

## ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ ГЛИБИННОГО НАВЧАННЯ ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ АВТОНОМНИХ МАШИН

О.С. Білоконь

Одними з найбільш частими задачами, що зустрічаються у побудові інтелектуальних систем машинного зору для побудови інтелектуальних автономних машин є задачі класифікації та регресії. Розв'язання задач класифікації використовуються для рефлекторних дій автономних машин. А розв'язання задач прогнозування можна застосовувати для побудови систем машинного зору для наділення інтелектуальних автономних машин знаннями з довкілля, які своєю чергою, важливі для запланованих прогнозованих рухів. Визначення класу екземплярів задач є важливою процедурою для ефективного проектування систем глибокого навчання. В цих рамках у роботі досліджено можливість застосування багатошарової нейронної мережі, в якості регресора для побудови елементарних функціональних відображень для подальшого прогнозування. У дослідженні окреслено особливості функціонування та комплектації спеціалізованої робототехнічної системи, яка розглядається у даній роботі, як інтелектуальна автономна машина або фізичний агент, згенерований набір даних точок за елементарними функціями, виконуються аналітичне моделювання та моделювання систем що навчаються. Побудовано графік вхідних даних, визначено архітектуру нейронної мережі, реалізовано алгоритм метода градієнтного спуску і на останок побудовані вихідні графіки: процесу навчання, прогнозування результатів та порівняльний графік прогнозованих результатів накладений на графік вхідних даних. У результаті дослідження проведено оцінку інтелектуальних можливостей машини до запланованих рухів.

Ключові слова: нейронні мережі, глибоке навчання, робототехнічні системи, автономні інтелектуальні машини, фізичні агенти, регресія.

Одними из наиболее частыми задачами, которые встречаются в построении интеллектуальных систем машинного зрения для построения автономных машин являются задачи классификации и регрессии. Решение задач классификации используются для рефлекторных действий автономных машин. А решения задач прогнозирования можно применять для построения систем машинного зрения для наделения интеллектуальных автономных машин знаниями из окружающей среды, которые в свою очередь, важны для запланированных прогнозируемых движений. Определение класса экземпляров задач является важной процедурой для эффективного проектирования систем глубокого обучения. В этих рамках в работе исследована возможность применения многослойной нейронной сети, в качестве регрессора для построения элементарных функциональных отображений для дальнейшего прогнозирования. В исследовании обозначены особенности функционирования и комплектации специализированной робототехнической системы, которая рассматривается в данной работе, как интеллектуальная автономная машина или физический агент, сгенерированный набор данных точек по элементарным функциям, выполняются аналитическое моделирование и обучающее моделирование. Построен график входных данных, определена архитектура нейронной сети, реализован алгоритм метода градиентного спуска и напоследок построены выходные графики: процесса обучения, прогнозирования результатов и сравнительный график прогнозируемых результатов наложенного на график входных данных. В результате исследования проведена оценка интеллектуальных возможностей машины в прогнозируемых движениях.

Ключевые слова: нейронная сеть, глубокое обучение, робототехническая система, автономная интеллектуальная машина, физические агенты, регрессия.

One of the most common tasks that arise in building intelligent machine vision systems for intellectually autonomous machines is the problems of classification and regression. Classification problems are used for the reflexive action of autonomous machines. Prediction tasks can be used to build machine vision systems to provide intelligent autonomous machines with environmental knowledge, which in turn is important for planned predictable movements. Defining a class of task instances is an important procedure for the effective design of deep learning systems. In this context, the possibility of using a multilayered neural network as a regressor to construct elementary functional mappings is explored for further prediction. The study outlines the peculiarities of functioning and configuration of a specialized robotics system, considered in this paper as an intelligent autonomous machine or physical agent, generates a set of data points for elementary functions, analytical modeling and modeling of training systems. Input graph was constructed, neural network architecture was defined, gradient descent algorithm was implemented, and output schedules were finally constructed: learning process, results prediction and comparative graph of predicted results superimposed on the input graph. As a result of the study, an assessment of the machine's intellectual ability to predict was made.

Key words: neural networks, deep learning, robotic systems, autonomous intellectual machines, physical agents, regression.

### Вступ

В процесі пошуку способів створення автономних машин (агентів) ключовою задачею є розв'язання проблеми здатності агентів до самостійного виконання складних завдань без втручання людини. Дослідники замислюються над тими засобами які дозволять машинам (агентам) самостійно сприймати навколишнє середовище та виконувати необхідні міркування. Дана здатність елементарна для людини і надто важка для машини (фізичного агента).

За для характеристики дослідження у контексті побудови фізичних агентів, які поведуть себе розумно в том сенсі, що дії виконавчих механізмів цих агентів будуть являти собою раціональну реакцію на дані одержувані через датчики. Надамо класифікацію атрибутів за критеріями рівня зворотних реакцій:

- рефлекторні дії,
- дії які сприймаються отриманими знаннями з довкілля [1].

Більшість фізичних агентів мають датчики (мікрофони, відеокамери, далекоміри і т. ін.) за допомогою яких автономні машини отримують дані з навколишнього середовища та приводи (колеса, крила, захоплення, синтез мови) за допомогою яких вони можуть надавати деякий вплив на навколишнє середовище. Тому можемо ввести та окреслити автономну інтелектуальну машину (фізичного агента) який забезпечить контекст для подальшого обговорення у теоретичних, аналітичних та практичних частинах роботи в рамках дослідження. Автономна інтелектуальна машина (фізичний агент) визначається функціонуванням її елементів та конструкцією.

Для цього слід, по-перше, окреслити особливості функціонування. Функціональну складову робототехнічної системи беремо зі зразка дезінфікуючого робота. Він має назву «Ультрафіолетовий дезінфекційний робот» (також його називають скорочено УФ-ДР або англ. Ultraviolet Disinfection Robot, UV-DR) – це автономний робот для дезінфекції, що застосовується в секторі охорони здоров'я та на виробничих лініях фармацевтичної індустрії. Робот використовується в основному в циклі очищення, але не обмежується тільки цією функцією, з метою зменшення розповсюдження госпітальних або внутрішньо лікарняних інфекційних захворювань, (англ. HAIs), вірусів, бактерій та інших видів або шкідливих органічних матеріалів. УФ-ДР – це мобільний робот, який може автономно керувати, концентрацією ультрафіолетового світла, що випромінює на задалегідь визначені інфекційні гарячі точки в кімнатах пацієнтів та інших частинах лікарняних палат або кімнат та приміщень, тим самим дезінфікуючи та знищуючи бактерії та вірус на всіх відкритих поверхнях. Час витримки становить десять хвилин, що знищує до 99 % бактерій [6].

По-друге, ультрафіолетовий дезінфекційний робот складається з наступних елементів: колісний привод – візок, лазерний сканер, апаратна частина комп'ютера, інтерфейс користувача, елементи живлення, ультрафіолетова лампа, відеокамера, яка забезпечує машинний зір [6]. Всі атрибути агента які ми визначили представляють минулі й теперішні дослідження вони ж є повністю незалежними один від одного. Дослідники націлені розробляти автономні машини, які мають спільні риси двох атрибутів – рефлекторний, і другий – розуміти дані, які надходять з довкілля та розробляти нові моделі реагування в наслідок процесу навчання, мета якого максимізація можливостей агента.

Теперішні робототехнічні успіхи вражають своєю технічною новизною і дивують своєю широкою функціональністю. Але це тільки на перший погляд. Всілякі маніпулятори, автономні машини жорстко запрограмовані на певні дії та в значною мірою залежать від людей. Але не дивлячись на зроблені дослідні успіхи робототехніка стикається з двома ключовими проблемами:

- обмеженою автономністю;
- вузькими параметрами раціональної реакції на зовнішнє середовище.

З метою подолання наявних проблем сучасні робототехнічні дослідження ставлять перед собою основну мету – це максимальна автономність, що веде до подолання залежності від людей.

Дослідницький підхід ділиться на два напрямки:

- інтелектуальна роботизація – дослідники вважають, що робот мусить мати необхідні знання про навколишнє середовище, а також ступінь передчасного планування своїх дій – для розв'язання цієї проблеми створюються роботи які ведуть докладні записи про своє оточення, містять перелік об'єктів і даних про місце положення. Дані дослідження залежать від прогресу досліджень в області представлення і збереження знань;
- реактивна роботизація – альтернативний підхід, розробка націлена на робототехнічні системи, які реагують в навколишньому середовищі, дана система, ще має назву (реактивні) робототехнічної системи [1].

Виходячи з цього дуже важливо усвідомити, що методологія дослідження в цій області ведеться за двома основними напрямками:

- теоретичний підхід – дослідники намагаються відшукати засоби розрахункових відображень різних характеристик інтелекту тварин і перш за все людини. Даний підхід веде до методології, орієнтованої на моделювання, оскільки основна мета досліджень полягає в тому що б розширити наше розуміння інтелекту, а це значить що акцент тут робиться на самих процесах, що покладені в його основі, а не на демонстрації їх зовнішніх проявів;

- інженерний підхід – основне завдання в розробки систем, які демонструють ту чи іншу інтелектуальну поведінку. Даний підхід веде до методології орієнтованої на демонстрацію певних дій, оскільки основна мета тут полягає у створенні автономної системи, що демонструє поведінку відповідне тих чи інших поставленому цілям.

**Методологія** спираючись на всесвітньо відомі підходи наше дослідження теж зорієнтовано на теоретичний та інженерний підхід:

- у теоретичному підході моделювання та розрахунки здійснюються у Wolfram Mathematica 12 – система комп'ютерної алгебри, широко використовувана в наукових, інженерних, математичних і комп'ютерних областях;

- у інженерному підході для демонстрації певних інтелектуальних дій системи розробляється програмний код мовою програмування Python.

В цьому дослідженні для розробки та побудови автономної машини, які можуть рухатись, як рефлекторно так і прогнозовано за мету береться огляд задачі регресії та дослідження можливостей застосування багаточислової нейронної мережі у регресійних задачах для побудови графіків наступних функцій:

$$f(x) = \frac{1}{x};$$

$$f(x) = \frac{1}{x}, 1 \leq x \leq 100;$$

$$f(x) = 3x^2 + 5.$$

Для досягнення мети слід виконати науково-дослідні завдання.

1. У теоретичній частині:
  - 1.1. розглянути основи теоретичної робототехніки та штучних нейронних мереж.
2. У аналітичній частині:
  - 2.1. розглянути основи глибокого навчання на прикладі елементарного алгоритму: лінійної регресії.
3. У практичній частині у процесі тестування реалізованого коду нейронної мережі мовою програмування Python виконати такі кроки:
  - 3.1. визначити архітектуру нейронної мережі;
  - 3.2. написати програмний код та представлення основних кроків алгоритму спроектованої нейронної мережі;
  - 3.3. запустити для тестування програму:
    - 3.3.1. тестуємо програму для першої функції;
    - 3.3.2. тестуємо програму для другої функції;
    - 3.3.3. тестуємо програму для третьої функції;
  - 3.4. зводимо дані прогресу навчання.

## Основна частина

### Теоретична частина

Саймон Хайкін виділяє три фундаментальні класи нейромережових архітектур [2]:

- одношарові мережі прямого поширення;
- багатшарові мережі прямого поширення;
- рекурентні мережі.

Нами будуть розглянуті одношарові й багатшарові нейронні мережі. У цьому дослідженні віддаємо пріоритет такій архітектурі з огляду на те, що на базі цих класів буде побудована нейронна мережа для прогнозування. У теоретичній частині ми намагаємось ввести в контекст основні поняття штучних нейронних мереж на яких базуються основи теоретичної робототехніки. Тут ми розглянемо одношарові та багатшарові мережі. Ми не претендуємо на широке охоплення матеріалу. Торкаємось тільки тих понять які будуть слугувати основній меті дослідження та проміжним цілям розділу, які своєю чергою обумовлені побудовою розрахункових моделей штучних нейронних мереж як одношарових, так і багатшарових.

**Штучні нейронні мережі** – математична модель основою якої є штучний нейрон, який здійснює нелінійне перетворення F суми добутку вхідних сигналів на вагові коефіцієнти [3].

Математична модель представлена

$$y = F(\sum_{i=1}^n w_i x_i) = F(S) = F(WX^T), \quad (1)$$

де  $W = (w_1, w_2, \dots, w_n)$  – вектор вагових коефіцієнтів,  $X = (x_1, x_2, \dots, x_n)$  – вектор вхідних сигналів;  $S = \sum_{i=1}^n w_i x_i$  – зважена сума; F – оператор нелінійного перетворення або функція активації.

Зважена сума S іноді представляють у вигляді  $S = \sum_{i=1}^n w_i x_i - T$ , де T – поріг нейронного елемента, що характеризує зрушення функції активації по осі абсцис [3, с.5-6].

Для розуміння поняття штучного нейрона пропонуємо його схему (рис. 1).

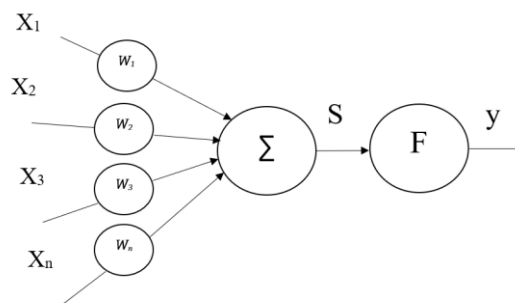


Рис. 1. Схема штучного нейрона

**Функція активації** – оператор нелінійного перетворення. Можуть виступати різні функції, які обираються відповідно задачі, що розглядається та типом нейронної мережі.

Одинична східчаста функція та біполярна східчаста функція (порогова функція) [3]:

$$y = \text{sign}(S) = f(x) = \begin{cases} 1, & S > 0 \\ 0, & S \leq 0 \end{cases}$$

або

$$y = \begin{cases} 1, & S > 0 \\ -1, & S \leq 0 \end{cases} \quad (2)$$

На рис. 2 візуалізовано значення функцій.

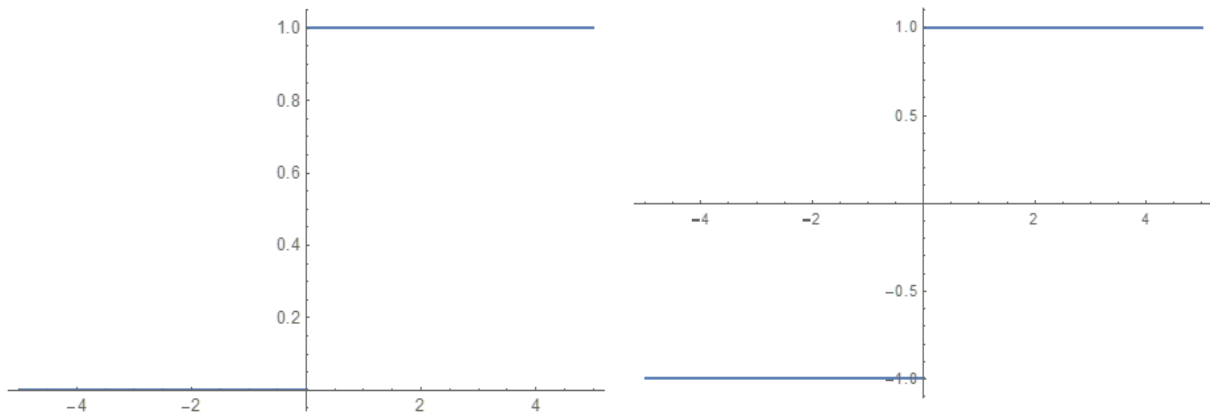


Рис. 2. Графік одинична східчаста функція та біполярна східчаста функція

**Сігмоїдна функція:**

$$y = \frac{1}{e^{-x} + 1} \quad (3)$$

Сігмоїдна функція характеризується монотонністю і всюди диференційованою її графік показано на рис. 3.

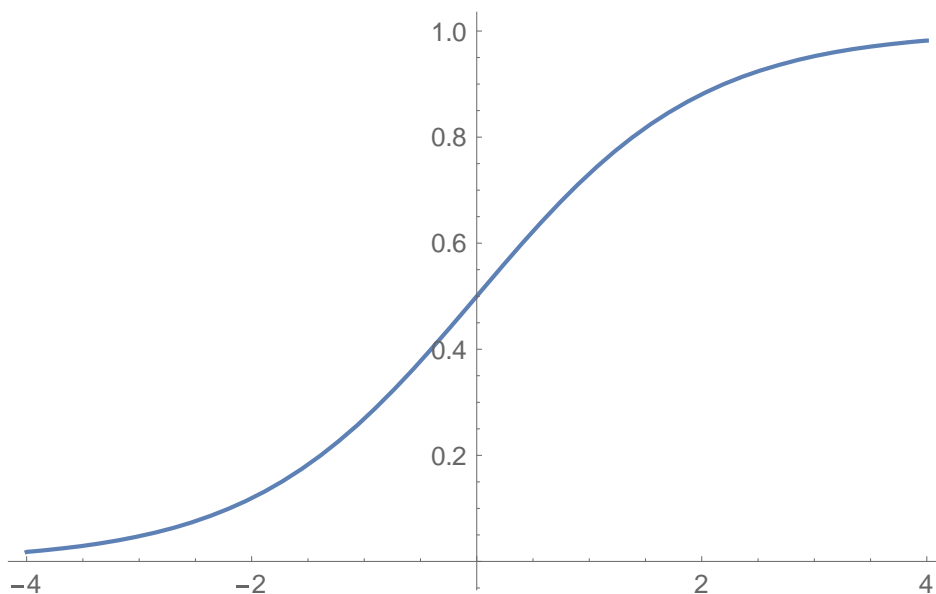


Рис. 3. Графік сігмоїдної функції

**Одношарова нейронна мережа** – це мережа у якій існує вхідний шар сигнал від якого передається на вихідний шар. Топологія такої мережи показана на рис. 4.

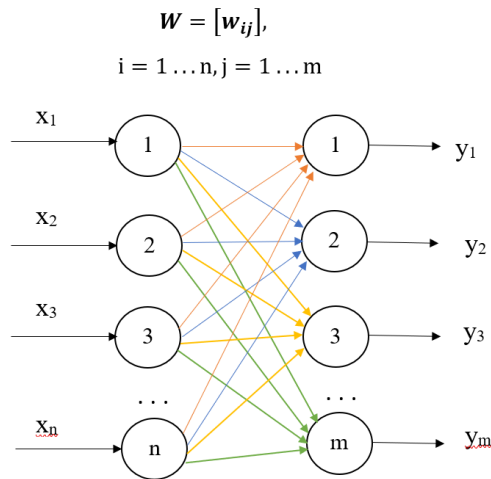


Рис. 4. Топологія одношарової нейронної мережі прямого розповсюдження

**Мережа прямого розповсюдження** – це мережі в яких сигнали передаються в одному напрямку.

Виходячи з топології, яка показана на рис. 4 кожен вхід  $x_i$  з'єднано з кожним нейроном єдиного шару нейронної мережі. Вихід  $j$ -го нейронного елемента розраховується за формулою [3].

$$y_j = F \left( \sum_{i=1}^n x_i w_{ij} - T_j \right), \quad (4)$$

де  $w_{ij}$  – вагові коефіцієнти між  $i$ -им входом та  $j$ -им нейроном. Таким чином вагові коефіцієнти нейронної мережі можна представити у вигляді матриці (5) розмірністю  $p \times m$ , де  $n$  – кількість входів нейронної мережі,  $m$  – кількість нейронних елементів [3]:

$$W = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{pmatrix}. \quad (5)$$

Основною задачею при використанні нейронної мережі є підбір вагових коефіцієнтів для розрахунку задачі, для цього використовують правила навчання нейронних мереж:

1. Правило навчання Хебба;
2. Персептрон Розенблатта;
3. Правило навчання Відрю – Хоффа.

**Багатшарова нейронна мережа** характеризується наявністю одного або декількох прихованих шарів. Топологія цієї мережі показана на рис. 5.

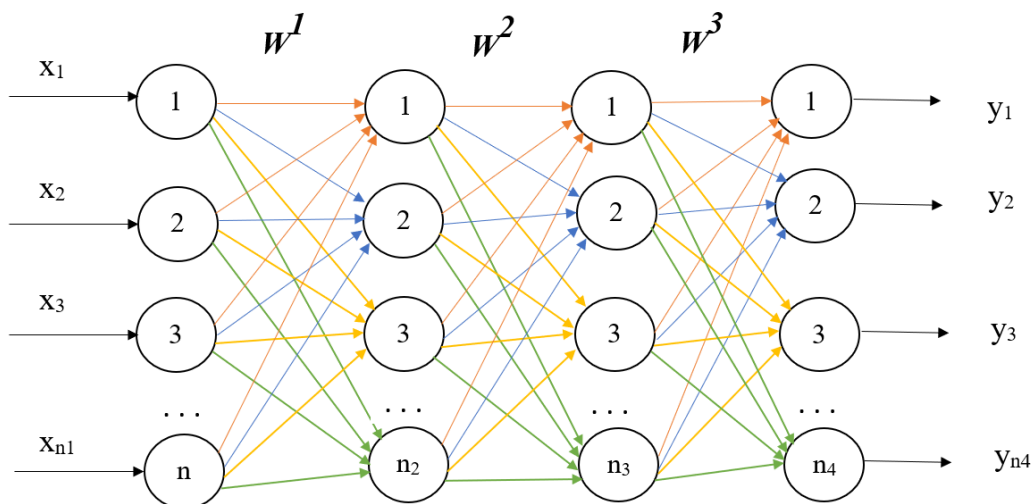


Рис. 5. Топологія багатшарової нейронної мережі

Також акцентується увага [3], що в основі цього типу нейронних мереж лежить теорема А.М. Колмогорова: будь-яку безперервну функцію  $n$ -змінних на одиничному відрізку  $[0,1]$  можна представити у вигляді суми кінцевого числа одновимірних функцій:

$$f(x_1, x_2, \dots, x_n) = \sum_{p=1}^{2n+1} g\left(\sum_{i=1}^n \lambda_i \varphi_p(x_i)\right), \quad (6)$$

де функції  $g$  та  $\varphi_p$  – одновимірні та безперервні,  $\lambda_i = \text{const}$  для всіх  $i$ .

Крім цього, до основних понять нейронних мереж слід додати алгоритм зворотного поширення помилки та представити його стислий опис [3].

Алгоритм зворотного поширення помилки використовується для навчання багатoshарових нейронних мереж. В процесі роботи алгоритму виділяють дві фази [3]:

- фаза прямого розповсюдження сигналу, що входить;
- фаза зворотного розповсюдження сигналу.

Алгоритм зворотного розповсюдження помилки налічує наступні кроки [3]:

- 1) крок – задається швидкість навчання;
- 2) випадково ініціалізувати вагові коефіцієнти та поріг;
- 3) на вхід по чергово подаються вхідні образи та для кожного образу виконуються наступні операції
  - 3.1. фаза прямого розповсюдження сигналу, що входить;
  - 3.2. фаза зворотного розповсюдження сигналу;

- 4) модифікуємо вагові коефіцієнти та пороги нейронної мережі;
- 5) розраховується середньо квадратична помилка мережі;

6) цикл для перевірки помилки – якщо середньо квадратична помилка більше мінімальної повертаємось до кроку 3).

Отже, у теоретичній частині розглянуто фундаментальні класи нейромережових архітектур: одношарові мережі прямого поширення та багатoshарові мережі прямого поширення, математичну модель штучної нейронної мережі, функції активації, правила навчання нейронних мереж: правило навчання Хебба, Персептрон Розенблатта, правило навчання Відроу – Хоффа. Також було розглянуто теорему А.М. Колмогорова про будь-яку безперервну функцію, досліджено алгоритм зворотного розповсюдження помилки. Крім цього представлено та досліджено топології нейронних мереж. Даний теоретичний базис дозволяє нам спроектувати багатoshарову нейронну мережу для задачі прогнозування для подальшої перевірки гіпотези про поєднання рефлексивних та запланованих рухів автономних машин.

## Аналітична частина

Слід зазначити та ввести у контекст, що аналітичну частину дослідження націлено на розв’язання задач та для перевірки теоретичних відомостей для осмислення алгоритмів машинного навчання та глибоких мереж і відтворенні їх в інтелектуальних автономних машинах. Теоретична частина в якій ведені основні поняття нейронних мереж та визначення алгоритму машинного навчання – окреслені абстрактною формою. Для подання конкретики наведемо приклад розрахунку лінійної регресії.

По-перше, слід зазначити, що лінійна регресія задається рівнянням виду:

$$y = ax + b \quad (7)$$

Тому для огляду прикладу розрахунку задаємо точки на координатній прямій (рис. 6).

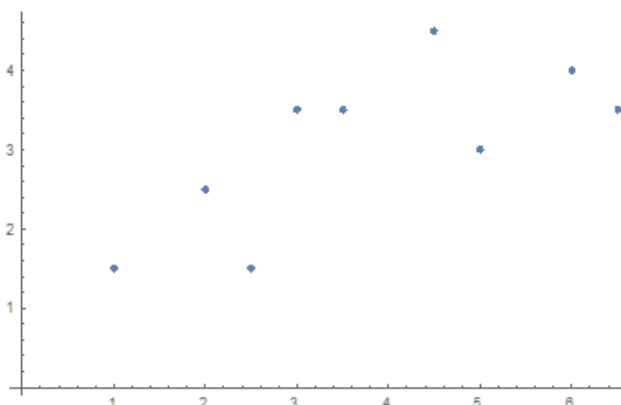


Рис. 6. Графік точок

Випишемо у табл. 1 координати точок

Таблиця 1.

<b>x</b>	1	2	3	4,5	6	6,5	5	3,5	2,5
<b>y</b>	1,5	2,5	3,5	4,5	4	3,5	3	3,5	1,5

Нам потрібно знайти лінію яка ближче всього проходить біля заданих точок – це можливо зробити за формулою суми квадратів помилок [5].

$$sse = \sum_{i=1}^n (y_i - \hat{y}_i) \quad (8)$$

Для побудови лінії регресії використовуючи дані координат точок слід розрахувати суму квадратів відхилення від середнього знаходимо розкид значень  $x_i$  та  $y_i$  рівняння виду [5]:

$$ss_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2, \quad (9)$$

$$ss_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2. \quad (10)$$

Визначаємо попарний добуток рівняння [5]:

$$ss_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}). \quad (11)$$

Знаходимо коефіцієнти  $a$  та  $b$  рівняння, де  $\bar{x}$  та  $\bar{y}$  середні значення [5]:

$$a = \bar{y} - b\bar{x}, \quad (12)$$

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \text{ або } b = \frac{ss_{xy}}{ss_{xx}}. \quad (13)$$

Результати розрахунків наведемо у табл. 2:

Таблиця 2.

$x_i$	$y_i$	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})^2$	$(x_i - \bar{x})(y_i - \bar{y})$
1	1,5	-2,7	-2,05	7,29	4,20	5,53
2	2,5	-1,7	-1,05	2,89	1,10	1,78
3	3,5	-0,7	-0,05	0,49	0,0025	0,03
4,5	4,5	0,8	1,45	0,64	2,10	1,16
6	4	2,3	2,95	5,29	8,70	6,78
6,5	3,5	2,8	3,45	7,84	11,90	9,66
5	3	1,3	1,95	1,69	3,80	2,53
3,5	3,5	-0,2	0,45	0,04	0,20	-0,09
2,5	1,5	-1,2	-0,55	1,44	0,30	0,66
$\sum_{i=1}^n x_i = 34$ $\bar{x} = 3,7$ $\bar{y} = 3,05$	$\sum_{i=1}^n y_i = 27,5$	–	–	$ss_{xx} = 27,61$	$ss_{yy} = 32,30$	$ss_{xy} = 28,04$

Результати розрахунку показано на рис. 7.

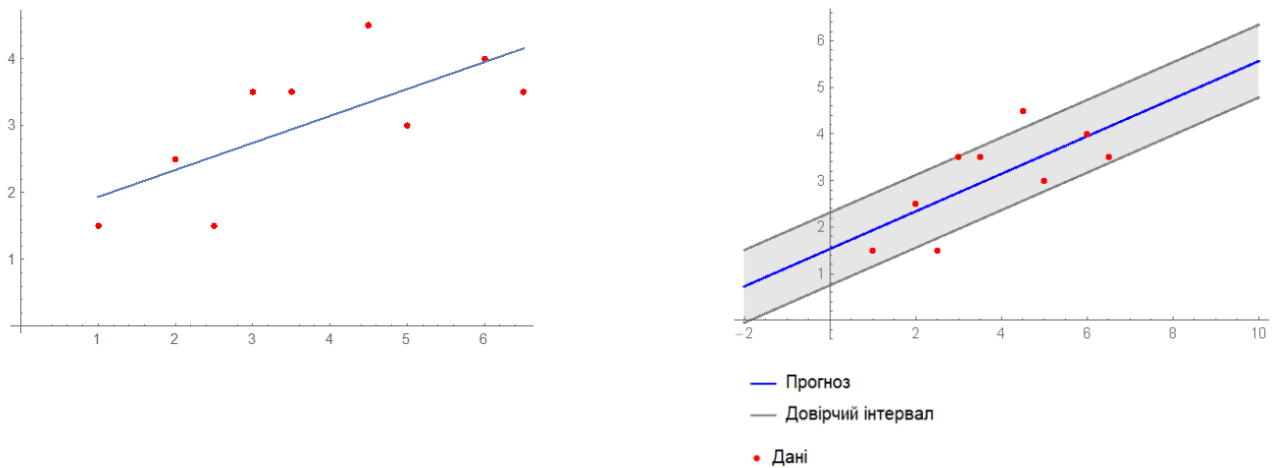


Рис. 7. Графік результатів розрахунків

Отже, в аналітичній частині ми провели розрахунок лінійної регресії за координатами точок. Нам вдалось успішно перевірити теоретичні відомості, щодо алгоритму машинного навчання на базі лінійної регресії. Ми змогли осмислити алгоритм та принципи його побудови. На цих принципах нам надається можливість проаналізувати взаємозв'язок між залежною змінною та декількома незалежними змінними, що дає можливість перейти до аналізу множинної лінійної регресії та нелінійної регресії дані якої описуються функціями, які зазначались вище у меті дослідження.

### Практична частина

З метою дослідження можливостей застосування багатошарової нейронної мережі у регресійних задачах слід, по-перше, спроектувати нейронну мережу та визначити її архітектуру.

У дослідженні була обрана архітектура, яка складається з 10 вузлів вхідного шару, 6 вузлів прихованого шару та одного вузла вихідного шару. Її показано на рис. 8.

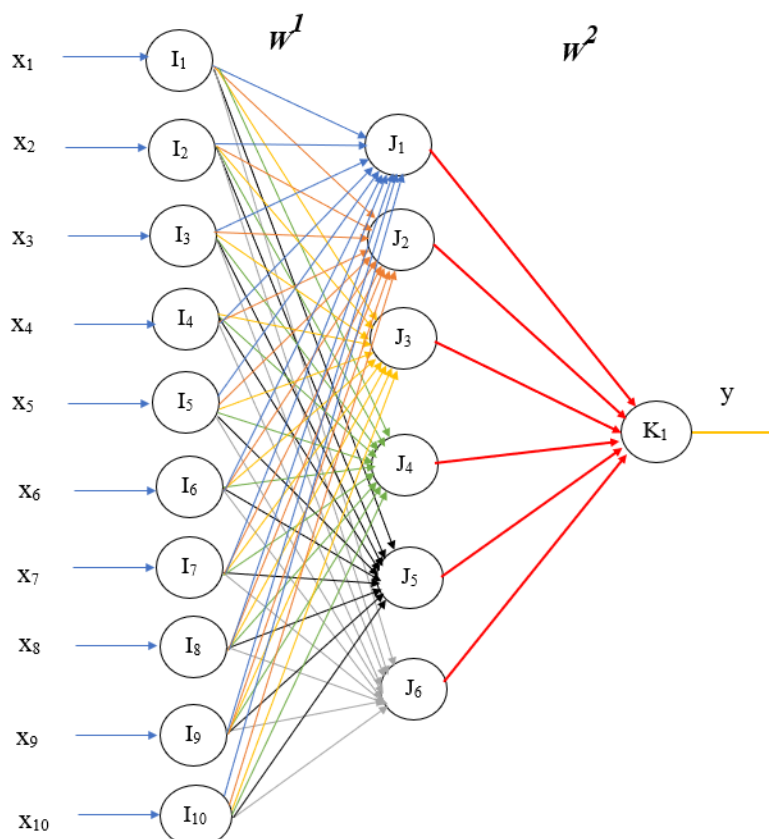


Рис. 8. Архітектура нейронної мережі



Крім цього написано програмний код мовою Python, тут представимо основні кроки алгоритму нейронної мережі:

- 1) імпортуємо бібліотеки (фреймворки):
  - 1.1. математичний пакет,
  - 1.2. пакет для роботи з багатовимірними масивами,
  - 1.3. пакет для візуалізації даних,
  - 1.4. бібліотека базових алгоритмів нейронних мереж;
- 2) задаємо змінні для генерування даних: min, max, кількість точок:
  - 2.1. присвоюємо змінній X дані = min, max, кількість точок,
  - 2.2. присвоюємо змінній Y функції  $f(x) = \frac{1}{x}$ ,  $f(x) = \frac{1}{x}$   $1 \leq x \leq 100$ ,  $f(x) = 3x^2 + 5$ ,
  - 2.3. нормалізуємо дані для змінної Y;
- 3) створюємо тренувальний набір даних:
  - 3.1. для змінної данні та мітки присвоюємо вектори значень;
- 4) будуємо графік вхідних даних;
- 5) визначаємо архітектуру нейронної мережі;
- 6) встановимо алгоритм градієнтного спуску;
- 7) тренуємо нейронну мережу;
- 8) запуск нейронної мережі для тренувальних точок даних;
- 9) будуємо графік прогресу навчання;
- 10) будуємо графік прогнозованих результатів.

По-перше, запускаємо програму для тестування на даних функції  $f(x) = \frac{1}{x}$ , яка має графік виду рис. 9.

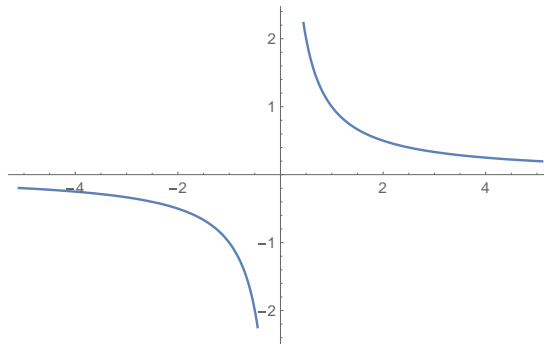


Рис. 9. Графік функції  $f(x) = \frac{1}{x}$

Отримуємо графіки вхідних даних, прогресу навчання та зведеного графіку для функції  $f(x) = \frac{1}{x}$  (рис. 10).

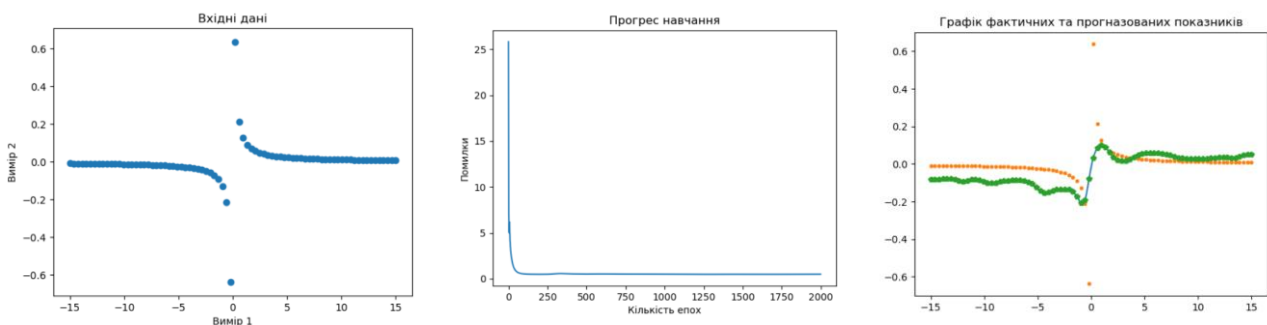


Рис. 10. Графіки вхідних даних, прогресу навчання та зведеного графіка для функції  $f(x) = \frac{1}{x}$

По-друге, запускаємо програму для тестування на даних функції  $f(x) = \frac{1}{x}$  з інтервалом  $1 \leq x \leq 100$ , яка має графік виду (рис. 11).

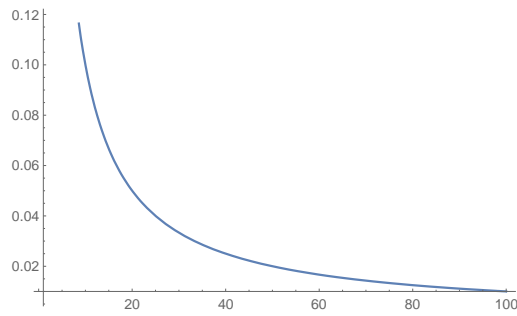


Рис. 11. Графік функції  $f(x) = \frac{1}{x}$  з інтервалом  $1 \leq x \leq 100$

Отримуємо наступні зображення рис. 12.

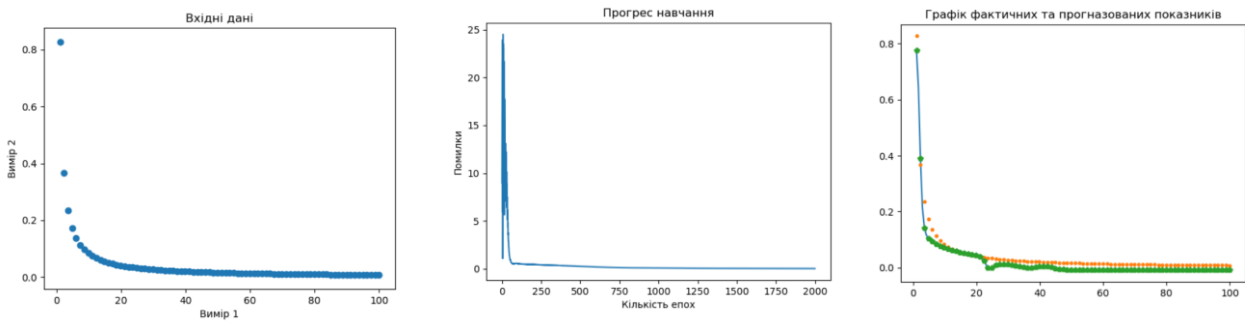


Рис. 12. Графіки вхідних даних, прогресу навчання та зведений графік для функції  $f(x) = \frac{1}{x}$  з інтервалом  $1 \leq x \leq 100$

На останньому етапі тестування нейронної мережі запускаємо програму для даних функції  $f(x) = 3x^2 + 5$ , яка має графік виду (рис. 13).

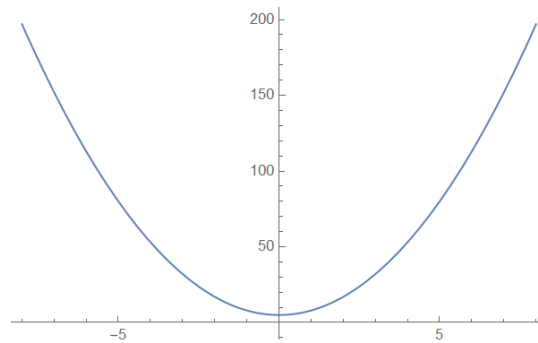


Рис. 13. Графік функції  $f(x) = 3x^2 + 5$

На виході отримуємо наступні зображення прогнозів на рис. 14.

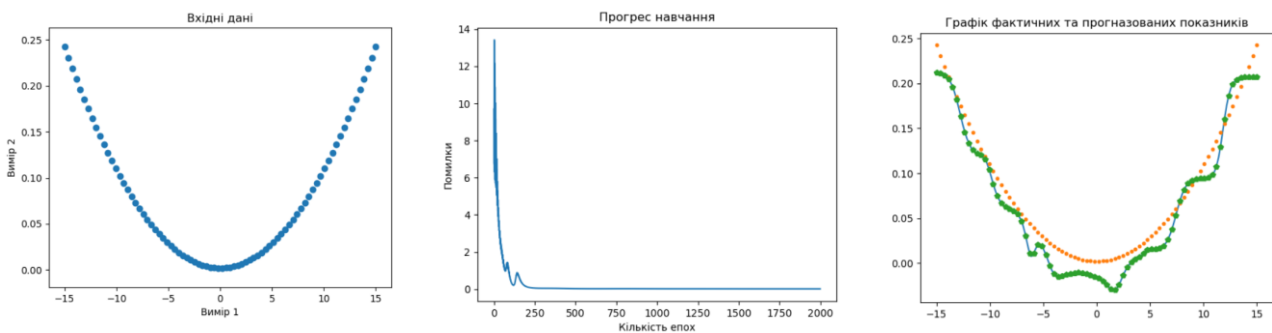


Рис. 14. Графіки вхідних даних, прогресу навчання та зведений графік для функції  $f(y) = 3x^2 + 5$

Після успішного тестування програми ми отримуємо наступні зведені дані прогресу навчання (табл. 3).

Таблиця 3.

Кількість епох	Дані помилок		
	$f(x) = \frac{1}{x}$	$f(x) = \frac{1}{x}$ $1 \leq x \leq 100$	$f(y) = 3x^2 + 5$
100	0,5282363532224323	0,5406480920775694	0,6246294861714475
200	0,48695747945416895	0,43574360382096666	0,13425399128064702
300	0,5411510545681447	0,37805762443111024	0,0459809956629007
400	0,5353050094749539	0,31999922315951124	0,0397333087869715
500	0,5146937074607045	0,2555387910111747	0,0271056798876749
600	0,5233212915850461	0,19273663941747654	0,022291652316069784
700	0,5152123499401748	0,14296077455638403	0,022701425558254655
800	0,5101269594445456	0,10920177537453199	0,024013486848812476
900	0,5073113077703687	0,08762008155242104	0,022501133612715676
1000	0,5020754530813775	0,07347612125174832	0,01985128976848194
1100	0,49619747904510625	0,06340339740213358	0,0184380374757029
1200	0,4887874775235817	0,05539822200329508	0,018226112156396062
1300	0,48488413171481237	0,0484461311965539	0,018130466072103137
1400	0,4885291477150965	0,04245004428281425	0,0174042779882214
1500	0,488722802843056	0,03941221780836337	0,016382902471254516
1600	0,48700145775355136	0,03584027967408876	0,015663204482686113
1700	0,4862369075399846	0,03058729819245275	0,015354596661953098
1800	0,4878413937160495	0,027751337962742893	0,01518032476369293
1900	0,492258671328267	0,02539066761023108	0,014870013532043098
2000	0,49718767979294287	0,023609023933740696	0,01443033518591383

У практичному розділі нам вдалося провести тестування нейронної мережі для елементарних функцій з метою прогнозування. Отримано графічні дані прогнозованих результатів. На зведеному графіку, отриманого у процесі роботи, системи для першої функції ми бачимо, що функція та її дані, що апроксимуються страждає від недонавчання – вона не вловлює властиві даним кривину, друга функція, що апроксимує дані краще узагальнюється на нових точках та показує кривину і третя функція так само як і перша не вловлює кривину. Отже, спроектована нами нейронна мережа з її обмеженою кількістю параметрів не дає чіткого прогнозування. Тому в подальших дослідженнях слід збільшувати параметри нейронних мереж для поліпшення результатів прогнозу.

## Висновки

У теоретичній частині розглянуто фундаментальні класи нейромережевої архітектури: одношарові мережі прямого поширення та багатошарові мережі прямого поширення, математичну модель штучної нейронної мережі, функції активації, правила навчання нейронних мереж: правило навчання Хебба, Персептрон Розенблатта, правило навчання Відроу – Хоффа. Також розглянуто теорему А.М. Колмогорова про будь-яку безперервну функцію, досліджено алгоритм зворотного розповсюдження помилки. Крім цього представлено та досліджено топології нейронних мереж. Даний теоретичний базис дозволяє нам спроектувати багатошарову нейронну мережу для задач прогнозування для подальшої перевірки гіпотези про поєднання рефлексивних та запланованих рухів автономних машин.

У аналітичній частині ми провели розрахунок лінійної регресії за координатами точок. Нам вдалось успішно перевірити теоретичні відомості, щодо алгоритму машинного навчання на базі лінійної регресії. Ми змогли осмислити алгоритм та принципи його побудови. На цих принципах нам надається можливість проаналізувати взаємозв'язок між залежною змінною та декількома незалежними змінними, що дає нам

можливість перейти до аналізу множинної лінійної регресії та нелінійної регресії, дані якої описуються функціями, які зазначались вище у меті дослідження.

У практичному розділі нам вдалося провести тестування нейронної мережі для елементарних функцій з метою прогнозування. Отримано графічні дані прогнозованих результатів. На зведеному графіку, отриманого у процесі роботи, системи для першої функції ми бачимо, що функція та її дані, що апроксимуються страждає від недонавчання – вона не вловлює властиві даним кривину, друга функція, що апроксимує дані краще узагальнюється на нових точках та показує кривину і третя функція так само як і перша не вловлює кривизну. Отже, спроектована нами нейронна мережа з її обмеженою кількістю параметрів не дає чіткого прогнозування. Тому у подальших дослідженнях слід збільшувати параметри нейронних мереж для поліпшення результатів прогнозу.

## Література

1. Брукшир Дж. Гленн, Брилов Деннис. Компьютерные науки. Базовый курс, 13-е изд.: Пер. с англ. СПб.: ООО “Диалектика”, 2019. 1104 с.
2. Хайкин Саймон. Нейронные сети: полный курс, 2-е изд.: Пер с англ. СПб.: ООО “Диалектика”, 2019. 1104 с.
3. Ахтеров А.В., Кирильченко А.А. Основы теоретической робототехники. Искусственные нейронные сети. Обзор. Препринты ИПМ им. М.В. Келдыша. 2008. № 2. 20 с. URL: <http://library.keldysh.ru/preprint.asp?id=2008-2>
4. Джоши Пратик. Искусственный интеллект с примерами на Python.: Пер. с англ. СПб.: ООО “Диалектика”, 2019. 448 с.
5. Драйпер Н., Смит Г. Прикладной регрессионный анализ: В 2-х кн, Кн.1/ Пер, с англ, 2-е изд., перераб, И доп. М.: *Финансы и статистика*, 1986. 366 с. (Математико-статические методы за рубежом).
6. Білоконь О. Стан та перспективи розвитку дезінфекційних робототехнічних системи. Матеріали VIII Всеукраїнської науково-практичної конференції «Глушковські читання» 2019.11.29. С. 16–19.

## References

1. Brookshear J. Glenn, Brylow Dennis. Computer science an overview. Basic course, 13th ed.: Per. from English. St. Petersburg.: Dialectics LLC, 2019. 1104 p.: ill. Parl. Titus. English.
2. Khaykin Simon. Neural networks: full course, 2nd ed.: Transl. From English. St. Petersburg.: Dialectics LLC, 2019. 1104 p.: ill. Parl. Titus. English.
3. Akhterov A.V., Kirilchenko A.A. Fundamentals of Theoretical Robotics. Artificial neural networks. Overview. KIAM Preprints M.V. Keldysh. 2008. N 2. 20 p. URL: <http://library.keldysh.ru/preprint.asp?id=2008-2>.
4. Joshi, Prateek. Artificial intelligence with Python. : Per. from English. SPb. : LLC “Dialectics”, 2019. 448 p. Paral. tit. English.
5. Draper N., Smith G. Applied regression analysis: In 2 kn, Kn, 1 / Per, from English, - 2nd ed., Revised, And ext., - M, Finances and statistics, 1986. 366 p. silt, (Mathematical-static methods abroad).
6. Bilokon O. Status and prospects of development of disinfection robotic systems. Materials VIII All-Ukrainian scientific-practical conference “V.M. Glushkov reading” 2019.11.29. P. 16–19.

Одержано 10.03.2020

### Про автора:

Білоконь Олександр Сергійович,  
молодший науковий співробітник,  
аспірант Інституту кібернетики імені В.М. Глушкова НАН України,  
03187, м. Київ-187,  
проспект Академіка Глушкова, 40.  
Кількість наукових публікацій в українських виданнях – 3.  
<https://orcid.org/0000-0002-9658-5730>

### Місце роботи автора:

Національна академія наук України,  
01601, м. Київ,  
вул. Володимирська, 54.  
Тел.: (38)(044) 239-65-43.  
E-mail: sashabilokon82@gmail.com