NEURAL NETWORKS IN OPTIMIZATION OF THE SOFTWARE DEVELOPMENT PROCESS

Yaroslav A. Honchar
Scientific supervisor — ph.d., Andrii Babii
Kharkiv National University of Radio Electronics
(61166, Kharkiv, pr. Nauki, 14, Software Engineering Department,
tel. 38 (057) 702-14-46)
e-mail: yaroslav.honchar@nure.ua

Software and hardware increasingly take a part of our life in the information era. Projects are filling with functionality, are becoming more significant and difficult to manage. The development time and the number of errors continuously increases. The development process includes steps as following: requirements analysis, design, implementation, testing, documentation, and support. They require significant effort that needs to be appropriately shared among all workers. But neural networks can help optimize every step of tedious software development.

Firstly, neural networks can simplify the requirements analysis process. We need to skip previous products' requirements and specifications that are more understandable for the developers through neural model. This one can process customer needs into more precise requirements, that can be rated by the customer. Developers will be able to quickly understand processed requirements and create a product that is more appropriate to the customer's vision.

The next step is to divide the work among the workers equally. To do this, we can create a model based on its previously entered tasks and their complexity. After training it will be determining tasks implementation effort.

Working with clear requirements designers better understand the user needs and design a better layout. In this case, neural networks can be used to provide designers with suggestions for developing such a product. This model can be created from the previously established projects and users' feedback on relevance, functionality, and convenience.

Used on implementation process, neural networks can analyse the program code for effective methods of solving tasks. There are two main stages of neural network software optimization.

Firstly, neural networks can analyze source code to identify unused sets of program code (which is not executed while the program is running) and variables to increase performance and reduce memory usage. Then the program code can be analyzed for the similarity of the code functions and known by neural network algorithms to check for the presence of faster algorithms that perform the same tasks. This process is required because it is advisable to apply a better algorithm. This high-level optimization can drastically change the program operation speed and memory usage, mainly when it touches the computational complexity.

Secondly, neural networks can analyze the binary code of an instruction set of compiled programs for the presence of unused memory areas. Then the instruction sets are analysing to find a faster counterpart. In this case, the neural network must remove redundant instructions, as well as find complicated preparations for the computer and replace them with more simple ones. A trained neural network can perform low-level optimization and work as a high-optimization compiler.

The next stage of software development is testing. Neural networks can help with stability testing. We should train the model with a set of codes where some errors appeared. This model finds problem areas of the code and indicates to the programmers the error rate of their system and the most probable errors. It is highly recommended to test high-level optimized program firstly, and only then perform low-level optimization.

Neural networks can simplify the documentation process. Based on the written program code model should describe the interface functions. In combination with the previously established precise requirements, it can provide text suggestions for writing documentation following these requirements.

Neural networks can be applied in software support. Based on the previously asked questions and adequate answers model will try to support the user when he contacts the support service or help the consultant to form a solution quickly. A model can be tied to either specific products or all.

The neural network can help consultants to define the user's psychology state to answer the questions in more appropriate form. This data also can be used to refine the previous model.

The development process is becoming more complicated with the creation of large and high-functional projects. Neural networks can efficiently organize and simplify it, specifying product requirements, providing design suggestions and recommendations for optimizing algorithms during implementation, offering documentation texts and actively supporting users, helping both users and consultants to solve software-related problems.

REFERENCES

1. Gregory Levitin, “Computational Intelligence in Reliability Engineering, Springer-Verlag Berlin Heidelberg”, 2007.

2. S Dick, A Kandel, “Computational Intelligence in Software Quality Assurance”, World Scientific Publishing Co. Pt. Ltd., 2005.

3. Xin□She Yang, “Engineering Optimization: An Introduction with Metaheuristic Applications”, John Wiley & Sons, Inc., 2010.

4. K. Srinivasan, D. Fisher, “Machine learning approaches to estimating software development effort”, IEEE Transactions on Software Engineering Volume: 21, Issue: 2, Feb 1995.

5. K. -L. DuM. N. S. Swamy, “Neural Networks in a Softcomputing Framework”, Springer-Verlag London Limited, 2006.