DOCUMENT RESUME

ED 368 182 FL 021 928

AUTHOR Perkins, Kyle; And Others
TITLE Predicting Item Difficulty in a Reading Comprehension Test with an Artificial Neural Network.
PUB DATE Mar 94
NOTE 22p.; Paper presented at the Annual Language Testing Research Colloquium (16th, 1994).
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Computer Assisted Testing; Expert Systems; Item Analysis; Linguistic Theory; *Neurolinguistics; Predictive Measurement; *Predictive Validity; Reading Comprehension; *Test Items; *Test Reliability
IDENTIFIERS *Neural Networks

ABSTRACT

This paper reports the results of using a three-layer backpropagation artificial neural network to predict item difficulty in a reading comprehension test. Two network structures were developed, one with and one without a sigmoid function in the output processing unit. The data set, which consisted of a table of coded test items and corresponding item difficulties, was partitioned into a training set and a test set in order to train and test the neural networks. To demonstrate the consistency of the neural networks in predicting item difficulty, the training and test sets were repeated four times starting with a new set of initial weights. Additionally, the training and testing runs were repeated by switching the training set and the test set. The mean squared error values between the actual and predicted item difficulty demonstrated the consistency of the neural networks in predicting item difficulty for the multiple training and test runs. Significant correlations were obtained between the actual and predicted item difficulties and the Kruskal-Wallis test, indicating no significant difference in the ranks of actual and predicted values. (Author/MDM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 368 182

Predicting Item Difficulty in a Reading Comprehension Test with an Artificial Neural Network

Kyle Perkins[1], Lalit Gupta[2], and Ravi Tammana[2]

Southern Illinois University

Carbondale, Illinois 62901

[1]Department of Linguistics

[2]Department of Electrical Engineering

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Kyle Perkins
Lalit Gupta

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

FL021928

BEST COPY AVAILABLE

Abstract

This paper reports the results of using a three-layer backpropagation artificial neural network to predict item difficulty in a reading comprehension test. Two network structures were developed: one with the sigmoid function in the output processing unit and the other without the sigmoid function in the output processing unit. The data set which consisted of a table of coded test items and corresponding item difficulties was partitioned into a training set and a test set in order to train and test the neural networks. To demonstrate the consistency of the neural networks in predicting item difficulty, the training and testing runs were repeated four times starting with a new set of initial weights. Additionally, the training and testing runs were repeated by switching the training set and the test set. The mean squared error values between the actual and predicted item difficulty demonstrated the consistency of the neural networks in predicting item difficulty for the multiple training and testing runs. Significant correlations were obtained between the actual and predicted item difficulties and the Kruskal-Wallis test indicated no significant difference in the ranks of actual and predicted values.

I Introduction

This paper focuses on developing an artificial neural network (ANN) approach to predict item difficulty in a standardized reading comprehension test. The rationale for the study was motivated by many considerations which can be generally categorized into two broad areas: (1) reading research and (2) the ability of ANNs to outperform traditional statistical techniques such as multipl regression in prediction studies.

Reading research

Identifying the variables which uniquely account for significant variance in the percent correct obtained by examinees for each item in a standardized, group administered reading comprehension test is a major focus in reading research for the following reasons. There are many potential sources of difficulty in a reading comprehension test which may derive from the way in which the prose passages and the reading comprehension questions are constructed (Scheuneman, Gerritz, and Embretson, 1989). Test item writers do not usually control nor quantify the sources of difficulty and reading researchers are unsure of what factors account for the observed item difficulty in a multiple-choice reading comprehension test (Embretson and Wetzel, 1987). Researchers have noted that content and test development experts cannot reliably estimate the difficulty of a test item (Bejar, 1983).

ANN versus traditional statistical techniques

There is growing literature that suggests that ANNs outperform traditional statistical procedures such as multiple regression in prediction studies. Studies in which traditional statistical methods and ANNs have been compared show a favorable advantage for ANNs in time/forecasting (Sharda and Patil, 1992), processing control (Nelson and Illingworth, 1991), signal processing (Lapedes and Farber, 1987), and predicting an AIDS risk index (Lykins and Chance, 1992). A reason which has been offered to account for the better performance of ANN over multiple regression is that backpropagation networks (one type of ANN) are a form of nonlinear regression and are not bound to the functional fitting inherent in multiple regression which utilizes the least-mean-squared error to determine the best representative function in a data set (Lykins and Chance, 1992).

The validity of the studies in which multiple regression is used to predict item difficulty is not high. Perkins and Brutten (1991) correlated 24 variables with the item difficulty indices from a standardized reading comprehension test and obtained correlation

coefficients ranging from *0.603* to *-0.011*. Only five variables correlated significantly at the *0.05* level with item difficulty, and these five variables were retained for a stepwise multiple regression analysis in which item difficulty was the dependant variable. Only four of the five variables uniquely accounted for significant variance and the entire model accounted for *72.49* percent of variance in the test. When only four of *24* variables account for significant variance in item difficulty, the validity of a multiple regression study is not high. It is hypothesized that using variables in combination and introducing forms of non-linearity might improve the validity of item difficulty studies. Combining variables and introducing non-linearity can be accomplished in an ANN by manipulating the input variables and by employing the non-linear transfer functions of the neurons in an ANN. Thus, the purpose of the study reported in this paper was to train an ANN to predict item difficulty in a reading comprehension test and to compare the actual item difficulties with the predicted item difficulties in order to determine whether the two sets of values were statistically similar or different.

II Artificial neural networks

Interest in artificial neural networks as an alternative to conventional algorithmic techniques has grown rapidly in recent years. Artificial neural networks attempt to emulate sophisticated brain-like functions such as learning and generalization. Researchers from diverse fields such as engineering, science, statistics, and mathematics are actively involved in developing and applying artificial neural net models to solve problems in pattern recognition, signal processing, biological system modeling, data analysis, and optimization.

An artificial neural network is a large parallel information processing network composed of many simple non-linear processing elements. Information is stored in a distributed fashion throughout the interconnections of the network. Artificial neural nets are specified by the network topology, node characteristics, and the training or learning rules. A variety of artificial neural net models have been developed and these include backpropagation nets (Werbos, 1974; Parker, 1982; Rumelhart & McClelland, 1986), associative memory nets (Hopfield & Tank, 1985), adaptive resonance nets (Grossberg, 1987), self-organizing nets (Kohonen, 1988), and counterpropagation nets (Hecht-Nielsen, 1987). The networks differ in that they operate on binary/continuous valued inputs, use

unsupervised/supervised training, and perform classification, clustering or optimization tasks.

The backpropagation neural network

The neural network model selected for the item difficulty prediction problem addressed in this paper is the backpropagation network. A backpropagation network implements a modifiable function which maps a set of inputs to a set of outputs. The functional form is modified by adjusting the adaptable interconnection weights by means of the backpropagation training algorithm. Since the item difficulty prediction problem essentially involves determining the functional mapping between the 24 variables (input) and the item difficulty (output), a backpropagation neural network can be designed to solve the mapping of the 24 variables to the corresponding item difficulties (p-values).

Network architecture

The backpropagation network is a hierarchical feed-forward network system consisting of two or more fully interconnected layers of processing units (artificial neurons). A *N-H-M* backpropagation network refers to a three-layer network with *N*, *H*, and *M* processing units in the first, second, and third layers respectively. A *N-H-1* backpropagation network is shown in Figure 1. The first, second, and third layers illustrated in this figure are the input layer, the hidden layer, and the output layer respectively. Each processing unit is represented by a circle and each interconnection between processing units by an arrow. Each interconnection is weighted by an adaptive coefficient called the interconnection weight (not marked in the figure). The input to the network is represented by the *N*-dimensional vector $U=(u_1,u_2,...,u_N)$ and the output by y. For convenience, the outputs of the processing units in the input and hidden layers are labelled in the circles representing the processing units. By using a training algorithm to adapt the interconnection weights, the backpropagation network has the ability to implement a wide range of responses to the patterns in a given training set.

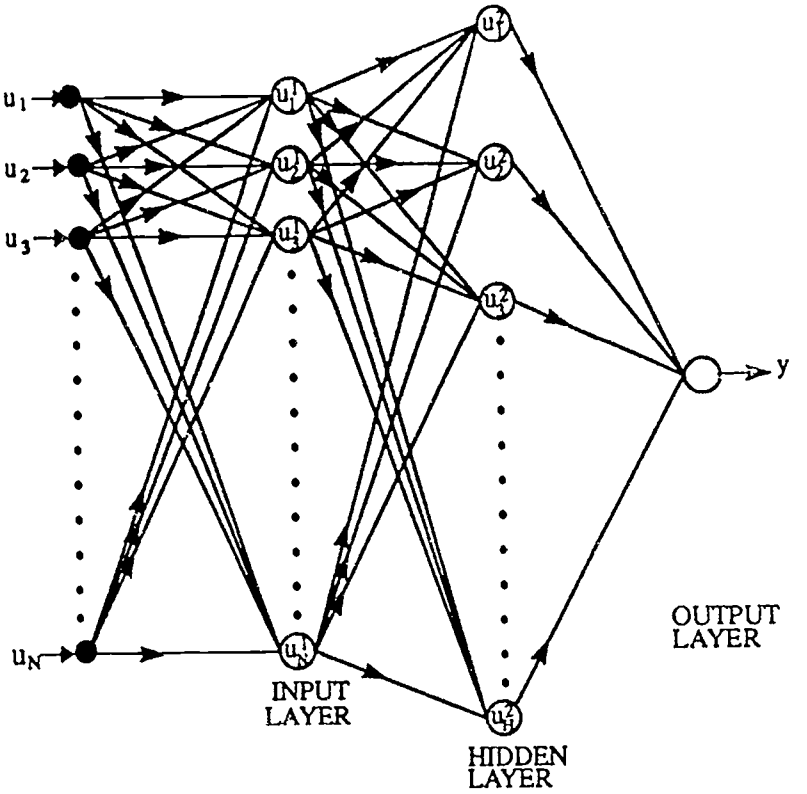

Fig. 1 A three-layer backpropagation neural network

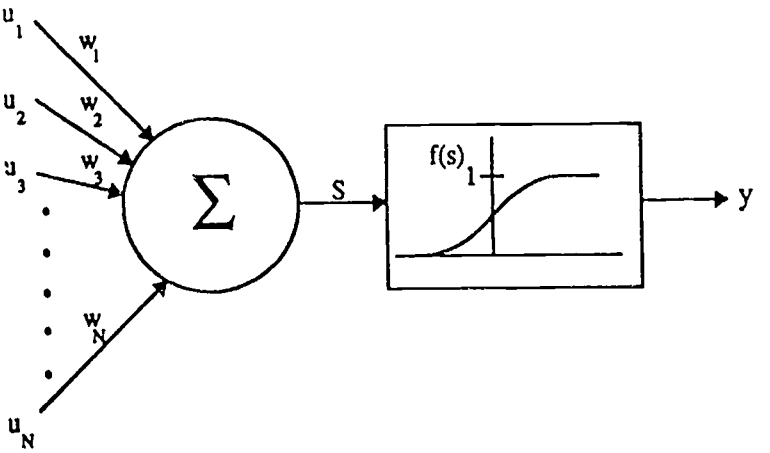

Fig. 2 A processing unit

Processing unit

Figure 2 shows a typical processing unit which consists of a summing unit and a non-linear sigmoidal activation function $f[.]$. The output y of the processing unit is given by

$y = f[S]$, where

$S = \sum_i w_i u_i$, and

$f[S] = 1/(1 + e^{-S})$.

That is, the processing unit first computes a weighted sum S of its inputs and then computes a function $f[S]$ of the weighted sum to give the activation level y (output) of the processing unit. Due to the sigmoidal function, the output y is limited to values between 0 and 1.

Network design issues

The design of a backpropagation network for a particular problem involves determining:

(1) The number of layers.

(2) The number of processing units in each layer.

(3) The format of the input to the network.

The backpropagation approximation theorem (Hecht-Nielsen, 1991) proves that three layers are generally enough to approximate the functional mapping required for most practical problems and, therefore, a backpropagation network with three layers is a good choice when no prior knowledge of the mapping is assumed. The number of processing units in the input layer is generally governed by the input dimension. No theory or rules exist to select the number of processing units in the hidden layer, therefore, the number of hidden layer units is determined empirically. The number of processing units in the output are problem-dependent; for example, pattern classification problems require the output layer to have one processing unit per pattern class (Gupta, Sayeh, & Tammana, 1990) or one processing unit per pattern feature (Gupta & Upadhye, 1991; Gupta et al., 1993). The prediction problem addressed in this paper requires one processing unit in the output layer. The input to the backpropagation network is a vector of real numbers and due to the fixed structure of the network, the input vector must have a fixed dimension.

Network training

The multilayer perceptron is trained under supervision using the backpropagation algorithm (Rumelhart & McClelland, 1986). The network is presented with pairs of vectors: the input

vector to the network and the desired network output vector for the input pattern vector. The network functions in two stages during training: a forward pass and a backward pass. In the forward pass, the input vector is presented to the network and the outputs of the units are propagated through each upper layer until the network output is generated. The difference (error) between the network output and the desired output is computed for each output unit and during the backward pass, a function of the error is fed back through the network layers to adjust the interconnection weights in order to minimize the error. The forward and backward passes are repeated until the network converges, that is, until a measure of the error is acceptably small. During training, the network gradually learns to produce the desired outputs. The backpropagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the desired network output and the actual network output. The backpropagation algorithm applied to train the neural network shown in Figure 1 is summarized below.

Network Dimensions:

Let the dimension of the input training vector and the network input layer be N and let H be the dimension of the hidden layer.

Network Initialization:

Set all the interconnection weights to small random values with zero mean. Typically, the weights are initialized to take random values between *-0.5* and *0.5*.

Apply Input and Set Desired Net Outputs:

Assume that the network is designed to predict M values and let $u_{i,m}$, $i=1,2,...,N$ represent a N dimensional training vector for the m*th* value. Let d_m be the desired network output for the input $u_{i,m}$. The presentation of the input may be done in several ways. One approach is to apply the input training vector $u_{i,m}$, set the desired network output d_m, and not change the training vector during the training iterations (an iteration is a forward pass and a backward pass) until the network converges to the desired output. The process is repeated for the remaining training vectors. Alternatively, the training vectors can be rotated cyclically from iteration to iteration. The desired network output must, therefore, be also set from iteration to iteration.

Compute Actual Unit Outputs (forward pass):

The processing of the outputs are carried out sequentially from the input layer to the output layer. The output of the *jth* unit in the input layer is given by:

$$u_j^1 = f[\sum_{i=1}^{N} w_{i,j} u_{i,m}], \quad 1 \le j \le N.$$

The outputs of the input layer are the inputs to the hidden layer and the output of the *kth* unit in the hidden layer is given by:

$$u_k^2 = f[\sum_{j=1}^{N} w_{j,k}^1 u_j^1], \quad 1 \le k \le H.$$

The outputs of the hidden layer are the inputs to the output layer and the network output in response to the input $u_{i,m}$ is given by:

$$y_m = f[\sum_{k=1}^{H} w_k^2 u_k^2].$$

In the above equations, $f[.]$ is the sigmoidal function, that is,

$$f[\alpha] = 1/[1 + e^{-\alpha}]$$

and $w_{i,j}$, $w_{j,k}^1$, and w_k^2 are the connection weights between the network input and the input layer, input and hidden layer, and the hidden layer and the output layer, respectively.

Update Weights (backward pass):

Hidden layer - Output layer interconnection weights:

The interconnection weights are updated sequentially from the output layer to the input layer. If $w_k^2(t)$ is the interconnection weight between *kth* unit in the hidden layer and the output layer unit at time *t*, then the weight at time *(t+1)* is given by:

$$w_k^2(t+1) = w_k^2(t) + \eta \delta_m u_k^2,$$

where

$$\delta_m = y_m(1-y_m)(d_m-y_m).$$

δ_m is the error for the output of the unit in the output layer when the input is $u_{i,m}$ and η is a gain term which controls the learning (adaptation) rate of the network. The gain term η is typically assigned values between *0* and *1*, however, the actual value selected between *0* and *1* is application-dependent. The gain term controls the convergence rate and the stability of the network and in practice, η is adjusted for fast adaptation and for obtaining stable estimates of the interconnection weights.

Input layer - Hidden layer interconnection weights:

The updated weights between the input and hidden layers are given by:

$$w^1_{j,k}(t+1) = w^1_{j,k}(t) + \eta\delta_k u^1_j .$$

where

$$\delta_k = u^2_k(1-u^2_k)(\Sigma^{M}_{m=1} \delta_m w^2_{k,m}).$$

δ_k is the error for the output of unit *k* in the hidden layer.

Input - Input layer interconnection weights:

The updated weights between the input and input layer are given by:

$$w_{i,j}(t+1) = w_{i,j}(t) + \eta\delta_j u_{i,m}.$$

where

$$\delta_j = u^1_j(1-u^1_j)(\Sigma^{H}_{k=1} \delta_k w^1_{j,k}).$$

δ_j is the error for unit *j* in the input layer.

The training set generally consists of a set of representative prototype vectors. The training prototypes could be a vector representation of the raw input data or a feature vector computed from the raw data. Network convergence can be tested in several ways (Gupta et al., 1992). The most practical test for network convergence is the error limit test, i.e., test if the absolute difference ε between the desired response and the response of each output unit is below a small specified error limit. Alternatively, training could be

terminated when the sum of the squares of the errors for all output units is below a specified limit.

Network testing

In the testing stage, the vector representing the input to be tested is presented to the input of the trained network and the network outputs are computed using one forward pass. In neural networks designed for classification problems, the input test pattern is assigned to the class of the network output that yields the maximum value (maximum response rule). For the prediction problem involving a single network output, the network is trained to predict m values. The value of the network output during testing is, therefore, the predicted value.

III Purpose and design of the study

The specific data set to be analyzed in this paper comes from a study by Perkins and Brutten (1991). Three classes of variables were examined: (1) various counts of text surface structure (Drum, Calfee, and Cook, 1981), (2) propositional analysis of the passages and item stems (Scheuneman and Gerritz, 1990; Scheuneman, Gerritz and Embretson, 1989) and (3) cognitive demand (Scheuneman, Gerritz and Embretson, 1989).

Text structure

The variables describing the structure of the texts included passage content (humanities, nonhumanities), the number of paragraphs per passage, the number of lines per passage, the number of test items per passage, the number of words per passage, the number of content words per passage, the number of sentences per passage, a passage word/sentence ratio, and the percent of content words per passage.

Propositional analysis

The following propositional counts were conducted for both the test passages and the item stems separately: the number of arguments, the number of modifiers, the number of predicates, arguments density (the number of arguments divided by the number of sentences), modifier density, predicate density, and combined density (the total number of propositions divided by the number of sentences).

Cognitive demand

Scheuneman, Gerritz and Embretson, (1989) used five cognitive process categories in their analysis which were modified as follows for the study reported herein:

0 = identify, recognize, name, discern, locate, match, exemplify, or illustrate a concrete piece of relevant information in the text which was given in an item stem.

1 = non-identification

- support/weaken a claim, procedure, or outcome; substantiate, demonstrate, prove, confirm, verify a result; negate, critique, contradict, or disprove a claim, procedure or outcome
- infer, conclude, induce, deduce, diagnose, distinguish, differentiate, contrast
- generalize, plausibly universalize, find common ground, transfer, apologize, apply, carry over
- problem/solve, calculate, inquire, experiment, evaluate, appraise, weigh, compare

(adapted from Scheuneman, Gerritz and Embretson, 1989, pp. 14-14).

If a test item required the reader to conduct a cognitive operation on a concrete, verbatim piece of information in the text, it was coded *0*. All other cognitive requirements were coded *1*.

IV Method

Subjects

Seventy students enrolled in intensive English classes participated as subjects for this study. The distribution of native languages was the following: Japanese, sixteen; Chinese, thirteen; Arabic, twelve; Korean, eleven; Spanish, nine; Thai, two; Turkish, two; Urdu, two; Hebrew, one; Indonesian, one; and Wolof, one.

Instrumentation

The elicitation instrument consisted of 29 reading comprehension items from the *Test of English as a Foreign Language*, Form 3LTFG (Educational Testing Service, 1990).

Data coding

Each of the twenty-nine reading comprehension test items was coded according to three sets of variables: text structure, propositional analysis of passages and stems, and cognitive demand. The two researchers coded the items independently and anonymously. Disagreements were adjudicated by a third party, and the consensus was recorded. Pearson correlations for the continuous variables and percentage agreement for the categorical variables were used to determine coding stability consistency. The coefficients ranged from *0.85* to *0.93*. The item difficulty of each item was calculated as the proportion of correct responses.

Network training and testing

The data available consisted of the *29* reading comprehension coded test items shown in Table 1. Two sets: a training set and a test set were created from the available data set. In order for the training set to be representative, items were selected by picking the first item and every other item in the table to give a total of *15* items in the training set. The remaining *14* items constituted the test set. A three-layer backpropagation network was designed with *24* input units (one for each of the *24* input variables) and *1* output unit for the predicted item difficulty (p-value). The number of units in the hidden layer was empirically determined to be *17*. The input data were normalized to take values between *0* and *1* by dividing each variable by its highest value in the table. Two variants of the *24-17-1* networks were implemented: one with the sigmoid function in the output processing unit and the other without the sigmoid function in the output processing unit. The rationale for implementing the network without the sigmoid function in the output unit was to determine the effect, if any, of the sigmoid function compressing heavily large and small input values. All processing units in the input and hidden layers used the sigmoid function. The gain term η used in training was *0.2* and the error limit test with ε=*0.05* was used to test for network convergence.

V Results

The neural networks were trained using the backpropagation training algorithm to output the desired p-values in the training set. The *14* items in the test set were tested and the results (Run 1) are shown in Tables 2 and 3. In order to demonstrate the consistency of the neural networks in predicting item difficulty, the two networks were trained starting with a

Table 1 Input data

Item number	Actual p-value	Passage content	Number of paragraphs in passage	Number of lines in passage	Number of items per passage	Number of passage arguments	Number of passage modifiers	Number of passage predicates	Passage argument density	Passage modifier density	Passage predicate density	Combined density	Number of words per passage	Number of content words per passage	Number of sentences per passage	Passage words/sentence ratio	Percent content words in passage	Number of arguments in question stem	Number of modifiers in question stem	Number of predicates in question stem	Stem argument density	Stem modifier density	Stem predicate density	Combined density	Cognitive demand
1	0.90	0.00	2.00	11.0	4.00	34.0	49.0	26.0	4.80	7.00	3.71	15.5	157	107	7.00	22.4	0.68	4.00	3.00	1.00	1.00	0.75	0.25	2.00	1.00
2	0.76	0.00	2.00	11.0	4.00	34.0	49.0	26.0	4.80	7.00	3.71	15.5	157	107	7.00	22.4	0.68	3.00	3.00	2.00	0.75	0.75	0.50	2.00	0.00
3	0.90	0.00	2.00	11.0	4.00	34.0	49.0	26.0	4.80	7.00	3.71	15.5	157	107	7.00	22.4	0.68	2.00	5.00	1.00	0.66	1.66	0.33	2.66	1.00
4	0.87	0.00	2.00	11.0	4.00	34.0	49.0	26.0	4.80	7.00	3.71	15.5	157	107	7.00	22.4	0.68	4.00	3.00	2.00	1.33	1.00	0.66	3.00	1.00
5	0.66	1.00	1.00	19.0	6.00	82.0	61.0	46.0	7.45	5.55	4.18	17.1	287	165	11.0	26.0	0.57	2.00	2.00	2.00	1.00	1.00	1.00	3.00	0.00
6	0.90	1.00	1.00	19.0	6.00	82.0	61.0	46.0	7.45	5.55	4.18	17.1	287	165	11.0	26.0	0.57	3.00	2.00	1.00	1.00	0.66	0.33	2.00	1.00
7	0.91	1.00	1.00	19.0	6.00	82.0	61.0	46.0	7.45	5.55	4.18	17.1	287	165	11.0	26.0	0.57	4.00	1.00	4.00	1.33	0.33	1.33	3.00	1.00
8	0.51	1.00	1.00	19.0	6.00	82.0	61.0	46.0	7.45	5.55	4.18	17.1	287	165	11.0	26.0	0.57	3.00	2.00	1.00	1.00	0.66	0.33	2.00	0.00
9	0.67	1.00	1.00	19.0	6.00	82.0	61.0	46.0	7.45	5.55	4.18	17.1	287	165	11.0	26.0	0.57	1.00	2.00	3.00	0.33	0.66	1.00	2.00	0.00
10	0.70	1.00	1.00	19.0	6.00	82.0	61.0	46.0	7.45	5.55	4.18	17.1	287	165	11.0	26.0	0.57	2.00	1.00	2.00	1.00	0.50	1.00	2.50	0.00
11	0.49	0.00	3.00	24.0	6.00	82.0	82.0	43.0	4.56	4.56	2.39	11.5	323	204	18.0	17.9	0.63	3.00	2.00	2.00	1.50	1.00	1.00	3.50	1.00
12	0.60	0.00	3.00	24.0	6.00	82.0	82.0	43.0	4.56	4.56	2.39	11.5	323	204	18.0	17.9	0.63	3.00	3.00	1.00	1.00	1.00	0.33	2.33	1.00
13	0.71	0.00	3.00	24.0	6.00	82.0	82.0	43.0	4.56	4.56	2.39	11.5	323	204	18.0	17.9	0.63	2.00	1.00	2.00	1.00	0.50	1.00	2.50	1.00
14	0.67	0.00	3.00	24.0	6.00	82.0	82.0	43.0	4.56	4.56	2.39	11.5	323	204	18.0	17.9	0.63	2.00	1.00	2.00	1.00	0.50	1.00	2.50	1.00
15	0.53	0.00	3.00	24.0	6.00	82.0	82.0	43.0	4.56	4.56	2.39	11.5	323	204	18.0	17.9	0.63	6.00	3.00	1.00	2.00	1.00	0.33	3.33	0.00
16	0.56	0.00	3.00	24.0	6.00	82.0	82.0	43.0	4.56	4.56	2.39	11.5	323	204	18.0	17.9	0.63	5.00	1.00	3.00	1.66	0.33	1.00	3.00	1.00
17	0.53	0.00	1.00	16.0	6.00	57.0	48.0	36.0	7.13	6.00	4.50	17.6	220	124	8.00	27.5	0.56	2.00	1.00	3.00	1.00	0.50	1.50	3.00	0.00
18	0.44	0.00	1.00	16.0	6.00	57.0	48.0	36.0	7.13	6.00	4.50	17.6	220	124	8.00	27.5	0.56	4.00	2.00	1.00	1.33	0.66	0.33	2.33	0.00
19	0.19	0.00	1.00	16.0	6.00	57.0	48.0	36.0	7.13	6.00	4.50	17.6	220	124	8.00	27.5	0.56	4.00	2.00	1.00	1.33	0.66	0.33	2.33	0.00
20	0.43	0.00	1.00	16.0	6.00	57.0	48.0	36.0	7.13	6.00	4.50	17.6	220	124	8.00	27.5	0.56	4.00	1.00	2.00	1.33	0.33	0.66	2.33	0.00
21	0.33	0.00	1.00	16.0	6.00	57.0	48.0	36.0	7.13	6.00	4.50	17.6	220	124	8.00	27.5	0.56	2.00	1.00	1.00	0.66	0.33	0.33	1.33	0.00
22	0.39	0.00	1.00	16.0	6.00	57.0	48.0	36.0	7.13	6.00	4.50	17.6	220	124	8.00	27.5	0.56	3.00	1.00	2.00	1.00	0.33	0.66	2.00	0.00
23	0.76	1.00	1.00	12.0	6.00	56.0	23.0	38.0	6.22	2.56	4.22	13.0	177	100	9.00	19.6	0.56	2.00	4.00	1.00	1.00	2.00	0.50	3.50	0.00
24	0.16	1.00	1.00	12.0	8.00	56.0	23.0	38.0	6.22	2.56	4.23	13.0	177	100	9.00	19.6	0.56	4.00	4.00	3.00	1.00	1.00	0.75	2.75	0.00
25	0.30	1.00	1.00	12.0	8.00	56.0	23.0	38.0	6.22	2.56	4.23	13.0	177	100	9.00	19.6	0.56	3.00	3.00	1.00	1.00	1.00	0.33	2.33	0.00
26	0.50	1.00	1.00	12.0	8.00	56.0	23.0	38.0	6.22	2.56	4.23	13.0	177	100	9.00	19.6	0.56	2.00	2.00	2.00	1.00	1.00	1.00	3.00	0.00
27	0.34	1.00	1.00	12.0	8.00	56.0	23.0	38.0	6.22	2.56	4.23	13.0	177	100	9.00	19.6	0.56	3.00	4.00	2.00	1.00	1.33	0.66	3.00	0.00
28	0.59	1.00	1.00	12.0	8.00	56.0	23.0	38.0	6.22	2.56	4.23	13.0	177	100	9.00	19.6	0.56	3.00	2.00	1.00	1.00	0.66	0.33	2.00	1.00
29	0.27	1.00	1.00	12.0	8.00	56.0	23.0	38.0	6.22	2.56	4.23	13.0	177	100	9.00	19.6	0.56	4.00	4.00	1.00	1.00	1.00	0.25	2.25	0.00

Table 2 Predicted p-values with the sigmoid function (original training set and test set)

Actual p-value	Predicted p-value			
	Run 1	Run 2	Run 3	Run 4
0.90	0.897	0.907	0.905	0.859
0.90	0.925	0.933	0.942	0.901
0.66	0.713	0.725	0.739	0.753
0.91	0.830	0.866	0.844	0.867
0.67	0.601	0.685	0.651	0.619
0.49	0.709	0.719	0.718	0.754
0.71	0.708	0.692	0.694	0.671
0.53	0.242	0.161	0.284	0.208
0.53	0.606	0.579	0.585	0.594
0.19	0.407	0.398	0.396	0.399
0.33	0.433	0.425	0.413	0.387
0.76	0.585	0.577	0.633	0.675
0.30	0.272	0.261	0.276	0.279
0.34	0.295	0.291	0.313	0.331
0.27	0.195	0.183	0.191	0.169
MSE	0.0165	0.0197	0.0134	0.0172

$\chi^2 = 5.23$, n.s., 0.01 level, df 4

Table 3 Predicted p-values without the sigmoid function (original training set and test set)

Actual p-value	Predicted p-value			
	Run 1	Run 2	Run 3	Run 4
0.90	0.837	0.886	0.878	0.889
0.90	0.924	0.987	0.955	0.983
0.66	0.691	0.717	0.698	0.747
0.91	0.868	0.905	0.863	0.856
0.67	0.587	0.678	0.635	0.597
0.49	0.688	0.685	0.678	0.751
0.71	0.704	0.686	0.704	0.679
0.53	0.288	0.233	0.392	0.257
0.53	0.583	0.550	0.512	0.528
0.19	0.411	0.410	0.401	0.402
0.33	0.423	0.435	0.420	0.401
0.76	0.628	0.547	0.561	0.635
0.30	0.313	0.277	0.284	0.346
0.34	0.323	0.302	0.286	0.348
0.27	0.229	0.178	0.167	0.191
MSE	0.0128	0.0169	0.0112	0.0161

$\chi^2 = 5.362$, n.s., 0.01 level, df 4

new set of initial weights taking random values between -0.5 and 0.5. This form of training was repeated three times and the results of testing the 14 items are also shown in Tables 2 and 3 (Runs 2-4). Additionally, the training set and test set were switched i.e. the new training set consisted of the 14 items in the original test set and the new test set consisted of the 15 items in the original training set. The training of the two networks was repeated using exactly the same set of initial weights used in the original 4 runs and the test results are shown in Tables 4 and 5. The mean squared error (MSE) computed from the actual and predicted p-values are also shown for each run in Tables 2-5. The MSE provides a measure for evaluating the consistency in the performance of the networks for the multiple training runs and also provides a measure for comparing the performances of the two networks. The small MSE values and the little difference in the MSE values obtained not only demonstrate how effective the networks are in predicting the item difficulty but also the consistency in the predictions from run to run. The average of the MSE values obtained for the networks with and without the sigmoid function (0.0129 and 0.0133 respectively) also show that the performance of the network with the sigmoid function is marginally superior to that of the network without the sigmoid function.

A correlation analysis and the Kruskal-Wallis test were also employed to assess how accurately the neural network predicted the item difficulty values. The correlation matrices corresponding to the runs in Tables 2-5 are shown in Tables 6-9 and all correlation coefficients reported in the tables are significant at the 0.01 level for a one-tailed test. The Kruskal-Wallis test, a nonparametric alternative to the one-way analysis of variance, was utilized to determine whether there was a difference between the actual p-values and the predicted p-values for different test runs. The Kruskal-Wallis test was selected because neither normality of distribution nor homogeneity of variance for the groups of p-values under study could be assumed. The Kruskal-Wallis test statistic is calculated from the sums of ranks for the different samples of p-values, and the interpretation in this paper is that of a hypothesis of equal means. For *df* 4 at the 0.01 level, the critical tabled value for the chi-square statistic is 13.277. For Tables 2-5, the calculated statistic for each table of values is smaller than the tabled value; therefore, it can be concluded that no significant difference in the ranks has been established and further that the predicted p-values are statistically equal to the actual p-values.

Table 4 Predicted p-values with the sigmoid function (switched training set and test set)

Actual p-value	Predicted p-value			
	Run 1	Run 2	Run 3	Run 4
0.76	0.773	0.818	0.814	0.775
0.87	0.887	0.878	0.882	0.908
0.90	0.862	0.780	0.739	0.769
0.51	0.526	0.357	0.491	0.512
0.70	0.686	0.644	0.622	0.651
0.60	0.581	0.597	0.547	0.613
0.67	0.716	0.699	0.693	0.737
0.56	0.637	0.600	0.535	0.584
0.44	0.239	0.224	0.232	0.236
0.43	0.292	0.302	0.267	0.284
0.39	0.323	0.341	0.312	0.330
0.16	0.267	0.336	0.298	0.332
0.50	0.407	0.433	0.418	0.439
0.59	0.689	0.677	0.542	0.619
MSE	0.0074	0.0096	0.0102	0.0091

$\chi^2 = 5.234$, n.s., 0.01 level, df 4

Table 5 Predicted p-values without the sigmoid function (switched training set and test set)

Actual p-value	Predicted p-value			
	Run 1	Run 2	Run 3	Run 4
0.76	0.829	0.852	0.864	0.805
0.87	0.861	0.863	0.843	0.936
0.90	0.846	0.729	0.628	0.662
0.51	0.553	0.569	0.486	0.539
0.70	0.687	0.666	0.625	0.627
0.60	0.605	0.592	0.501	0.576
0.67	0.724	0.713	0.696	0.717
0.56	0.606	0.652	0.553	0.488
0.44	0.238	0.222	0.221	0.200
0.43	0.301	0.323	0.286	0.254
0.39	0.332	0.351	0.333	0.299
0.16	0.267	0.341	0.294	0.315
0.50	0.394	0.439	0.425	0.441
0.59	0.534	0.613	0.433	0.464
MSE	0.0072	0.0107	0.0158	0.0154

$\chi^2 = 6.016$, n.s., 0.01 level, df 4

Table 6 Correlation matrix with the sigmoid function (original training set and test set)

		1	2	3	4	5
1	Actual p-value	-	0.850	0.836	0.879	0.726
2	Predicted p-value, Run 1		-	0.991	0.995	0.987
3	Predicted p-value, Run 2			-	0.991	0.987
4	Predicted p-value, Run 3				-	0.991
5	Predicted p-value, Run 4					-

all correlations are significant for df 13, $p<0.01$ for a one tailed test

Table 7 Correlation matrix without the sigmoid function (original training set and test set)

		1	2	3	4	5
1	Actual p-value	-	0.879	0.856	0.896	0.856
2	Predicted p-value, Run 1		-	0.890	0.981	0.989
3	Predicted p-value, Run 2			-	0.983	0.984
4	Predicted p-value, Run 3				-	0.975
5	Predicted p-value, Run 4					-

all correlations are significant for df 13, $p<0.01$ for a one tailed test

Table 8 Correlation matrix with the sigmoid function (switched training set and test set)

		1	2	3	4	5
1	Actual p-value	-	0.919	0.877	0.898	0.888
2	Predicted p-value, Run 1		-	0.987	0.969	0.982
3	Predicted p-value, Run 2			-	0.984	0.991
4	Predicted p-value, Run 3				-	0.991
5	Predicted p-value, Run 4					-

all correlations are significant for df 12, $p<0.01$ for a one tailed test

Table 9 Correlation matrix without the sigmoid function (switched training set and test set)

		1	2	3	4	5
1	Actual p-value	-	0.922	0.857	0.839	0.845
2	Predicted p-value, Run 1		-	0.977	0.952	0.953
3	Predicted p-value, Run 2			-	0.966	0.961
4	Predicted p-value, Run 3				-	0.976
5	Predicted p-value, Run 4					-

all correlations are significant for df 12, $p<0.01$ for a one tailed test

VI Discussion

The ability of the neural network to predict the p-values is highly dependent on the training set. The training set must be large enough to be representative of the data in order to approximate the desired input/output mapping during training. In the experiments conducted, the training set was relatively small; nevertheless, the prediction of the item difficulties was quite reasonable. A significant improvement in the prediction of the p-values by the neural networks can, therefore, be expected when a larger training set becomes available. No prior assumptions of the functional mapping between the 24 variables and the corresponding p-values, the significance of the variables, or the relationships between the variables were made. The functional mappng was approximated by the network during training with a part of the available data.

The next phase of our research will involve the identification of variables or types of variables for which predictions are most sensitive. By dropping variables and comparing the magnitude of change in predicted item difficulty, it should be possible to drastically reduce the number of predictor variables from 24 to a more manageable number and still maintain close concurrence between the actual and predicted values.

VII Conclusion

This paper focused on developing a neural network approach to predict item difficulty in a standardized reading comprehension test. The results obtained from the two backpropagation neural networks designed for the prediction problem clearly demonstrate that the networks can consistenily predict item difficulty with a high degree of success. The results of training an ANN to predict item difficulty in a reading comprehension test have direct application to pretesting and provide application to other items. The use of ANNs should further inform the testing community of what determines item difficulty and provide a basis for generalizing about how selected variables affect item difficulty.

VIII References

Bejar, I. 1983: Subject matter experts' assessment of item statistics. *Applied Psychological Measurement 7*, 303-310.

Drum, P.A., Calfee, R.C., and Cook, L.K. 1981: The effects of surface structure variables on performance in reading comprehension tests. *Reading Research Quarterly 16*, 486-514.

Educational Testing Service 1990: Test of English as a Foreign Language, Form 3LTF6. Princeton, N.J: Author.

Embretson, S.E., and Wetzel, C.D. 1987: Component latent trait models for paragraph comprehension tests. *Applied Psychological Measurement 11*, 175-193.

Grossberg, S. 1987: Competitive learning: From interactive activation to Adaptive resonance. *Cognitive Science 11*, 23-63.

Gupta, L., Sayeh, M.R., & Tammana, R. 1990: An artificial neural network approach to robust shape classification. *Pattern Recognition 23-6*, 563-568.

Gupta, L., & Upadhye, A.M. 1991: Non-linear alignment of neural net outputs for partial shape classification. *Pattern Recognition 24-10*, 943-948.

Gupta, L., Wang, J., Charles, A., & Kisatsky, P. 1992: Prototype selection rules for neural network training. *Pattern Recognition 25-11*, 1401-1408.

Gupta, L., Wang, J., Charles, A., & Kisatsky, P. in press: Three-layer perceptron based classifiers for the partial shape classification problem," *Pattern Recognition*.

Hecht-Nielsen, R. 1987: Counterpropagation networks. *Applied Optics 26(3)*, 4979-84.

Hecht-Nielsen, R. 1991: *Neurocomputing*, Addison-Wesley.

Hopfield, J.J., & Tank, D.W. 1985: Neural computations of decisions in optimization problems. *Biological Cybernetics 52*, 141-52.

Kohonen, T. 1988: *Self-organization and associative memory*. New York: Springer.

Lapedes, A., and Farber, R. 1987: *Nonlinear signal processing using neural networks: prediction and systems modeling* (LA-UR-87-2662). Los Alamos National Lab Technical Report. Los Alamos, N.M: Los Alamos National Lab.

Lykins, S. and Chance, D. 1992: Comparing artificial neural networks and multiple regression for predictive application. *Proceedings of the Eight Annual Conference on Applied Mathematics*, Edmond, OK: University of Central Oklahoma, 155-169

Nelson, M.M., and Illingworth, W.T. 1991: *A practical guide to neural nets*. Reading, M.A: Addision-Wesley.

Parker, D.B. 1982: *Learning logic*. Invention report S81-64, File 1, Office of Technology Licensing, Stanford University, Stanford, CA.

Perkins, K., and Brutten, S.R. 1991: A model of ESL reading comprehension difficulty. Paper presented at the meeting of the Language Testing in Europe International Conference, Jyvaskyla, Finland.

Rumelhart, D.E. & McClelland, J.D. 1986: *Parallel and distributed processing, I, II*, MIT Press, Cambridge, Mass.

Scheuneman, J., Gerritz, K., and Embretson, S. 1989 March: Effects of prose complexity on achievement test item difficulty. Paper presented at the meeting of the American Educational Research Association, San Francisco.

Scheuneman, J., Gerritz, K. 1990: Using item differential item functioning procedures to explore sources of item difficulty and group performance characteristics. *Journal of Educational Measurement 27*, 109-131

Sharda, R. and Patil, R.B. 1992: Connectionist approach to time-series prediction: an empirical test. *Journal of Intelligent Manufacturing 3, 317-323.*

Werbos, P.J. 1974: Beyond regression: new tools for prediction and analysis in the behavioral sciences. Masters' thesis, Harvard University.