

# The Development of a Self-assessment System for the Learners Answers with the Use of GPNN

John Pavlopoulos<sup>1</sup>, John Vrettaros<sup>1</sup>, George Vouros<sup>2</sup>, and Athanasios S. Drigas<sup>1</sup>

<sup>1</sup> NCSR DEMOKRITOS, Department of Applied Technologies,  
Patriarhou Grigoriou, 15310 Ag. Paraskevi, Greece

<sup>2</sup> Aegean University, Info and Communication Systems Eng,  
83200, Karlovassi, Samos, Greece

{dr, jvr}imm.demokritos.gr, annis.pavlo@gmail.com,  
georgev@aegean.gr

**Abstract.** The goal of this study is the development of an assessment system with the support of a Neural Network approach optimized with the use of Genetic Programming. The data used as training data are real data derived from an educational project. The developed system is proved capable of assessing data from both single select and multiple choice questions in an e-learning environment. The final result is the assessment of the learners' answers through various criteria.

**Keywords:** neural network, genetic programming, assessment, learners.

## 1 Introduction

This paper presents the development of an assessment system of the gained knowledge of students. In specific, the results of self-assessment exercises provided by a learning environment are examined, in order for the students to obtain the knowledge level they have possessed in each learning section solely and overall. The final aim is for the assessment system to be trained in order to play the role of an instructor. The assessment system is based on a Neural Network approach, optimized with the aid of Genetic Programming.

Neural Networks (NNs) mimic the way human brain functions. Through a large number of interconnections organized in layers, they can capture complex non linear relationships between input and output variables. NNs can be trained (that is adjust their parameters to a certain pattern recognition problem) in order to be able to generalize to unknown data (under linear or non linear relationships). Moreover, hybrid methods can include the use of evolutionary techniques (Genetic Programming or Genetic Algorithms), in order to optimize the architecture and the training parameters of the NNs.

Genetic Programming (GP) is inspired by natural evolution and provides a way to develop computer programs, such as appropriately designed and trained NNs, which produce some desired output for particular inputs [3]. In this paper, in order to produce the assessment system acquired, we examined whether a Genetic Programming

Neural Networks (GPNN) approach [4, 5] is able to model the assessment role of a pedagogical expert.

GPNN uses input and output data in order to train an initial population of NNs through GP. The training procedure stops when a minimum error point is reached or a maximum number of iterations is exceeded. Until now, it was being used as a powerful statistical pattern recognition tool through ten final GPNN models [4, 5]. However, its ability for quick convergence to the solution for linear and non linear relations between the input and the output, make GPNN a very good candidate for an expert system application.

GPNN Assessment System (GPNNAS) is a GPNN system that is trained with data, which consists of answers of students and their evaluation according to a pedagogical expert. The final purpose of the GPNNAS is to be able to evaluate the answer of a student according to some criteria. The final system consists of one Neural Network (NN) approach for each criterion, optimized with Genetic Programming so that each NN approach is able to evaluate the answer according to the specific criterion. Thus, the output of the assessment system for a question is an evaluation of a student's answer for each criterion.

The data generated by the learner going through a mini-test consists of a string of characters and values which are built based on certain criteria. The types of questions are both single-select and multi-select and have several answer options. The questions test learners against more than one sector while each question has a relevance value against every sector.

## 2 Data of the Expert System

The data of the developed system are real data that were extracted from the answers of learners from the Dedalos<sup>1</sup> educational project. The modeling of the data was proved to be precise.

Dedalos learners undertook a mini-test at the end of each module to assess their understanding of the learning points covered. Each mini-test comprises a series of multiple choice questions and each answer option selected provides the GPNN Assessment System (GPNNAS) with two types of data: test data and training data. Pedagogical experts have assigned educational values to the test and training data which, in turn, allows GPNNAS to assess the learner's understanding of the module.

The rest of this section describes these two data types and how values are assigned to them.

### 2.1 Purpose and Transmission of Test Data

Test data assesses how relevant a question is against one of the following areas of learning:

1. letter recognition and alphabetical order
2. spelling/vocabulary

---

<sup>1</sup> Dedalos: Teaching English as a second language to deaf people, whose first language is sign language, via e-learning tools. LEONARDO DA VINCI, Community Action Programme on Vocational Training, Second phase: 2000- 2006.

3. grammar/sentence structure
4. reading
5. writing

Test data also evaluates the answer options against the five areas of learning and specifies whether the answer is correct, partially correct or incorrect.

## 2.2 Assignment of Test Data

Firstly, each question is assigned a relevance value between 0 and 4 by a pedagogical expert. For example, the question “Which sign is in capital letters?” mainly tests the learner’s skills in section A and hence, receives a relevance value of 4 here. It is also about an underpinning reading skill at a low level and therefore it is given a relevance value of 1 in section D. It does not test spelling/vocabulary, grammar/sentence structure or writing at all, hence these sections receive a relevance value of 0.

**Table 1.** Relevance of questions

| Section Code | Section name                              | Relevance Value |
|--------------|---|-----------------|
| A            | Letter recognition and alphabetical order | 4               |
| B            | Spelling/vocabulary                       | 0               |
| C            | Grammar/sentence structure                | 0               |
| D            | Reading                                   | 1               |
| E            | Writing                                   | 0               |

Secondly, each answer option is assigned evaluation values. Evaluation values are also set against the five learning areas. However, the mini-tests comprise two types of multiple choice questions: single select and multi select. While the principle behind the assignment of evaluation values remains the same, a different form of the data set is sent to GPNNAS for each question type.

In single select questions there is only one correct answer. For example for the question “Which sign is in capital letters?” option 2 is the only correct answer and the evaluation values are assigned as follows:

**Table 2.** Evaluation of answers in single-select type questions

| Answer Code | Answer Options | Correct /Incorrect | Evaluation Values |    |    |     |    |
|-------------|----------------|--------------------|-------------------|----|----|-----|----|
|             |                |                    | A                 | B  | C  | D   | E  |
| 1           | Open           | Incorrect          | 0                 | -1 | -1 | 0   | -1 |
| 2           | NO ENTRY       | Correct            | 1.0               | -1 | -1 | 1.0 | -1 |
| 3           | Closed         | Incorrect          | 0                 | -1 | -1 | 0   | -1 |
| 4           | Staff Only     | Incorrect          | 0.3               | -1 | -1 | 0.3 | -1 |

Hence:

- **1.0** is assigned to cell 2A because the answer option is correct and the question is relevant to area A - Letter recognition and alphabetical order.
- **1.0** is assigned to cell 2D because the answer option is correct and the question is relevant to area D - Reading
- **0.3** is assigned to cell 4A because the answer option 'Staff Only' is partially correct as it contains two capital letters and the question is relevant to area A
- **0.3** is assigned to cell 4D because the answer option 'Staff Only' is partially correct as it contains two capital letters and the question is relevant to area D
- **0** is assigned where an answer option was wrong but the question is relevant to the learning area
- **-1** is assigned where an answer option was wrong and the question is not relevant to the learning area

In multiple select questions there can be two or more correct answers. For example, for the question "Which of these are capital letters?" there are three correct answers (options 2, 3 and 6) and the evaluation values are assigned as follows:

**Table 3.** Evaluation of answers in multi-select type questions

| Answer Code | Answer Options | Correct /Incorrect | Evaluation values |    |    |     |    |
|-------------|----------------|--------------------|-------------------|----|----|-----|----|
|             |                |                    | A                 | B  | C  | D   | E  |
| 1           | V              | Incorrect          | 0                 | -1 | -1 | 0   | -1 |
| 2           | G              | Correct            | 1.0               | -1 | -1 | 1.0 | -1 |
| 3           | C              | Correct            | 1.0               | -1 | -1 | 1.0 | -1 |
| 4           | P              | Incorrect          | 0                 | -1 | -1 | 0   | -1 |
| 5           | H              | Incorrect          | 0                 | -1 | -1 | 0   | -1 |
| 6           | B              | Correct            | 1.0               | -1 | -1 | 1.0 | -1 |

The question is primarily devised to test the learner's knowledge of area A - Letter recognition and alphabetical order and to a lesser extent knowledge of area D - Reading. The following values are assigned to the correct answer options (2, 3 and 6):

- Section A: **1.0** – because the answer is correct and the question is relevant to this area
- Section D: **1.0** – because the answer is correct and the question is relevant to this area
- Sections B, C and E: **-1** because the question is not relevant to these areas

### 3 Methodology

In order for a system to be able to evaluate correctly a learner's answer, it should be trained in a way that each input (answer pattern) can be related with a particular output (the evaluation). Neural Networks are capable of this task, once they are appropriately trained.

NNs are weighted interconnected networks of artificial neurons (computational models based on the biological neuron). The training procedure consists of modeling the structure of the NNs as well as defining the values of their weights. Although a gradient descent algorithm such as back-propagation is most often used as a training algorithm, an evolutionary algorithm such as GP has the potential to produce a global minimum of the weight space and thereby avoid local minima [7].

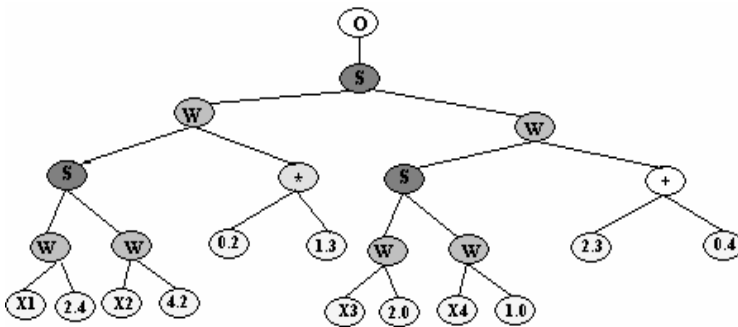
Such a hybrid methodology is GPNN, which produces an initial population of randomly generated NNs and then recombines them through GP operations (reproduction, crossover and mutation) in order for the fittest to survive. The extracted NN is considered to be the most appropriate one for the generalization of the input pattern to the output pattern.

### 3.1 Genetic Programming Neural Networks

GPNN was initially developed by Richie et al. [4] to improve upon the trial-and-error process of choosing an optimal architecture for a pure feed forward back propagation NN. However, the methodology was re-implemented at the Biosim Lab of the National Technical University of Athens, Greece in order to study the genetic and environmental underlay of diseases. In this paper is presented an application of this implementation which aims at training a system (the trained NNs) to evaluate automatically the answers of learners according to a number of criteria.

Optimization of NN architecture using GP was first proposed by Koza and Rice [1]. The use of binary expression trees allows for the flexibility of the GP to evolve a tree-like structure that adheres to the components of a NN (Fig.1). The GP is constrained in a way, that it uses standard GP operators, but retains the typical structure of a feed-forward NN. A set of rules is defined prior to network evolution, to ensure that the GP tree maintains a structure that represents a NN [1, 2]. The flexibility of the GPNN allows optimal network architectures to be generated in such a way that they will consist of the appropriate inputs, connections, and weights for a given data set [6].

The steps of the GPNN method are described in brief as follows. In step one, GPNN has a set of parameters that must be initialized before the beginning of the evolution of the NN models. These include, an independent variable input set, a list of



**Fig. 1.** The tree structure of a Neural Network. The o-node is the output node, the w-node is the weight node and the s-node is the activation function node.

mathematical functions, a fitness function, and finally the operating parameters of the GP. These operating parameters include the population size, and the number of generations. In step two, the training data are modeled according to the tested problem. In step three, the training of the GPNN begins by generating an initial population of random solutions. Each solution is a binary expression tree representation of a NN (Fig.1). In step four, each GPNN is evaluated on the training set and its recorded fitness. In step five, the best solutions are selected for crossover and reproduction, using a fitness-proportionate selection technique, called roulette wheel selection, based on the classification error of the training data [4, 5].

Classification error is defined as the proportion of individuals for whom the output was incorrectly specified. A predefined proportion of the best solutions are directly copied (reproduced) into the new generation. Another proportion of the solutions are used for crossover with other best solutions and finally the last solutions are mutated. The extracted NN, which is the best-so-far solution, is considered to be capable of classifying the data with the minimum error. In the last step, the best-so-far solution is being held and the new generation, which is equal in size to the original population, begins the cycle again. This continues until some criterion is met, and at that point the GPNN stops. This criterion is either a classification error of zero (best-so-far solution) or the maximum number of generations reached (error message).

### 3.2 Application of GPNN

Until now, GPNN was mostly used for pattern recognition in the field of Bioinformatics [4, 5]. However, this GPNN application aims at modeling the classification of the answers of learners and thus, the NNs are expected to associate each answer (NN input) with an evaluation (NN output).

The training procedure of the assessment system for each question consisted of training six NNs, one for each of the five criteria and one for the overall performance. The inputs of the NNs (answer patterns) consisted of binary strings representing different answer codes. Inside the binary string, the 1's represented the correct choices according to the user while the 0's the false ones. For example, the NN input string 1-0-0-0, for a single select question, would indicate that the learner selected the first choice as the correct one. The output of each NN (answer evaluation) could either be negative, indicating an irrelevant criterion, or a number from the space [0, 1], representing the evaluation of the learner's answer according to the specific criterion.

GPNNAS, in its pattern operation has been applied for both a question of single select and a question of multi select type and has modeled the data successfully proving the system's capability of modeling this kind of data. The single select type question was "Which sign is in capital letters?" and there were four possible student answers, while the multi select type question was "Which of these are capital letters?" and there were nine possible answers.

For each criterion, the initial NN population was set to be 100 NNs while the generations, through which NNs' genetic recombination took place, were set to be 50. The training procedure of the three NNs that were responsible for the first three criteria of the single select type question is depicted in Fig.3. As it can be observed,

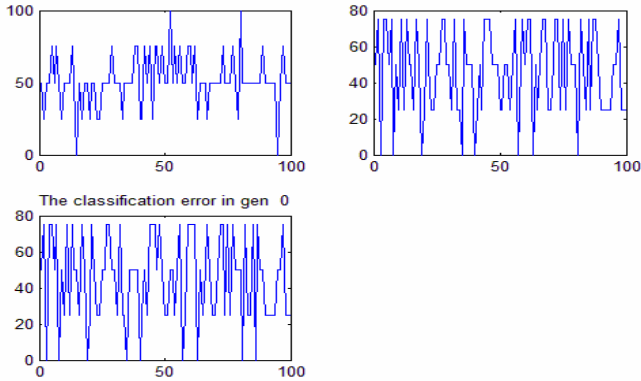


Fig. 3. The classification error of 100 NNs of the initial population after 0 Generations

the 3<sup>rd</sup> fittest NN (classification error 0) was found inside the initial population (generation 0) and needed no GP operations, indicating the simplicity of the modeled functions.

The answers of the learners were uploaded via a web page to the main server (Fig. 4), wherein they were encoded in an appropriate form and were processed by the GPNNAS.

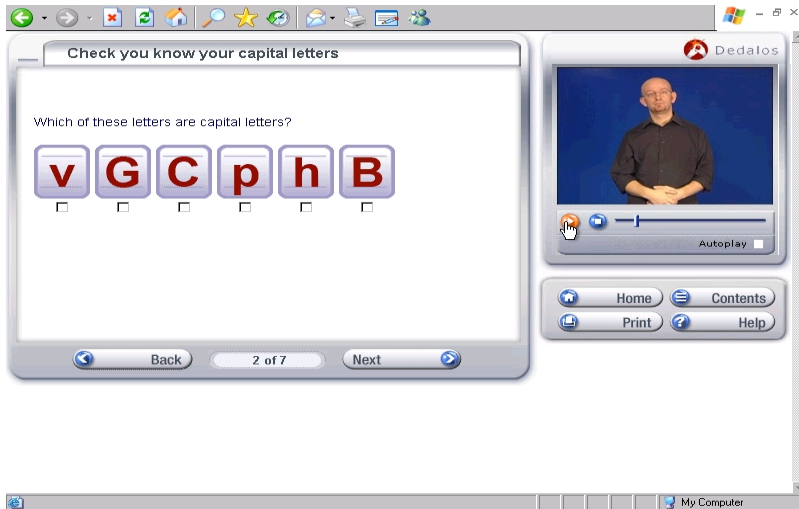


Fig. 4. The question interface for the Dedalos e-learning environment

The output of the system was the learner’s evaluation for the five criteria examined as well as for the learner’s overall performance. Furthermore, the evaluation was presented to the learner through a bar diagram (Fig. 5), forwarding intelligibility of the results for the user.

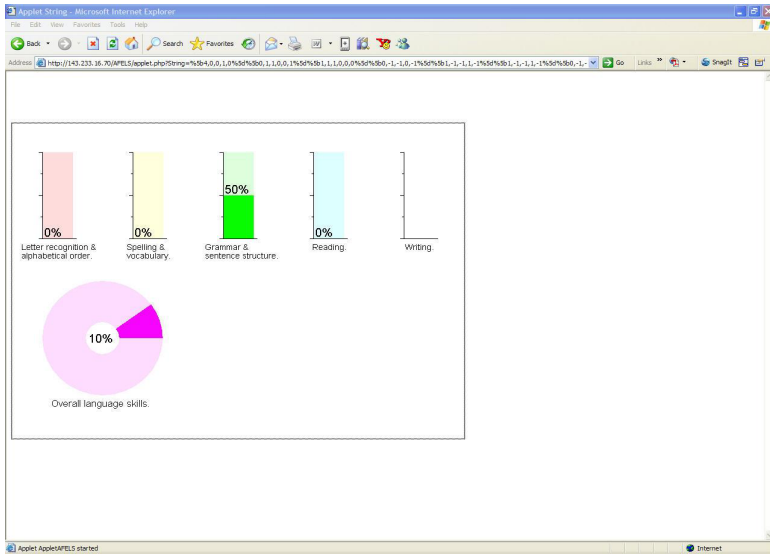


Fig. 5. Classification form of the results

## 4 Future Work

In this paper, a hybrid expert system with use of GPNN is developed for the evaluation of learners' answers according to a number of criteria. Thus, the assessment data could be represented to the learner in a meaningful and useful way in order to help the learner improve his skills in the cognitive sections where he showed low performance in the relevant test. The application of the GPNN methodology for e-learning purposes allows for generalization of the assessment process which could lead to the implementation of an intelligent e-Tutor. The system was applied and evaluated successfully learners' answers, which were derived from an educational project for the teaching of English as a second language to deaf people whose first language is the sign language. The next challenge is a fully automated training procedure wherein the training data will be presented to the assessment system online and the system could be trained in real time, as well as over different and more complicated kinds of tests. Thus, an e-learning system could be implemented that could serve various kinds of learners who need to improve their learning abilities according to various criteria.

## References

1. Koza, J.R., Rice, J.P.: Genetic generation of both the weights and architecture for a neural network. In: IJCNN 1991-Seattle International Joint Conference on Neural Networks, vol. 2, pp. 397–404 (1991)
2. Koza, J.R.: Survey of genetic algorithms and genetic programming. In: WESCON 1995 Conference: Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies, pp. 589–594 (1995)



3. Koza, J.R.: Genetic Programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Stanford University Computer Science Department (1990)
4. Ritchie, M.D., Motsinger, A.A., Bush, W.S., Coffey, C.S., Moore, J.H.: Genetic Programming neural networks: A powerful bioinformatics tool for human genetics. *Applied Soft Computing* 7, 471–479 (2007)
5. Ritchie, M.D., White, B.C., Parker, J.S., Hahn, L.W., Moore, J.H.: Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. *BMC Bioinformatics* 4(1), rec.NO 28 (2003)
6. Spears, W.M.: A Study of Crossover Operators in Genetic Programming. In: Raś, Z.W., Zemankova, M. (eds.) *ISMIS 1991*. LNCS, vol. 542, pp. 409–418. Springer, Heidelberg (1991)
7. Siddique, M.N.H., Tokhi, M.O.: Training Neural Networks: Backpropagation vs. Genetic Algorithms. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2673–2678 (2001)