

Intelligent Decision-Making Technology for Choice MVC Models using Neural Networks



Krasyukova N.L.¹, Boyko A.N.^{2,*}, Plotitsyna L.A.³, Zudenkova S.A.⁴, Danilkevich M.A.⁵

¹Financial University under the Government of the Russian Federation, 125993, Russia

krasyukova@mail.ru

²Financial University under the Government of the Russian Federation, 125993, Russia

boykoan@mail.ru

³Financial University under the Government of the Russian Federation, 125993, Russia

plotitsyna@mail.ru

⁴Financial University under the Government of the Russian Federation, 125993, Russia

zudenkova@mail.ru

⁵Financial University under the Government of the Russian Federation, 125993, Russia

danilkevich@mail.ru

ABSTRACT

Today, the development of any web application based on the MVC architecture can not be imagined without its main component-the model. The convenience and thoughtfulness of the created model allows you to improve the quality and speed of the task to be solved.

However, today most developers use standard interface elements or self-written models, which leads to long development and possible synchronization problems. In order to ensure that you can quickly adapt and select your own set of user model elements for a specific task, you need to create an automated module that allows you to create software code models based on the MVC architecture.

By using the MVC architecture, it is possible to separate the model from its visualization (representation, view). This separation increases the ability to reuse the code.

The most useful application of this concept is when the user needs to see the same data simultaneously in different contexts and / or from different perspectives. As a result, the development of a software module that will automate the creation of models based on the MVC architecture is an urgent task.

Key words: Neural network, MVC, developing, architecture, modeling

1. INTRODUCTION

Smart technologies are various approaches and tools that use knowledge to achieve a specific goal in efficiency. The computer is now at the center of almost every economic operation in the developed world. Computer technologies are also rapidly reaching developing countries, due to the rapid spread of mobile phones. Soon the entire planet will be connected, and most economic

transactions around the world will be carried out through a computer Data systems that were once created for accounting, inventory control, and invoicing now have other important applications that can improve our daily lives and simultaneously stimulate the global economy [4].

The decision support system (DSS) is a computer-based automated system that aims to help people making decisions in difficult conditions for a complete and objective analysis of subject activities. In other words, it provides information (in printed form, either on a monitor screen, or by sound) based on input data, which helps people quickly and accurately assess the situation and make a decision. DSSS arose from the merger of management information systems and database management systems.

Different methods are used in the DSS to analyze and develop offers. These can be: information search, data mining, case-based reasoning, simulation, neural networks, situational analysis, cognitive modeling, and others. Some of these methods have been developed within the framework of artificial intelligence. If the basis of the work of the DSS is based on artificial intelligence methods, then they talk about DSS or ISPR.

2. MATERIALS AND METHODS

The classes of systems that are close to the DSS are expert systems and automated control systems. The decision-making process (figure 1) generally has 5 stages. Each stage takes a finite amount of time and resources.

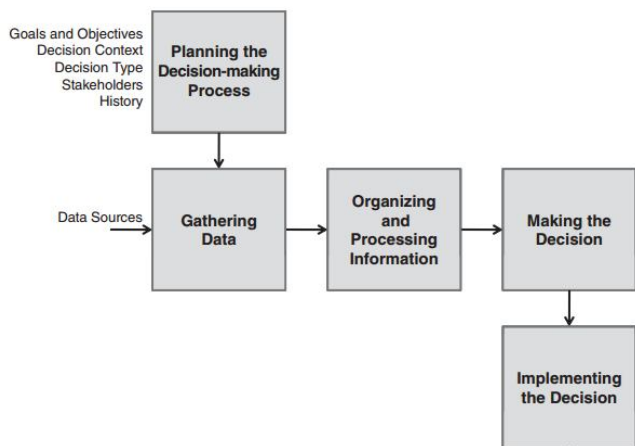


Figure 1: Schematic representation of the decision-making process

There are three types of solutions [15-18]:

- structured-routine decisions, in the sense that the context is quite clear and the scope of action is known. In many cases, there are standards, global or corporate, that describe the methods of decision-making. Such decisions have usually been made in the past;
- weakly structured-similar decisions may have been made before, but the new circumstances differ from the previous ones so much that the success of the previous decision does not guarantee the correct choice. Usually you can borrow from them, if not the methods themselves, then at least the direction of thinking. This group includes many system engineering solutions;
- unstructured-complex problems that are unique and have no analogues. For example, a decision about new technologies.

By management area the solution can be divided into three groups:

- decisions on functioning-are made by practical specialists (engineers, analysts, architects, testers, etc.), usually structured and slightly structured. All procedures and algorithms are usually described in detail in the guidelines;
- administrative and management decisions – the main level at which decisions related to system engineering are made (the level of chief engineer, program Manager, system engineer). There are usually policies, heuristics, and logical relationships that point the system engineer in the direction of finding a solution [1];
- solutions in the field of strategic planning – strategic planning) - the level of management or the entire enterprise. For weakly structured solutions, the ideas of causality (establishing causal relationships) are usually used. At this level, investment decisions and decisions under uncertainty are usually made [5].

Information search is the process of identifying in a certain set of documents (texts) all those that are dedicated to the specified topic (subject), meet a pre-defined search condition (query) or contain the necessary (corresponding to information needs) facts, information, data [2].

The search process includes a sequence of operations aimed at collecting, processing, and providing information.

In General, the search for information consists of four stages:

- definition (clarification) of information needs and formulation of information requests;
- determination of the set of possible holders of information arrays (sources);
- withdrawal of the information of the detected data sets;
- familiarization with the information received and evaluation of search results.

Artificial neural networks are divided into the following types of architecture:

Neural networks of direct propagation (Figure 2) and perceptrons are very direct, they transmit information from input to output.

Neural networks are often described as a layer cake, where each layer consists of input, hidden, and output cells. Cells of one layer are not connected to each other, and neighboring layers are usually completely connected. The simplest neural network has two input cells and one output cell, and can be used as a model of logic gates.

Direct propagation neural networks are usually trained using the error propagation method, in which the network receives multiple inputs and outputs [3-6].

This process is called learning with a teacher, and it differs from learning without a teacher in that in the second case, the network composes a set of source data independently.

The above error is the difference between input and output. If the network has a sufficient number of hidden neurons, it is theoretically able to simulate the interaction between input and output data. In practice, such networks are rarely used, but they are often combined with other types to get new ones [9].



Figure 2: Direct propagation neural networks

Networks of radial basis functions (Figure 3) - uses radial basis functions as activation functions. It doesn't stand out any more.

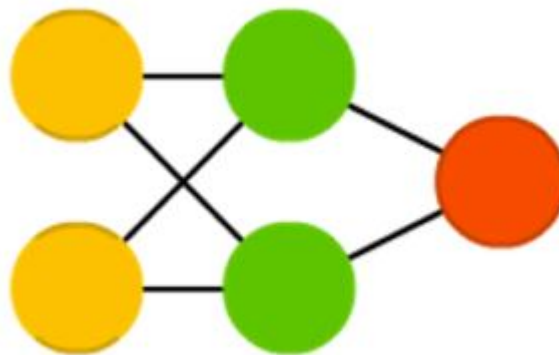


Figure 3: Network of radial basis functions

The Hopfield neural network (Figure 4) is a fully connected neural network with a symmetric matrix of connections.

The Hopfield network is single-layer and consists of N artificial neurons. Each neuron of the system can accept one of two States at the input and output (which is similar to the output of a neuron with a threshold activation function) [13,14].

When receiving input data, each node is an input, during training it becomes hidden, and then becomes an output.

The network is trained as follows: the values of neurons are set according to the desired pattern, and then the weights are calculated, which do not change in the future. After the network has learned from one or more templates, it will always be reduced to one of them (but not always to the desired one). It stabilizes depending on the total "energy" and "temperature" of the network. Each neuron has its own activation threshold, which depends on the temperature, when passing which the neuron takes one of two values (usually -1 or 1, sometimes 0 or 1) [9-11].

Such a network is often called a network with associative memory; just as a person seeing half a table can imagine the other half of the table, so this network, getting a table that is half noisy, restores it to full.

Because of their bipolar nature, Hopfield neural networks are sometimes called spins. Each neuron is connected to all other neurons. The interaction of neurons in the network is described by the expression:

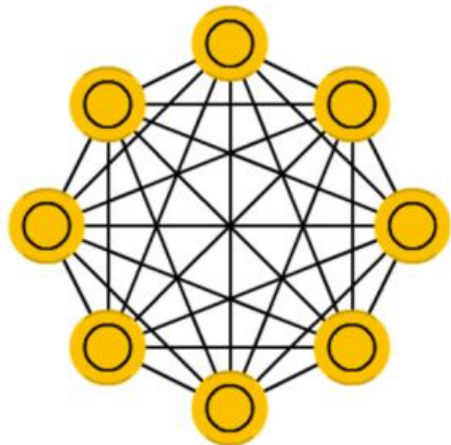


Figure 4: Hopfield Neural network

Markov chains (Figure 5) are the precursors of Boltzmann machines and Hopfield networks. Their content can be explained as follows: what are my chances of getting into one of the subsequent nodes, if I am in this one? Each subsequent state depends only on the previous one [8]. Although Markov chains are not actually a neural network, they are very similar.

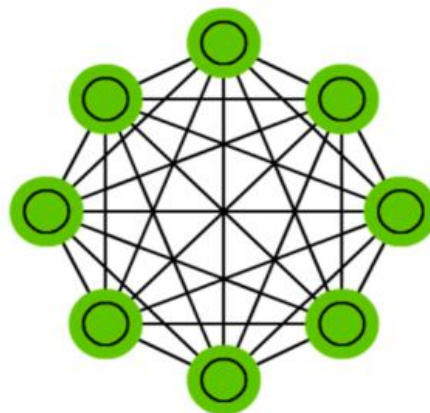


Figure 5: Markov Chains

The Boltzmann machine (figure 6) is very similar to the Hopfield network, but in it some neurons are marked as incoming and some as hidden. Input neurons then become output neurons [12].

The Boltzmann machine is a stochastic network. The training is based on the method of back propagation of error algorithm or a comparative difference. In General, the learning process is very similar to that of the Hopfield network.

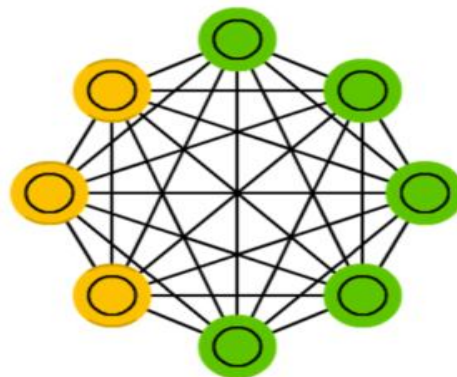


Figure 6: Boltzmann Machine

Based on the above, we can present a classical neural network as a computational graph containing (figure 7):

- input vertexes x ;
- vertices are neurons with their output values a ;
- vertexes responsible for bias b ;
- edges, multiply the output value of the previous layer by the corresponding coefficients of the weight matrix w ;
- hypothesis-the result of the last layer output.

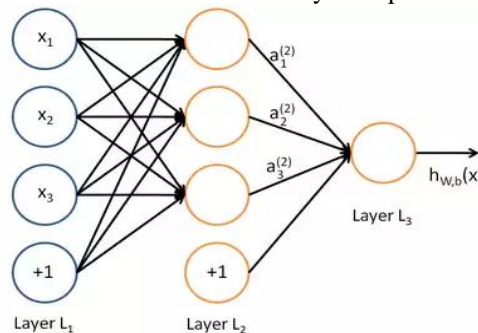


Figure 7: Representation of the neural network architecture used

3. CONCLUSION

Having analyzed the currently existing intelligent technologies, as well as intelligent technologies for object-oriented programming, their disadvantages and advantages were identified.

On the basis of this was carried out the development of intelligent decision-making: developed a method for training a neural network integrated intelligent system in the development of models for the pattern Model View Controller is also the chosen architecture of the neural network for choice of programming models for the pattern Model View Controller neural networks to create models. Based on the above, we can say that our neural network passed the test and works correctly.

Modern machine learning algorithms, such as neural networks that can independently find nonlinear patterns in data. But in order for this to happen, there must be a lot of data. Sometimes a lot.

Based on the principle of operation of the neural network, we can conclude that the more variations in software configurations of libraries and frameworks, as well as the ratio of models to fields, the better it will produce results.

REFERENCES

1. Y. LeCun, Y. Bengio, G. Hinton, **Deep learning**. *Nature* 521, 436–444 (2015). doi:10.1038/nature14539pmid:26017442
2. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, C. I. Sánchez, **A survey on deep learning in medical image analysis**. *Med. Image Anal.* 42, 60–88 (2017). doi:10.1016/j.media.2017.07.005pmid:28778026
3. A. Graves, A. Mohamed, G. Hinton, in **Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE, 2013)**, pp. 6645–6649.
4. K. Cho et al., **Learning phrase representations using RNN encoder-decoder for statistical machine translation**. *arXiv:1406.1078* [cs.CL] (3 June 2014).
5. A. Krizhevsky, I. Sutskever, G. E. Hinton, in **Advances in Neural Information Processing Systems 25** (NIPS 2012), F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds. (Curran Associates, 2012), pp. 1097–1105.
6. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, **Mastering the game of Go with deep neural networks and tree search**. *Nature* 529, 484–489 (2016). doi:10.1038/nature16961pmid:26819042
7. U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, D. Psaltis, **Learning approach to optical tomography**. *Optica* 2, 517 (2015). doi:10.1364/OPTICA.2.000517
8. Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, A. Ozcan, **Deep learning microscopy**. *Optica* 4, 1437 (2017). doi:10.1364/OPTICA.4.001437
9. K. H. Jin, M. T. McCann, E. Froustey, M. Unser, **Deep convolutional neural network for inverse problems in imaging**. *IEEE Trans. Image Process.* 26, 4509–4522 (2017). doi:10.1109/TIP.2017.2713099pmid:28641250
10. Y. Rivenson, Y. Zhang, H. Gunaydin, D. Teng, A. Ozcan, **Phase recovery and holographic image reconstruction using deep learning in neural networks**. *Light Sci. Appl.* 7, 17141 (2018). doi:10.1038/lsa.2017.141
11. A. Sinha, J. Lee, S. Li, G. Barbastathis, **Lensless computational imaging through deep learning**. *Optica* 4, 1117 (2017). doi:10.1364/OPTICA.4.001117
12. K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, F. Knoll, **Learning a variational network for reconstruction of accelerated MRI data**. *Magn. Reson. Med.* 79, 3055–3071 (2018). doi:10.1002/mrm.26977pmid:29115689
13. Y. Rivenson, H. CeylanKoydemir, H. Wang, Z. Wei, Z. Ren, H. Günaydin, Y. Zhang, Z. Göröcs, K. Liang, D. Tseng, A. Ozcan, **Deep learning enhanced mobile-phone microscopy**. *ACS Photonics* 5, 2354–2364 (2018). doi:10.1021/acsp Photonics.8b00146
14. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, **Gradient-based learning applied to document recognition**. *Proc. IEEE* 86, 2278–2324 (1998). doi:10.1109/5.726791
15. D. Ciregan, U. Meier, J. Schmidhuber, in **Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)** (IEEE, 2012), pp. 3642–3649.
16. C.-Y. Lee, P. W. Gallagher, Z. Tu, in **Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS, 2016)**, pp. 464–472.
17. M. A. Ranzato, C. Poultney, S. Chopra, Y. LeCun, in **Advances in Neural Information Processing Systems 20** (NIPS 2007), J. C. Platt, D. Koller, Y. Singer, S. T. Roweis, Eds. (Curran Associates, 2007), pp. 1137–1144.
18. H. Xiao, K. Rasul, R. Vollgraf, **Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms**. *arXiv:1708.07747* [cs.LG] (25 August 2017).