USE OF NEURAL NETWORKS IN ASSESSING KNOWLEDGE AND SKILLS OF UNIVERSITY STUDENTS

E. Hadzhikolev, K. Yotov, M. Trankov, S. Hadzhikoleva
Students’ assessment

- True scores
- Reliability, Validity
- Fair assessment
  - Differentiation and comprehensiveness of the assessment
  - Fair and equitable assessment practices for all students
- Students put effort to only learn the things that will help them score higher
- Correlation between the students' motivation to perform well and the results achieved
Bloom’s taxonomy

- **Six cognitive levels** - Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation
- **Each level is characterized by numerous mental skills**
- The levels are arranged hierarchically - from simpler to more complex ones
- When acquiring new knowledge, *it is believed that a learner must go through all levels successively*
Higher order thinking skills

- **Lower Order Thinking Skills (LOTS)**
  - Covers the lower three levels of Bloom’s Taxonomy - Knowledge, Comprehension and Application
  - Must be achieved first

- **Higher order thinking skills (HOTS)**
  - Non-algorithmic, complex mode of thinking
  - Creative, innovative and critical thinking
  - Corresponds to the upper three levels of Bloom’s Taxonomy - Analysis, Synthesis, and Evaluation
Setting the experiment

- *In programming training, assessment is often complex and involves several components aimed at evaluating the acquired theoretical knowledge and practical skills*

- In general, the assessment components participate in forming the final assessment with different weight

- **Our experiment aimed at facilitating educators in forming assessments based on multiple assessment components**

- The conducted research and analysis were carried out in the course of training undergraduate students studying in "Internet programming with PHP and MySQL" course
Assessment components

- **Basic Theoretical knowledge (T_LOTS).** Assessed is the knowledge acquired, the ability to understand and apply the studied theoretical material. Open-ended test questions of the “choose the right answer” type are used. The questions require distinguishing, listing and comparing objects; explaining and exemplifying concepts; reproducing, clarifying and using concepts, etc.

- **Basic Practical Knowledge (P_LOTS).** Assessed are the abilities to understand and apply the studied syntactic constructs, libraries, etc. The test questions are mostly related to predicting the result of executing a short program code, modifying a code, finding errors, applying examples, etc.

- **Theoretical Skills (T_HOTS).** Students answer questions in free form (text, diagrams, tables, etc.), demonstrating their ability to analyze, synthesize, and evaluate. The questions are related to explanation, interpretation, graphical presentation and program code modification; comparison and critical analysis of solutions; design of software components; summarizing, proving, drawing conclusions, assessing, etc.

- **Practical Skills (P_HOTS).** Assessed are the students’ practical skills to solve relatively new tasks, to design and develop web applications.
## Assessment components

<table>
<thead>
<tr>
<th>Assessment component</th>
<th>Symbol</th>
<th>Assessment scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical knowledge – T_LOTS</td>
<td>x1</td>
<td>integer in the interval [0, 30]</td>
</tr>
<tr>
<td>Practical knowledge – P_LOTS</td>
<td>x2</td>
<td>integer in the interval [0, 15]</td>
</tr>
<tr>
<td>Theoretical skills – T_HOTS</td>
<td>x3</td>
<td>integer in the interval [0, 15]</td>
</tr>
<tr>
<td>Practical skills – P_HOTS</td>
<td>x4</td>
<td>real number in the interval [2, 6]</td>
</tr>
<tr>
<td>Final assessment</td>
<td>finalGrade</td>
<td>integer in the interval [2, 6]</td>
</tr>
</tbody>
</table>
Assessment

- The education in "Internet programming with PHP and MySQL" is usually conducted with large courses, with the number of students around 60-80 people.

- For the exercises, students are divided into small groups (15 people), each of which can be taught and assessed by a different teacher.

- The experiment assessed 130 students trained in two specialties. One educator gave lectures and assessed the components $x_1$, $x_2$, and $x_3$, and three different instructors taught exercises and assessed the $x_4$ component.

- While the test components based on the lectures - $x_1$, $x_2$ and $x_3$ are relatively objective indicators, the $x_4$ component can be affected by subjective influences related to the teaching and assessment methods of different educators.
In our experiment, there are dependencies between the individual assessment components - between the theoretical LOTS and HOTS, as well as between the practical LOTS and HOTS.

Situations of serious discrepancies between the pairs \((x_1, x_3)\) and \((x_2, x_4)\) raise doubts about the students' knowledge and skills.

For example, with maximum allowable results \((30, 15)\) for the pair \((x_1, x_3)\), serious discrepancies show in the following results obtained by students: \((24, 2), (25, 3), (27, 5)\) etc. Respectively, with maximum allowable results \((15, 6)\) for the pair \((x_2, x_4)\), some students have obtained results \((3, 5.5), (4, 6), (1, 4), (0, 4), (10, 2.25)\) etc.

These results show serious deviations in both directions - high practical HOTS and relatively low LOTS or vice versa - relatively high practical LOTS and poor HOTS.
The diagrams represent the distribution of rounded scores for theory, practice and final grades.

The diagrams in fig. 1 show that part of the exercise grades of one teacher (fig. 1.a) differ significantly from the theoretical grades, while the scores of the other two teachers are comparable.

An interesting result (fig. 2.c) is that final grades and the theory scores for all students are relatively normally distributed around the average.
Need for new solutions

- Creating an automated algorithm for fair assessment with available specific values of the assessment components requires a non-standard solution.
- In general, the idea of describing all specific cases and their corresponding solutions is not applicable.
- The task we set for ourselves is to find a common approach for automated assessment.
- Experiments with artificial intelligence methods were carried out to solve this task.
Using machine learning for automated assessment

- There are numerous AI models and algorithms for machine learning: Artificial neural networks, Decision tree, Random forest, and many others.
- The primary process for them is training based on existing examples of input data and the corresponding target output values.
- In our case, the main input data are the assessment components values $x_1$-$x_4$, and the outputs are the respective final grades for each arranged four ($x_1, x_2, x_3, x_4$) as formed by the teacher.
- For education methods, some of the sample data is used for education and the other part – for validation and testing.
- Upon completion of the training, the implemented solutions can be applied to new input data.
- In our case, we expect that when a new combination of values is set for the selected components, the correct final assessment will be determined automatically.
Choosing factors for the algorithms

- The **main factors** influencing the final assessment are the components $x_1$-$x_4$.

- The dependencies between the components were not explicitly used in the software implementation. They have an influence on the final assessment formation, and respectively, in the learning process based on the input-output patterns.

- While working on the evaluation task, **numerous experiments were carried out**. Some of them have proved the groundlessness of the doubts that it is possible to ignore any of the $x_1$-$x_4$ components and still obtain the same final grades.
Training data for machine learning

- In the experiments initially conducted, the available data for the assessment of the 130 students proved to be insufficient. Test results showed high error rates.

- Therefore, questions arose as to how much data for effective training in machine learning algorithms is needed and whether we can automatically increase their quantity.

- To provide additional training, validation and testing examples, we used the GARP method for space compaction of input-output patterns developed during the study. (The GARP method is not the subject of this presentation and is just briefly discussed here for completeness of the presentation)
The GARP method

- Using the Generation of additional ray's points (GARP) method, we create multiple additional input-output patterns on the segments defined by already known input-output patterns.

- By increasing the number of input-output patterns, we provide a better basis for training in machine learning algorithms.

- Fig. 3 illustrates the basic concepts used in the development of two-dimensional factor space.

- In the mathematical model, we consider the more general case of an n-dimensional factor space in which the values of the factors are real numbers, i.e. each factor \( x_i \in \mathbb{R}, i = 1..n \).

- Solving the more general case, we not only get a solution for the particular assessment task, but we also provide a general approach that can be applied in other similar cases.
The GARP method

The main symbols and concepts in the model are:

- $FS^n \subset R^n$ is a subset of an $n$-dimensional Euclidean space of factors $x_1, x_2, ..., x_n$
- Any arbitrary point $P_p(x_{1p}, x_{2p}, ..., x_{np})$ of $FS^n$ is presented as a combination of values of the relevant factors.
- The set of input-output patterns $FSL^{n+1} = (FS^n \times L) \subset R^{n+1}$, where $L \subset R$ is the set of possible values of the final grades.
- Each point of $FSL^{n+1}$ is of type $N_p(P_p, L_p) = N_p(x_{1p}, x_{2p}, ..., x_{np}, L_p)$. The value $L_p$ indicates the assessment of the relevant set of factors $(x_{1p}, x_{2p}, ..., x_{np})$.

The main result of mathematical modeling is that for each two points of input-output patterns $N_p(P_p, L_p)$ and $N_q(P_q, L_q)$ of $FSL^{n+1}$ we can build a sequence $\{N_{pqj}(P_{pqj}, L_{pqj})\}_{j=0}^m$ of $m + 1$ additional points whose components are:

- $(1)\ P_{pqj} = P_{pqj}(x_{1p} + \delta_{1j}, x_2 + \delta_{2j}, ..., x_n + \delta_{nj}) \in FS^n, \delta_{ij} = \frac{j(x_{iq} - x_{ip})}{m}, i = 1..n, j = 0..m$ and
- $(2)\ L_{pqj} = L_p + \frac{j(L_q - L_p)}{m} \in R, j = 0..m$. 
The GARP algorithm - basic steps

- **A set of input data is generated for the algorithm**: a set of $s$ input-output patterns of the type $\{N_k(P_k, L_k)\}_{k=1}^s$.

- **The set of input-output patterns is arranged under the following conditions**:
  - the first element is the one closest to the center of the coordinate system in $FS^n$;
  - each subsequent element in the ordered set is defined as:
    - the last element of the ordered elements set is selected;
    - its closest neighbor belonging to the unordered set is found;
    - the found element from the unordered set is subtracted and added at the end of the ordered set.

- **Additional input-output patterns are generated** for each two adjacent elements in the ordered set of input-output patterns using the formulas (1) and (2).
Multiple experiments were performed using MATLAB’s Neural Net Fitting tool to solve the problem of finding an artificial neural network based on 130 input-output patterns.

- The first set of experiments included only the initially specified 130 input-output patterns. The basic parameters for the optimal NN built during the experiments are:
  - 75 neurons in the hidden layer;
  - training method used: Levenberg-Marquardt;
  - activation function for the neurons in the hidden layer: hyperbolic tangent $Tansig(x) = \frac{2}{1+e^{-2x}} - 1$;
  - activation function for the output neuron: $Purelin(x) = x$.
  - error rate: $1e-5$.

Despite the low error rate, NN showed weaknesses during additional testing. In cases where the set test points of $FS^4$ are relatively farther from the points used in the training, the network made assessments different from the teacher’s ones.
Automated assessment through artificial neural networks at MATLAB

In the second set of experiments we used the GARP method to generate additional patterns on 50 of the input-output patterns. We received a total of 7400 education patterns. The characteristics of the optimal NN obtained in the experiments are:

- 100 neurons in the hidden layer;
- training method used: Levenberg-Marquardt;
- activation function for the neurons in the hidden layer: hyperbolic tangent $Tansig(x) = \frac{2}{1+e^{-2x}} - 1$;
- activation function for the output neuron: $Purelin(x) = x$.
- error rate: 1e-7.

Unlike the first set of experiments, the NN thus created showed no errors during the additional tests. In the last, most important verification phase, the NN was tested with the 70 real-life examples not used in training. There were only 2 discrepancies between the teacher’s assessment and the NN assessment.
Experiments with other AI methods

- In addition to NN, experiments with other machine learning algorithms were conducted. For software implementation we used Python language and libraries of tools for data analysis and machine learning Pandas and Scikit-learn.

- Like with the neural networks experiments, the results from Python's initial development using the original 130 input-output patterns were not encouraging. The accuracy of algorithm execution was below 70%.

- When using the additional samples generated by the GARP method, the tested machine learning algorithms showed better results: Linear Regression - 81.82%; Logistic Regression - 75.36%; Decision Tree - 96.80%; Support Vector Machine - 91.80%; Naive Bayes - 71.16%; K-Nearest Neighbors - 98.67%; K-Means - 72.59%; Random Forest - 99.03%; Gradient Boosting Algorithms (GBM) - 97.82%.

- Classification algorithms such as Decision Tree, K-Nearest Neighbors, Random Forest, and Gradient Boosting Algorithms proved to be suitable for use in evaluation tasks similar to ours.
Conclusions

- Assessing student achievements is an extremely complex and responsible task. The assessment should reflect the different aspects of the trainees' schooling - theoretical knowledge and practical skills. Lower order thinking skills and Higher order thinking skills need to be assessed purposefully. To achieve this, the educator has to provide different opportunities for the students to demonstrate their knowledge and skills.

- In some cases, the evaluation is complex and includes multiple assessment components with functional dependencies between them. In case of major or incomprehensible differences in the grades of a learner, the educator should make an assessment based on his/her personal experience. In such cases the assessment can be automated using different machine learning algorithms.

- The experiment presented in this article is a specific example in this regard. The neural network apparatus was used to predict students' final grade, based on 4 assessment components used to assess practical and theoretical HOTS and LOTS. The results obtained with the software experiments are rather a help to the teacher than indisputable final student grades.
Thank you!