Using Back-Propagation (BPN) neural networks for basic knowledge of the English language diagnosis

John Vrettaros[1], George Vouros[2], Athanasios S. Drigas[1]
[1] N.C.S.R. DEMOKRITOS,
Institute of Informatics and Telecommunications
Net Media Lab
Ag. Paraskevi, 15310, Athens, Greece
[2] Aegean University, Info and Communication Systems Eng,
83200, Karlovassi, Samos, Greece
{dr,jvr}@iit.demokritos.gr, georgev@aegean.gr

Abstract. This article studies the expediency of using neural networks technology and the development of back-propagation networks (BPN) models for modeling automated evaluation of the answers and progress of deaf students' that possess basic knowledge of the English language and computer skills, within a virtual e-learning environment. The performance of the developed neural models is evaluated with the correlation factor between the neural networks' response values and the real value data as well as the percentage measurement of the error between the neural networks' estimate values and the real value data during its training process and afterwards with unknown data that weren't used in the training process.

Keywords: Neural Networks, Back-propagation Network models, diagnosis.

1 Introduction

One of the most promising branches of current educational research is student diagnosis which according to VanLehn [10] is the process of inferring students' internal characteristics from their recognizable behavior. Student diagnosis is considered more than necessary as according to learning theories adaptive learning is a fruitful approach if one wishes to obtain effective tutoring. That is the reason why one of the latest trends in educational software development is attempting to implement a simulation of the way that a human teacher adapts his tutoring to the individualization of each student [7]. In order for this kind of intelligence to be achieved, researchers have adopted many Artificial Intelligence methods. The most famous among them are neural networks, fuzzy logic as well as several search methods such as genetic algorithms.

Neural networks are on the top of the researchers' choice since they provide a system the ability to recognize patterns, to derive meaning from vague data and to identify matching in similar cases [8]. Fuzzy set theory is widely used since it can deal in a reliable way with human uncertainty and it obtains smooth modelling of human decision – making. Genetic algorithms are ideal for efficient expert knowledge representation. Finally, Neuro – Fuzzy synergism is getting more and more popular in this area since it seems to overcome obstacles that come up when each of the methods involved is solely applied [9]. Below we present several typical examples of the application of these methods in student's diagnosis.

Weon and Kim [5], implemented a system aiming at individual learning evaluation by making use of fuzzy linguistic variables representing each question. Finally, assessment came up taking into consideration the membership degree of uncertainty factors.

Vrettaros et al. [6] introduced a diagnostic system of taxonomies using fuzzy logic. Filled up questionnaires were processed in order for students' classification into one of the predefined knowledge levels to be achieved according to the given answers.

Stathakopoulou et al. [11] attempted to infer students' individual characteristics and use them in order not only to create but also to update the student model. This neural network - based fuzzy model is consisted of a fuzzy component and a neural network trained through actual students' profiles.

Hommsi et al. [4], presented an Adaptive and Intelligent Web – Based Educational System (AIWBES) based on Fuzzy – ART2 neural network and Hidden Markov Model (HMM), which is a stochastic method. The goal of the system described by the authors was to assess the learners' knowledge level and to obtain students' classification in one of six different levels taking into account several parameters.

2 Description of the assessment process

In general, the aim of the e-learning environment that is studied is the realization of an e-educational system for the evaluation of the teaching of the English language to deaf people. The crucial point in the definition of the theoretical and technical aspects of the correlation between the e-learning models' subsystems is the definition of the configuration and the characteristics of their corresponding interconnections. According to the structural and operational details of the e-learning procedure the performance factor of the expert system and the collaboration of the e-learning model is the encoding and the contents of the inputs and outputs of the expert system as well as the structure, the standardization and the content of the database questions that require further attention and skilful handling.

The evaluation procedure for the teaching of English to deaf students relies on the attainment of the ESOL (English for Speakers of Other Languages) models (levels 1 and 2). These levels comprise five sections, which in ascending order are [A], [B], [C], [D] and [E]. Section [A] represents the Letter Recognition and their Alphabetical Order, section [B] represents the Spelling and the Vocabulary, section [C] represents the Grammar and the Sentence Structure, section [D] represents the Reading and section [E] represents the Writing.

According to the ESOL e-learning environment specifications the input / output parameters of an expert system can be determined without doubt, while simultaneously their translation is simple and quite straightforward.

As far as the input is concerned, there are five pairs of parameters in total and per question, which are: $a = \ a_{val}, a_{rel}$, $b = \ b_{val}, b_{rel}$, $c = \ c_{val}, c_{rel}$, $d = \ d_{val}, d_{rel}$ and $e = \ e_{val}, e_{rel}$. In other words, every pair corresponds to a language section of a certain level. Parameter a describes the recognition of the letters and the alphabetical order of section A , parameter b relates to the spelling/vocabulary of section B , parameter c represents the grammar/sentence structure for section C , the corresponding parameter for reading in section D is d , while the writing skill of section E is quantified with parameter e. The abbreviation index val (value) represents the evaluation of the specific section based on a particular response, while the abbreviation index rel (relevance) recognizes the level of relevance /weight of a specific question with respect to the contents of a section.

The evaluation values of the input parameters a_{val} , b_{val} , c_{val} , d_{val} and e_{val} originate from the range $S = \ -1 \ \bigcup \ 0, \ 1$. When a section is not examined by a question of the corresponding parameter, then the range is characterized by the value -1. A wrong answer according to a certain section leads to a corresponding value of zero (0), while the value of a sections' parameter is (1) if the chosen answer is correct according to that section. Accordingly, answers that are partially correct based on some sections have their values placed in between.

On the other hand, someone can argue that the relevance parameters a_{rel} , b_{rel} , c_{rel} , d_{rel} and e_{rel} characterize the question itself rather than its possible answers. Although this is true, the negotiation with relative parameters as part of a given answer is handy from an evaluation viewpoint. As a result, the relevance/weight is considered range between $0, \ 1$, where the value zero (0) or values close to zero mean low relevance, while the value one (1) or values close to one mean high relevance. Accordingly, the other weight values range somewhere in between. It must be pointed out, that the relevance parameters are the same for all the answers of a specific question.

The most skilful method for the provision of information in an expert system in relation with these sections is the sequence in array format of the ten values for the parameters of the input pairs:

a_{val}	a_{rel}	b_{val}	b_{rel}	c_{val}	c_{rel}	d_{val}	d_{rel}	e_{val}	e_{rel}

As an example, let's consider that a question shows low relevance in section A, high relevance in section C and medium relevance in section B. Moreover, let's assume that the question under consideration does not include information about sections D and E. Now let's consider an answer to the previous question, which is correct according to section A, partially correct according to section C, wrong according to section B. Obviously, it does not contain information relating to sections D and E. Such an answer ends in a ten value sequence, which comprises elements that are defined in the range $S = \ -1 \cup 0,\ 1$. In addition, it is obvious that the aforementioned ten value sequence can be directly encoded as an arithmetic string similar to the one below:

a_{val}	a_{rel}	b_{val}	b_{rel}	c_{val}	c_{rel}	d_{val}	d_{rel}	e_{val}	e_{rel}
1	0.1	0	0.5	0.7	0.9	-1	0	-1	0

This way, this particular arithmetic string can easily be used as an input to an expert system.

As far as the output is concerned, from observations and/or monitoring of the operational and correlating characteristics of the expert system, the conclusion that is derived is that that output parameters of the system are six (6) namely y_1, y_2, y_3, y_4, y_5 and y_6. The first five parameters are the evaluation/estimation of the lingual skills per section, while the sixth parameter represents the total estimation of the user for the sum of all of the lingual skills, as follows:

y_1 = letter recognition and alphabetical order skills

y_2 = spelling / vocabulary skills

y_3 = grammar / sentence structure skills

y_4 = reading skills

y_5 = writing skills

y_6 = total lingual skills **(essentially, it is a weighted average of y_1 - y_5, *representing a general estimation of the linguistic level of the student, as a professional pedagogic would have determined in a real case scenario*).**

It is obvious that the output parameters are continuous. Due to the fact that the expert system outputs represent certain estimation in connection with a specific lingual section, the evaluation is considered to be normalized in the range $0,\ 1$. The translation of the final arithmetic values is simple: zero means no lingual skills, one means perfect lingual skills, while all the other lingual skills levels may be estimated with similar arithmetic insertions. The output values, which are already arithmetically encoded, may be inserted in the e-learning environment as a six value arrayed sequence:

y_1	y_2	y_3	y_4	y_5	y_6

Let's assume that the next estimation is real for a particular student:

0.6 = letter recognition and alphabetical order skills
0.4 = spelling / vocabulary skills
0.2 = grammar / sentence structure skills
0.5 = reading skills
0.3 = writing skills
0.4 = total lingual skills

This particular six value arrayed sequence which comprises continuous elements could be directly encoded as an arithmetic string similar to the one below:

y_1	y_2	y_3	y_4	y_5	y_6
0.6	*0.4*	*0.2*	*0.5*	*0.3*	*0.4*

This way the final outputs are available straight away to rest of the e-learning environment.

The above (according to the ESOL specifications) skilful encoding and inputs and outputs content as well as the structure, standardization and content of the database questions is based on the use of neural networks technology for the e-learning expert system modeling for the automated estimation of the evaluation values of the teaching of English to deaf people.

The neural networks technology has been applied with success to many estimation problems with similar input/output characteristics. This study examines the expediency of applying this technology for modeling the automated evaluation of the deaf students' answers in the form of questions divided in five sections within the e-learning environment of the expert system.

In this study, section 2 provides a description of the neural networks characteristics and details about the development of a typical BPN model. In section 3, various Back-Propagation Networks (BPN) type neural networks models are developed for the automated evaluation of the deaf students' education and progress and their results are presented, determining hence, the expediency of the use of neural networks technology in such problems. In section 4, general neural networks technological issues are presented about generalization and over-fitting for the successful development and realization of effective models. Finally, in section 5 conclusions are presented and ideas are suggested for further research.

3 Development of back-propagation network (BPN) type neural networks for the automated evaluation of deaf students' progress

In this study, the general schema of input/output data for supervised learning and their connection with neural networks results from the definition of ten (10) variables namely, v, v', w, w', x, x', y, y', z and z', which belong in the range $S = -1 \cup 0,\ 1$ for the input and with the definition of six (6) variables, namely,

h, i, j, k, l and n that belong in the range $0, 1$, for the output. Hence, each input/output model from the total education data could be similar to the next model (where the symbol $\varnothing$ denotes the blank value):

Ten (10) Input Values	a	b	c	d	e	
	(v, v')	(w, w')	(x, x')	(y, y')	(z, z')	
Six (6) Output Values	y_1	y_2	y_3	y_4	y_5	y_6
	h *if* $v \neq -1$ *else* $\varnothing$	i *if* $w \neq -1$ *else* $\varnothing$	j *if* $x \neq -1$ *else* $\varnothing$	k *if* $y \neq -1$ *else* $\varnothing$	l *if* $z \neq -1$ *else* $\varnothing$	n

A sample of real values for seven (7) models from the sum of the education data is depicted in the following Table 1.

Table 1. A sample of supervised learning education data

pattern (#)			1	2	3	4	5	6	7
Input Variables	a	a_{val}	0.6	0.1	1.0	0.9	0.6	0.1	0.1
		a_{rel}	0.5	0.5	0.5	0.7	0.7	0.7	0.2
	b	b_{val}	0.8	0.7	0.1	-1	-1	0.8	-1
		b_{rel}	0.8	0.8	0.8	0	0	0.6	0
	c	c_{val}	0.6	0.8	0.2	-1	-1	0.9	0,1
		c_{rel}	0.1	0.1	0.1	0	0	0.5	0.8
	d	d_{val}	-1	-1	-1	0.9	0.1	0.6	-1
		d_{rel}	0	0	0	0.5	0.5	0.4	0
	e	e_{val}	-1	-1	-1	0.6	0.8	0.2	0.9
		e_{rel}	0	0	0	0.3	0.3	0.1	0.3
Output Variables	y_1		0.9	0.2	0.3	0.5	0.8	0.4	0.7
	y_2		0.7	0.8	0.2	$\varnothing$	$\varnothing$	0.1	$\varnothing$
	y_3		0.4	0.5	0.2	$\varnothing$	$\varnothing$	0.1	0.5
	y_4		$\varnothing$	$\varnothing$	$\varnothing$	0.9	0.0	0.1	$\varnothing$
	y_5		$\varnothing$	$\varnothing$	$\varnothing$	0.6	0.2	0.6	0.8

	y_6	0.1	0.5	0.7	0.1	0.2	0.9	0.7

It is obvious that the information that is contained in such education data, should be gathered and processed by a professional pedagogist, due to the fact that such a person seems to be the most suitable and competent one to create the aforementioned content.

For the development of typical back-propagation network (BPN) neural networks models, this study used the modeling and simulation software package Predict by NeuralWare, which is described in reference [1]. This software package provides the necessary mechanisms for the automated modifications and executes the scaling and limiting procedures to the users' input/output data. The basic idea for the modification of data, for the better identification of the statistical distribution of the input variables for the neural networks' training and for the maximization of its performance, is described in references [2, 3].

The software package Predict by NeuralWare demands that the presentation of the variables of the input vector, x, to be of the following pattern:

$$x = a_{val} \quad a_{rel} \quad b_{val} \quad b_{rel} \quad c_{val} \quad c_{rel} \quad d_{val} \quad d_{rel} \quad e_{val} \quad e_{rel}$$

with values for the evaluation parameters a_{val}, b_{val}, c_{val}, d_{val} and e_{val} that originate from the aforementioned range $S = -1 \cup 0, 1$ and with values for the relevance parameters a_{rel}, b_{rel}, c_{rel}, d_{rel} and e_{rel} that originate from the aforementioned range $0, 1$. For the variables of output vector, y, the software package Predict by NeuralWare demands that the presentation is of the following pattern:

$$y = y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6$$

with values that originate from the aforementioned normalized range $0, 1$.

BPN network Architecture #1: The BPN network architecture #1 is described below along with the respective transfer functions.

Nodes	**Hid 1**	**Hid 2**	**Hid 3**	**Out 1**	**Out 2**	**Out 3**	**Out 4**	**Out 5**	**Out 6**
Transfer Functions $[f_h \; I \; , f_o \; I \;]$	Tanh	Tanh	Tanh	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid

For the construction of the network several kinds of transformations of input and output variables were used. Several categories of continuous transformations are functions with type *Linear*, *Inverse* (1/x), *ln* (ln x/(1-x)), *Logical* και *Log*.

Continuous (Tmin Tmax Method Imin Imax)

Tmin and Tmax stand for minimum and maximum transformed values respectively, Method determines the way that data are mapped in the range of the transformation, while Imin and Imax stand for minimum and maximum variables when using unprocessed data. Transformation Islit gives value Tmax if String appears in data while it gives value Tmin if not.

The following holds about Logical transformation,

$$y = \begin{cases} T_{\max} & if \quad x \geq \dfrac{I_{\min} + I_{\max}}{2} \\ T_{\min} & otherwise \end{cases} \qquad (1)$$

The performance of the BPN network Architecture #1 is evaluated by comparing the results of the trained neural network for all of the input/output data, using the recall phase. Table 2 presents the real values of the 36 output data, the corresponding values of the BPN#1 neural network and the corresponding indication of the correct or wrong answer (estimation) of the BPN#1 neural network (considering that the real input value 10 and the real output values 10, 14 and 20 correspond to initial values -1):

Table 2. BPN network Architecture #1 - (Recall Phase Results) - Single Select Questions

	Test#, Question #	Real Output Values						Output Values BPN#1						BPN#1
		Y1	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	**Y1**	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	**Correct or Wrong Indication**
1	t1,q2	0	10	10	0	10	0	0	14	20	0	10	0	**Correct**
2	t1,q2	1	10	10	1	10	1	1.1	14	20	1	10	1	**Correct**
3	t1,q2	0	10	10	0	10	0	0	14	20	0	10	0	**Correct**
4	t1,q2	0.3	10	10	0.3	10	0.3	0.2	14	20	0.3	10	0.3	**Correct**
5	t1,q3	0	0	0	0	0	0	-0	-0	0	0	0	0	**Correct**
6	t1,q3	0	0	0.5	0	0	0.2	-0	-0	0	0	0	0	**Wrong**
7	t1,q3	0	0	0	0	0	0	-0	-0	0	0	0	0	**Correct**
8	t1,q3	1	1	1	1	1	1	1	0.9	1.4	1	0.9	1	**Correct**

9	t1,q4	1	1	1	1	1	1	1	0.9	1.4	1	0.9	1	**Correct**
10	t1,q4	0	0	0	0	0	0	-0	-0	0	0	0	0	**Correct**
11	t1,q4	0	0	0.5	0	0	0.2	0	0	0.5	0	0	0.3	**Correct**
12	t1,q4	0	0	0.5	0	0	0.2	0	0	0.5	0	0	0.3	**Correct**
13	t1,q5	0	0	0	0	0	0	-0	-0	0	0	0	-0	**Correct**
14	t1,q5	1	1	1	1	1	1	0.8	1.1	0.6	1	0.9	1	**Correct ?**
15	t1,q5	0	0	0	0	0	0	-0	-0	0	0	0	-0	**Correct**
16	t1,q5	0	0	0	0	0	0	-0	-0	0	0	0	-0	**Correct**
17	t3,q1	10	0	0	0	10	0	10	0.1	0.1	0.2	10	0	**Correct ?**
18	t3,q1	10	0	0	0	10	0	10	0.1	0.1	0.2	10	0	**Correct ?**
19	t3,q1	10	0	0	0	10	0	10	0.1	0.1	0.2	10	0	**Correct ?**
20	t3,q1	10	1	1	1	10	1	10	0.1	0.1	0.2	10	0	**Correct ?**
21	t3,q2	10	0.5	0	0	10	0.2	10	0.5	0	0	10	0.2	**Correct**
22	t3,q2	10	0	0	0	10	0	10	0	0	0	10	0	**Correct**
23	t3,q2	10	1	1	1	10	1	10	1.2	0.7	1	10	1	**Correct ?**
24	t3,q2	10	0	0	0	10	0	10	0	0	0	10	0	**Correct**
25	t3,q3	10	0	0.5	0	10	0.2	10	0	0.3	0	10	0.1	**Correct ?**
26	t3,q3	10	1	1	1	10	1	10	0.9	1.1	1	10	1	**Correct**
27	t3,q3	10	0.5	0	0	10	0.2	10	0.4	0	0	10	0.2	**Correct**
28	t3,q3	10	0	0	0	10	0	10	-0	0	0	10	0	**Correct**
29	t3,q4	10	1	1	1	10	1	10	1.1	1.4	1	10	1	**Correct ?**
30	t3,q4	10	0	0	0	10	0	10	-0	0	0	10	0	**Correct**
31	t3,q4	10	0.5	0	0	10	0.2	10	0.4	0	0	10	0.2	**Correct**
32	t3,q4	10	0	0	0	10	0	10	-0	0	0	10	0	**Correct**
33	t3,q5	10	0	0.5	0	10	0.2	10	0	0.3	0	10	0.1	**Correct ?**

34	t3,q5	10	1	1	1	10	1	10	1.1	1.4	1	10	1	**Correct ?**
35	t3,q5	10	0	0	0	10	0	10	-0	0	0	10	0	**Correct**
36	t3,q5	10	0	0	0	10	0	10	-0	0	0	10	0	**Correct**

According to the table presented above (Table 2) the BPN network Architecture #1 truly learned 27 out of a total of 28 input/output models, hence a percentage of 27/28 = 94.43%, which is comparable with the correlation factor percentage during its training, which is 95.28%. In addition, from table 2 one can derive that the BPN network Architecture #1 estimates correctly almost 8 of the 8 output values.

BPN network Architecture #2: The BPN network architecture #2 is described below along with the output neurons and their respective transfer functions.

Nodes	**Out 1**	**Out 2**	**Out 3**	**Out 4**	**Out 5**	**Out 6**
Transfer Functions $\left[f_h \;\; I \;\; , f_o \;\; I \;\; \right]$	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid

For the construction of the network several kinds of transformations of the input and output variables were used. Several categories of continuous transformations are functions with type *Linear*, *Inverse, Rlogical (Reverse Logical), Tanh*, και *Log*. The respective formats of those transformations are presented below:

(Tmin Tmax Method Imin Imax)

Tmin and Tmax stand for minimum and maximum transformed values, Method determines the way that data are mapped in the range of the transformation, while Imin and Imax stand for minimum and maximum variables when using unprocessed data.

The following holds about Rl*ogical* transformation:

$$y = \begin{cases} T_{\max} & if \quad x \leq \dfrac{I_{\min} + I_{\max}}{2} \\ T_{\min} & otherwise \end{cases} \qquad (2)$$

The performance of the BPN network Architecture #2 is evaluated by comparing the results of the trained neural network for all of the input/output data, using the recall phase. Table 3 presents the real values of the 36 input/output data, the corresponding values of the BPN#2 neural network and the corresponding indication of the correct or wrong answer (estimation) of the BPN#1 neural network (considering that the real input value 10 and the real output values 9 and 10 correspond to initial values -1):

Table 3. BPN network Architecture #2 - (Recall Phase Results) - Single Select Questions

	Test#, Question#	Real Output Values						Output BPN#2						BPN#2 Correct or Wrong Indication
		Y1	Y2	Y3	Y4	Y5	Y6	Y1	Y2	Y3	Y4	Y5	Y6	
1	t1,q2	0	10	10	0	10	0	0	9.4	9.1	0	10	0	Correct
2	t1,q2	1	10	10	1	10	1	1.5	9.9	9	1	10	0.9	Correct ?
3	t1,q2	0	10	10	0	10	0	0	9.4	9.1	0	10	0	Correct
4	t1,q2	0.3	10	10	0.3	10	0.3	0.2	9.9	9.1	0.3	10	0.3	Correct
5	t1,q3	0	0	0	0	0	0	0	0	0.1	-0	0	0	Correct
6	t1,q3	0	0	0.5	0	0	0.2	0	0	0.1	-0	0	0	Wrong
7	t1,q3	0	0	0	0	0	0	0	0	0.1	-0	0	0	Correct
8	t1,q3	1	1	1	1	1	1	0.9	1	1.4	1	1	1	Wrong
9	t1,q4	1	1	1	1	1	1	0.9	1	1.4	1	1	1	Wrong
10	t1,q4	0	0	0	0	0	0	0	0	0.1	-0	0	0	Correct
11	t1,q4	0	0	0.5	0	0	0.2	0	0	0.5	-0	0	0.2	Correct
12	t1,q4	0	0	0.5	0	0	0.2	0	0	0.5	-0	0	0.2	Correct
13	t1,q5	0	0	0	0	0	0	0	0	0	-0	0	0	Correct
14	t1,q5	1	1	1	1	1	1	1.1	1	0.6	1	1.1	1	Wrong
15	t1,q5	0	0	0	0	0	0	0	0	0	-0	0	0	Correct
16	t1,q5	0	0	0	0	0	0	0	0	0	-0	0	0	Correct
17	t3,q1	10	0	0	0	10	0	9.7	0.1	0.1	0	10	0	Correct ?
18	t3,q1	10	0	0	0	10	0	9.7	0.1	0.1	0	10	0	Correct ?
19	t3,q1	10	0	0	0	10	0	9.7	0.1	0.1	0	10	0	Correct

20	**t3,q1**	**10**	**1**	**1**	**1**	**10**	**1**	**9.7**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
21	**t3,q2**	**10**	**0.5**	**0**	**0**	**10**	**0.2**	**9.7**	**0.5**	**0.2**	**0.1**	**10**	**0**	**Wrong**
22	**t3,q2**	**10**	**0**	**0**	**0**	**10**	**0**	**9.7**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
23	**t3,q2**	**10**	**1**	**1**	**1**	**10**	**1**	**10**	**9.4**	**2.6**	**1**	**10**	**1**	**Wrong**
24	**t3,q2**	**10**	**0**	**0**	**0**	**10**	**0**	**9.7**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
25	**t3,q3**	**10**	**0**	**0.5**	**0**	**10**	**0.2**	**9.1**	**0.4**	**0.8**	**0**	**10**	**0.4**	**Wrong**
26	**t3,q3**	**10**	**1**	**1**	**1**	**10**	**1**	**10**	**9.4**	**2.4**	**1**	**10**	**1**	**Wrong**
27	**t3,q3**	**10**	**0.5**	**0**	**0**	**10**	**0.2**	**9.3**	**0.6**	**0.2**	**0.1**	**10**	**0.1**	**Correct ?**
28	**t3,q3**	**10**	**0**	**0**	**0**	**10**	**0**	**9.2**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
29	**t3,q4**	**10**	**1**	**1**	**1**	**10**	**1**	**10**	**9.4**	**2.4**	**1**	**10**	**1**	**Wrong**
30	**t3,q4**	**10**	**0**	**0**	**0**	**10**	**0**	**9.2**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
31	**t3,q4**	**10**	**0.5**	**0**	**0**	**10**	**0.2**	**9.3**	**0.6**	**0.2**	**0.1**	**10**	**0.1**	**Correct ?**
32	**t3,q4**	**10**	**0**	**0**	**0**	**10**	**0**	**9.2**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
33	**t3,q5**	**10**	**0**	**0.5**	**0**	**10**	**0.2**	**9.1**	**0.4**	**0.8**	**0**	**10**	**0.4**	**Wrong**
34	**t3,q5**	**10**	**1**	**1**	**1**	**10**	**1**	**10**	**9.4**	**2.4**	**1**	**10**	**1**	**Wrong**
35	**t3,q5**	**10**	**0**	**0**	**0**	**10**	**0**	**9.2**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**
36	**t3,q5**	**10**	**0**	**0**	**0**	**10**	**0**	**9.2**	**0.1**	**0.1**	**0**	**10**	**0**	**Correct**

According to the table presented above the BPN network Architecture #2 truly learned 16 out of a total of 20 input/output models, hence a percentage of 16/20 = 80%, which is comparable to the correlation factor percentage during its training, which is 93.60%. In addition, from this table (Table 3) one can derive that the BPN network Architecture #2 estimates correctly almost 9 of the 16 output values, hence a percentage of 9/16=56.25%. The results of the assessment of BPN network Architecture #2 were not unexpected since in this architecture neurons in the hidden layer were not chosen.

BPN network Architecture #3: For the development of BPN network Architecture #3, the real values of the available output data were encoded in literals. The encoding used turns value 0 into literal zero, value -1 into literal N, value 0.1 into literal one, value 0.3 into literal 3, value 0.5 into literal 5 as it appears in Table 4 below. Very often, the use of literal encoding in neural networks is being adopted in order to enhance the training of the neural network separating or classifying in different categories each output.

Table 4: Inputs / Outputs of Single Select Questions

	Test#, Question #	**Input Variables**										**Output Variables**					
		Aeval	Arel	Beval	Brel	Ceval	Crel	Deval	Drel	Eeval	Erel	Y1	Y2	Y3	Y4	Y5	Y6
1	t1,q2	0	4	-1	0	-1	0	0	1	-1	0	zero	N	N	zero	N	zero
2	t1,q2	1	4	-1	0	-1	0	1	1	-1	0	one	N	N	one	N	one
3	t1,q2	0	4	-1	0	-1	0	0	1	-1	0	zero	N	N	zero	N	zero
4	t1,q2	0.3	4	-1	0	-1	0	0.3	1	-1	0	three	N	N	three	N	three
5	t1,q3	0	1	0	1	0	4	0	1	0	2	zero	zero	zero	zero	zero	zero
6	t1,q3	0	1	0	1	0	4	0	1	0	2	zero	zero	five	zero	zero	two
7	t1,q3	0	1	0	1	0	4	0	1	0	2	zero	zero	zero	zero	zero	zero
8	t1,q3	1	1	1	1	1	4	1	1	1	2	one	one	one	one	one	one
9	t1,q4	1	1	1	1	1	4	1	1	1	2	one	one	one	one	one	one
10	t1,q4	0	1	0	1	0	4	0	1	0	2	zero	zero	zero	zero	zero	zero
11	t1,q4	0	1	0	1	0.5	4	0	1	0	2	zero	zero	five	zero	zero	two
12	t1,q4	0	1	0	1	0.5	4	0	1	0	2	zero	zero	five	zero	zero	two
13	t1,q5	0	4	0	1	0	4	0	1	0	2	zero	zero	zero	zero	zero	zero
14	t1,q5	1	4	1	1	1	4	1	1	1	2	one	one	one	one	one	one
15	t1,q5	0	4	0	1	0	4	0	1	0	2	zero	zero	zero	zero	zero	zero
16	t1,q5	0	4	0	1	0	4	0	1	0	2	zero	zero	zero	zero	zero	zero
17	t3,q1	-1	0	0	1	0	4	0	2	-1	0	N	zero	zero	zero	N	zero
18	t3,q1	-1	0	0	1	0	4	0	2	-1	0	N	zero	zero	zero	N	zero
19	t3,q1	-1	0	0	1	0	4	0	2	-1	0	N	zero	zero	zero	N	zero
20	t3,q1	-1	0	0	1	0	4	0	2	-1	0	N	one	one	one	N	one
21	t3,q2	-1	0	0.5	2	0	4	0	2	-1	0	N	five	zero	zero	N	two
22	t3,q2	-1	0	0	2	0	4	0	2	-1	0	N	zero	zero	zero	N	zero
23	t3,q2	-1	0	1	2	1	4	1	2	-1	0	N	one	one	one	N	one
24	t3,q2	-1	0	0	2	0	4	0	2	-1	0	N	zero	zero	zero	N	zero
25	t3,q3	-1	0	0	4	0.5	2	0	1	-1	0	N	zero	five	zero	N	two
26	t3,q3	-1	0	1	4	1	4	1	1	-1	0	N	one	one	one	N	one
27	t3,q3	-1	0	0.5	4	0	2	0	1	-1	0	N	five	zero	zero	N	two
28	t3,q3	-1	0	0	4	0	2	0	1	-1	0	N	zero	zero	zero	N	zero
29	t3,q4	-1	0	1	4	1	2	1	1	-1	0	N	one	one	one	N	one
30	t3,q4	-1	0	0	4	0	2	0	1	-1	0	N	zero	zero	zero	N	zero
31	t3,q4	-1	0	0.5	4	0	2	0	1	-1	0	N	five	zero	zero	N	two
32	t3,q4	-1	0	0	4	0	2	0	1	-1	0	N	zero	zero	zero	N	zero
33	t3,q5	-1	0	0	4	0.5	2	0	1	-1	0	N	zero	five	zero	N	two
34	t3,q5	-1	0	1	4	1	2	1	1	-1	0	N	one	one	one	N	one
35	t3,q5	-1	0	0	4	0	2	0	1	-1	0	N	zero	zero	zero	N	zero
36	t3,q5	-1	0	0	4	0	2	0	1	-1	0	N	zero	zero	zero	N	zero

For the construction of the network, transformations T_{in} and T_o of the output and input variables were used. Several categories of continuous transformations are functions with type *Linear*, *Inverse*, *Tanh*, *Log*, *Logical*, *Islit* (*Is literal*) and *Fzraw*

(Fuzzy center on raw data). The respective formats of those transformations are presented in the table below:

Table 5. Kinds of transformations

Continuous	(Tmin	Tmax	Method	Imin	Imax)
Islit (Is literal)	(Tmin	Tmax	Literal (String)		
Fuzzy	(Tmin	Tmax	Fleft	Fcenter	Fright)

Tmin and Tmax stand for minimum and maximum transformed values, Method determines the way that data are mapped in the range of the transformation, while Imin and Imax stand for minimum and maximum variables when using unprocessed data. Transformation Islit gives value Tmax if String appears in data while it gives value Tmin if not. Fleft, Fcenter and Fright are respectively the left, center and right fuzzy sets that are presented in the figure below.

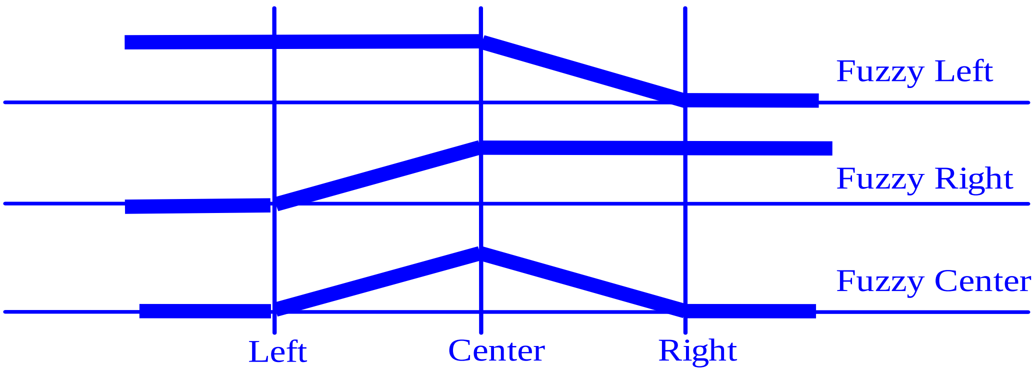

Figure 1: Fuzzy transformations

The developed BPN neural network used the first 28 of the 36 available data from Table 4 for its training and the remaining 8 were used for its evaluation trials. Its training was based on the maximization of the correlation with the adaptive gradient-descent technique and its final architecture is of this pattern: 26-2-22/0.9622. This pattern means that the final trained neural network comprises 26 neurons in the input (added inputs due to the modifications), 2 neurons in the hidden layer and 22 neurons in the output (added outputs due to the modifications) with a correlation factor of 0.9622.

The performance of the BPN network Architecture #3 is evaluated by comparing the results of the trained neural network for all of the input/output data, using the recall phase. Table 4 presents the real values of the 36 input/output data, the corresponding values of the BPN#3 neural network and the corresponding indication of the correct or wrong answer (estimation) of the BPN#3 neural network:

Table 6. BPN network Architecture #3 – Recall Phase Results - Single Select Questions

	Test#,	**Real Output Values (verbally encoded)**						**BPN#3 Output Values (verbally encoded)**						**BPN#3 Correct or Wrong Indication**
		Y1	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	**Y1**	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**	
1	t1,q2	zero	N	N	zero	N	zero	zero	n	n	zero	n	zero	**Correct**
2	t1,q2	one	N	N	one	N	one	one	n	n	one	n	one	**Correct**
3	t1,q2	zero	N	N	zero	N	zero	zero	n	n	zero	n	zero	**Correct**
4	t1,q2	three	N	N	three	N	three	three	n	n	three	n	three	**Correct**
5	t1,q3	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	**Correct**
6	t1,q3	zero	zero	five	zero	zero	two	zero	zero	zero	zero	zero	zero	**Correct**
7	t1,q3	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	**Correct**
8	t1,q3	one	one	one	one	one	one	one	one	one	one	one	one	**Correct**
9	t1,q4	one	one	one	one	one	one	one	one	one	one	one	one	**Correct**
10	t1,q4	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	**Correct**
11	t1,q4	zero	zero	five	zero	zero	two	zero	zero	five	zero	zero	two	**Correct**
12	t1,q4	zero	zero	five	zero	zero	two	zero	zero	five	zero	zero	two	**Correct**
13	t1,q5	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	**Correct**
14	t1,q5	one	one	one	one	one	one	one	one	one	one	one	one	**Correct**
15	t1,q5	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	**Correct**
16	t1,q5	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	zero	**Correct**
17	t3,q1	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
18	t3,q1	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
19	t3,q1	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
20	t3,q1	N	one	one	one	N	one	n	zero	zero	zero	n	zero	**Wrong**
21	t3,q2	N	five	zero	zero	N	two	n	five	zero	zero	n	two	**Correct**
22	t3,q2	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
23	t3,q2	N	one	one	one	N	one	n	one	one	one	n	one	**Correct**
24	t3,q2	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
25	t3,q3	N	zero	five	zero	N	two	n	zero	five	zero	n	two	**Correct**
26	t3,q3	N	one	one	one	N	one	n	one	one	one	n	one	**Correct**
27	t3,q3	N	five	zero	zero	N	two	n	five	zero	zero	n	two	**Correct**
28	t3,q3	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
29	t3,q4	N	one	one	one	N	one	n	one	one	one	n	one	**Correct**
30	t3,q4	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
31	t3,q4	N	five	zero	zero	N	two	n	five	zero	zero	n	two	**Correct**
32	t3,q4	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
33	t3,q5	N	zero	five	zero	N	two	n	zero	five	zero	n	two	**Correct**
34	t3,q5	N	one	one	one	N	one	n	one	one	one	n	one	**Correct**
35	t3,q5	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**
36	t3,q5	N	zero	zero	zero	N	zero	n	zero	zero	zero	n	zero	**Correct**

From the above table (Table 6) one can derive that the BPN network Architecture #3 (with verbally encoded outputs) truly learned 27 out of a total of 28 model inputs/outputs, hence a percentage of 27/28 = 96.43%, which is comparable with the correlation factor percentage during its training, which is 96.22%. In addition, from

Table 6 one can derive that the BPN network Architecture #3 estimates correctly almost 8 of the 8 output values, hence, a 8/8 = 100.0% percentage.

After making use of three different efficient BPN network Architectures, we came to the conclusion that the third one had the most successful performance of all and that is why we chose present it in a more thorough way in the present paper. Taking into account the results of the application of the three architectures, one could claim that neural network technology can be used for developing a model for automated evaluation of deaf students' answers and progress that possess basic knowledge of the English language and computer skills within a virtual e - learning environment.

However, this study focused on the development of typical BPN models based only on a limited dataset. It is regarded as worthwhile to develop more BPN models, which are trained for wider dataset and to determine the level of their best performance using all existing performance improvement techniques of the neural networks. In general though, the acceptance of a model depends on its performance, which may be evaluated based on statistical methods provided that the remaining errors between the models' output values and the real values follow approximately the normal distribution.

In this study the effort was focused merely on the examination of the expediency of applying neural networks technology for modeling the automated evaluation of the performance and progress of deaf students who possess basic knowledge of English and computer skills, within a virtual e-learning environment. In the current cases of BPN simulations though, together with the small sum of available input/output data, the aim was to exhaust all the possibilities in order to derive the best BPN model.

In general, the procedure of finding the best model and the successful development of a real calculating application with neural networks includes a procedure with many repetitive steps.

The first step is the definition of the neural networks system aim. In other words, a taxonomy problem, a problem that minimizes the sum of the square values of the error differences between the prediction values and the real values in the output, or a problem that interpolates well in the whole range of the input parameters.

The second step is where the data quality is examined (checking for consistency, missing fields, formats, range of values, and for other characteristics, faults, or inadequacies) and the analysis, remodeling and transformation of the variables so that their distributions match the distributions of the dependent variables.

The third step is the choice step, according to the aims of the system, the sum of the training data for the adjustment of the model parameters, the sum of the test data to determine how well the model interpolates and for avoiding over-fitting and the sum of the check data for the estimation of the performance of the model in a real application environment.

In later stages there are more steps, such as the choice of the input variables step as well as the creation and training of the network step and finally, the check model step.

In addition to the aforementioned steps, there are many differentiations regarding the type of the neural network, architectures and neurons that need to be examined. For example [9], the sum, beyond being a simple sum of the inputs times the weights, can also be a cumulative sum and/or a sum that calculates the minimum or maximum of inputs times the weights, the majority of the inputs times the weights, the product of the inputs times the weights, the Euclidean distance between the weights and the source outputs etc, or in the case of projection networks to calculate different normalization functions. The continuous transfer function can be a linear gain function, or a non-linear function such as sigmoid, hyperbolic tangent, sine and exponential function, a threshold that permits the passing through of information provided that the joint final input rises to a certain value like functions such as percetron, signum, signum0, step and psi, or even functions that are specifically imposed by the neural network, such as probabilistic neural networks, radial basis functions, general regression neural networks, Spatio-Temporal Pattern Recognition, Boltzmann and Digital Neural Network Architecture.

Moreover, during the procedure of creating any model, a very special issue is that of model generalization or in other words how well the model interpolates from a new sum of inputs to the corresponding outputs. The opposite of generalization is over-fitting, which is more apparent in complex models with a large number of free parameters. For neural networks (where there is no restriction regarding weights), the number of free parameters equals the number of interconnections, the over-fitting can be reduced with the use of techniques such as preserving a small number of neurons, using an independent sum of test data for performance control on a regular basis during training and interruption of training when degradation occurs. Furthermore, another technique is the increase of the data records size (the "degrees of freedom") and decrease of the number of input variables (variable selection process), or finally, with the implementation of the weight decay and noise factors method, correction of the gradient descent during the learning phase.

Having proved the expediency of the BPN type neural networks technology for modeling the evaluation of the educational progress of deaf students in an e-learning environment, additional simulations are required for the further increase of the performance according to the aforementioned generic procedures and issues for the realization and real implementation of neural networks models.

4 Conclusions

In this study, using the ESOL specifications for the skilful encoding and the inputs and outputs content and also the structure, standardization and content of the database questions of an e-learning expert system, three BPN type models were developed for the estimation of the expediency of applying neural network technology towards the automated estimation of the values, for the evaluation and progress of teaching English to deaf people.

Based on the performances of the three type BPN architectures that were developed, namely, BPN network Architecture #1, BPN network Architecture #2 and BPN network Architecture #3, it was derived that the neural networks technology can be used towards the development of such an automated model for the evaluation of the answers and the progress of deaf students that possess basic knowledge of the English language and also basic computer skills, within a virtual e-learning environment.

However, it is regarded that this estimation would be more accurate and would have more certitude if the available input/output data were more. Additionally, in this study only BPN type neural network models were tested and it is thought of as necessary to develop additional models for this particular problem and to determine the level for their best performance using a larger sum of available input/output data as well as all the performance improvement techniques for the neural networks.

References

1. NeuralWorks Predict: Computer Software package for applications of Neural Networks, NeuralWare Inc., USA.
2. Tukey . J. W.: Exploratory Data Analysis, Addison-Wesley (1977)
3. Samprit C., Bertram P.: Regression Analysis by Example, John Wiley & Sons (1991)
4. Homsi M., Lutfi R., Carro R. M., Ghias B.: Student modeling using NN – HMM for EFL course
5. Weon S., Kim J.: Learning achievement evaluation strategy using fuzzy membership function (2001)
6. Vrettaros J., Vouros G., Drigas A.: Development of a diagnostic system of taxonomies using fuzzy logic – case SOLO (useful for e-learning system) (2004)
7. Vrettaros J., Pavlopoulos J., Vouros G., Drigas, S.: The development of a self-assessment system for the learners answers with the use of GPNN (2008)
8. Frias – Martinez, E., Magoulas G., Chen S., Macredie R.: Modeling human behavior in User – Adaptive Systems: Recent Advances using soft computing techniques (2005)
9. Al Hamadi A. S., Milne R. H.: A neuro – fuzzy classification approach to the assessment of student performance (2004)
10. VanLehn K.: Student Modeling. In: Polson, M. C., Richardson, J. J. (eds.): Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Associates Hillsdale, New Jersey, 55-78 (1988)
11. Stathakopoulou R., Magoulas G., Grigoriadou M., Samarakou, M.: Neuro –Fuzzy knowledge processing in intelligent learning environments for improved student diagnosis (2005)