High-speed Fuzzy Inference Machine Learning Device Based on Single-Layer Area Ratio Defuzzifier

Maxim B. Bobyr[1,*,†], Bogdan Bondarenko[2] and Alexandr Malyshev[3]

[1]Southwest State University of Russia (SWSU), 94, 50 Let Oktyabrya St, Kursk, 305000, Russian Federation
[2]Southwest State University of Russia (SWSU), 94, 50 Let Oktyabrya St, Kursk, 305000, Russian Federation
[2]Southwest State University of Russia (SWSU), 94, 50 Let Oktyabrya St, Kursk, 305000, Russian Federation

Abstract

This article discusses a variant of a High-speed Fuzzy Inference Machine Learning Device Based on Single-Layer Area Ratio Defuzzifier. The developed mathematical model is presented, the steps by which this system operates are described. The results of modeling in Simulink are also presented and show a 1.5x increase in training speed.

Keywords

machine learning, single-layer area ratio defuzzifier, fuzzy logic

1. Introduction

In an era of rapidly advancing technology, traditional binary logic systems often fall short when dealing with the complexity and uncertainty of real-world scenarios [1], [2], [3], [4]. Fuzzy logic systems, which mimic human reasoning by handling partial truths and uncertainties, have emerged as a powerful alternative. By integrating machine learning techniques, these systems are now capable of adapting and improving their performance over time, creating a dynamic approach to problem-solving. This article explores the foundations of fuzzy logic systems with learning capabilities, their advantages, and their growing applications across various fields, from robotics to data analysis [5], [6].

2. High-speed Fuzzy Inference Machine Learning Device Based on Single-Layer Area Ratio Defuzzifier

The main purpose of the developed High-speed Fuzzy Inference Machine Learning Device is to implement the system learning function and improve computing performance. This is achieved by adding feedback from the training unit to the defuzzification unit, which allows for the training of the fuzzy logic device [7], [8]. Some operations in the defuzzification unit were also excluded, which reduced the computational performance time of the defuzzification process to 180 ns.

The result of a high-speed fuzzy logic inference machine learning device based on a single-layer defuzzifier of the area ratio method is the generation and transformation of input data into a single specified crisp value at the output of the fuzzy logic system. This type of device can be used for image classification or thermocouple control tasks [9], [10].

Also, an ontological model of neuro-fuzzy learning based on the area ratio method was developed:

$$O = \left\langle \underset{in=1}{\overset{2}{O_{in}}},\ \underset{im=1}{\overset{6}{O_{im}}},\ \underset{def=1}{\overset{5}{O_{def}}},\ \underset{l=1}{\overset{2}{O_{l}}}, \right\rangle \tag{1}$$

The 2024 Sixth Doctoral Symposium on Intelligence Enabled Research (DoSIER 2024), November 28–29, 2024, Jalpaiguri, India

*Corresponding author.
†These authors contributed equally.
maxbobyr@gmail.com (M. B. Bobyr); sikersinko@gmail.com (B. Bondarenko); alta76@yandex.ru (A. Malyshev)
0000-0002-5400-6817 (M. B. Bobyr); 0000-0001-5415-9015 (B. Bondarenko); 0000-0002-9938-3456 (A. Malyshev)

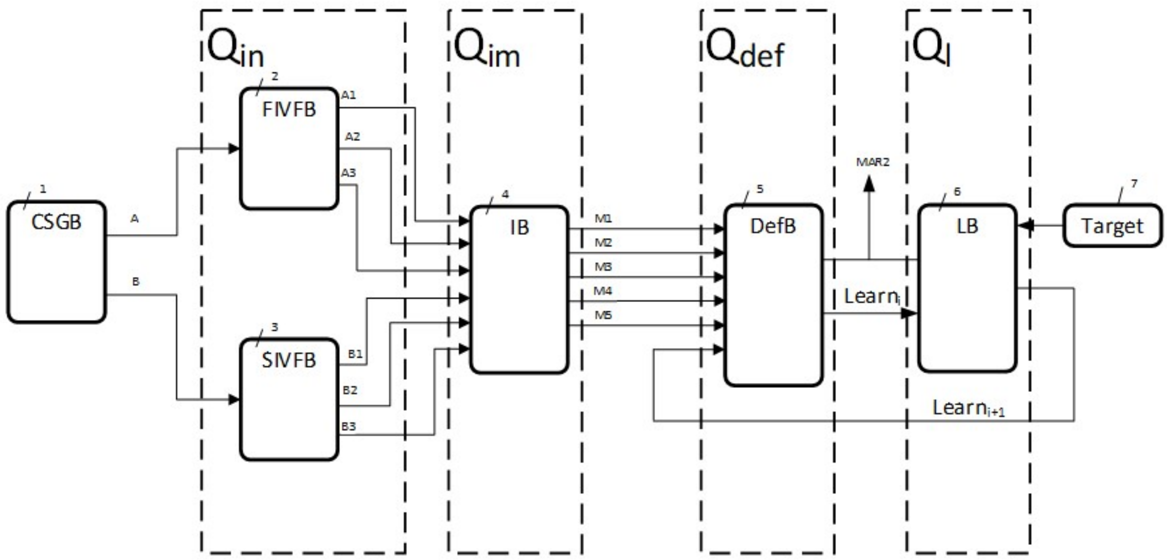


Figure 1: High-speed Fuzzy Inference Machine Learning Device Based on Single-Layer Area Ratio Defuzzifier

where O_{in} is ontological model of input variable (*in*=1…2, where *in* is number of input variables); O_{im} is ontological model of the implication block (*im*=1…6, where *im* is number of implication methods); O_{def} is ontological defuzzification model (*def*=1…5, where *def* is number of defuzzification methods); O_l is ontological model of the training block (*l*=1…2, where *l* is number of training methods).

The working principle of the high-speed fuzzy inference machine learning device (Fig. 1) based on the single-layer area ratio defuzzifier consists of five steps [11], [12]. The values of two variables «A» and «B» are generated in block 1 – Control Signal Generator Block (CSGB). Then the variable «A» goes to block 2 – First Input Variable Fuzzification Block (FIVFB), and the variable «B» is transferred to block 3 – Second Input Variable Fuzzification Block (SIVFB). In FIVFB 2, the first three triangular functions «A1», «A2», «A3» are generated in First, Second and Third Triangular Function Formation Blocks (FTFFB, STFFB, TTFFB) respectively. In SIVFB 3, the second three triangular functions «B1», «B2», «B3» are generated in FTFFB, STFFB, TTFFB too. Each of the triangular membership functions has a form similar to Fig. 3. The triangular functions «A1», «A2», «A3», «B1», «B2», «B3» are transferred to block 4 – Implication Block (IB), where they are subject to implication. As a result, at the output of IB 4, five variables «M1», «M2», «M3», «M4», «M5» are transferred to block 5 – Defuzzification Block (DefB). In DefB 5, the defuzzification process takes place, during which the resulting variable «MAR2» is calculated. And after receiving MAR2, the calculations go to block 6 – Learning Block (LB).

The calculation of the resulting variable "MAR2" (Fig.1) in the high-speed fuzzy inference machine learning device based on the single-layer area ratio defuzzifier [13], [14] is carried out by the steps described below.

Step 1. Generation of input variables by counters (Fig.2):

$$A = clck_1 \times gain_1 + a \tag{2}$$

$$B = clck_2 \times gain_2 + b \tag{3}$$

where $clck_1$ is clock generator (block 1.2), $gain_1$ is signal booster unit (block 1.5), $clck_2$ is clock generator (block 1.4), $gain_2$ is signal booster unit (block 1.6), *a* is initial value of the first membership function (block 1.1), *b* is initial value of the second membership function (block 1.3).

Step 2. The formation of the triangular membership function (Fig.3 and Fig.4) is calculated using the formula:

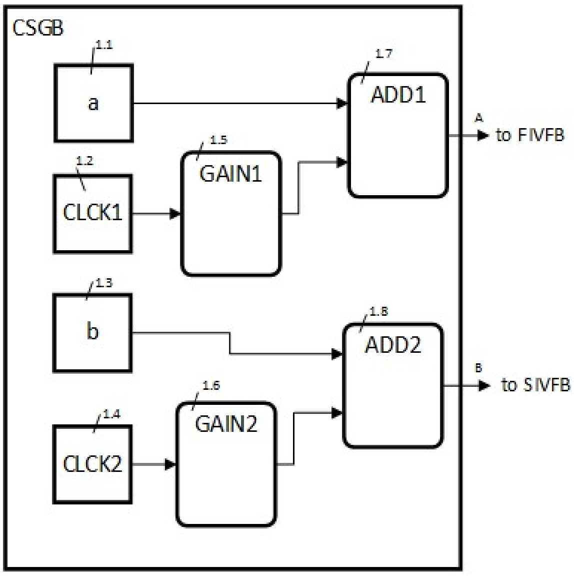


Figure 2: Control signal generator block

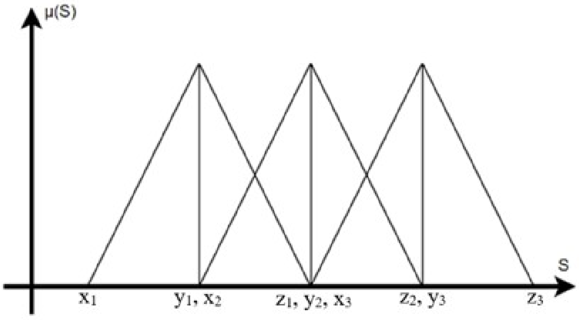


Figure 3: Graph of the triangular membership function

$$\mu(s,x,y,z) = \begin{cases} \frac{s-x}{y-x}\text{, if } x < s < y \\ \frac{z-s}{z-y}\text{, if } y < s < z \\ 0\text{, else} \end{cases} \tag{4}$$

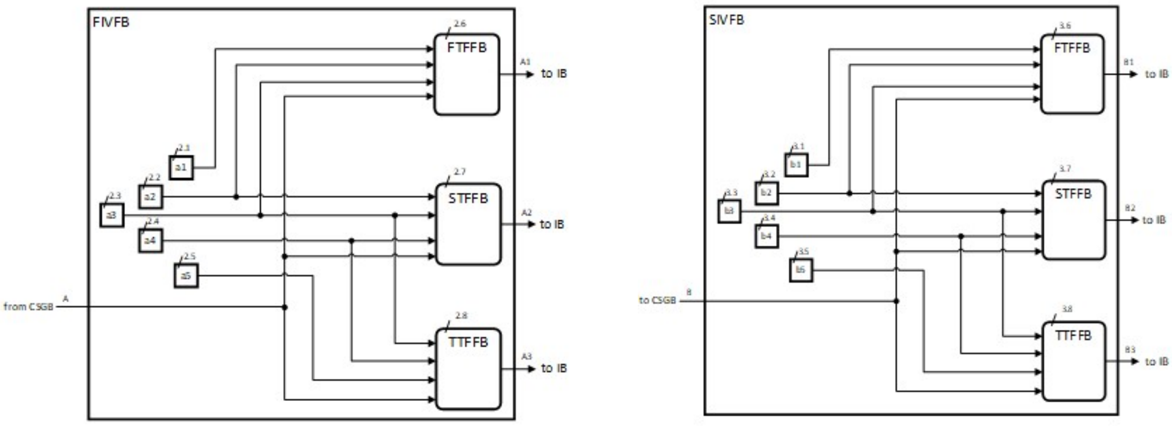
where s is input value A or B coming from the block 1 (Fig.3 and Fig.4), x, y, z are membership function labels. Their value stored in blocks 2.1÷2.5 and 3.1÷3.5. The variables for «x», «y», and «z» are selected depending on the required type of membership function. In our case, a triangular membership function is used, which is shown in Fig.3. An example of their calculation is presented above Eq.(4). This stage of calculation is called composition and occurs in FTFFB, STFFB, TTFFB (blocks 2.6-2.8, 3.6-3.8). As a result of this operation, three output values «A1», «A2» and «A3» are formed at the output of the FIVFB block, and at the output of the SIVFB block «B1», «B2», and «B3» values (Fig.4).

The implementation of Eq.(4) is presented in the form of logical blocks in Fig.4.

Step 3. Functions «A1», «A2», «A3», «B1», «B2», «B3» come from the outputs of the FIVFB, SIVFB blocks to the inputs of the IB block. The implication process [15], [16] (Fig.6) of the input variables is calculated according to the established fuzzy rules:

$$M_1 = \min(A_1, B_1) \tag{5}$$

$$M_2 = \max(\min(A_1, B_2), \min(A_2, B_1)) \tag{6}$$



a) b)

Figure 4: Input Variable Fuzzification Block: (a) First. (b) Second method.

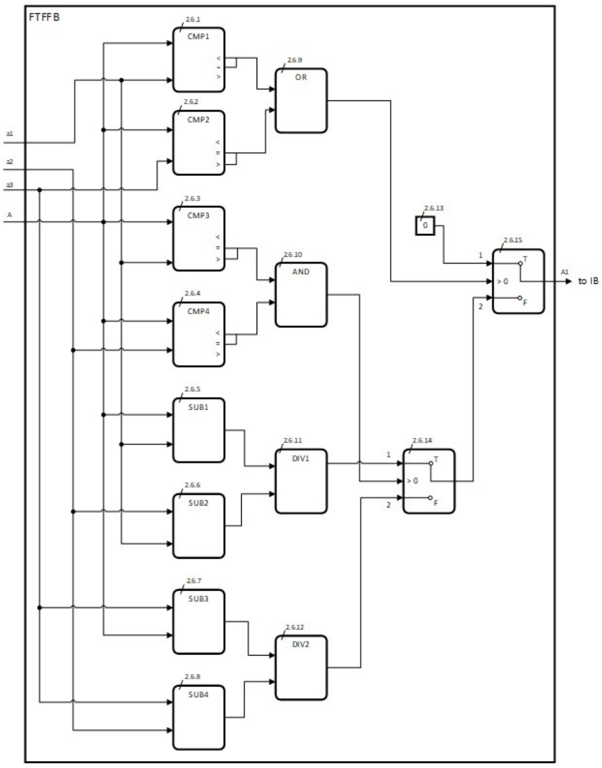


Figure 5: First Triangular Function Formation Block

$$M_3 = \max(\min(A_1, B_3), \min(A_2, B_2), \min(A_3, B_1)) \tag{7}$$

$$M_4 = \max(\min(A_2, B_3), \min(A_3, B_2)) \tag{8}$$

$$M_5 = \min(A_3, B_3) \tag{9}$$

The output values of the block are M1, M2, M3, M4, M5 (Fig.7), and the variable M1 is calculated in block 4.1, M2 is calculated in block 4.10, M3 is calculated in block 4.11, M4 is calculated in block 4.12, M5 is calculated in block 4.9.

Step 4. The defuzzification process (Fig.8). Determining the output value after defuzzification based on the area ratio method according to Eq.10:

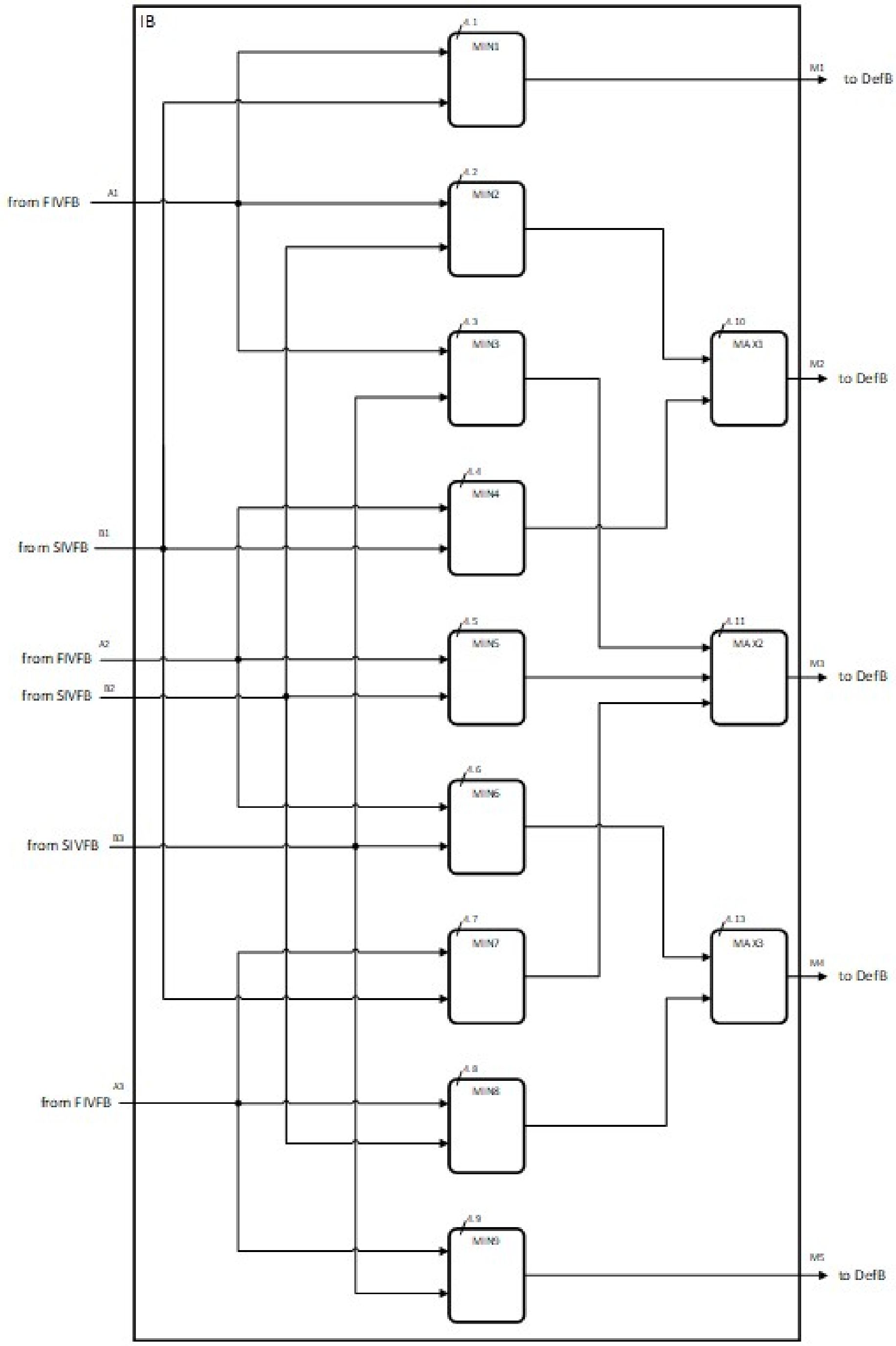


Figure 6: Implication Block

$$MAR2 = \left[\frac{\sum_{i=1}^{n} M_i}{n \times \mathrm{Learn}_{i+1}} \times (Y_{\max} - Y_{\min}) \right] + Y_{\min} \tag{10}$$

where M_i is value transmitted from the IB block 4, *n* is number of output fuzzy membership functions (*n* = 5) (Fig.7), $Learn_{i+1}$ is weighting factor (default 1), Y_{max} is maximum value of the output function (block 5.4), Y_{min} is minimun value of the output function (block 5.5) (Fig.7 and Fig.8: Y_{max} = 250, Y_{min} = 210).

To find the difference between Y_{max} and Y_{min}, input values Y_{max} and Y_{min} are fed to the inputs of subtraction block SUB 5.7. To calculate Eq.10 ten-digit values Y_{min} and Y_{max} are fed to the input of the subtractor SUB 5.7. The output value of the subtractor SUB 5.7, which determines the value of the

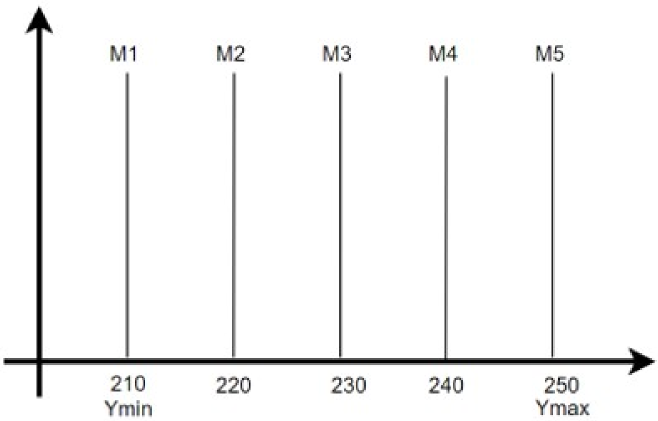


Figure 7: Graph of the output singleton membership function

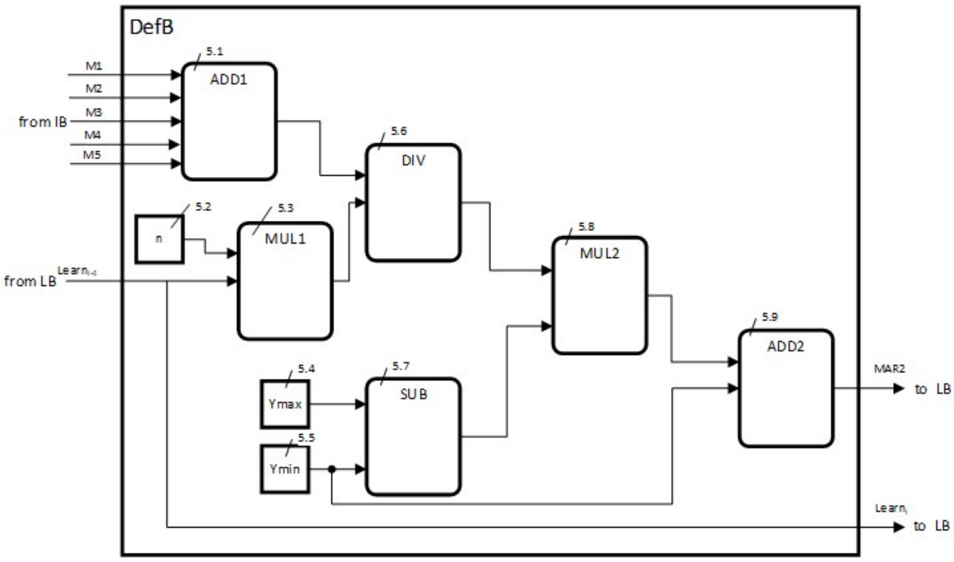


Figure 8: Defuzzification block

domain of definition of the output membership function, is fed to the input of the multiplier MUL 5.6, and D, obtained at the output of the divider DIV 5.3, is fed to the second input of the multiplier MUL 5.6. The output of the multiplier MUL 5.6 is connected to the input of the adder ADD2 5.8. The value Y_{min} is fed to the second input of the adder ADD2 5.8. At the output of the adder ADD2 5.8, the output ten-bit value is calculated after defuzzification based on the area ratio method "MAR2" [17].

Step 5. The training process (Fig.9). The output value after learning is determined according to the formula:

$$\text{Learn}_{i+1} = \text{Learn}_i + ([\text{MAR2} - Y_{\text{target}}] \times \text{Mult}), \text{until MAR2} - Y_{\text{target}}| \leq T \tag{11}$$

where *Mult* is learning rate (default 0.04), T is threshold coefficient (default 0.01) (Fig.9, block 6.2), Y_{target} is expected output value after defuzzification.

3. Simulation Results

The above mathematical model was simulated in the Simulink software for modeling, simulation and analysis of multidomain dynamic systems. The results of the training process simulation are shown in

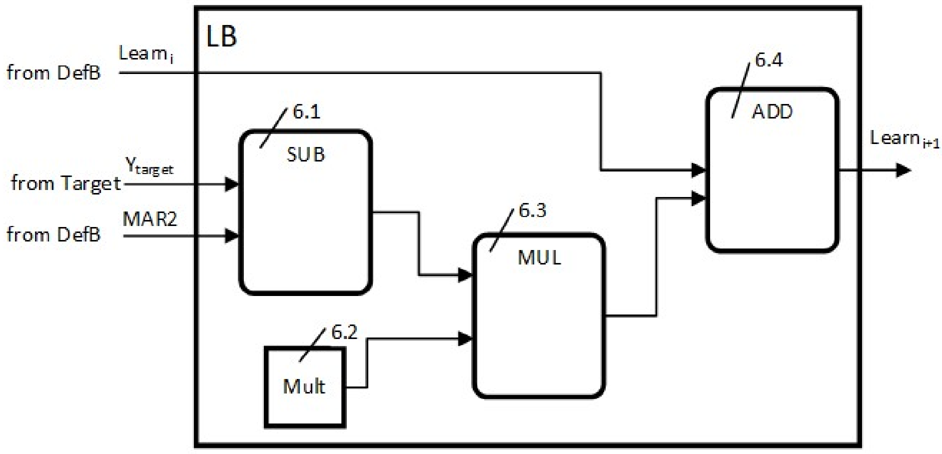


Figure 9: Learning Block

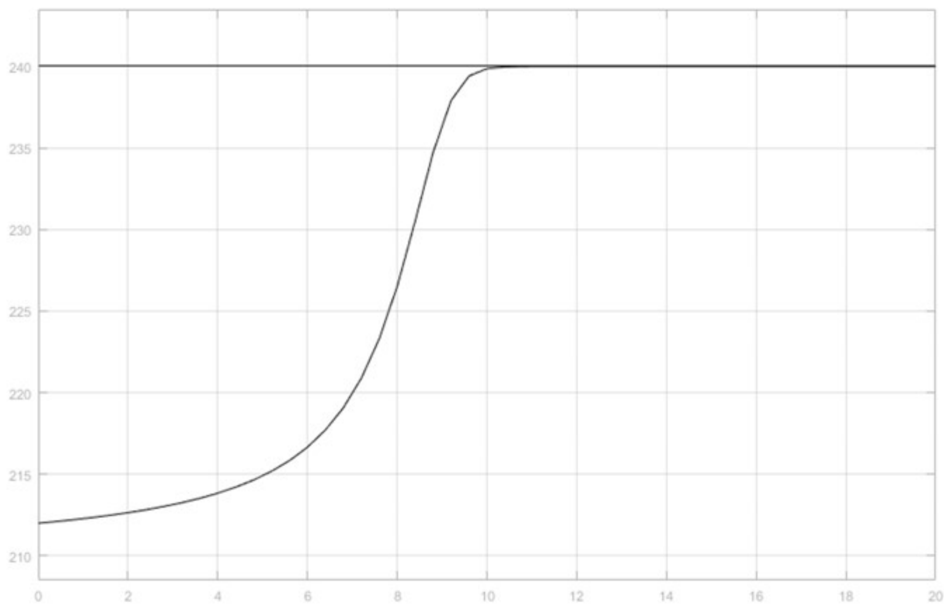


Figure 10: Simulation result of Single-Layer Area Ratio Defuzzifier

Fig.10.

The system performance was also simulated for other defuzzification methods. Fig.11 shows the simulation results: 1 is Single-layer area ratio defuzzifier, 2 is Multi-layer area ratio defuzzifier, 3 is Defuzzifier based on the center of gravity method. The result of the training process of a single-layer area ratio defuzzifier: 10 cycles; multi-layer area ratio defuzzifier: 11.35 cycles; defuzzifier based on the center of gravity method: 15.5 cycles.

The simulation results show that the speed of the single-layer area ratio defuzzifier is 1.55 times higher than that of the defuzzifier based on the center of gravity method, and 1.135 times higher than that of the multi-layer area ratio defuzzifier.

4. Conclusion

The developed method is described, the ontological and mathematical model of High-speed Fuzzy Inference Machine Learning Device Based on Single-Layer Area Ratio Defuzzifier is presented. In

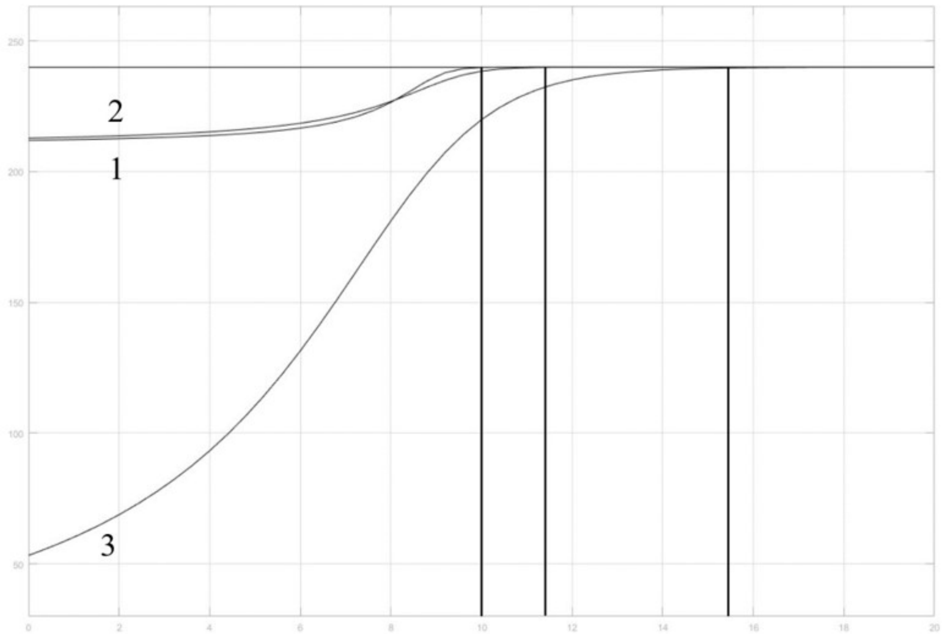


Figure 11: Comparison of the results of the training process with different defuzzifiers

addition, the proposed system was simulated in Simulink and compared with the performance of the center-of-gravity defuzzifier and the multilayer modification of the area ratio method.

Thus, the high-speed fuzzy inference machine learning device based on the single-layer area ratio defuzzifier allows determining a single value after defuzzification, provides increased performance by simplifying the defuzzifier structure, and allows training the fuzzy inference system to the target value.

5. Acknowledgments

The work was prepared as part of the implementation of the RSF project No. 24-21-00055. The authors are grateful to the Foundation for their support.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Y. Liu, H. Huang, Brand digital marketing under intranet security control based on the machine learning classification algorithm, Security and Communication Networks (2021). doi:10.1155/2021/9977221.
[2] W. V. Leekwijck, E. E. Kerre, Defuzzification: Criteria and classification, Fuzzy Sets and Systems 108 (1999) 159–178. doi:10.1016/s0165-0114(97)00337-0.
[3] A. A. Romanov, A. A. Filippov, N. G. Yarushkina, Adaptive fuzzy predictive approach in control, Mathematics 11 (2023). doi:10.3390/math11040875.
[4] A. Piegat, Fuzzy Modeling and Control, 1 ed., Physica Heidelberg, 2001. doi:10.1007/978-3-7908-1824-6.
[5] S. Eti, S. Yuksel, H. Dincer, D. Pamucar, M. Deveci, G. O. Olaru, A machine learning and fuzzy logic model for optimizing digital transformation in renewable energy: Insights into industrial

information integration, Journal of Industrial Information Integration 42 (2024). doi:10.1016/j.jii.2024.100734.

[6] H. N. Do, H. T. Phan, N. T. Nguyen, Multimodal sentiment analysis using deep learning and fuzzy logic: A comprehensive survey, Applied Soft Computing 167 (2024). doi:10.1016/j.asoc.2024.112279.

[7] M. Alateeq, W. Pedrycz, Logic-oriented fuzzy neural networks: A survey, Expert Systems with Applications 257 (2024). doi:10.1016/j.eswa.2024.125120.

[8] H. Yin, C. Liu, W. Wu, K. Song, Y. Dan, G. Cheng, An integrated framework for criticality evaluation of oil and gas pipelines based on fuzzy logic inference and machine learning, Journal of Natural Gas Science and Engineering 96 (2021). doi:10.1016/j.jngse.2021.104264.

[9] A. M. T. Marimuthu, J. Rakkiyannan, L. Jakkamputi, S. Vaithiyanathan, S. Gnanasekaran, Vibration-based fault diagnosis study on a hydraulic brake system using fuzzy logic with histogram features, SDHM Structural Durability and Health Monitoring 16 (2022) 383–396. doi:10.32604/sdhm.2022.011396.

[10] D. Rengasamy, J. M. Mase, A. Kumar, B. Rothwell, M. Torres, M. R. Alexander, D. A. Winkler, G. P. Figueredo, Feature importance in machine learning models: A fuzzy information fusion approach, Neurocomputing 511 (2022) 163–174. doi:10.1016/j.neucom.2022.09.053.

[11] D.-N. Truong, J.-S. Chou, Fuzzy adaptive jellyfish search-optimized stacking machine learning for engineering planning and design, Automation in Construction 143 (2022). doi:10.1016/j.autcon.2022.104579.

[12] M. M. Kamruzzaman, S. Alanazi, M. Alruwaili, I. Alrashdi, Y. Alhwaiti, N. Alshammari, Fuzzy-assisted machine learning framework for the fog-computing system in remote healthcare monitoring, Measurement 195 (2022). doi:10.1016/j.measurement.2022.111085.

[13] M. V. Bobyr, N. A. Milostnaya, S. A. Kulabuhov, A method of defuzzification based on the approach of areas' ratio, Applied Soft Computing 59 (2017) 19–32. doi:10.1016/j.asoc.2017.05.040.

[14] M. Bobyr, N. Milostnaya, V. Bulatnikov, The fuzzy filter based on the method of areas' ratio, Applied Soft Computing 117 (2022). doi:10.1016/j.asoc.2022.108449.

[15] E. V. Krieken, E. Acar, F. V. Harmelen, Analyzing differentiable fuzzy logic operators, Artificial Intelligence 302 (2022). doi:10.1016/j.artint.2021.103602.

[16] M. Bobyr, A. Arkhipov, S. Emelyanov, N. Milostnaya, A method for creating a depth map based on a three-level fuzzy model, Engineering Applications of Artificial Intelligence 117 (2023). doi:10.1016/j.engappai.2022.105629.

[17] M. Bobyr, S. Emelyanov, A nonlinear method of learning neuro-fuzzy models for dynamic control systems, Applied Soft Computing Journal 88 (2020). doi:10.1016/j.asoc.2019.106030.