DOI: https://doi.org/10.5281/zenodo.14777596

UDC: 004.735

ANALYSIS OF FUZZY NEURAL CONTROL SYSTEMS FOR TELECOMMUNICATION NETWORKS

Alevtina Aleksandrovna Muradova
TUIT named after Muhammad al-Khwarizmi, PhD, associate professor of the Department of Telecommunication Engineering
a.muradova1982@inbox.ru

Svetlana Aleksandrovna Sadchikova
TUIT named after Muhammad al-Khwarizmi, PhD, associate professor of the Department of Telecommunication Engineering
sadchikova047@gmail.com

ABSTRACT

The article presents an analysis of fuzzy neural control systems for telecommunication networks. Neural networks and fuzzy logic can solve problems that are beyond the capabilities of traditional control systems in network telecommunications. This article examines the operation of such systems and the advantages they provide in areas such as high-speed processing of images, files, packets, and other types of multimedia information.

Keywords: *artificial neural networks (ANN), model complex relationships, two-dimensional vector, fuzzy logic algorithms, real-time control systems, data structure.*

АНАЛИЗ НЕЧЕТКИХ НЕЙРОННЫХ СИСТЕМ УПРАВЛЕНИЯ ДЛЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ

АННОТАЦИЯ

В статье представлен анализ нечетких нейронных систем управления для телекоммуникационных сетей. Нейронные сети и нечеткая логика позволяют решать задачи, выходящие за рамки возможностей традиционных систем управления в сетях телекоммуникации. В статье рассматривается работа таких систем и преимущества, которые они предоставляют в таких областях,

как высокоскоростная обработка изображений, файлов, пакетов и других видов мультимедийной информации.

Ключевые слова: *искусственные нейронные сети (ИНС), сложные моделируемые отношения, двумерный вектор, алгоритмы нечеткой логики, системы управления в реальном времени, структура данных.*

INTRODUCTION

While engineers working in the field of automatic control were busy transitioning from traditional electromechanical and analog control technologies to digital mechatronic control systems that integrate computerized algorithms for analysis and decision making, new computer technologies appeared on the horizon that could cause even more significant changes. Neural networks and fuzzy logic have already found wide application and will soon change the way automatic control systems are built and programmed.

Traditional computers have a von Neumann architecture, which is based on the sequential processing and execution of explicitly given commands. Artificial neural networks (ANN) are built on a different architecture. They are assembled from very simple processor units combined into a system with a high level of parallelism. This system executes implicit commands based on pattern recognition on data inputs from external sources. Fuzzy logic also turns traditional ideas upside down. Instead of precise measurements that establish a value’s position on a given scale (temperature is 23°C), fuzzy information indicates the degree of membership in ill-defined overlapping sets (on the colder side of warm). Computers (inference engines) that use these concepts are able to solve complex problems that are beyond the capabilities of traditional control systems. An artificial neural network (ANN) - an interconnected collection of artificial ‘neurons’ that uses a mathematical or computational model to process information based on the connectivity of the computers [1-3].

RESEARCH OBJECT AND METHODS

In most cases, an ANN is an adaptive system that changes its structure in response to external or internal information passing through the network. Instead of calculating numerical results from numerical inputs, ANNs model complex relationships between inputs and outputs or discover patterns in the data. Elementary nodes (also called neurons, neurodes, processing elements, or units) are connected together to form a network of nodes. Their usefulness comes from the ability to implement inference algorithms that change the strengths or weights of network connections to produce the desired signal flow.

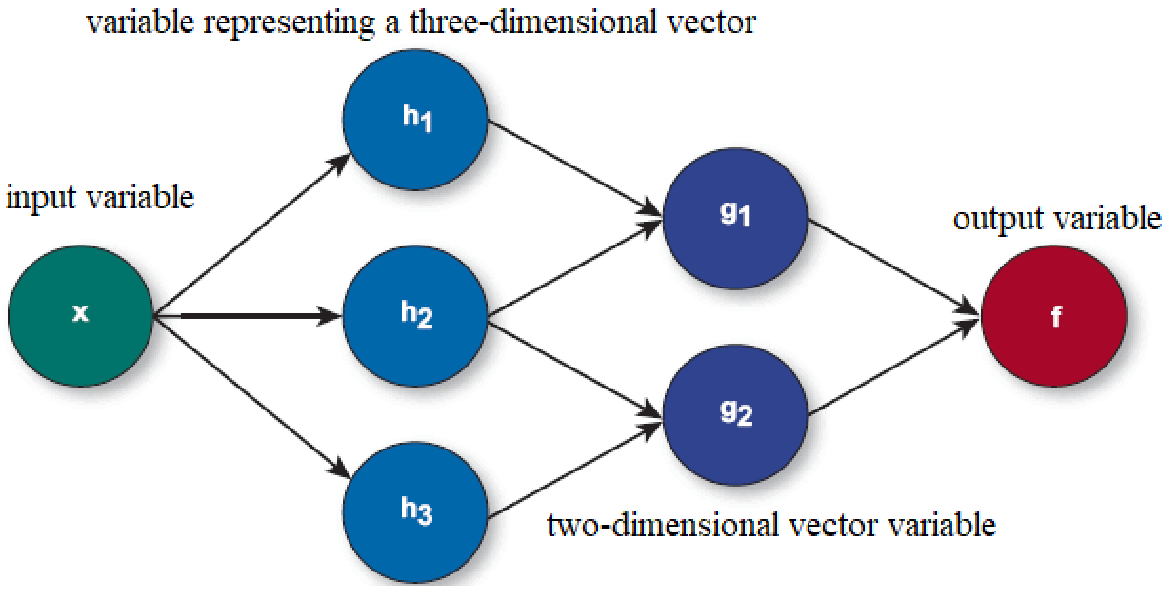

Fig.1. Artificial neural network architecture

In this example of an artificial neural network, the variable h, which is a three-dimensional vector, depends on the input variable x. Next, g, a two-dimensional vector variable, depends on h, and finally the output variable f depends on g. The most interesting feature is the learning capability, which in practice means optimizing some quantity, often called the "price", which indicates the correctness of the result in the context of the problem being solved. For example, the price in the classic traveling salesman problem is the time required to travel the entire sales area, stopping at all the required points, and arrive at the starting point. The shorter route gives the best solution.

To solve this problem, von Neumann computers must establish all possible routes, and then try each route in turn, adding up the time delays to determine the total delay for that route. After calculating the sums for all possible routes, the computer simply chooses the shortest one. In contrast, ANNs consider all routes in parallel to find configurations that minimize the total travel time. Using these configurations minimizes the final route. Learning consists of identifying configurations that, based on previous experience, provide route optimization strategies [4,5].

Fuzzy logic is derived from fuzzy set theory, which deals with reasoning that is approximate rather than exact. Truth in fuzzy logic indicates membership in ill-defined sets. In fuzzy logic, decisions can be made based on ill-defined, but nonetheless very important, characteristics. Fuzzy logic allows membership values to vary between 0 and 1 inclusive, as well as the use of vague concepts such as "a little", "somewhat", and "very". This allows partial membership in a set in a special way. A basic application can be described by subranges of a continuous variable. For example, the temperature range of an anti-lock braking system may have several separate membership functions defining the temperature intervals necessary for proper brake control. Each function represents a temperature value that is a truth value between 0 and 1. These truth values can then be used to select a brake control method.

RESEARCH RESULTS AND THEIR DISCUSSION

Fast Fuzzy Logic for Real-Time Control. While any microcontroller or computer can implement fuzzy logic algorithms in software, doing so can be inefficient due to its slow speed and memory requirements. Jim Seebigtroth, an automotive systems engineer in the Transportation and Standard Products Group, a microcontroller division of Freescale Semiconductor, says the company’s HC12 and HCS12 microcontrollers solve this problem very efficiently by adding four instructions specifically designed to implement the core parts of a fuzzy logic inference engine [6,7].

The core program for the general-purpose inference engine, which processes unweighted rules, is approximately 57 bytes of object code (approximately 24 lines of assembly code). Sibigtroth notes that the 25 MHz HCS12 can execute a complete inference sequence for two inputs and one output, with seven labels for each input and output, in about 20 µs. An equivalent program for the 8 MHz MC68HC11 (without fuzzy logic instructions) would require approximately 250 bytes of object code and about 750 µs of execution time. Even if the MC68HC11 could process the program at the same speed as the HCS12, the fuzzy logic instructions make the program 4 times smaller and reduce the execution time by 12 times. Such short recognition intervals make it possible to use fuzzy logic algorithms in real-time control systems without expensive hardware or large programs.

Image Processing. A powerful control system can be created using fuzzy logic-based ANN decision making. It is obvious that the two concepts work well together: an inference algorithm with three fuzzy states (cold, warm, hot) could be implemented in hardware using truth values (0.8, 0.2, 0.0) as inputs to three neurons, each representing one of the three sets. Each neuron processes the input value according to its function and obtains an output value, which will then be the input value for the second layer of neurons, and so on.

For example, a neural computer for image processing can remove numerous restrictions on video recording, lighting, and hardware settings. This degree of freedom is possible because the neural network allows you to build a recognition mechanism by learning from examples. As a result, the system can be trained to recognize good and bad products in strong and weak lighting, when they are located at different angles. The inference engine starts by “scoring” the lighting conditions (establishing the degree of similarity to other lighting conditions under which the system knows how to act). The system then makes a decision about the content of the image using criteria based on these lighting conditions. Since the system treats lighting conditions as fuzzy concepts, the inference engine easily infers new conditions from known examples [8-10].

The more examples the system learns, the more experience the image processing engine gains. This learning process can be quite easily automated, for example by pre-sorting parts with similar properties into groups to learn areas of similarity and difference. These observed similarities and differences can then provide information to the ANN, whose task is to sort incoming parts into these categories. Thus, the success of the system does not depend on the cost of the equipment, but on the number of images needed to train and build a robust inference engine.

An image processing neural computer is suitable for applications where diagnostics rely on the experience and expert judgment of the operator, rather than on models and algorithms. The processor can build a recognition engine from simple annotations of the image made by the operator, then extract features or feature vectors from the annotated objects and feed them to the neural network. Feature vectors describing visible objects can be as simple as pixel row values, histogram or intensity distributions, intensity distribution profiles, or gradients along the corresponding axes. More complex features can include elements of the wavelet transform and the fast Fourier transform.

After training on examples, the neural network is capable of generalization and can classify situations never seen before by associating them with similar situations from examples. On the other hand, if the system tends to be too free and generalize situations, its behavior can be corrected at any time by learning counterexamples. From the neural network point of view, this operation consists of reducing the areas of influence of existing neurons to accommodate new examples that are in conflict with the existing mapping of the solution space.

An important factor determining the acceptance of ANNs is independent and adaptive learning. This means that the device must be able to learn an object with little or no human intervention. In the future, for example, dolls could recognize the face of a child unwrapping them for the first time and ask for their name. Self-learning for a cell phone could involve learning the fingerprint of its first owner. Owner identification could also be enhanced by combining face, fingerprint, and speech recognition in a single device. In a self-learning environment, the device must build its own recognition mechanism that works best in its operating environment. For example, an intelligent doll must recognize its original owner regardless of their hair and skin color, location, or season [11,12].

Initially, the mechanism must use all the feature extraction techniques it knows. This will lead to the formation of a series of intermediate mechanisms, each designed to identify the same categories of objects, but based on the observation of different features (color, graininess, contrast, edge thickness). The overall mechanism can then evaluate the performance of the intermediate mechanisms, selecting those that provide the best performance and/or accuracy.

PiscesVMK manufactures process equipment for fish processing on board and in coastal factories. The company's customers are fish processing vessels that fish all year round in the North Sea and the Atlantic Ocean for a variety of fish species. These customers want to fill their holds as quickly as possible with the highest quality catch while using a minimum number of workers. Typically, the fish are brought on board using nets and unloaded into containers on a conveyor belt, which takes them through cleaning, cutting and filleting machines. Possible deviations include the wrong species, damaged fish, more than one fish in a container and its incorrect position before entering the cutting machine. Implementing such inspections with traditional image processing is difficult because the dimensions, shape and volume are difficult to describe mathematically. In addition, these parameters can vary depending on the location and season.

Pisces has installed more than 20 systems based on the Iris intelligent camera from Matrox and the Cognit Sight recognition engine from General Vision. The camera is mounted above the conveyor so that the fish pass underneath it just before entering the filleting machine. The camera is linked to a Siemens Simatic S7-224 controller (PLC) and to a local area network (LAN). A strobe light source mounted next to the camera is triggered each time a new container comes into view. The camera's connection to the LAN is necessary to perform three operations: adjusting the converter to ensure focus and proper image contrast, training the recognition mechanism, and accessing statistics that continuously report the number of good and bad fish.

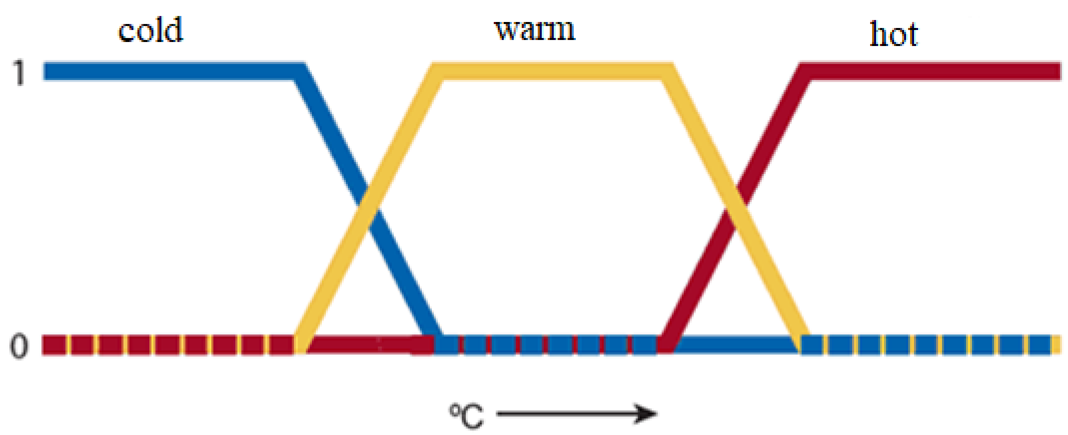

Fig.2. Data structure of fuzzy logic

In this figure, “cold”, “warm” and “hot” represent sets covering the entire temperature scale. A point on this scale has three “truth values” – one for each of the three sets – which indicate its relative similarity to each set. For a particular temperature, the three truth values (0.8, 0.2, 0.0) can be interpreted as describing the temperature of objects in terms of “quite cold”, “lukewarm” and “not hot”.

SCIENTIFIC RESEARCH RESULTS AND CONCLUSION

The transducer is configured only once, when the camera is installed in a waterproof housing. Training is performed at the beginning of each swim using fish samples from the first catch or by loading an existing file.

Once the camera has a knowledge base, it can start recognizing fish autonomously, without communication with a personal computer. The ANN sorts them into categories of “accepted”, “rejected”, “for processing” or “empty”. This signal is sent to a PLC, which controls two brushes that direct the appropriate fish to the disposal or processing tanks. The PLC is also linked to a magnetic sensor that generates a trigger signal every time a fish container passes under the camera. Pisces has now installed over 20 systems on 5 different fishing fleets in Norway, Iceland, Scotland and Denmark. The system evaluates 360 conveyor containers per minute on herring lines, but it can work even faster. A network of 80 neurons achieved 98% accuracy in classifying 16 tons of fish. Fishermen are happy with the system because of its reliability, flexibility and ease of use. The benefits include shorter sailing times, higher quality catches and income distributed among fewer fishermen [13,14].

FINAL CONCLUSION

In discrete manufacturing, neural networks have found applications in vehicle control, telecommunication network monitoring systems, pattern recognition in radar systems, personal recognition, object recognition, handwriting, gestures and speech. Fuzzy logic is already used to control cars and other vehicle subsystems such as ABS and cruise control, as well as air conditioning, cameras, digital image processing, computer game AI, and pattern recognition in remote sensor systems. Similar “soft computing” technologies are also used to create reliable respirator battery chargers. In the continuous and batch manufacturing industries, fuzzy logic and neural networks are the basis for some self-tuning controllers. Some microcontrollers and microprocessors are optimized for fuzzy logic, so that systems can run even faster.

REFERENCES

1. Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. ISBN 978-0-387-31073-2.
2. Ferrie, C., & Kaiser, S. (2019). *Neural Networks for Babies*. Sourcebooks. ISBN 978-1-4926-7120-6.
3. Bozinovski, S. (2014). Modeling mechanisms of cognition-emotion interaction in artificial neural networks. *Procedia Computer Science*. p. 255-263.
4. Jospin, L.V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-On Bayesian Neural Networks-A Tutorial for Deep Learning Users. *IEEE Computational Intelligence Magazine*. Vol. 17, №2. pp. 29-48. doi: 10.1109/mci.2022.3155327.
5. Ferreira, C. (2006). Designing Neural Networks Using Gene Expression Programming. *Applied Soft Computing Technologies: The Challenge of Complexity*. Springer-Verlag. pp. 517-536.
6. Alaloul, W.S., & Qureshi, A.H. (2019). Data Processing Using Artificial Neural Networks. *Dynamic Data Assimilation - Beating the Uncertainties*. doi: 10.5772/intechopen.91935.
7. Ganesan, N. (2010). Application of Neural Networks in Diagnosing Cancer Disease Using Demographic Data. *International Journal of Computer Applications*. 1 (26): 81 - 97. doi:10.5120/476-783.
8. Nabian, M.A., & Meidani, H. (2017). Deep Learning for Accelerated Reliability Analysis of Infrastructure Networks. *Computer-Aided Civil and Infrastructure Engineering*. 33 (6): 443 - 458. doi:10.1111/mice.12359. S2CID 36661983.
9. Weng, J. (2012). *Natural and Artificial Intelligence: Introduction to Computational Brain-Mind.* Wayback Machine, BMI Press, ISBN 978-0-9858757-2-5.
10. Tahmasebi, H. (2012). A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation. *Computers & Geosciences*. 42: 18-27. doi: 10.1016/j.cageo.2012.02.004.
11. Bansod, N.A, Kulkarni, M., & Patil, S. H. (2005). *Soft Computing- A Fuzzy Logic Approach*. In Bharati Vidyapeeth College of Engineering (ed.). Soft Computing. Allied Publishers. p. 73. ISBN 978-81-7764-632-0.
12. Gerla, G. (2016). Comments on some theories of fuzzy computation. *International Journal of General Systems.* 45 (4): 372–392. doi: 10.1080/03081079.2015.1076403.
13. Francesco, V. (2017). Fuzzy Multiple Attribute Conditions in fsQCA: Problems and Solutions. *Sociological Methods & Research*. 49 (2): 312-355. doi: 10.1177/0049124117729693.
14. Di, S., & Bruno, N. (2013). On the Need of a Standard Language for Designing Fuzzy Systems. *Studies in Fuzziness and Soft Computing.* Vol. 296. pp. 3-15. doi:10.1007/978-3-642-35488-5.