

ХАОТИЧНІ СТАНИ БАГАТОШАРОВОЇ НЕЙРОННОЇ МЕРЕЖІ

С. Свелеба¹, І. Катеринчук¹, І. Куньо¹, І. Карпа¹, О. Семотюк²,
Я. Шмигельський¹, Н. Свелеба¹, В. Куньо³

¹Львівський національний університет імені Івана Франка,
вул. Ген. Тарнавського, 107, 79017 Львів, Україна
incomlviv@gmail.com

²Українська академія друкарства,
вул. Підголоско, 19, 79020 Львів, Україна

³Львівський інститут економіки і туризму,
вул. Менцинського, 8, 79007 Львів, Україна

⁴Національний університет «Львівська політехніка»,
вул. С. Бандери, 12, 79013 Львів, Україна

В роботі проведено дослідження впливу швидкості навчання η на процес навчання багатошарової нейронної мережі. Програма для багатошарової нейронної мережі була написана на мові Python. 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 частоти. Показано, що функція похибки навчання характеризується біфуркаційними процесами, які приводять до хаотичного стану при $\eta > 0,8$. Визначено оптимальне значення швидкості навчання, яке визначає появу процесу подвоєння кількості локальних мінімумів. Збільшення кількості прихованих шарів та кількості нейронів в кожному шарі не приводить до кардинальної зміни діаграми логістичної функції, а отже і оптимальної величини швидкості навчання. Показано, що збільшення кількості прихованих шарів, як і кількості нейронів в них, при збільшенні швидкості навчання супроводжується появою хаотичного стану, який характеризується відсутністю процесу навчання. Використовуючи Фур'є розклад функції похибки, в роботі досліджено процес біфуркації в залежності від швидкості навчання, і на базі цих досліджень розроблена програма для визначення величини оптимальної швидкості навчання нейронної мережі.

Ключові слова: багатошарова нейронна мережа, оптимальна швидкість навчання, біфуркації, хаотичні стани.

Зростання об'єму інформації, що обробляється, спричинює застосування багатошарових нейронних мереж (НМ) та згорткових НМ. Нейронні мережі з глибинним навчанням все більше використовують для систем розпізнавання образів. Оскільки більшість задач, на виконання яких направлені НМ, є задачі на прийняття рішень, то будемо розглядати НМ, в яких використовуються метод навчання з вчителем зі зворотнім поширенням похибки. Даний метод пов'язаний з підбором ваг, які на початку навчання задаються рандомно. В даній праці ми будемо розглядати тільки багатошарові НМ. В багатошарових НМ корекція ваг здійснюється в кожному прихованому шарі від

однієї епохи до іншої поки не буде досягнуто заданої точності навчання. Процес навчання визначається рядом параметрів – швидкість навчання, кількість епох, кількість вхідних і вихідних величин тощо. Збільшення кількості вхідних і вихідних величин спричиняє зростання часу навчання та прийняття рішення. Значний термін навчання, а отже і прийняття рішення, є однією з негативних ознак НМ. Зменшення часу навчання може бути досягнуто шляхом зменшення кількості епох необхідних для навчання. Одним із методів зменшення кількості ітерацій, необхідних для досягнення заданої точності, є підвищення швидкості навчання. Відомо [1-3] кілька ефективних алгоритмів підбору швидкості навчання для оптимізації процесу навчання. Це залежність швидкості навчання η від кількості епох [1]. В більшості випадків величина η або покладається постійною, і тоді звичайно послідовність значень ваг w_k сходиться в окіл оптимального значення w , або вона є спадаючою функцією часу так, як це робиться в стохастичних методах оптимізації та адаптації [2]. Дана процедура виконується доти, поки значення керованих змінних не стабілізуються, або поки помилка не зменшиться до прийняттого значення. Слід зазначити, що дана процедура характеризується повільною швидкістю збіжності і може потрапляти в локальні мінімуми цільового функціонала [2]. Це і залежність швидкості навчання від часу навчання [3]. Це і множення параметра навчання на деякий коефіцієнт залежний від кількості вхідних параметрів.

При підході до глобального мінімуму простежується збільшення локальних мінімумів, які впливають на алгоритм навчання НМ. Зростання локальних мінімумів при підході до глобального мінімуму зумовлює розглядати НМ, як систему в якій появляються невизначені стани (хаотичні стани). Величини локальних і глобального мінімумів стають тотожними, і кількість таких локальних мінімумів зростає. Тобто в першому наближенні НМ в околі глобального мінімуму, можна розглядати як систему що характеризується невизначеністю. Такий режим системи є стохастичним з появою стаціонарних та хаотичних розв'язків. Даний режим характерний, як для одношарових, так і для багатошарових НМ. Відомо [4], що дієвим методом уникнення локальних мінімумів є підбір параметра навчання (швидкості навчання) для таких систем, та зменшення кількості необхідних епох для навчання.

Кожний шар даної НМ будемо розглядати як окрему детерміновану систему, для якої дослідимо діаграму розгалуження з використанням функції відображення виду:

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

де n – крок, η – параметр, який визначає швидкість навчання.

Її нерухомі точки:

$$x_{1,2}^* = -1 \pm \sqrt{1 + \eta},$$

а мультиплікатори:

$$\rho_{1,2} = 1 \mp 2\sqrt{1 + \eta}.$$

Вибір даного логістичного відображення обумовлений тим, що воно описує процес подвоєння частоти коливань [6]. В нашому випадку цей процес зумовлений виникненням локальних мінімумів при підході до глобального мінімуму.

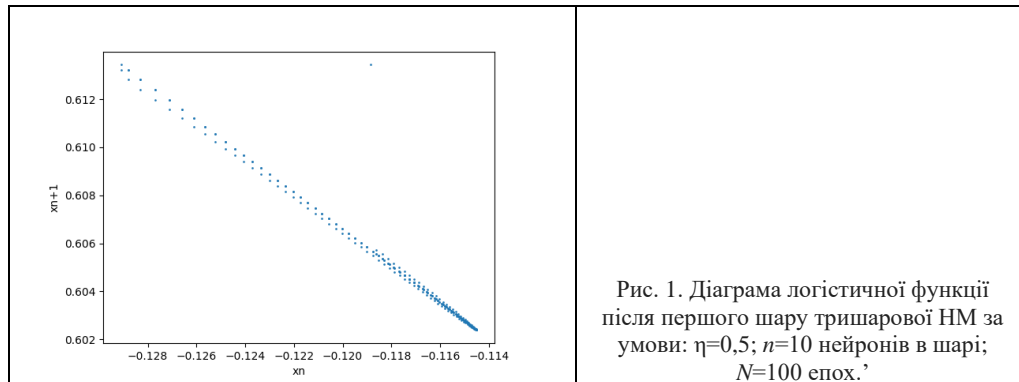
Завданням даної роботи є встановлення алгоритму автоматичного підбору швидкості навчання для НМ. З цією метою в середовищі Python розроблена програма для багатошарової нейронної мережі, яка передбачає задання кількості прихованих

шарів і кількості нейронів в них, та зміною параметра навчання в інтервалі $0,001 \div 10$. Цей інтервал зміни η вибирався з урахування даних роботи [5], де було проведено дослідження впливу швидкості навчання на процес навчання в багатошарових НМ. У додатку наведено код програми для багатошарової НМ з можливістю побудови діаграми розгалуження.

Для одновимірних відображень існує два способи зміни стійкості нерухомої точки, коли мультиплікатор точки $\rho = +1$ і $\rho = -1$. Однак число пов'язаних з цим біфуркацій (подвоєнь) помітно більше. Це пояснюється тим, що в них часто бере участь більше однієї нерухомої точки. Такій ситуації відповідають 4 варіанти біфуркацій: дотична біфуркація (складка, сідло-вузлова); транскритична біфуркація; виллоподібна біфуркація (біфуркація втрати симетрії); біфуркація подвоєння.

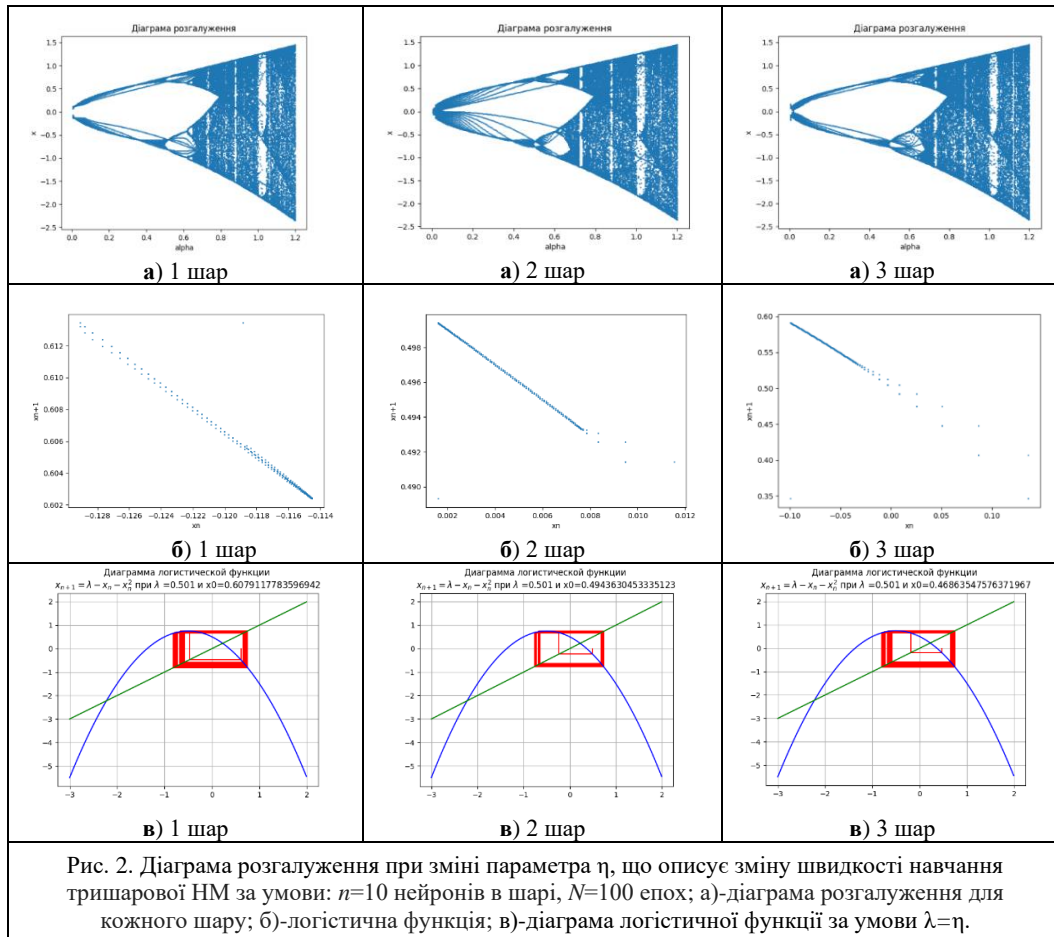
Входження системи в непередбачуваний режим описується каскадом біфуркацій, які йдуть одна за одною. Каскад біфуркацій веде послідовно до появи вибору між двома рішеннями, потім чотирма і так далі, система починає коливатися в хаотичному, турбулентному режимі послідовного подвоєння кількості можливих значень. Будемо розглядати біфуркації подвоєння кількості локальних мінімумів та появу хаосу в точкових відображеннях. Відображення - це функція, яка показує залежність наступних значень параметрів системи від попередніх значень. При зміні зовнішнього параметра η точкові відображення демонструють досить складну поведінку, яка стає хаотичною при досить великих η .

Розглянемо спочатку тришарову нейронну мережу і дослідимо її як детерміновану систему. На рис. 1 наведено діаграму логістичної функції при постійному значенні параметра η . Дана діаграма логістичної функції отримує свою адекватну поведінку лише за умови коли логістичне відображення задається ітераційною формулою для біфуркації подвоєння.



Перш за все зазначимо, що дана логістична функція найбільш адекватно описує дану систему і свідчить про те, що утворення локальних мінімумів можна в першому наближенні описати, як процес подвоєння.

Наведена на рис. 2 діаграма розгалуження для кожного шару тришарової НМ засвідчує, що кожний шар можна розглядати, як окрему систему для якої характерне існування хаотичних і стаціонарних розв'язків.

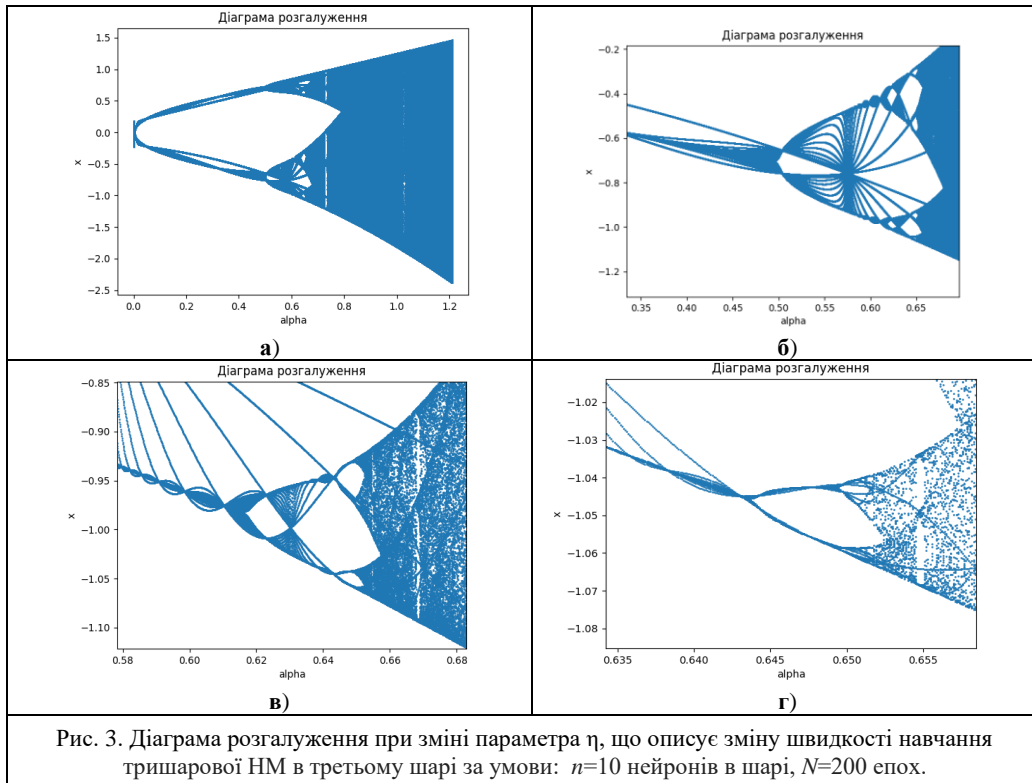


В залежності від значення параметра η відображення має різну кількість нерухомих точок. При $0 < \eta < 0,4$ число нерухомих точок не змінюється, але обидві нерухоми точки є нестабільними. Оскільки відображення обмежене, то відсутність точкового атрактора означає утворення складнішого атрактора типу граничного циклу (рис. 2,б). Хоча саме відображення не має стійких нерухомих точок, його квадрат такі стійкі нерухоми точки можливо має. Тому на біфуркаційній діаграмі в цій області простежується розгалуження ліній. При значеннях параметра $\eta > 0,4$ (рис. 2, а), граничний цикл втрачає стійкість. При таких значеннях параметра стійкі нерухоми точки потрібно шукати у відображеннях вищих порядків.

Ситуація з періодом граничного циклу відображень вищих порядків існує в певній області параметра η , а потім змінюється – відбувається збільшення локальних мінімумів, і так далі.

При значеннях параметру $\eta > 0,78$ (рис. 2,а) жоден із граничних циклів не має стабільності. З цього значення починається хаотична поведінка системи. Значення x_n змінюється неперіодично. Такий хаос називається детермінованим, оскільки існує чіткий строго визначений закон, за яким можна визначити значення змінної на будь-якій ітерації, починаючи від вибраного початкового значення, однак в її поведінці немає періодичності. Якщо в області стійкості граничних циклів поведінка відображень слабо залежала від початкової точки, то в області хаосу мала зміна початкового значення призводить до значної зміни значення n -тої ітерації [7].

В області хаосу існують несподівані вікна, коли при певних значеннях параметру η поведінка відображення стає регулярною (рис. 2,а). Фрактальність даного відображення пов'язана з подібністю процесу подвоєння граничних циклів. Так на рис. 3,б-г наведено в збільшеному масштабі ділянки подвоєння граничних циклів. Згідно даної залежності цей процес проходить по єдиному сценарію.



На рис. рис. 2,б,в наведено діаграму логістичної функції при $\eta=0,5$ і $x_0 = 0,6$. Дане відображення характеризується появою стійкої точки x_1^* , яка відмінна від точки $x^* = 0$. В розглянутій тришаровій НМ в кожному шарі діаграма логістичної функції є подібною і засвідчує що точка $x^* = 0$ не є стійкою, а стійкою точкою є x_1^* .

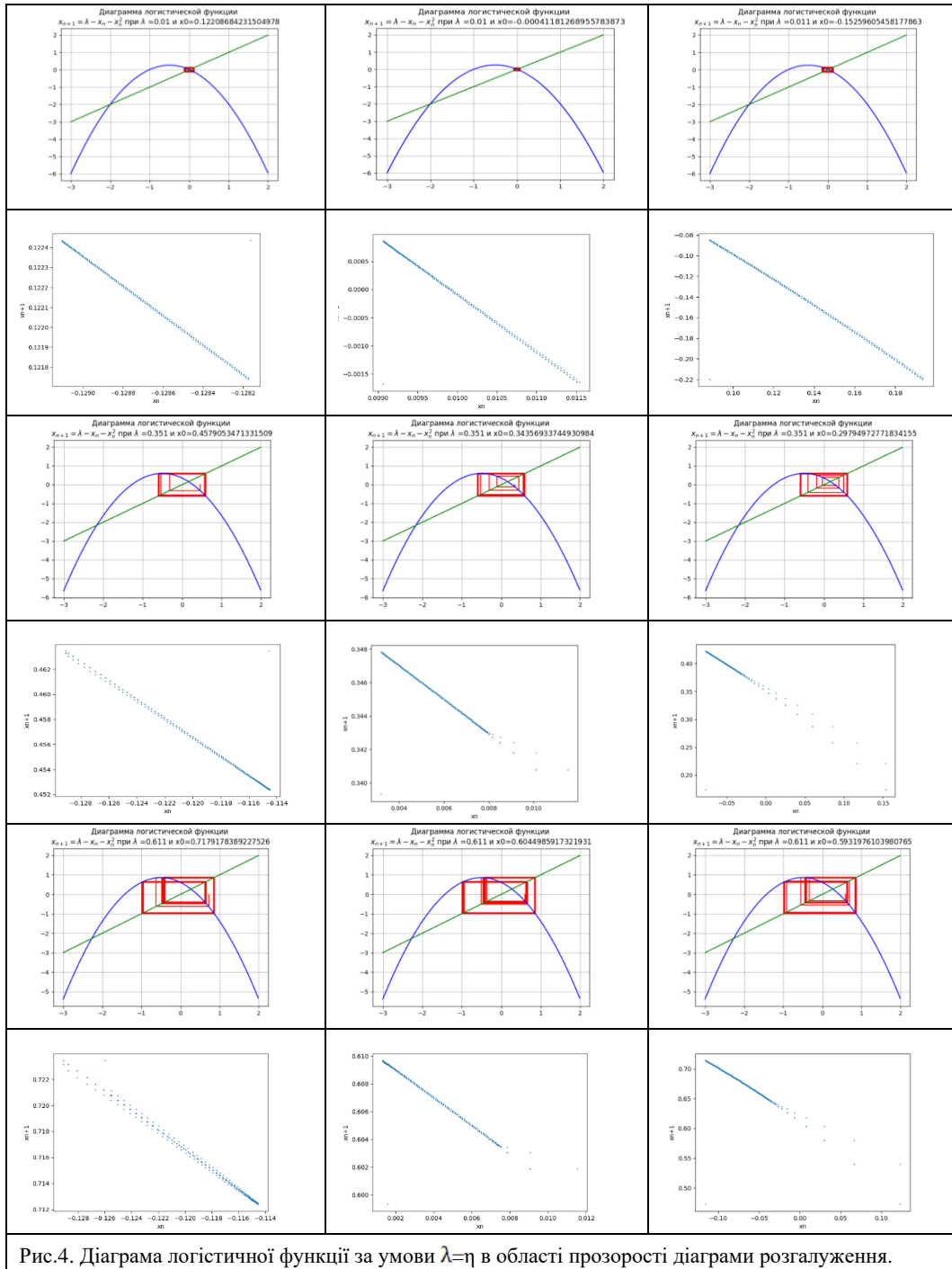


Рис.4. Діаграма логістичної функції за умови $\lambda = \eta$ в області прозорості діаграми розгалуження.

На рис. 4 наведено діаграми логістичної функції в області прозорості діаграми розгалуження. Згідно даних діаграм стійка точка $x^* = 0$ є відсутньою, що вказує на те що початково вже в околі глобального мінімуму існують кілька локальних мінімумів. Про це також засвідчують і діаграми розгалуження кожного прихованого шару. При $\eta=0,62$ починає відбуватись процес подвоєння кількості локальних мінімумів. Тобто при переході параметра η через значення $\eta=0,62$ 2-кратний цикл стає 4-кратним і так далі. При кінцевому значенні $\eta=1,2$ в системі є нестійкі цикли всіх можливих порядків (рис. 5). Отже в подальшому ($\eta>0,6$), внаслідок процесу подвоєння дана система переходить до хаосу (рис. 5). Така картина ідентична для кожного прихованого шару багатошарової НМ. Характерно, що для таких систем спостерігаються вікна прозорості, які в нашому випадку добре прослідковується при $\eta=1,03$ (рис. 4, рис. 1). Отже оптимальним значенням швидкості навчання для тришарової НМ з 10 нейронами в кожному шарі є значення $\eta \approx 0,6$. Збільшення кількості нейронів в кожному шарі в першому наближенні, згідно діаграм розгалуження (рис. 5) не приводить до суттєвих змін. А саме не змінюються ні вікно прозорості на діаграмі розгалуження, ні її загальна картина.

Виходячи із теорії зворотного поширення похибки, збільшення кількості нейронів в прихованому шарі, повинне стимулювати появу більш вираженого хаосу («розвинутого» хаосу). З цією метою було розглянуто характер поведінки логістичної функції при збільшенні кількості нейронів у прихованому шарі. На рис. 7 наведено вигляд логістичної функції для НМ з трьома прихованими шарами при різній кількості нейронів у прихованому шарі.

Згідно з рис. 7 збільшення кількості нейронів в шарі спричиняє збільшення хаотичності системи, тобто стимулює появу розвинутого хаотичного стану. Аналогічна картина спостерігається і при одночасному збільшенні як кількості нейронів у прихованому шарі так і кількості прихованих шарів.

Виходячи із характеру поведінки діаграми логістичної функції (рис. 7), поява хаотичності, можливо, обумовлена як збільшенням кількості локальних мінімумів (кількості рівноважних значень функції похибки), так і хаотичним значенням функції похибки при рандомному підборі ваг на початковому етапі навчання. Останній тип хаосу скоріш за все буде розвиватись при збільшенні кількості нейронів в прихованому шарі. На рис. 8 наведено залежність похибки навчання нейронної мережі від кількості епох та швидкості навчання. При збільшенні кількості нейронів в прихованому шарі простежується поява хаотичного стану, який починає зароджуватися при великих значеннях швидкості навчання і при малих значеннях кількості епох (рис. 8, в,г). Подальше збільшення кількості нейронів у прихованому шарі спричиняє збільшення області хаотичності на діаграмі залежності похибки навчання нейронної мережі від кількості епох і швидкості навчання. Даний хаотичний стан характеризується переходом в стан який відзначається відсутністю процесу навчання (рис. 8, д,є). Хаотичність даного стану визначається тим, що при незначній зміні вхідних параметрів (таких як кількість навчаючих пар, кількість нейронів в прихованому шарі) відбуваються значні зміни в області існування хаотичного стану на діаграмі залежності похибки навчання нейронної мережі від кількості епох та швидкості навчання (рис. 9).

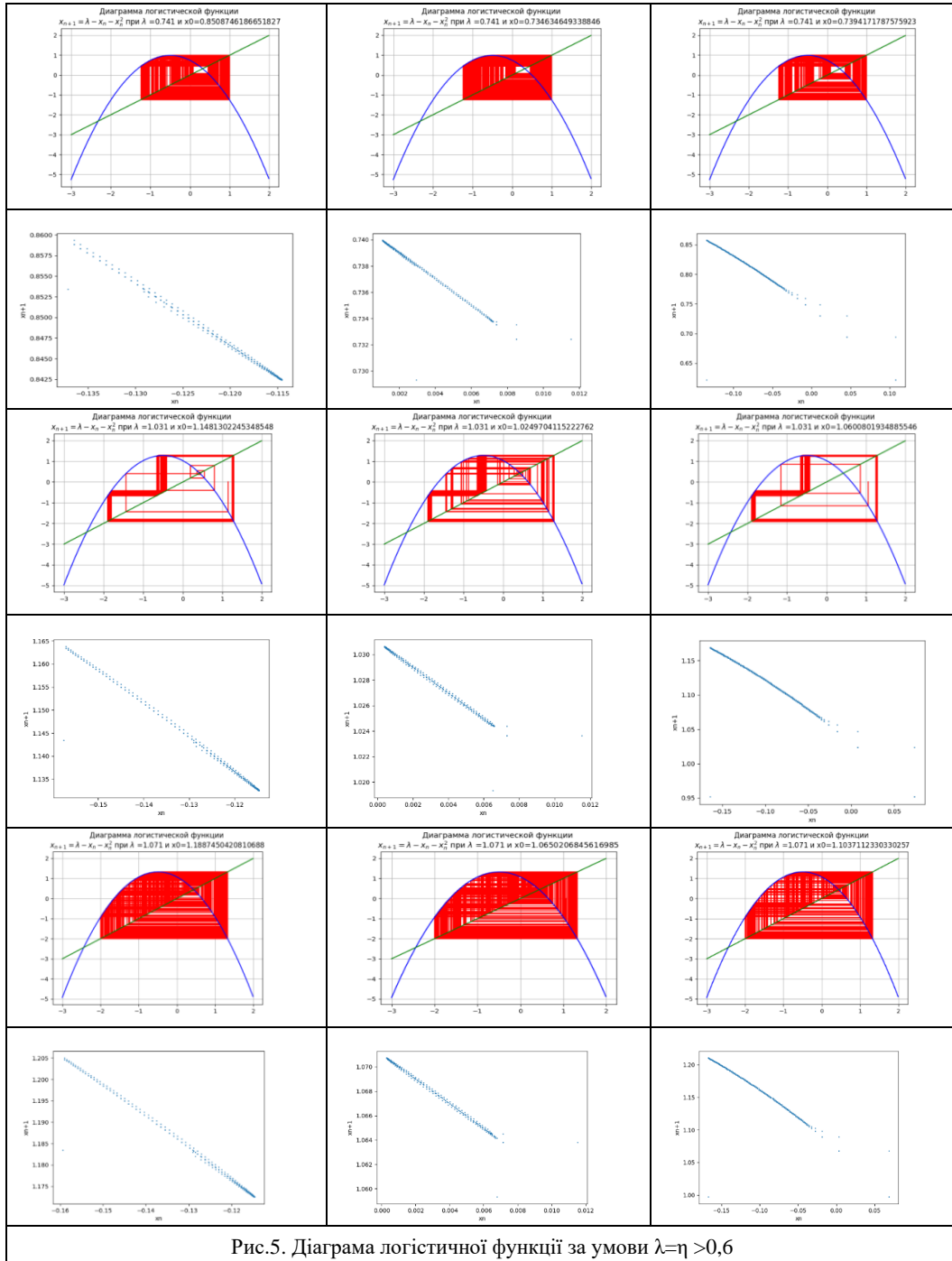
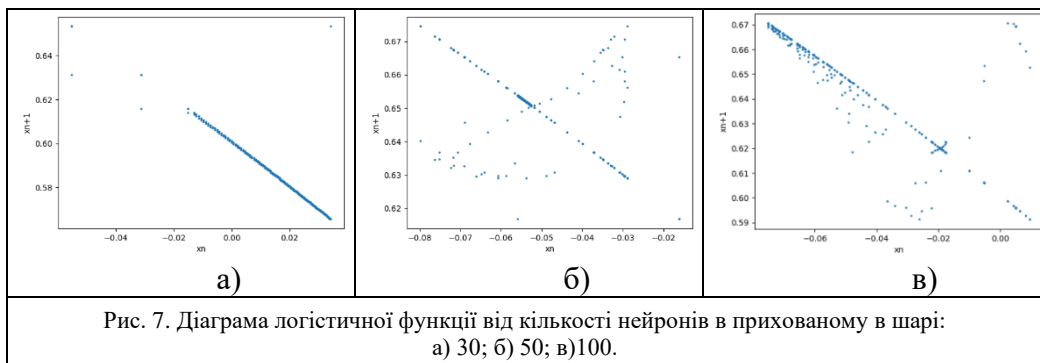
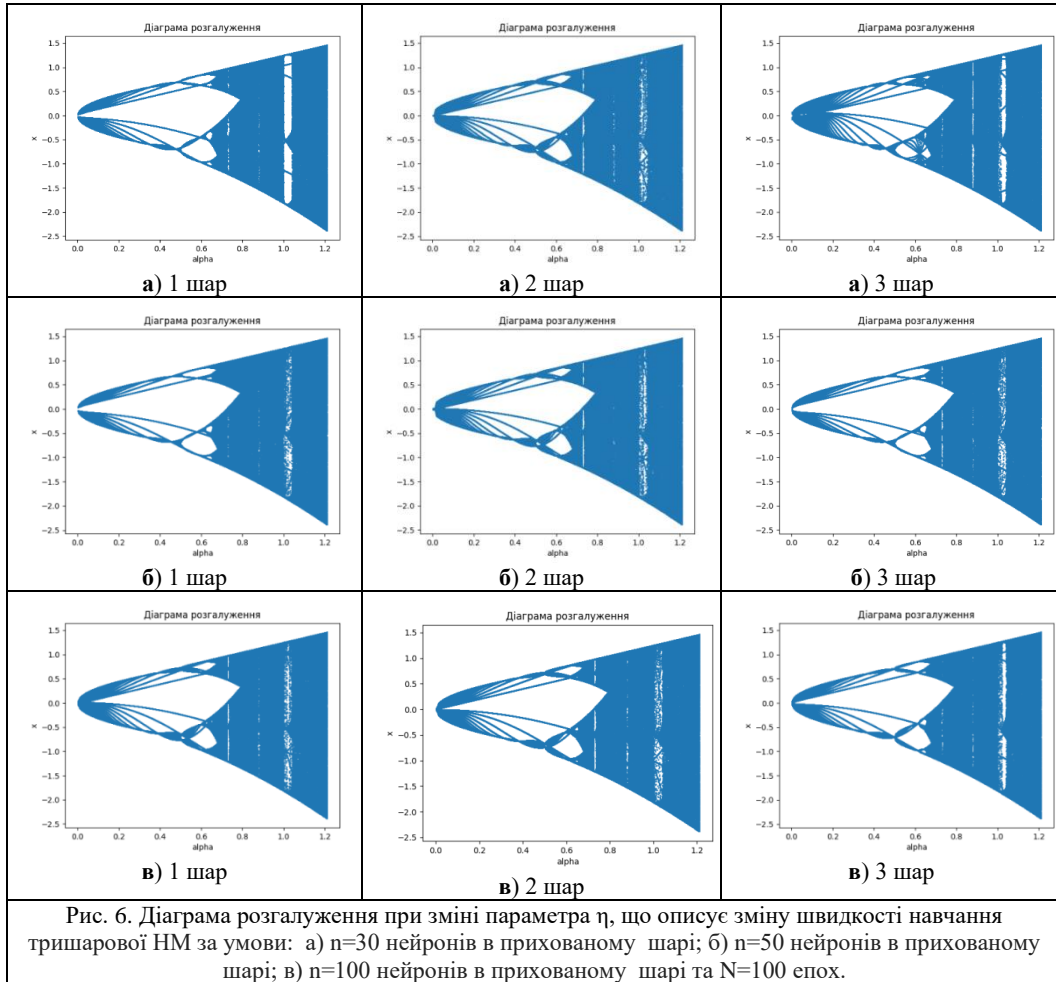


Рис.5. Діаграма логістичної функції за умови $\lambda = \eta > 0,6$



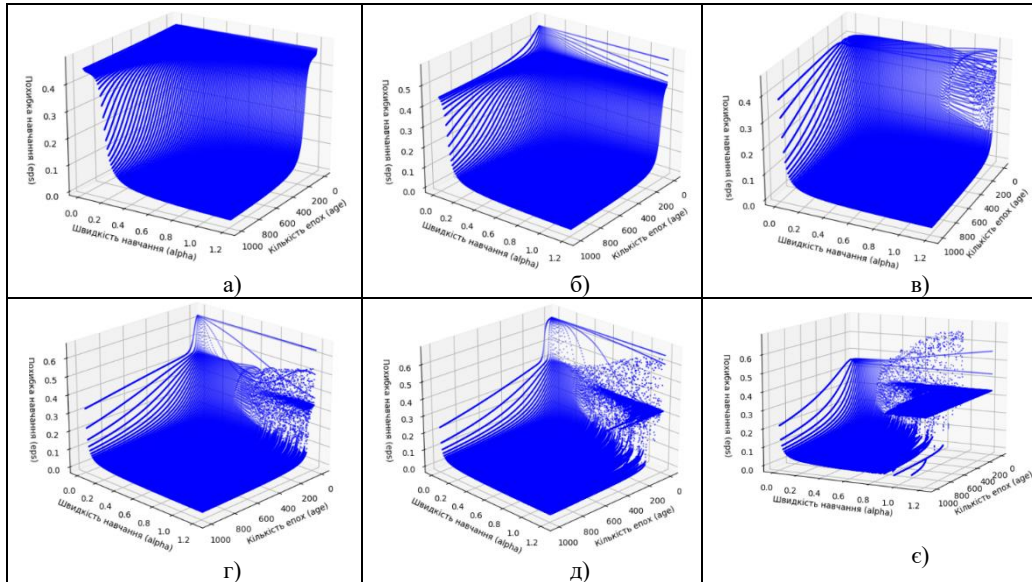


Рис. 8. Залежність похибки навчання нейронної мережі від кількості епох та швидкості навчання за умови: $n=3$ - приховані шари; а) 3 нейрони в прихованому шарі; б) 10 нейронів в прихованому шарі; в) 30 нейронів в прихованому шарі; г) 50 нейронів в прихованому шарі; д) 70 нейронів в прихованому шарі; є) 100 нейронів в прихованому шарі.

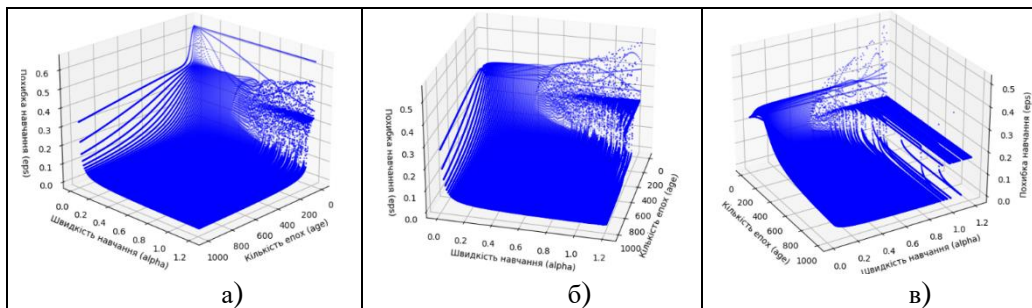
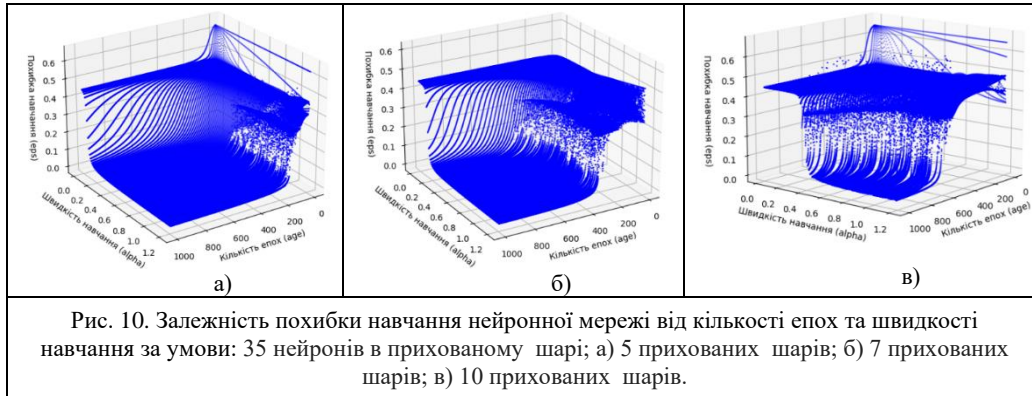


Рис. 9. Залежність похибки навчання нейронної мережі від кількості епох та швидкості навчання за умови: $n=3$ - прихованих шара; 50 нейронів в прихованому шарі; а) 3 навчальні пари; б) 4 навчальні пари; в) 6 навчальних пар.

Збільшення кількості прихованих шарів аналогічним чином стимулює збільшення хаотичного стану на діаграмі залежності похибки навчання нейронної мережі від кількості епох та швидкості навчання (рис. 10).

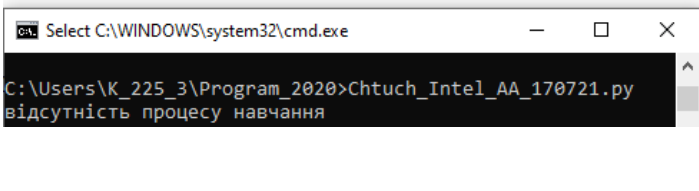
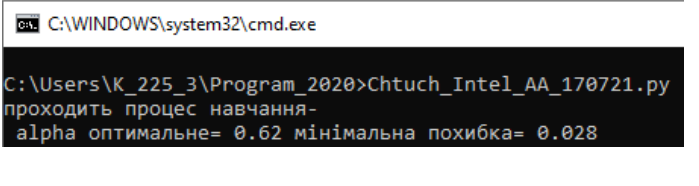


Отже в багатошаровій нейронній мережі можуть виникати хаотичні стани, які зумовлені як збільшенням швидкості навчання так і збільшенням кількості прихованих шарів і нейронів в них. Всі отримані хаотичні стани є чутливими до зміни параметрів НМ. Незначна зміна параметрів НМ спричиняє суттєві зміни стану НМ. Поява локальних розв'язків (локальних мінімумів) внаслідок збільшення швидкості навчання зумовлює появу біфуркацій залежності похибки навчання від кількості епох. Така динаміка величини похибки від кількості епох при постійній швидкості навчання приводить до хаотичного стану, і засвідчує, що отримати похибку нульової величини є неможливо. Тобто є граничне значення величини похибки яку можна досягнути. Виходячи із досліджень діаграми розгалуження дана величина похибки практично не залежить від конфігурації багатошарової НМ. Другий хаотичний стан, який виникає в наслідок збільшення, як кількості прихованих шарів так і кількості нейронів в них, спричиняє перехід НМ в стан який характеризується відсутністю процесу навчання. Особливо це проявляється при зміні кількості прихованих шарів.

Одним із способів вирішення даної проблеми (уникнення хаотичних станів НМ) є автоматичне визначення мінімальної кількості розв'язків на діаграмі логістичної функції, тобто визначення значення швидкості навчання, при якому проходить процес подвоєння цих розв'язків. Цей механізм передбачає відсутність хаотичного стану, який зумовлює появу стану, при якому відсутнє навчання НМ. Алгоритм вирішення даної задачі полягає у визначенні кількості розв'язків на діаграмі логістичної функції при даному значенні швидкості навчання, за якого проходить процес збільшення кількості біфуркацій в два або більше разів, і визначає оптимальне значення швидкості навчання та оптимальну величину похибки. В разі появи хаосу, зумовленого збільшенням кількості прихованих шарів чи кількістю нейронів в них, НМ виводить значення «відсутність процесу навчання». Враховуючи поведінку логістичної функції для даного хаосу, його присутність зумовлює наявність значної величини похибки навчання (30÷50%). Виявлення його можна здійснити методом виключення. В Додатку наведено код програми, який реалізує даний алгоритм. В табл. 1 наведено результати роботи даного коду програми. Згідно з табл. 1, коли «проходить процес навчання» для даної багатошарової НМ визначаються оптимальний параметр навчання та величина похибки навчання. Коли процес навчання унеможливлений існуванням хаотичного стану другого типу, система виводить «відсутність процесу навчання». Таким чином підбір

оптимальної швидкості навчання може приводити до зменшення кількості епох, а отже зменшення часу навчання НМ. На думку авторів розгляд швидкості навчання як величини, що може змінюватись від епохи до епохи, в деякій мірі може вирішувати розглянуті в даній статті задачі.

Таблиця.1. Результати роботи тришарової НМ.

Режим роботи тришарової НМ	Результат роботи програми
Відсутній процес навчання НМ за умови: кількість прихованих шарів-10; кількість нейронів у прихованому шарі -30; кількість епох -1000	
Проходить процес навчання НМ за умови: кількість прихованих шарів-3; кількість нейронів у прихованому шарі -3; кількість епох -1000	

Список використаних джерел

- [1] Руденко О. Г. Штучні нейронні мережі / О. Г. Руденко, Є. В. Бодяньський. – Харків: Компанія СМІТ, 2006. – 404 с.
- [2] Субботін С. О. Нейронні мережі: теорія та практика: навч. посіб. / С. О. Субботін. – Житомир: Вид. О.О. Євенок, 2020. – 184 с.
- [3] Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия - Телеком, 2001. – 382 с.
- [4] Субботін С. О. Нейронні мережі: навчальний посібник / С. О. Субботін, А.О. Олійник; під заг. ред. проф. С.О. Субботіна. – Запоріжжя: ЗНТУ, 2014. – 132 с.
- [5] Yuriy Olenych, Sergiy Sveleba, Ivan Katerynychuk, Ivan Kunio, Ivan Karpa Features of deep studyneural network / [Електронний ресурс]. – Режим доступу: <https://openreviewhub.org/lea/paper-2019/features-deep-study-neural-network#>
- [6] [Информационная энтропия хаоса / Хабр \(habr.com\)](#)
- [7] Кузнецов А.П. Динамические системы и бифуркации – Саратов: ООО Издательский центр «Наука», 2015. – 168 с.
- [8] Хазова Ю.А. Элементы теории бифуркаций. Часть 1. Динамические системы: учебно-методическое пособие / Ю.А. Хазова – ФГАОУ ВО «Крымский федеральный университет имени В. И. Вернадского». – Симферополь, 2019. – 55 с.

Додаток

Код програми для багат шарової НМ з можливістю побудови діаграми розгалуження
<pre> import numpy as np import array as arr from matplotlib import pyplot as plt # оголошення за задання необхідних змінних hiddenSize = 15 # кількість нейронів на прихованому шарі alpha = 0.001 # швидкість навчання (коефіцієнт альфа) eps = 0.001 # бажана точність навчання num = 100 # кількість шарів arr_eps = [] # масив для значень похибки навчання arr_age = [] # масив для значень кількості епох навчання # задання тренувального набору даних x = np.array([[1,0,1], [0,1,1], [0,0,1]]) y = np.array([[0], [0], [1]]) def sigmoid(x): # функція активації return 1 / (1 + np.exp(-x)) def sigmoid_output_to_derivative(output): # метод обчислення похідної від функції активації return output * (1 - output) def gen_synapse(x, y, hiddenSize, num): # генерація початкових ваг synapse = [] np.random.seed(1) for i in range(num): if i == 0: synapse.append(2 * np.random.random((len(x[0]),hiddenSize)) - 1) elif i == num - 1: synapse.append(2 * np.random.random((hiddenSize,len(y[0]))) - 1) else: synapse.append(2 * np.random.random((hiddenSize,hiddenSize)) - 1) return synapse chastota=arr.array('d',[0]) xx=arr.array('d',[0]) delta_n=arr.array('d',[0]) delta=arr.array('d',[0]) xx=arr.array('d',[0]) xxx=arr.array('d',[0]) def training(x, y, alpha, eps, hiddenSize, synapse, num, arr_age, arr_eps): # метод навчання нейронної мережі </pre>

```

alphas=[]
for alpha in np.arange(0.001,1.2111,0.001):
    delta=[]

    #print("Навчання нейронної мережі")
    age = 1
    while True:
        age += 1
        layers = []
        for i in range (num + 1):
            if i == 0:
                layers.append(x)
            else:
                layers.append(sigmoid(np.dot(layers[i - 1],synapse[i - 1])))
        layer_errors = []
        layer_deltas = []

        layer_errors.append(layers[num] - y)
        e = np.mean(np.abs(layer_errors[0]))

        if (age % 1) == 0:
            arr_age.append(age)
            arr_eps.append(e)
            #print("Похибка на " + str(age) + " ітерації: " + str(e))

        if(age > 100):
            break

        if (e < eps):
            #print("Точність " + str(round(e, 4)) + " досягнута за " + str(age) + " епох(и)")
            break

        layer_deltas.append(layer_errors[0] * sigmoid_output_to_derivative(layers[num]))

        for i in range (num - 1):
            layer_errors.append(layer_deltas[i].dot(synapse[num - 1 - i].T))
            layer_deltas.append(layer_errors[i + 1] * sigmoid_output_to_derivative(layers[num - 1 - i]))

        for i in range (num):
            synapse[num - 1 - i] -= alpha * (layers[num - 1 - i].T.dot(layer_deltas[i]))
            #print('np.mean(synapse[num - 1 - i])'+str(synapse[num - 1 - i])+'='+str(np.mean(synapse[num -
1 ])))

            delta.append(np.mean(synapse[num - 1 ]))
            #print(delta)

        chastota.append(alpha-delta[age-1]-delta[age-1]*delta[age-1])
        xx.append(delta[age])

    data=(min(delta)+max(delta))/2
    for delta0 in delta:
        delta0=data
        delta0=alpha-delta0-delta0*delta0
        delta_n.append(delta0)
        xxx.append(alpha)
        #print('data0',data0,alpha)
        data=delta0

```

```

synapse = gen_synapse(x, y, hiddenSize, num)
training(x, y, alpha, eps, hiddenSize, synapse, num, arr_age, arr_eps)
delta_n=np.array(delta_n)
#delta=np.array(delta)
chastota=np.array(chastota)
xx=np.array(xx)
xxx=np.array(xxx)
print(len(xxx),len(delta_n))
plt.scatter(xxx, delta_n, s=1.0, alpha=0.9)
plt.title("Діаграма розгалуження")
plt.xlabel(u"alpha")
plt.ylabel(u"x")
plt.show()

plt.scatter(xx, chastota, s=1, alpha=0.9)
plt.xlabel(u"xn")
plt.ylabel(u"xn+1")
plt.show()

```

Код програми для визначення наявності або відсутності процесу навчання, визначення значення оптимальної швидкості навчання і величини похибки при цьому.

```

data =delta_n
ps =np.abs(np.fft.fft(data))
m=len(ps)
mm=len(epsilon)
time_step =1
freqs = np.fft.fftfreq(data.size, time_step)
idx = np.argsort(freqs)

for i in range(mm):
    if ps[idx[i]]>20000:
        if epsilon[idx[i]]<0.2:
            alpha_optimalne=xxx[idx[i]]
            epsilon_min=epsilon[idx[i]]
            print( 'проходить процес навчання-', 'alpha оптимальне=',alpha_optimalne, 'мінімальна
похибка=',epsilon_min)
            break
        else:
            print('відсутність процесу навчання')
            break

```

CHAOTIC STATES OF A MULTILAYER NEURAL NETWORK

**S. Sveleba¹, I. Katerynychuk¹, I. Kunyo¹, I. Karpa¹, O. Semotyuk², Ya. Shmygelsky¹,
N. Sveleba³, V. Kunyo⁴**

¹*Ivan Franko National University of Lviv,
107 Tarnavsky St.,
UA-79017 Lviv, Ukraine
incomlviv@gmail.com*

²*Ukrainian Academy of Printing,
19 Pid Holoskom St.,
UA-79020 Lviv, Ukraine*

³*Lviv Institute of Economy and Tourism,
8 Mentsynskyi St., UA-79007 Lviv, Ukraine*

⁴*Lviv Polytechnic National University,
12 Bandera St., UA-79013, Lviv, Ukraine*

The study of the influence of learning speed η on the learning process of a multilayer neural network was studied. The program for the multilayer neural network was written in Python. We determined the speed of learning at which the best learning is observed. To analyze the impact of learning speed on the learning process, a logistic function was used, which describes the doubling of frequency. It is shown that the learning error function is characterized by bifurcation processes that lead to a chaotic state at $\eta > 0.8$. The optimal value of the learning speed is determined, which determines the appearance of the process of doubling the number of local minima. Increasing the number of hidden layers and the number of neurons in each layer does not lead to a radical change in the diagram of the logistics function, and hence the optimal value of the learning speed. It is shown that the increase in the number of hidden layers, as well as the number of neurons in them, with increasing speed of learning is accompanied by the appearance of a chaotic state. The chaotic state is characterized by the lack of a learning process. Using the Fourier transform of the error function, the bifurcation process depending on the learning speed was investigated. On the basis of these researches the program for definition of size of optimum speed of training of a neural network is developed. The text of the article adds program codes for a multilayer neural network with the ability to build a branching diagram and a program to determine the presence or absence of the learning process, taking into account the value of optimal learning speed and error.

Key words: multilayer neural network, optimal learning speed, bifurcations, chaotic states.

*Стаття: надійшла до редакції 20.10.2021,
доопрацьована 03.11.2021,
прийнята до друку 04.11.2021*