ISSN 2224-087X. Електроніка та інформаційні технології. 2022. Випуск 17. С. 36–53
Electronics and Information Technologies. 2022. Issue 17. P. 36–53

UDK 548.0:535 DOI: https://doi.org/10.30970/eli.17.4

SPECIFICS OF THE LEARNING ERROR DEPENDENCE OF MULTILAYERED NEURAL NETWORKS FROM THE ACTIVATION FUNCTION DURING THE PROCESS OF PRINTED DIGITS IDENTIFICATION

S. Sveleba[1], I. Katerynchuk[1], I. Kuno[1], O. Semotiuk[2], Ya. Shmyhelskyy[1] and N. Sveleba[1]

[1]Ivan Franko National University of Lviv,
Universytetska str.1, Lviv, 79000, Ukraine,
incomlviv@gmail.com
[2] Ukrainian Academy of Printing,
Pid Goloskom str.19, Lviv, 79020, Ukraine

In this paper, we investigated the learning process during an identification of printed digits from the type of an activation function. The study of the activation function type and the number of iterations in the learning process of the neural system was carried out using the Fourier spectra analysis of the learning error function and branching diagrams. For this purpose, a program for the multilayer neural network was developed in the Python software, which involves setting the number of hidden layers as well as the number of neurons inside them and changes in the learning rate. The learning rate was considered as a constant, and its optimal value, where the best learning rate is observed, was determined. To analyze the learning rate effect on the educational process, we used a logistic function describing the frequency doubling process. It is shown that the learning error function is characterized by bifurcation processes leading to a chaotic state when η>0.8. The optimal learning rate value that determines the emergence of the doubling process of the local minima number is determined. It was found that the sigmoidal activation function (as compared to the activation functions ReLU and hyperbolic tangent) best satisfies the learning process of the three-layer neural network for recognizing digits, given an array of 4x7 zeros and ones. Compared to other activation functions, there is an insignificant change in the learning error during the transition from one digit to another. It is shown that an increase in the number of hidden layers does not lead to a sharp increase during the learning error. An increase in the learning iterations number is accompanied by the appearance of periodic dependences of the logistic function value of the learning rate, the period of which is a variable of the number of iterations and the learning rate. Using Fourier spectra of the error function from the learning rate value, it can be argued that an increase in the number of iterations leads to an increase in the number of harmonics, which eventually leads to the appearance of a chaotic state of the neural network.

Keywords: Multilayered neural network, activation function, optimal learning rate, digital identification.

1. Introduction

It is known [1] that deep learning is a branch of machine learning and is based on appropriate algorithms. Deep neural networks are used to solve various problems related to clustering, approximation and events forecasting, language identification, text processing, and others.

The classical error backpropagation algorithm works well with two-layer and three-layer neural networks, however, as the depth of the network is further increased, there may be some problems [2]. Indeed, as it was pointed out in [3], an increase in both the number of hidden layers and the number of neurons in them stimulates an increase in the chaotic state of the neural network (NN) learning error versus the epoch number and the learning rate diagram. A chaotic state, which appears as a result of increasing both the number of hidden layers and the number of neurons in them, leads to the NN transition into a state, characterized by the learning process absence. It is especially evident when the number of hidden layers changes. One of the reasons for such behavior of a neural multilayer network is the so-called gradient decay [2]. During the error spread from the source layer to the target layer, the current result is multiplied by the activation function derivative. Using a traditional sigmoid activation function whose derivative has a range of fewer than 5 units, the error can be close to zero after running through several layers. On the contrary, if we take an activation function where the derivative is unbounded (such as the hyperbolic tangent), there can be an increase in the learning error, leading to unstable network learning [4]. Hence, let us consider the type effect (sigmoidal, hyperbolic tangent, ReLU function), activation function, and number of iterations on the learning process of the neural system.

2. . ReLU Activation (rectified linear unit)

The activation function ReLU (rectified linear unit) has gained much attention in recent years. Its derivative is either one or zero, and therefore no gradient expansion or attenuation can occur. Moreover, the use of this function leads to thinning of the scales. The positive aspects of this activation function, according to the paper [1]:

1. Sigmoid and hyperbolic tangent requires a large number of system resources to perform operations, such as lifting to a degree, which with a large number of layers and neurons slows down the learning process, while ReLU can be implemented through a simple threshold transformation of matrix activity to zero.

2. The use of ReLU significantly increases the convergence speed of stochastic gradient descent compared to sigmoid and hyperbolic tangents. This is considered to be due to the linear nature and lack of this function's saturation.

The downsides. Unfortunately, ReLU is not always reliable enough in the learning process anyway. For example, a large gradient passing through ReLU may lead to such an update of the weights that this neuron is never activated again. This problem is solved by choosing the correct learning rate. Currently, there are several varieties of this function, determined by its parameters [5]. In particular, in [6], on handwritten digits identification based on mnist and multilayer neural network, the activation function ReLU was used. It increased the learning rate of the network compared to sigmoid, and the learning error was less.

In this paper, we will study the effect of the type of activation function and the number of iterations on the learning process of the neural system by analyzing the Fourier spectra of the learning error function and branching diagrams. For this purpose, we developed a program for the multilayer neural network in Python software, which involves setting the number of hidden layers and the number of neurons in them and changing the learning parameter in the range of $0.001 \div 10$. This interval of variation η was chosen taking into account the data obtained in [7], where the study of the learning rate impact on the learning process in the multilayered NN was conducted. Each layer of this NN will be considered as a separate deterministic system, for which we investigate the branching diagram using the mapping form function:

$$x_{n+1} = \eta - x_n - x_n^{\,2},$$

where n — is a step, η — a parameter that determines the learning rate.

Its fixed points:

$$x_{1,2} = -1 \pm (\eta + 1)^{1/2},$$

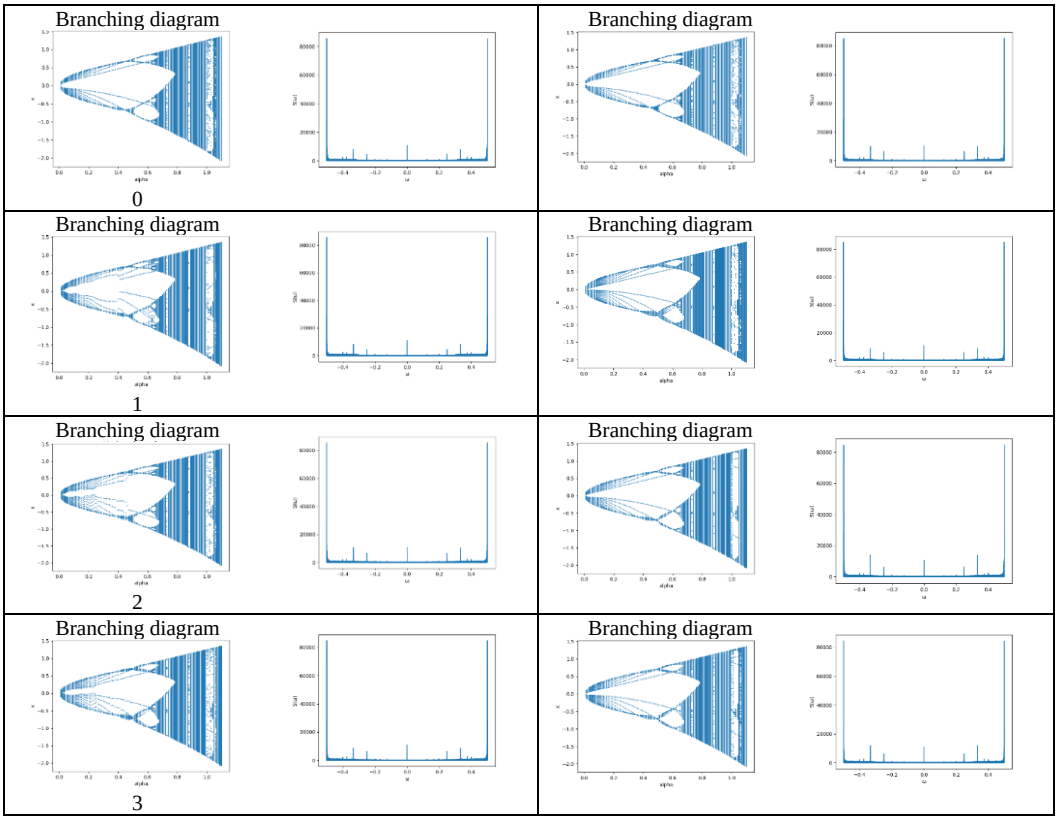
the eigenvalues, which can be calculated as follows:

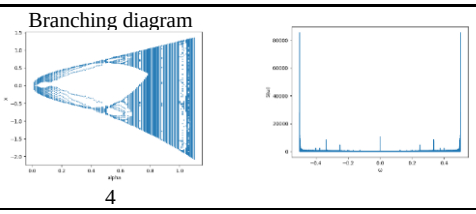
$$\rho_1 = 1 - 2(\eta + 1)^{1/2}.$$

The choice of such a logistic representation is determined by the fact that it describes the doubling process of the oscillation frequency [8]. In our case, this process is caused by the emergence of local minima while approaching the global minimum and repeatedly passing through the global minimum.

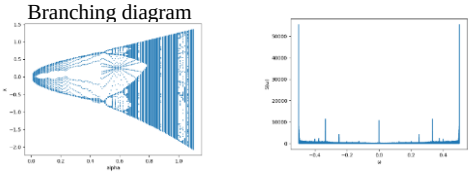
3. Results and analysis

Fig. 1 and Fig. 2 show the branching diagrams and Fourier spectra for the activation functions, which are described by the sigmoid and hyperbolic tangents, respectively. The NN had three layers and 28 neurons in each layer, assuming that the activation function coefficient is c=1. For the ReLU activation function, if c=1, the learning process was missing.

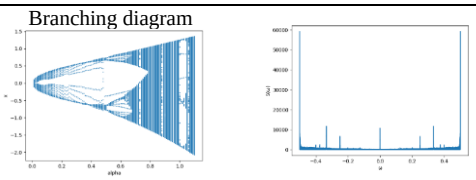




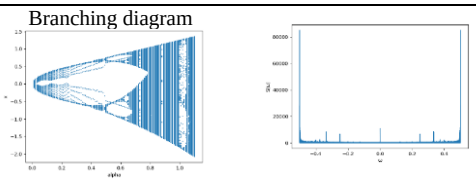
4



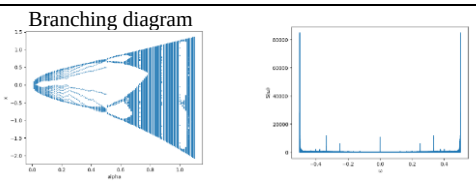
5



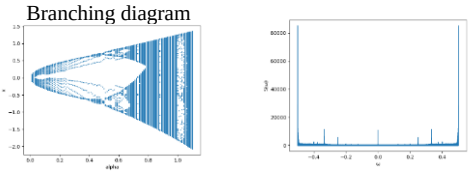
6



7

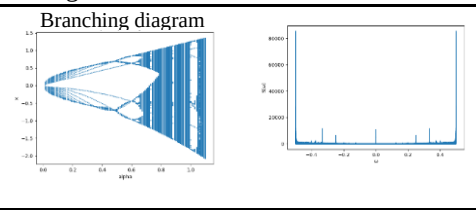


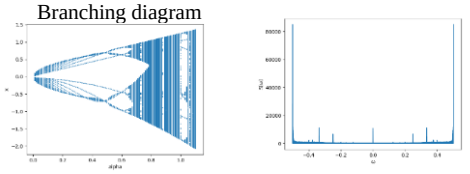
8

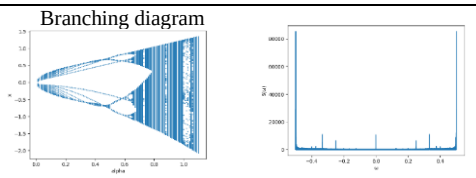


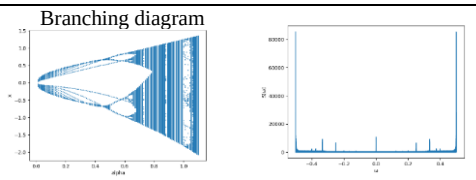
9

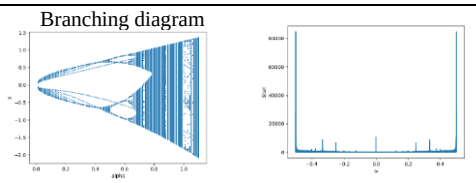
Figure 1. Fourier spectra and branching diagrams for printed digits given by an index of 4x7 zeros and ones, under the following conditions: hyperbolic tangent activation function, three hidden layers with 28 neurons per layer, 100 iterations, C=1.











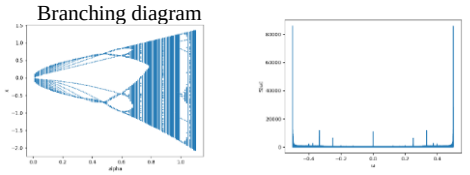


Figure 2. Fourier spectra and branching diagrams for printed digits defined by an index of 4x7 zeros and ones, with the following conditions: sigmoidal activation function, three hidden layers with 28 neurons per layer, 100 iterations, C=1.

According to Fig. 1 and Fig. 2, with 100 iterations of the three-layer neural network, the best learning process occurs with a sigmoidal activation function. The learning error is 0.1%. According to the branching diagrams, when a sigmoid is used as the activation function, they are similar and indicate that the learning process for all digits is almost the same with identical learning errors (Fig. 3). When the hyperbolic tangent is used as the activation function (Fig. 1), the learning process for different digits proceeds with different accuracy. It is indicated by the branching diagrams for digits "4", "5", and "9". Taking into account the Fourier spectra shown in Fig. 1 and Fig. 2, a slight difference in the magnitude of the first and second harmonics and the amount of fuzziness can be observed in favor of the sigmoidal activation function.

digit = 0; optimum alpha = 0.55; minimum error = 0.00117 digit = 1; optimum alpha = 0.55; minimum error = 0.00117 digit = 2; optimum alpha = 0.55; minimum error = 0.00118 digit = 3; optimum alpha = 0.55; minimum error = 0.00122 digit = 4; optimum alpha = 0.55; minimum error = 0.00118 digit = 5; optimum alpha = 0.56; minimum error = 0.00115 digit = 6; optimum alpha = 0.55; minimum error = 0.00117 digit = 7; optimum alpha = 0.55; minimum error = 0.00113 digit = 8; optimum alpha = 0.55; minimum error = 0.00123 digit = 9; optimum alpha = 0.55; minimum error = 0.00121	Sigmoidal activation function, three hidden layers with 28 neurons per layer, 100 iterations, C=1.
no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes	Hyperbolic tangent activation function, three hidden layers with 28 neurons per layer, 100 iterations, C=1.

Figure 3. The optimal learning rate and the minimum learning error rate for the two activation functions.

digit = 0; optimum alpha = 0.55; minimum error = 0.00759 digit = 1; optimum alpha = 0.55; minimum error = 0.00292 digit = 2; optimum alpha = 0.55; minimum error = 0.00364 digit = 3; optimum alpha = 0.55; minimum error = 0.00235 digit = 4; optimum alpha = 0.55; minimum error = 0.00504 digit = 5; optimum alpha = 0.55; minimum error = 0.00455 digit = 6; optimum alpha = 0.56; minimum error = 0.0052 digit = 7; optimum alpha = 0.55; minimum error = 0.00319 digit = 8; optimum alpha = 0.55; minimum error = 0.00237 digit = 9; optimum alpha = 0.55; minimum error = 0.06594	Hyperbolic tangent activation function, three hidden layers with 28 neurons per layer, 1000 iterations, C=0.5.
digit = 0; optimum alpha = 0.55; minimum error = 0.00235 digit = 1; optimum alpha = 0.56; minimum error = 0.00239 digit = 2; optimum alpha = 0.56; minimum error = 0.00228 digit = 3; optimum alpha = 0.55; minimum error = 0.00237 digit = 4; optimum alpha = 0.55; minimum error = 0.00246 digit = 5; optimum alpha = 0.55; minimum error = 0.00239 digit = 6; optimum alpha = 0.56; minimum error = 0.00244 digit = 7; optimum alpha = 0.55; minimum error = 0.00231 digit = 8; optimum alpha = 0.55; minimum error = 0.00246 digit = 9; optimum alpha = 0.55; minimum error = 0.00244	Sigmoidal activation function, three hidden layers with 28 neurons per layer, 1000 iterations, C=0.5.

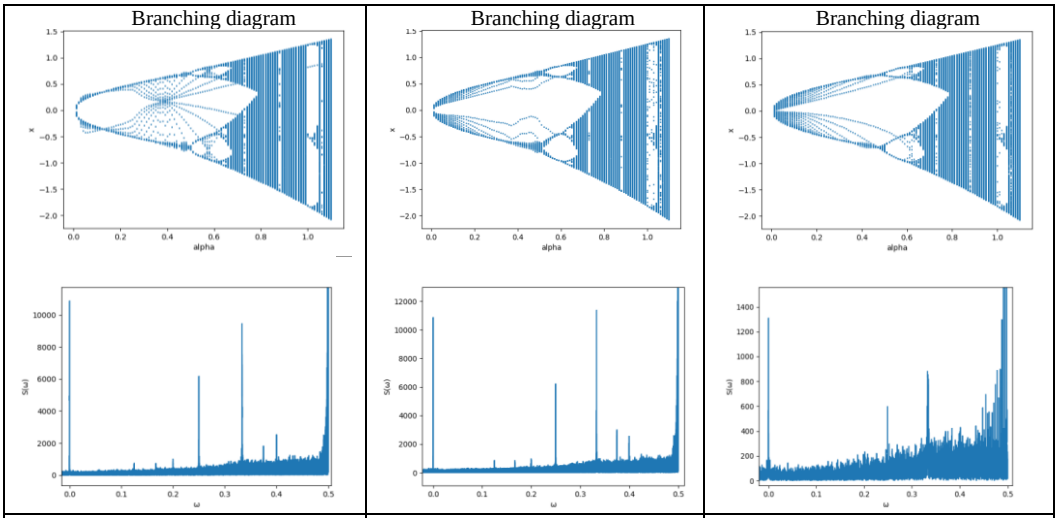
Figure 4. Optimal learning rate and minimum learning error for two activation functions.

It is clear that when the weights change, the slope angle of the activation function graph also changes. It is useful if we model different interconnection densities between inputs and outputs. The constant $\alpha=C$ determines the slope of the activation function. As α decreases, the sigmoid becomes flatter, and as α increases, the sigmoid function approaches the unit jump function. Decreasing the value of parameter $\alpha=C$ to the C=0.5 causes an increase in the learning error to ≈0.22%, and this error rate is the same for all digits. When applied as an activation function, the hyperbolic tangent entails an increase in the learning error (Fig. 4). It should also be noted that the error rate is different for each digit and reaches a maximum value of ≈0.7%. In particular, we should note the digits "0", "4", "5" and "9", for which the learning process is accompanied by a greater, almost twofold, learning error compared to other digits.

A further decrease in the value of constant C to 0.1 entails a slight increase in the learning error for the sigmoidal activation function to 1.8% and the hyperbolic tangent a slight decrease in the learning error to 0.2%. Although for some digits ("3" and "7"), there is an increase in the learning error to ≈1.1÷1.3% (Fig. 5). For these digits, we will consider the behavior of branching diagrams and Fourier spectra in a scaled-up way (Fig.6). The branching diagrams shown in Fig. 6 are the same, except there is a larger opacity window for the activation function, which is described by the hyperbolic tangent. As for the Fourier spectra, when the activation function is a hyperbolic tangent, the observed noisiness (arising from the higher harmonics existence) is much smaller with the simultaneous presence of a significant increase in the first and second harmonic meanings. It determines the magnitude of the minimum error and the optimum learning rate (Fig. 6). The value of the obtained optimal learning rate, according to Fig. 3 - Fig. 5, practically does not depend on the type of the activation function and the iteration rate.

no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes	ReLU_digital_9_28neur_layer_3layers _alpha_0.1-1.2_1000_iter_Phase Diagram C+0.01 ReLU activation function, three hidden layers with 28 neurons per layer, 1000 iterations, C=0.1
digit = 0; optimum alpha = 0.55; minimum error = 0.01796 digit = 1; optimum alpha = 0.55; minimum error = 0.01769 digit = 2; optimum alpha = 0.55; minimum error = 0.01718 digit = 3; optimum alpha = 0.55; minimum error = 0.01745 digit = 4; optimum alpha = 0.55; minimum error = 0.01832 digit = 5; optimum alpha = 0.55; minimum error = 0.01744 digit = 6; optimum alpha = 0.55; minimum error = 0.017 digit = 7; optimum alpha = 0.55; minimum error = 0.01711 digit = 8; optimum alpha = 0.55; minimum error = 0.01849 digit = 9; optimum alpha = 0.55; minimum error = 0.01845	Sigmoidal activation function, three hidden layers with 28 neurons per layer, 1000 iterations, C=0.1
digit = 0; optimum alpha = 0.55; minimum error = 0.0022 digit = 1; optimum alpha = 0.55; minimum error = 0.00174 digit = 2; optimum alpha = 0.55; minimum error = 0.00199 digit = 3; optimum alpha = 0.55; minimum error = 0.01274 digit = 4; optimum alpha = 0.56; minimum error = 0.00241 digit = 5; optimum alpha = 0.55; minimum error = 0.00186 digit = 6; optimum alpha = 0.55; minimum error = 0.00161 digit = 7; optimum alpha = 0.55; minimum error = 0.01119 digit = 8; optimum alpha = 0.55; minimum error = 0.00159 digit = 9; optimum alpha = 0.55; minimum error = 0.00189	Hyperbolic tangent activation function, three hidden layers with 28 neurons per layer, 1000 iterations, C=0.1

Figure 5. Optimal learning rate value and minimum learning error for the three activation functions.



Sigmoid activation function by C=0.1 value	Hyperbolic tangent activation function by C=0.1 value	ReLU activation function by C=0.01 value

Figure 6. Branching diagrams and Fourier spectra in enlarged scale for the printed digit "7" given by an index of 4x7 zeros and ones, assuming three hidden layers with 28 neurons per layer, 1000 iterations.

Fig. 6 shows the branching diagrams for three-layered NNs for different activation functions.

Depending on the learning rate parameter (η) value, the mapping has a different number of fixed points. When $0 < \eta < 0.4$, the number of fixed points does not change, but both fixed points are unstable. Since the reflection is bounded, the absence of a point attractor implies the formation of a more complex attractor of the limit cycle type (as shown in [3]). Although the reflection itself has no stable fixed points, its squaring of such stable fixed points is possible. Therefore, the bifunctional diagram in this area shows the line branching. For the values of $\eta >$ 0.4 (Fig. 6), the limit cycle loses its stability. At such parameter values, stable fixed points should be sought in higher-order representations. Such a situation with the period of the limit cycle of higher-order mappings exists in a certain region of the parameter η and then changes - there is a further increase in local minima, and so on.

For values of $\eta > 0.78$ (Fig. 6), none of the critical cycles is stable. The chaotic behavior of the system starts from this point.

One of the ways to solve the problem of avoiding the chaotic states of the neural network is to automatically define the minimum number of solutions on the diagram of the logistic function, i.e., to define the value of the training rate, at which these solutions (the number of local minima) are doubling. This mechanism assumes the absence of a chaotic state, causing the state appearance where there is no neural network learning. The algorithm for solving this problem consists in defining the number of solutions on the logistic function diagram at a given value of the training rate, at which the process of doubling the number of bifurcations takes place and determines the optimal value of the training rate and the optimal value of the error. As mentioned by [5], the use of the ReLU activation function significantly increases the con-

vergence speed of the stochastic gradient descent as compared to sigmoid and hyperbolic tangent, which led to a reduction in the number of iterations to achieve a given learning accuracy. According to [5], the activation function ReLU increased the learning rate of the network compared to sigmoid, and the learning error was less. Therefore, let us consider the dependence of the learning accuracy and the optimal value of the learning rate on the number of iterations for each activation function.

Iterations number	hyperbolic tangent with C=0.1	ReLU C=0.1
1000	digit = 0; optimum alpha = 0.55; minimum error = 0.0022 digit = 1; optimum alpha = 0.55; minimum error = 0.00174 digit = 2; optimum alpha = 0.55; minimum error = 0.00199 digit = 3; optimum alpha = 0.55; minimum error = 0.01274 digit = 4; optimum alpha = 0.56; minimum error = 0.00241 digit = 5; optimum alpha = 0.55; minimum error = 0.00186 digit = 6; optimum alpha = 0.55; minimum error = 0.00161 digit = 7; optimum alpha = 0.55; minimum error = 0.01119 digit = 8; optimum alpha = 0.55; minimum error = 0.00159 digit = 9; optimum alpha = 0.55; minimum error = 0.00189	no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes
100	digit = 0; optimum alpha = 0.55; minimum error = 0.06338 digit = 1; optimum alpha = 0.56; minimum error = 0.04913 digit = 2; optimum alpha = 0.55; minimum error = 0.07122 digit = 3; optimum alpha = 0.55; minimum error = 0.06865 digit = 4; optimum alpha = 0.56; minimum error = 0.02207 digit = 5; optimum alpha = 0.55; minimum error = 0.02615 digit = 6; optimum alpha = 0.55; minimum error = 0.01866 digit = 7; optimum alpha = 0.56; minimum error = 0.02001 digit = 8; optimum alpha = 0.55; minimum error = 0.02568 digit = 9; optimum alpha = 0.55; minimum error = 0.05277	digit = 0; optimum alpha = 0.55; minimum error = 0.00178 digit = 1; optimum alpha = 0.55; minimum error = 0.00737 digit = 2; optimum alpha = 0.55; minimum error = 0.00423 digit = 3; optimum alpha = 0.55; minimum error = 0.00197 digit = 4; optimum alpha = 0.55; minimum error = 0.00384 digit = 5; optimum alpha = 0.55; minimum error = 0.00204 digit = 6; optimum alpha = 0.55; minimum error = 0.00272 digit = 7; optimum alpha = 0.55; minimum error = 0.0058 digit = 8; optimum alpha = 0.55; minimum error = 5e-05 digit = 9; optimum alpha = 0.55; minimum error = 0.00659
10	digit = 0; optimum alpha = 0.58; minimum error = 0.06623 digit = 1; optimum alpha = 0.53; minimum error = 0.06076 digit = 2; optimum alpha = 0.53; minimum error = 0.06683 digit = 3; optimum alpha = 0.53; minimum error = 0.08125 digit = 4; optimum alpha = 0.53; minimum error = 0.06486 digit = 5; optimum alpha = 0.53; minimum error = 0.07324 digit = 6; optimum alpha = 0.53; minimum error = 0.07832 digit = 7; optimum alpha = 0.53; minimum error = 0.08034 digit = 8; optimum alpha = 0.58; minimum error = 0.07294 digit = 9; optimum alpha = 0.53; minimum error = 0.08781	digit = 0; optimum alpha = 0.53; minimum error = 0.02215 digit = 1; optimum alpha = 0.53; minimum error = 0.02354 digit = 2; optimum alpha = 0.53; minimum error = 0.02792 digit = 3; optimum alpha = 0.53; minimum error = 0.0278 digit = 4; optimum alpha = 0.58; minimum error = 0.00855 digit = 5; optimum alpha = 0.53; minimum error = 0.0289 digit = 6; optimum alpha = 0.53; minimum error = 0.02654 digit = 7; optimum alpha = 0.58; minimum error = 0.01707 digit = 8; optimum alpha = 0.58; minimum error = 0.01077 digit = 9; optimum alpha = 0.53; minimum error = 0.01835
5	digit = 0; optimum alpha = 0.58; minimum error = 0.06623 digit = 1; optimum alpha = 0.53; minimum error = 0.06076 digit = 2; optimum alpha = 0.53; minimum error = 0.06683 digit = 3; optimum alpha = 0.53; minimum error = 0.08125 digit = 4; optimum alpha = 0.53; minimum error = 0.06486 digit = 5; optimum alpha = 0.53; minimum error = 0.07324 digit = 6; optimum alpha = 0.53; minimum error = 0.07832 digit = 7; optimum alpha = 0.53; minimum error = 0.08034 digit = 8; optimum alpha = 0.58; minimum error = 0.07294 digit = 9; optimum alpha = 0.53; minimum error = 0.08781	digit = 0; optimum alpha = 0.56; minimum error = 0.05368 digit = 1; optimum alpha = 0.56; minimum error = 0.03204 digit = 2; optimum alpha = 0.56; minimum error = 0.04235 digit = 3; optimum alpha = 0.56; minimum error = 0.05658 digit = 4; optimum alpha = 0.56; minimum error = 0.01315 digit = 5; optimum alpha = 0.56; minimum error = 0.0655 digit = 6; optimum alpha = 0.56; minimum error = 0.0539 digit = 7; optimum alpha = 0.56; minimum error = 0.0281 digit = 8; optimum alpha = 0.56; minimum error = 0.04212 digit = 9; optimum alpha = 0.56; minimum error = 0.03111

Figure 7. Dependence of the optimal learning rate and the minimum learning error on the iteration count for two different activation functions.

Fig.7 and Fig.8 show the dependence of the optimal learning rate and the minimum learning error for each digit of a given dimension of 4x7 zeros and ones for each activation function

and iteration number. Comparing the ReLU activation function with C=0.1 (this function corresponds to the best learning of a neural system with a given architecture (3 hidden layers with 28 neurons per layer)) with the hyperbolic tangent of C=0.1 at different iterations, we can make the following conclusions:

Firstly, with an increasing number of iterations (from 5 to 100), the training error decreases practically twice (for the hyperbolic tangent at 5 iterations for the digit "9", the error ≈8.8% to at 100 iterations the error ≈5.2%; for ReLU at 5, iterations for the digit " 9" the error ≈3.1% to at 100 iterations (error ≈0.65%) for the hyperbolic tangent, and for ReLU practically by 5 times.

Second, for these activation functions (hyperbolic tangent and ReLU), a non-monotonic change in the learning error is traced both from the number of iterations and during the transition from digit to digit (Fig. 7).

Consequently, when comparing these two activation functions, the ReLU activation function of C=0.1 was the best at training this neural network for the identification of the printed digit. Regarding the optimal learning rate, this parameter remained almost unchanged when changing both the activation function and the iteration number and remained equal of 0.5.

Comparing the activation functions, the sigmoidal function with the C=1 and ReLU with C=0.1 parameters, we can state that if at 100 iterations when applying the ReLU activation function, the best learning of the neural network is generally observed (approximately at 0.1%), then at a decrease in the number of iterations (Fig. 8) the best learning when applying the sigmoid function as an activation function is traceable. Thus, in particular, at 5 iterations the learning error at the application of the sigmoidal activation function is ≈2.5%, while at the application of the ReLU activation function, the error grew to ≈4%. It should be noted that when using the ReLU activation function, non-monotonic behavior of the learning error is observed when passing from one digit to another (the smallest value of the learning error for the digit "7" ≈2.8% to the largest for the digit "5"≈6.5%, Fig. 8).

Consequently, the sigmoidal activation function handles the learning process of the neural network better than the hyperbolic tangent and ReLU. In addition, it does not lead to a jump dependence of the learning error when moving from one digit to another. It is an important factor when recognizing digits at insignificant (100) values of the number of iterations.

The obtained value of the optimal value of the learning rate is the same for these activation functions. Decreasing the number of iterations leads to an insignificant increase for all the activation functions used.

Let us consider how a change in the number of hidden layers will affect the learning error when using the sigmoidal and ReLU activation functions. Fig. 9 shows the dependence of the learning error on the number of hidden layers, with 28 neurons in each layer, for 10 iterations. An increase in the number of hidden layers leads to an increase in the learning error of ≈0.1% at 3 layers to ≈2% for the 10-layer network at the sigmoidal activation function and from a learning error of ≈2% at 3 layers to ≈4% for the 10-layer network at the ReLU activation function.

Therefore, when the number of hidden layers increases, the smallest value of the learning error should be expected for the sigmoidal activation function.

Iterations number	Sigmoid C=1	ReLU C=0.1
1000	digit = 0; optimum alpha = 0.55; minimum error = 0.00117 digit = 1; optimum alpha = 0.55; minimum error = 0.00117 digit = 2; optimum alpha = 0.55; minimum error = 0.00118 digit = 3; optimum alpha = 0.55; minimum error = 0.00122 digit = 4; optimum alpha = 0.55; minimum error = 0.00118 digit = 5; optimum alpha = 0.56; minimum error = 0.00115 digit = 6; optimum alpha = 0.55; minimum error = 0.00117 digit = 7; optimum alpha = 0.55; minimum error = 0.00113 digit = 8; optimum alpha = 0.55; minimum error = 0.00123 digit = 9; optimum alpha = 0.55; minimum error = 0.00121	no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes
100	digit = 0; optimum alpha = 0.55; minimum error = 0.00399 digit = 1; optimum alpha = 0.56; minimum error = 0.00393 digit = 2; optimum alpha = 0.55; minimum error = 0.00405 digit = 3; optimum alpha = 0.55; minimum error = 0.00421 digit = 4; optimum alpha = 0.55; minimum error = 0.00409 digit = 5; optimum alpha = 0.55; minimum error = 0.00398 digit = 6; optimum alpha = 0.55; minimum error = 0.00399 digit = 7; optimum alpha = 0.55; minimum error = 0.00389 digit = 8; optimum alpha = 0.55; minimum error = 0.00423 digit = 9; optimum alpha = 0.55; minimum error = 0.00412	digit = 0; optimum alpha = 0.55; minimum error = 0.00178 digit = 1; optimum alpha = 0.55; minimum error = 0.00737 digit = 2; optimum alpha = 0.55; minimum error = 0.00423 digit = 3; optimum alpha = 0.55; minimum error = 0.00197 digit = 4; optimum alpha = 0.55; minimum error = 0.00384 digit = 5; optimum alpha = 0.55; minimum error = 0.00204 digit = 6; optimum alpha = 0.55; minimum error = 0.00272 digit = 7; optimum alpha = 0.55; minimum error = 0.0058 digit = 8; optimum alpha = 0.55; minimum error = 5e-05 digit = 9; optimum alpha = 0.55; minimum error = 0.00659
10	digit = 0; optimum alpha = 0.53; minimum error = 0.01586 digit = 1; optimum alpha = 0.53; minimum error = 0.01628 digit = 2; optimum alpha = 0.53; minimum error = 0.01616 digit = 3; optimum alpha = 0.53; minimum error = 0.01689 digit = 4; optimum alpha = 0.53; minimum error = 0.01644 digit = 5; optimum alpha = 0.53; minimum error = 0.0158 digit = 6; optimum alpha = 0.53; minimum error = 0.01614 digit = 7; optimum alpha = 0.53; minimum error = 0.01548 digit = 8; optimum alpha = 0.53; minimum error = 0.01701 digit = 9; optimum alpha = 0.53; minimum error = 0.01654	digit = 0; optimum alpha = 0.53; minimum error = 0.02215 digit = 1; optimum alpha = 0.53; minimum error = 0.02354 digit = 2; optimum alpha = 0.53; minimum error = 0.02792 digit = 3; optimum alpha = 0.53; minimum error = 0.0278 digit = 4; optimum alpha = 0.58; minimum error = 0.00855 digit = 5; optimum alpha = 0.53; minimum error = 0.0289 digit = 6; optimum alpha = 0.53; minimum error = 0.02654 digit = 7; optimum alpha = 0.58; minimum error = 0.01707 digit = 8; optimum alpha = 0.58; minimum error = 0.01077 digit = 9; optimum alpha = 0.53; minimum error = 0.01835
5	digit = 0; optimum alpha = 0.56; minimum error = 0.0242 digit = 1; optimum alpha = 0.56; minimum error = 0.02507 digit = 2; optimum alpha = 0.56; minimum error = 0.02463 digit = 3; optimum alpha = 0.56; minimum error = 0.02566 digit = 4; optimum alpha = 0.56; minimum error = 0.0251 digit = 5; optimum alpha = 0.56; minimum error = 0.02409 digit = 6; optimum alpha = 0.56; minimum error = 0.02459 digit = 7; optimum alpha = 0.56; minimum error = 0.0236 digit = 8; optimum alpha = 0.56; minimum error = 0.02595 digit = 9; optimum alpha = 0.56; minimum error = 0.02538	digit = 0; optimum alpha = 0.56; minimum error = 0.05368 digit = 1; optimum alpha = 0.56; minimum error = 0.03204 digit = 2; optimum alpha = 0.56; minimum error = 0.04235 digit = 3; optimum alpha = 0.56; minimum error = 0.05658 digit = 4; optimum alpha = 0.56; minimum error = 0.01315 digit = 5; optimum alpha = 0.56; minimum error = 0.0655 digit = 6; optimum alpha = 0.56; minimum error = 0.0539 digit = 7; optimum alpha = 0.56; minimum error = 0.0281 digit = 8; optimum alpha = 0.56; minimum error = 0.04212 digit = 9; optimum alpha = 0.56; minimum error = 0.03111

Figure 8. Dependence of the optimal learning rate and the minimum learning error on the iteration numbers for two different activation functions.

Number of hidden layers	Sigmoid C=1, 10 iterations	ReLU C=0.1, 10 iterations
3	digit = 0; optimum alpha = 0.53; minimum error = 0.01586 digit = 1; optimum alpha = 0.53; minimum error = 0.01628 digit = 2; optimum alpha = 0.53; minimum error = 0.01616 digit = 3; optimum alpha = 0.53; minimum error = 0.01689 digit = 4; optimum alpha = 0.53; minimum error = 0.01644 digit = 5; optimum alpha = 0.53; minimum error = 0.0158 digit = 6; optimum alpha = 0.53; minimum error = 0.01614 digit = 7; optimum alpha = 0.53; minimum error = 0.01548 digit = 8; optimum alpha = 0.53; minimum error = 0.01701 digit = 9; optimum alpha = 0.53; minimum error = 0.01654	digit = 0; optimum alpha = 0.53; minimum error = 0.02215 digit = 1; optimum alpha = 0.53; minimum error = 0.02354 digit = 2; optimum alpha = 0.53; minimum error = 0.02792 digit = 3; optimum alpha = 0.53; minimum error = 0.0278 digit = 4; optimum alpha = 0.58; minimum error = 0.00855 digit = 5; optimum alpha = 0.53; minimum error = 0.0289 digit = 6; optimum alpha = 0.53; minimum error = 0.02654 digit = 7; optimum alpha = 0.58; minimum error = 0.01707 digit = 8; optimum alpha = 0.580; minimum error = 0.01077 digit = 9; optimum alpha = 0.53; minimum error = 0.01835
5	digit = 0; optimum alpha = 0.54; minimum error = 0.01562 digit = 1; optimum alpha = 0.57; minimum error = 0.0153 digit = 2; optimum alpha = 0.54; minimum error = 0.01563 digit = 3; optimum alpha = 0.54; minimum error = 0.01629 digit = 4; optimum alpha = 0.54; minimum error = 0.01623 digit = 5; optimum alpha = 0.54; minimum error = 0.01566 digit = 6; optimum alpha = 0.54; minimum error = 0.01604 digit = 7; optimum alpha = 0.54; minimum error = 0.01542 digit = 8; optimum alpha = 0.54; minimum error = 0.01573 digit = 9; optimum alpha = 0.54; minimum error = 0.01662	digit = 0; optimum alpha = 0.54; minimum error = 0.06366 digit = 1; optimum alpha = 0.57; minimum error = 0.05063 digit = 2; optimum alpha = 0.54; minimum error = 0.04663 digit = 3; optimum alpha = 0.54; minimum error = 0.08859 digit = 4; optimum alpha = 0.57; minimum error = 0.04265 digit = 5; optimum alpha = 0.54; minimum error = 0.08635 digit = 6; optimum alpha = 0.54; minimum error = 0.06389 digit = 7; optimum alpha = 0.54; minimum error = 0.03489 digit = 8; optimum alpha = 0.54; minimum error = 0.07446 digit = 9; optimum alpha = 0.54; minimum error = 0.04216
10	digit = 0; optimum alpha = 0.56; minimum error = 0.02088 digit = 1; optimum alpha = 0.56; minimum error = 0.02034 digit = 2; optimum alpha = 0.56; minimum error = 0.02741 digit = 3; optimum alpha = 0.56; minimum error = 0.06126 no learning processes digit = 5; optimum alpha = 0.56; minimum error = 0.01843 digit = 6; optimum alpha = 0.56; minimum error = 0.01904 digit = 7; optimum alpha = 0.56; minimum error = 0.03639 digit = 8; optimum alpha = 0.56; minimum error = 0.02022 digit = 9; optimum alpha = 0.56; minimum error = 0.02688	no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes no learning processes

Figure 9. Dependence of the optimal learning rate and the minimum learning error on the hidden layers number of the neural network for two different activation functions.

To clarify the reason for such a difference, let us consider branching diagrams and Fourier spectra for small iteration numbers values (less than 7). In other words, we will consider the logistic function value behavior from the value of the learning rate alpha. For this purpose, let us consider a neural network containing 3 hidden layers with 28 neurons per layer and the learning process for each activation function at 3 and 4, and 5 iterations (Fig. 10-Fig. 12). The C parameter for each function was chosen according to the previously obtained results, namely the values that corresponded to the best neural network learning (activation function: Sigmoid - C=1.0, hyperbolic tangent - C=1.0, ReLU - C=0.1).

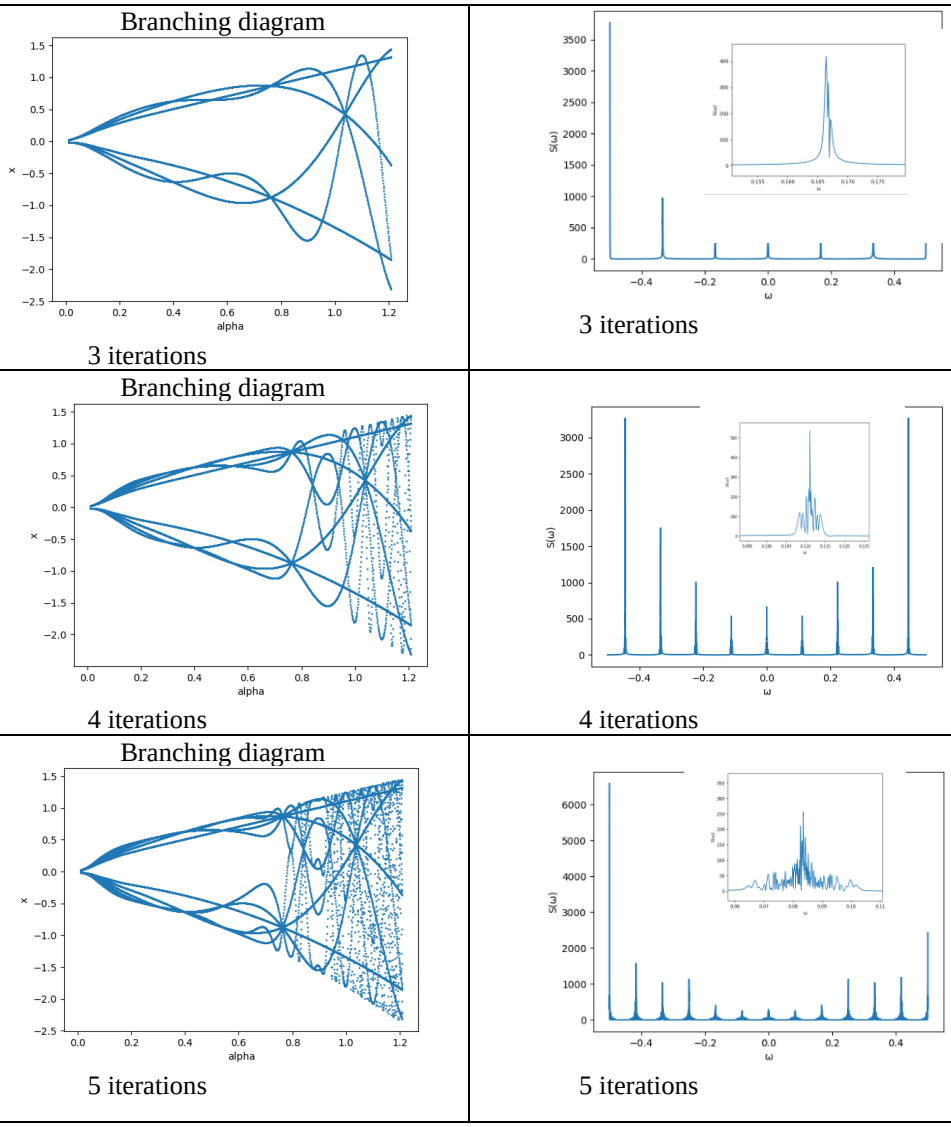
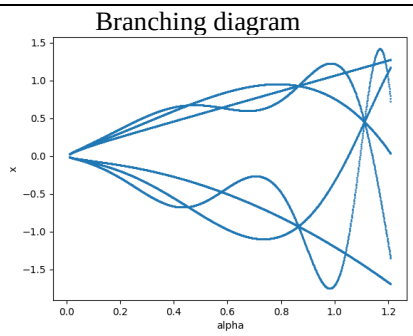
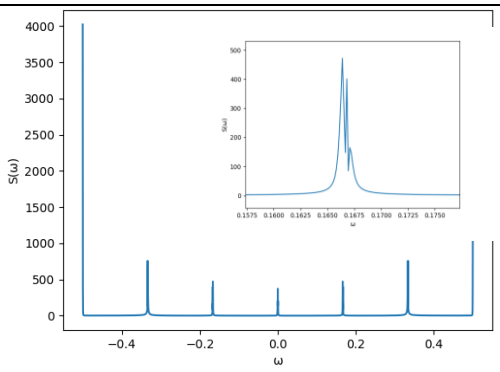


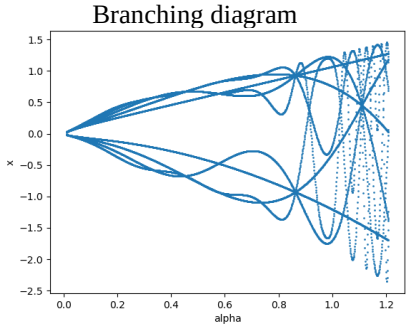
Figure 10. Branching diagrams and Fourier spectra (close-up inset spectrum of one band) for the printed digit "0" given by an array of 4x7 zeros and ones, assuming three hidden layers with 28 neurons per layer, for ReLU activation function with a C=0.1 parameter, on the iterations number.



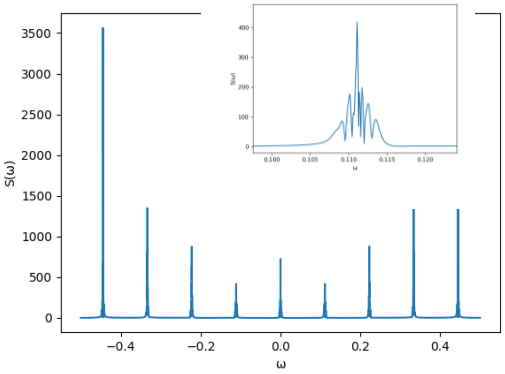
3 iterations



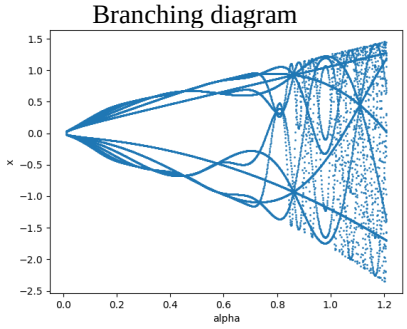
3 iterations



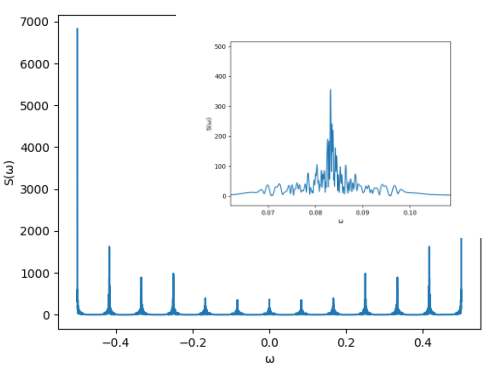
4 iterations



4 iterations



5 iterations



5 iterations

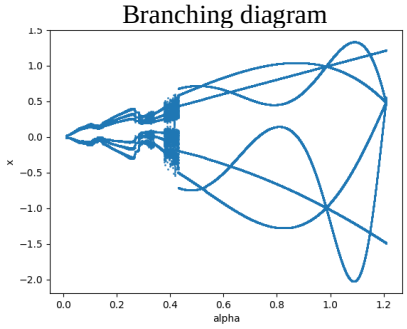
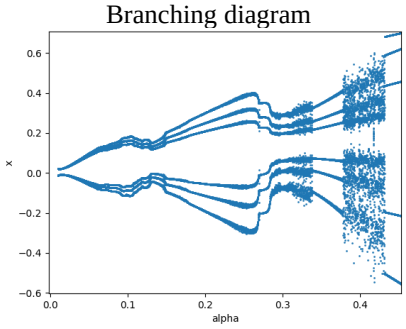
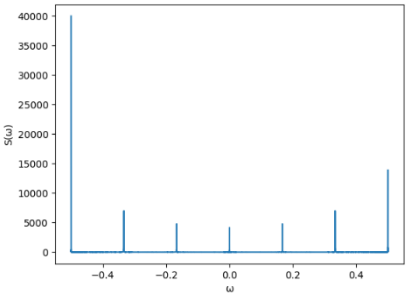
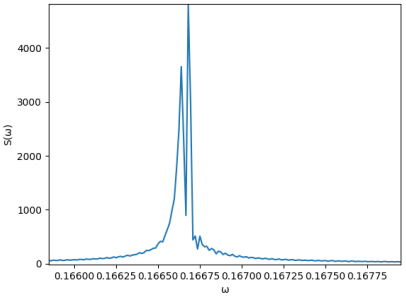
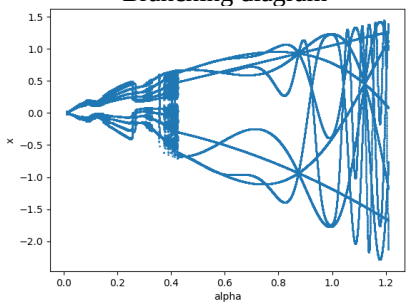
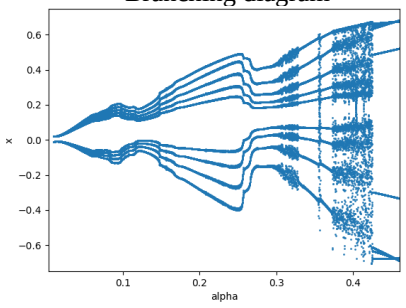
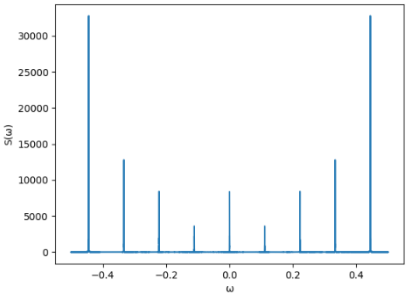
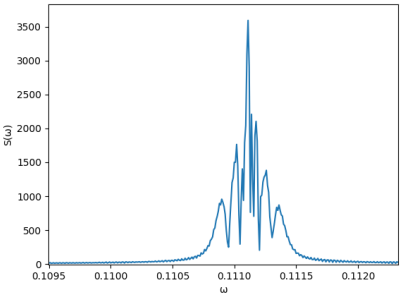
Figure 11. Branching diagrams and Fourier spectra (close-up inset spectrum of one band) for the printed digit "0" given by an array of 4x7 zeros and ones, assuming three-hidden layers with 28 neurons per layer, for a sigmoidal activation function with a *C*=1 parameter, from the iterations number

According to Fig. 10, for the ReLU activation function at 3 iterations, the branching diagram shows a split characterized by the existence of three curves. The first curve is a monotonically descending dependence; the second curve is a monotonically descending dependence with the one extremum point existence. The third curve is monotonic, and the function describing it is a periodic function (Fig. 10). The Fourier spectra of this network display a characteristic behavior, namely the existence of an appropriate number of spectral bands and harmonics (Fig. 10). Increasing by one the iterations number (5 iterations) entails the appearance of the fourth curve, which is characterized by a clearly defined periodicity with different periods. It is also displayed on the Fourier spectra, resulting in the appearance of an additional band, and all bands are characterized by the appearance of additional harmonics.

As can be easily seen, their number is equal to the sum of all existing curve harmonics (iterations) (Fig. 10). By increasing the iterations number by one more (5 iterations), an additional curve emerges, and the number of harmonics increases (Fig. 10). The curves obtained this way are forming a branching diagram in the final case. Consequently, an increase in the number of iterations leads to the appearance of curves described by periodic dependence of the logistic function value on the learning rate, and this periodicity changes both when passing from one curve to another and when changing the learning rate. It should be noted that there may be a case when, in a certain range of learning rate values, a multiplicity of existing periodic dependences (of the logistic function from the learning rate) will be observed. In the authors' opinion, this feature leads to the appearance of transparency regions, i.e., to a decrease in the harmonic's number. Such a decrease in the number of harmonics is accompanied by an increase in the power spectrum value of the basic harmonics, as indicated by the Fourier spectra. It, in turn, leads to a decrease in the value of the minimum learning error.

When applying the sigmoidal activation function to the learning process, we obtained similar dependences (Fig. 11) as for the ReLU function. We also observed the emergence of periodic dependences with a variable period on the learning rate, as well as the presence of three-node points where all the existing curves intersect. The difference in these two activation functions' applications is evident only when the number of iterations is ≥ 5, namely, in an increase in the power spectrum value of the basic harmonics. It leads, as noted above, to a better learning process (lower learning error) of the neural network.

Contrary to the previous two activation functions, the hyperbolic tangent activation function leads to two clearly expressed chaotic neural network states in the range of learning rate values alpha $\approx$ 0.3-0.43 (Fig. 12). At alpha >0.45, the characteristic behavior of the three curves is observed, as for the activation functions given above. An increase in the Iterations number ("curves") leads to a decrease in the range of chaotic state existence and, like the mentioned above activation functions, also to the appearance of curves described by periodic functions with a variable period (Fig. 12). The increase in the hidden layers number and the decrease in the *C* parameter value for a given activation function led to a chaotic state disappearance process for the neural network.

Branching diagram  3 iterations	Branching diagram  3 iterations
 3 iterations	 3 iterations
Branching diagram  4 iterations	Branching diagram  4 iterations
 4 iterations	 4 iterations

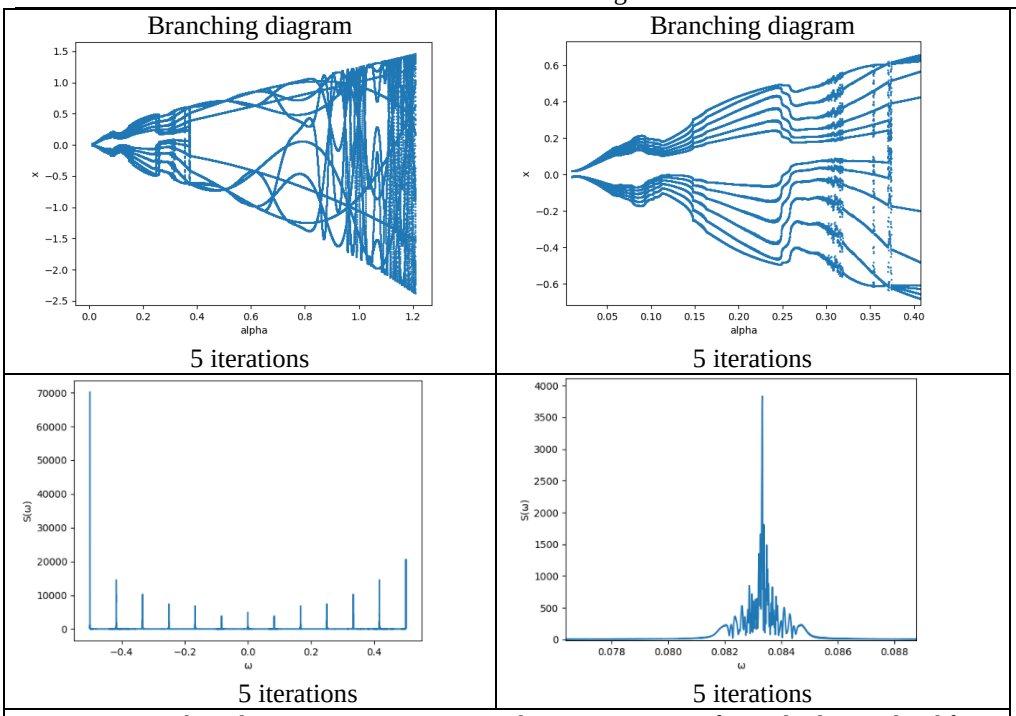


Figure 12. Branching diagrams, Fourier spectra, and Fourier spectrum of a single close-up band for a printed digit "0" given by an array of 4x7 zeros and ones, assuming three-hidden layers with 28 neurons per layer, for a hyperbolic tangent activation function with a *C*=1 parameter, from the iterations number.

4. Conclusions

To summarize all the above, we can say that the sigmoidal activation function has the best effect on the training of a three-layer neural network for the identification of digits defined by an array of 4x7 zeros and ones. Compared to other activation functions, there is an insignificant change in the learning error for it during the transition from one digit to another. The increase in the number of hidden layers does not lead to a sharp increase in the learning error. An increase in the number of learning iterations is followed by the appearance of periodic dependences of the logistic value of the learning rate function, the period of which is a variable of the iterations number and the learning rate. Based on the Fourier spectra of the error function on the learning rate value, it can be argued that an increase in the iterations number leads to an increase in the number of the harmonic, which eventually leads to the chaotic state of the neural network. The reasons for the chaotic state appearance depend on the value of the logistic function from the learning rate, at values alpha<0.5 due to the hyperbolic tangent activation function will be discussed in the following paper.

REFERENCES

[1] *S. Mogilnyj* Machine learning with the use of microcomputers: teaching book editors: O. Lisovyj and others. - K., 2019. – 226 p.
[2] *I. Goodfellow, Y. Bengio, A. Courville*. Deep Learning, 2016 URL: http://www.deeplearningbook.org
[3] *S. Sveleba, I. Katerynchuk, I. Kuno, I. Karpa, O. Semotyuk, Ya. Shmygelsky, N. Sveleba, V. Kuno*. Chaotic states of a multilayer neural network, Electronics and information technologies. 2021. Issue 16. P. 20–35.
[4] *X.-S. Wei*. Must Know Tips/Tricks in Deep Neural Networks URL: http://www.lamda.nju.edu.cn/weixs/project/CNNTricks/CNNTricks.html
[5] *J. Brownlee*. A Gentle Introduction to the Rectified Linear Unit (ReLU) URL: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
[6] *M. Nielsen* Neural Networks and Deep Learning Chapter 1: Using neural nets to recognize handwritten digits URL: http://neuralnetworksanddeeplearning.com/chap1.html
[7] *Yu. Olenych, S. Sveleba, I. Katerynchuk, I. Kunio, I. Karpa*. Features of deep studyneural network. 2019. URL: https://openreviewhub.org/lea/paper-2019/features-deep-study-neural-network#
[8] *Yu. Taranenko* Information entropy of chaos URL: https://habr.com/ru/post/447874/

ОСОБЛИВОСТІ ЗАЛЕЖНОСТІ ПОХИБКИ НАВЧАННЯ БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ ВІД ФУНКЦІЇ АКТИВАЦІЇ В ПРОЦЕСІ РОЗПІЗНАВАННЯ ДРУКОВАНИХ ЦИФР

С. Свелеба[1], І. Катеринчук[1], І. Куньо[1], О. Семотюк[2], Я. Шмигельський[1] та Н. Свелеба[1]

[1]Львівський національний університет імені Івана Франка,
вул. Ген. Тарнавського, 107, 79017 Львів, Україна

[2]Українська Академія Друкарства
вул. Під Голоском,19, 79020 Львів, Україна

В роботі проведено дослідження процесу навчання при розпізнаванні друкованих цифр залежно від виду функції активації. Дослідження впливу виду функції активації, кількість ітерацій на процес навчання нейронної системи здійснювався з використанням аналізу Фур'є спектрів функції похибки навчання та діаграм розгалуження. З цією метою в середовищі Phyton розроблена програма для багатошарової нейронної мережі, яка передбачає задання кількості прихованих шарів і кількості нейронів в них, та швидкості навчання Швидкість навчання розглядалась, як постійна величина і визначалось її оптимальне значення, при якому спостерігається найкраще навчання. Для аналізу впливу швидкості навчання на процес навчання, використовувалась логістична функція, яка описує процес подвоєння ча-

стоти. Показано, що функція похибки навчання характеризується біфуркаційними процесами, які приводять до хаотичного стану при η>0,8. Визначено оптимальне значення швидкості навчання, яке визначає появу процесу подвоєння кількості локальних мінімумів. Встановлено, що сигмоїдальна функція активації (в порівнянні з функціями активації ReLU та гіперболічний тангенс) найкраще задовольняє процес навчання трьох шарової нейронної мережі для розпізнавання цифр заданих масивом 4х7 нулів і одиниць. Порівняно з іншими функціями активації для неї спостерігається незначна зміна похибки навчання при переході від однієї цифри до іншої. Показано, що збільшення кількості прихованих шарів не приводить до різкого збільшення похибки навчання. Збільшення кількості ітерацій навчання супроводжується появою періодичних залежностей величини логістичної функції від швидкості навчання, період яких є змінною величиною від кількості ітерацій та швидкості навчання. Використовуючи Фур'є спектри функції похибки від величини швидкості навчання, можна стверджувати, що збільшення кількості ітерацій спричиняє збільшення кількості гармонік, які в кінцевому випадку приводять до появи хаотичного стану нейронної мережі.

Ключові слова: Багатошарова нейронна мережа, функція активації, оптимальна швидкість навчання, розпізнавання цифр.

Стаття: надійшла до редакції 02.05.2022,
прийнята до друку 05.05.2022