

Міністерство освіти і науки України  
Державний вищий навчальний заклад  
«Приазовський державний технічний університет»  
Кафедра біомедичної інженерії

Верескун М.В.

## **МАШИНЕ НАВЧАННЯ**

Методичні вказівки  
до самостійного вивчення дисципліни  
«Технології програмування»  
частина 2 «Навчання машин та штучний інтелект»  
для студентів напрямку 163 «Біомедична інженерія»  
усіх форм навчання



**Розроблено в рамках проекту «Erasmus+ (CBHE) BioArt: «Інноваційна мультидисциплінарна навчальна програма зі штучних імплантів для біоінженерії для рівнів бакалавр та магістр»**

**Developed in the frame of project «Erasmus+ (CBHE) BioArt: Innovative Multidisciplinary Curriculum in Artificial Implants for Bio-Engineering BSc/MSc Degrees» (586114-EPP- 1-2017- 1-ES- EPPKA2-CBHE- JP)**

Маріуполь  
2019

УДК 004.423(076)

Навчання машин та штучний інтелект: методичні вказівки до самостійного вивчення дисципліни «Технології програмування» ч. 2 «Навчання машин та штучний інтелект» для здобувачів освітнього рівня бакалавра за напрямом 163 «Біомедична інженерія» усіх форм навчання / уклад. М.В. Верескун. – Маріуполь : ПДТУ, 2019. – 24 с.

Методичні вказівки до виконання лабораторних робіт з дисципліни «Технології програмування» ч. 2 «Навчання машин та штучний інтелект» призначені для студентів напряму 163 «Біомедична інженерія» та спрямовані на опанування теоретичних і практичних знань у вивченні методів штучного інтелекту, зокрема машинного навчання.

Укладач М.В. Верескун, докт. екон. наук, доцент

Рецензент Д.С. Міроненко, канд. техн. наук,

Рекомендовано  
на засіданні кафедри біомедичної інженерії,  
протокол № 21 від 24 чеврня 2019 р.

Затверджено  
на засіданні методичної комісії факультету інформаційних  
технологій,  
протокол № 10 від 24 червня 2019 р.

© ДВНЗ «ПДТУ», 2019

© М.В. Верескун, 2019

## ЗМІСТ

ВСТУП.....	4
1 ЗАГАЛЬНІ МЕТОДИЧНІ РЕКОМЕНДАЦІЇ САМОСТІЙНОГО ВИВЧЕННЯ ДИСЦИПЛІНИ.....	3 6
2 МЕТОДИЧНІ ВКАЗІВКИ ДО САМОСТІЙНОГО ВИВЧЕННЯ ТЕОРЕТИЧНОГО МАТЕРІАЛУ.....	7
3 РЕКОМЕНДАЦІЇ З САМОСТІЙНОГО ВИКОНАННЯ ПРАКТИЧНИХ ЗАВДАНЬ.....	15
3.1 Вступ.....	15
3.2 Основні принципи.....	15
3.4 Завдання для індивідуальної роботи.....	22
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	23

## ВСТУП

Метою викладання дисципліни «Технології програмування» в ч. 2 «Навчання машин та штучний інтелект» є формування у студентів теоретичних знань і навичок при вивченні окремих галузей штучного інтелекту, зокрема, машинного навчання. Подання студентам широкого кола методів та алгоритмів у контексті машинного сприйняття та навчання, вивчення основ теорії навчання машин, включаючи дискримінантний, кластерний і регресійний аналіз. Підготувати студента до ефективного використання сучасних методів машинного навчання у подальшій професійній діяльності; допомогти оволодіти практичними навичками вирішення завдань інтелектуального аналізу даних за допомогою сучасних програмних засобів.

Основними завданнями дисципліни є:

- набуття теоретичних знань та практичних умінь з формування базового уявлення про галузі застосування систем штучного інтелекту;
- набуття вмінь і навичок розв'язання задач з використанням систем штучного інтелекту;
- опанування теоретичних і практичних питань створення та застосування систем штучного інтелекту, експертних систем; вивчення механізмів обробки і подання знань в інтелектуальних системах.

В результаті вивчення дисципліни студент повинен отримати наступні елементи компетентностей:

**Знання про:**

- тенденції і перспективи розвитку систем штучного інтелекту;
- принципи побудови та технологію розробки систем штучного інтелекту;
- моделі та методи розв'язання задач у слабоформалізованих галузях;
- основні поняття інженерії знань;
- моделі обробки та подання знань;

- принципи побудови нейронних мереж і підходи до навчання в нейронних мережах;

### **Вміння:**

- використовувати системи штучного інтелекту для розв'язання прикладних задач у різних предметних галузях;
- проектувати системи штучного інтелекту, експертні системи, бази знань;
- застосовувати методи розв'язання задач у слабоформалізованих галузях;
- формалізувати знання за допомогою різних способів їх подання;
- використовувати різні методи навчання.

### ***Способи мислення:***

- теоретичне мислення - це вид мислення, який полягає у пізнанні законів, правил. Воно відображає істотне у явищах, об'єктах, зв'язках між ними на рівні закономірностей і тенденцій;
- абстрактне (словесно-логічне) мислення – відбувається з опорою на поняття, судження, з допомогою логіки, не використовуючи емпіричних даних;
- аналітичне мислення - це вид мислення, який розгорнутий у часі, має чітко виражені етапи, достатньою мірою усвідомлені суб'єктом;
- репродуктивне мислення - це вид мислення, який забезпечує розв'язання задачі, спираючись на відтворення вже відомих людині способів.

### ***Професійні, світоглядні і громадянські якості:***

- здатність до абстрактного мислення, аналізу та синтезу на відповідних рівнях;
- здатність застосовувати знання у практичних ситуаціях;
- знання та розуміння предметної області та професійної діяльності;

- навички використання інформаційних і комунікаційних технологій;
- здатність вчитися і оволодівати сучасними знаннями;
- здатність до пошуку, оброблення та узагальнення інформації з різних джерел;
- здатність працювати в команді та особисто;
- навички міжособистісної взаємодії;
- здатність розробляти та управляти проектами;
- навички здійснення безпечної діяльності;
- здатність оцінювати та забезпечувати якість виконуваних робіт;

## 1 ЗАГАЛЬНІ МЕТОДИЧНІ РЕКОМЕНДАЦІЇ З САМОСТІЙНОГО ВИВЧЕННЯ ДИСЦИПЛІНИ

Завдання для самостійної роботи призначене для підготовки і виконання індивідуального (самостійного домашнього завдання) для підвищення ефективності, поглиблення знань і кращого засвоєння основних понять курсу.

На самостійну роботу студента денної форми навчання відводиться 21 година, розподіл за видами наведено у таблиці 1.1.

Таблиця 1.1 – Розрахунок часу для самостійної роботи студента за видами

№ з/п	Види роботи	Кількість годин
1	Опрацювання програмного матеріалу, що викладається на лекціях та тем самостійного опрацювання	6
3	Підготовка до практичних занять	6
4	Виконання індивідуальних завдань	9
	<b>Разом</b>	<b>21</b>

Для виконання самостійного домашнього завдання необхідно спочатку ознайомитись з теоретичним матеріалом, що наведено у підрозділах 3.1 – 3.6.

## **2 МЕТОДИЧНІ ВКАЗІВКИ ДО САМОСТІЙНОГО ВИВЧЕННЯ ТЕОРЕТИЧНОГО МАТЕРІАЛУ**

Самостійне вивчення теоретичного матеріалу дисципліни «Технології програмування» передбачає самостійне ознайомлення з основними ключовими аспектами, що наведені нижче. До кожної теми наведено рекомендовану літературу, яка найбільш повно розкриває внутрішній зміст теми.

### **1. Вступ до навчання машин. Базові поняття навчання машин.**

Завдання машинного навчання. Проблеми машинного навчання: класифікації та розмітки даних, реконструкції структури даних, прогнозування і часових рядів, регресії та безперервних даних. Базові поняття Big Data, Business Intelligence, Data Mining, Data Science, Machine Learning, Artificial Intelligence.

### **2. Стратегії та методи навчання машин.**

Стратегії індуктивного та дедуктивного навчання. Контрольоване навчання, або навчання з учителем (supervised learning); неконтрольоване навчання (unsupervised learning), або навчання без учителя; навчання з підкріпленням (reinforcement learning). Інші методи навчання: активне, багатозадачне, різноманітне, трансферне. Глибоке навчання, як метод поєднання алгоритмів навчання з вчителем і без вчителя.

### **3. Загальні положення систем штучного інтелекту.**

Базові поняття штучного інтелекту. Інтелектуальні задачі. Різні трактування інтелектуальної діяльності. Визначення штучного інтелекту за М. Мінським. Тест Тьюринга. Біологічні та соціальні моделі інтелекту. Агенти. Галузі застосування штучного інтелекту. Класифікація систем штучного інтелекту.

### **4. Апарат штучних нейронних мереж.**

Структура й властивості штучного нейрона. Класифікація нейронних мереж і їх властивостей. Топології нейронних мереж.



Прості одношарові мережі. Персептрон Розенблата. Функції активації. Процес навчання нейронних мереж. Навчання з вчителем. Застосування навченої нейронної мережі. Основні функціональні можливості програм моделювання нейронних мереж. Застосування штучних нейронних мереж.

#### **5. Використання різних видів нейронних мереж. Генетичні алгоритми.**

Топології багатошарових нейронних мереж. Алгоритми навчання мережі. Алгоритм зворотнього поширення помилки. Критерії адекватності та оцінки моделей MLP. Архітектура мереж Хопфілда. Принцип роботи мереж Хопфілда. Алгоритм навчання мережі Хопфілда. Класифікація асоціативних мереж. Мережі ВАР. Принцип роботи мережі ВАР. Алгоритм навчання мережі ВАР. Рішення задач ідентифікації. Генетичні алгоритми.

# 3 РЕКОМЕНДАЦІЇ З САМОСТІЙНОГО ВИКОНАННЯ ПРАКТИЧНИХ ЗАВДАНЬ

## 3.1 Вступ

На даний момент в світі бурхливо розвивається нова прикладна область математики, що спеціалізується на штучних нейронних мережах. Актуальність досліджень в цьому напрямку обумовлена можливостями застосування нейронних мереж в самих різних областях.

Широта завдань, що вирішуються нейронними мережами (НС), пояснюється багато в чому тим, що можливість навчання дозволяє зробити функціонування системи на порядок ефективнішим.

Нейронні мережі можуть сильно відрізнятись один від одного, проте у них є кілька спільних рис. Основу кожної нейронної мережі складають прості, найчастіше однотипні, елементи (комірки), що імітують роботу нейронів мозку (аксони і дендрити). Нейрон може визначатися своїм поточним станом за аналогією з нервовими клітинами головного мозку, які можуть бути порушені або загальмовані. Нейрон має групу синапсів вхідні зв'язків, з'єднаних з виходами інших нейронів, а також має аксон - вихідну зв'язок даного нейрона, з якої сигнал (збудження або гальмування) надходить на синапси наступних нейронів

## 3.2 Основні принципи

Розпізнавання тексту є однією з актуальних задач нейронних мереж. Для прикладу розглянуто розпізнавання цифр, набір еталонних образів приведено нижче. Прикладом такого набору є послідовність з десяти цифр від 0 до 9. У цьому прикладі число образів  $M = 10$ . У разі, коли кожен клас образів характеризується лише своїм еталоном, маємо число класів, також рівне  $M$ .

Кожен образ формується у вигляді графічного файлу в бітовому форматі. Тип файлу (розширення) визначається використовуваними в середовищі MATLAB типами графічних файлів. Рекомендується використовувати розширення *tif*.

Для створення графічних файлів образів зручно використовувати середу «Adobe Photoshop». В цьому випадку при створенні кожного файлу необхідно виконати наступну послідовність операцій:

1) створити новий файл, задавши його параметри:

- ім'я: XXXX;
- ширина: N1 пікселів;
- висота: N2 пікселів;
- колірний режим: бітовий формат.

Значення  $N1 = 10$ ,  $N2 = 12$ .



Рисунок 3.1 – Приклади графічних символів цифр

Створення і навчання НС у середовищі MATLAB.

На даному етапі виконання роботи в середовищі MATLAB проводиться за допомогою програми *sg\_newff*, яка реалізує наступні функції:

- формування числових масивів еталонних образів, використовуваних в якості навчальних;
- підготовка даних, необхідних для створення НС;
- створення НС, завдання параметрів навчання НС і навчання НС.

Еталонний образ кожного символу представлений у вигляді вектора-стовпця  $[N, 1]$ , число елементів  $N$  якого дорівнює числу ознак (інакше кажучи,  $N$  – розмірність простору ознак). Такий вектор-стовпець формується з двовимірного

масиву-зображення  $[N1, N2]$ , який, в свою чергу, формується при зчитуванні графічного файлу образу за допомогою команд:

`imread (FILENAME)` – процедура читання графічного файлу;

`X = reshape (A, [N, 1])` – процедура перетворення двовимірного масиву  $A [N1, N2]$  в одновимірний вектор-стовпець  $X [N, 1]$ , де  $N = N1 * N2$ .

Процедура множення масиву на 1 призводить до зміни типу елементів масиву з `logical` (для елементів бітового формату) на `double`.

Для задовільної роботи НС недостатньо формування лише одного навчального образу для кожного класу (типу символу) образів. Це пов'язано з тим, що розпізнаються образи (на етапі роботи НС в режимі розпізнавання) завжди відрізняються від навчальних по ряду причин:

- відмінність шрифтів і стилів друкованих символів;
- похибки сканування і неточності суміщення символу і вікна сканування;
- низька якість друку, дефекти паперу і т. д.

В силу зазначених причин для надійного розпізнавання образів НС слід навчати на досить представницькому безлічі образів, що входять в один і той же клас.

У програмі `sg_newff` формування додаткових навчальних образів проводиться шляхом незначного спотворення еталонних образів, зчитувальних з графічних файлів. Спотворення образу-еталона кожного класу реалізується шляхом додавання до нього рівномірного (по площі зображення) шуму типу «Сіль і перець», що представляє собою випадкове спотворення окремих пікселів зображення. Ступінь спотворення характеризується числом  $p = [0; 1]$ , що визначає частку перевернутих пікселів.

Такий підхід при формуванні образів дозволяє, по-перше, швидко отримувати велике число навчальних образів, і, по-друге, регулювати (шляхом зміни значення  $p$ ) ступінь розкиду безлічі образів в межах одного класу.

Підготовка даних, необхідних для створення НС, включає в себе:

1) формування двовимірного масиву навчальних образів  $XR [N, K]$ , кожен стовпець якого є набором  $N$  ознак одного образу, а число стовпців  $K$  дорівнює числу навчальних образів;

2) формування двовимірного масиву бажаних відгуків  $YR [NY, K]$ , де  $NY$  – число виходів НС (тобто, число нейронів вихідного шару);  $K$  – число навчальних образів. Відгук  $YR[:, k]$  (в загальному випадку – вектор-стовпець) відповідає  $k$ -му навчальному образу – вектору  $XR[:, k]$ ;

3) формування двовимірного масиву  $R [N, 2]$ , що визначає мінімальне  $R(n, 1)$  і максимальне  $R(n, 2)$  значення  $n$ -го ознаки,  $n = 1, \dots, N$ .

Створення НС. У загальному випадку НС net створюється за допомогою команди:

```
net = nnnnn (P1, P2, ... PL), де  
nnnnn – тип НС;  
P1, ..., PL – параметри НС.
```

У даній роботі використовується НС типу багатошарового перцептрона `newff`, яка задається командою:

```
net = newff (R, [A1 A2 ... AL], {F1 F2 ... FL}, BTF, PF),
```

де  $R$  – масив мінімальних і максимальних значень вхідних нейронів (ознак);

$A_i$  – число нейронів  $i$ -го шару, починаючи з першого прихованого шару,  $i = 1, \dots, L$ ;

$F_i$  – функція активації нейронів  $i$ -го шару, за замовчуванням 'tansig';

BTF – функція навчання мережі, за замовчуванням 'trainlm';

PF – критерій зупинки, за замовчуванням 'mse' (мінімум ско).

Додаткові параметри, які використовуються при створенні мережі:

```
net.performFcn = 'msereg' – навчання НС проводиться за допомогою методу регуляризації;
```

```
net.performParam.ratio = 0.1 – значення параметра регуляризації;
```

`net.trainParam.show = 5` – число епох, через яке проводиться висновок параметрів процедури навчання;

`net.trainParam.epochs = 500` – максимальне число епох при навчанні НС;

`net.trainParam.goal = 0.02` – значення цільової функції, після досягнення якого процес навчання НС припиняється.

Процес навчання НС запускається командою:

`net = train (net, XR, YR).`

Для вирішення завдань розпізнавання друкованих символів рекомендується використовувати тришарову НС (один прихований шар) з числом нейронів:

$N = 120$  – у вхідному шарі;

$A1 = 20$  – в прихованому (проміжному) шарі;

$A2 = 1$  – в вихідному шарі.

При використанні більшого числа нейронів процедура навчання НС може зайняти занадто багато часу. Для рекомендованих значень параметрів НС (в тому числі і додаткових) і загальній кількості навчальних образів (для всіх заданих класів)  $K = 100 \dots 200$  час навчання НС становить  $5 \dots 20$  хв.

Розпізнавання друкованих символів за допомогою навченої НС.

Робота НС, тобто формування відгуку  $Y$  при вхідній дії у вигляді вектора-стовпця  $X [N, 1]$  проводиться командою:

`Y = sim (net, X).`

У разі, коли бажаний відгук приймає цілочисельні значення, рекомендується використовувати округлення до найближчого цілого, тобто

`Y = round (sim (net, X)).`

Тестування роботи НС при розпізнаванні друкованих символів з різним ступенем спотворення проводиться за допомогою програми `sr_work`, вихідними даними для якої є:

`SX.tif` – ім'я графічного файлу образу-еталона;

`N` – число пікселів зображення образу;

`NT` – число тестованих образів, отриманих шляхом спотворення еталона;

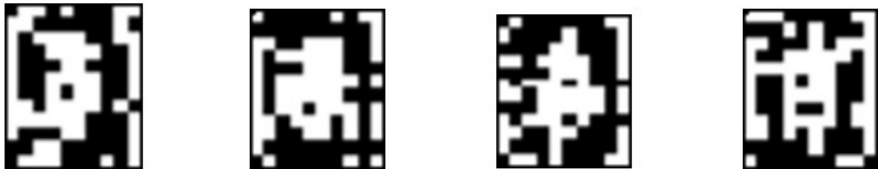
`P` – частка перекручених пікселів [0; 1].

На рис. 3.2 представлені деякі приклади розпізнавання символів, зображених на рис. 3.1, за допомогою навченої НС. Навчання проводилося при числі навчальних образів  $M = 10$  для кожного виду символу і параметрі спотворення символів  $p = 0,1$ .



Результат розпізнавання: «2» «3» «5»

Рисунок 3.2 – Невірні розпізнавання символу «0», спотвореного 20 % шуму «Сіль і Перець»



Результати розпізнавання: «0» «0» «0» «0»

Рисунок 3.3 – Правильні розпізнавання символу «0», спотвореного 20 % шуму «Сіль і Перець»



Результат розпізнавання: «3» «5» «6»

Рисунок 3.4 – Невірні розпізнавання символу «4», спотвореного 20 % шуму «Сіль і Перець»



Результати розпізнавання: «4» «4» «4» «4» «4»

Рисунок 3.5 – Правильні розпізнавання символу «4», спотвореного 20 % шуму «Сіль і Перець»



Результати розпізнавання: «8» «8» «6»

Рисунок 3.6 – Результати розпізнавання символу «8», спотвореного 10 % шуму «Сіль і Перець»

Результати розпізнавання символів, представлені на рис. 3.2 – 3.6, демонструють гарне розпізнавання за допомогою НС навіть при сильному спотворенні (параметр  $p > 0,1$ ). Для об'єктивної оцінки якості роботи НС необхідно обчислення імовірнісних характеристик розпізнавання. При правильному



виборі параметрів навчання мережі та використання не менше 100 навчальних образів можна отримати ймовірність правильного розпізнавання символів близько 0,6 ... 0,9 (в залежності від виду розпізнається символу) при параметрі спотворення  $p = 0,1 \dots 0,2$ .

### 3.4 Завдання для індивідуальної роботи

1) Підготувати графічні файли еталонних образів для символів, заданих викладачем.

2) У середовищі MATLAB створити і навчити НС, призначену для розпізнавання друкованих символів.

3) Дослідити залежність якості роботи НС від:

- ступеня спотворення символів (параметр  $p$ );

- числа нейронів в прихованому шарі.

Якість роботи НС характеризується можливостями правильної класифікації  $P_{np}$  способу  $i$ -го класу,  $i = 1, \dots, M$ . Оцінка ймовірностей  $\hat{P}_{np(i)}$  здійснюється за формулою:

$$\hat{P}_{np(i)} = \frac{N_{np}}{N_0}, \quad (3.1)$$

де  $N_{np}$  - число правильних розпознавань способу  $i$ -го класу;

$N_0$  – загальне число розпознавань образів  $i$ -го класу.

Число  $N_{np}$  визначається експериментально при запуску програми `sr_work` при значеннях  $N_0 = 10 \dots 100$ .

## СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

### Базові

1. Stephen Marsland. Machine Learning: An Algorithmic Perspective / Stephen Marsland. – 2015. – 452 p.,
2. Yaser S. Abu-Mostafa. Learning from data / Yaser S. Abu-Mostafa. – 2017. – 215 p.
3. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – 2016. – 800 p.
4. LISA lab, University of Montreal Deep Learning Tutorial. – 2015. – 173 p.
5. [Neil Wilkins](#) Artificial Intelligence: An Essential Beginner's Guide to AI, Machine Learning, Robotics, The Internet of Things, Neural Networks, Deep Learning, Reinforcement Learning, and Our Future Paperback. – Publisher: Bravex Publications, 2019, 112 p.
6. [Mariya Yao](#) Applied Artificial Intelligence: A Handbook For Business Leaders Kindle Edition, Publisher: TOPBOTS, 2018, 246 p.
7. [Richie Dorsey](#) Machine Learning for Beginners: A Complete Guide for Getting Started with Machine Learning Kindle Edition.- Publisher: Amazon.com Services LLC, 2019, 162 p.
8. [Svein Linge](#), [Hans Petter Langtangen](#) Programming for Computations - Python: A Gentle Introduction to Numerical Simulations with Python (Texts in Computational Science and Engineering Book 15) 1st ed. 2016 Edition, Kindle Edition. – Publisher: Springer; 1st ed. 2016 edition (July 25, 2016), 232 pages
9. [Stephen Richard](#) Data Analysis from Scratch with Python: The Complete Beginner's Guide for Machine Learning Techniques and A Step By Step NLP using Python Guide To Expert (Including Programming Interview Questions) Kindle Edition.- Publisher: Amazon.com Services LLC, 2019, 134 p.

### Допоміжні

1. Tom M. Mitchell. Machine Learning [Електронний ресурс] / Tom M. Mitchell. – Режим доступу: <http://www.cs.cmu.edu/~tom/mlbook.html>
2. Feldman, R. The text mining handbook: advanced approaches in analyzing unstructured data [Текст] / R. Feldman, J. Sanger. – Cambridge University Press, 2007. – 410 p.
3. Люгер, Д. Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / Д. Люгер. – Издательский дом «Вильямс», 4е изд. М.: – 2003. – 864 с.
4. Рассел, С. Искусственный интеллект. Современный подход [Текст] / С. Рассел, П. Норвиг, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.
5. Bezdek, J.C. Pattern Recognition with Fuzzy Objective Function Algorithms [Текст] / J.C. Bezdek // N.Y.: Plenum Press, 1981. – 272 p.
6. Люгер, Дж.О. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж.О. Люгер. - М.: Диалектика, 2016. - 864 с.
7. Нильсон, Н. Принципы искусственного интеллекта / Н. Нильсон. - М.: Радио и связь, 2015. - 373 с.
8. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. - М.: Вильямс, 2016. - 578 с.
9. Слэйгл, Дж. Искусственный интеллект / Дж. Слэйгл. - М.: Мир, 2016. - 320 с.
10. Тей, А. Логический подход к искусственному интеллекту / А. Тей, П. Грибомон, и др.. - М.: Мир, 2015. - 432 с.
11. Акинин, М.В. Нейросетевые системы искусственного интеллекта в задачах обработки изображений / М.В. Акинин, М.Б. Никифоров, А.И. Таганов. - М.: ГЛТ, 2016. - 152 с.
12. Raheem, N. (2019). Big Data : A Tutorial-Based Approach (Vol. First edition). Boca Raton: Chapman and Hall/CRC.
13. Акинин, М. В. Нейросетевые системы искусственного интеллекта в задачах обработки изображений / М.В. Акинин, М.Б. Никифоров, А.И. Таганов. - М.: РиС, 2016. - 152 с.

### **Інформаційні ресурси**

14. Towards Data Science, <https://towardsdatascience.com/>
15. DataSet, <https://www.kaggle.com/kernels>.
16. Scikit-learn tutorial: statistical-learning for scientific data processing, <http://gael-varoquaux.info/scikit-learn-tutorial/>
17. Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>, 2015. Online; accessed 11-December-16
18. Andrew Moore. Statistical Data Mining Tutorials [<http://www.autonlab.org/tutorials/>]
19. Ускоренный курс машинного обучения с API TensorFlowhttps, <http://developers.google.com/machine-learning/crash-course/>

***This project has been funded with support from the European Commission. This publication / communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained there in.***