symmetry

MDPI

Article

Neural networks application for processing of the data from the FMICW radars

Lubos Rejfek [1], Tan N. Nguyen [2,*], Pavel Chmelar [1], Ladislav Beran [1] and Phuong T. Tran [2]

[1] Faculty of Electrical Engineering and Informatics, University of Pardubice, 532 10 Pardubice, Czech Republic; Lubos.Rejfek@upce.cz (L.R.); pavel.chmelar@student.upce.cz (P.C.); ladislav.beran@student.upce.cz (L.B.)

[2] Wireless Communications Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam; tranthanhphuong@tdtu.edu.vn

* Correspondence: nguyennhattan@tdtu.edu.vn; Tel.: +84-283-775-5028

Received: date; Accepted: date; Published: date

Abstract: In this paper the results of the Neural Networks and machine learning applications for radar signal processing are presented. The radar output from the primary radar signal processing is represented as a 2D image composed from echoes of the targets and noise background. The Frequency Modulated Interrupted Continuous Wave (FMICW) radar PCDR35 (Portable Cloud Doppler Radar at the frequency 35.4 GHz) was used. Presently, the processing is realized via a National Instruments industrial computer. The neural network of the proposed system is using four or five (optional for the user) signal processing steps. These steps are 2D spectrum filtration, thresholding, unification of the target, target area transforming to the rectangular shape (optional step), and target board line detection. The proposed neural network was tested with sets of four cases (100 tests for every case). This neural network provides image processing of the 2D spectrum. The results obtained from this new system are much better than the results of our previous algorithm.

Keywords: FMICW radar, Radar signal analysis, Neural network, 2D spectrum; Image processing

1. Introduction

The radar PCDR 35 was developed for the Institute of Atmospheric Physics Czech Academy of Sciences. This radar is based on the industrial computer from the National Instruments company. The memory of this system is very limited. The secondary processing describes the data and only important information is saved (number of the targets, distances, reflected powers, Doppler shifts). We are describing the algorithm for the automatic evaluation of the signal from this radar by using neural networks. After comparison with our previous algorithm [1], The benefit of this algorithm is its simple implementation on the Field Programmable Gate Array (FPGA) system and thanks to this, we can analyze signals much faster.

2. FMICW radar description

The FMICW system works as a combination of the pulse radar and FMCW (Frequency Modulated Continuous Wave) radar. FMCW systems are described for example in [2, 3]. The principle of the FMICW radar is described in [4], but this system was developed for the measurement of the ionosphere, and the pulse part of the system does not have a big influence on the signals. The block diagram of the FMICW radar is shown in figure 1. The signal is generated by the sweeping generator and connected to the output amplifier. Switching of the system is realized via PIN diodes; one is for the connection to the transmitting amplifier and second is for the receiver. A detailed description of the radar PCDR 35 blocks was realized in [5]. The radar PCDR 35 was developed for the measurement in short distances (less than 10 km) and we must include the influence of pulses in this case, as shown in figure 2. Because we have different lengths of the reflected signals, we must do corrections of the power. This was described for example in [5]. The distance in FMICW radars is calculated by equation (1).

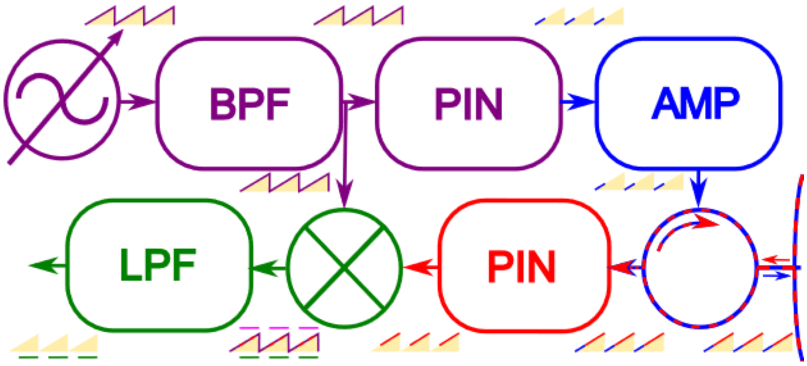

Figure 1. Block diagram of FMICW radar. [1]

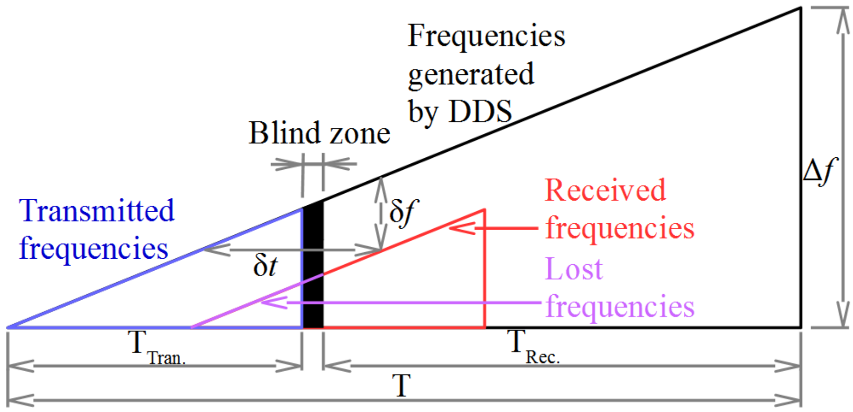

Figure 2. Timing diagram of FMICW radar. [6]

$$R = \frac{c_0 \cdot \delta f \cdot T}{2 \cdot \Delta f} + e(\delta f), \quad (1)$$

where c_0 is the light speed, δf is the output frequency, T is the measurement period, Δf is the sweep range and $e(\delta f)$ is the measurement error.

Frequencies are obtained during the primary signal processing of the radar. The frequency analysis can be realized by the non-parametric, or parametric methods [7]. The parametric AR model is described for example in [8]. More parametric and non-parametric methods are described in [9]. 2D FFT can be used for the estimation of the target velocities, this algorithm is described in [10, 11]. The principle is shown in figure 3. Measured data are sorted in a matrix, after every measurement is transformed by the Fast Fourier Transform (FFT). The results from this process are the range profiles for the times of measurement. The next step is applying the FFT on the time dimension. This transformation changes the time dimensions to Doppler shift dimensions. The Doppler shift resolution is defined by equation (2). The testing measurements of the radar PCDR 35 and sensitivity analysis were presented in [12]. Examples of four cases of 2D spectra are shown in figure 4, where PSD is power spectral density. The clutter elimination is described in [13], where the static clutter detection, and the meteorological clutter is canceled by the Airborne Moving Target Identification (AMTI) filter. It represents selected velocities for all distances in the case of the signal 2D spectrum.

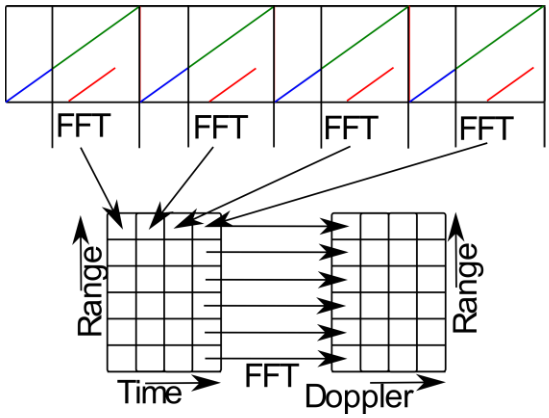

Figure 3. Principle of the 2D FFT from the signal. [10]

$$\delta D = \frac{1}{T \cdot N}, \tag{2}$$

where T is the time between measurements (periods), and N represents the number of measurements.

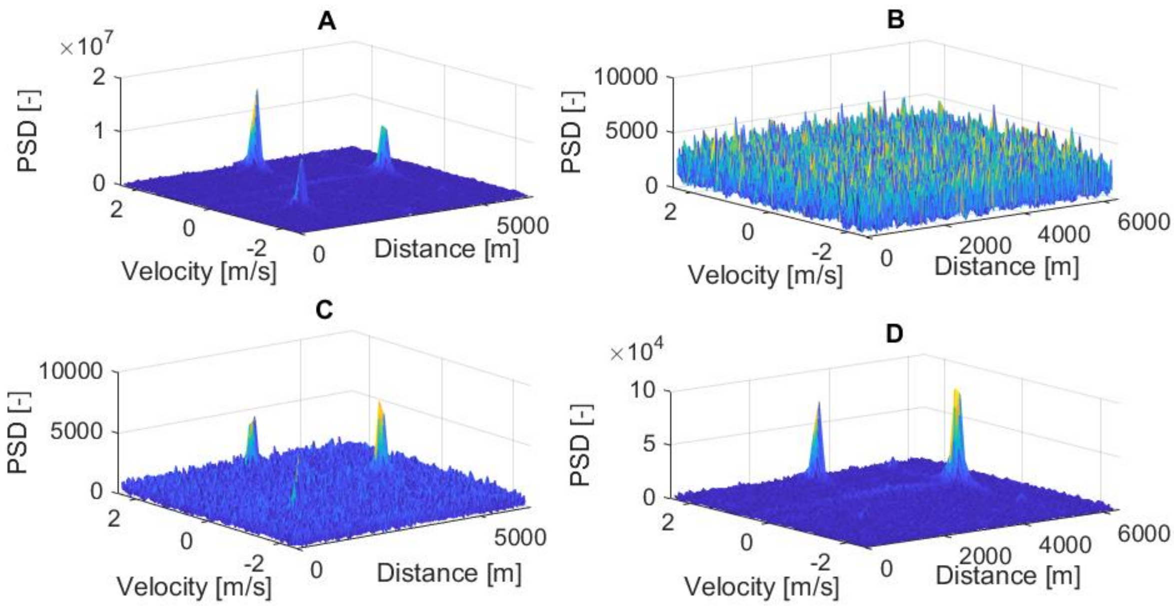

Figure 4. 2D spectra of the four situation cases: (**a – left top**) 2D spectrum of the signals with three strong echoes, (**b – right top**) 2D spectrum of the signals with noise only, (**c – left bottom**) 2D spectrum of the signal with three weak echoes (**d – right bottom**) 2D spectrum of the signal contains one weak and two strong echoes.

3. Neural networks

The neuron model is shown in figure 5. We can see, that the neuron has N inputs ($x_1, \dots x_N$) and one output (y). Every input is multiplied by the input weight ($w_1, \dots w_N$), this value is changing during the learning process. The output has application function Y, which decides the output value. Activation functions can be different and usually are nonlinear (signum function, limited linear function, standard logistic function, hyperbolic tangents). Examples of these functions are described in [14]. The neuron bias is represented by the input b, and this signal is also multiplied by weight w_0. If we look at the neuron model, we can see that neuron is based on principles of digital filters. The mathematical description of one neuron is in the equation (3). The benefit of this model is the easy application on the FPGA with faster processing speed. Applications of the neurons and neural networks on the FPGA are described in [15].

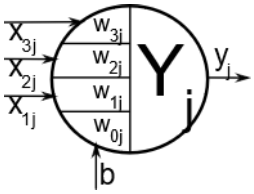

Figure 5. Formal neuron with bias. [16]

$$y_j = Y_j\left(b \cdot w_{0j} + \sum_{i=1}^{N}\left\{w_{ij} \cdot x_{ij}\right\}\right), \tag{3}$$

where Y_j is the application function of a neuron, x_{ij} represents i-th neuron input, w_{ij} represents the weight for the input (synapse), *b* represents bias and N represents number of inputs.

The Hebbian learning is based on the increasing and decreasing connections between two neurons (changing of the input weights). The increasing and decreasing is, in turn, controlled by the inputs and outputs. Training signals are used in the learning sequence. Results are set by the user and if an input and an output are related, then the connection strengthens (the weight is increased), but if they are not related, the connection weakens (the weight is decreased). This style is called learning with a teacher. If we use all combinations, the system remembers all solutions.

The neural network is composed from the basic neurons. Neurons can be sorted into more layers. These layers are sorted into groups known as the input layer, hidden layers and output layer. An example of the neuron network in topology (3-3-4-2) is shown in figure 6. This is the neural network with more layers. The connection between neurons can skip to any layer or can be realized as the back loop.

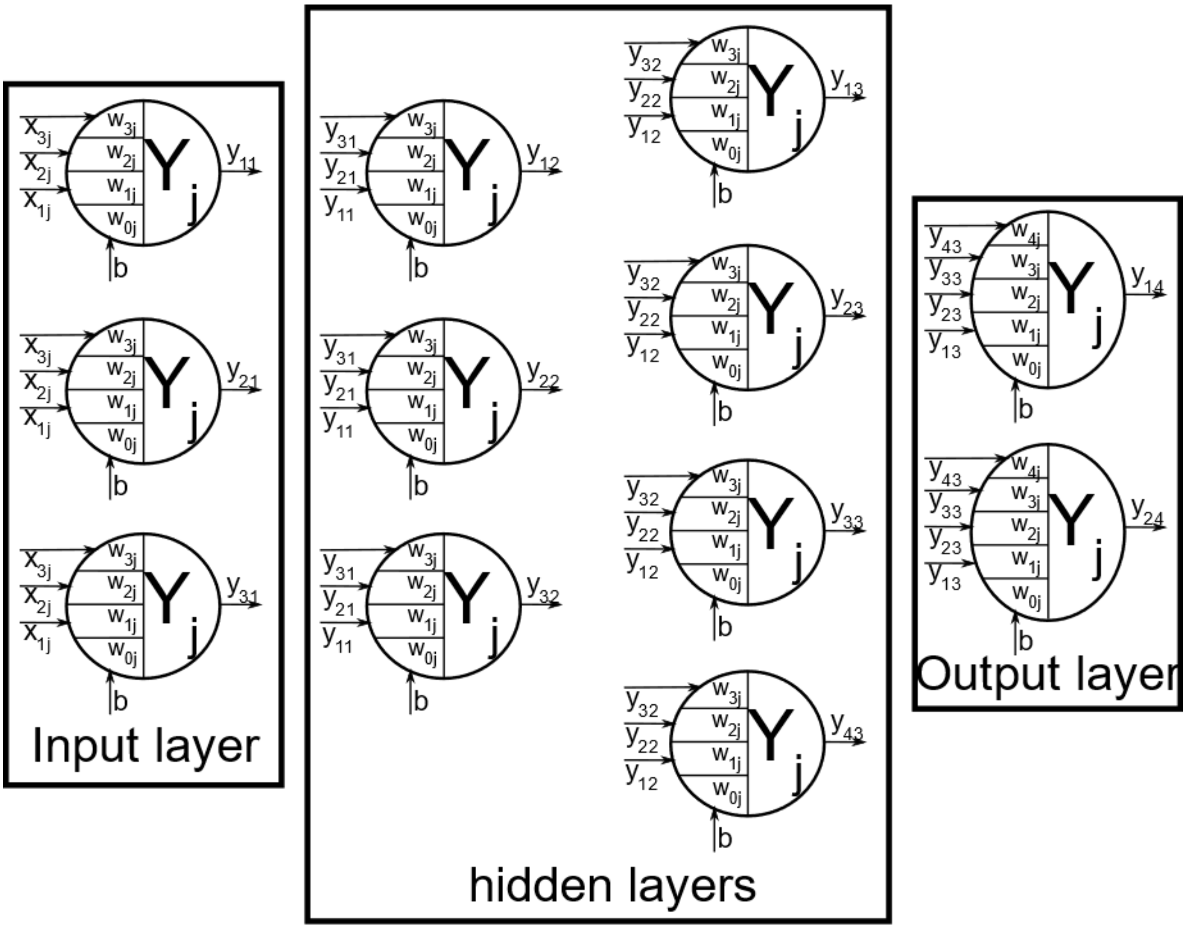

Figure 6. An example of a multilayer neural network architecture (3-3-4-2).

4. Application of the neural networks for the radar signal processing

The neural network for our algorithm is composed from three, or four layers (optional). Every layer has a specific function. The input layer is for the 2D spectrum filtration and the thresholding. The second layer is for the target unification, when any target is split. The third layer can be transformation of the target area to the rectangular shape (this layer is optional). The last layer is the target board line detection.

4.1. Filtration and thresholding

Neurons in this layer transform the 2D spectrum (the original spectrum is represented by the spectral components sizes) to a binary matrix, where 1 represents the positive detection and -1 represents the negative detection in the tested cell. In this layer there are used two types of neurons, one type is used for the filtration and the second type is used for the threshold value estimation. This value is used in the filtration neurons for the activation function. From this, it is obvious that numbers of neurons for the filtration and the thresholding in the input layer are functions of the 2D spectrum size. The model of this neuron is shown in figure 7. Input signals are in the input vector $\vec{X_j}$. This vector has n elements and these elements are obtained from the distance profile. Synapses are saved in the vector $\vec{W_j}$, and activation function Y_j is the signum function. The neuron's model for the threshold value estimation is shown in figure 8. The activation function Y_j of this neuron is linear, the input is the matrix of the 2D spectrum from the radar measured data. This neuron has common weight for all inputs, because all inputs have the same priority. This neuron definition is shown in equation (4), where TV represents the threshold value (5).

Teaching the neurons is described in the algorithm in figure 9. The 2D spectrum is manually processed and transformed to the mask, where 1 represents positive detection (target) and -1 represents negative detection (noise). This mask is used for the neural network Hebbian learning. As described in figure 7, the input vector is obtained from the 2D spectrum. The used vector has 21 elements and all inputs have the same Doppler shifts. Weights are increased in the case of positive detection 1 and decreased in the case of negative detection -1. The first threshold value is estimated according to equation (5). At the start of the algorithm, the weight is chosen as 1. The next step is 2D spectrum processing by using the neural network and the obtained result is compared with the mask. If results match, the algorithm is finished. If the results do not match, the weight is modified according to the results. If the neural network does not detect any targets, the weight is divided by 1.4 and if the neural network generates false alerts, the signal is multiplied by 1.4 (this value was chosen experimentally). The next step is the threshold value recalculation from the new weight and the 2D spectrum is processed again.

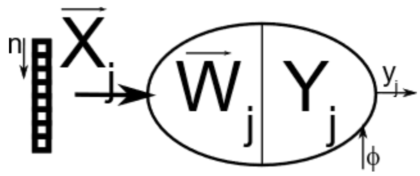

Figure 7. Neuron for the filtration and thresholding.

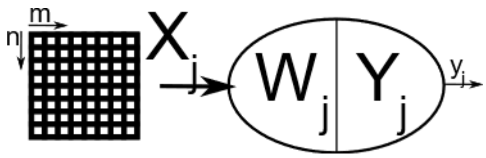

Figure 8. Neuron for the threshold level estimation.

$$y_j = Y_j\left(w_j \cdot \sum_{i=1}^{N} \left\{ x_{ij} \right\}; TV \right), \tag{4}$$

$$TV = w_j \cdot \sum_{n=1}^{N} \sum_{m=1}^{M} \left\{ 2DFFT_{nm} \right\}, \tag{5}$$

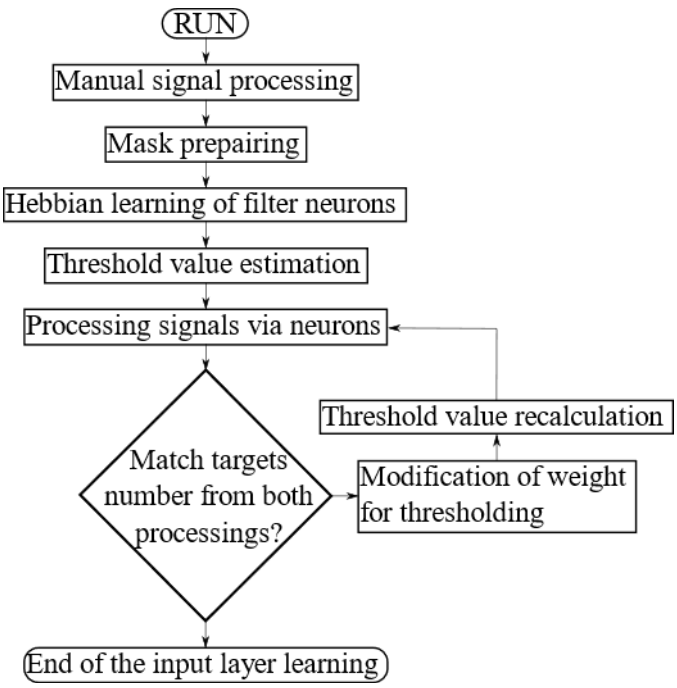

Figure 9. Algorithm for the learning of the input layer of the neural network for the radar signal processing.

After the input network learning, we will obtain weight parameters for the filtration neurons. These weights are shown in the graph in figure 10, and these parameters will be set by the equation (3). Weights are saved in vector $\overrightarrow{W_j}$ from figure 7. The weight for the neuron threshold value estimation was estimated from the learning as $113.3817189 \cdot 10^{-6}$. Test of the layer application is shown in figure 11. The 2D spectrum with three targets was used for input to this layer.

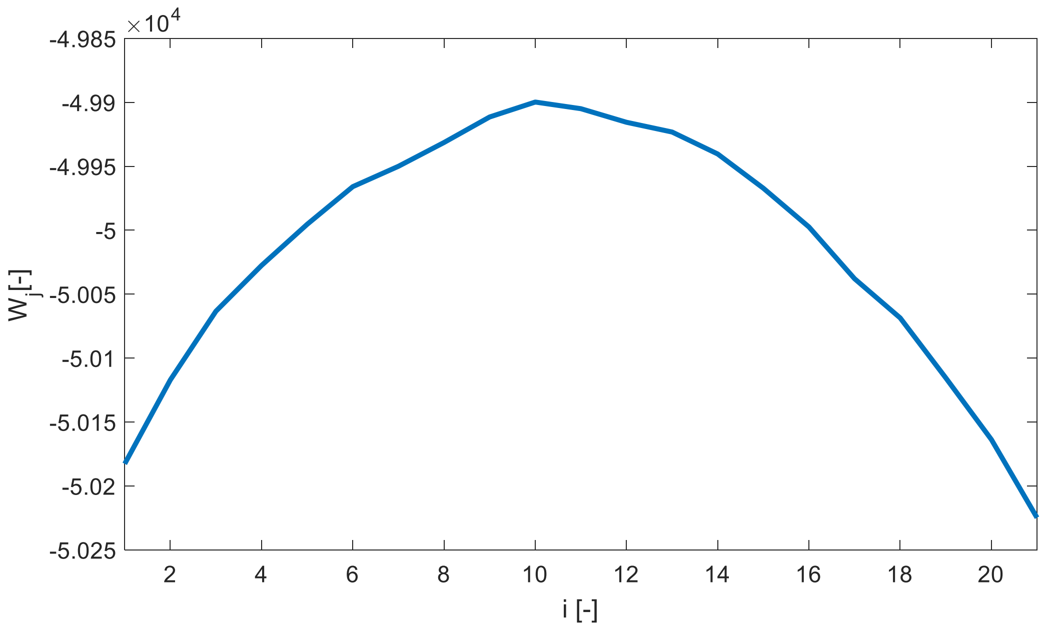

Figure 10. Synapses weight vector parameters for filtration of the 2D spectra. (20 inputs)

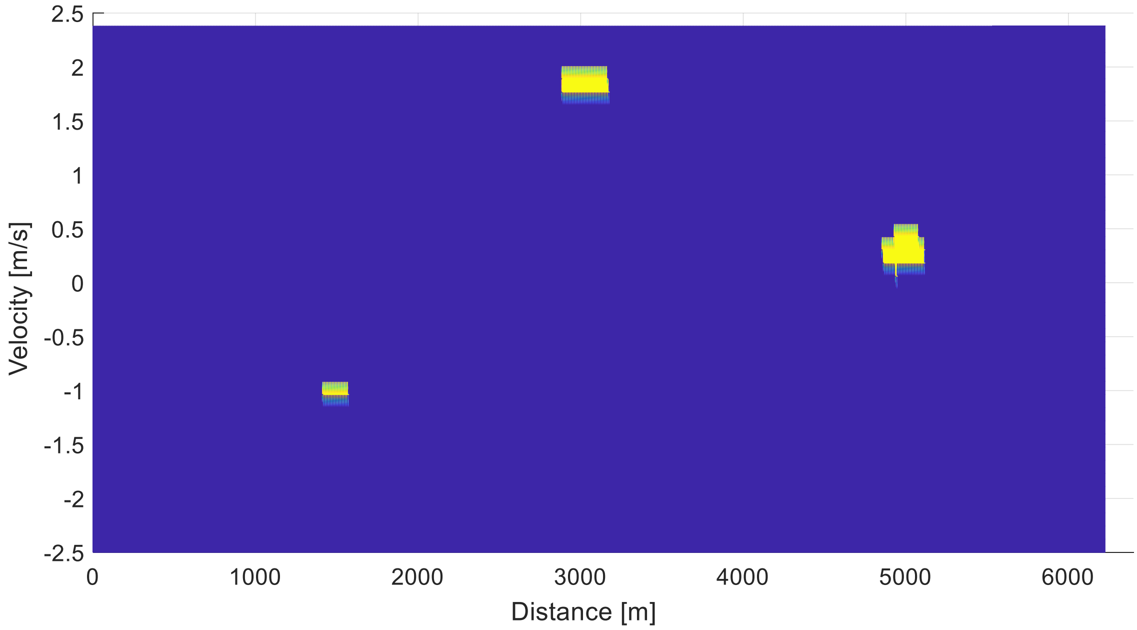

Figure 11. Test of the neurons in the input layer (filtration and thresholding of the 2D spectrum): 2D spectrum with 2 strong echoes and one weak echo is used for input in this figure.

4.2. Unification of the targets

The target can be split during the thresholding process, and we must make target unification in this case. For this we are using neural sub networks. One of these neural networks is shown in figure 12. Neurons in the first layer are described by the equation (6) and the output layer is described by the equation (7). For all neurons the activation function signum is used. Input signals are two vectors of five elements. The first vector has elements placed before the tested element in the range dimension, and the second input vector has elements following the tested element in the range dimension. The neuron in the second layer is the OR function with three inputs, where one input is the cell state before the unification process and the next two inputs are outputs from the first layer neurons. During the Hebbian learning we used a negative combination twice for the better setting. Test of this neural sub layer is shown in figure 13, where we can see that both splits were removed, and the target is again united.

$$y_j = Y_j\left(\sum_{i=1}^{5}\{x_{ij}\} + 4\right), \tag{6}$$

$$y_j = Y_j\left(3 \cdot \sum_{i=1}^{2}\{x_{ij}\} + 5\right), \tag{7}$$

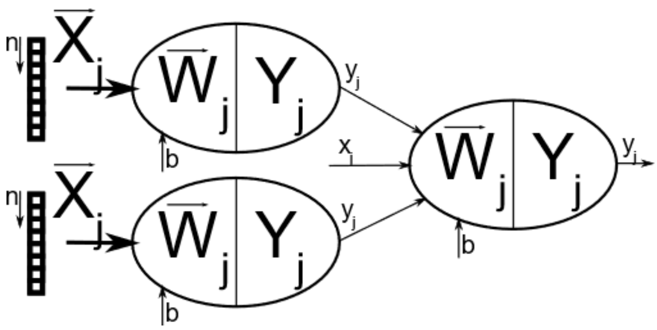

Figure 12. Neural network for unification of the targets.

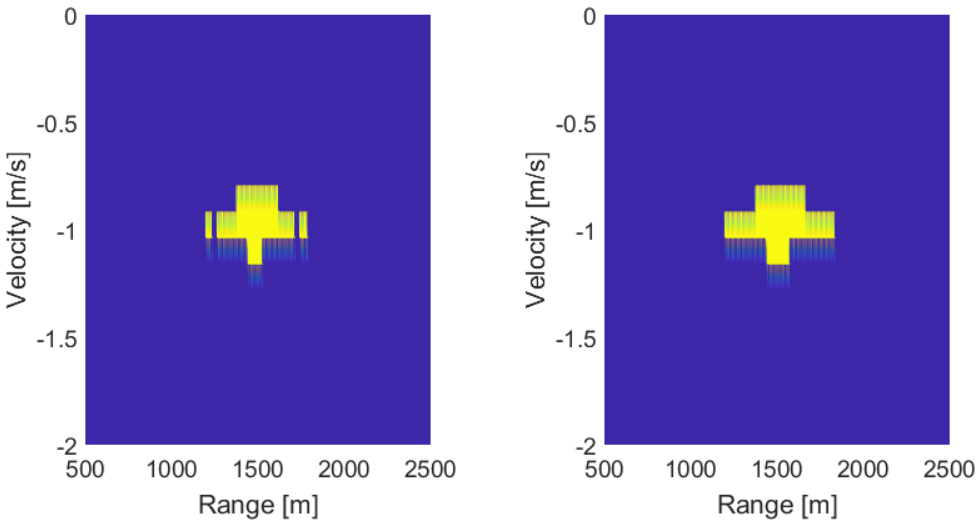

Figure 13. Test of the neural sub network for unification of the targets: (**a**) Target is split to three parts (**b**) Target was restore via neural sub network.

4.3. Detection of the target board lines

The neurons, with nine inputs in this layer are organized in the 3x3 matrix of elements. Inputs are connected from layer 1 (inputs can be 1, or -1). The activation function is the signum function. The model of this neuron is similar to the neuron from figure 8, only another activation function and different bias is used. This layer is used for the definition of the target board lines. The neuron tests if the middle element is -1 and around are any elements with 1. If these conditions are not successful, the output is -1. If conditions are successful, the output is 1. After the Hebbian learning we obtained an equation, which cannot be used, because in the case when all inputs are -1 the output is wrong. The layer generated the positive board line of the target and the first condition was not successful. Then we used this case more times for the Hebbian learning and we trained this neural network until the neural network started to analyze this case correctly. We obtained equation (8) from this training. Input x_{9j} represents the middle element in the matrix 3x3, the weight for this element is -8, the weight for the other elements is 1. This is because these elements have same priority and the value -0 is for bias (for the Hebbian learning we were using 8 cases for positive detection and 8 cases for negative detection). Application of this layer is shown in figure 14. Input for testing of this layer is the output from the thresholding layer, which was shown in figure 11 after unification of the targets. We can see, that the 2D spectrum was analyzed correctly.

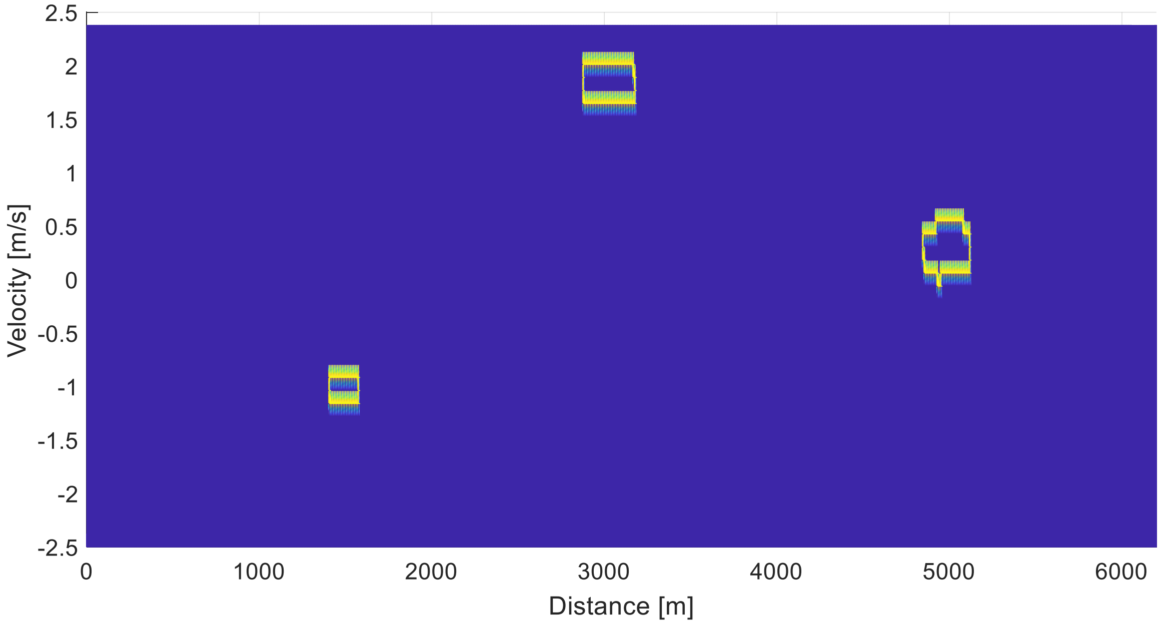

Figure 14. Board lines of the targets detected in the output from the unification neural sub network.

$$y_j = Y_j\left(-8 \cdot w_{9j} + \sum_{i=1}^{8}\left\{x_{ij}\right\} - 0\right), \tag{8}$$

4.4. Marking of the targets position

This part is used for the target marking in the signal. The processing is composed from the two steps. The first step is transformation of the target area and the settings of the rectangular shape. The second step is the board lines detection in the rectangular shape.

The neural sub network for the transformation of the area is shown in figure 15. The first layer is from neurons described by equation (9). This is the OR function which depends on the vector length. For the Doppler shift dimension a vector where *n* is 21 elements is used, and for the range dimension a vector where *m* is 81 elements is used. The tested element is in the vector middle in both cases. The equation for the OR neurons which are used for the transformation of the area is described by (9). It was derived from the Hebbian learning for more lengths of the input vectors. The output layer is the AND neuron, which is described by equation (10). The signum function is used for both neuron types in this neural sub network. For the second step the same layer is used as in the case when we detected the board lines of the targets after the thresholding. Output from the application of this layer is shown in figure 16.

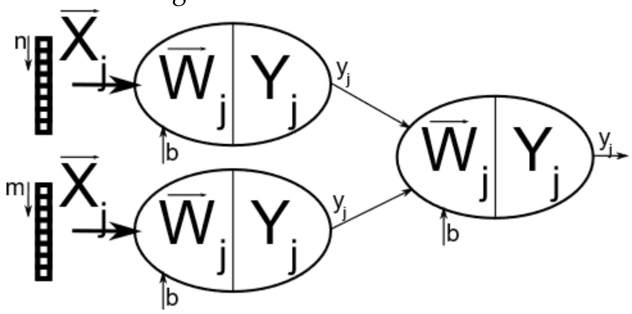

Figure 15. Neural sub network for transforming of the area to rectangular shape.

$$y_j = Y_j\left(\sum_{i=1}^{I}\{x_{ij}\} + I - 1\right), \tag{9}$$

$$y_j = Y_j\left(\sum_{i=1}^{2}\{x_{ij}\} - 1\right), \tag{10}$$

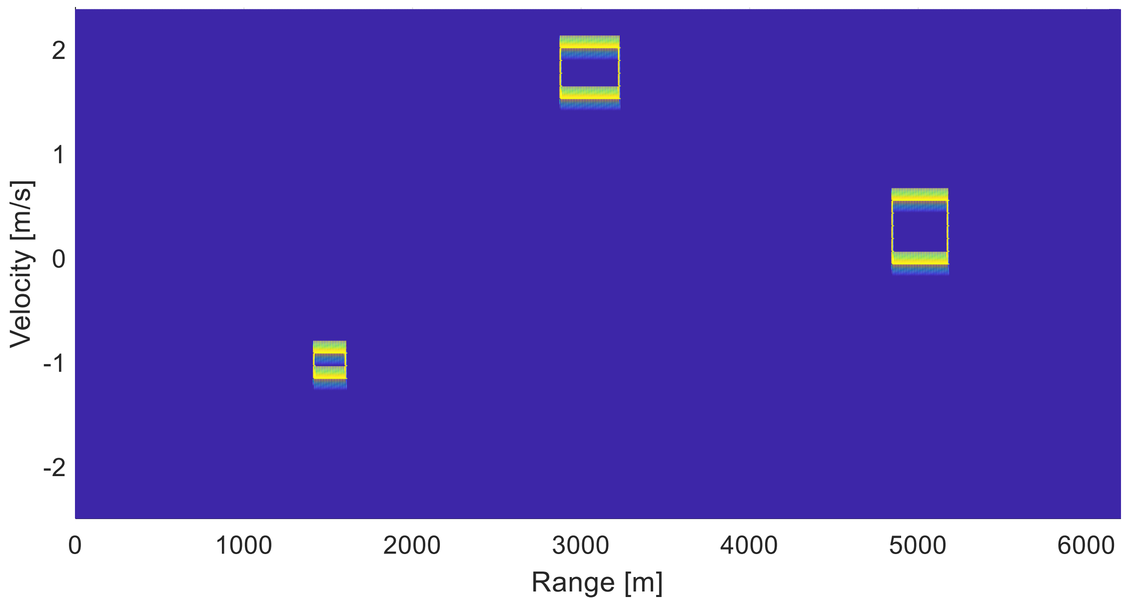

Figure 16. Board lines of the targets (rectangular shape) in the output from the proposed system. One weak and two strong echoes were in the input.

5. Description of the proposed neural network

Proposed parts of the neural network were connected to the final system. This final neural network is shown in figure 17. Neurons placed in the rectangular shape represents this neuron matrix, and outputs from the neural network are realized by the marking blocks. The input matrix size is 665x41 elements (665 elements represents the samples of one measurement, it is the function of the sampling frequency and radar range. 41 samples represent the realizations, less samples are not enough for Doppler measurements, more samples make long response time of the system). The output signal is the matrix with the same dimensions. The processed signals example is in figure 18. Targets in the 2D spectrum were marked by the red lines. The case in the right top position is only with noise and no target is marked.

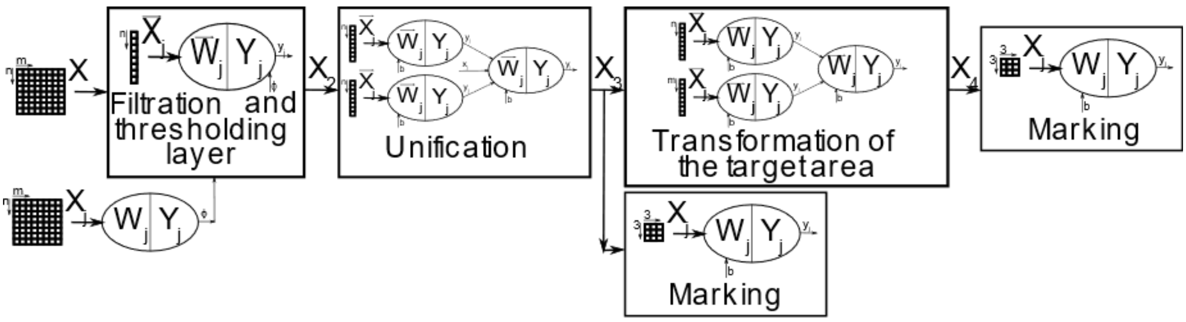

Figure 17. Neural network proposed for the FMICW radar signal processing.

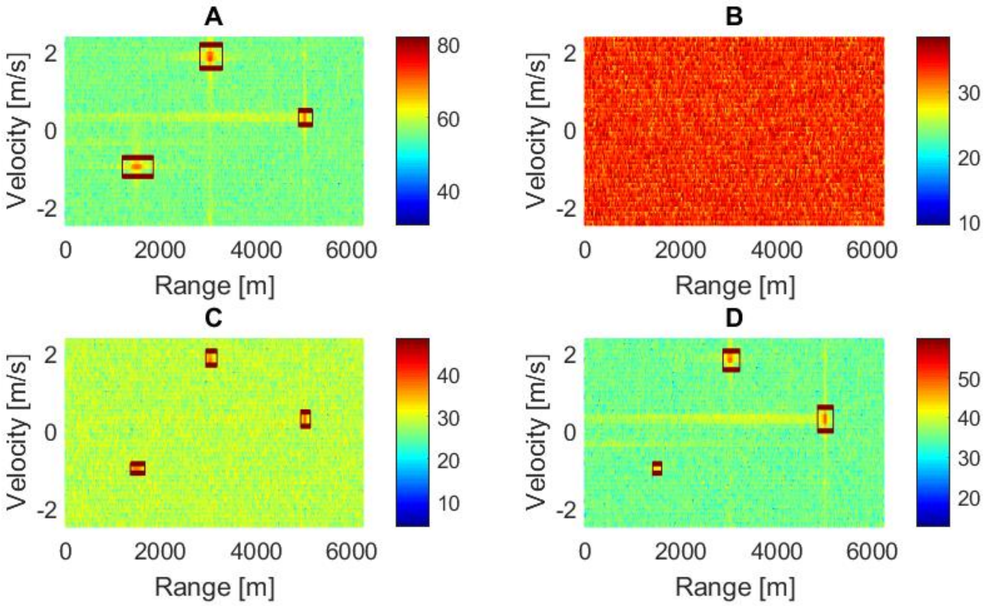

Figure 18. Marking of the targets in 2D spectrum: (**a**) processing of the signal with three strong echoes, (**b**) processing of the signal without echoes, (**c**) processing of the signal with three weak echoes and (**d**) processing of the signal with one weak and two strong echoes.

6. The algorithm test on the training sequence data and discussion

The proposed neural network was tested on the simulated data. These data were created by our simulator, which is based on the real system PCDR 35. Real measurement data and comparison with simulated data is shown in [17]. In the results there is a difference, where the measured data contains a noise level dependent on the distance, and this is not reflected in the simulator. This is caused by the wrong impedance in the radar output, but this was corrected in cooperation with the BTV Klimkovice company.

Results of the proposed neural network are presented in table 1. We tested the neural network in four cases: three strong targets, noise, three weak targets, and one weak with two strong targets. For every tested case 100 sets of the simulated data were used. From the table 1 we can see very good results. Table 2 is added for comparison, where the unification layer in the neural network was not used. We can see, that the results are unacceptable. For the test, the same data were used, which we used for the validation of our previous algorithm, which is published in [1]. This algorithm results are in table 3. From comparison of table 1 and table 3 we can see, that the neural network results are much better than previous algorithm in two cases, and the same results are in the next two cases.

Table 1. Results of the proposed neural network.

Cases	Totally right	Lost targets	False alerts
3 strong targets	97	0	3
Noise	100	-	0
3 weak targets	96	4	4
1 weak and 2 strong targets	94	0	6

Table 2. Results of the proposed neural network without unification of the targets.

Cases	Totally right	Lost targets	False alerts
3 strong targets	33	0	77
Noise	100	-	0
3 weak targets	95	4	5

1 weak and 2 strong targets	94	0	6

Table 3. Validation of the previous algorithm published in [1].

Cases	Number of targets	Distances	Doppler shifts
3 strong targets	94	94	94
Noise	100	-	-
3 weak targets	96	96	96
1 weak and 2 strong targets	84	84	84

After revision of the wrongly processed cases, we found that false alerts in the case of three strong targets were caused by the using of short vectors for the target unification. We tried to extend these vectors, and we used equation (11) for the input to this neural sub network. After the test, we obtained results from table 4. We can see that the results are better for this target type, but from the process we can estimated that in any case the distance between main target and side lobe of this target can be bigger, then we will obtain wrong results again. Much longer vector length we also cannot use, because we can do unification of more targets to the one target. We can see, that results are good, but it can happen that processing can contain errors, but these mistakes are rare. After checking the wrongly processed cases for the last situation, we can see, that the distance between the original target position and the detected side lobe is very big and cannot be removed by the neural sub network for unification. One example is shown in figure 19. When we checked the lost targets, we observed, that targets were lost, because the simulated target position was only the one point from the area. This marking causes the target loss, because the area is very small in this case. If we included board lines to the target area, the target is marked correctly. Thanks to this we can obtain results in table 5. Targets are detected correctly, only they are not in the middle of the marked area. From this we can see, that the used neural network has much better results than previous algorithms.

$$y_j = Y_j\left(\sum_{i=1}^{7}\left\{x_{ij}\right\}+6\right), \tag{11}$$

Table 4. Modification of the neural network for improving the strong target detections.

Cases	Totally right	Lost targets	False alerts
3 strong targets	100	0	0
Noise	100	-	0
3 weak targets	96	4	4
1 weak and 2 strong targets	94	0	6

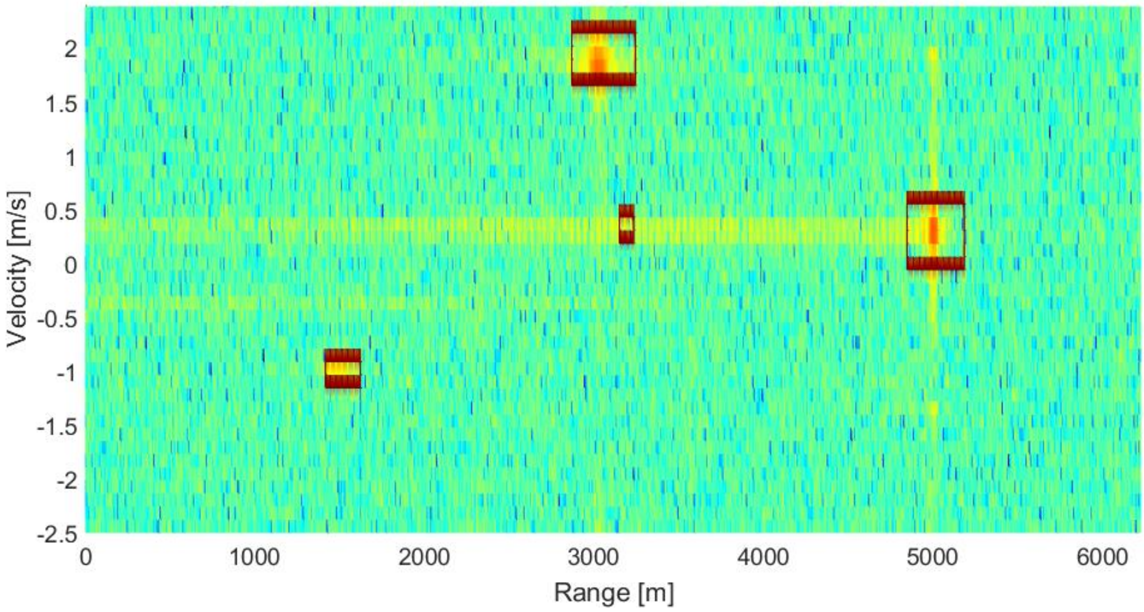

Figure 19. False alert in the processed 2D spectrum.

Table 4. Modification of the neural network for improving the weak targets detection.

Cases	Totally right	Lost targets	False alerts
3 strong targets	100	0	0
Noise	100	-	0
3 weak targets	100	0	0
1 weak and 2 strong targets	94	0	6

An example of another work in this field can be found in [18]. These authors measure position and Doppler shifts of the targets in this research. They are using for detection the CFAR (Constant false alarm rate) method and tracking of the target. Setting of their algorithm and the evaluation study are not published. Another approach is used in [19], where other authors use another type of the algorithm for processing of the velocity measurement. The second group of authors are using target route tracking for velocity estimation of the targets in this research, but they did not describe their algorithm exactly. From results which they published, we can see, that they also have problems with false alerts and losing of the targets. Efficiency of this algorithm was not published. The better research on this topic is described in [20]. Here, authors describe their algorithm and they made the evaluation of the success. Their algorithm has, according to tests, very good results. These results are comparable with our algorithm except for Doppler shifts or velocities. This is because they did not include these measurements in their research.

7. Conclusion

In this paper we described the neural network for the FMICW radar signal processing. The neural network can be easily used for implementation on the FPGA with the speed processing benefit. Use of the neural network improved the threshold level estimation. Before, we used median, and it was very time consuming in comparison with the neural network, where the sum of elements is multiplied by the weight. This way is faster for the PC processing in comparison with the previous way.

The two outputs are from the neural network; one for the precision marking of the targets and one for the marking by the rectangular shape. The first output is better for extremely big target detection (the rain cell). The second one is good for the point targets. Because we tested this neural network with the same data as our older algorithm, we can easily compare these two approaches for radar signal processing. From the results we can see, that the using of this neural network is much better than our previous algorithm, which we used before. In the case of one weak and two strong targets the improvement is around 10 %.

Author Contributions: LR created the main idea, developed the algorithm and prepared the main part of the article, TNN edited the text to template form and helped with programming Hebbian learning of the layers. PCH worked at the image processing, he is coauthor of the main idea (board lines detection, symmetry detection – it was not finally used) and helped with completion of the text and did language revision. LB taught us processing of the signal via neural networks and prepared theoretical part about neural networks and TTP was our supervisor during the work on the described project. He helped us via discussion of the problem and helped with completion of the text and did language revision.

Funding: This paper was supported by the internal student grant university of Pardubice: 60120/20/SG690021.

Acknowledgments: Thanks to the MAREW 2019 conference unknown member for his suggestion to use the neural network for our system and thanks to Charles Hooper for final language revision.

References

1. [1] REJFEK, L., FISER, O., CHMELAR, P., PITAS, K., BEZOUSEK, P., PHUONG, T.T., DONG, S.T.CH., "Automatic Analysis of the Signals from the FMICW Radars," 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, Czech Republic, 2019, pp. 1-4. doi: 10.1109/RADIOELEK.2019.8733449
2. [2] NEEMAT, S., KRASNOV, O. and YAROVOY, A., 2019. An interference mitigation technique for FMCW radar using beat-frequencies interpolation in the STFT domain. IEEE Transactions on Microwave Theory and Techniques, 67(3), pp. 1207-1220.
3. [3] BEZOUSEK, P., HAJEK, M. and POLA, M., 2010. Effects of signal distortion in a FMCW radar on range resolution, Proceedings of 15th Conference Microwave Techniques, COMITE 2010 2010, pp. 113-116.
4. [4] NOZAKI, K., FMCW Ionosonde for the SEALION project, Journal of the National Institute of Information and Communications Technology, vol. 56, no. 1-4, pp. 287-298.
5. [5] REJFEK, L., FISER, O., MATOUSEK, D., BERAN, L. and CHMELAR, P., 2017. Correction of received power for Doppler measurements by FMICW radars, Proceedings Elmar - International Symposium Electronics in Marine 2017, pp. 107-110.
6. [6] REJFEK, L., BEZOUSEK, P., FISER, O. and BRAZDA, V., 2014. FMICW radar simulator at the frequency 35.4 GHz, 2014 24th International Conference Radioelektronika, RADIOELEKTRONIKA 2014 - Proceedings 2014.
7. [7] SMEKAL, Z., "Číslicové zpracování signálů," Vysoké Učení Technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009, skriptum, p. 208
8. [8] CHMELAROVA, N. and TYKHONOV, V.A., 2016. Composite vector stochastic processes model in the task of signals' recognition, 2016 26th International Conference Radioelektronika, RADIOELEKTRONIKA 2016 2016, pp. 203-206.
9. [9] STOICA, P. and MOSSES, R., Introduction to Spectral Analysis, Upper Saddle River, NJ: Prentice Hall, 1997.
10. [10] OTTO, T., Presentation "Principle of FMCW radars,", Published on Jul 26, 2012 at https://www.slideshare.net/tobiasotto/principle-of-fmcw-radars
11. [11] SEDIONO, W., 2013. Method of measuring Doppler shift of moving targets using FMCW maritime radar, Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2013 2013, pp. 378-381.
12. [12] REJFEK, L., FISER, O., MATOUSEK, D., BERAN, L. and CHMELAR, P., 2017. Sensitivity analysis of PCDR35 radar, Proceedings Elmar - International Symposium Electronics in Marine 2017, pp. 111-114.
13. [13] REZNICEK, M., BEZOUSEK, P. and ZALABSKY, T. AMTI filter design for radar with variable pulse repetition period, Journal of Electrical Engineering, 2016, vol. 67, no. 2, s. 131-136. ISSN: 1335-3632.
14. [14] HIKAWA, H., 2003. A Digital Hardware Pulse-Mode Neuron With Piecewise Linear activation Function. IEEE Transactions on Neural Networks, 14(5), pp. 1028-1037.

15. [15] LIU, Y., LIU, S., WANG, Y., LOMBARDI, F. and HAN, J., 2018. A Stochastic Computational Multi-Layer Perceptron with Backward Propagation. IEEE Transactions on Computers, 67(9), pp. 1273-1286.
16. [16] VOLNA, E. Scriptum “Neuronové sítě 1.” Ostrava: Ostravská univerzita, 2002.
17. [17] MANDLIK, M. and BRAZDA, V., 2015. FMICW radar simulator, Proceedings of 25th International Conference Radioelektronika, RADIOELEKTRONIKA 2015 2015, pp. 317-320.
18. [18] KREJCI, T. and MANDLIK, M., 2017. Close vehicle warning for bicyclists based on FMCW radar, 2017 27th International Conference Radioelektronika, RADIOELEKTRONIKA 2017, 2017.
19. [19] HONG, D.-. and YANG, C.-., 2012. Algorithm design for detection and tracking of multiple targets using FMCW radar, Program Book - OCEANS 2012 MTS/IEEE Yeosu: The Living Ocean and Coast - Diversity of Resources and Sustainable Activities 2012.
20. [20] YULIAN, D., HIDAYAT, R., NUGROHO, H.A., LESTARI, A.A. and PRASAJA, F., 2018. Automated ship detection with image enhancement and feature extraction in FMCW marine radars, Proceeding - 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications, ICRAMET 2017 2018, pp. 58-63.