | -- write \( a_4 \)

else proceed

until \( i = n \)

-----

for \( i = 1 \) to 40

tmp1 = \( a_3[i] \)
tmp2 = \( a_4[i] + 52 \)

for \( j = 1 \) to 114

\( INPUT[i*2-1,j] = ori[tmp1,j] \)
\( INPUT[i*2,j] = ori[tmp2,j] \)

next

\( TARGET[i*2-1,1] = ori[tmp1,115] \)
\( TARGET[i*2,1] = ori[tmp2,115] \)

next

for \( i = 1 \) to 12

tmp2 = \( i + 40 \)

\( tmp3 = a_3[tmp2] \)

for \( j = 1 \) to 114

\( INPUTTEST[i,j] = ori[tmp3,j] \)

next

\( tarTest[i,1] = ori[tmp3,115] \)

next

for \( i = 13 \) to 20

tmp2 = \( i + 28 \)

\( tmp3 = a_4[tmp2] + 52 \)

for \( j = 1 \) to 114

\( INPUTTEST[i,j] = ori[tmp3,j] \)

next

\( tarTest[i,1] = ori[tmp3,115] \)

next

--

-- random selection ends

-- small random initial weights to "break symmetry"

-- Initialize the random weights

for \( i = 1 \) to \( n_2 \)

for \( j = 1 \) to 114

\( W12[i,j] = 0.02 \times ran() \)

next

\( W23[1,i] = 0.03 \times ran() \)

next

---------------------------------------
-- set experiment parameters
lrate2=0.002  lrate3=0.002  --set learning rate
r2=0.001  r3=0.0005 --set momentum
NN=16000  -- 16000 trials

-------------
display N15 | display C7 | -- set display colors
display R | -- and thick dots
-------------
drun  | -- make a simulation run (NN trials)

write 'type go for a recall test'
-- STOP
---

---

label jjj | -- try all
NNold=NN | NN=20 | -- save old NN for new runs
lrate2=0 | lrate3=0 | r2=0 | r3=0
t=1  | TMAX=NN-1

-------------
drun RECALL
cnt=0  -- calculate test accuracy
for i=1 to ntest
   if err[i]=0 then cnt=cnt+1 | else proceed
next
rate=cnt/ntest
write rate | -- write test accuracy
NN=NNold | TMAX=NN-1 | -- restore NN

-------------

DYNAMIC
-------------

iRow=t | Vector layer1=INPUT#
Vector layer2=tanh(w12*layer1+bias2)
Vector layer3=tanh(w23*layer2+bias3)
out1=layer3[1]

-------------

Vector error=TARGET#-layer3 | -- backpropagation
Vector delta3=e
Vector delta3=error*(1-layer3*layer3)
Vector delta2=w23%*delta3*(1-layer2^2)

-- accumulate temporary weights, biases for M trials
--
DELTA W12=lrate2*delta2*layer1+r2*w12
DELTA W23=lrate3*delta3*layer2+r3*w23
Vectr delta BIAS2=lrate2*delta2+r2*bias2
Vectr delta BIAS3=lrate3*delta3+r3*bias3

ERROR=2*error[1] | -- offset error display

SAMPLE M | -- update actual weights, biases only ...

MATRIX w12=W12 | MATRIX w23=W23 | -- update
Vector bias2=BIAS2 | Vector bias3=BIAS3

dispt ERROR | -- display error measure

iRow=t | Vector layer1=INPUTTEST# | -- recall run
Vector layer2=sat(w12*layer1+bias2)
Vector layer3=sat(w23*layer2+bias3)

out1=layer3[1]
Vector error2=tartest#-layer3 | out2=error2[1]
type out1,out2 | -- test for correct truth table
store err=error2[1]

Appendix D

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

estimates a posteriori probabilities

display N14 | display C7 | -- set display colors
display R | -- and thick dots

nx=114 | N=2 | -- 114 nodes in input layer and 2 nodes in output layer
ntrain=80 | ntest=20 | -- 80 patterns for training and 20 patterns for test

build the ANN model

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:\annodata\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[i]=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
MATRIX TARGET=1 |  --  (binary-classifier rows)

---

next

--

-- random selection ends

---

lrate=1.75 |  c=0.2

lrate1=lrate*c

NN=8000 |  -- 8000 trials

---

drun

write 'type go for successive recall runs' |  -- STOP

display F

----------

label recall
t=1 |  NN=2 |  TMAX=NN-1 |  restore |  -- reset the read point

for i=1 to ntest

drun RECALL

ii=i+1 |  --  test pattern output to the screen

write 'current test vector',ii

for jj=1 to N

if q[jj]=1 then output[i]=jj |  write output[i], 'is estimated _
to be the best match train vector' |  else proceed

next

write '-----------------------'

next

cnt=0 |  -- counter: count the correct outputs

for i=1 to 11

if output[i]=1 then cnt=cnt+1 |  else proceed

next

for i=12 to 19

if output[i]=2 then cnt=cnt+1 |  else proceed

next

if output[20]=1 then cnt=cnt+1 |  else proceed

rate=cnt/ntest

write rate |  -- outprint test accuracy

connect 'e:\anndata\ssd_output_time5.txt' as output 3

for i=1 to ntest

write #3,output[i]

next

disconnect 3

NN=8000 |  TMAX=NN-1 |  restore

---

DYNAMIC
iRow=t | Vector x=INPUT#
Vector v=exp(c*W*x) | DOT vsum=v*1 -- use softmax function
Vector v=v/vsum | -- probability estimate
Vector error=TARGET#-v | -- calculate error
Vector q^=v | Vector q=swtch(q) | -- enhance the larger to 1 and reduce the smaller to 0
Vector error2=TARGET#-q

Vector v1=error*v*(1-v)
DELTA W=lrate1*v1*x | -- update weights

--
DOT enormsq=error*error
ENORMSQ=enormsq
dispt ENORMSQ

label RECALL
iRow=t | Vector x=INPUTTEST# | -- testing process forwards propagation
Vector v=exp(c*W*x) | DOT vsum=v*1
Vector v=v/vsum | -- probability estimate
Vector q^=v | Vector q=swtch(q) | -- binary selector

Appendix E

DESIREE program for describing combination of social skill deficit and dysregulation of anger/aggression model
n2=68 --68 neurons in hidden layer
ntrain=71 | ntest=18 -- 71 patterns for training and 18 patterns for test
-- build ANN model
ARRAY layer1[114],layer2[n2],layer3[2] -- 114 neuron in input layer
ARRAY w12[n2,114],w23[2,n2],bias2[n2],bias3[2]
ARRAY W12[n2,114],W23[2,n2],BIAS2[n2],BIAS3[2]-- auxiliary weights W and bias B
ARRAY INPUT[ntrain,114],TARGET[ntrain,2],INPUTTEST[ntest,114]
ARRAY tarest[ntest,2],ori[89,116],errsd[ntest],errpsy[ntest]
ARRAY error[2],delta3[2],delta2[n2],error2[2]
n=89
ARRAY a3[n]
-- import data into matrix ori[89,116]
connect 'e:\anndata\0505\two_89_114.txt' as input 3
for i=1 to 89
  for j=1 to 116
    input #3, ori[i,j]
    if eof then go to 303 | else proceed
    next
  next
disconnect 3
i=0 |-- random selection for training set and test set
repeat
  a1=ran()/(ran()+1)
  a2=round(a1*n)
  if a2<>0 then i=1 | a3[i]=a2 | else proceed
  until i=1
repeat
  a1=ran()/(ran()+1)
  a2=round(a1*n)
  f=0
  for j=1 to i
    if a2=a3[j] then f=1 | else proceed
    next
  if a2<>0 then f=1 | else proceed
  if f=0 then i=i+1 | a3[i]=a2 | -- write a3
  else proceed
  until i=n
-----
for i=1 to ntrain
  tmp1=a3[i]
  for j=1 to 114
    INPUT[i,j]=ori[tmp1,j]
    next
  TARGET[i,1]=ori[tmp1,115]
  TARGET[i,2]=ori[tmp1,116]
next|-- random selection ends
-- small random initial weights to "break symmetry"
-- Initialize the random weights
for i=1 to n2
  for j=1 to 114
    W12[i,j]=0.02*ran()
    next
  for j=1 to 2
    W23[j,i]=0.03*ran()
    next
  next
next
-- set experiment parameters
lrate2=0.002 | lrate3=0.002 -- set learning rate
r2=0.001 | r3=0.0005 -- set momentum
NN=16000 -16000 trials

--

display N15 | display C7 | -- set display colors
display R | -- and thick dots

-- make a simulation run (NN trials)

write 'type go for a recall test'

-- STOP

--

label jji | -- try all
NNold=NN | NN=ntest | -- save old NN for new runs
lrate2=0 | lrate3=0 | r2=0 | r3=0
t=1 | TMAX=NN-1

for i=1 to ntest |-- fill the test array and test target array
tmp2=i+ntrain
tmp3=a3[tmp2]
for j=1 to 114
INPUTTEST[i,j]=ori[tmp3,j]
next
tartest[i,1]=ori[tmp3,115]
tartest[i,2]=ori[tmp3,116]
next

--

for i=1 to ntest |-- result calculation
cntssd=0 | cntpsy=0 | cnttwo=0 |-result calculation
for i=1 to ntest
cntssd=cntssd+1 | else proceed
cntpsy=cntpsy+1 | else proceed
cnttwo=cnttwo+1 | else proceed

ssdrate=cntssd/18
psyrate=cntpsy/18
tworate=cnttwo/18

write tworate |-- outprint test accuracy
NN=NNold | TMAX=NN-1 | -- restore NN
DYNAMIC

weight-learning run

iRow=t | Vector layer1=INPUT# |-- forwards propagation

Vector layer2=tanh(w12*layer1+bias2)
Vector layer3=tanh(w23*layer2+bias3)
out1=layer3[1]

Vector error=TARGET#-layer3 | -- backpropagation

Vector delta3=error*(1-layer3^2)
Vector delta2=w23%*delta3*(1-layer2^2)

DELTA W12=lrate2*delta2*layer1+r2*w12
DELTA W23=lrate3*delta3*layer2+r3*w23

Vector bias2=BIAS2 | Vector bias3=BIAS3

MATRIX w12=W12 | MATRIX w23=W23 | -- update

Vector layer2=sat(w12*layer1+bias2)
Vector layer3=sat(w23*layer2+bias3)

Vector error2=tartest#-layer3

out1=layer3[1]
out2=layer3[2]

store errssd=error2[1] | -- store for result calculation
store errpsy=error2[2]