

## **APPLICATION OF HYBRID NEURAL NETWORK SYSTEM IN IMAGE PROCESSING**

**Łukasz RYKAŁA**

lukasz.rykala@wat.edu.pl

<https://orcid.org/0000-0002-2301-3280>

Military University of Technology  
Faculty of Mechanical Engineering

***Abstract:** The article describes the use of feedforward two-layer neural network and recursive neural network with feedback, i.e. Hopfield network within a system capable of performing associative memory functions. It has been shown that these structures can be used successfully alone, however, using them together in series in the listed above order significantly improves system performance.*

***Keywords:** neural networks, Hopfield network, image processing, artificial intelligence*

### **INTRODUCTION**

Feedforward neural networks are the basic structure of artificial neural networks in which there is no feedback. The way these types of networks operate is determined by their name. Networks with one-way connections are successfully used, among others in robotics, transport, control, and also widely understood image processing (Korbicz, Obuchowicz, Uciński, 1994; Muszyńska, 2015; Rykała, 2018; Kijek, Brzezinski, Rykała, Zelkowski, 2018).

The Hopfield network is a recursive neural network with feedback, i.e. a neural network in which there is feedback between layers. Due to its function, the Hopfield network is often called auto-associative or auto-association memory. It enables, among others restoring images from their fragments, solving optimization tasks, etc (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006; Tadeusiewicz, Szaleniec, 2015).

In turn, associative memory is linked with one of the basic functions of the human brain and in this context consists of restoring all information based on associations (from the available fragment of memory). By concentrating, people can understand very indistinct speech or read illegible writing. The main task of the mentioned network is to remember a set of “patterns” in such a way that the system, during the presentation of new inputs, can generate a response that will correspond to one of the previously remembered “patterns” (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006).

In the Hopfield network, all output signals are treated as network inputs and all input signals carry feedback signals to all neurons. In the mentioned system there is no feedback of the neuron with its input signal. However, it is possible for the output signal from a given neuron to influence its value in the future due to the feedback of additional intermediary neurons. The action of additional neurons stabilizes such a signal (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006).

The described network consists of  $N$  neurons, of which at the discrete-time  $k$  only one active processing neuron  $p$  is active, which is selected randomly from all neurons with the same probability. Changes in the state of the network occur only at discrete moments (Korbicz, Obuchowicz, Uciński, 1994).

The principle of the processing neuron can be described by the following dependencies:

$$\varphi_p(k) = \sum_{j=1}^N \omega_{pj} x_j(k) - \varphi_p \quad (1)$$

$$x_p(k+1) = \begin{cases} 1, & \varphi(k) > 0 \\ x_p(k), & \varphi(k) = 0 \\ 0, & \varphi(k) < 0 \end{cases} \quad (2)$$

Where (Korbicz, Obuchowicz, Uciński, 1994):

- $x_{pj}(k)$  – output of the  $j$ th processing element at the time  $k$ ,
- $w_{pj}$  – the weight of the connection between the  $j$ th output signal and the  $p$ -element input signal,
- $\varphi_p(k)$  – a threshold value of the neuron.

The Hopfield network can be presented in many ways, most often it is taken as a system in the form of direct coupling of the output with the input (Fig. 1).

Two modes of operation can be distinguished in the operation of the Hopfield network (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006; Tadeusiewicz, Szaleniec, 2015):

- learning mode,
- playback mode.

The process of learning the network consists of such a selection of weights of network connections that the presented “pattern” becomes one of the so-called attractors (stable states). In order to modify the weights in the Hopfield network, the Hebb rule is used. The usage of the learning mode causes that the weight values are the average values of all “learning patterns”, i.e. information about a given sample is included in all network weights. This is the basic advantage of the Hopfield network, which is a factor affecting the network’s resistance to damage. Changing the weights of connections or damaging one of the neurons has a slight effect on the network. For this reason, the Hopfield network has found wide applications in systems where its damage would have serious consequences, e.g. spacecraft control systems, etc. (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006; Tadeusiewicz, Szaleniec, 2015).

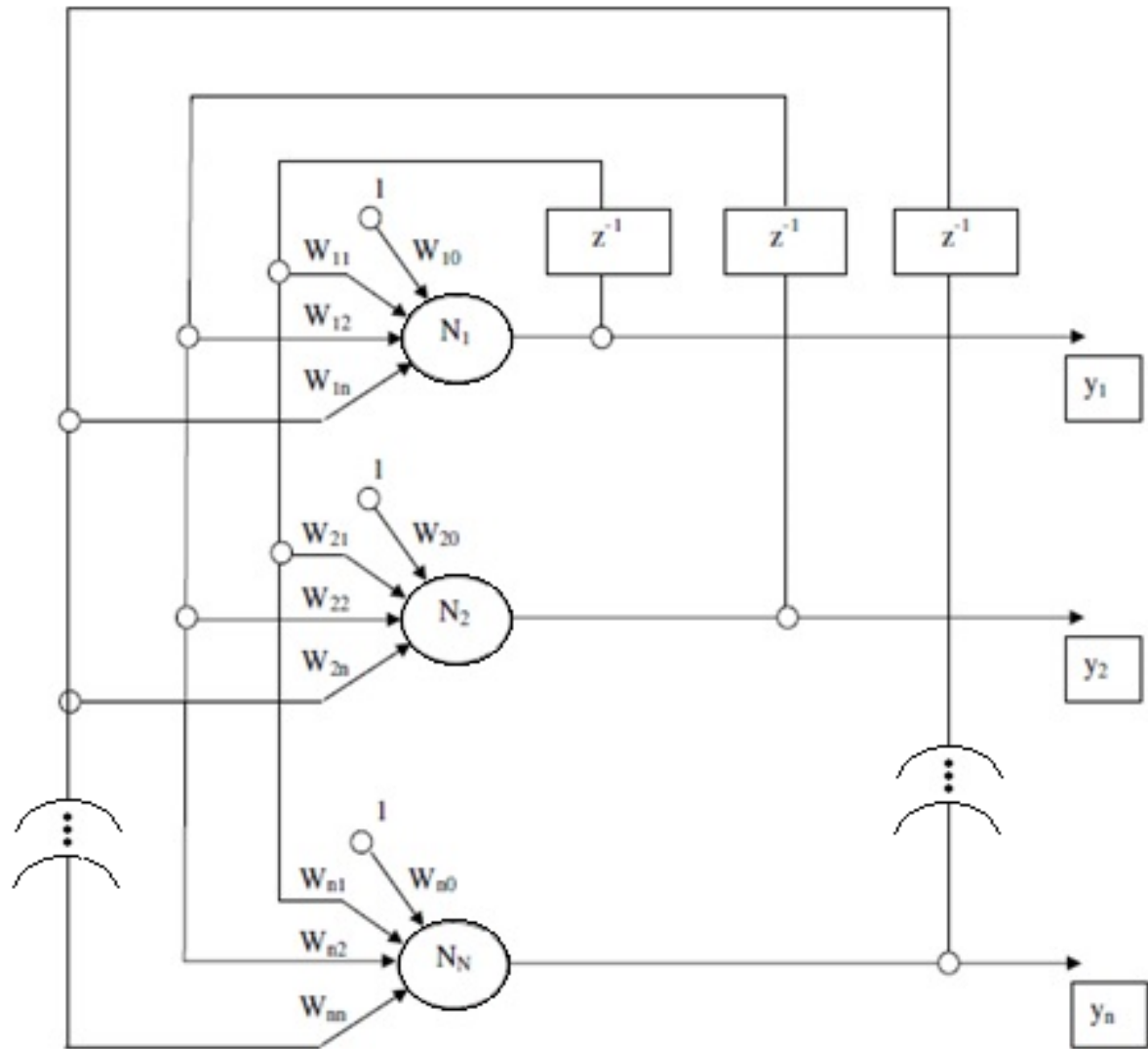


Fig. 1. Hopfield network diagram

Source: Drawn by the authors.

The playback process begins at the initial moment of time  $k = 0$  from the introduction of input signals to neurons, which are the initial state of the network. It is an iterative process and the state of the entire network changes in a way that reduces the value of the network's "energy" function (Lapunov function). The length of this process depends on the size of the network and the shape of local minima, and after its completion, the network reaches a stable state. After determining the answer, which is the determination of the minimum energy function, the playback process is interrupted and the network status is sent to its output. However, not always the correct output is obtained at the network output, which corresponds to one of the previously remembered patterns. One of the reasons may be false attractors, i.e. indirect local minima, in which the process may get stuck. This results in a response at the output of the network that does not correspond to any neuron involved in the learning process. The second reason for the incorrect reproduction of patterns at the network

output may be the ability to mix different components of stored patterns (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006).

A very important parameter of associative memory is its capacity, which is the maximum number of “patterns” stored and restored with the maximum error specified. For an error of 1% and using the Hebb rule in the learning process, the maximum memory capacity is 13.8% of the number of neurons that make up associative memory. This capacity may decrease as a result of an incorrect initial state of the network, which as a result of feedback causes further error duplication, which is equivalent to a decrease in memory capacity (Korbicz, Obuchowicz, Uciński, 1994).

Hopfield networks find numerous applications in solving optimization tasks and in the generation of specific signal sequences in a sequence. Thanks to auto-association, a trained Hopfield network can automatically correct or complete incomplete data. The network can reproduce the exact shape of an approaching object when the camera has recorded an incomplete or noisy image. This is of great importance in military operations, where, based on such an incomplete picture, the system must determine whether the object belongs to the enemy or not. The high efficiency of the Hopfield network is since the network reproduces reference signals from its memory resources, and the distorted input signal is only supposed to direct it when choosing the right image from all the analyzed ones (Korbicz, Obuchowicz, Uciński, 1994; Osowski, 2006; Tadeusiewicz, Szaleniec, 2015).

## **1. TRAINING OF HOPFIELD NETWORK**

For the training of Hopfield network, 7 letters of the Latin alphabet (A, E, I, N, T, V, X) were used, created in grayscale size 50 x 50 pixels in the bitmap format. Then, the loaded images into the memory are scaled using the scale factor  $k_x = 255$  to the range of values [0,1]. After scaling, each pixel of a given image is represented by values in the range [0,1], so the whole image appears in memory as a 50 x 50 matrix. To create an input matrix for a Hopfield network, “disassemble” of previously loaded matrices into individual vectors must be done (2500 values). From the individual vectors created in this way, a 7 x 2500 matrix can be “assembled”, which, after transposition, is the signal input into the Hopfield network. As noted before, the learning process of the Hopfield network is always stable, which is a big advantage of this network. However, this process is time-consuming, the time to train the Hopfield network lasted about 4 minutes (toolbox Neural Networks, Matlab/Simulink software). The process of teaching the Hopfield network is followed by the process of “assembling” images from the network output. This time, the images are multiplied by the  $k_x$  factor and transformed into a form that allows them to be displayed in graphic form (Fig. 2).

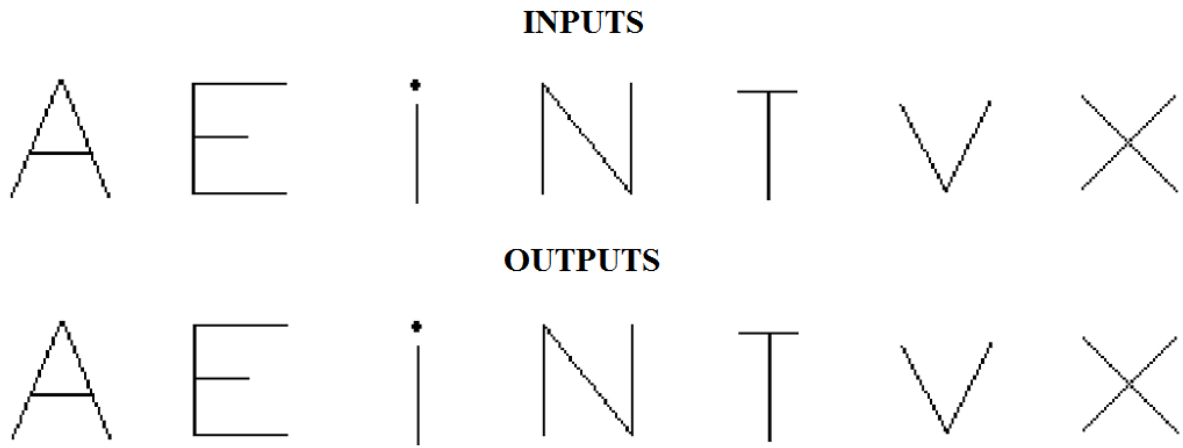


Fig. 2. Results of the Hopfield network test  
Source: Drawn by the authors.

The Hopfield network test presented in Fig. 2. illustrates the possibility of using the Hopfield network to learn images. The top line of images shows the images used as “patterns” for teaching the network. The bottom line, on the other hand, shows the responses of the already trained network to the listed images. The mapping of letters across the network is correct.

The following quality indicators were used to evaluate the quality of the mapping:

$$E = \frac{1}{2} \sum_{i=1}^{50} \sum_{j=1}^{50} (x_{ij} - y_{ij})^2 \quad (3)$$

where:

- $x_{ij}$  – elements of a two-dimensional matrix of image patterns (input),
- $y_{ij}$  - elements of a two-dimensional matrix of the image obtained as a result of the network (output).

Mapping errors of individual letters calculated according to the formula (3):

- $E_A = 0.1320$ ,
- $E_E = 0.1121$ ,
- $E_I = 0.2856$ ,
- $E_N = 0.1164$ ,
- $E_T = 0.2212$ ,
- $E_V = 0.2115$ ,
- $E_X = 0.2032$ .

Low values of mapping errors prove the correctness of the mapping graphically shown in Figure 2.

## 2. SIMULATION OF HOPFIELD NETWORK

The current section is analogous to the example discussed in the previous section. The only difference between these two cases is that in this one for the training of the Hopfield network distorted images of the letter E are used (Fig. 3).

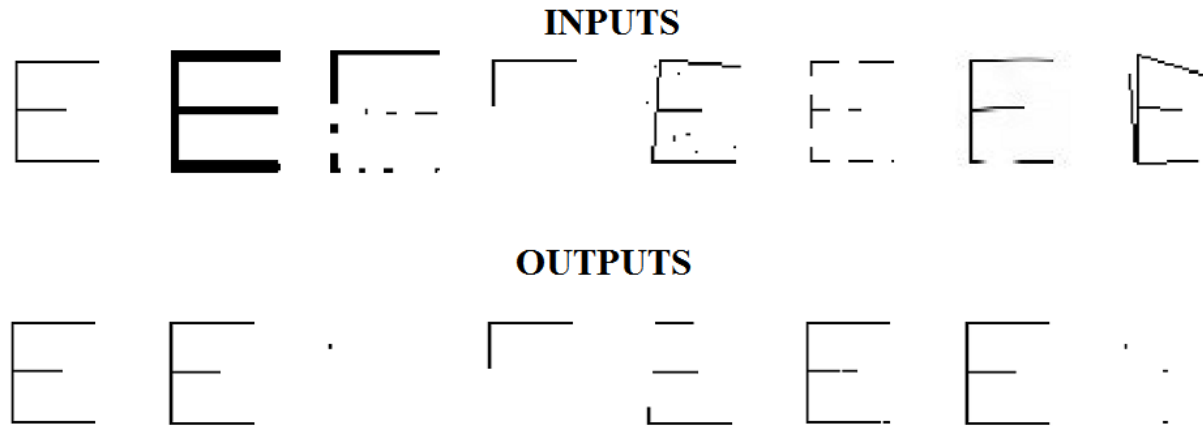


Fig. 3. Results of training and simulation of Hopfield network

Source: Drawn by the authors.

Figure 3 presents graphical results of the operation of the Hopfield network in the case that the input to the network is distorted images. The first column of images from the top left shows the pattern of the letter E and from below left the network response to the given image. It is easy to see that the network copes well with the trained object (image 1), object scaling (image 2) and slight interference (images 6 and 7). The Hopfield network, on the other hand, is unable to deal with the combination of reduced or increased elements in combination with various image distortions, e.g. noise (images 3, 4, 8) or “breaking out” in the letter (image 5).

Mapping errors of individual letters calculated according to the formula (3):

- $E_{E1} = 0,$
- $E_{E2} = 0,$
- $E_{E3} = 64,$
- $E_{E4} = 38.5,$
- $E_{E5} = 27,$
- $E_{E6} = 1,$
- $E_{E7} = 0,$
- $E_{E8} = 62.$

The large values of some of the presented above errors confirm the conclusion about the need for filtration of noisy images for the input of the Hopfield network.

### **3. TRAINING AND SIMULATION OF THE FEEDFORWARD NETWORK**

Filtration of the images can be done using feedforward neural networks. Parameters of the created feedforward network:

- number of neurons of the first layer of neurons: 44,
- number of neurons of the last layer of neurons: 11,
- the function of activating the first layer of neurons: logistic,
- the function of activating the last layer of neurons: logistic,
- goal: mean squared error (mse),
- network teaching method: gradient method with momentum and adaptive learning factor.

Parameters of the neural network training:

- maximum number of epochs: 1000,
- minimum error value (goal):  $10^{-6}$ ,
- minimum gradient value:  $10^{-12}$ ,
- the initial value of the learning rate: 0.99.

The selection of logistic functions as a function of activating both layers of neural networks is not accidental. In the case of images, the unipolar activation function significantly speeds up the learning process itself, because any values of pixel brightness in grayscale must be positive. In turn, the selection of 44 neurons of the input layer allows obtaining very low values of the target function with a relatively low number of iterations (Fig. 4). Also, a gradient method with momentum and adaptive learning factor was selected based on experiments.

As can be seen in Figure 4, the network only needed 135 epochs to learn the network with the objective function:  $mse = 0.000604$ . Network training has been interrupted due to a very low gradient  $<10^{-12}$ .

The analysis of figure 5 shows that the objective function achieved a minimal error in the 132 epoch, while the first graph of Figure 6 did not show the current gradient value in the 135 epoch due to the adopted scale. In turn, the bottom graph shows a gradual increase in the value of the learning coefficient, which at the end of learning in the last epoch assumed a value of about 478,74.

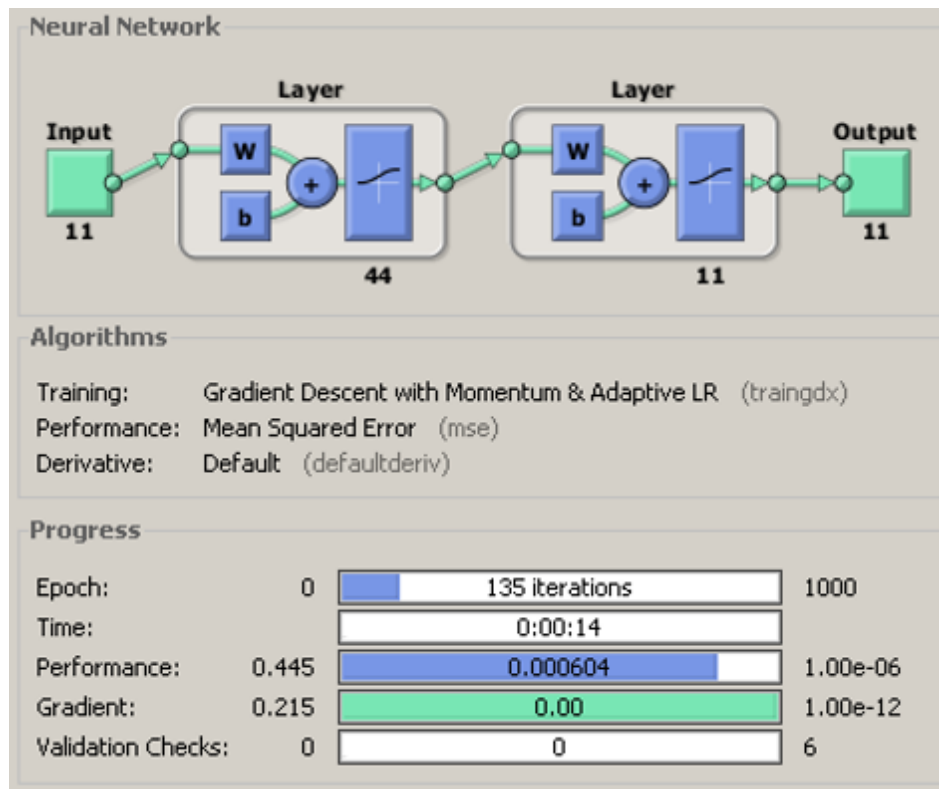


Fig. 4. Feedforward network learning process  
Source: Drawn by the authors.

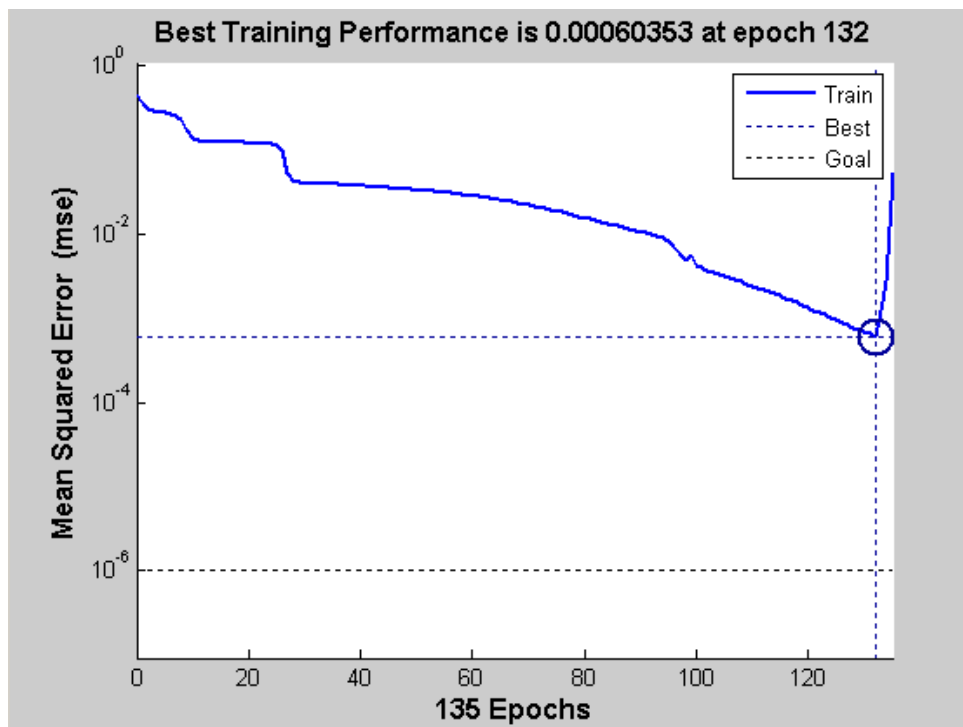


Fig. 5. The behavior of the objective function depending on the epochs  
Source: Drawn by the authors.



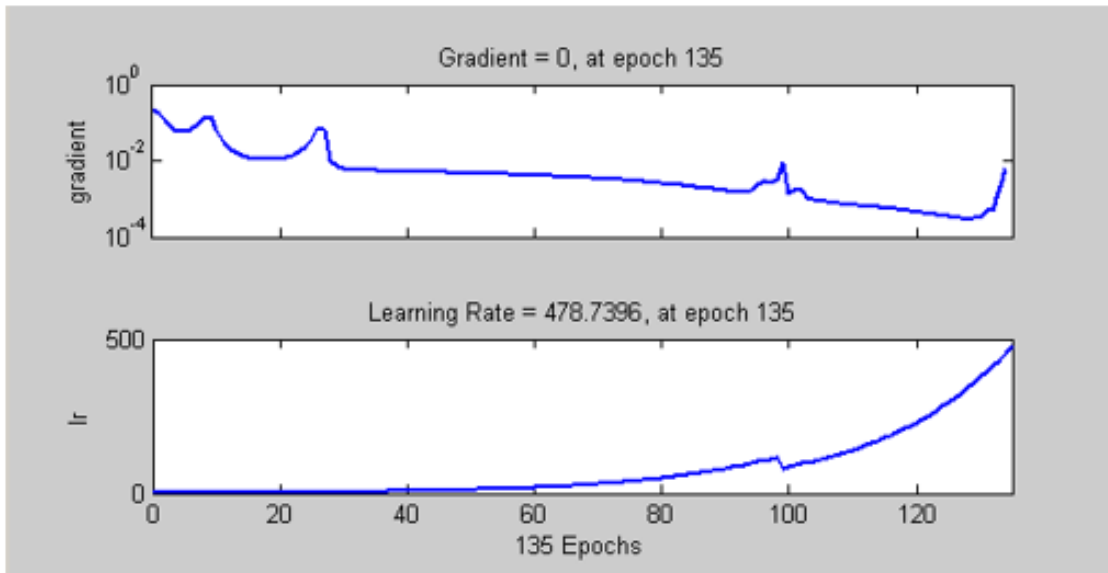


Fig. 6. Behavior of the gradient function and learning rate values during learning depending on the epochs

Source: Drawn by the authors.

Figure 7 shows distorted images of the letter E used for learning the feedforward network (the top row) and the network's response to the given disturbed image (the bottom row). The network correctly maps learned images.

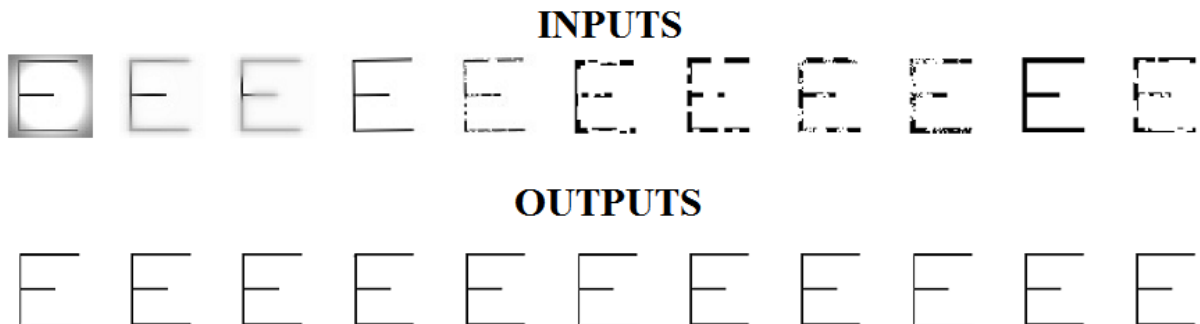


Fig. 7. Results of the feedforward network test

Source: Drawn by the authors.

Mapping errors of individual letters calculated according to the formula (3):

- $E_{E1} = 1,$
- $E_{E2} = 0,$
- $E_{E3} = 0,$
- $E_{E4} = 0.5000,$
- $E_{E5} = 0,$
- $E_{E6} = 0,$
- $E_{E7} = 0,$
- $E_{E8} = 0,$
- $E_{E9} = 0,$

- $E_{E10} = 0$ ,
- $E_{E11} = 0$ .

Zero error value for most images and minimum error values for the first and fourth mapping confirms the previous conclusion regarding the correctness of the mapping of the network.

#### **4. CONCEPT OF THE HYBRID SYSTEM**

The concept of the described hybrid system is shown in figure 8. Two neural networks: feedforward (FF) and Hopfield are connected in series in the mentioned order.

The output of the feedforward neural is also the input for Hopfield network (Fig. 8).

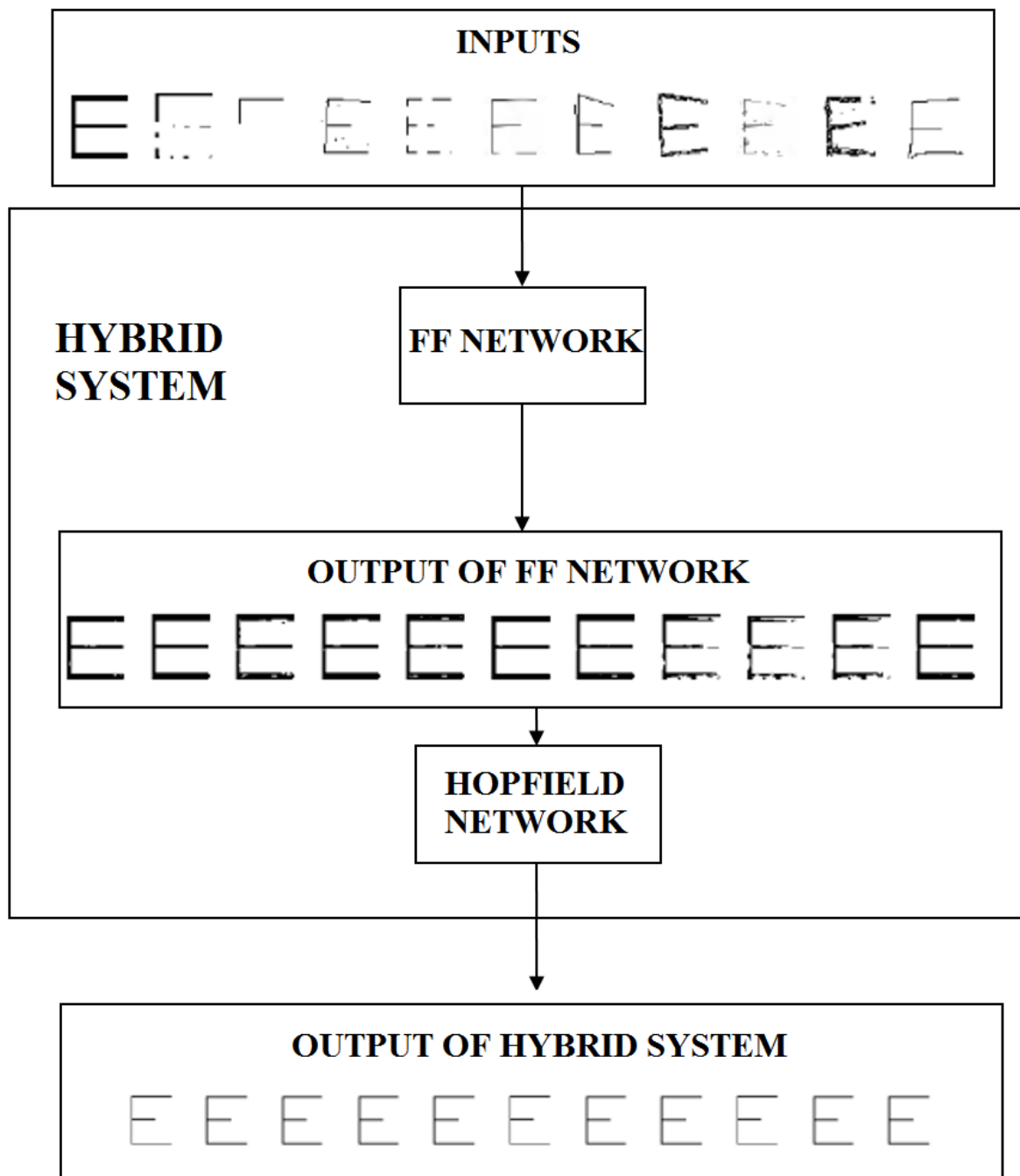


Fig. 8. Concept of the hybrid system

Source: Drawn by the authors.

## 5. RESULTS OF THE HYBRID SYSTEM

On the feedforward network input was given images that were not used to teach the network. As can be seen in figure 9, there are incomplete parts of the letter E (element 3), several distorted and “torn out” images of the letter E (elements 2, 4, 5, 6) as well as images where the shape of the letter E is disturbed (elements 8, 9, 10, 11).

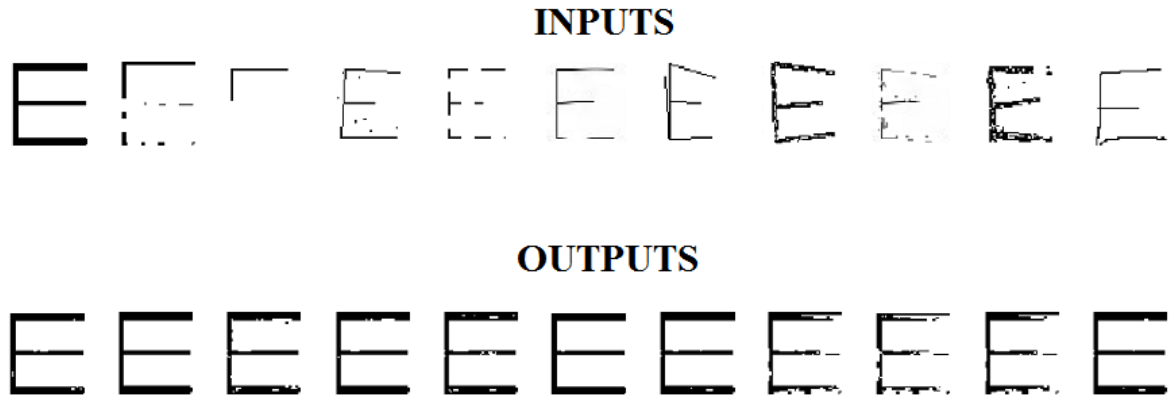


Fig. 9. Results of feedforward's part of the hybrid system  
Source: Drawn by the authors.

Feedforward's part of the hybrid system errors according to the formula (3):

- $E_{F1} = 1,$
- $E_{F2} = 276,$
- $E_{F3} = 280.5,$
- $E_{F4} = 273.5,$
- $E_{F5} = 283.5,$
- $E_{F6} = 261,$
- $E_{F7} = 279.5,$
- $E_{F8} = 275,$
- $E_{F9} = 182.5,$
- $E_{F10} = 161,$
- $E_{F11} = 179.$

Despite the relatively large values of mapping errors, the feedforward network can correctly map the shape of a given letter.

The inputs and outputs for Hopfield network are shown in figure 10.

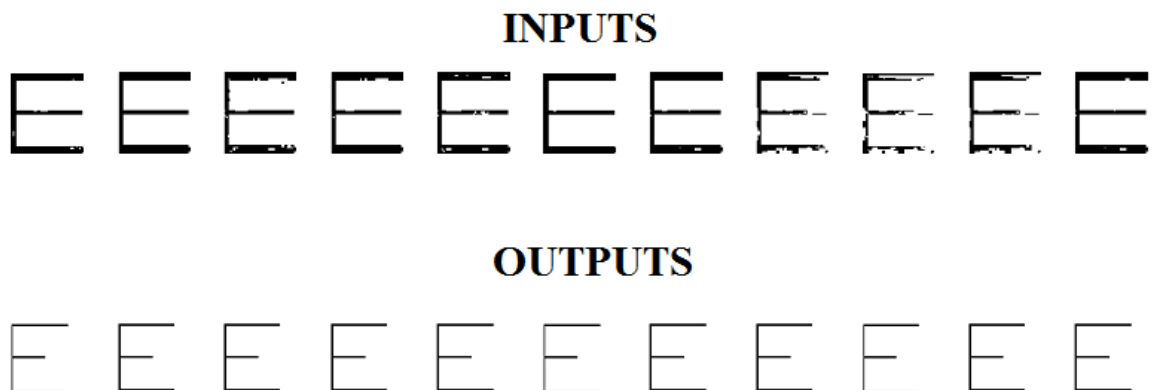


Fig. 10. Results of Hopfield's part of the hybrid system  
Source: Drawn by the authors.

The images provided as an input for Hopfield network have a clear shape of the analyzed letter, which facilitates the operation of the Hopfield network. In the end, Hopfield network correctly maps the recognized one and thus confirms the correctness of the solution.

Hopfield's part of the hybrid system errors according to the formula (3):

- $E_{H1} = 6.04,$
- $E_{H2} = 6.02,$
- $E_{H3} = 6.80,$
- $E_{H4} = 6.87,$
- $E_{H5} = 5.60,$
- $E_{H6} = 5.99,$
- $E_{H7} = 5.87,$
- $E_{H8} = 4.03,$
- $E_{H9} = 4.83,$
- $E_{H10} = 3.86,$
- $E_{H11} = 6.08.$

Small values of mapping errors confirm the accuracy of the entire system as well as the accuracy of the Hopfield network itself.

## **CONCLUSIONS**

The numerous advantages and disadvantages of the Hopfield network demonstrated in the article allowed to create a system capable of recognizing letters consisting of two neural networks: a two-layer feedforward network and a Hopfield network. The simulations show that the Hopfield network is suitable for the reasoning element at the end of the system. The feedforward network used in turn has great generalization capabilities, which allowed it to be used as a “filter” for the Hopfield network.

The Hopfield network acting as auto-associative memory can learn to map many letters, which is a huge advantage of this structure. In turn, the feedforward network should be taught using as many distorted images as possible, which significantly increases the effort needed to properly train the network. However, the mapping of the feedforward network of a given letter does not have to be close to the pattern, because it is enough for the mapping to approximate the shape of a given letter. The Hopfield network handles such interference very well, as demonstrated in this article.

The created system can be easily expanded by providing a larger number of letters on both networks and thus can work well as an OCR character recognition system.

## REFERENCES

- [1] KORBICZ, J., OBUCHOWICZ, A., & UCIŃSKI, D. (1994). *Sztuczne sieci neuronowe: podstawy i zastosowania*. Akademicka Oficyna Wydawnicza PLJ. 30 - 112.
- [2] MUSZYŃSKA, M. (2015). Neuronowe modelowanie nieliniowości aktorów. *Przegląd Mechaniczny*, (1), 30-36.
- [3] RYKAŁA, Ł. (2018). Modelowanie ruchu niejednorodnych brył z wykorzystaniem metod sztucznej inteligencji. *Przegląd Mechaniczny*.
- [4] KIJEK, M., BRZEZINSKI, M., RYKAŁA, L., ZELKOWSKI, J., (2018). *Neural Algorithm of Driver Selection for Transport Tasks*. In *Proceedings of the 22nd International Conference, Transport Means* (pp. 489-494).
- [5] OSOWSKI, S. (2006). *Sieci neuronowe do przetwarzania informacji*, Oficyna Wyd. Politechniki Warszawskiej, Warszawa. 220 - 260.
- [6] TADEUSIEWICZ, R., & SZALENIEC, M. (2015). *Leksykon sieci neuronowych*. Projekt Nauka. Fundacja na rzecz promocji nauki polskiej.