ШТУЧНИЙ ІНТЕЛЕКТ

Штучний інтелект (ШІ) — розділ комп'ютерної лінгвістики та інформатики, що опікується формалізацією проблем та завдань, які подібні до дій, що виконує людина. Здатність інженерної системи обробляти, застосовувати та вдосконалювати здобуті знання та вміння.

Штучний інтелект був заснований як академічна дисципліна в 1956 році, і за роки, що минули, він пережив кілька хвиль оптимізму, за якими слідували розчарування та втрата фінансування (відома як « [зима штучного інтелекту](https://en.wikipedia.org/wiki/AI_winter) »), а потім нові підходи, успіх і відновлення фінансування. Дослідження штучного інтелекту випробували та відкинули багато різних підходів, включаючи симуляцію мозку, [моделювання вирішення людських проблем](https://en.wikipedia.org/wiki/Symbolic_AI#Cognitive_simulation), [формальну логіку](https://en.wikipedia.org/wiki/Symbolic_AI#Logic-based), [великі бази даних знань](https://en.wikipedia.org/wiki/Symbolic_AI#Knowledge-based_systems) та імітацію поведінки тварин.

Різні підгалузі досліджень ШІ зосереджені навколо конкретних цілей і використання конкретних інструментів. Традиційні цілі досліджень штучного інтелекту включають:

* Міркування – використання стратегії, вирішення головоломок та вміння робити судження в умовах [невизначеності](https://en.wikipedia.org/wiki/Uncertainty).
* Представлення знань – представлення [інформації](https://en.wikipedia.org/wiki/Information) про [світ](https://en.wikipedia.org/wiki/World) у формі, яку комп’ютерна система може використовувати для вирішення складних завдань.
* Планування (планування штучного інтелекту) – галузь [штучного інтелекту](https://en.wikipedia.org/wiki/Artificial_intelligence), яка стосується реалізації [стратегій](https://en.wikipedia.org/wiki/Strategy) або послідовностей дій
* Вміння вчитись.
* Обробка природної мови ( NLP ) - здатність «розуміти» зміст документів, включаючи [контекстуальні](https://en.wikipedia.org/wiki/Context_(language_use)) нюанси мови в них.
* Вміння відчувати (бачити, чути і т.п.) та діяти (пересувати та маніпулювати об’єктами).

У [штучному інтелекті](https://en.wikipedia.org/wiki/Artificial_intelligence) інтелектуальний агент ( ІА ) —[сприймає](https://en.wikipedia.org/wiki/Machine_perception) навколишнє середовище, виконує дії [автономно](https://en.wikipedia.org/wiki/Autonomous), щоб досягти цілей, і може покращити свою продуктивність шляхом [навчання](https://en.wikipedia.org/wiki/Machine_learning) або отримання [знань](https://en.wikipedia.org/wiki/Knowledge_representation). Інтелектуальний агент може бути простим або складним: [термостат](https://en.wikipedia.org/wiki/Thermostat) або інша [система керування](https://en.wikipedia.org/wiki/Control_system) вважається прикладом інтелектуального агента, як і [людина](https://en.wikipedia.org/wiki/Human_being), як і будь-яка система, яка відповідає цьому визначенню, наприклад [фірма](https://en.wikipedia.org/wiki/Firm) або [держава](https://en.wikipedia.org/wiki/State_(polity)).

Типи ШІ

Оскільки дослідження штучного інтелекту мають на меті змусити машини імітувати функціонування, подібне до людини, ступінь, до якого система штучного інтелекту може відтворювати людські можливості, використовується як критерій для визначення типів штучного інтелекту.

Один тип заснований на класифікації штучного інтелекту та машин із підтримкою штучного інтелекту на основі їхньої схожості з людським розумом і здатності «мислити» і, можливо, навіть «відчувати» як люди. Згідно з цією системою класифікації існує чотири типи ШІ або систем на основі ШІ:

1. Реактивні машини – Це найстаріші форми систем ШІ з надзвичайно обмеженими можливостями. Вони імітують здатність людського розуму реагувати на різні види стимулів. Ці машини не мають функцій пам’яті. Це означає, що такі машини не можуть використовувати раніше отриманий досвід для інформування про свої поточні дії, тобто ці машини не мають здатності «навчатися».
2. Машини з обмеженою пам’яттю – це машини, які, окрім можливостей суто реактивних машин, також здатні навчатися на історичних даних для прийняття рішень. Майже всі існуючі програми, про які ми знаємо, належать до цієї категорії ШІ. Усі сучасні системи штучного інтелекту, наприклад ті, що використовують глибоке навчання, навчаються великими обсягами навчальних даних, які вони зберігають у своїй пам’яті, щоб сформувати еталонну модель для вирішення майбутніх проблем.
3. Теорія розуму – теоретично, ШІ рівня розуму зможе краще зрозуміти сутності, з якими він взаємодіє, розпізнаючи їхні потреби, емоції, переконання та процеси мислення. Досягнення теорії розумового рівня штучного інтелекту також потребуватиме розвитку в інших галузях штучного інтелекту. Це пояснюється тим, що для того, щоб по-справжньому зрозуміти людські потреби, машини зі штучним інтелектом повинні сприймати людей як тих, чий розум може бути сформований багатьма факторами, по суті, «розуміючи» людей.
4. Самосвідомий ШІ – Це останній етап розвитку штучного інтелекту, який наразі існує лише гіпотетично. Штучний інтелект, що усвідомлює себе, який, зрозуміло, є штучним інтелектом, який настільки схожий на людський мозок, що розвинув самосвідомість. Створення такого типу штучного інтелекту, до реалізації якого залишилися десятиліття, якщо не століття, є і завжди буде кінцевою метою всіх досліджень ШІ.

Другий тип класифікації ШІ опирається на його можливості до навчання, саморозвитку та об’єму інформації, з якою він може працювати:

1. Вузький ШІ (ANI) – цей тип штучного інтелекту представляє весь існуючий ШІ, включаючи навіть найскладніший і потужний ШІ, який будь-коли створювався на сьогоднішній день. Штучний вузький інтелект відноситься до систем штучного інтелекту, які можуть виконувати певне завдання лише автономно, використовуючи людські можливості. Ці машини не можуть робити нічого більше, ніж те, на що вони запрограмовані, і тому мають дуже обмежений або вузький діапазон компетенцій. Згідно з вищезгаданою системою класифікації, ці системи відповідають усім ШІ з реактивною та обмеженою пам’яттю. Навіть найскладніший штучний інтелект, який використовує машинне та глибоке навчання для навчання, підпадає під ANI.
2. Загальний ШІ ( AGI ) — це тип гіпотетичного [інтелектуального агента](https://en.wikipedia.org/wiki/Intelligent_agent). Концепція AGI полягає в тому, що він може навчитися виконувати будь-яке інтелектуальне завдання, яке можуть виконати [люди](https://en.wikipedia.org/wiki/Human_beings) чи тварини. Ці системи зможуть самостійно створювати численні компетенції та створювати зв’язки та узагальнення між областями, значно скорочуючи час, необхідний для навчання. Це зробить системи штучного інтелекту такими ж спроможними, як і люди, відтворюючи наші багатофункціональні можливості. Загальний ШІ може згадуватись як сильний/потужний/повний ШІ.
3. Штучний супер інтелект (ASI). Розвиток штучного суперінтелекту, ймовірно, стане вершиною досліджень штучного інтелекту, оскільки AGI стане, безумовно, найефективнішою формою інтелекту на землі. ASI, окрім копіювання багатогранного інтелекту людей, буде надзвичайно кращим у всьому, що вони роблять, завдяки значно більшому об’єму пам’яті, швидшій обробці й аналізу даних, а також можливостям прийняття рішень.

Технології з підтримкою ШІ та машинного навчання вже використовуються в:

* [медицині](https://pubmed.ncbi.nlm.nih.gov/30617339/) – аналіз різних речовин та симуляція їх роботи;
* [транспорті](https://www.ifc.org/wps/wcm/connect/7c21eaf5-7d18-43b7-bce1-864e3e42de2b/EMCompass-Note-75-AI-making-transport-safer-in-Emerging-Markets.pdf?MOD=AJPERES&CVID=mV7VCeN) – автопілот;
* [робототехніці](https://scienceexchange.caltech.edu/topics/artificial-intelligence-research/autonomous-ai-cars-drones) – технології розумного будинку;
* [науці](https://www.ai4science.caltech.edu/) – проведення великої кількості обрахунків, симуляція об’єктів, від атомів до космічних тіл;
* [освіті](https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-019-0171-0);
* [збройних силах](https://www.rand.org/pubs/research_reports/RR3139-1.html) – системи автоматичного наведення, керування безпілотниками;
* [фінансах](https://scienceexchange.caltech.edu/topics/artificial-intelligence-research/artificial-intelligence-experts/yaser-abu-mostafa-finance-recession-artificial-intelligence) та [їх регулюванні](https://www.imf.org/en/News/Articles/2021/10/29/sp102921-ai-and-regtech) – аналіз графіків, створення прогнозів.
* [сільському господарстві](https://cmr.berkeley.edu/2020/03/ai-agriculture/) – керування технікою.
* [розвагах](https://arxiv.org/abs/1905.04175) – комп’ютерні ігри, «ШІ-художники»: midjourney, dall-e, stable-diffusion;
* роздрібній торгівлі;
* обслуговуванні клієнтів;
* [виробництві](https://www.sap.com/insights/what-is-a-smart-factory.html) – автоматизація.

МАШИННЕ НАВЧАННЯ

Машинне навчання (ML) – це підгалузь штучного інтелекту в галузі інформатики, яка часто застосовує статистичні прийоми для надання комп'ютерам здатності «навчатися» з даних, без того, щоби бути програмованими явно. Назву «машинне навчання» було започатковано 1959 року Артуром Семюелем.

Підходи до машинного навчання традиційно поділяються на три великі категорії:

1. [Контрольоване навчання](https://en.wikipedia.org/wiki/Supervised_learning): комп’ютеру надаються приклади вхідних даних і бажаних виходів, наданих «вчителем», і мета полягає в тому, щоб вивчити загальне правило, яке [відображає](https://en.wikipedia.org/wiki/Map_(mathematics)) вхідні дані та виходи.

Алгоритми навчання під наглядом створюють математичну модель набору даних, яка містить як вхідні, так і бажані вихідні дані. Дані відомі як [навчальні дані](https://en.wikipedia.org/wiki/Training_data) та складаються з набору навчальних прикладів. Кожен навчальний приклад має один або кілька входів і бажаний вихід, також відомий як контрольний сигнал. У математичній моделі кожен приклад навчання представлено масивом [або](https://en.wikipedia.org/wiki/Array_data_structure) вектором, який іноді називають [вектором ознак](https://en.wikipedia.org/wiki/Feature_vector), а дані навчання представлені [матрицею](https://en.wikipedia.org/wiki/Matrix_(mathematics)). Завдяки [ітераційній оптимізації](https://en.wikipedia.org/wiki/Mathematical_optimization#Computational_optimization_techniques) цільової [функції](https://en.wikipedia.org/wiki/Loss_function) алгоритми керованого навчання вивчають функцію, яку можна використовувати для прогнозування результату, пов’язаного з новими входами. Оптимальна функція дозволить алгоритму правильно визначати вихід для вхідних даних, які не були частиною навчальних даних. Кажуть, що алгоритм, який з часом покращує точність своїх виходів або прогнозів, навчився виконувати це завдання.

Типи алгоритмів навчання під наглядом включають:

* [активне навчання](https://en.wikipedia.org/wiki/Active_learning_(machine_learning)) – алгоритм навчання може в інтерактивному режимі запитувати користувача (або інше джерело інформації), щоб позначити нові точки даних бажаними виходами.
* [класифікацію](https://en.wikipedia.org/wiki/Statistical_classification) – проблема визначення, до якої з набору категорій [(](https://en.wikipedia.org/wiki/Categorical_data) субсукупностей) належить віднести [спостереження (однина) або спостереження (множина).](https://en.wikipedia.org/wiki/Observation)Прикладами є віднесення певної електронної пошти до класу [«спам» або «не спам»](https://en.wikipedia.org/wiki/Spam_filtering).
* [регресію](https://en.wikipedia.org/wiki/Regression_analysis) – набір статистичних процесів для [оцінки](https://en.wikipedia.org/wiki/Estimation_theory) зв’язків між [залежною змінною](https://en.wikipedia.org/wiki/Dependent_variable) (часто її називають змінною «результат» або «відповідь» або «міткою» на мові машинного навчання) та однією чи кількома [незалежними змінними](https://en.wikipedia.org/wiki/Independent_variable) ( часто називають «предикторами», «коваріатами», «пояснювальними змінними» або «особливими ознаками»). Найпоширенішою формою регресійного аналізу є [лінійна регресія](https://en.wikipedia.org/wiki/Linear_regression), у якій знаходить лінію (або більш складну [лінійну комбінацію](https://en.wikipedia.org/wiki/Linear_combination)), яка найбільше відповідає даним згідно з певним математичним критерієм.

1. [Неконтрольоване навчання](https://en.wikipedia.org/wiki/Unsupervised_learning): алгоритму навчання не призначаються мітки, тому він сам знаходить структуру вхідних даних. Навчання без нагляду може бути самоціллю (виявлення прихованих шаблонів у даних) або засобом досягнення мети ( [навчання функцій](https://en.wikipedia.org/wiki/Feature_learning) ).

Алгоритми неконтрольованого навчання беруть набір даних, який містить лише вхідні дані, і знаходять у даних структуру, як-от групування чи кластеризацію точок даних. Таким чином, алгоритми вивчають тестові дані, які не були позначені чи класифіковані. Замість того, щоб реагувати на відгуки, алгоритми неконтрольованого навчання виявляють спільні риси в даних і реагують на наявність або відсутність таких спільних рис у кожній новій частині даних. Основним застосуванням неконтрольованого навчання є оцінка [щільності](https://en.wikipedia.org/wiki/Density_estimation) в [статистиці](https://en.wikipedia.org/wiki/Statistics), наприклад, визначення [функції щільності ймовірності](https://en.wikipedia.org/wiki/Probability_density_function).

Кластерний аналіз — це розподіл набору спостережень на підмножини (так звані кластери ), щоб спостереження в одному кластері були подібними відповідно до одного або кількох заздалегідь визначених критеріїв, тоді як спостереження, взяті з різних кластерів, були несхожими.

1. [Навчання з підкріпленням](https://en.wikipedia.org/wiki/Reinforcement_learning): комп’ютерна програма взаємодіє з динамічним середовищем, у якому вона повинна виконати певну мету (наприклад, [керувати транспортним засобом](https://en.wikipedia.org/wiki/Autonomous_car) або грати проти суперника). Переміщаючись у просторі проблем, програмі надається зворотний зв’язок, аналогічний винагородам, які вона намагається максимізувати.

Навчання з підкріпленням відрізняється від навчання під наглядом тим, що не потрібно подавати позначені пари входу/виходу, а також не потрібно явно виправляти субоптимальні дії. Натомість увага зосереджена на пошуку балансу між дослідженням і експлуатацією.

Середовище зазвичай описується у формі [Марковського процесу прийняття рішень](https://en.wikipedia.org/wiki/Markov_decision_process) (MDP), оскільки багато алгоритмів навчання з підкріпленням для цього контексту використовують методи [динамічного програмування.](https://en.wikipedia.org/wiki/Dynamic_programming) Основна відмінність між класичними методами динамічного програмування та алгоритмами навчання з підкріпленням полягає в тому, що останні не передбачають знання точної математичної моделі MDP, і вони націлені на великі MDP, де точні методи стають неможливими.

У математиці Марковський процес прийняття рішень ( MDP ) – це [стохастичний процес](https://en.wikipedia.org/wiki/Stochastic)[управління](https://en.wikipedia.org/wiki/Optimal_control_theory)[з дискретним часом](https://en.wikipedia.org/wiki/Discrete-time). Він забезпечує математичну основу для моделювання [прийняття рішень](https://en.wikipedia.org/wiki/Decision_making) у ситуаціях, коли результати є частково [випадковими](https://en.wikipedia.org/wiki/Randomness#In_mathematics), а частково знаходяться під контролем особи, яка приймає рішення.

Динамічне програмування – спрощення складної проблеми шляхом її розбиття на простіші підпроблеми [рекурсивним](https://en.wikipedia.org/wiki/Recursion) способом. Хоча деякі проблеми прийняття рішень неможливо розібрати таким чином, рішення, які охоплюють кілька моментів часу, часто розбиваються рекурсивно.

Хоча кожен алгоритм має переваги та обмеження, жоден з них не підійде для вирішення всіх проблем.

Виконання машинного навчання може включати створення [моделі](https://en.wikipedia.org/wiki/Statistical_model) , яка навчається на деяких навчальних даних, а потім може обробляти додаткові дані для прогнозування.

Для підвищення ефективності вище описаних методів, також використовують:

* Штучні нейронні мережі ( ШНМ (з англ. ANM), також скорочено до [нейронних мереж](https://en.wikipedia.org/wiki/Neural_network) (НМ)) – базується на сукупності з’єднаних одиниць або вузлів, які називаються [штучними нейронами](https://en.wikipedia.org/wiki/Artificial_neuron), які приблизно моделюють [нейрони](https://en.wikipedia.org/wiki/Neuron) біологічного мозку. Кожне з’єднання, як [синапси](https://en.wikipedia.org/wiki/Synapse) в біологічному мозку, може передавати сигнал до інших нейронів. Штучний нейрон отримує сигнали, потім обробляє їх і може сигналізувати підключеним до нього нейронам. «Сигнал» у з’єднанні є [дійсним числом](https://en.wikipedia.org/wiki/Real_number) , а вихід кожного нейрона обчислюється за допомогою деякої нелінійної функції суми його входів. З'єднання називаються ребрами . Нейрони та ребра зазвичай мають [вагу](https://en.wikipedia.org/wiki/Weighting" \o "Зважування)який коригується в міру навчання. Вага збільшує або зменшує силу сигналу під час підключення.
* Дерево рішень – це підхід до контрольованого навчання, який використовується в [статистиці](https://en.wikipedia.org/wiki/Statistics), [аналізі даних](https://en.wikipedia.org/wiki/Data_mining) і [машинному навчанні](https://en.wikipedia.org/wiki/Machine_learning). У цьому формалізмі класифікаційне або регресійне [дерево рішень](https://en.wikipedia.org/wiki/Decision_tree) використовується як [прогностична модель](https://en.wikipedia.org/wiki/Predictive_model), щоб зробити висновки щодо набору спостережень.
* Машини опорних векторів (SVM) або мережі опорних векторів, являють собою набір пов’язаних [контрольованих](https://en.wikipedia.org/wiki/Supervised_learning) методів навчання, які використовуються для класифікації та регресії. На основі набору навчальних прикладів, кожен з яких позначено як належний до однієї з двох категорій, навчальний алгоритм SVM будує модель, яка передбачає, чи належить новий приклад до однієї категорії. Алгоритм навчання SVM є неімовірнісним[,](https://en.wikipedia.org/wiki/Probabilistic_classification) двійковим[,](https://en.wikipedia.org/wiki/Binary_classifier) лінійним [класифікатором](https://en.wikipedia.org/wiki/Linear_classifier).

Вбудоване машинне навчання — це підгалузь машинного навчання, де модель машинного навчання виконується на [вбудованих системах](https://en.wikipedia.org/wiki/Embedded_systems) з обмеженими обчислювальними ресурсами, як-от [переносні комп’ютери](https://en.wikipedia.org/wiki/Wearable_computer), [периферійні пристрої](https://en.wikipedia.org/wiki/Edge_device) та [мікроконтролери](https://en.wikipedia.org/wiki/Microcontrollers). Запуск моделі машинного навчання у вбудованих пристроях усуває потребу в передачі та зберіганні даних на хмарних серверах для подальшої обробки, відтепер, зменшуючи порушення даних і витоки конфіденційності через передачу даних, а також мінімізує крадіжки інтелектуальна власність, персональні дані та комерційна таємниця. Вбудоване машинне навчання можна застосувати за допомогою кількох методів, включаючи [апаратне прискорення](https://en.wikipedia.org/wiki/Hardware_acceleration), з використанням [наближених обчислень](https://en.wikipedia.org/wiki/Approximate_computing), оптимізація моделей машинного навчання та багато іншого.

Апаратне прискорення — це використання [апаратного забезпечення комп’ютера](https://en.wikipedia.org/wiki/Computer_hardware), призначеного для більш ефективного виконання певних функцій порівняно з [програмним забезпеченням, що працює на](https://en.wikipedia.org/wiki/Software)[центральному процесорі](https://en.wikipedia.org/wiki/Central_processing_unit) загального призначення (CPU).

Наближене обчислення є новою парадигмою для енергоефективного та/або високопродуктивного дизайну. Він включає в себе безліч методів обчислення, які повертають можливо неточний результат, а не гарантовано точний результат, які можуть бути використані для програм, де приблизний результат є достатнім для своєї мети.

Машинне навчання використовується в

* пошукових системах Інтернету
* фільтрах електронної пошти для сортування спаму
* веб-сайтах для персоналізованих рекомендацій
* банківському програмному забезпеченні для виявлення незвичайних транзакцій
* багатьох програмах на телефонах, таких як розпізнавання голосу.

ГЛИБОКЕ НАВЧАННЯ

Глибоке навчання (DL) – це підмножина  [машинного навчання](https://www.ibm.com/topics/machine-learning) , яка, по суті, є нейронною мережею з трьома або більше рівнями. Ці нейронні мережі намагаються змоделювати поведінку людського мозку, дозволяючи йому «навчатися» на великих обсягах даних. Хоча нейронна мережа з одним шаром все ще може робити приблизні прогнози, додаткові приховані шари можуть допомогти оптимізувати та покращити точність.

DL керує багатьма  додатками та службами [штучного інтелекту (AI)](https://www.ibm.com/topics/artificial-intelligence) , які покращують автоматизацію, виконуючи аналітичні та фізичні завдання без втручання людини. Технологія глибокого навчання лежить в основі повсякденних продуктів і послуг (таких як цифрові помічники, пульти дистанційного керування телевізором із голосовою підтримкою та виявлення шахрайства з кредитними картками), а також нових технологій (таких як безпілотні автомобілі).

Підходи до навчання за допомогою DL ідентичні до тих, що використовуються у ML, а саме: контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням, - тому, що DL є підмножиною ML.

Якщо модель для машинного навчання потребує ретельної попередньої підготовки, то при створенні моделі для глибокого навчання можна пропустити тривалу підготовку, тобто, DL  усуває частину попередньої обробки даних, яка зазвичай пов’язана з машинним навчанням. Такі алгоритми можуть приймати й обробляти неструктуровані дані, як-от текст і зображення, і автоматизують вилучення функцій, усуваючи певну залежність від експертів-людей. Наприклад, є набір фотографій різних домашніх тварин, і потрібно класифікувати їх за категоріями «кіт», «собака», «хом’як» тощо. Алгоритми глибокого навчання можуть визначити, які особливості (наприклад, вуха) є найважливішими, щоб відрізнити кожну тварину від іншої. У машинному навчанні ця ієрархія функцій встановлюється вручну людиною-експертом.

Глибока нейронна мережа (DNN) — це [штучна нейронна мережа](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANM) із кількома рівнями між вхідним і вихідним рівнями. Існують різні типи нейронних мереж, але вони завжди складаються з тих самих компонентів: нейронів, синапсів, вагових коефіцієнтів, зміщень і функцій. Ці компоненти в цілому функціонують подібно до людського мозку, і їх можна навчити, як і будь-який інший алгоритм ML.

Нейронні мережі глибокого навчання, або штучні нейронні мережі, намагаються імітувати людський мозок за допомогою комбінації вхідних даних, ваг і зміщення. Ці елементи працюють разом, щоб точно розпізнавати, класифікувати та описувати об’єкти в даних.

Глибокі нейронні мережі складаються з кількох шарів взаємопов’язаних вузлів, кожен з яких будується на попередньому шарі для уточнення й оптимізації прогнозу або категоризації. Цей хід обчислень через мережу називається прямим поширенням. Вхідний і вихідний шари глибокої нейронної мережі називаються  видимими  шарами, а шари між ними – прихованими. Вхідний рівень – це місце, де модель глибокого навчання отримує дані для обробки, а вихідний рівень – це місце, де робиться остаточний прогноз або класифікація.

Інший процес, який називається зворотним поширенням,  використовує алгоритми, такі як градієнтний спуск, для обчислення помилок у передбаченнях, а потім коригує ваги та зміщення функції, переміщаючись назад через шари, щоб навчити модель. Разом пряме та зворотне поширення дозволяють нейронній мережі робити прогнози та відповідно виправляти будь-які помилки. З часом алгоритм стає точнішим.

Основні кроки в мережі зворотного поширення:

1. Проходження через мережу від входу до виходу, обчислюючи вихід прихованих шарів і вихідний рівень.
2. Обчислення похідної функції вартості відносно вхідного та прихованого шарів у вихідному шарі.
3. Неодноразове оновлення вагових коефіцієнтів, поки вони не збігаються або модель не пройшла достатньо ітерацій.

Пряме та зворотне поширення є найпростішими в реалізації та розумінні архітектурами для глибокого навчання нейронних мереж. Однак алгоритми глибокого навчання неймовірно складні, і існують різні типи нейронних мереж для вирішення конкретних проблем або наборів даних. Наприклад:

* Згорткові нейронні мережі (CNN) – клас глибинних [штучних нейронних мереж](https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0) [прямого поширення](https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D0%B3%D0%BE_%D0%BF%D0%BE%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8F), який успішно застосовується для аналізу візуальних зображень. CNN використовують різновид [багатошарових перцептронів](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D1%88%D0%B0%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD), розроблений так, щоби вимагати використання мінімального обсягу [попередньої обробки](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80).

Багатошаровий перцептрон Румельхарта — окремий випадок [перцептрона Розенблатта](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD), в якому один алгоритм [зворотного поширення помилки](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B7%D0%B2%D0%BE%D1%80%D0%BE%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BF%D0%BE%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_%D0%BF%D0%BE%D0%BC%D0%B8%D0%BB%D0%BA%D0%B8) навчає всі шари.

* Рекурсивні нейронні мережі (RNN) — це клас [глибинних нейронних мереж](https://uk.wikipedia.org/wiki/%D0%93%D0%BB%D0%B8%D0%B1%D0%B8%D0%BD%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), створюваних [рекурсивним](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D1%96%D1%8F) застосуванням одного й того ж набору ваг до структури, щоби здійснювати [структурове передбачування](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BE%D0%B2%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B1%D0%B0%D1%87%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) вхідних структур мінливого розміру, або скалярне передбачування на них, шляхом обходу заданої структури в [топологічній послідовності](https://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BF%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B5_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). RNN були успішними, наприклад, в навчанні послідовнісних та деревних структур в [обробці природної мови](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0_%D0%BF%D1%80%D0%B8%D1%80%D0%BE%D0%B4%D0%BD%D0%BE%D1%97_%D0%BC%D0%BE%D0%B2%D0%B8), головним чином неперервних представлень фраз та речень на основі [векторного представлення слів](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B5_%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D1%81%D0%BB%D1%96%D0%B2).

Структурове передбачування (structured prediction) — це [узагальнювальний термін](https://uk.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D1%96%D0%BD) для методик [керованого](https://uk.wikipedia.org/wiki/%D0%9A%D0%B5%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F) машинного навчання, які включають [передбачування](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B1%D0%B0%D1%87%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) структурованих об'єктів, а не скалярних [дискретних](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F) або [дійснозначних](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%96%D0%B9%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7) значень.

Глибоке навчання застосовують в:

* Правоохоронних органах – розпізнаванні мовлення, розпізнавання зображень;
* Фінансовій сфері – прогнозна аналітика, оцінка бізнес-ризиків, виявлення шахрайства;
* Медицині – пошук ліків, симуляція впливу ліків та вірусів, передбачення потенційно небезпечних патогенів;
* Обслуговуванні – віртуальні помічники, чат боти.
* Біоінформатика – анотація генів.

КОМП’ЮТЕРНИЙ ЗІР

Одним із найпотужніших і найпривабливіших типів ШІ є комп’ютерний зір (CV) – сфера штучного інтелекту (ШІ), яка дозволяє комп’ютерам і системам отримувати значущу інформацію з цифрових зображень, відео та інших візуальних вхідних даних — і виконувати дії або давати рекомендації на основі цієї інформації. Якщо ШІ дозволяє комп’ютерам мислити, комп’ютерний зір дозволяє їм бачити, спостерігати та розуміти.

Комп’ютерний зір використовується для виконання наступних завдань:

* Класифікація зображень бачить зображення і може його класифікувати (собака, яблуко, обличчя людини). Точніше, він здатний точно передбачити, що дане зображення належить до певного класу. Наприклад, компанія соціальних медіа може використовувати його для автоматичної ідентифікації та відокремлення небажаних зображень, завантажених користувачами.
* Виявлення об’єктів може використовувати класифікацію зображень для ідентифікації певного класу зображень, а потім виявлення та таблиці їх появи на зображенні чи відео. Приклади включають виявлення пошкоджень на складальній лінії або ідентифікацію обладнання, яке потребує технічного обслуговування.
* Відстеження об’єкта стежить за об’єктом після його виявлення. Це завдання часто виконується за допомогою послідовних зображень або відео в реальному часі. Автономним транспортним засобам, наприклад, потрібно не тільки класифікувати та виявляти такі об’єкти, як пішоходи, інші автомобілі та дорожню інфраструктуру, їм потрібно відстежувати їх рух, щоб уникнути зіткнень і дотримуватися правил дорожнього руху.
* Пошук зображень на основі вмісту використовує комп’ютерний зір для перегляду, пошуку та отримання зображень із великих сховищ даних на основі вмісту зображень, а не пов’язаних із ними тегів метаданих. Це завдання може включати автоматичне анотування зображення, яке замінює ручне тегування зображення. Ці завдання можна використовувати для систем [управління цифровими активами](https://www.ibm.com/topics/digital-asset-management) та можуть підвищити точність пошуку та вилучення.
* Ідентифікація  – розпізнається окремий екземпляр об’єкта. Приклади включають ідентифікацію обличчя чи відбитків пальців конкретної людини, [ідентифікацію рукописних цифр](https://en.wikipedia.org/wiki/Handwriting_recognition) або конкретного автомобіля.
* [Оцінка пози](https://en.wikipedia.org/wiki/Pose_(computer_vision))  – оцінка положення або орієнтації конкретного об’єкта відносно камери. Прикладом застосування цієї техніки може бути допомога руці робота у вилученні об’єктів із конвеєрної стрічки на[складальній лінії](https://en.wikipedia.org/wiki/Assembly_line).
* [Оптичне розпізнавання символів](https://en.wikipedia.org/wiki/Optical_character_recognition) (OCR) – ідентифікація[символів](https://en.wikipedia.org/wiki/Character_(computing)) у зображеннях друкованого чи рукописного тексту, як правило, з метою кодування тексту у форматі, зручнішому для редагування чи[індексування](https://en.wikipedia.org/wiki/Search_index) ( наприклад, [ASCII](https://en.wikipedia.org/wiki/ASCII) ). Пов’язаним завданням є зчитування двовимірних кодів, таких як[матриця даних](https://en.wikipedia.org/wiki/Data_Matrix) і[QR-](https://en.wikipedia.org/wiki/QR_code)коди.
* [Розпізнавання облич](https://en.wikipedia.org/wiki/Facial_recognition_system)  – технологія, яка дозволяє зіставляти обличчя на цифрових зображеннях або відеокадрах з базою даних облич, яка зараз широко використовується для блокування обличчя мобільного телефону, розумного замка дверей тощо.
* Розпізнавання шаблонів (PR) – це автоматизоване розпізнавання [шаблонів](https://en.wikipedia.org/wiki/Pattern) і закономірностей у [даних](https://en.wikipedia.org/wiki/Data). PR має застосування в [аналізі статистичних даних](https://en.wikipedia.org/wiki/Data_analysis), [обробці сигналів](https://en.wikipedia.org/wiki/Signal_processing), [аналізі зображень](https://en.wikipedia.org/wiki/Image_analysis), [пошуку інформації](https://en.wikipedia.org/wiki/Information_retrieval), [біоінформатиці](https://en.wikipedia.org/wiki/Bioinformatics), [стисненні даних](https://en.wikipedia.org/wiki/Data_compression), [комп'ютерній графіці](https://en.wikipedia.org/wiki/Computer_graphics) та [машинному навчанні](https://en.wikipedia.org/wiki/Machine_learning). Наприклад, PR може відрізнити людину від іншого об’єкту.
* [Розпізнавання активності людини](https://en.wikipedia.org/wiki/Activity_recognition) – стосується розпізнавання діяльності із серії відеокадрів, наприклад, якщо людина бере предмет або йде.

Організація системи комп’ютерного зору сильно залежить від прикладних програм. Деякі системи є автономними програмами, які вирішують конкретну проблему вимірювання або виявлення, тоді як інші становлять підсистему більшої конструкції, яка, наприклад, також містить підсистеми для керування механічними приводами, плануванням, інформаційними базами даних, людино-машинними інтерфейсами тощо. Конкретна реалізація системи комп’ютерного зору також залежить від того, чи є її функціональні можливості попередньо визначеними, чи деякі її частини можна вивчати чи змінювати під час роботи. Багато функцій є унікальними для програми. Проте є типові функції, які є в багатьох системах комп’ютерного зору. Наприклад:

1. Отримання зображення – цифрове зображення створюється одним або кількома [датчиками зображення](https://en.wikipedia.org/wiki/Image_sensor) , які, окрім різних типів світлочутливих камер, включають [датчики дальності](https://en.wikipedia.org/wiki/Rangefinder_camera) , томографічні пристрої, радар, ультразвукові камери тощо. Залежно від типу датчика, отримані дані зображення є звичайним 2D-зображенням, 3D-об’ємом або послідовністю зображень.
2. Попередня обробка – перш ніж метод комп’ютерного зору можна буде застосувати до даних зображення з метою вилучення певної інформації, зазвичай необхідно обробити дані, щоб переконатися, що вони задовольняють певні припущення, які передбачає метод.
3. [Вилучення функцій](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)) – із даних зображення витягуються функції зображення різного рівня складності. Типовими прикладами таких функцій є:

* Лінії, [ребра](https://en.wikipedia.org/wiki/Edge_detection) та [виступи](https://en.wikipedia.org/wiki/Ridge_detection) .
* Локалізовані [точки інтересу,](https://en.wikipedia.org/wiki/Interest_point_detection) такі як [кути](https://en.wikipedia.org/wiki/Corner_detection) , [краплі](https://en.wikipedia.org/wiki/Blob_detection) або точки.

Більш складні функції можуть бути пов’язані з текстурою, формою або рухом.

1. [Виявлення](https://en.wikipedia.org/wiki/Object_detection) / [сегментація](https://en.wikipedia.org/wiki/Image_segmentation) – на певному етапі обробки приймається рішення про те, які точки або області зображення є релевантними для подальшої обробки. Приклади:

* Вибір певного набору точок інтересу.
* Сегментація однієї чи кількох областей зображення, які містять певний об’єкт інтересу.

1. Високорівнева обробка – на цьому кроці зазвичай вводиться невеликий набір даних, наприклад набір точок або область зображення, яка, як передбачається, містить певний об’єкт. Решта обробки це:

* Перевірка того, що дані задовольняють припущенням на основі моделі та конкретної програми.
* Оцінка специфічних для програми параметрів, таких як положення або розмір об’єкта.
* Класифікація зображень.
* [Реєстрація зображення](https://en.wikipedia.org/wiki/Image_registration) – порівняння та поєднання двох різних видів одного і того ж об’єкта.

1. Прийняття рішень. Прийняття остаточного рішення, необхідного для заявки, наприклад:

* Пройшов/не пройшов перевірку програм автоматичної перевірки.
* Збіг/не збіг у програмах розпізнавання.
* Позначити для подальшої перевірки людиною.

Бібліотеки для роботи з комп’ютерним зором

Бібліотека комп’ютерного зору – це набір попередньо написаного коду та даних, які використовуються для створення або оптимізації комп’ютерної програми. Бібліотеки комп’ютерного зору численні та адаптовані до конкретних потреб або мов програмування.

Бібліотеки для роботи з CV є для різних мов програмування, наприклад:

* PHP-OpenCV для PHP.
* ruby-OpenCV для Ruby.
* common-cv для CommonLisp.
* gocv для Go.
* cv-rs для Rust.
* EmguCV для C#.

Тобто, працювати з комп’ютерним зором можна на більшості популярних, і не дуже, мов програмування.

Наявність таких великих і зручних бібліотек як NumPy та PyTorch, який також можна використовувати для роботи з CV і він є основою для деяких бібліотек зі списку нижче, для Python і його популярність у сфері ШІ вплинуло і на те, що більшість популярних, на цей момент, бібліотек для роботи з CV написані для цієї мови програмування. Наприклад:

* [Scikit-Image](https://scikit-image.org/) вважається найзручнішою та природною бібліотекою Python, яка є «розширенням» Scikit-Learn. Це один із найбільш часто використовуваних інструментів для контрольованого та неконтрольованого машинного навчання. Scikit-Learn — це пакет Python, який використовується для обробки зображень і роботи з масивами NumPy як об’єктами зображень. Оскільки це Naturally Python і використовує модуль Scikit-cuda, Scikit-Image є безкоштовним і має обмеження.
* [Pillow](https://python-pillow.org/) – бібліотека з відкритим кодом для мови програмування Python. Його можна використовувати в Windows, Mac OS X і Linux. Бібліотека зображень Python надає інтерпретатору Python можливості обробки зображень, а її бібліотеку зображень модифіковано для швидкого доступу до даних. Його можна використовувати як на мовах C, так і на Python, і він має Python Wrapper. Здебільшого використовується для читання та збереження зображень різних форматів, Pillow також містить різні базові перетворення зображень, такі як обертання, об’єднання, масштабування тощо.
* Як розширення бібліотеки PyTorch, [TorchVision](https://pytorch.org/vision/stable/index.html) містить найпоширеніші трансформації зображень для комп’ютерного зору. Він також містить набори даних і архітектури моделей для нейронних мереж комп’ютерного зору. Однією з головних цілей TorchVision є забезпечення природного способу використання трансформацій зображень комп’ютерного зору за допомогою моделей PyTorch без їх перетворення в масив NumPy і назад. Його пакет містить загальні набори даних, архітектури моделей і регулярні перетворення зображень комп’ютерного зору. TorchVision — це природно Python, і його можна використовувати для мов Python і C++.
* [MMCV](https://mmcv.readthedocs.io/en/latest/) — це тип розширення PyTorch, який забезпечує обробку й перетворення зображень/відео, візуалізацію зображень і анотацій, а також багато архітектур CNN. Він підтримує такі системи, як Linux, Windows і macOS, і є одним із найкорисніших наборів інструментів для дослідників комп’ютерного зору. Він використовується для Python, C++ і CUDA і має Python Wrapper.
* [Keras](https://keras.io/)  — це бібліотека програмного забезпечення з відкритим вихідним кодом на основі Python, яка особливо корисна для початківців, оскільки дозволяє швидко створювати моделі нейронних мереж і забезпечує підтримку серверної частини. Це набір модульних будівельних блоків, які інженери комп’ютерного зору можуть використовувати для швидкого збирання виробничого рівня, найсучаснішого навчання та конвеєрів висновків. Маючи понад 400 000 окремих користувачів, Keras має потужну підтримку спільноти. Використовує TensorFlow.
* [MATLAB](https://mathworks.com/products/matlab.html) — це скорочення від Matrix Laboratory, і це платна платформа програмування, яка підходить для різних програм, таких як машинне навчання, глибоке навчання, обробка зображень, відео та сигналів. Постачається з  [набором інструментів](https://mathworks.com/products/computer-vision.html), який містить кілька функцій, програм і алгоритмів, які допомагають виконувати завдання, пов’язані з комп’ютерним зором.

OpenCV

Не зважаючи на те, що існує велика кількість бібліотек для роботи з CV, найстарішою і, при цьому, найпопулярнішою є OpenCV – бібліотека комп’ютерного зору та машинного навчання з відкритим кодом. OpenCV було створено, щоб забезпечити загальну інфраструктуру для програм комп’ютерного зору та прискорити використання машинного сприйняття в комерційних продуктах. Будучи ліцензованим продуктом Apache 2, OpenCV дозволяє компаніям легко використовувати та змінювати код.

Це кросплатформна бібліотека, яка підтримує Windows, Linux, Android і macOS і може використовуватися різними мовами, такими як Python, Java, C++ тощо. OpenCV має Python Wrapper і використовує модель CUDA для GPU. Спочатку розроблений компанією Intel, тепер його можна безкоштовно використовувати за ліцензією BSD з відкритим кодом. Він також містить деякі моделі, які можна перетворити на моделі TensorFlow.

Бібліотека OpenCV включає в себе:

* Набори інструментів для 2D і 3D функцій.
* Додаток для розпізнавання обличчя.
* Розпізнавання жестів.
* Розуміння руху.
* Виявлення об’єктів.
* Сегментація та розпізнавання.

Перелічені вище можливості є лише прикладом функціональності цієї бібліотеки.

OpenCV використовується:

1. Приватними компаніями: Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota та багато інших.
2. У стартапах: Applied Minds, VideoSurf і Zeitera та багатьох інших.
3. У багатьох країнах:

* охоплює діапазон від з’єднання зображень вуличного перегляду до виявлення вторгнень у відеоспостереження в Ізраїлі.
* Допомога роботам орієнтуватися та підбирати предмети в лабораторії Willow Garage (США).
* Виявлення випадків утоплення в басейні в Європі.
* Інтерактивне мистецтво в Іспанії та США.
* Перевірка етикеток на продуктах на заводах по всьому світу.
* Швидке виявлення обличчя в Японії.

ВИСНОВКИ

Концепція ШІ загалом є великою темою, яка стосується не лише сфери IT і це зумовлено тим, що створення чогось, що не є живим фізично, але володіє інтелектом спонукає спеціалістів різних сфер працювати над цим питанням.

Зараз існує багато типів, методів навчання та призначень ШІ як в повсякденному житті, так і в вузьких сферах.

Штучні нейронні мережі які навчають за допомогою різних методів машинного навчання і є тим ШІ, який можна побачити в багатьох сферах сучасного світу.

Від вузького ШІ призначеного для конкретних цілей до гіпотетичного AGI навченого за допомогою DL, можливості якого будуть, як мінімум, рівними людським в плані можливостей та мислення або супер ШІ, який є фінальним етапом у вивчені ШІ.

Зважаючи на розрахункові можливості сучасних комп’ютерів та загальний розвиток технологій найкращим, що сьогодні може представити людство є вузький ШІ та комп’ютерний зір, яким можна його наділити. Тобто, зараз ми можемо навчити машину бачити та виконувати різноманітні дії в залежності від її призначення.

У сучасному світі робота з ШІ є одним з пріоритетних завдань, яке несе за собою велику кількість проблем, від підходів до навчання до питань: «Чи може ШІ володіти особистістю?».

Через охоплення великої кількості наукових сфер та актуальності, робота з ШІ вже є хорошим способом для заробітку і можливістю залити слід в історії людства для спеціалістів різних сфер та рівнів.

СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ

1. Штучний інтелект. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Artificial_intelligence> (дата звернення 16.07.2023)
2. Інтелектуальний агент. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Intelligent_agent> (дата звернення 16.07.2023)
3. Загальний штучний інтелект. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Artificial_general_intelligence> (дата звернення 16.07.2023)
4. Представлення знань. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning> (дата звернення 16.07.2023)
5. Планування штучного інтелекту. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Automated_planning_and_scheduling> (дата звернення 16.07.2023)
6. Обробка природньої мови. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Natural_language_processing> (дата звернення 16.07.2023)
7. Типи штучного інтелекту. Веб-сайт. URL : <https://www.forbes.com/sites/cognitiveworld/2019/06/19/7-types-of-artificial-intelligence/?sh=4c0f1170233e> (дата звернення 19.07.2023)
8. Типи штучного інтелекту. Веб-сайт. URL : <https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/types-of-artificial-intelligence> (дата звернення 19.07.2023)
9. Де ШІ використовується зараз? Веб-сайт. URL : <https://scienceexchange.caltech.edu/topics/artificial-intelligence-research/artificial-intelligence-everyday-life-uses> (дата звернення 17.07.2023)
10. Що таке машинне навчання і де його використовують? Веб-сайт. URL : <https://royalsociety.org/topics-policy/projects/machine-learning/videos-and-background-information/> (дата звернення 17.07.2023)
11. Машинне навчання. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Machine_learning> (дата звернення 17.07.2023)
12. Машинне навчання: Контрольоване навчання. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Supervised_learning> (дата звернення 17.07.2023)
13. Контрольоване навчання: Активне навчання. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Active_learning_(machine_learning)> (дата звернення 17.07.2023)
14. Контрольоване навчання: Класифікація. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Statistical_classification> (дата звернення 17.07.2023)
15. Контрольоване навчання: Регресійний аналіз. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Regression_analysis> (дата звернення 17.07.2023)
16. Машинне навчання: Неконтрольоване навчання. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Unsupervised_learning> (дата звернення 17.07.2023)
17. Кластерний аналіз. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Cluster_analysis> (дата звернення 17.07.2023)
18. Машинне навчання: Навчання з підкріпленням. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Reinforcement_learning> (дата звернення 17.07.2023)
19. Марковський процес прийняття рішень. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Markov_decision_process> (дата звернення 17.07.2023)
20. Динамічне програмування. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Dynamic_programming> (дата звернення 17.07.2023)
21. Штучна нейронна мережа. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Artificial_neural_network> (дата звернення 18.07.2023)
22. Дерево рішень. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Decision_tree_learning> (дата звернення 18.07.2023)
23. Опорна векторна машина. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Support_vector_machine> (дата звернення 18.07.2023)
24. Апаратне прискорення. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Hardware_acceleration> (дата звернення 18.07.2023)
25. Наближене обчислення. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Approximate_computing> (дата звернення 18.07.2023)
26. Що таке глибоке навчання і де його застосовують? Веб-сайт. URL : <https://www.ibm.com/topics/deep-learning> (дата звернення 18.07.2023)
27. Глибоке навчання. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Deep_learning> (дата звернення 18.07.2023)
28. Зворотне поширення. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Backpropagation> (дата звернення 18.07.2023)
29. Згорткова нейронна мережа. Веб-сайт. URL : <https://uk.wikipedia.org/wiki/%D0%97%D0%B3%D0%BE%D1%80%D1%82%D0%BA%D0%BE%D0%B2%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0> (дата звернення 18.07.2023)
30. Багатошаровий перцептрон Румельхарта. Веб-сайт. URL : <https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D1%88%D0%B0%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D1%86%D0%B5%D0%BF%D1%82%D1%80%D0%BE%D0%BD> (дата звернення 18.07.2023)
31. Рекурсивна нейронна мережа. Веб-сайт. URL : <https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D0%B8%D0%B2%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0> (дата звернення 18.07.2023)
32. Структурове передбачування. Веб-сайт. URL : <https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BE%D0%B2%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B1%D0%B0%D1%87%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F> (дата звернення 18.07.2023)
33. Комп’ютерний зір. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Computer_vision> (дата звернення 20.07.2023)
34. Розпізнавання шаблонів. Веб-сайт. URL : <https://en.wikipedia.org/wiki/Pattern_recognition> (дата звернення 20.07.2023)
35. Що таке комп’ютерний зір? Веб-сайт. URL : <https://www.ibm.com/topics/computer-vision> (дата звернення 20.07.2023)
36. Усе, що ви хотіли знати про комп’ютерний зір. Веб-сайт. URL : <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e> (дата звернення 20.07.2023)
37. Найкращі бібліотеки для комп’ютерного зору. Веб-сайт. URL : <https://www.superannotate.com/blog/computer-vision-libraries> (дата звернення 20.07.2023)
38. Про - OpenCV. Веб-сайт. URL : <https://opencv.org/about/> (дата звернення 20.07.2023)
39. Що таке OpenCV і чому він такий популярний?. Веб-сайт. URL :<https://medium.com/analytics-vidhya/what-and-why-opencv-3b807ade73a0> (дата звернення 20.07.2023)