

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних та практичних робіт
з дисципліни «Теорія штучного інтелекту»
для студентів спеціальності 126 –
«Інформаційні системи та технології»

Міністерство освіти і науки України
Вінницький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних та практичних робіт
з дисципліни «Теорія штучного інтелекту»
для студентів спеціальності 126 –
«Інформаційні системи та технології»

Електронне видання
комбінованого (локального та мережного) використання

Вінниця
ВНТУ
2021

Рекомендовано до видання Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 10 від 20.05.2021 р.)

Рецензенти:

П. І. Кулаков, доктор технічних наук, професор ВНТУ

А. Я. Кулик, доктор технічних наук, професор ВНМУ ім. М. Пирогова

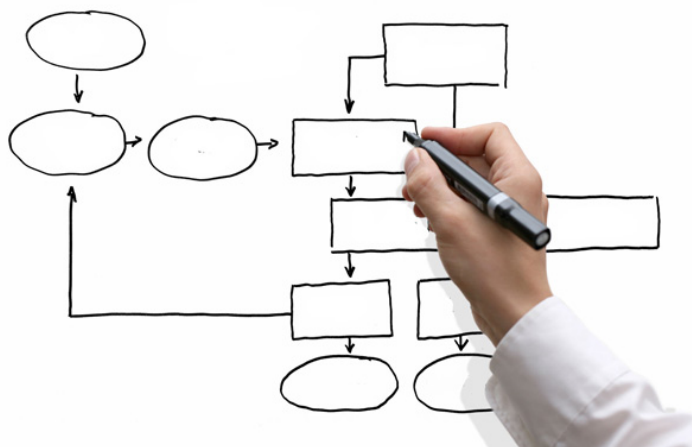
Методичні вказівки до виконання лабораторних та практичних робіт з дисципліни «Теорія штучного інтелекту» для студентів спеціальності 126 – «Інформаційні системи та технології» [Електронний ресурс] / Уклад. Ю. Ю. Іванов. Вінниця : ВНТУ, 2021. (PDF 30 с.)

У методичних вказівках наведено завдання на лабораторні та практичні роботи з дисципліни «Теорія штучного інтелекту», а також рекомендована для їхнього виконання література. Сформовано блоки контрольних питань, які можна використовувати для підготовки до контрольних заходів.

ЗМІСТ

Захист та структура звіту лабораторних робіт	4
Критерії оцінювання знань, умінь та навичок студентів.....	5
Вступ.....	6
Практична робота.....	8
Лабораторна робота № 1. Регресійний аналіз. Визначення оптимальної регресійної моделі	10
Лабораторна робота № 2. Математичні методи безумовної оптимізації.....	14
Лабораторна робота № 3. Екстракція знань із експериментальних даних методом дерев рішень	18
Лабораторна робота № 4. Машинне навчання. Штучні нейронні мережі	22
Рекомендована література	26
Оцінювання студентами матеріалів курсу та рівня викладання.....	29

ЗАХИСТ ТА СТРУКТУРА ЗВІТУ ЛАБОРАТОРНИХ РОБІТ



Студенти виконують *підготовчу роботу* відповідно до теми завдання, щоб зрозуміти теоретичні та практичні аспекти, математичну основу, застосування цих задач у професійній діяльності як тестових полігонів. Вони вивчають конспект лекцій, рекомендовані матеріали (література, аудіо-, відеофайли) тощо. Глибоке вивчення теоретичного матеріалу, набуті навички програмування та вміння використовувати пакети прикладних програм допоможуть успішно виконати поставлені задачі.

За результатами виконаної роботи на аркушах формату А4 оформлюється *звіт*, у якому відображаються всі результати роботи. Він має містити такі *елементи*: титульний аркуш, аналіз літератури та ключового матеріалу, формалізована постановка задачі, розв'язання задачі (числовий приклад) та алгоритми методів, програмне забезпечення, результати роботи, висновки, список використаної літератури. Оформлення звіту необхідно виконувати охайно за *правилами ДСТУ 3008:2015 та ДСТУ ГОСТ 7.1:2006* [1, 2]. Спочатку звіт необхідно подати викладачу, після виконання коригувань студент може захистити роботу. Також потрібно показати роботу програмного забезпечення на ЕОМ та відповісти на контрольні запитання, задані викладачем.

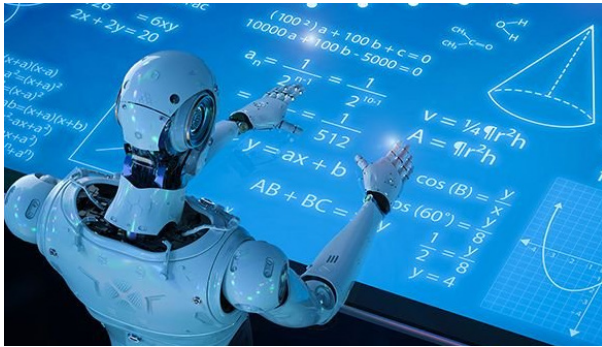
Студент має впевнено та чітко *презентувати* виконану ним роботу, показати її суть, знайомство з ключовою літературою; вміти розв'язати поставлену задачу та визначити методи її вирішення; володіти відповідним термінологічним апаратом; мати прийнятний рівень мовної грамотності, включаючи володіння функціональним стилем викладення матеріалу. Чітке виконання поставлених вимог та використання аргументованих відповідей на запитання викладача з посиланнями на роботи вчених допоможе підвищити оцінку.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ, УМІНЬ ТА НАВИЧОК СТУДЕНТІВ

Рівень компетен-тності	За нац. шкалою	За шка-лою ЕКТС	Критерії оцінювання
IV Високий (творчий) «5»	відмінно «5»	A	Виставляється, якщо під час відповіді на питання виявлено всебічні, систематизовані, глибокі знання матеріалу, який виноситься на контроль, уміння вільно виконувати завдання, передбачені програмою, знання основної і додаткової літератури, передбаченої програмою на рівні творчого використання.
III Достатній (конструк-тивний) «4»	добре «4+»	B	Повні знання з питань і задач, що стоять перед студентом. Уміння викладати основні ідеї. Вміння професійно відстоювати свою точку зору. Припускаються несуттєві неточності у викладенні матеріалу та у відповідях.
	добре «4»	C	Достатньо повні знання з поставлених питань і задач. Вміння викладати основні ідеї. Здатність самостійно застосовувати вивчений матеріал на рівні стандартних ситуацій, наводити власні приклади на підтвердження власних тверджень. Вміння доводити правильність своїх рішень. Несуттєві неточності у відповідях та деякі не-раціональності при програмуванні задач.
II Середній (репро-дуктив-ний) «3»	задовільно «3+»	D	Студент може відтворити значну частину теоретичного матеріалу, виявляє знання та розу-міння основних положень, з допомогою викла-дача може аналізувати матеріал, робити висно-вки та розробляти програмні блоки. Пояснення неповні, неаконічні, не завжди точні. Відповіді на питання неповні, містять неточності, при програмуванні застосовуються не найраціональ-ніші рішення.
	задовільно «3»	E	Задовільні знання програмного матеріалу на рівні вищому за початковий. Здатність за допо-могою викладача логічно відтворювати значну частину матеріалу. Під час відповіді на запи-тання виникають труднощі у деяких положен-нях, відповіді не повні, програми пишуться не-раціонально, не використовуються всі ефектив-ні засоби програмування.
I Низький «2»	незадовільно з можливістю повторного складання «2»	FX	Теорією володіє на рівні фрагментів, викладає матеріал уривчасто. Помітна складність в об-грунтуванні рішень. Самостійно, без допомоги викладача, не може сформувати алгоритм рі-шення задачі. Програми не раціональні та нее-фективні.
	незадовільно з обов'язковим повторним вивченням «2»	F	Теорією володіє на рівні фрагментів, викладає матеріал уривчасто. Не може обгрунтувати рі-шення задачі, на запитання викладача дає не-правильні відповіді. Самостійно не може сфор-мувати алгоритм рішення задачі.

ВСТУП

Що мислимо – то можливо, що можливо – то мислимо.
Г. В. Лейбніц



Методи штучного інтелекту – одна з найважливіших складових сучасної науки. З одного боку, вона містить фундаментальні основи математики – теорію множин, математичну логіку, теорію алгоритмів, теорію графів, науку про дані, машинне навчання тощо; з іншого боку, слугує базою для численних до-

датків в науці, економіці, техніці, соціальній сфері [3–9]. *Актуальність та необхідність вивчення* дисципліни зумовлена потребою пошуку прихованих закономірностей у множині даних, рішенням інтелектуальних задач пошуку інформації в корпоративних базах даних, прийняття рішень на основі використання систем правил та експертних систем, нейронних мереж, евристичних стратегій та алгоритмів. *Практична спрямованість* дисципліни зумовлена актуальною потребою в опануванні інформаційного світогляду фахівця, використанні розділів сучасної науки в сфері штучного інтелекту; побудові систем управління складними процесами, програмуванні додатків для використання на сучасній обчислювальній техніці.

У цьому курсі студентам запропоновано у якості тестових полігонів низку теоретичних та практичних задач з різних науково-практичних галузей. Варто зазначити, що в умовах обмеженого часу досить важко охопити всі теми такого об'ємного курсу. **Метою дисципліни** є вироблення системи знань, умінь і навичок у студентів, як майбутніх фахівців, щодо пошуку рішення у просторі станів, використання понять штучного інтелекту, експертних систем, машинного навчання для роботи з прикладними пакетами математичного моделювання або розробленим програмним забезпеченням у різних галузях професійної діяльності. Обов'язковим є виконання для кожної задачі числового прикладу (наприклад, в системах математичного моделювання – *MatLab*, *MathCad*, *Mathematica*, *Maple* тощо), підвищує бали розробка програмного забезпечення з обґрунтуванням вибору технологічної платформи для дослідження ефективності роботи методу розв'язання задачі.

Програмне забезпечення розробляється з використанням принципів структурного, функціонального, об'єктно-орієнтованого програмування. Програмний комплекс має бути гнучким, наприклад, надавати можливість змінювати параметри вхідних даних; містити графічне меню, яке

забезпечує можливість введення початкових умов задачі, вибір параметрів та збереження отриманого розв'язку.

Таким чином, студенти мають розширити свої знання за цим курсом та вдосконалити навички програмування, що знадобиться їм у професійній діяльності.

ПРАКТИЧНА РОБОТА



Основне завдання дисципліни полягає у виконанні *самостійного пошуку* студентом (або за допомогою викладача) певної цікавої *науково-дослідницької задачі*, її аналізу, розв'язанню (*теоретично та програмно*).

Студенти виконують *підготовчу роботу відповідно до теми завдання*, щоб зрозуміти теоретичні та практичні аспекти, математичну основу, застосування цих задач у професійній діяльності як тестових полігонів. Вони вивчають потрібні матеріали (література, аудіо-, відеофайли) тощо. Глибоке вивчення теоретичного матеріалу, набуті навички програмування та вміння використовувати пакети прикладних програм допоможуть успішно провести практичне заняття. Обов'язковим є виконання для кожної задачі числового прикладу (наприклад, у системах математичного моделювання – *MatLab, MathCad, Mathematica* тощо) та розроблення програмного забезпечення.

Студент має *оформити звіт* з виконаної роботи, *презентацію* та *виступити* перед аудиторією, яку представляє група студента та викладач дисципліни. Попередньо тема виступу обговорюється з викладачем курсу.

Спершу звіт необхідно подати викладачу, після виконання коригувань студент може захистити роботу. Оформлення звіту необхідно виконувати охайно за правилами *ДСТУ 3008:2015, ДСТУ ГОСТ 7.1:2006* [1, 2] для наукових робіт. Він має містити такі *елементи*: титульний аркуш, аналіз літератури та ключового матеріалу, формалізована постановка задачі, розв'язання задачі (числовий приклад) та алгоритми методів, програмне забезпечення, результати роботи, висновки, список використаної літератури.

У *презентації* студент наглядно показує результати своєї роботи. Для отримання *відмінної оцінки* необхідно виступити перед аудиторією, яку представляє група студента та викладач курсу. Студент має впевнено і точно презентувати виконану ним роботу.

Виконання такого завдання допоможе сформувати у студентів вміння використовувати методи та засоби викладання, організовувати навчальний процес, навчитися без сорому виступати перед аудиторією, вміти активізувати пізнавальну діяльність, стимулювати до критичного та творчого мислення, самостійного пошуку й дослідження за цим курсом. Студенти покращать свої знання та вдосконалять навички програмування, що знадобиться їм у професійній діяльності.

ЛАБОРАТОРНА РОБОТА № 1



Тема: Регресійний аналіз. Визначення оптимальної регресійної моделі.

Мета: розв'язати задачу регресії, навчитися вибирати оптимальну регресійну модель для прогнозування на основі вибірки даних, а також застосовувати відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [10–16], у яких подано теоретичні основи регресійного аналізу та проаналізовано відповідну задачу.

Завдання

Подати математичну постановку задачі *регресії (regression task)* для певної вибірки даних. Проаналізувати та теоретично обґрунтувати:

- основи аналізу вибірки даних (розбиття на навчальну, валідаційну та тестову вибірки);
- метод найменших квадратів (*least squares method*) у формі системи рівнянь та матричному вигляді;
- принцип зовнішнього доповнення (*external complement principle*) для пошуку оптимальної регресійної моделі;
- основи статистичного оцінювання для отриманого регресійного рівняння:
 - перевірка якості рівняння регресії (коефіцієнт множинної кореляції R , корінь середньої квадратичної помилки $RMSE$);
 - перевірка значущості рівняння регресії (F -критерій Фішера);
 - аналіз статистичної значущості параметрів моделі (t -критерій Стюдента);
 - перевірка виконання передумов методу найменших квадратів (наявність / відсутність автокореляції з використанням dw -критерію Дарбіна-Уотсона, гомо- або гетероскедатичність випадкової складової за допомогою тестів Гольдфельда-Квандта, Глейзера, рангової кореляції Спірмена, двостороннього критерію Фішера тощо).

Розв'язати такі задачі:

Задача 1. Отримати двофакторну регресійну залежність $y = f(x_1, x_2)$ для однієї із задач регресії репозиторію машинного навчання *UCI Machine Learning Repository* Каліфорнійського університету в Ірвіні (використати посилання <http://www.ics.uci.edu/~mlearn/MLRepository.html>):

1. З табл. 1.1 за виразом $N = (L + [\text{остання цифра року}]) \bmod 24$ (L – варіант студента за списком) вибрати задачу та сформулювати її математичну та змістовну постановки.

Таблиця 1.1 – Варіанти завдань

№	Задача	Фактори впливу x_1 та x_2
1	<i>Auto MPG</i>	<i>Horsepower</i> <i>Acceleration</i>
2	<i>Auto MPG</i>	<i>Displacement</i> <i>Year</i>
3	<i>Auto MPG</i>	<i>Displacement</i> <i>Horsepower</i>
4	<i>Auto MPG</i>	<i>Horsepower</i> <i>Year</i>
5	<i>Auto MPG</i>	<i>Weight</i> <i>Acceleration</i>
6	<i>Housing</i>	<i>NOX</i> <i>LSTAT</i>
7	<i>Housing</i>	<i>RM</i> <i>PTRATIO</i>
8	<i>Housing</i>	<i>PTRATIO</i> <i>LSTAT</i>
9	<i>Housing</i>	<i>B</i> <i>LSTAT</i>
10	<i>Wine Quality (Red)</i>	<i>pH</i> <i>Alcohol</i>
11	<i>Wine Quality (Red)</i>	<i>Fixed acidity</i> <i>Alcohol</i>
12	<i>Wine Quality (White)</i>	<i>Sulphates</i> <i>Alcohol</i>
13	<i>Wine Quality (White)</i>	<i>pH</i> <i>Alcohol</i>
14	<i>Yacht Hydrodynamics</i>	<i>Length-displacement ratio</i> <i>Froude number</i>
15	<i>Yacht Hydrodynamics</i>	<i>Froude number</i> <i>Longitudinal position of the center of buoyancy</i>
16	<i>SkillCraM Master Table</i>	<i>LeagueIndex</i> <i>ComplexUnitsMade</i>
17	<i>SkillCraM Master Table</i>	<i>ComplexUnitsMade</i> <i>Age</i>
18	<i>SkillCraM Master Table</i>	<i>HoursPerWeek</i> <i>MinimapRightClicks</i>
19	<i>SkillCraM Master Table</i>	<i>APM: Action per minute</i> <i>SelectByHotkeys</i>

Продовження таблиці 1.1

20	<i>Physicochemical Properties of Protein Tertiary Structure</i>	<i>Total surface area</i> <i>Fractional area of exposed non polar residue</i>
21	<i>Physicochemical Properties of Protein Tertiary Structure</i>	<i>Total surface area</i> <i>Fractional area of exposed non polar part of residue</i>
22	<i>Physicochemical Properties of Protein Tertiary Structure</i>	<i>Non polar exposed area</i> <i>Secondary structure penalty</i>
23	<i>Physicochemical Properties of Protein Tertiary Structure</i>	<i>Fractional area of exposed non polar residue</i> <i>Secondary structure penalty.</i>
24	<i>Physicochemical Properties of Protein Tertiary Structure</i>	<i>Fractional area of exposed non polar part of residue</i> <i>Secondary structure penalty</i>

2 Розбити дані на навчальну та тестову вибірки. Перевірити їх репрезентативність (однорідність даних).

Примітка. У навчальну вибірку можна включити, наприклад, непарні рядки даних, а в тестову – парні. Для перевірки репрезентативності вибірок розрахувати математичне сподівання (*mean*) та дисперсію (*std*) даних за кожною змінною. Якщо результати розрахунків вказують на приблизно однакові значення цих показників для навчальної та тестової вибірок, то можна вважати їх репрезентативними.

3. Зобразити експериментальні дані у формі однофакторних залежностей $y = f(x_1)$ та $y = f(x_2)$.

4. Знайти коефіцієнти 5 регресійних залежностей, серед яких 2 моделі згенерувати самостійно. Перевірити програмні розрахунки вручну для однієї з заданих моделей за допомогою скалярної та матричної форми методу найменших квадратів. Обов'язковими є моделі

$$y_1(x_1, x_2) = C_0 + C_1x_1 + C_2x_2 + C_3x_1x_2;$$

$$y_2(x_1, x_2) = C_0 + C_1x_1 + C_2x_2 + C_3x_1^2 + C_4x_2^2;$$

$$y_3(x_1, x_2) = C_0 + C_1x_1 + C_2x_2 + C_3x_1^2 + C_4x_2^2 + C_5x_1x_2.$$

5. Вибрати оптимальну регресійну модель та обґрунтувати свій вибір. Візуалізувати оптимальну регресійну модель та експериментальні дані. Кінцеві результати роботи відобразити у вигляді табл. 1.2.

Таблиця 1.2 – Результати регресійного аналізу

№	Регресійна модель	$RMSE_{tr}$	R_{tr}	$RMSE_{test}$	R_{test}
1					
2					
3					
4					
5					

Задача 2. Спрогнозувати розвиток пандемії у світі або епідемії *COVID-19* (коронавірусної інфекції) у певній країні, назву якої попередньо написати викладачу для уникнення повторень. Для розв'язання задачі використати апарат регресії та нормальний закон розподілу Гауса у формі

$$y(t) = b \cdot \exp(-(t - a)^2 \cdot c), \quad (1.1)$$

де t – щоденні відмітки часу; $y(t)$ – прогнозована кількість хворих у момент часу t ; a, b, c – параметри регресії.

Для розрахунків використати дані з відкритих джерел, наприклад:

- <https://ourworldindata.org/coronavirus>;
- <https://github.com/owid/covid-19-data/tree/master/public/data>;
- <https://www.ecdc.europa.eu/en/covid-19/data>;
- <https://data.world/datasets/covid-19>;
- <https://www.worldometers.info/coronavirus/>.

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab, MathCAD, Mathematica, Maple, Python Anaconda* тощо [40-47]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива. Видатні вчені.
2. Основи аналізу вибірки даних. Недонавчання та перенавчання.
3. Навчальна, валідаційна та тестова вибірки.
4. Репрезентативність вибірок даних.
5. Поняття регресійного аналізу. Математична постановка задачі регресії.
6. Метод найменших квадратів. Виведення ключових формул у скалярному та матричному вигляді.
7. Складність розрахунку коефіцієнтів регресії.
8. Методи розв'язання систем лінійних рівнянь.
9. Перевірка якості рівняння регресії. Параметри оцінювання моделі.
10. Перевірка значущості рівняння регресії (F -критерій Фішера).
11. Аналіз статистичної значущості параметрів моделі (t -критерій Стьюдента)
12. Перевірка виконання передумов методу найменших квадратів (автокореляція, гомо- та гетероскедатичність випадкової складової). dw -критерій Дарбіна-Уотсона. Тести Гольдфеля-Квандта, рангової кореляції Спірмена, двостороннього критерію Фішера.
13. Принцип зовнішнього доповнення.
14. Критерії вибору оптимальної регресійної моделі.
15. Практичне застосування регресійного аналізу.
16. Особливості розв'язання задачі регресії в пакетах математичного моделювання та середовищах програмування.

ЛАБОРАТОРНА РОБОТА № 2



Тема: Математичні методи безумовної оптимізації.

Мета: розв'язати задачу безумовної глобальної оптимізації з використанням математичного аналізу та класичних пошукових методів, навчитися застосовувати дані методи та відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [10, 17–21], у яких подано теоретичні основи теорії оптимізації, методи пошуку екстремуму функції та проаналізовано відповідну задачу.

Завдання

Подати загальну математичну постановку задачі глобальної оптимізації. Виділити задачі умовної та безумовної глобальної оптимізації (*constrained / unconstrained global optimization task*). Проаналізувати та теоретично обґрунтувати:

- аналітичний метод пошуку екстремуму функції багатьох змінних;
- пошукові методи безумовної оптимізації (*search techniques for optimization*) з табл. 2.1;

Таблиця 2.1 – Пошукові методи оптимізації

№	Нульовий порядок ($k_1 = 4$)	Перший порядок ($k_2 = 7$)	Другий порядок ($k_3 = 3$)
1	Хук-Дживс	Градiєнтний спуск	Ньютон
2	Нелдер-Мід	Покоординатний спуск	Ньютон-Рафсон
3	Пауелл	Гаусс-Зейдель	Левенберг-Марквардт
4	Випадковий пошук	Флетчер-Рівс	-
5	-	Девідон-Флетчер-Пауелл	-
6	-	Кубічна інтерполяція	-
7	-	Бройден-Флетчер-Гольдарб-Шенно	-

- основи стохастичних методів оптимізації (метаевристик) або ройового інтелекту (*swarm intelligence*);
- теорему Вульперта-МакКріді про відсутність безкоштовних сніданків (*No Free Lunch Theorem*).

Розв'язати задачу:

Задача 1. Дослідити залежність часу розв'язання задачі безумовної оптимізації t від кількості керованих змінних n :

1. Розв'язати задачу оптимізації, використавши поняття математичного аналізу. Для цього вибрати вихідні дані з табл. 2.2 за виразом $N = (L + [\text{остання цифра року}]) \bmod 10$, де L – варіант студента за списком. Побудувати двовимірний та тривимірний графіки функцій, відобразити на них глобальний оптимум.

Таблиця 2.2 – Вихідні умови задачі оптимізації

№	Цільова функція $f(x)$	Початкова точка x^0
1	$y = \sum_{i=1}^n (x_i - n)^2 / i^2$	$x_i^0 = 2i, i = \overline{1, n}$
2	$y = \sum_{i=1}^n (x_i + 7i)^2 / i$	$x_i^0 = 0, i = \overline{1, n}$
3	$y = \sum_{i=1}^n (x_i - i)^4$	$x_i^0 = \frac{i}{2}, i = \overline{1, n}$
4	$y = \sum_{i=1}^n (x_i + i)^2 / 0,1i^2$	$x_i^0 = i, i = \overline{1, n}$
5	$y = \sum_{i=1}^n (x_i + i)^4 / i$	$x_i^0 = 5i, i = \overline{1, n}$
6	$y = \sum_{i=1}^n (0,1x_i + 2i)^2$	$x_i^0 = 3, i = \overline{1, n}$
7	$y = \sum_{i=1}^n (i \cdot x_i)^2$	$x_i^0 = 2i, i = \overline{1, n}$
8	$y = \sum_{i=1}^n (i^2 - x_i^2)^4$	$x_i^0 = 5i, i = \overline{1, n}$
9	$y = 10 \cdot n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2\pi x_i))$	$x_i^0 = 3, i = \overline{1, n}$
10	$y = -\frac{1}{n} \cdot \sum_{i=1}^n (x_i \cdot \sin(\sqrt{ x_i }))$	$x_i^0 = 10i, i = \overline{1, n}$

2. Вибрати метод оптимізації (табл. 2.1) з кожного класу (стовпця таблиці) за допомогою виразу $N = (L + [\text{остання цифра року}]) \bmod k_i$, де L – варіант студента за списком, а k_i – кількість методів у класі. Навести детальні алгоритми роботи вибраних методів оптимізації. Виконати на папері вручну низку ітерацій та показати результати.

3. Експерименти для функції з табл. 1.1 провести, використавши від $n = \overline{2,30}$ керованих змінних. Змінюючи стартові точки та параметри алгоритму, дослідити їх вплив на ефективність роботи, зафіксувати час роботи та похибки обчислень. Результати подати у табл. 2.3.

Для перевірки роботи програми використати набір з трьох класичних тестових функцій із роботи [21] або за посиланнями:

– http://infinity77.net/global_optimization/index.html (*Global Optimization Benchmarks*);

– <https://www.sfu.ca/~ssurjano/optimization.html> (*Virtual Library of Simulation Experiments: Test Functions and Datasets. Optimization Problems*).

Експерименти проводити, не використовуючи готові пакети та функції оптимізації. Але їх можна застосувати для перевірки результатів роботи.

Таблиця 2.3 – Результати експериментів

Кількість керованих змінних n	Стартові точки, параметри алгоритму	Результати обчислень $f(x^*)$ (найбільше, найменше, середнє значення функції)	Час пошуку оптимуму t , сек (найбільший, найменший, середній час)	Похибки обчислень
2				
...
n				

4. За допомогою методу найменших квадратів апроксимувати отримані експериментальні дані функцією $t = f(n)$. Записати отримане регресійне рівняння та проаналізувати його.

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *MathCAD*, *Mathematica*, *Maple*, *Python Anaconda* тощо [40–47]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива. Видатні вчені.
2. Поняття оптимізації. Математична постановка задачі оптимізації.
3. Екстремум функції.
4. Градієнт та антиградієнт.

5. Критерій Сільвестра. Власні значення матриці Гесе.
6. Матриця Гесе. Квадратична форма.
7. Алгоритм ідентифікації оптимуму на основі понять математичного аналізу.
8. Методи умовної і безумовної оптимізації.
9. Класифікація методів безумовної оптимізації функцій багатьох змінних.
10. Особливості пошукових алгоритмів оптимізації багатовимірних функцій. Їх переваги та недоліки.
11. Особливості стохастичної оптимізації. Ройовий інтелект.
12. Метаевристики. Загальна метаевристична схема.
13. Як залежить час оптимізації від розмірності задачі?
14. Чи завжди при збільшенні розмірності задачі збільшується час пошуку оптимуму?
15. Які фактори впливають на час розв'язання задачі оптимізації?
16. Практичне застосування пошукових методів оптимізації.
17. Як вибрати «найкращий» оптимізатор? Теорема про відсутність безкоштовних сніданків.
18. Особливості розв'язання задачі оптимізації в пакетах математичного моделювання та середовищах програмування.

ЛАБОРАТОРНА РОБОТА № 3



Тема: Екстракція знань із експериментальних даних методом дерев рішень.

Мета: розв'язати задачу екстракції знань із даних, навчитися використовувати дерева рішень, а також застосовувати відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [10, 12, 22–26], у яких подано теоретичні основи екстракції знань та застосування дерев рішень.

Завдання

Подати математичну постановку задачі екстракції знань (*knowledge extraction*) з експериментальних даних. Розглянути застосування дерев рішень (*decision tree*) у задачах регресії та класифікації. Проаналізувати та теоретично обґрунтувати:

- алгоритми побудови дерев рішень:
 - *ID3*;
 - *C4.5*;
 - *CART*;
- алгоритм відсікання (*pruning*) для дикого (неоптимального) дерева;
- генерацію підвибірок (*subsample generation*):
 - випадкові підвибірки (*bagging, bootstrap aggregation*);
 - перехресне затвердження (*cross-validation*);
 - адаптивний підхід або підсилювання (*boosting*).
- алгоритм «випадковий ліс» (*random forest*).

Розв'язати задачу:

Задача 1. Синтезувати дерево рішень, яке описує залежність між вхідними $X = (x_1, \dots, x_n)^T$ та вихідною змінною Y для однієї із задач класифікації репозиторію машинного навчання *UCI Machine Learning Repository* Каліфорнійського університету в Ірвіні (використати посилання <http://www.ics.uci.edu/~mllearn/MLRepository.html>):

1. З табл. 3.1 за виразом $N = (L + [\text{остання цифра року}]) \bmod 33$ (L – варіант студента за списком) вибрати задачу та сформулювати її математичну та змістовну постановки.

Таблиця 3.1 – Варіанти завдань

№	Задача
1	<i>Adult</i>
2	<i>Amazon commerce reviews set</i>
3	<i>Annealing</i>
4	<i>Bank marketing</i>
5	<i>Cardiotocography</i>
6	<i>Chess (King-Rook vs. King)</i>
7	<i>Credit approval</i>
8	<i>Cylinder bands</i>
9	<i>Ecoli</i>
10	<i>Glass identification</i>
11	<i>Image segmentation</i>
12	<i>Low resolution spectrometer</i>
13	<i>Madelon</i>
14	<i>MAGIC gamma telescope</i>
15	<i>Mechanical analysis</i>
16	<i>MiniBooNE particle identification</i>
17	<i>Nursery</i>
18	<i>Page blocks classification</i>
19	<i>Pen-based recognition of handwritten digits</i>
20	<i>Poker hand</i>
21	<i>Spambase</i>
22	<i>SPECTF heart</i>
23	<i>Statlog (German credit data)</i>
24	<i>Statlog (Heart)</i>
25	<i>Statlog (Landsat satellite)</i>
26	<i>Statlog (Shuttle)</i>
27	<i>Statlog (Vehicle silhouettes)</i>
28	<i>Steel plates faults</i>
29	<i>University</i>
30	<i>Vertebral column</i>
31	<i>Wall-following robot navigation data</i>
32	<i>Wine</i>
33	<i>Yeast</i>

2 Розбити дані на навчальну та тестову вибірки. Перевірити їх репрезентативність (однорідність даних).

Примітка. У навчальну вибірку можна включити, наприклад, непарні рядки даних, а в тестову – парні. Для перевірки репрезентативності вибірок розрахувати математичне сподівання (*mean*) та дисперсію (*std*) даних за кожною змінною. Якщо результати розрахунків вказують на приблизно однакові значення цих показників для навчальної та тестової вибірок, то можна вважати їх репрезентативними.

3. Синтезувати дерево рішень для класифікації об'єктів з вибірки.

4. Відобразити графічно залежність частоти виникнення помилок класифікації $error_{DT}$ від розміру дерева рішень (кількості рівнів дерева N_{level} та вершин N_{edge}), тобто $error_{DT} = f(N_{level})$ та $error_{DT} = f(N_{edge})$, для навчальної та тестової вибірок.

5. Проаналізувати результати та визначити мінімальну частоту виникнення помилок, а також знайти оптимальний розмір дерева рішень. Візуалізувати оптимальне дерево та його правила.

6. Розрахувати матриці сплутувань (*confusion matrix*) найкращого дерева рішень на навчальній та тестовій вибірках. Зробити висновки про розроблений класифікатор.

7. Порівняти відомі результати для даної вибірки з отриманими у своїй роботі. Зробити відповідні висновки.

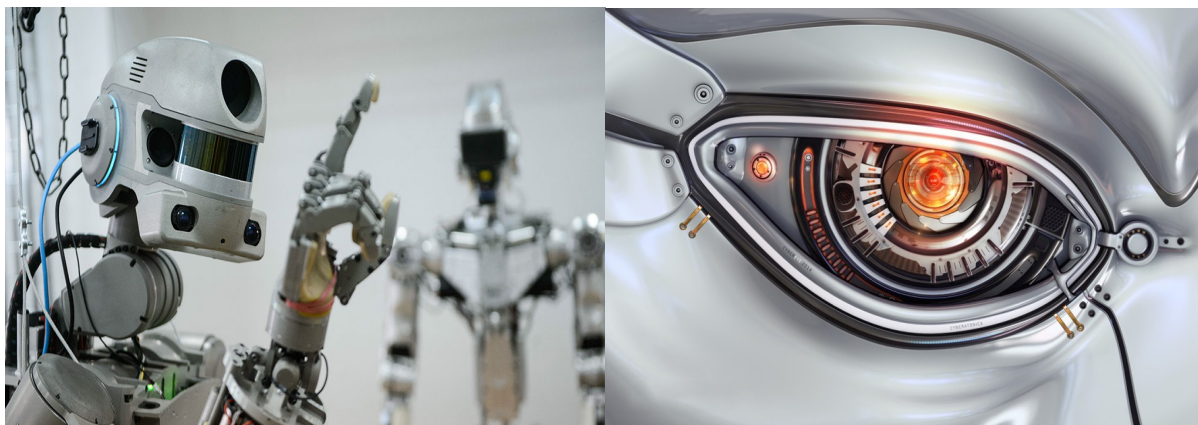
Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *MathCAD*, *Mathematica*, *Maple*, *Weka*, *R*, *Python Anaconda* тощо [40–47]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива. Видатні вчені.
2. Основи аналізу вибірки даних. Недонавчання та перенавчання.
3. Навчальна, валідаційна та тестова вибірки.
4. Репрезентативність вибірок даних.
5. Види шкал для вибірок.
6. Постановка задачі екстракції знань з експериментальних даних.
7. Постановка задачі класифікації.
8. Оцінювання якості класифікатора. Похибка класифікатора.
9. Дерево прийняття рішення.
10. Правила побудови дерева. Підхід «розділяй та володарюй».
11. Методи генерації підвибірок (випадкові підвибірки, перехресне затвердження адаптивний підхід).

12. Алгоритми побудови дерев рішень (*ID3*, *C4.5*, *CART*, *NewId*, *ITrule*, *CHAID*, *CN2*). Математична основа.
13. Алгоритм відсікання для неоптимального дерева;
14. Алгоритм «випадковий ліс».
15. Переваги та недоліки дерев рішень.
16. Оцінювання похибок класифікатора на матрицях сплутування.
17. Практичне застосування дерев рішень.
18. Особливості використання дерев рішень в пакетах математичного моделювання та середовищах програмування.

ЛАБОРАТОРНА РОБОТА № 4



Тема: Машинне навчання. Штучні нейронні мережі.

Мета: навчитися використовувати класичні архітектури штучних нейронних мереж, математичний апарат для їх оптимізації та відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [10, 27–39], у яких подано теоретичні основи розробки штучних нейронних мереж та розглянуто задачу їх навчання.

Завдання

Подати математичну постановку задачі навчання штучних нейронних мереж (*neural network training*). Проаналізувати та теоретично обґрунтувати алгоритми навчання:

- нейромережі Хопфілда (*Hopfield*);
- нейромережі Коско (*Kosko*);
- нейромережі Хеммінга-Ліппмана (*Hamming-Lippmann*);
- нейромережі Кохонена (*Kohonen Self-Organizing Maps*);
- багатошарового перцептрона (*perceptron*) та згорткової (*convolutional*) нейромережі на основі зворотного поширення похибки (*backpropagation*);
- на основі градієнтної оптимізації першого порядку (*gradient-based optimization of neural network*) із табл. 4.1.

Розв'язати такі задачі:

Задача 1. Розробити нейромережі Хопфілда (автоасоціативна пам'ять, модель Літтла), Коско (гетероасоціативна пам'ять) та Хеммінга-Ліппмана. Навчити їх розпізнавати чорно-біле зображення літер, які входять до імені студента, підписи або бар-коди (*QR, Data Matrix, Aztec*).

Задача 2. Розробити нейромережу Кохонена та розв'язати за її допомогою задачу кластеризації (кластерного аналізу) даних про успішність студентів вашої навчальної групи (n студентів).

Таблиця 4.1 – Нейромережеві градієнтні методи оптимізації

№	Перший порядок ($k = 10$)
1	Стохастичний градієнтний спуск (<i>Stochastic Gradient Descent, SGD</i>)
2	Класичний метод моментів (<i>classical momentum</i>)
3	Прискорений градієнтний метод Нестерова (<i>Nesterov Accelerated Gradient, NAG</i>)
4	Метод адаптивного градієнта (<i>AdaGrad</i>)
5	<i>AdaDelta</i>
6	Метод ковзного середнього (<i>Root Mean Square Propagation, RMSProp</i>)
7	Метод адаптивної оцінки моментів (<i>Adaptive Moment Estimation, ADAM</i>)
8	<i>AdaMax</i>
9	Прискорений по Нестерову метод адаптивної оцінки моментів (<i>Nesterov-accelerated Adaptive Moment Estimation, NADAM</i>)
10	<i>AMSGrad</i>

Вихідна вибірка подана в табл. 4.2. Генерувати дані по заліках з використанням функції $k = rand > 0,5$ (залік); для даних по дисциплінах $k = randint(...)$ від 0 до 100 балів; якщо у студента незалік або бальна оцінка з дисципліни менша 60, то він не отримує стипендію (коефіцієнт дорівнює нулю), якщо сума його оцінок складає 300-349 (коеф. 1,00), 350-409 (коеф. 1,25), 410-429 (коеф. 1,50), 430-449 (коеф. 1,75), 450-500 (коеф. 2,00). Дані нормалізувати лінійно від 0 до 1.

Таблиця 4.2 – Початкова вибірка даних про успішність студентів

№	Прізвище	Стать, x_1	Отрим. всі зал., x_2	Рейтинг за дисциплінами					Коеф. стипендії, x_8
				1, x_3	2, x_4	3, x_5	4, x_6	5, x_7	
1	Іванов	М (1)	Ні (0)	72	66	57	91	60	0,00
2	Корольов	М (1)	Так (1)	75	98	86	82	85	1,50
3	Летова	Ж (0)	Так (1)	85	89	85	92	85	1,75
...
n	Марчук	Ж (0)	Ні (0)	54	60	21	16	17	0,00

Для навчання використати алгоритм послідовних наближень. Задати коефіцієнт швидкості навчання ν , який з кожною епохою на n прикладів зменшується на показник l . Наприклад, якщо $\nu = 0,3$, а $l = 0,05$, то з різним коефіцієнтом швидкості навчання буде виконано 6 епох, на кожній з яких буде n коригувань ваг одного з нейронів. Початкові вагові коефіцієнти задані в табл. 4.3, тобто структурно мережа складається з одного шару нейронів (7 входів; 4 виходи). Ваги також можна ініціалізувати випадковим чином. У якості вхідних змінних використати множину $x = \{x_1, \dots, x_7\}$, інформація про змінну x_8 не використовується для навчання, але буде задіяна в ході кластеризації. Розподіл прикладів здійснюється за 4 кластерами. Після навчання нейромережі виконати кластерний аналіз (табл. 4.4).

Таблиця 4.3 – Початкові вагові коефіцієнти мережі Кохонена

Кластер j	Вагові коефіцієнти w_{ij}						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,06	0,06	0,21	0,52	0,36	0,55	0,57
2	0,00	1,00	0,50	0,80	0,80	0,80	0,73
3	1,00	0,00	0,04	0,48	0,26	0,22	0,42
4	1,00	0,99	0,69	0,77	0,79	0,78	0,81

Таблиця 4.4 – Результати кластерного аналізу

Кластер	Розмір кластеру	Стать	Отрим. всі заліки	Середній рейтинг	Середній коеф. стипендії	Опис кластеру
1						
2						
3						
4						

Задача 3. Розробити багатoshаровий перцептрон або згорткову нейромережу, які дозволять розв'язати задачу реалізації певної логічної функції від трьох змінних (табл. 4.5). Вибір варіанта завдання здійснити за формулою $N = (L + [\text{остання цифра року}]) \bmod 7$, де L – варіант студента за списком. Вихід логічної функції згенерувати випадковим чином.

Таблиця 4.5 – Варіанти завдань

№	Функція
1	Кон'юнкція
2	Диз'юнкція
3	Імплікація
4	Еквівалентність
5	Нерівнозначність
6	Стрілка Пірса
7	Штих Шеффера

Задачу навчання мережі розв'язати, використавши методи оптимізації з табл. 4.1. Вибір здійснити за виразом $N = (L + [\text{остання цифра року}]) \bmod 10$. Гіперпараметри нейромережі налаштувати самостійно.

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *Mathematica*, *Python Anaconda*: фреймворки *TensorFlow*, *Keras*, *Theano*, *Caffe* тощо [40–47]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива розвитку штучного інтелекту (спадний та висхідний напрями) та машинного навчання. Видатні вчені.
2. Зв'язок біології та комп'ютерної науки на прикладі нейронних мереж.
3. Структура головного та спинного мозку.
4. Пам'ять та увага. Емоції.
5. Структура біологічного нейрона.
6. Математична модель штучного нейрону.
7. Які задачі дозволяє розв'язати штучна нейронна мережа?
8. Класифікація нейронних мереж і їх властивості.
9. Поняття оптимізації, екстремуму та похідної.
10. Навчання нейромережі. Недонавчання та перенавчання.
11. Застосування навченої нейронної мережі.
12. Архітектура нейронних мереж Хопфілда, Коско, Хеммінга-Ліппмана, Кохонена, багатошаровий перцептрон.
13. Принцип «*Winner Takes All*» («Переможець отримує все»).
14. Глибоке машинне навчання. Згорткова нейромережа.
15. Операції згортки та пуллінгу, дропауту.
16. Алгоритми навчання нейромереж Хопфілда, Коско, Хеммінга-Ліппмана, Кохонена.
17. Ланцюгове правило (*chain rule*) диференціювання складних функцій.
18. Алгоритм зворотного поширення похибки.
19. Особливості нейромережових оптимізаторів.
20. Інерційні (метод моментів, Нестерова) та адаптивні (*Adagrad*, *ADAM*, *Adamax* тощо) методи оптимізації.
21. Особливості стохастичної оптимізації нейромереж. Загальна метаевристична схема.
22. Практичне застосування методів навчання нейромереж.
23. Як вибрати «найкращий» оптимізатор? Теорема про відсутність безкоштовних сніданків.
24. Артистичні нейронні мережі.
25. Ігрові нейроконтролери.
26. Апаратна база для роботи з нейромережами. Процесори та відеокарти.
27. Особливості використання нейромереж в пакетах математичного моделювання та середовищах програмування.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА



1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки та техніки. Структура та правила оформлювання / В. Земцева, Ю. Поліщук, Р. Санченко, Л. Шрамко, А. Ямчук. — К. : ДП Український інститут науково-технічної і економічної інформації, 2016. — 31 с.
2. ДСТУ ГОСТ 7.1:2006. Бібліографічний запис, бібліографічний опис. Загальні вимоги та правила складання / О. К. Галевич, І. М. Штогрин. — Львів, 2008. — 20 с.
3. Henderson Н. Encyclopedia of Computer Science and Technology / Н. Henderson. — New-York, 2003. — 593 p.
4. Эндрю А. М. Искусственный интеллект / Эндрю А. М. — М. : Мир, 1985. — 267 с.
5. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норvig. — М. : Вильямс, 2006. — 1408 с.
6. Хокинг С. Мир в ореховой скорлупке / Хокинг С. — СПб. : Амфора, 2007. — 218 с.
7. Пенроуз Р. Новый ум короля: о компьютерах, мышлении и законах физики / Пенроуз Р. — М. : Едиториал УРСС, 2003. — 384 с.
8. Нікольський Ю. В. Системи штучного інтелекту : навчальний посібник / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина. — Л. : Магнолія, 2013. — 279 с.
9. Джонс М. Т. Программирование искусственного интеллекта в приложениях / Джонс М. Т. — М. : ДМК Пресс, 2011. — 312 с.
10. Іванов Ю. Ю. Методи штучного інтелекту: лекції, алгоритми та задачі [Електронний ресурс] / Іванов Ю. Ю. — Вінниця, 2020. — 112 с. — Режим доступу : https://iq.vntu.edu.ua/method/read_url.php?id=745450&tbl_num=2&card_id=1071.
11. Орлова И. В. Экономико-математические методы и модели: компьютерное моделирование. — М. : Вузовский учебник, 2007. — 365 с.

12. Штовба С. Д. Інтелектуальні технології ідентифікації залежностей / С. Д. Штовба, В. В. Мазуренко. — Вінниця : ВНТУ, 2014. — 113 с.
13. Снитюк В. Е. Прогнозирование. Модели, методы, алгоритмы: учебное пособие / Снитюк В. Е. — К. : «Маклаут», 2008. — 364 с.
14. Крилик Л. В. Інтерполяція та апроксимація табличних даних / Л. В. Крилик, І. В. Богач, М. О. Прокопова. — Вінниця : ВНТУ, 2013. — 110 с.
15. Chapra S.C. Numerical Methods for Engineers / S.C. Chapra, R.P. Canale. — New-York: McGraw-Hill, 2010. — 994 с.
16. Хемминг Р.У. Численные методы для научных работников и инженеров / Р.У. Хемминг. — М.: Наука, 1972. — 400 с.
17. Пантелеев А. В. Методы оптимизации в примерах и задачах / А. В. Пантелеев, Т. А. Летова. — М. : Высш. шк., 2005. — 544 с.
18. Жалдак М. І. Основи теорії і методів оптимізації / М. І. Жалдак, Ю. В. Триус. — Черкаси : Брама-Україна, 2005. — 608 с.
19. Mathews J.H. Numerical Methods Using MATLAB / J.H. Mathews, K.D. Fink. — Prentice Hall, 1999. — 336 p.
20. Многометрическая оптимизация. Методы многомерной оптимизации [Электронный ресурс]. — Режим доступа : <https://www.intuit.ru/studies/courses/1020/188/lecture/4931?page=2>.
21. Jamil M. A Literature Survey of Benchmark Functions For Global Optimization Problems / M. Jamil, Xin-She Yang // International Journal of Mathematical Modelling and Numerical Optimization. — 2013. — V. 4. — P. 150–194.
22. StatSoft. Электронный учебник по статистике [Электронный ресурс] / Деревья классификации. — Режим доступа : <http://www.statsoft.ru/home/textbook/modules/stclatre.html>.
23. Дюк В. Data Mining: учебный курс / В. Дюк, А. Самойленко. — СПб. : Питер, 2001. — 368 с.
24. BaseGroup Labs. Технологии анализа данных [Электронный ресурс] / Деревья решений. — Режим доступа : http://www.basegroup.ru/library/analysis/tree/math_cart_part1/.
25. Quinlan R. Induction of Decision Trees // Machine Learning 1 / R. Quinlan. — Boston: Kluwer Academic Publishers, 1986. — P. 81-106.
26. Breiman L. Classification and Regression Trees / L. Breiman, J. Friedman, R. Olshen, C. Stone. — Wadsworth, Belmont, CA, 1984. — 400 p.
27. Lippmann R. An Introduction to Computing with Neural Nets / R. Lippmann // IEEE ASSP Magazine. — V. 4. — 1987. — P. 4-22.
28. Алгоритм розпізнавання та коригування пошкоджених QR-кодів / Р. Н. Кветний, В. В. Півошенко, Ю. Ю. Іванов, А. С. Васюра // Інформаційні технології та комп'ютерна інженерія. — Вінниця : ВНТУ, 2019. — № 2. — С. 25–32.
29. Каллан Р. Основные концепции нейронных сетей / Каллан Р. — М. : Вильямс, 2001. — 287 с.

30. Люгер Д. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем / Люгер Д. Ф. — М. : Вильямс, 2003. — 864 с.
31. Fausett L. Fundamentals of Neural Networks: Architectures, Algorithms, and Applications / L. Fausett. — USA: Prentice-Hall Inc, 1994. — 476 p.
32. Портал искусственного интеллекта [Электронный ресурс] / Нейронные сети. — Режим доступа : <http://neuronus.com/>.
33. Осовский С. Нейронные сети для обработки информации / Осовский С. — М. : Финансы и статистика, 2002. — 344 с.
34. Хайкин С. Нейронные сети: полный курс / Хайкин С. — М. : Вильямс, 2008. — 1104 с.
35. Рашид Т. Создаем нейронную сеть / Рашид Т. — М. : Издательство «Вильямс», 2017. — 272 с.
36. Deep Learning Tutorial. — Montreal (Canada): LISA Lab, University of Montreal, 2015. — 173 p.
37. Гудфеллоу Я. Глубокое обучение / Я. Гудфеллоу, А. Курвилль, И. Бенджио. — М. : ДМК Пресс, 2017. — 654 с.
38. Prakash J. Back-Propagation is very simple. Who made it Complicated? / J. Prakash. — 2017. — 29 p.
39. Gradient Descent Optimization Algorithms [Web resource]. — Access mode: <https://www.kdnuggets.com/2019/06/gradient-descent-algorithms-cheat-sheet.html>.
40. Ануфриев И. Е. MATLAB 7 / И. Е. Ануфриев, А. Б. Смирнов, Е. Н. Смирнова. — СПб. : БХВ-Петербург, 2005. — 1104 с.
41. MATLAB Programming Fundamentals [Web Resource]. — Massachusetts: MathWorks, 2021. — Access mode : https://www.mathworks.com/help/pdf_doc/matlab/matlab_prog.pdf.
42. Бринк Х. Машинное обучение / Х. Бринк, Дж. Ричардс, М. Феверолф. — СПб. : Питер, 2017. — 336 с.
43. Рашка С. Python и машинное обучение / Рашка С. — М. : ДМК Пресс, 2017. — 418 с.
44. Лутц М. Программирование на Python : 4-е издание / Лутц М. — СПб. : Символ-Плюс, 2010. — 1280 с.
45. Центр документации Wolfram Mathematica [Web Resource] / Wolfram Language & System. — Access mode : <http://reference.wolfram.com/language/>.
46. Numerical Recipes in C: The Art of Scientific Computing / W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. — Cambridge: Cambridge University Press, 2002. — 949 p.
47. Макаров Е. Г. Инженерные расчеты в Mathcad 15. Учебный курс / Макаров Е. Г. — СПб. : Питер, 2011. — 400 с.

ОЦІНЮВАННЯ СТУДЕНТАМИ МАТЕРІАЛІВ КУРСУ ТА РІВНЯ ВИКЛАДАННЯ

Після закінчення курсу необхідно дати відповіді на задані запитання, вписавши свої коментарі.

1. Наскільки Ви задоволені змістом дисципліни в цілому? Пропозиції до наповнення змісту курсу та критика.

Коментар

2. Як Ви оцінюєте якість викладання навчального матеріалу викладачем? Що Ви запропонували б змінити в методичному і змістовному плані для вдосконалення викладання цієї дисципліни?

Коментар

3. Що Ви вважаєте найскладнішим під час вивчення курсу (тема, задача, метод тощо)?

Коментар

4. Чи вважаєте Ви, що отримані знання будуть корисними у подальшому під час навчання / роботи? Що було найбільш корисним з точки зору подальшого навчання та / або застосування у практичній діяльності?

Коментар

*Електронне навчальне видання
комбінованого використання
Можна використовувати в локальному та мережному режимах*

**Методичні вказівки
до виконання лабораторних та практичних робіт
з дисципліни «Теорія штучного інтелекту»
для студентів спеціальності 126 –
«Інформаційні системи та технології»**

Укладач *Іванов Юрій Юрійович*

Рукопис оформив *Ю. Іванов*

Редактор *О. Ткачук*

Оригінал-макет виготовив *Г. Багдасар'ян*

Підписано до видання 08.07.2021 р.
Гарнітура Times New Roman.
Зам. № P2021-017.

Видавець та виготовлювач
Вінницький національний технічний університет,
інформаційний редакційно-видавничий центр.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95,
м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.