Навчально-методична література

Міністерство освіти і науки України
Тернопільський національний технічний університет
ім. Івана Пулюя
Кафедра комп’ютерно-інтегрованих технологій

КОНСПЕКТ ЛЕКЦІЙ

з дисципліни

«КОМП’ЮТЕРНІ СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ»

для студентів спеціальності G7 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» денної та заочної форми здобуття освіти

Тернопіль – 2025

Конспект лекцій з дисципліни «Комп’ютерні системи штучного інтелекту» для студентів спеціальності G7 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» денної та заочної форми здобуття освіти / уклад. І. С. Дідич. // ТНТУ. – 2025. – С. 92.

Укладач: доктор філософії, доц., Ірина ДІДИЧ
Рецензент: д.т.н., проф., Олег ЯСНІЙ

Відповідальний за випуск: доктор філософії, доц., Ірина ДІДИЧ

Конспект лекцій з дисципліни «Комп’ютерні системи штучного інтелекту» для студентів спеціальності G7 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» денної та заочної форми здобуття освіти розглянуто і схвалено на засіданні кафедри комп’ютерно-інтегрованих технологій
Протокол № 1 від “ 28 ” серпня 2025 року.

Конспект лекцій з дисципліни з дисципліни «Комп’ютерні системи штучного інтелекту» для студентів спеціальності G7 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» денної та заочної форми здобуття освіти схвалено та рекомендовано до друку науково-методичною комісією факультету прикладних інформаційних технологій та електроінженерії
Протокол № 1 від « 29 » серпня 2025 року

ЗМІСТ

ТЕМА 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ ПРО СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ

Штучний інтелект (ШІ) – це галузь інформатики, що займається розробкою систем, здатних виконувати завдання, які традиційно вимагають людського інтелекту. Попри відсутність єдиного визначення ШІ (адже природа людського інтелекту досі залишається дискусійною у філософії), існує кілька підходів до розуміння, чим займається ця наука. У цій лекції розглянуто основні концепції штучного інтелекту, сфери застосування інтелектуальних технологій, роль ШІ в автоматизації, історію становлення і сучасний стан галузі, питання доцільності використання ШІ, а також поняття інтелектуального агенту.

1. Підходи до розуміння інтелекту та поняття штучного інтелекту

Штучний інтелект як науковий напрям не має загальноприйнятого суворого визначення. Майже кожен дослідник пропонує власне бачення того, що слід розуміти під ШІ, і кожне визначення акцентує різні аспекти. На рисунку 1 показано взаємозв'язок між трьома ключовими поняттями: штучний інтелект (AI), машинне навчання (ML) та глибинне навчання (DL). Штучний інтелект охоплює всі підходи до створення розумних систем, машинне навчання є його підмножиною, що реалізує алгоритми навчання на даних, а глибинне навчання є спеціальним напрямом ML, що використовує штучні нейронні мережі для автоматичного формування висновків без участі людини.

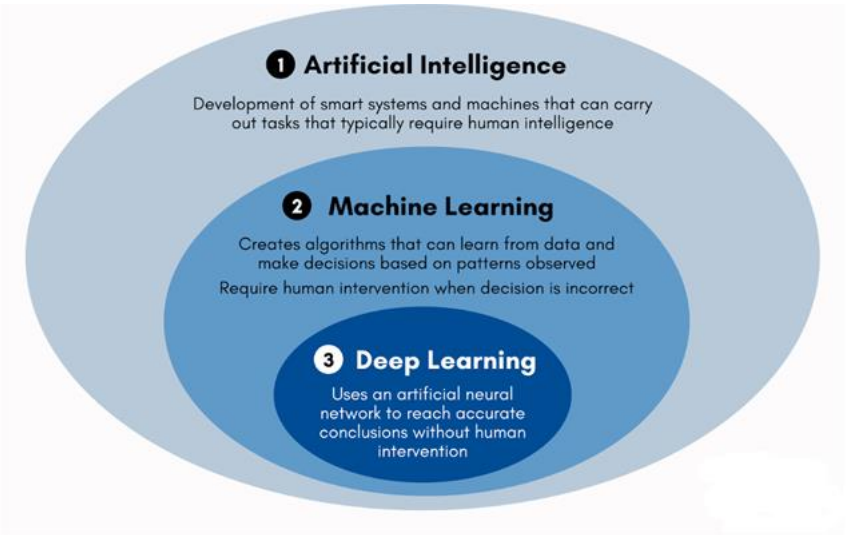


Рисунок 1. Взаємозв'язок понять: штучний інтелект, машинне навчання та глибинне навчання

Основними підходами до розуміння інтелекту та визначення ШІ є:

Інтелект як здатність вирішувати складні задачі. Згідно з одним підходом, штучний інтелект вивчає методи розв'язання завдань, що потребують людського розуміння та інтелектуальних зусиль. Тобто метою є навчити комп'ютер вирішувати задачі на зразок тих, що містяться в тестах на IQ, опанувати аналогії, дедукцію, індукцію, накопичувати знання та застосовувати їх на практиці. Таким чином, будь-яка задача, для якої невідомий алгоритм вирішення, апріорі належить до сфери ШІ.

ШІ як пошук рішень для погано формалізованих задач. Інший підхід визначає ШІ через коло задач, для яких не існує точного або ефективного алгоритму розв'язання за прийнятний час (наприклад, NP-повні задачі). У таких випадках використовуються евристичні методи та машинне навчання. Інакше кажучи, ШІ займається вирішенням нетривіальних проблем, що не піддаються класичному алгоритмічному розв'язанню в силу обмежень часу чи пам'яті.

Імітація людського мозку та поведінки. Історично одним із перших було визначення ШІ як моделювання людської вищої нервової діяльності на комп'ютері. У цьому підході наголошується, що машини можуть бути створені за аналогією з людським мозком – наприклад, шляхом побудови штучних нейронних мереж, що наслідують структуру і функції біологічних нейронів. Так само популярною є ідея імітації людської поведінки: вважається, що машина вважається інтелектуальною, якщо її поведінка не відрізняється від людської. Класичним прикладом є тест Тюрінга, згідно з яким комп'ютер можна визнати розумним, якщо при спілкуванні людина не зможе відрізнити його від іншої людини.

ШІ як системи, що оперують знаннями та навчаються. Сучасні визначення часто описують ШІ як системи, здатні маніпулювати знаннями і навчатися з досвіду. Йдеться про створення програм, які накопичують бази знань, вміють робити логічні висновки та самовдосконалюються. Приклад – експертні системи, які накопичують знання фахівців і можуть приймати рішення на рівні експерта людини. Здатність до навчання взагалі розглядається як один із ключових критеріїв інтелектуальних систем.

Агентно-орієнтований підхід. Починаючи з 1990-х років, поширення набув підхід, в центрі уваги якого – інтелектуальний агент, що діє в певному середовищі. У цьому розумінні ШІ – це наука про методи, які дозволяють агенту виживати і досягати своїх цілей у довкіллі. Наголос робиться на алгоритмах пошуку рішень і прийняття оптимальних дій, аби агент поводився раціонально. По суті, цей підхід визначає інтелект як раціональну дію: машина вважається інтелектуальною, якщо вона здатна діяти оптимально для досягнення своїх цілей у складних умовах.

Крім наведених підходів, у філософії ШІ виділяють поняття сильного та слабкого штучного інтелекту. Слабкий ШІ – це система, яка лише імітує мислення та розв'язує конкретні завдання, тоді як сильний ШІ означає гіпотетичну машину, здатну мислити, розуміти та усвідомлювати на рівні людини. Питання "чи може машина мислити?" було поставлене ще Аланом Тюрінгом у 1950 році, і відтоді дискусія триває. Прихильники сильного ШІ вважають, що у майбутньому комп'ютери зможуть досягти людського рівня розумності, тоді як прихильники слабкого ШІ розглядають теперішні системи лише як інструменти, що моделюють окремі інтелектуальні функції, не володіючи справжнім розумінням.

2. Сучасний стан та коротка історія розвитку ШІ

Історія штучного інтелекту як окремої галузі науки налічує менше століття, проте ідеї створення штучного розуму виникали ще у міфах та фантазіях минулого. Як наука, ШІ був започаткований у 1956 році – саме тоді на Дартмутській конференції було вперше публічно вжито термін "artificial intelligence" і окреслено програму досліджень. Перші десятиліття розвитку ШІ пройшли під знаком ентузіазму та великих сподівань. У 1950-60-х роках були створені перші програми, що демонстрували елементи інтелекту: алгоритми доведення теорем, моделі гри в шахи, системи розв'язування алгебраїчних задач. Одним з проривів стала поява у 1966 році робота Shakey – першого мобільного робота, здатного аналізувати оточення і самостійно планувати свої дії. Проєкт Shakey об'єднав різні напрямки (комп'ютерний зір, планування, логічне виведення) і показав, що машина може володіти базовою автономною поведінкою. Цей робот вважається "першою електронною особистістю" та важливою віхою в еволюції ШІ.

Однак розвиток ШІ йшов нерівномірно, синусоїдально – за періодами ейфорії наставали етапи розчарування, так звані "зими ШІ". В 1970-х роках перший спад інтересу був спричинений тим, що реальні результати не виправдали завищених очікувань: виявилося, що багатьом "розумним" задачам (наприклад, загальне розуміння мови) машини тоді ще не могли навчитися через обмеженість обчислювальних ресурсів і нерозуміння складності проблеми. У 1980-х стався новий підйом завдяки успіхам експертних систем – програм, які на основі баз знань експертів давали поради у вузьких галузях (медицина, геологія, фінанси). Експертні системи здобули популярність у промисловості та бізнесі, що спричинило інвестиційний бум. Проте наприкінці 1980-х – на початку 1990-х ринок експертних систем обвалився (наступила друга "зима ШІ"), знову ж через обмеження: такі програми погано масштабувалися і не могли навчатися, їх створення було дорогим.

Ситуація радикально змінилася з кінця 1990-х – початку 2000-х років, коли на авансцену вийшли методи машинного навчання та значно зросли обчислювальні потужності комп'ютерів. 1997 року відбулася символічна подія – шаховий суперкомп'ютер IBM Deep Blue переміг чемпіона світу Гаррі Каспарова, продемонструвавши силу алгоритмів і обчислень. У 2000-х прогрес ШІ

прискорився завдяки експоненційному зростанню даних (епоха *Big Data*) та удосконаленню алгоритмів нейронних мереж. Ключовим поворотним моментом став 2012 рік: нейромережа глибокого навчання виграла з великим відривом змагання з розпізнавання зображень ImageNet. Це започаткувало бум глибокого навчання (deep learning) – багатошарові нейронні мережі почали встановлювати нові рекорди в комп'ютерному зорі, розпізнаванні мови, машинному перекладі. В 2016 році програмний ШІ-агент AlphaGo компанії DeepMind уперше переміг чемпіона світу з гри го – до того вважалося, що ця древня гра надто складна для комп'ютера через астрономічну кількість можливих комбінацій. Перемога AlphaGo стала символом того, що машинний інтелект перевершує людський у дедалі складніших завданнях, де потрібна не тільки сила обчислень, а й елементи інтуїції (го вимагала стратегічного бачення позиції). У наступні роки системи ШІ досягли чи перевершили людей у ряді специфічних сфер: від аналізу медичних знімків та складання юридичних документів до управління складними технічними об'єктами.

Сьогодні розвиток штучного інтелекту перебуває на черговому підйомі і справляє все більший вплив на різні сторони життя. Сучасні алгоритми здатні навчатися на величезних обсягах даних і демонструють вражаючі результати. Одним із найпомітніших трендів останніх років є генеративні моделі ШІ, зокрема великі мовні моделі (LLM). У 2022–2023 роках широкого розголосу набув чат-бот ChatGPT від OpenAI, створений на основі моделі GPT-3.5, а згодом і GPT-4. Ця система показала здатність підтримувати змістовну розмову, писати тексти різного стилю, розв'язувати задачі і навіть складати програмний код. Деякі дослідники в березні 2023 року назвали нецензуровану версію ChatGPT на базі GPT-4 (доступну для розробників) ранньою та неповною версією сильного ШІ (AGI). Тобто виникла думка, що ми наближаємося до створення систем, близьких за універсальністю до людського інтелекту. Хоча це твердження дискусійне, безсумнівно одне: моделі на кшталт GPT-4 продемонстрували якісно новий рівень здібностей машин до обробки мови, знань і логіки.

Стрімкий прогрес ШІ супроводжується величезним інтересом з боку як бізнесу, так і держав. Інвестиції в штучний інтелект зростають експоненційно, технологічні корпорації змагаються у створенні дедалі потужніших моделей. ШІ проник у смартфони, побутову техніку, онлайн-сервіси – користувачі часто навіть не підозрюють, що при пошуку в Інтернеті, фільтрації спаму чи рекомендації фільмів їм допомагають алгоритми машинного навчання. На думку деяких футурологів, зокрема Рея Курцвейла, ШІ може досягти рівня людського інтелекту вже в найближчі десятиліття (Курцвейл прогнозував до 2029 року). Хоча ці прогнози залишаються спірними, загальна тенденція така, що машинний інтелект нарощує можливості в геометричній прогресії.

Сьогодні ШІ перевершує людей у багатьох вузьких завданнях і активно асистує фахівцям. Важливо відзначити, що успіхи ШІ стимулюють і увагу до питань

регулювання та безпеки. Різні країни та міжнародні організації розпочали роботу над нормами використання ШІ. Так, 31 жовтня 2023 року Організація Об'єднаних Націй створила консультативний орган з питань штучного інтелекту для вироблення рекомендацій щодо його безпечного застосування. А 2 лютого 2024 року держави Європейського Союзу одноголосно схвалили проєкт закону про ШІ – це, фактично, перша спроба законодавчо окреслити рамки технології на наднаціональному рівні. Такий інтерес регуляторів підкреслює, наскільки впливовим фактором став ШІ у суспільстві.

Таким чином, сучасний стан ШІ характеризується широким впровадженням, значними досягненнями та новими викликами. Штучний інтелект уже нині допомагає лікарям, інженерам, вченим, споживачам у різних ролях – від невидимого "двигуна" сервісів до автономних агентів. І хоча до створення справжнього універсального штучного розуму (AGI) ще є наукові та технологічні бар'єри, темпи розвитку дають підстави говорити про наближення ери, де ШІ стане повсюдним інфраструктурним елементом, подібно до електрики чи Інтернету.

3. Місце штучного інтелекту в автоматизації

Розвиток штучного інтелекту тісно пов'язаний з прогресом автоматизації. Автоматизація передбачає передачу машинами рутинних або складних для людини функцій – спочатку це були суто механічні чи розрахункові операції, але з появою ШІ автоматизація перейшла на новий рівень, включивши неструктуровані задачі, що вимагають елементів мислення. Місце штучного інтелекту в автоматизації можна охарактеризувати через кілька аспектів: Від алгоритмічної автоматизації до інтелектуальної. Традиційні автоматизовані системи діяли за чітко заданими алгоритмами: машина виконує наперед визначену послідовність команд і не виходить за її межі. Штучний інтелект дозволив додати гнучкість і адаптивність. Якщо раніше автоматизація обмежувалась строго формалізованими процесами (наприклад, конвеєр на заводі виконує одні й ті самі дії), то з використанням ШІ автоматика здатна реагувати на зміни умов, навчатися на нових даних і приймати рішення в реальному часі. Таким чином, ШІ став ключем до інтелектуалізації автоматизованих систем. Комп'ютерні програми тепер можуть самостійно обирати стратегію дій з-поміж багатьох варіантів, працюючи в умовах неповної інформації чи невизначеності, чого вимагали нові завдання автоматизації.

Промислові роботи та роботизація виробництва. Як уже згадувалося, промислові роботи – один з найважливіших напрямів автоматизації – набули "інтелектуальних" можливостей завдяки ШІ. Сьогодні промисловий робот оснащений не лише механічним маніпулятором, а й системами зору (камери, датчики), сенсорами силового навантаження, контролерами з програмами штучного інтелекту. Це дозволяє йому не тільки повторювати запрограмовані рухи, а й самостійно коригувати дії залежно від ситуації. Наприклад, робот може розпізнавати деталі за допомогою машинного зору і підлаштовувати траєкторію

руху для їх схоплювання. У такий спосіб ШІ підвищує гнучкість автоматизації: одна й та сама роботизована лінія може переналаштовуватися під різні продукти, навчатися нових операцій. Промислові роботи з високим рівнем “інтелектуальності” стали не тільки засобом підвищення ефективності, а й спричинили глибокі соціально-економічні зміни у сфері праці, адже дедалі більше рутинних або небезпечних робіт виконують машини. Варто підкреслити, що поширення таких роботів величезне – світове їх число обчислюється сотнями тисяч і продовжує зростати, що свідчить про ключову роль ШІ у сучасній автоматизації.

Розумні автоматизовані системи управління. ШІ інтегрується в АСУ – автоматизовані системи управління підприємствами, технологічними процесами, інфраструктурою. Якщо класичні АСУ виконували контроль за суворими правилами, то зараз впроваджуються інтелектуальні модулі, що аналізують поточні дані та прогнозують розвиток ситуації. Наприклад, на електростанціях системи з елементами ШІ можуть передбачати перевантаження мережі і завчасно перерозподіляти потужності, в міському господарстві – керувати освітленням чи опаленням залежно від прогнозу погоди та споживання. ШІ робить автоматизацію адаптивною: система не просто діє за програмою, а й оптимізує свою роботу, навчаючись на історичних даних. За допомогою ШІ забезпечується оптимальний і життєздатний вибір комбінації сенсорів та виконавчих механізмів, а також координація їх спільного функціонування. У результаті автоматизовані процеси стають значно ефективнішими, ніж при жорсткій програмі дій.

Людина в контурі “людина-машина”. Цікавим аспектом є зовнішня інтелектуалізація комп’ютерів – коли штучний інтелект використовується, щоб зробити автоматизовані системи зручнішими для людини. Мова про інтелектуальні інтерфейси та системи підтримки рішень. Наприклад, на сучасних виробництвах оператор все ще відіграє роль, але інтелектуальні помічники підказують йому оптимальні дії, відслідковують помилки, пропонують рішення нестандартних ситуацій. Такий симбіоз людини і ШІ підвищує загальну надійність автоматизованих комплексів. Як зазначають дослідники, впровадження інтелектуального інтерфейсу в АСУ та системи проєктування виводить ефективність використання цих систем на новий рівень. Отже, ШІ займає місце не лише “всередині” автономних автоматів, а й на стику “людина–машина”, полегшуючи їх взаємодію.

В даний час виділяють чотири основні напрямки, за якими ведуться дослідження в галузі ШІ:

- моделювання окремих функцій творчих процесів;
- зовнішня інтелектуалізація комп'ютерів;
- внутрішня інтелектуалізація комп'ютерів;

- цілеспрямована поведінка роботів та різноманітних електронних пристроїв (смартфонів, гаджетів та ін.).

Перший напрямок почав розвиватися в ШІ раніше за інші; саме він і породив цей термін: моделювання на комп'ютерах окремих функцій творчих процесів (гра в шахи, шашки, доміно та ін., автоматичний доказ теорем, автоматичний синтез програм, аналіз і синтез музичних творів, автоматичний переклад, розпізнавання образів і ін.).

Другий напрямок утворюють фундаментальні і прикладні дослідження, спрямовані на підвищення рівня взаємодії людини з комп'ютерними системами і пристроями в рамках вдосконалення функцій діалогового інтерфейсу. Інтелектуальний інтерфейс виводить на новий рівень ефективність використання автоматизованих систем управління (АСУ), систем автоматизованого проектування (САПР), автоматизованих систем наукових досліджень (АСНД) і оперативного управління виробництвом в цілому.

Промислові роботи стали не тільки однією з рушійних сил автоматизації, а й найважливішим засобом здійснення глибоких соціально– економічних змін в сфері праці. Розробка і виробництво промислових роботів з високим рівнем інтелектуальності та багатофункціональності, які мають вражаючу надточність, дозволили підняти продуктивність праці на небувалу висоту. Кількість промислових роботів, що випускаються фірмами Японії (FANUC, KAWASAKI, MOTOMAN, OTC DAIHEN, PANASONIC), Німеччини (KUKA), США (KC ROBOTICS, TRITON MANUFACTURING, KAMAN CORPORATION), Швеції (ABB) і ряду інших країн вже перевищує сотні тисяч найменувань (рис. 2).

Використовуючи акумульовані в комп'ютерах і базах даних знання про розвиток предметних областей, які їх цікавлять, фахівці багатьох галузей науки отримують можливість, не виходячи за межі мови своїх особистих предметних областей (підмов природної мови), здійснювати розпізнавання та діагностику процесів в складних системах, приймати оптимальні рішення, формулювати плани дій, висувати гіпотези, а також виявляти закономірності в результатах спостережень. Ці можливості реалізуються експертними системами, які стали інтенсивно поширюватися в важко формалізованих галузях знань, наприклад, медицині, освіті, технічних дисциплінах і т.д.

Рисунок 2. Роботи японської компанії FANUC, призначені для зварювання, навантаження, сортування, транспортування та інших точних робіт

Третій напрям використання штучного інтелекту вирішує проблеми побудови комп'ютерів нових поколінь, оскільки для вирішення завдань створення застосувань ШІ важливі і самі апаратні засоби, і нові методи обробки символьної інформації. Як правило, в подібних системах використовується інформація, представлена в символьній формі: літери, слова, знаки, рисунки. Це відрізняє область штучного інтелекту від областей, в яких традиційно комп'ютерам довіряється обробка даних в числовій формі. У системах ШІ передбачається наявність вибору між багатьма варіантами можливих рішень в умовах невизначеності, що вимагає принципово нових архітектурних побудов обчислювальних пристроїв і програмних компонентів для управління ними.

Так, наприклад, для операційної системи iOS смартфонів iPhone (Apple) розроблений персональний помічник і питально–відповідаюча система Siri (sɪəri; рос. Сірі, від англ. **S**peech **I**nterpretation and **R**ecognition **I**nterface – інтерфейс розпізнавання і інтерпретації мови). Цей застосунок використовує обробку природної мови, щоб відповідати на питання і видавати рекомендації. Крім того, Сірі пристосовується до кожного користувача індивідуально, вивчаючи його вподобання та телефонні контакти протягом довгого часу (рис. 3).

Рисунок 3. Приклад результату взаємодії користувача смартфону з персональним помічником Сірі

Четвертий напрямок додатків ШІ пов'язано з створення інтелектуальних роботів для різних областей діяльності людини. Ця науковотехнічна проблема вимагає розробки, як спеціалізованих обчислювальних засобів, так і цілого комплексу складних технічних, механічних, програмних і енергетичних систем: сенсорів, рушіїв і т.д. Як і всі системи ШІ, інтелектуальні роботи орієнтовані на використання знань. Наприклад, знання про зовнішнє середовище надходять в бортові комп'ютери роботів від численних сенсорів (зорових, акустичних, радіолокаційних, тактильних і т.д.) (рис. 4).

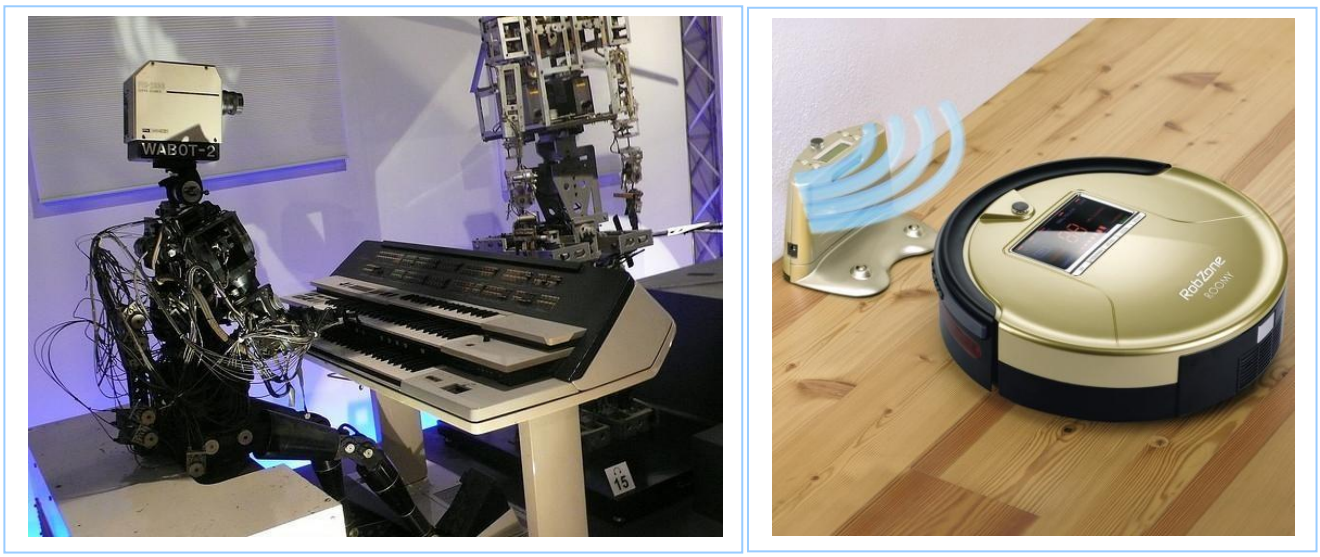


Рисунок 4. Робот–піаніст (зліва) і робот–пилосос (праворуч)

Штучний інтелект, як основа нової інформаційної технології, примножує інтелектуальні ресурси суспільства, оскільки взаємодія користувача з обчислювальними пристроями своєю професійною мовою підвищує рівень інтелекту користувача і розширює його можливості в сфері формування нових елементів логічного висновку. І якщо раніше комп'ютери були основою індустрії обробки даних, то зараз, у зв'язку з активним використанням ідей і методів ШІ, стало правомірним говорити про створення на базі комп'ютерних пристроїв індустрії робототехніки і технологій проектування інтелектуальних систем. Зазначимо основні галузі застосування штучного інтелекту.

Підсумовуючи, штучний інтелект сьогодні є невід’ємною складовою сучасної автоматизації. Якщо епоху обчислювальної техніки XX століття інколи називали “ери обробки даних”, то завдяки ідеям та методам ШІ правомірно говорити про формування нової індустрії – робототехніки та інтелектуальних систем. Саме інтеграція ШІ дозволяє автоматизувати ті сфери, які раніше потребували обов’язкової участі людини, і створювати системи, здатні самостійно планувати і приймати рішення у складному середовищі. Штучний інтелект не замінює класичну автоматизацію, а доповнює і розширює її, надаючи автоматизованим системам риси гнучкості, навчання та здатності до творчого розв’язання задач.

Отже, штучний інтелект можна узагальнено визначити як галузь науки і технологій, що прагне навчити машини “мислити” або діяти розумно. Це охоплює як розробку алгоритмів для вирішення інтелектуальних задач, так і побудову моделей, що імітують людське пізнання, та створення автономних агентів, здатних раціонально діяти в навколишньому середовищі.

4. Сфери застосування інтелектуальних технологій

Штучний інтелект із концептуальної сфери наукових досліджень перетворився на практичний інструмент, що знаходить застосування в різноманітних галузях. Сучасні інтелектуальні технології впроваджуються скрізь – від промисловості й медицини до освіти, бізнесу та повсякденного життя. Розглянемо основні сфери застосування ШІ та приклади їх використання:

- **Промисловість та виробництво.** У виробничому секторі ШІ використовується для автоматизації складних технологічних процесів, прогнозування обслуговування обладнання та оптимізації логістики. Знаковим прикладом є промислові роботи – роботизовані маніпулятори на конвеєрах і заводах. Сучасні промислові роботи, оснащені елементами штучного інтелекту, стали одними з головних рушійних сил автоматизації виробництва. Вони здатні виконувати зварювання, складання, сортування, транспортування деталей з надзвичайною точністю, що підняло продуктивність праці на небувалий рівень. Так, у світі вже використовується сотні тисяч роботів у різних країнах (Японія, Німеччина, США, Швеція тощо) для виконання рутинних та точних операцій на виробництві.

- **Транспорт і робототехніка.** Інтелектуальні системи керування транспортом охоплюють як автономні транспортні засоби (безпілотні автомобілі, дрони), так і оптимізацію трафіку. Наприклад, у автомобілебудуванні ШІ допомагає реалізувати автопілот – програму, що на основі сенсорів та камер здатна самостійно вести автомобіль. Хоча повністю автономні автомобілі все ще проходять випробування, елементи ШІ (системи допомоги водієві, адаптивний круїз-контроль, розпізнавання дорожніх знаків) вже масово використовуються. У сфері робототехніки ШІ дозволив створити побутових роботів (наприклад, робот-пилосос), роботів-сервісменів, військових і рятувальних роботів, які можуть орієнтуватися в просторі, розпізнавати перешкоди та приймати рішення в реальному часі. Таким чином, штучний інтелект розширює можливості автоматизованих машин, надаючи їм адаптивність та автономність.

- **Медицина та охорона здоров'я.** Інтелектуальні технології здійснили революцію в медичній діагностиці, прогнозуванні та персоналізованому лікуванні. Експертні системи та сучасні алгоритми глибокого навчання здатні аналізувати медичні знімки (МРТ, рентген, КТ) і ставити попередній діагноз, допомагаючи лікарям виявляти захворювання на ранніх стадіях. Наприклад, у 2018 році система ШІ (BioMind) з точністю 87% за 18 хвилин поставила діагнози за знімками пухлин головного мозку, перевершивши команду провідних лікарів, яка досягла лише 66% точності за 50 хвилин. Такі результати демонструють потенціал ШІ в медицині: від аналізу медичних даних і зображень до розробки нових лікарських препаратів (алгоритми для

дизайну білків та пошуку ліків уже генерують перспективні молекули). Крім того, чат-боти та голосові помічники на основі ШІ застосовуються для консультування пацієнтів, моніторингу стану здоров'я, а робототехніка – для допомоги в хірургії (робот-хірург Da Vinci) та реабілітації.

- **Освіта і наука.** ШІ стає невід'ємним інструментом в освіті, забезпечуючи персоналізоване навчання. Інтелектуальні навчальні системи (так звані *intelligent tutoring systems*) підлаштовують матеріал під рівень та прогрес кожного учня, надають підказки і рекомендації. Наприклад, адаптивні платформи навчання використовують алгоритми машинного навчання, щоб визначити прогалини в знаннях студента і запропонувати вправи для їх заповнення. У наукових дослідженнях ШІ допомагає в аналізі великих масивів даних (наприклад, виявлення закономірностей у біологічних чи фізичних експериментах), генерації гіпотез та навіть у відкриттях. Показовим є те, що алгоритми DeepMind у 2021 році допомогли математикам розв'язати складні задачі в теорії вузлів та теорії представлень, виявивши нові зв'язки, які були недоступні людському розуму Це свідчить про здатність ШІ сприяти науковому прогресу, особливо в міждисциплінарних сферах.

- **Бізнес, фінанси та обслуговування.** У комерційному секторі штучний інтелект використовується для аналізу даних і підтримки прийняття рішень. Бізнес-аналітика на основі ШІ дозволяє прогнозувати ринок, сегментувати клієнтів, виявляти шахрайство у фінансових транзакціях. Банки застосовують алгоритми машинного навчання для оцінки кредитоспроможності клієнтів та управління ризиками. В обслуговуванні клієнтів масово впроваджуються віртуальні помічники та чат-боти. Ці програми розуміють природну мову та можуть спілкуватися з користувачами, відповідаючи на типові запитання або виконуючи прості операції (наприклад, оформлення замовлення, бронювання квитків). Яскравий приклад – голосові асистенти у смартфонах (Apple Siri, Google Assistant, Amazon Alexa): вони використовують технології обробки природної мови для відповідей на запити і навіть адаптуються до користувача, вивчаючи його вподобання з часом. Такі системи значно підвищують якість сервісу та доступність послуг.

- **Розваги, медіа та творчість.** ШІ дедалі частіше виступає як інструмент для створення контенту. У кіно- та ігровій індустрії алгоритми штучного інтелекту генерують реалістичну графіку, моделюють поведінку неігрових персонажів у відеоіграх та навіть пишуть сценарії. Вже існують приклади, коли повністю згенерованим ШІ текстам та зображенням надається увага на рівні з людськими творами: у 2023 році український журнал опублікував номер, контент якого цілком згенерований штучним інтелектом, а у 2024 році на виставці цифрового мистецтва в Лондоні був представлений витвір, створений алгоритмом на базі GPT-4. Це свідчить про зростання ролі ШІ у

сфері творчості – від музики (генерація мелодій) до живопису (нейромережі, що малюють картини) та літератури (мовні моделі, що пишуть тексти). Водночас у галузі ігор штучний інтелект давно став стандартним компонентом: типові задачі ігрового ШІ включають знаходження шляху для персонажа, імітацію реалістичної поведінки опонентів, прийняття стратегічних рішень тощо.

Отже, інтелектуальні технології проникли майже в усі сфери діяльності людини. Вони забезпечують оптимальні та адаптивні рішення в управлінні складними системами, допомагають обробляти величезні масиви інформації та підтримувати прийняття рішень у невизначених ситуаціях. Завдяки ШІ комп'ютери нині не просто рахують чи зберігають дані – вони набувають рис "інтелектуальних помічників" людини, підсилюючи наші можливості в різних галузях – від щоденних побутових справ до наукових відкриттів.

5. Представлення знань в системах штучного інтелекту

В основі розробки і використання обчислювальної техніки традиційно лежать такі поняття, як *програми* і *дані*. При цьому перші призначені для обробки другого. На ранніх етапах розвитку даної галузі програміст, як правило, сам розробляв програму і сам вводив в неї необхідні дані.

Потім відбулася серйозна зміна в їх взаємодії – з появою *баз даних* різної структури (ієрархічних, мережевих, реляційних, об'єктноорієнтованих) і систем управління базами даних (СКБД) *дані* були відокремлені від *програм*. Для вирішення такого завдання використовувалися засоби опису даних, що містяться в мовах програмування. Такі мови, як ФОРТРАН і АЛГОЛ, мали засоби опису відносно простих структур даних в пам'яті електронно-обчислювальних машин (ЕОМ). Наприклад, цілі – INTEGER, дійсні – REAL і т.д. Більш складні засоби опису ієрархічних структур даних вбудовані в мови КОБОЛ, СІ, ПАСКАЛЬ. Модула-2, АДА і ін. В цих мовах також є засоби для конструювання структур даних самим користувачем.

Паралельно з вищевказаними процесами розвивалися концепції представлення даних у зовнішній пам'яті великих за розмірами ЕОМ, які паралельно з удосконаленням компонентів елементної бази і переходу від ламп, діодів і тріодів до інтегральних мікросхем трансформувалися в невеликі і компактні персональні комп'ютери (рис. 5).

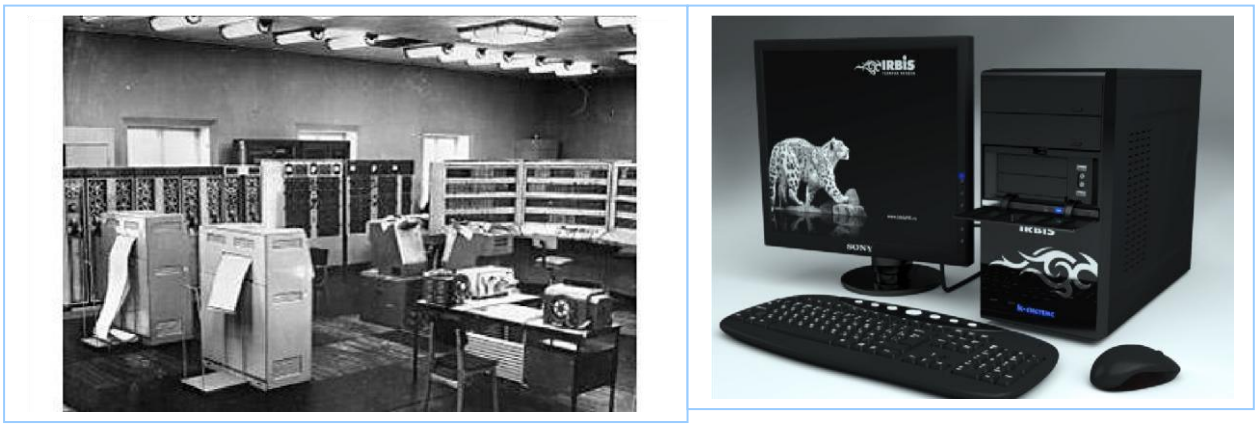


Рисунок 5. ЕОМ БЕСМ–6 (зліва) і персональний комп'ютер (праворуч)

Іншим компонентом, що дозволив остаточно відокремити дані від програм, став пристрій, який називається магнітний диск. Революційність його появи пояснювалася тим, що після знеструмлення ЕОМ всі дані, присутні в її пам'яті, пропадали.

Магнітний диск забезпечив зберігання інформації і після вимикання комп'ютера. Більш того, накопичувач на жорстких магнітних дисках (НЖМД) може бути витягнутий з одного комп'ютера і підключений до іншого, де він може надавати доступ не тільки до записаних на ньому даних, але й до операційних систем і прикладних програм, записаних на ньому до цього (рис. 6).

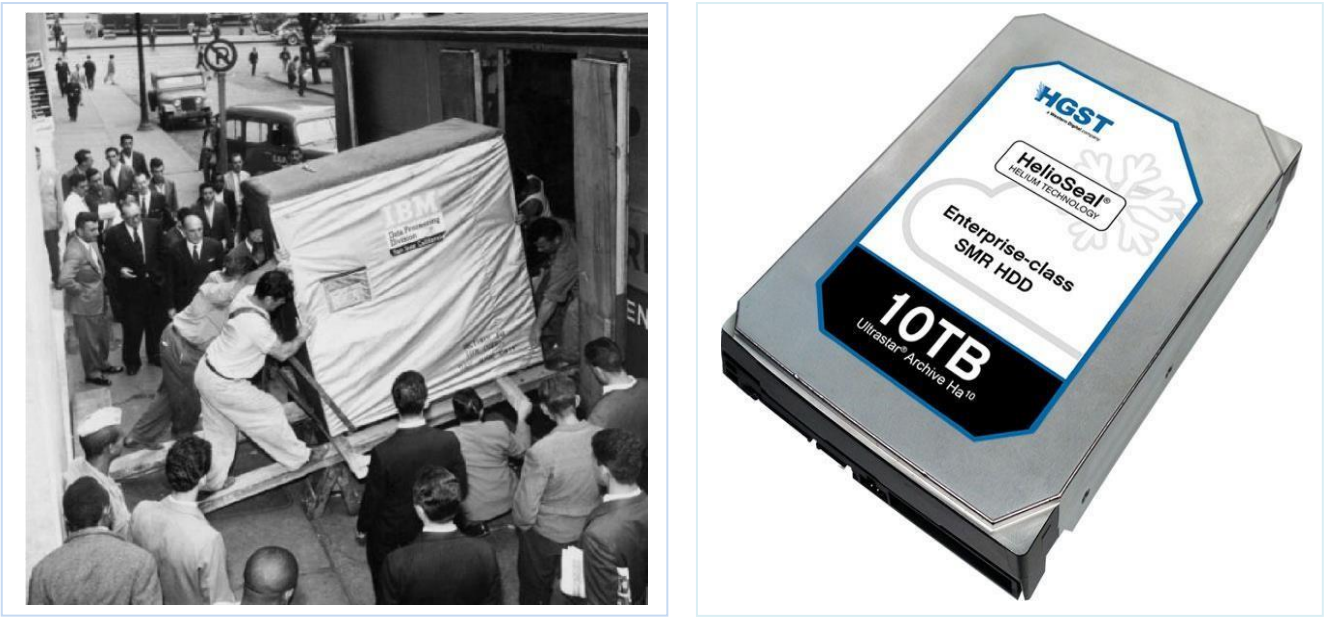


Рисунок 6. Завантаження для перевезення 5-мегабайтного жорсткого диска компанії IBM, 1965 рік (США) (ліворуч) та 10-ти терабайтний диск фірми Western Digital, 2016 рік (праворуч)

На цьому етапі фундаментальним поняттям інформатики стало поняття інформаційного масиву – *файлу*, що є спеціально організованою структурою даних, яка розпізнається комп'ютером як єдине ціле, має ім'я і містить всі необхідні структуровані записи даних про різні об'єкти, з якими веде роботу комп'ютер. Файл можна трактувати як інформаційну модель деякого об'єкту, наприклад, програми, документа, таблиці, музичного твору і ін.

Таким чином, уявлення даних у зовнішній пам'яті комп'ютера пройшло через три етапи:

1) на першому етапі, способи формування записів даних в файлах, ведення файлів і організація доступу до них повністю визначалися в конкретних програмах користувача;

2) на другому етапі, управління файлами і організацію доступу до них стали здійснювати операційні системи комп'ютерів;

3) на третьому, файли стали елементами створюваних баз даних і розвинених систем управління ними (СУБД – системи управління базами даних).

4) При цьому, була реалізована можливість ефективної роботи з великими базами даних (зокрема, з інтегрованими базами, що містять різнорідні дані), що обробляються в інтересах цілого підприємства, галузі і т.д., і призначеними для використання в прикладних задачах.

Таким чином, на першому етапі розвитку методів обробки даних створення, підтримка і організація доступу до них, як на логічному, так і на фізичному рівнях, цілком покладалися або на розробника, або на користувача кожної окремої програми. Робота з даними, що відносяться до конкретної програми, а тим більше їх використання в інших програмах були вкрай трудомісткими і малоефективними.

Поліпшення становища сталося на другому етапі, коли частина турбот, що були пов’язані з обробкою даних у зовнішній пам'яті комп'ютерів (в основному на фізичному рівні) взяла на себе операційна система (ОС). Однак робота з інтегрованими даними стала реальністю лише на третьому етапі розвитку інформаційних систем (ІС), коли з'явилася можливість ефективно організовувати бази даних зі складною структурою, а в рамках СУБД були розроблені потужні засоби для роботи з наборами різноманітних відомостей про навколишній світ. Це зробило виправданим і ефективним існування в інформаційних системах наборів даних, незалежних від прикладних програм, в яких ці дані створюються і використовуються, а також дозволило технологічно відокремити один від одного різні програми (що створюють, підтримують і використовують дані). З'явилася можливість ефективно зв'язувати програми для обробки даних з самими цими даними і вже викликати програми на основі і виходячи з існуючих даних, а не навпаки, як було раніше. Для забезпечення функціонування величезних масивів

даних, накопичених і продовжують накопичуватися організаціями та науково–дослідними інститутами в даний час використовуються т.зв. центри обробки даних (дата-центри, серверні ферми, «хмарні» дата центри і т.д.) (рис. 7).

Рисунок 7. Зовнішній (зліва) і внутрішній вигляд, зі стойками (шафами) дискових накопичувачів (праворуч), одного з дата-центрів компанії Google

І, нарешті, СУБД забезпечили засоби для створення в кожному програмному комплексі проміжного шару програмного забезпечення, що відокремлює власне прикладні програми від використовуваних даних, ефективно реалізує пошук, розміщення та інші операції над даними і тим самим звільняє від цієї діяльності прикладних програмістів. Проміжний шар частково складається з системних програм СУБД, частково добудовується користувачем. Засоби для цього надаються спеціалізованими мовами опису даних і мовами маніпулювання даними, що включаються в СУБД. Такі мови, як, наприклад, SQL (Structured Query Language – мова структурованих запитів), доповнюють традиційні мови програмування засобами для організації та обробки великих груп (наборів) даних. Щоб забезпечити можливість прямого використання цих засобів в прикладних програмах на традиційних мовах програмування, мова опису даних і мова маніпулювання даними часто оформлюються як розширення розвиненої мови програмування – т.зв. «мови, що включається».

В даний час можна говорити про новий етап представлення даних в пам'яті комп'ютера – про створення інформаційно-обчислювальних мереж і на їх основі – розподілених баз даних колективного користування. Це призводить як до зниження витрат на створення і введення баз даних, так і до підвищення якості інформації, що зберігається, оскільки для ведення баз даних можливо залучення більш кваліфікованих працівників. Одночасно різко зростає доступність цієї інформації для користувачів.

З появою систем ШІ з'явилося нові поняття – *«знання»* і *«база знань»* (БЗ). У теорії штучного інтелекту знання – це сукупність інформації про світ, властивості об'єктів, закономірності процесів і явищ, а також правила використання їх для прийняття рішень. Головна відмінність *знань* від *даних* полягає в їх структурованості і активності.

Поява в базі нових фактів або встановлення нових зв'язків може стати джерелом змін в прийнятті рішень. Виникла необхідність якимось чином співвіднести звичні поняття «дані і бази даних» з поняттями «знання і бази знань». Безсумнівно, що дані і структура бази даних певною мірою відображають знання про предметну область і її структуру. Проте, є специфічні ознаки, що відрізняють знання від даних. Такими специфічними ознаками знань у зв'язку з представленням їх в комп'ютерах виділяють наступні чотири ознаки:

- внутрішня інтерпретованість;
- структурованість;
- зв’язність;
- активність.

Якщо звернутися до наборів даних, то деякі із зазначених ознак, властиві знанням, будуть справедливими і для них. Наприклад, перша ознака – *інтерпретованість* – явно проглядається у реляційній базі даних, де імена стовпців є атрибутами відносин, імена яких вказані в рядках. Внутрішня інтерпретованість передбачає можливість установки для елемента даних пов'язаної з ним системою імен. Система імен включає в себе індивідуальне ім'я, яке привласнене даній інформаційній одиниці. Наявність системи «надлишкових» імен дозволяє системі штучного інтелекту знати, що зберігається в її базі знань, а, отже, вміти відповідати на нечіткі питання про вміст бази знань.

Другу ознаку – *структурованість* – можна розглядати як властивість декомпозиції складних об'єктів на більш прості і встановлення зв'язків між простими об'єктами, що означає використання відносин «частина-ціле», «клас-підклас», «рід-вид» і т.д. Відносини подібного роду зустрічаються і в ієрархічних, і мережевих базах даних. Ці ж відносини можуть бути реалізовані і в реляційних (табличних) базах даних.

Для третьої ознаки знань – *зв'язності* – практично не можна знайти аналогів у звичайних базах даних. Як правило, наші знання пов'язані не тільки у сенсі структури. Вони відображають закономірності організації взаємодії фактів, процесів, явищ і причинно-наслідкові зв'язки між ними. Зв’язність характеризує можливість встановлення між інформаційними одиницями найрізноманітніших відносин (чітких, нечітких, бінарних, складових і ін.), які визначають семантику і прагматику зв'язків явищ і фактів, а також відносин, що визначають зміст даної системи в цілому.

Що стосується четвертої ознаки – *активності*, то ситуація складається таким чином, що при використанні комп'ютерів – нові знання породжуються програмами (тобто програмно), а дані пасивно зберігаються в пам'яті. Людині властива пізнавальна активність, іншими словами, знання людини активні. І це принципово відрізняє знання від даних. Наприклад, виявлення суперечностей в знаннях стає спонукальною причиною їх подолання і появи нових знань. Таким самим стимулом активності є неповнота знань, що відображається в необхідності їх поповнення. Таким чином, головна відмінність *знань* від *даних* полягає в їх зв'язності і активності, а поява в базі нових фактів, або встановлення нових зв'язків може стати джерелом змін у прийнятті рішень.

Знання, які використовуються для створення системи ШІ, які забезпечує її роботу, зберігаються, модифікуються і виробляються у ній, можуть бути визначені різноманітним чином. У теперішній час використовують три визначення. Знання – це:

- результат, отриманий пізнавальним чином;
- система суджень з основною і єдиною організацією, основана на об’єктивній закономірності;
- формалізована інформація, на яку посилаються або використовують у процесі логічного виводу.

Процес рішення задач за допомогою найпростішої моделі системи ШІ представлений на рис 8.

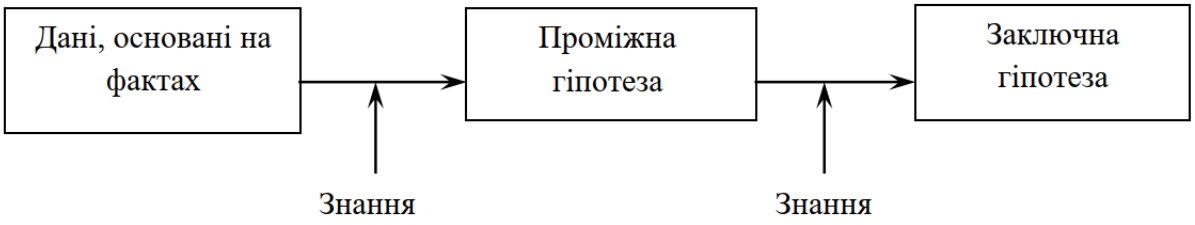


Рисунок 8. Зв’язок між знаннями і висновком при рішенні інтелектуальної проблеми.

У представленій на рисунку постановці задачі, знання – це інформація, на яку посилаються, коли роблять різноманітні висновки на основі існуючих даних за допомогою *логічних висновків*. Якщо такі дії виконуються на основі використання програмних засобів, то знання – це обов’язкова інформація представлена у певній формі. Важливо те, що постановка і рішення будь-якої задачі, пов’язаної з обробкою даних і знань, завжди пов’язані з її «зануренням» у відповідну предметну область.

Предметна область — це частина реального світу, що розглядається у межах заздалегідь визначеного (використовуваного розробником) контексту. Зазвичай це множина всіх предметів, властивості яких і відношення між якими розглядаються у науковій теорії або галузі практичної діяльності спеціаліста. Подумки, предметна область представляється такою, що складається з реальних або абстрактних об'єктів, званих *сутностями*. Так, наприклад, вирішуючи задачу складання розкладу обробки деталей на металорізальних верстатах, ми залучаємо у предметну область, з одного боку, такі сутності, як конкретні верстати, деталі, інтервали часу їх обробки, а з іншого боку – загальні поняття «верстат», «деталь», «тип верстату» і т.д. Об'єднана сукупність самої предметної області, її уявлень і сутностей, а також задач, що розв'язуються у цій області, визначається поняттям *проблемна область*.

При цьому слід розуміти, що *знання* – це закономірності предметної області (принципи, зв'язки, закони), отримані в результаті практичної діяльності і професійного досвіду, що дозволяють фахівцям ставити і вирішувати завдання в цій галузі. Важливо також зазначити, що при вирішенні задач в деякій предметній області знань, останню зручно розділити на дві великі категорії – *факти* і *евристику*.

Перша категорія вказує, зазвичай, на добре відомі в даній галузі обставини, тому знання цієї категорії іноді називають текстовими, маючи на увазі достатню їх освітленість у спеціальній літературі або підручниках.

Друга категорія знань ґрунтується на власному досвіді фахівця у даній предметній області (т.зв. *експерта*), і цей досвід накопичений у результаті багаторічної практики. У так званих експертних системах евристичні знання відіграють вирішальну роль у підвищенні ефективності систем. Іншими словами, в цю категорію входять такі знання, як «засоби зосередження», «засоби видалення непотрібних ідей», «способи використання нечіткої інформації» і т.д., що дозволяють з більшою ефективністю вирішувати завдання. Проте, через недостатню наукову обґрунтованість та відсутність вичерпних відомостей користуватися такими знаннями потрібно обачно.

Знання, крім того, можна розділити на *факти* (фактичні знання) і *правила* (знання для прийняття рішення). Під фактами маються на увазі знання типу «А це А» і вони характерні для баз даних. Під правилами розуміються знання виду «ЯКЩО А – ТО Б». Крім них існують так звані метазнання (знання про знання). Поняття «метазнання» вказує на знання, які стосуються способів використання знань, і знання, які стосуються властивостей знань. Це поняття необхідно для управління базою знань і логічним висновком, виконання функцій ототожнення, навчання і т.д. Дані та структури даних далеко не в повній мірі відображають особливості предметних областей. Хоча, взагалі кажучи, чітку грань між даними і знаннями провести можна не завжди, проте, відмінності між даними і знаннями існують, і ці відмінності привели до появи спеціальних формалізмів у вигляді

моделей представлення знань в комп'ютерах, що відображають в тій чи іншій мірі основні ознаки, що характеризують знання.

6. Доцільність використання ШІ

Питання про доцільність та межі використання штучного інтелекту є надзвичайно важливим у сучасних умовах. З одного боку, ШІ приносить величезні переваги – підвищує ефективність, автоматизує рутинні процеси, відкриває нові можливості в науці та техніці. З іншого боку, його застосування пов’язане з ризиками та викликами, що потребують зваженого підходу. Розгляньмо, в чому полягає доречність впровадження ШІ та які фактори слід враховувати, приймаючи рішення про його використання.

Обґрунтування використання ШІ: переваги. Штучний інтелект доцільно застосовувати там, де він дає якісний виграш порівняно з традиційними методами. Наприклад, якщо задачу неможливо вирішити фіксованим алгоритмом через її складність або варіативність, залучення ШІ виправдане (такими були багато задач із області розпізнавання образів, перекладу, стратегічних ігор – і ШІ успішно їх опанував). Використання ШІ особливо доцільне при роботі з великими масивами даних: алгоритми машинного навчання можуть знайти неявні закономірності, тренди, зробити прогнози, які людині було б дуже складно отримати вручну. Так само ШІ ефективний у задачах, що потребують високої швидкості реакції або безперервної цілодобової роботи – в таких умовах автоматизація з елементами інтелекту перевершує людські можливості. У багатьох сферах (медицина, транспорт, виробництво) впровадження ШІ вже довело свою результативність у вигляді зниження витрат, підвищення точності та якості результату, скорочення часу виконання робіт. Тому в цілому можна стверджувати: доцільно використовувати ШІ там, де він дозволяє робити те, що раніше було неможливо або суттєво покращує існуючі процеси.

Межі і обмеження: коли ШІ може бути недоречним. Водночас важливо усвідомлювати, що ШІ – не панацея і не завжди його застосування виправдане. Існують ситуації, де людський фактор принципово важливий: наприклад, прийняття етичних або творчих рішень, що виходять за рамки формалізованих критеріїв. Надмірна автоматизація може призвести до знеособлення процесів, втрати контролю людиною над критичними рішеннями. Бувають завдання настільки прості або добре формалізовані, що впровадження ШІ лише ускладнить систему і збільшить витрати. Крім того, якість роботи ШІ залежить від даних – якщо дані навчання містять помилки або упередження, алгоритм відтворюватиме їх. Отже, при розв’язанні вузьких та чітко визначених задач часто ефективніші класичні алгоритми, а ШІ доцільніше залишити для складних випадків. Інший аспект – витрати і ресурси: розробка й підтримка систем ШІ може бути дорогою,

вона потребує фахівців, обчислювальних потужностей. Для малого бізнесу чи задач з невеликим масштабом простіші рішення іноді економічно вигідніші. Таким чином, при вирішенні питання “чи варто застосовувати ШІ?” слід проводити аналіз витрат і вигод: якщо інтелектуальна система не дає суттєвого покращення або вимагає невиправдано великих ресурсів, її впровадження може бути недоцільним.

Етичні та соціальні міркування. Окремо треба зважати на етичність використання ШІ. Є сфери, де люди висловлюють обґрунтовані занепокоєння – скажімо, автономна зброя, системи масового стеження, прийняття юридично значущих рішень (кредити, вироки) алгоритмами без належного нагляду. Вчений Джеффрі Гінтон, відомий як “хрещений батько ШІ”, у травні 2023 року залишив компанію Google саме для того, щоб відкрито застерегти суспільство про ризики, пов’язані з розвитком ШІ (рис.9). Зокрема, він та інші експерти попереджають про небезпеку створення автономних бойових систем на базі ШІ, а також про можливість неконтрольованого розповсюдження надто потужних ШІ-моделей. У квітні 2023 року понад 1000 фахівців у галузі ШІ (включно з І. Маском і С. Возняком) підписали відкритого листа із закликом оголосити мораторій на 6 місяців на навчання найпотужніших ШІ-систем з огляду на необхідність розв’язання етичних і безпекових проблем. Ці події підкреслюють: питання “чи доцільно впроваджувати ШІ” має вирішуватися не лише техніко-економічним аналізом, але й з урахуванням суспільних наслідків. Там, де ШІ може нашкодити – підмінити собою людський контакт (як у догляді за хворими), порушити приватність чи права людей, – його використання потребує особливо ретельного обґрунтування і контролю.

Рисунок 9. Видатний науковець у галузі штучного інтелекту.

Відповідальне впровадження ШІ. Сьогодні набуває популярності концепція Responsible AI – відповідального ШІ. Великі технологічні компанії задекларували принципи етичного та безпечного розвитку штучного інтелекту. У

липні 2023 року такі корпорації, як Microsoft, Google та OpenAI, погодили набір заходів щодо безпеки і прозорості ШІ: надання можливості незалежного тестування моделей, інвестування в кібербезпеку, маркування контенту, створеного ШІ (водяні знаки на згенерованих зображеннях/аудіо), і спрямування передових систем ШІ на розв'язання суспільно значущих проблем. Цей крок демонструє розуміння індустрією, що довіра до ШІ є ключовою умовою його подальшої успішної інтеграції. Для розробників і користувачів це означає: доцільно використовувати ШІ, коли існують засоби контролю його роботи, механізми перевірки результатів та можливість втручання людини у критичних ситуаціях. Лише за таких умов можна по-справжньому скористатися перевагами ШІ, мінімізувавши потенційні ризики.

Підсумовуючи, доцільність використання ШІ визначається балансом між його користю та можливими ризиками. Штучний інтелект варто впроваджувати там, де він приносить реальне покращення – підвищує ефективність, безпеку, точність – і водночас потрібно забезпечувати, щоб використання ШІ було прозорим, контрольованим та етичним. Як образно зазначають, ШІ – це дуже потужний інструмент, а потужні інструменти потребують мудрого застосування. Тому головне завдання суспільства і спеціалістів – навчитися використовувати ШІ на благо, залишаючи людині роль остаточного арбітра у важливих рішеннях.

7. Поняття інтелектуального агенту

Однією з базових концепцій сучасного штучного інтелекту є інтелектуальний агент. Цей термін відображає уніфікований підхід до побудови систем ШІ, розглядаючи їх як агентів, що функціонують у певному середовищі. Інтелектуальний агент – це автономна система, здатна сприймати оточуюче середовище, аналізувати ситуацію та виконувати дії для досягнення поставлених цілей. У формальному визначенні, інтелектуальний агент – це система, яка сприймає своє середовище і здійснює дії, що максимізують її шанси на успіх. З цього визначення випливає кілька ключових рис інтелектуального агенту:

- **Наявність сенсорів (сприйняття):** Агент отримує інформацію про середовище через датчики або канали вводу. Це можуть бути фізичні сенсори (камери, мікрофони, датчики температури – для роботів) або програмні "сенсори" (запити до баз даних, отримання даних із Інтернету – для програмних агентів). Сприйняття дає агенту образ стану середовища, на основі якого він буде діяти.

- **Наявність ефекторів (дії):** Агент впливає на середовище через виконавчі механізми. Фізичний агент (наприклад, робот) має ефектори у вигляді моторів, маніпуляторів, коліс тощо – щоб рухатися і змінювати світ довкола. Програмний агент може вживати дії у віртуальному середовищі – надсилати

команди, змінювати дані, генерувати повідомлення. Головне, що внаслідок дій агента змінюється стан середовища (чи то фізичного, чи інформаційного).

- **Автономність та цілеспрямованість:** Інтелектуальний агент функціонує без постійного втручання людини, самостійно обираючи, яку дію виконати далі. В нього є мета або набір цілей, закладених розробником чи сформованих у процесі навчання, і він прагне їх досягти, взаємодіючи із середовищем. Автономність означає, що агент сам приймає рішення на основі своєї програми (стратегії), яка може враховувати як поточні сприйняття, так і накопичений досвід. Наприклад, інтелектуальний пошуковий інтернет-агент може мати мету знайти певну інформацію і автономно переглядати веб-сторінки, поки не виконає завдання.

- **Раціональність:** Агент називається раціональним, якщо він діє так, щоб максимально наблизитися до своєї мети (або максимізувати деяку корисність). Це не обов'язково означає ідеальність – раціональність визначається з огляду на знання агента та обмеження. Але припускається, що інтелектуальний агент вибирає найкращу доступну дію відповідно до своєї оцінки ситуації. В теорії агентів введено поняття функції корисності або цільової функції, яка ранжує можливі стани середовища за ступенем досягнення мети; раціональний агент діє так, щоб очікуваний результат мав найбільшу корисність.

Простий приклад інтелектуального агенту – робот-пилосос. Його середовище – квартира; сенсори – далекоміри, камери або інфрачервоні датчики, що дозволяють виявляти перешкоди та забруднення; ефектори – коліщатка для руху і механізм всмоктування пилу. Мета – прибрати підлогу. Робот-пилосос сприймає кімнату, виявляє сміття, будує маршрут і пересувається, поки не збере сміття, уникаючи зіткнень. Він автономний (його не потрібно постійно скеровувати) і раціональний у своїх діях (намагається охопити всю площу і не повторювати пройдене). Інший приклад – програмний торговий агент на біржі: він отримує дані про ціни (сенсори), може купувати або продавати акції (ефектори – надсилання заявок), має мету максимізувати прибуток і діє автономно за заданою стратегією, реагуючи на ринкові зміни.

Важливо, що поняття інтелектуального агенту є достатньо широким: під нього підпадають і люди, і організації, і програми. Людину теж можна розглядати як агента з біологічними сенсорами та ефекторами. Це узагальнення корисне тим, що дозволяє вивчати принципи інтелекту незалежно від конкретної реалізації. В рамках ШІ розроблено типологію агентів: реактивні агенти, які реагують безпосередньо на поточні стимули; агенти, що використовують модель середовища (вони зберігають внутрішній стан, який відображає попередній досвід); цілеспрямовані агенти, що діють, маючи явно задану мету; агенти, які навчаються, здатні покращувати свою поведінку з досвідом; та багатоагентні системи, де кілька

агентів взаємодіють між собою. На практиці інтелектуальні системи можуть поєднувати ці підходи – наприклад, сучасний робот має і реактивні рефлекси (уникання зіткнення), і цілепокладання (досягти цілі), і навчання для адаптації.

Отже, інтелектуальний агент – це фундаментальна абстракція, що допомагає описати будь-яку систему ШІ як суб'єкт, що самостійно сприймає, аналізує та діє. Такий підхід дозволяє формалізувати задачу штучного інтелекту як конструювання раціональних агентів, здатних ефективно працювати у заданому середовищі. Від простих програм-агентів, що фільтрують електронну пошту, до автономних роботів-дослідників на Марсі – усі вони підпадають під це визначення. Концепція інтелектуального агенту є наріжним каменем сучасного ШІ, оскільки об'єднує різні напрямки (сприйняття, планування, навчання, прийняття рішень) в єдину модель "агент-середовище". Розуміння цієї концепції є ключем до проєктування нових інтелектуальних систем та аналізу їхньої поведінки.

Контрольні запитання:

1. У чому полягають основні підходи до розуміння сутності штучного інтелекту та які критерії визначають його "інтелектуальність"?
2. Які відмінності між поняттями штучний інтелект (AI), машинне навчання (ML) та глибинне навчання (DL), і як вони взаємопов'язані?
3. Яким чином тест Тюрінга історично вплинув на формування концепції "розумної машини"?
4. Охарактеризуйте відмінності між сильним і слабким штучним інтелектом. Які аргументи наводяться на користь існування кожного з них?
5. Проаналізуйте основні етапи розвитку штучного інтелекту - від перших експертних систем до сучасних генеративних моделей. Які події стали переломними?
6. Яке місце займає штучний інтелект у процесах автоматизації? Як ШІ трансформує класичні автоматизовані системи управління (АСУ)?
7. У чому полягає відмінність між поняттями дані та знання в контексті систем штучного інтелекту? Які ознаки відрізняють бази знань від баз даних?
8. Розкрийте сутність поняття інтелектуальний агент. Які його ключові компоненти і яким чином він взаємодіє з навколишнім середовищем?
9. Які сфери сучасної діяльності людини зазнали найбільшого впливу від упровадження інтелектуальних технологій? Наведіть приклади.

ТЕМА 2. МЕТОДИ МАШИННОГО НАВЧАННЯ

1. Загальні питання машинного навчання

Машинне навчання (МН) – це великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних «навчатись». Термін «Машинне навчання» ввів американський вчений у галузі ШІ Arthur Samuel (Артур Самуель) в 1959 як «здатність комп'ютера вчитися, не будучи явно запрограмованим.

Мета МН – передбачити результат за вхідними даними. Чим різноманітніші вхідні дані, тим простіше алгоритму знайти закономірності і тим точніший результат.

На рис.1 представлені основні складові машинного навчання.

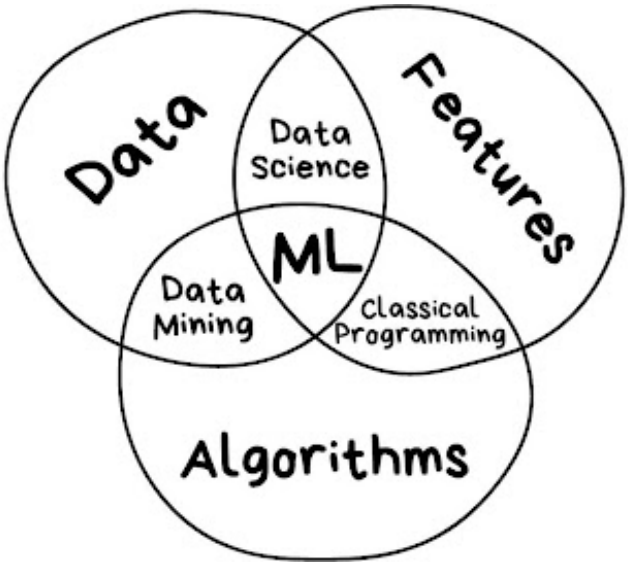


Рисунок 1 – Складові машинного навчання

МН включає класичну теорію розпізнавання образів, штучні нейронні мережі (ШНМ) і підгалузь ШНМ – глибоке навчання.

Три кити машинного навчання

Data (Дані). Хочемо виявляти спам – потрібні приклади спам-листів, передбачати курси акцій – потрібна історія цін, дізнатися інтереси користувача – потрібні його лайки або пости. Даних потрібно якомога більше.

Дані збирають як тільки можуть. Хтось вручну, даних менше, зате без помилок. Автоматично –все, що знайшлося. Великі компанії типу Google, використовують своїх же користувачів для безкоштовної розмітки. За хорошими наборами даних (datasets) йде велике полювання.

Features (Ознаки). Ознаками можуть бути: пробіг автомобіля, стать користувача, ціна акцій, навіть лічильник частоти появи слова в тексті. Алгоритму потрібно, з чим конкретно працювати. Добре, коли дані просто лежать в таблицях – назви їх колонок і є features. Коли ознак багато, модель працює повільно і неефективно. Найчастіше відбір правильних ознак займає більше часу, ніж все інше навчання. Але бувають і зворотні ситуації, коли

розробник сам відбирає тільки «правильні» на його погляд ознаки і вносить в модель суб'єктивність, що теж може призвести до неефективності моделі.

Algorithm (Алгоритм). Зазвичай, одну й ту ж задачу майже завжди можна розв'язати різними методами (способами). Від вибору методу залежить точність, швидкість роботи і розмір готової моделі. В кожній підгалузі МН алгоритми мають свою специфіку, можуть називатись правилами розв'язку в теорії розпізнавання образів або алгоритмами навчання в штучних нейронних мережах.

2. Типи машинного навчання.

Класифікація основних задач розпізнавання образів:

- Image recognition (Розпізнавання зображень).
- Speech recognition (Розпізнавання мовлення).
- Situation recognition (Розпізнавання ситуацій).
- State recognition (Розпізнавання станів).

Всередині цих основних напрямів теорії розпізнавання образів виділяються піднапрями і їх дуже багато, наприклад, в розпізнавання зображень виділяють:

- розпізнавання рентгенограм;
- розпізнавання раку шкіри;
- розпізнавання літер (Optical character recognition (OCR));
- розпізнавання штрих-кодів;
- розпізнавання автомобільних номерів;
- розпізнавання обличь;
- розпізнавання жестів;
- розпізнавання емоцій
- розпізнавання локальних ділянок земної кори

На рис. 2 подана класифікація МН за способом формування даних для алгоритму навчання.

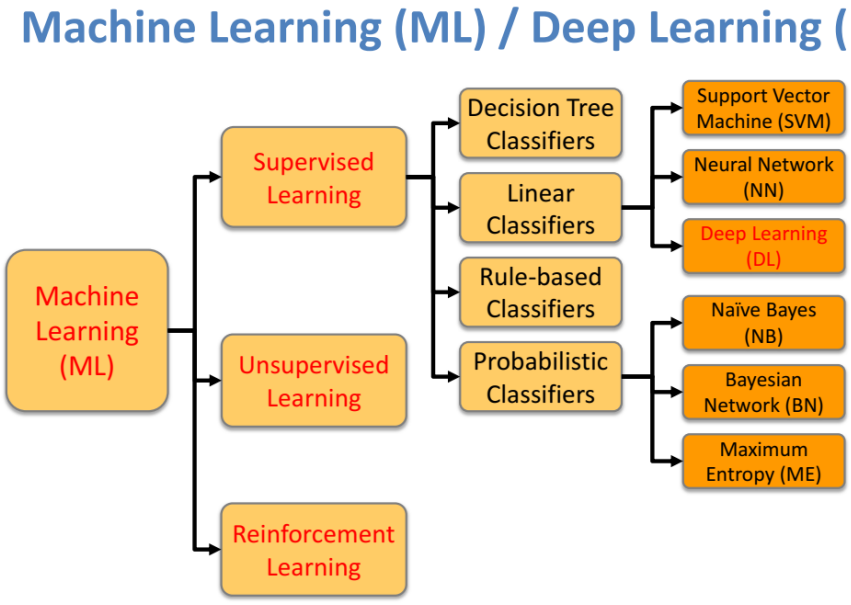


Рисунок 2 – Класифікація машинного навчання за типом даних навчання

1. Supervised Learning (Навчання з учителем)

У цьому випадку алгоритм має «наставника» – учителя, повідомляє йому як правильно вчинити. Вчитель заздалегідь відмічає всі потрібні дані і відповідний цим даним результат, щоб алгоритм навчився на конкретних прикладах. Наприклад, він показує, що на цій світлині автомобіль, на іншій – велосипед.

У термінах машинного навчання вчитель – це саме втручання людини в процес обробки інформації. З учителем машина вчиться набагато краще й швидше, тому для вирішення практичних завдань такі алгоритми використовують частіше.

2. Unsupervised Learning (Навчання без вчителя)

Завдання алгоритму при навчанні без вчителя – знайти зв'язки між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати класифікацію даних. Навчання без вчителя використовується, наприклад, в рекомендаційних системах, коли в інтернет-магазині на основі аналізу попередніх покупок покупцеві пропонуються товари, які можуть зацікавити його з більшою ймовірністю, ніж інші.

У реальності добре розмічені дані – це велика рідкість, тому для їх розмітки зазвичай використовують або спеціальні сервіси, у яких реальні люди з країн з дешевою робочою силою (зазвичай Індії або Китаю) за мінімальну плату вручну класифікують дані, або спеціальні алгоритми для розмітки (які, в свою чергу, можуть також використовувати машинне навчання).

3. Reinforcement Learning (Навчання з підкріпленням)

Таке навчання є окремим випадком навчання з вчителем, але вчителем в даному випадку є «середовище». Алгоритм (в цій ситуації часто називають «агент») не має попередньої інформації про середовище, але має можливість сприймати це середовище через сенсори і здійснювати в ньому деякі дії. Середовище реагує на ці дії і тим самим надає агенту через сенсори агента дані, які дозволяють йому реагувати на них і вчитися. Фактично агент і середовище утворюють систему зі зворотним зв'язком.

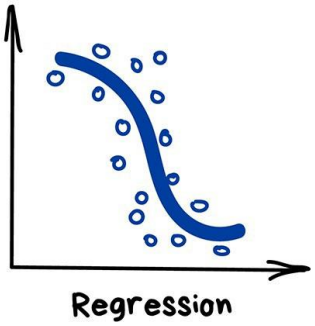
Навчання з підкріпленням використовується для вирішення більш складних завдань, ніж навчання з учителем і без вчителя. Воно використовується, наприклад, в системах навігації для роботів, які навчаються уникати зіткнень з перешкодами шляхом набуття досвіду, отримуючи зворотний зв'язок при кожному зіткненні. Навчання з підкріпленням використовується також в логістиці, при складанні графіків і плануванні завдань, при навчанні машини логічним іграм (покер, нарди, го і ін.).

3. Типи задач машинного навчання:

Регресія

Регресію дуже полюбляють аналітики та фінансисти, тому що вона може показати залежності між різними факторами процесу. Наприклад, як на ціну будинку впливає той чи інший район розташування? Або що більше впливає на вартість авто: рік випуску чи пробіг? Алгоритм намагається побудувати криву на

графіку, яка відображає залежність. Але, на відміну від людини з крейдою та дошкою, робить вона це з застосуванням математики.



Моделі регресійного аналізу зазвичай використовують, щоб показати або передбачити взаємозв'язок між процесом та тим, що цей процес може спровокувати. Тут варто пам'ятати, що така кореляція – не завжди причинність, тобто навіть пряма у простій лінійній регресії, яка добре відображає залежності між даними, може не дати конкретної відповіді про причинно-наслідковий зв'язок. Саме тому регресійний аналіз не використовують для **інтерпретації** причинно-наслідкових зв'язків між змінними.

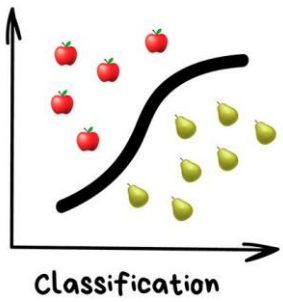
Алгоритмів регресійного аналізу існує багато, і використовують їх дуже давно. Сьогодні регресію використовують для:

- прогнозу вартості цінних паперів;
- аналізу попиту, обсягу продажів;
- медичних діагнозів;
- будь-яких залежностей даних від часу (часові ряди).

Популярні алгоритми: Лінійна, Поліноміальна Регресія, Логістична Регресія.

Класифікація (розпізнавання образів)

Алгоритми класифікації дозволяють розділити об'єкти відповідно до зазначених заздалегідь класів, наприклад розділити котів і собак, музику за жанром, шкарпетки за кольорами. Сьогодні алгоритми класифікації використовують для великої кількості завдань: визначення мови, спам-фільтрів, визначення шахрайства (коли зловмисник сплачує за послуги вкраденими коштами), пошуку схожих документів тощо.



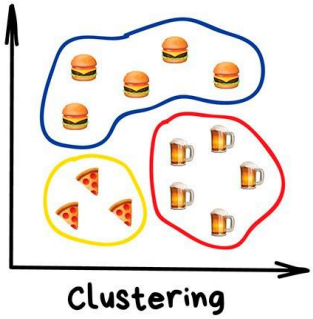
Щоб класифікація спрацювала, потрібно мати розмічені дані з категоріями і ознаками, які алгоритм буде вчитися визначати. Залежно від певних ознак алгоритм визначає, до якого з класів можна віднести об'єкт.

Найпростішим із технічного погляду завданням класифікації є бінарна класифікація – коли об'єкти потрібно розподілити між двома класами. Наприклад, на вході є транспортні засоби, та ми знаємо, що це або автомобілі, або велосипеди. У багатокласовій класифікації кількість класів може досягати декількох тисяч, і рішення стає значно складнішим. Також бувають класи, що перетинаються, – у таких випадках об'єкт може одночасно належати до декількох класів, і нечіткі класи – коли належність до того чи іншого класу визначається ступенем (зазвичай від 0 до 1).

Популярні алгоритми: Наївний Баєс, Дерева Рішень, K-найближчих сусідів, Метод Опорних Векторів.

Кластеризація

Алгоритми кластеризації дозволяють розділити об’єкти у випадку, коли класи заздалегідь не зазначені, а кластери мають бути сформовані за схожістю елементів. Алгоритми кластеризації використовують у завданнях сегментації ринку, для аналізу нових даних, стиснення зображень.



Алгоритм визначає схожі ознаки у об’єктів та групує їх у кластери. Кількість кластерів, як із класами, може бути визначена дослідником або самою машиною. Також дослідник може задати ознаки, за якими потрібно розділити вибірку, або машина визначить їх сама.

Кількість кластерів можна задати вручну або довірити це алгоритму. Теоретично, об’єкти, які перебувають в одному і тому самому кластері, повинні мати схожі властивості і/або особливості, тоді як об’єкти в різних групах повинні мати дуже різні властивості і/або особливості.

Прогнозування

Прогностичних моделей існує багато, і всі вони вирішують завдання передбачення часових рядів – знаходження майбутніх значень залежно від часу. Алгоритм при побудові прогнозу включає в себе наступні елементи:

а) Аналіз часової послідовності на предмет наявності пропущених значень і значень, що випадають, та корекція цих значень.

б) Визначення наявності тренду і його типу. Визначення періодичності в послідовності.

в) Перевірка послідовності на стаціонарність, тобто що процес у майбутньому буде розвиватися так само, як в минулому і сьогоденні. У реальній природі таких процесів не існує, але процеси, характеристики яких змінюються дуже повільно, можна зарахувати до стаціонарних.

ґ) Аналіз послідовності на предмет необхідності попередньої обробки. Усі дані повинні мати однакові характеристики й не відрізнятися один від одного.

д) Визначення параметрів моделі. Прогноз на підставі обраної моделі.

е) Оцінка точності прогнозування моделі.

є) Аналіз похибки обраної моделі.

ж) Визначення адекватності обраної моделі і, якщо результат виявиться незадовільним (недостовірним чи не достатньо точним), її заміна і повернення до попередніх пунктів.

Прогнозування використовують у медичній діагностиці, оцінці кредитоспроможності, передбаченні попиту, під час прийняття рішень на фінансових ринках.

Зменшення розмірності

Зменшення розмірності – це зведення більшого числа ознак до меншого для зручності їх подальшого використання. Використовується для побудови рекомендаційних систем, пошуку схожих документів чи візуалізації. Практична користь методів зменшення розмірності в тому, що можна отримати абстракцію, об’єднавши кілька ознак в одну.

Ще одне застосування завдань зменшення розмірності – рекомендаційні системи, алгоритми, що намагаються передбачити, які об’єкти будуть цікаві користувачеві, маючи певні дані про його профіль. Такі системи не враховують оцінку всіх користувачів, а спираються лише на переваги самого користувача або користувачів зі схожим портретом, і пропонують вам, приміром, подивитися мультфільм, серіал про маніяка або фільм про пошук дзену.

Виявлення аномалій

Мета – виявити аномальні відхилення від стандартних випадків. На перший погляд, завдання дуже схоже на завдання класифікації, проте воно має істотну відмінність: аномалії – явище рідкісне, тому вибірок, на яких можна навчити машинну модель або дуже мало, або немає зовсім. Саме тому методи класифікації тут не працюють. На практиці аномалії допомагають виявити шахрайство в банку, медичні проблеми або помилки в тексті.

Пошук правил

Завдання пошуку правил шукає закономірності в потоці даних. Істотним є передбачення поведінки користувача на інтернет-ресурсах. За яким товаром він повернеться? Які товари можна продати «навздогін» до вже заданого? На які розділи сайту направити користувача, щоб він залишив у вас більше грошей? На ці та інші питання може відповісти машинне навчання, коли алгоритм проаналізує інших користувачів, їхній досвід та поведінку.

4. Розпізнавання образів (Pattern recognition)

Задача розпізнавання образів (РО) – це задача віднесення вхідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних або автоматичне розпізнавання шаблонів і закономірностей у даних. Іноді мова йде про теорію розпізнавання образів – напрям штучного інтелекту, що створює теоретичні основи класифікації та ідентифікації предметів, явищ, процесів, сигналів, ситуацій тощо. Віднесенням об’єкта до того чи іншого класу може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі. Віднесення об’єкта до певного класу відображає найтиповішу проблему класифікації, і, коли говорять про розпізнавання образів, найчастіше мають на увазі саме цю проблему.

Терміни теорії розпізнавання образів

Поняття класу

Класом у теорії розпізнавання образів прийнято називати сукупність об’єктів, які мають ті чи інші спільні ознаки. Про ознаки мова йшла в попередньому розділі. Клас може об’єднувати фізично існуючі сутності (наприклад, людина, яблуко) або бути абстрактним поняттям (горе, економічний крах і т.і.). Ознаки, що дають можливість відрізнити представників одного класу від іншого, прийнято називати інформативними ознаками. Ознаки, спільні для всіх представників класу, називатимемо інваріантами класу.

Іноді вживають більш формальне визначення класу: класом називається сукупність об’єктів, пов’язаних між собою деякими відносинами еквівалентності, або, в крайньому разі, толерантності. Відносинами

еквівалентності називається відношення, яке є симетричним, рефлексивним і транзитивним.

Можна виокремити такі основні властивості класів:

1. Усі представники класу мають певний набір спільних ознак (випливає з визначення).

2. Змінюваність реалізацій класів. По-перше, різні об'єкти, що належать одному класу, можуть бути не схожими між собою. Вони повинні мати спільні інваріантні ознаки, але всі інші ознаки можуть як завгодно варіювати. По-друге, один і той самий об'єкт може змінюватися з часом і навіть поступово переходити від одного класу до іншого (наприклад, перетворення пуголовка на жабу). Усе це свідчить про те, що розпізнати чіткі межі класу часто неможливо.

3. Ознайомлення з деякою скінченою кількістю представників одного класу дає можливість впізнавати інших представників цього класу. Взагалі кажучи, ця властивість може не виконуватися. Але якщо ця властивість виконується (принаймні, теоретично), це є дуже сильним і важливим твердженням. Воно по суті означає можливість навчатися на прикладах, тобто на основі спостереження певної кількості прикладів (можливо, разом з контрприкладами), сформулювати правило розпізнавання, яке дає змогу відрізняти представників даного класу від представників іншого (можливо, з певною достовірністю тобто з певним процентом помилок).

Якщо навчання на прикладах неможливе або неефективне, правило розпізнавання інколи можна задати явно. Якщо це зробити не вдається, єдиною можливістю для надійного розпізнавання залишається запам'ятовування всіх можливих представників даного класу. Цей випадок не становить інтересу з теоретичної точки зору, і його можна реалізувати лише, якщо кількість можливих представників не є надто великою.

Означення 1. Образом (*Pattern*) називається модель, яка відображає властивості об'єкта, що розпізнається. Образ характеризується множиною ознак розпізнавання, які утворюють структурований вектор-реалізацію образу. Математично образ розглядається як точка в N-вимірному просторі, де N – кількість ознак образу.

Означення 2. Ознакою (*Feature*) розпізнавання називається характеристика певної властивості об'єкта.

Означення 3. Вектором-реалізацією образу називається структурована послідовність ознак розпізнавання, що подається у вигляді вектора-рядка.

$x=(x_1, x_2, \ldots, x_N)$

або вектора-колонки

$x=(x_1, x_2, \ldots, x_N)^T$

Означення 4. Правилом розв'язку (classifier) називається математичний вираз або алгоритм визначення належності реалізації образу одному з класів розпізнавання із заданого алфавіту.

Означення 5. Міра відстані (метрика, міра близькості) відстань між двома об'єктами в N -вимірному просторі ознак. В алгоритмах розпізнавання найчастіше використовуються такі міри:

Евклідова $d(x_m, x_l) = \sqrt{\sum_{1=1}^{N} (x_m^i - x_l^i)^2}$

Манхеттенська $d(x_m, x_l) = \sum_{i=1}^{N} \left| x_m^i - x_l^i \right|$

Узагальнена $d(x_m, x_l) = \sqrt[r]{\sum_{i=1}^{N} \left(x_m^i - x_l^i \right)^r}$

Означення 6. Помилка 1-го роду – це віднесення образу не до того класу, до якого він належить в дійсності.

В задачі діагностики має назву «Помилкова тривога».

Приклад: авторизований *користувач* класифікується як *порушник*.

Означення 7. Помилка 2-го роду – це віднесення образу до якогось певного класу, до якого він насправді не належить.

В задачі діагностики має назву «Пропуск цілі».

Приклад: *порушник* класифікується як авторизований *користувач*.

З наведених означень випливає, що в просторі ознак клас є множиною точок. Якщо в просторі ознак ми маємо два класи, то правилом розв'язку буде гіперплощина або інша поверхня, що розділяє ці класи. Тепер, коли ми хочемо визначити до якого класу належить невідомий об'єкт необхідно визначити по який бік поверхні знаходиться цей об'єкт.

Роздільність класів розпізнавання

Гіпотеза компактності

В задачах класифікації це – припущення про те, що образи схожих об'єктів набагато частіше знаходяться в одному класі, ніж в різних; або – класи утворюють компактно локалізовані підмножини в просторі образів об'єктів (рис.3). Це також означає, що межа між класами має досить просту форму.

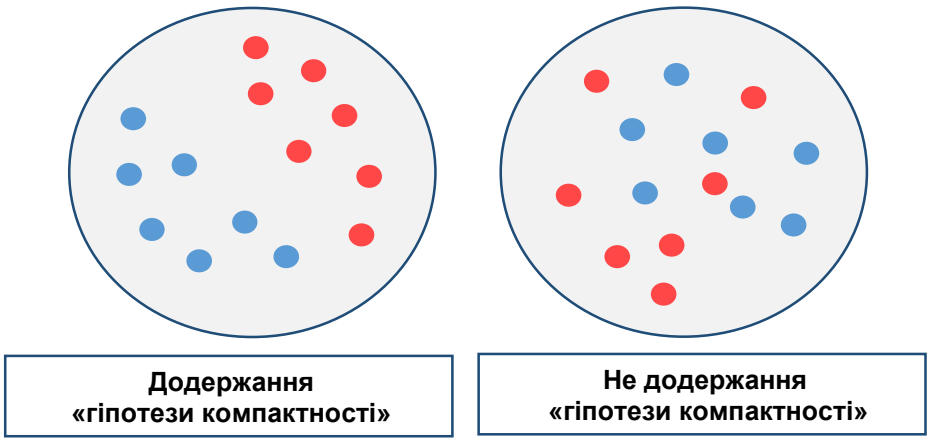


Рисунок 3 – Ілюстрація гіпотези компактності

5. Постановка задачі розпізнавання.

Загальна постановка задачі розпізнавання

Знайти правило розв'язку, що відносить образ об'єкту або процесу у виді істотних ознак, що характеризують цей об'єкт або процес до певного класу.

Математична постановка задачі розпізнавання

Нехай Ω – простір образів;

$\omega \in \Omega$ – образ;

$M = \{1, 2, ..., m\}$ – номери класів Ω_1 , Ω_2 , Ω_m, таких що $\Omega_i \cap \Omega_j = \varnothing$, якщо $j \neq i$ та $\bigcup_{i=1}^{m} \Omega_i = \Omega$;

$p: \Omega \rightarrow M$ – правило розв'язку, що є невідомим;

X – простір ознак, тобто векторний простір, точками якого є вектори ознак образів;

$x: \Omega \rightarrow X$ – функція, що ставить у відповідність образу ω його вектор ознак $x(\omega)$.

Задача розпізнавання з учителем (supervised classification) полягає у тому, щоб на підставі множини прецедентів (p_j, x_j), $j = 1, \ldots, N$, яка називається навчальною вибіркою (training sample) побудувати правило розв'язку p, що мінімізує кількість помилок.

Алгоритм розпізнавання представляється як абстрактна функціональна система R з трьох компонент:

$R = \{\Omega, X, P\}$,

де

$\Omega = \{\Omega_k\}$, $k = 1, \ldots, K$ – алфавіт класів – множина категорій за якими розподіляються образи,

$X = \{x_j\}$, $j = 1, \ldots, N$ – словник ознак – множина характеристик, з яких складається опис образу,

$P = \{p_i\}$, $i = 1,\ldots,L$ – множина правил розв'язку (прийняття рішень).

Функціонування системи розпізнавання зводиться до наступного:

- на вхід подається образ – деяка конфігурація з елементів множини X;
- до X застосовується певна послідовність правил з P;
- в результаті конфігурація позначається індексом, що відповідає одному з елементів множини Ω.

Якість функціонування системи визначається тим, наскільки часто присвоєний образу індекс збігається з очікуваним результатом.

Найчастіше система розпізнавання образів за аналогією з біонічною системою функціонує в двох режимах:

- навчання;
- класифікації (інтерпретації).

Процес створення системи розпізнавання образів

Для розпізнавання образів необхідно:

- визначити множину ознак, які відображають істотні властивості об'єктів, що будуть розпізнаватись;

• визначити класи об’єктів, до яких треба відносити образи при розпізнаванні;

• на множині відомих образів визначити правило розв’язку, що відносить реалізацію образу до відповідного класу (навчити систему розпізнавання).

Процес створення системи розпізнавання образів складається з таких кроків (рис.4):

• Вибір ознак (feature selection) – визначення найбільш інформативних ознак образів, які відображають істотні властивості об’єктів (в цей набір можуть входити функції від ознак).

• Генерування ознак (feature generation) – вимірювання або обчислення числових ознак, що характеризують об’єкт.

• Побудова класифікатора (classifier construction) – конструювання правила розв’язку, на підставі якого здійснюється класифікація.

• Оцінка якості класифікації (classifier estimation) – обчислення показників правильності класифікації (точність, чутливість, специфічність, помилки першого та другого роду).

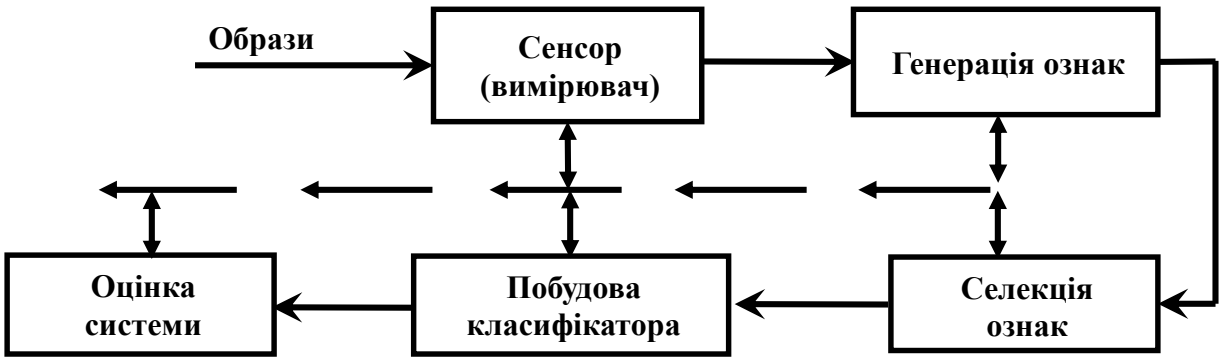


Рисунок 4 – Процес створення системи розпізнавання образів

6. Методи теорії розпізнавання.

Класифікація методів розпізнавання за математичною моделлю:

1) Алгебраїчний

Основа – прості правила розв’язку у виді нерівностей, що зв’язують ознаки. Недолік – невисока достовірність розпізнавання, оскільки не враховує неконтрольовані фактори.

2) Геометричний

Основа – образ представляється точкою в n-вимірнім просторі ознак. Класи в n-вимірнім просторі розділяються гіперплощинами. Належність до класу визначається відстанню образу до центру класу або до гіперплощини. Характеризується універсальністю, наочністю та простотою інтерпретації алгоритмів розпізнавання.

3) Статистичний

Базується на статистичних характеристиках даних. Використовується формула Байеєса.

4) Біологічний

Штучні нейронні мережі. Алгоритми в рамках цього підходу моделюють когнітивні процеси, що відбуваються у нервових клітинах мозку. Недолік – висока чутливість до багатовимірності простору ознак розпізнавання.

5) Мережевий

Семантичні мережі, фрейми, мережі Петрі, дерево рішень тощо.

Перевага – простота моделі, можливість її розширення та ускладнення.

Недолік – складність побудови правил розв'язку.

6) Нечіткий

Дозволяє моделювати процеси розпізнавання образів, що апріорно перетинаються в просторі ознак розпізнавання. Але не пристосований до оптимізації параметрів функціонування системи розпізнавання;

Класифікація методів розпізнавання за типом даних

Методи порівняння з еталоном

Застосовуються у випадках, коли кожному класу Ω_k можна зіставити кінцевий набір еталонних образів

$E_k = \{e_m, m = 1, ..., M_k\}$.

Процес розпізнавання полягає в зіставленні образів, що надходять на вхід пристрою розпізнавання або алгоритму, з еталонами E_k класів Ω_k, на основі обраної міри схожості (близькості).

Методи загальних властивостей

Застосовується в тих випадках, коли множина образів кожного класу занадто велика, щоб отримати надійний опис кінцевого числа еталонів, але можна виявити достатню кількість відмінних рис класів за кінцевими вибірками образів. Виявлені властивості кодуються на основі відповідної моделі і зберігаються в пам'яті у вигляді деяких структур, функцій або відносин.

Структурний метод

При структурних методах, реалізації образу описуються не множиною їх значень (навчальна вибірка типу об'єкт-властивість) або їх відношень (навчальна вибірка типу відношення-подібність), а їх структурою.

Структурні методи використовуються при сегментації (визначенні меж) різних текстів, при усномовному розпізнаванні за послідовністю фонем, тощо.

Головна перевага:

можливість подати велику кількість реалізацій у вигляді малопотужної множини непохідних елементів і граматичних правил.

Недоліки:

- відсутність прямих вирішальних правил;
- обмеженість використання через те, що аналізується не весь образ, а лише його фрагмент, що ускладнює процес прийняття рішень.

Метод січних площин

Алгоритм навчання, заснований на методі січних площин, полягає у відокремленні образів за допомогою частин гіперплощин. Алгоритм складається з наступних етапів.

1. Навчання (формування гіперплощин для відокремлення множин):

a. проведення січних площин;
b. вилучення зайвих гіперплощин;
c. вилучення зайвих частин площини.

2. Розпізнавання нових об’єктів.

Припустимо, що потрібно навчити комп’ютер розпізнавати 3 образи *a*, *b*, та *c*.

Проведення січних площин. У комп’ютер вводять коди двох точок (рис. 5), які належать різним образам та проводять довільну пряму, яка їх відокремлює.

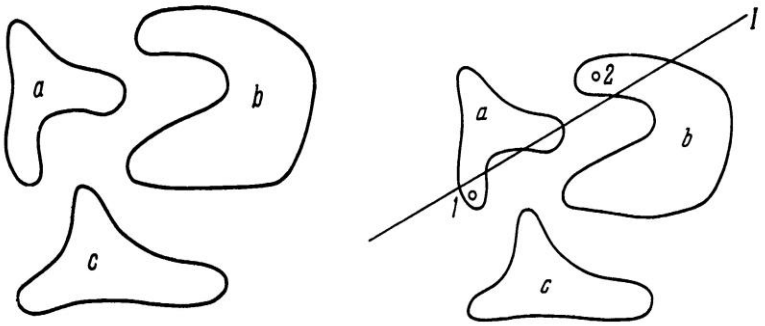


Рисунок 5 – Проведення січної площини

Далі береться третій об’єкт (рис. 6) і перевіряється правильність класифікації відносно проведеної прямої.

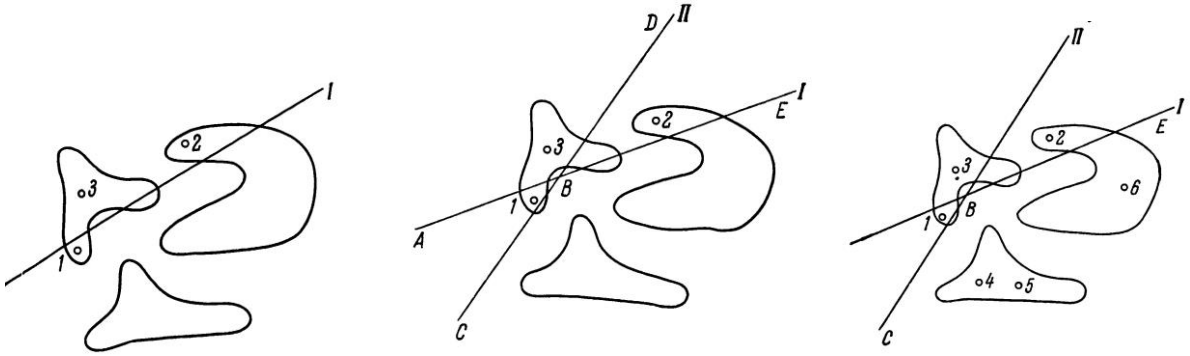


Рисунок 6 – Проведення другої січної площини

Оскільки третя точка належить тій самій півплощині, що й друга, то приходимо до висновку, що поточний класифікатор працює невірно. Для відокремлення точок 2 та 3 проводиться пряма II. Після цього площина розбивається на 4 частини. Ті частини, у який лежать об’єкти 1 та 3 (області *ABC* та *ABD*) відносимо до класу *a*, область *DBE* — до класу *b*.

При перевірці нової точки можливі три випадки:

1) виникає протиріччя (як це було для точки 3);
2) протиріччя не виникає, точка потрапляє у свою частину простору;
3) протиріччя не виникає, точка потрапляє у вільну (непозначену) частину простору (точки 4 і 5); У третьому випадку (точки 4 та 5 на рис. 6) машина відносить вільну частину площини до відповідного образу.

Для відокремлення точки 6 (образ *b*) від точок 4 та 5 (образ *c*) потрібно провести дві нові прямі (прямі III та IV на рис. 7).

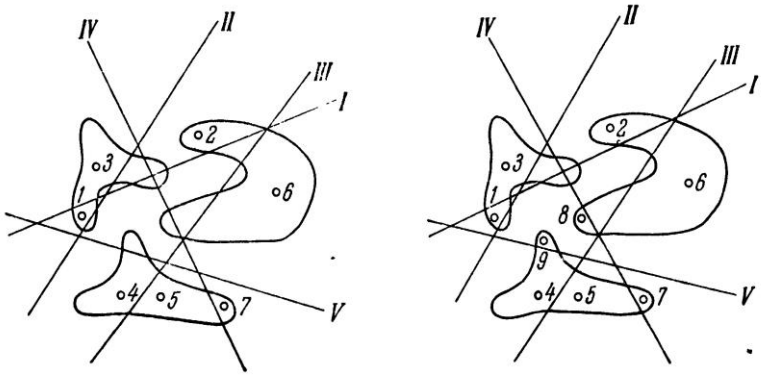


Рисунок 7 – Проведення нових січних площин

Подальші кроки вищенаведеного процесу навчання наведені на рис. 8, де вказано проміжкові та кінцевий результат стадії а). Незафарбовані ділянки не відносяться розпізнавальною системою до жодного із трьох наведених образів

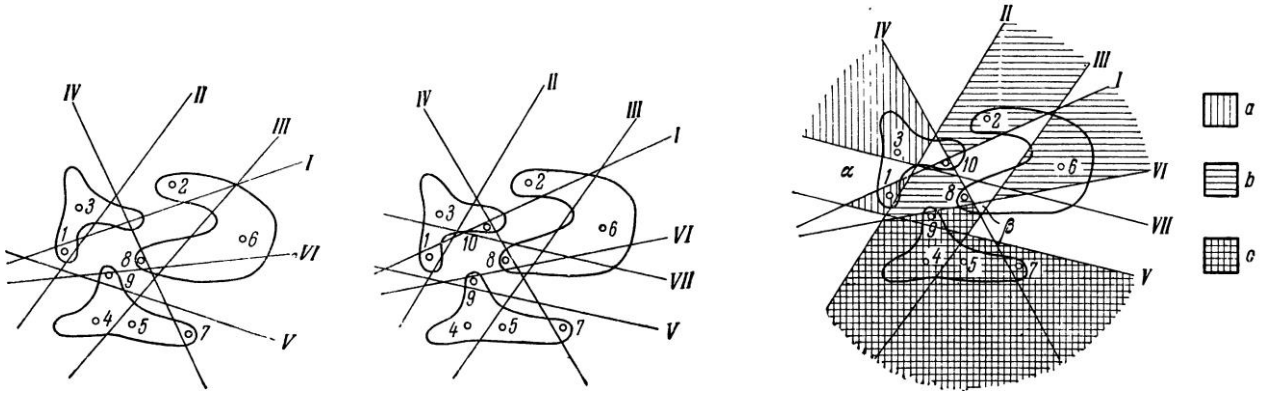


Рисунок 8 – Результат проведення площин

Видалення зайвих площин. З рис. 8 видно, що гіперплощини (прямі) I, III та V можуть бути вилученні цілком, оскільки вони не містять частин, відкидання яких призводить до появи протиріч. Наприклад, пряма I використовується лише для відокремлення об’єктів 1 та 2 — представників різних образів *a* та *b*. Але ці об’єкти відокремлює також пряма IV (або пряма VI). Розбиття після відкидання зайвих площин наведено на рис. 9. Слід зазначити, що у результаті частка вільних частин площини зменшилися.

Проте не можна вважати, що процес навчання завершений, оскільки ще залишилися "порожні" частини площини. Для їх вилучення використовується гіпотеза компактності. Природнім є віднести "порожні" області до того образу, що і яка-небудь суміжна зафарбована частина. Цей процес називають процесом вилучення зайвих частин. Для деяких вільних частин площини цей процес однозначний (область α на рис. 8), для інших — ні (усі області на рис. 9). Рекомендується проводити це вилучення поступово проглядаючи усі гіперплощини (прямі на рис. 9) та "порожні шматки", які до них прилягають.

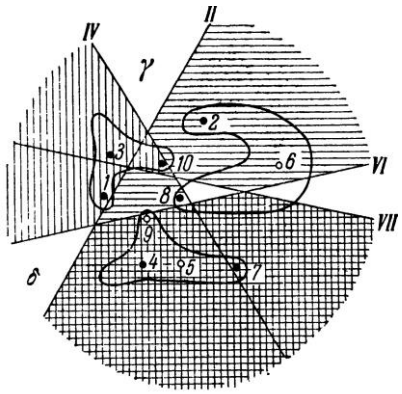


Рисунок 9 – Вилучення зайвих частин

Тому спочатку виконуємо перевірку усіх частин площини II, потім усіх частин площини IV і т.д. Кінцевий результат наведений на рис. 10.

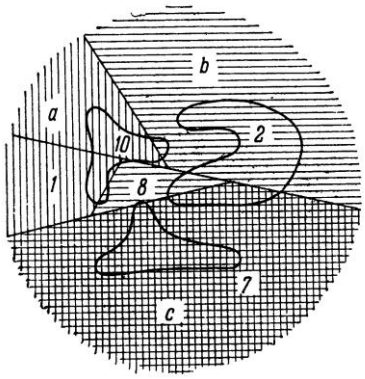


Рисунок 10 – Кінцевий результат злиття

Метод потенціалів для розпізнавання образів

Точковий електричний заряд у однорідному середовищі створює електричне поле, зображене на рис. 11. Радіальні лінії — це силові лінії поля, концентричні кола – лінії однакового потенціалу.

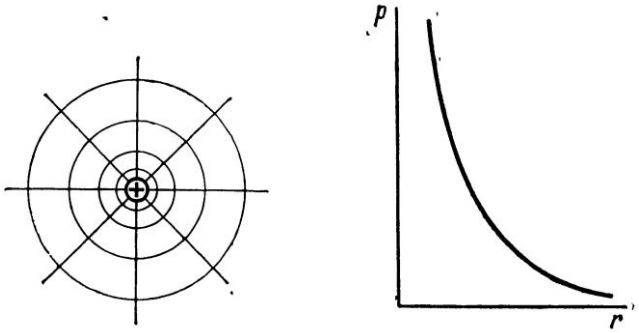


Рисунок 11 – Електричне поле точкового заряду

Потенціал p у кожній точці простору визначається співвідношенням

$$p = a\frac{q}{r^2},$$

де a – деяка стала, q – величина заряду, r – відстань від заданої точки до заряду.

Крива зміни потенціалу як функції відстані наведена на рис. 11. Потенціал зменшується по мірі віддалення від його джерела. Якщо поле утворене кількома зарядами, то потенціал в кожній точці рівний сумі потенціалів, які створюються кожним із зарядів.

Припустимо, що у просторі розташовані дві компактні групи зарядів. У одній групі — позитивні, у другій — негативні. На рис. 12 показаний розподіл потенціалів у околі цих зарядів, на рис. 13 ці потенціали алгебраїчно просумовані.

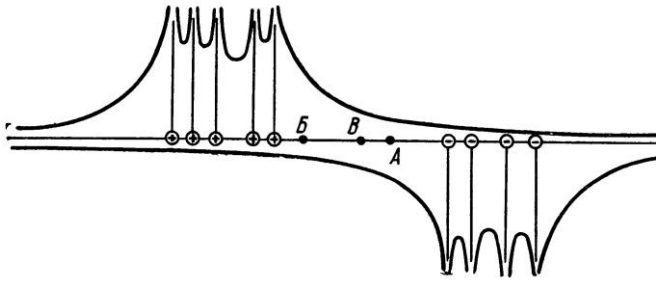


Рисунок 12 – Розподіл потенціалів кількох зарядів

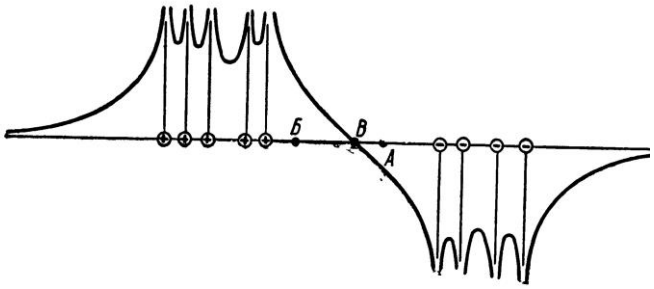


Рисунок 13 – Результат підсумовування потенціалів

Точку можна віднести до тієї чи іншої множини точок у залежності від того, який знак має сумарний потенціал поля у цій точці.

Вищенаведені міркування по аналогії можна перенести на точки простору рецепторів. Кожній точці, яка з'являється у процесі навчання, поставимо у відповідність деяку функцію, аналогічну по формі до електричного потенціалу. Такою функцією може бути, наприклад, функція

$$\varphi(D) = \frac{1}{1+\alpha D^2},$$

де α – деякий коефіцієнт, D – відстань (у деякій метриці) між точкою-джерелом потенціалу та точкою, у якій обчислюється потенціал. Наприклад, у якості D можна використовувати евклідову відстань, віддаль Хеммінга, манхеттенську метрику, тощо.

Нехай джерелами потенціалів є група точок, які відповідають деякому образу a. Тоді можна вважати, що середній потенціал, які створюють у деякій точці простору джерела цього образу, характеризує віддаль від цієї точки до усього образу у цілому.

Спираючись на гіпотезу компактності, можна запропонувати наступне правило розпізнавання: точку відносимо до того образу, середній потенціал якого у цій точці є максимальним.

Алгоритм розпізнавання на основі методу потенціалів

1. У процесі навчання запам'ятовуються координати усіх точок та належність їх до відповідних образів $a_1,....,a_m$.

2. Розпізнавання.

a. Для точки **x**, яка підлягає розпізнаванню, обчислюється потенціали кожного образу, тобто суми

$$\Phi_i(x) = \frac{1}{n_i} \sum_{a \subset \alpha_i} \varphi_a(x), \quad i = 1, 2, ..., m,$$

де m – кількість різних образів, n_i – кількість точок відповідного образу, використаних у процесі навчання, $\varphi_a(x) = \frac{1}{1 + \alpha D^2(a, x)}$ – потенціал, який створює точка **a** у точці **x**.

b. Порівнюються величини $\Phi_1(x), \Phi_2(x), \ldots, \Phi_m(x)$, і точка **x** відноситься до того образу, потенціал якого є найбільшим.

У випадку двох образів a_1 та a_2 рішення можна приймати на основі значення знаку функції

$$\Delta\Phi(x) = \Phi_{a_1}(x) - \Phi_{a2}(x)$$

Застосування вищенаведеного алгоритму для розпізнавання цифр, зображених за допомогою рецепторного поля 6×10 у середньому дало змогу досягнути правильного розпізнавання у 85% випадків за умови, що $n_1 = n_2 = ... = n_{10} = 12$, причому подальше збільшення обсягу навчальної вибірки практично не впливає на якість розпізнавання.

Покращена модифікація алгоритму навчання. Алгоритм розпізнавання може бути покращений, шляхом введення поняття «ваги точки» та доповненням кроку навчання наступною операцією. Після запам'ятовування усіх об'єктів їм присвоюють початкову вагу 1. Далі до елементів навчальної вибірки застосовують крок 2 алгоритму навчання, причому потенціали середні потенціали обчислюють за формулою

$$\Phi_i(x) = \frac{1}{n_i} \sum_{a \subset \alpha_i} \varphi_a(x) w(a), \quad i = 1, 2, ..., m,$$

де w(**a**) — вага точки **a**.

Якщо при розпізнаванні об'єкту **a** відбувається помилка, то вагу w(**a**) відповідного об'єкту збільшують на деяку величину, наприклад на одиницю. Потім застосовується ще один такий самий цикл перевірки та корекції ваг і т.д. Цикли повторюються до тих пір, поки усі відомі фігури не будуть розпізнані правильно.

Застосування покращеного алгоритму до розпізнавання дало змогу підвищити відсоток правильного розпізнавання нових зображень цифр до 89%.

Метод найближчого сусіда.

1. Випадок одного еталону.

В деяких задачах об'єкти деяких класів (образів) мають *тенденцію до групування навколо деякого об'єкту, який є типовим* або репрезентативним для відповідного образу. Типовим прикладом є задача зчитування банківських чеків, для якої вектори, які відповідають образам кожного класу, будуть майже ідентичними.

Розглянемо M класів, які допускають зображення за допомогою еталонних представників $\mathbf{z}_1,\ldots, \mathbf{z}_M$. Евклідова відстань між довільним вектором $\mathbf{x}$ та цими образами обчислюється за формулою

$$D_i = \sqrt{\sum_{j=1}^{n}(x_j - z_{ij})^2}$$

або $D_i = \|x - z_i\| = \sqrt{(x-z_i)'(x-z_i)}$.

Вектор $\mathbf{x}$ відноситься до класу ω_i, якщо умова $D_i < D_j$ виконується для усіх $j \neq i$. Перетворимо формулу обчислення відстані

$$D_i^2 = \|x - z_i\|^2 = (x-z_i)'(x-z_i) = x'x - 2x'z_i + z_i'z_i = x'x - 2\left(x'z_i - \frac{1}{2}z_i'z_i\right).$$

Остання формула показує, що вибір мінімальної відстані до класу еквівалентний максимізації величини

$$x'x - 2\left(x'z_i - \frac{1}{2}z_i'z_i\right).$$

Тому функції рішень можна визначити як

$$d_i = x'z_i - \frac{1}{2}z_i'z_i, \quad i = 1, 2, ..., M.$$

де $d_i(\mathbf{x})$ – лінійні функції рішень. Якщо покласти

$$w_{ij} = z_{ij}, \quad j = 1, 2, \ldots, n,$$
$$w_{i,n+1} = -(1/2)\,\mathbf{z}_i'\mathbf{z}_i$$

і $\mathbf{x} = (x_1, x_n, 1)$, то функції рішень можна записати у виді:

$$d_i(x) = w'x, \quad i = 1, 2, ..., M$$

де $\mathbf{w}_i\,(w_{i1},\ldots, w_{in}, w_{i,n+1})$. На рис. 14 зображена відокремлююча межа для випадку двох класів.

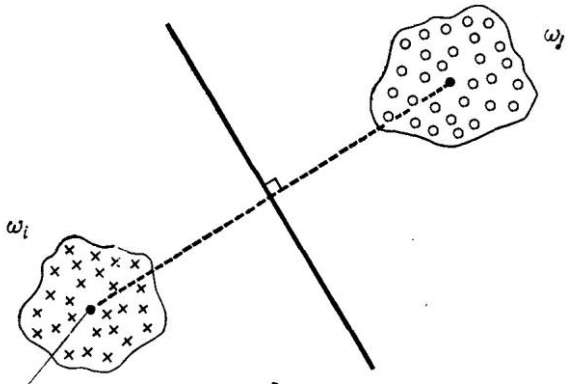


Рисунок 14 – Розділення двох класів

Межа між класами є гіперплощиною, яка рівновіддалена від еталонних представників класів. Описаний підхід ще називають кореляцією або співставленням із кластером.

2. Випадок кількох еталонів.

У цьому випадку кожний клас ω_i можна охарактеризувати еталонами $\mathbf{z}^1{}_i,,\ldots,\ \mathbf{z}_i^{Ni}$, N_i – кількість еталонів i-го класу. Одним із найпростіших підходів є метод "найближчого сусіда" (НС-правило). Відстань між i-им класом та об'єктом $\mathbf{x}$ можна обчислити за формулою:

$$D_i = \min_i \|x - z_i\|, \quad i = 1, 2, ..., N_i.$$

тобто з використанням відстані до найближчого до $\mathbf{x}$ еталона класу. Функції рішень можна записати у виді:

$$d_i = \max_i \{x' z_i^l - \frac{1}{2}(z_i^l)' z_i^l\}, \quad l = 1, 2, ..., N_i.$$

і процедура віднесення нового об'єкта до класу така сама, як і раніше. На рис. 14 показані межі класів у випадку двох класів, які містять по два еталони кожний.

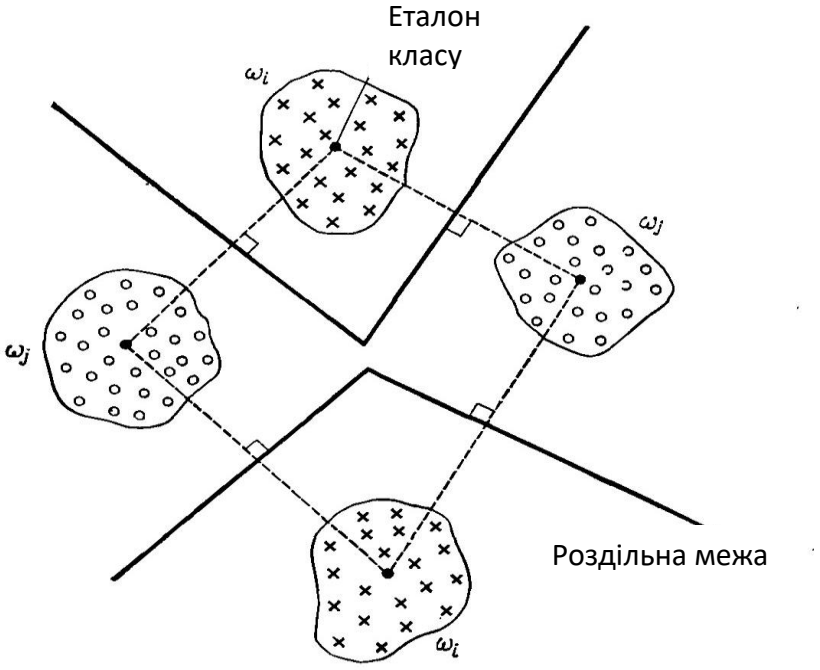


Рисунок 14 – Розділення двох класів з двома еталонами

Отриманий класифікатор є частинним випадком *кусково-лінійного класифікатора*.

3. Узагальнення принципів класифікації по мінімуму відстані. Метод *k*-найближчих сусідів.

НС-правило може бути узагальнено на *k-НС правило*, згідно якого обчислюються *k*-найближчих сусідів і об'єкт відноситься до того класу, до якого відноситься найбільша кількість з *k*-найближчих еталонів.

7. Класифікація на основі байєсівської теорії рішень

Байєсовський підхід виходить із статистичної природи спостережень. За основу береться припущення про існування ймовірнісного заходу на просторі образів, який або відомий, або може бути оцінений. Мета полягає у розробці такого класифікатора, який правильно визначатиме найбільш ймовірний клас для пробного образу. Тоді завдання полягає у визначенні "найвірогіднішого" класу.

Задано M класів $\Omega_1, \Omega_2, ..., \Omega_M$, і навіть $P(\Omega_i|x)$, i = 1, 2, ..., M – ймовірність те, що невідомий образ, представлений вектором ознак x, належить класу Ω_i .

$P(\Omega i|x)$ називається апостеріорною ймовірністю, оскільки задає розподіл ознаки класу після експерименту (a posteriori – тобто після того, як значення вектора ознак x було отримано).

Розглянемо випадок двох класів Ω_1 та Ω_2. Природно вибрати вирішальне правило в такий спосіб: об'єкт відносимо до того класу, у якого апостеріорна ймовірність вище. Таке правило класифікації максимально апостеріорної ймовірності називається Байєсовським: якщо $P(\Omega_1|x) > P(\Omega_2|x)$, то x класифікується в Ω_1, інакше в Ω_2 . Таким чином, для Байєсовського вирішального правила необхідно отримати апостеріорні ймовірності $P(\Omega_i|x)$, i = 1, 2. Це можна зробити за допомогою формули Байєса.

Формула Байєса, отримана Т. Байєсом в 1763 році, дозволяє обчислити апостеріорні ймовірності подій через апріорні ймовірності та функції правдоподібності.

Нехай $A_1, A_2, ..., A_n$ – а повна група несумісних подій.

$$\bigcup A_i = \Omega \; . \; A_i \cap A_j = \emptyset,$$

при $i \neq j$. Тоді апостеріорна ймовірність має вид:

$$P(A_i \mid B) = \frac{P(A_i)P(B \mid A_i)}{\sum_{i=1}^{n} P(A_i)P(B \mid A_i)},$$

де $P(A_i)$ – апріорна ймовірність події A_i , $P(B|A_i)$ – умовна ймовірність події B за умови, що сталася подія A_i .

Розглянемо отримання апостеріорної ймовірності $P(\Omega|x)$, знаючи $P(\Omega)$ та $P(x|\Omega)$.

Для незалежних подій A і B можна записати:

$$P(AB) = P(A|B)P(B), \qquad P(AB) = P(B|A)P(A),$$

звідси:

$$P(A|B)P(B) = P(B|A)P(A),$$

або

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)},$$

Якщо P(A) і P(A|B) описуються щільностями p(x) і p(x|B), то

$$P(B \mid x) = \frac{p(x \mid B)P(B)}{p(x)} \Rightarrow P(\Omega_i \mid x) = \frac{p(x \mid B)P(\Omega_i)}{p(x)}$$

При перевірці класифікації порівняння $P(\Omega_1|x)$ та $P(\Omega_2|x)$ еквівалентно порівнянню $p(x|\Omega_1)P(\Omega_1)$ та $p(x|\Omega_2)P(\Omega_2)$. У випадку, коли $P(\Omega_1|x) = P(\Omega_2|x)$ вважається, що міра множини x дорівнює нулю.

Таким чином, завдання порівняння за апостеріорною ймовірністю зводиться до обчислення величин $P(\Omega_1)$, $P(\Omega_2)$, $p(x|\Omega 1)$, $p(x|\Omega_2)$. Вважатимемо, що в нас достатньо даних для визначення ймовірності приналежності об'єкта кожному з класів $P(\Omega_i)$, i = 1, 2. Такі ймовірності називаються апріорними ймовірностями класів. А також вважатимемо, що відомі функції розподілу вектора ознак для кожного класу $P(x|\Omega i)$, i = 1, 2.

Вони називаються функціями правдоподібності x стосовно Ω_i. Якщо апріорні ймовірності та функції правдоподібності невідомі, їх можна оцінити методами Якщо апріорні ймовірності та функції правдоподібності невідомі, їх можна оцінити методами математичної статистики на множині прецедентів. Наприклад,

$$P(\Omega_i) \approx \frac{N_i}{N},$$

де N_i число прецедентів з Ω_i, i = 1,2. N – загальна кількість прецедентів. $P(x|\Omega_i)$ може бути наближено до гістограми розподілу вектора ознак для прецедентів з класу Ω_i.

У теорії статистичних рішень всі види вирішальних правил для $M \geq 2$ класів засновані на формуванні відношення правдоподібності L і його порівняння з певним порогом c, значення якого визначається обраним критерієм якості:

$$L = \frac{f_n(x_1,\ldots,x_n \mid \Omega_1)}{f_n(x_1,\ldots,x_n \mid \Omega_2)} \geq c$$

де $f_n(x_1,\ldots,x_n \mid \Omega_1)$ – умовна n-мірна щільність ймовірності вибіркових значень $x_1\ldots,x_n$ при умові їх належності до класу Ω_i. В статистичному розпізнаванні ці щільності, в принципі, не відомі, і в формулу) підставляються їх оцінки $\hat{f}_n(x_1,\ldots,x_n \mid \Omega_1)$.

Таким чином, в правилі розв'язку з порогом c порівнюється оцінка відношення правдоподібності

Отже, Байєсовський підхід до статистичних завдань ґрунтується на припущенні про існування певного розподілу ймовірностей для кожної ознаки. Недоліком цього є необхідність постулювання як існування апріорного розподілу для невідомого параметра, і знання його форми.

Мінімізація середнього ризику

Розглянемо правила прийняття рішень у задачі статистичної класифікації на прикладі двох класів та однієї ознаки, заданої дійсним значенням вимірювання X. Завдання полягає у визначенні на шкалі X інтервалів Ω_1 та Ω_2, на яких прийматимуться рішення на користь першого та другого класу

відповідно. Для простоти надалі класи та відповідні їм області прийняття рішення позначатимемо одним і тим самим символом Ω.

Припустимо, що отримана вся необхідна статистична інформація про класи: функції щільності статистичного розподілу $f_1(x)$ і $f_2(x)$ (рис. 15), а також апріорні ймовірності $P(\Omega_1)$ та $P(\Omega_2)$ появи даних класів.

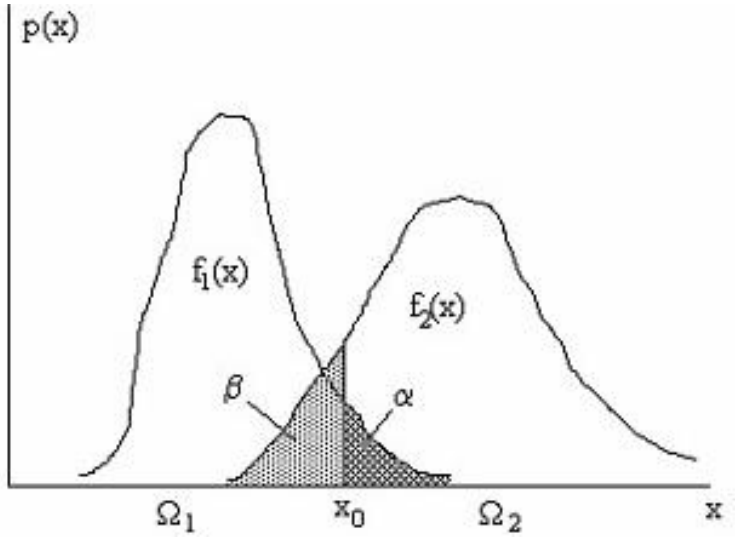


Рисунок 15 – Статистична гіпотеза для двох класів

Нехай ми вибрали деяку точку x_0, що розділяє всю шкалу значень ознаки x на два інтервали: $(-\infty, x_0]$ відповідає області рішень на користь класу Ω_1, (x_0, ∞) – області рішень на користь класу Ω_2. несуперечна безліч припущень про властивості випадкової величини ξ називається статистичною гіпотезою, в даному випадку ми розглядаємо дві статистичні гіпотези: ξ, що приймає значення x, має розподіл із щільністю $f_1(x)$, тобто належить класу Ω_1, – це гіпотеза H_1 проти альтернативи H_2: ξ має розподіл із щільністю $f_2(x)$, тобто належить класу Ω_2.

Імовірність появи будь-якого значення x відрізняється від нуля і для першого, і для другого класу на всій множині X, тому при прийнятті рішення щодо належності деякого значення x до одного з класів можуть виникнути 4 ситуації.

1. Приймаємо гіпотезу H1 і вона вірна.
2. Приймаємо H2, але правильна H1.
3. Приймаємо H2 і вона вірна.
4. Приймаємо H1, але правильна H2.

Імовірність виникнення ситуації 1 відповідає площі під $f_1(x)$ на напівінтервалі $(-\infty, x_0]$, ситуації 2 – площі під $f_1(x)$ на інтервалі (x_0, ∞), ситуації 3 – площі під $f_2(x)$ на (x_0, ∞), ситуації 4 – площі під $f_2(x)$ на $(-\infty, x_0)$ Сумарна площа під $f_1(x)$ та $f_2(x)$ для ситуацій 2 і 4 – це повна ймовірність помилок у нашій схемі прийняття рішень. У випадку двох альтернативних гіпотез помилку, що відповідає ситуації 2, зазвичай називають **помилкою першого роду** (α), помилку, що відповідає ситуації 4, – **помилкою другого роду** (β).Взагалі кажучи, поняття помилок першого та другого роду симетрично і залежить від

того, яка гіпотеза є основною, а яка – альтернативною. Якби H_2 була основною гіпотезою, помилка першого роду відповідала б ситуації 4.

Байєсівська стратегія мінімального середнього ризику.

Щоб побудувати класифікатор на основі описаної схеми, ми повинні виробити правило оптимального розбиття всієї множини X на ділянці прийняття рішення на користь кожного класу. У нашому конкретному випадку – правило вибору точки x_0, що розділяє. Оскільки відмінність між класами пов'язана з частотою появи тих чи інших значень x цих класів, природно вибрати точку x_0 в такий спосіб, щоб за багаторазовому пред'явленні класифікатору образів x усереднена за сукупністю рішень помилка була мінімальною.

Для цього запровадимо "вартість" кожного прийнятого рішення. Для описаних вище чотирьох ситуацій із класами Ω_1 та Ω_2 запишемо відповідні вартісні коефіцієнти у виді матриці:

$$c = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

Діагональні елементи такої вартісної матриці відповідають випадкам правильної класифікації (ситуації 1 та 3), два інших – помилкам (ситуації 2 та 4). Вартісні коефіцієнти можна поставити так, щоб вони відображали наші переваги щодо тієї чи іншої ситуації. Найчастіше обмежуються умовою
$c_{11} = c_{22} = 0$ і $c_{12} = c_{21} = 1$, тобто розглядають лише втрати від помилок.

Розглядаючи вартісні коефіцієнти як наш "ризик" у кожній із можливих ситуацій (які можна розглядати як випадкові події), введемо поняття середнього ризику для описаних чотирьох випадків:

$$R = \sum_{k=1}^{K} P(\Omega_k) \sum_{l=1}^{K} c_{kl} P(c_{kl})$$

Тут c_{kl} – це платіжні коефіцієнти з (8.1), а $P(c_{kl})$ – ймовірність кожної з чотирьох подій (ймовірність відповідної плати подію). У разі коли $c_{11}=c_{22}=0$ і $c_{12}=c_{21}=1$ (тобто «плата» – це штраф за помилки), функцію R називають також функцією втрат.

Розглянемо, що є ймовірності $P(c_{kl})$. Наприклад, ймовірність плати c_{11} є ймовірність одночасного здійснення двох подій: ξ належить до $f_1(x)$ (ймовірність цієї події у Ω_1 є $\int_{-\infty}^{x_0} f_1(x)dx +$ та появи самого класу Ω_1 (апріорна ймовірність цієї події – $P(\Omega_1)$.

Отже, формулу ризику можна записати так:

$$R = c_{11}P(\Omega_1)\int_{-\infty}^{x_0} f_1(x)dx + c_{12}P(\Omega_1)\int_{x_0}^{\infty} f_1(x)dx + c_{22}P(\Omega_2)\int_{x_0}^{\infty} f_2(x)dx + c_{21}P(\Omega_2)\int_{-\infty}^{x_0} f_1(x)dx$$

Мінімум R у точці x_0 досягається за умови $\left.\frac{dR}{dx}\right|_{x=x_0} = 0$.

Візьмемо похідну в точці x_0, враховуючи, що $\int f(x)dx=F(x)$, $F(-\infty)=0$, $F(\infty)=1$:

$$\frac{dR}{dx}\Big|_{x=x_0} = c_{11}P(\Omega_1)f_1(x) - c_{12}P(\Omega_1)f_1(x) - c_{22}P(\Omega_2)f_2(x) + c_{21}P(\Omega_2)f_2(x)$$

Звідси маємо наступне співвідношення для $x = x_0$:

$$\frac{P(\Omega_1)f_1(x)}{P(\Omega_2)f_2(x)} = \frac{c_{21} - c_{22}}{c_{12} - c_{11}} = \lambda$$

Цей вираз називається відношенням правдоподібності, а величина λ – коефіцієнтом правдоподібності. При значеннях $\frac{P(\Omega_1)f_1(x)}{P(\Omega_2)f_2(x)} \geq \lambda$ рішення приймається на користь Ω_1, при $\frac{P(\Omega_1)f_1(x)}{P(\Omega_2)f_2(x)} < \lambda$ – на користь Ω_2.

Якщо покласти, що $c_{11} = c_{22} = 0$ і $c_{12} = c_{21} = 1$, отримаємо:

$$\frac{P(\Omega_1)f_1(x)}{P(\Omega_2)f_2(x)} = 1$$

або, у логарифмічній формі,

$$\ln\frac{P(\Omega_1)f_1(x)}{P(\Omega_2)f_2(x)} = 0$$

Якщо значення ознаки для обох класів розподілено за нормальним законом

$$f(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

із середніми m_1, m_2 та середньоквадратичними відхиленнями σ_1, σ_2 відповідно, відношення правдоподібності у логарифмічній формі має вид:

$$\ln\frac{P(\Omega_1)}{P(\Omega_1)} + \ln\frac{\sigma_2}{\sigma_1} - \frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2} = 0$$

Тобто x_0 є розв`язком квадратного рівняння. Випадок, коли рівняння має два дійсні корені, представлений на рис. 16.

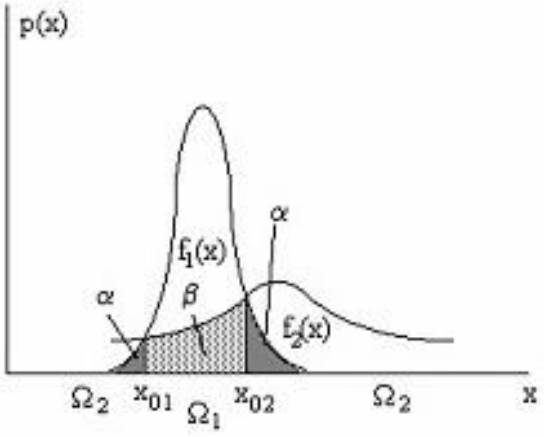


Рисунок 16 – Поділ класів із різними дисперсіями ознаки.

8. Ансамблеві методи

Типи ансамблевих методів у машинному навчанні

Методи ансамблю допомагають створити декілька моделей, а потім об'єднати їх для отримання поліпшених результатів. Деякі ансамблеві методи класифікуються на такі групи:

1. Послідовні методи

У такому методі ансамблю існують послідовно генеровані базові учні, в яких існує залежність даних. Усі інші дані базового учня мають певну залежність від попередніх даних. Отже, попередні помилкові дані налаштовуються виходячи з його ваги для покращення продуктивності загальної системи.

Приклад : Підвищення, (Бустінг, англ. Boosting)

2. Паралельний метод

У такому методі ансамблю базовий учень генерується в паралельному порядку, в якому залежності від даних немає. Усі дані базового учня формуються незалежно.

Приклад : укладання (стекінг, stacking)

3. Однорідний (гомогенний) ансамбль

Такий метод ансамблю - це поєднання однотипних класифікаторів. Але набір даних різний для кожного класифікатора. Це змусить більш точно працювати комбіновану модель після узагальнення результатів від кожної моделі. Цей тип ансамблевого методу працює з великою кількістю наборів даних. У гомогенному методі метод вибору особливостей однаковий для різних даних тренувань.

Приклад: Народні методи, такі як розфасування та підсилення, входять в однорідний ансамбль.

4. Гетерогенний ансамбль

Такий метод ансамблю - це поєднання різних типів класифікаторів або моделей машинного навчання, в яких кожен класифікатор будується на одних і тих же даних. Такий метод працює для невеликих наборах даних. У неоднорідному методі вибір особливостей для одних і тих же навчальних даних різний. Загальний результат цього ансамблевого методу здійснюється шляхом усереднення всіх результатів кожної комбінованої моделі.

Чотири перевірених способи робити ансамблів.

Рисунок 17 – Основні типи ансамблевих методів

Bagging (пакування в мішок чи сумку)

Bagging, він же Bootstrap AGGregatING. Ідея – навчаємо один алгоритм багато разів на випадкових вибірках з вхідних даних (навчальної вибірки). В кінці усереднюємо відповіді.

Дані в випадкових вибірках можуть повторюватися. Тобто з набору 1-2-3 ми можемо робити вибірки 2-2-3, 1-2-2, 3-1-2 і так поки не набридне. На них ми навчаємо один і той же алгоритм кілька разів, а в кінці знаходимо відповідь простим голосуванням.

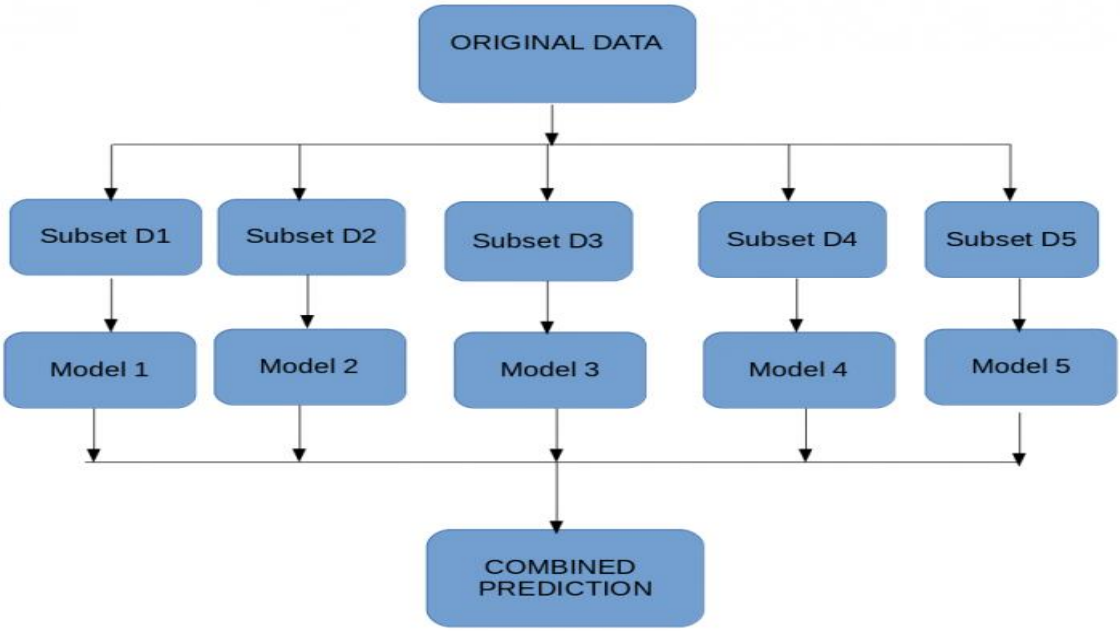


Рисунок 18 – Структура алгоритмів Bagging

Random Forest

Найпопулярніший приклад бегінга – алгоритм Random Forest (рис. 19).

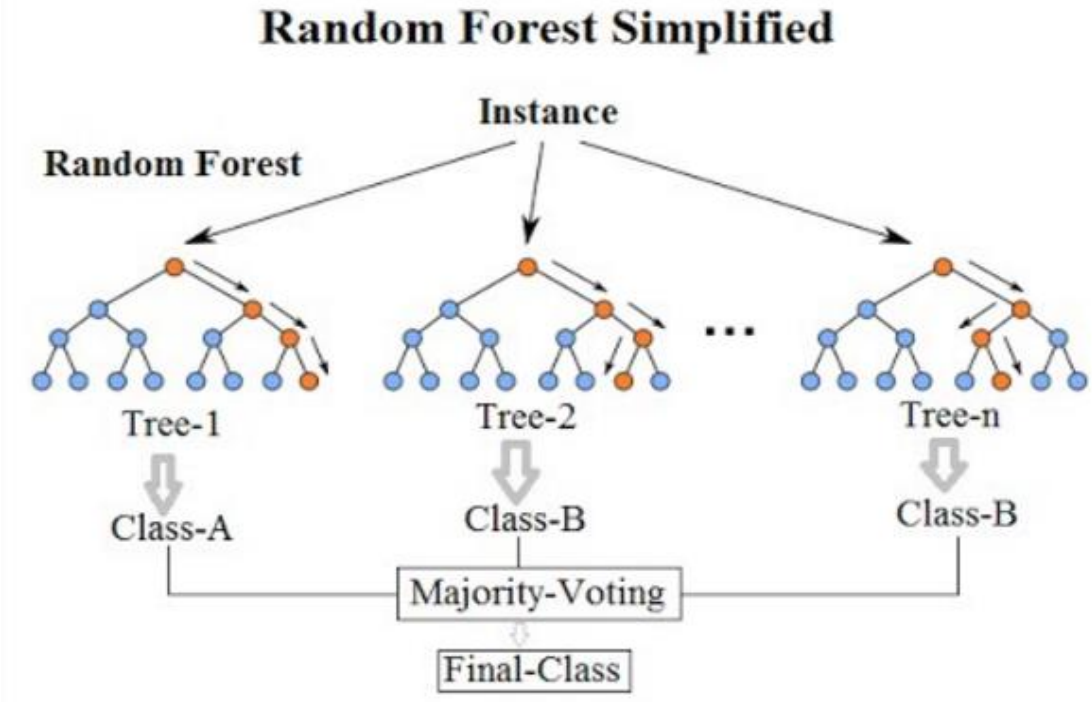


Рисунок 19 – Структура алгоритмів Random Forest

Цей метод ансамблю поєднує дві моделі машинного навчання, тобто завантаження та агрегацію, в єдину модель ансамблю. Мета методу – зменшити велику дисперсію моделі. Нехай є великий набір даних (скажімо, 1000 зразків) підвибіркою (скажімо, 10 підвібірок, кожна містить 100 зразків даних). Кілька дерев рішень будуються на кожній підвибірки. Для ефективності моделі кожне окреме дерево рішень нарощується в глибину. Результати кожного дерева рішень узагальнюються, щоб отримати остаточний прогноз. Дисперсія узагальнених даних зменшується. Точність прогнозування моделі в методі упаковки залежить від кількості використовуваного дерева рішень. Різні підвідбірки даних створюються випадковим чином. Вихід кожного дерева має високу кореляцію.

Перевага бегінгу, наприклад, перед нейромережами – можливість працювати паралельно.

Boosting (підвищення)

Boosting – ітераційний алгоритм, що реалізує «сильний» класифікатор, який дозволяє досягти довільно малої помилки навчання (на навчальній вибірці) на основі композиції "слабких" класифікаторів, кожен з яких кращий, ніж просто вгадування, тобто ймовірність правильної класифікації більше 0,5.

Ключова ідея: використання вагових версій одних і тих же навчальних даних замість випадкового вибору їх підмножини.

У Boosting рівна вага (рівномірний розподіл ймовірностей) надається вибірковим навчальним даним (скажімо, D1) на самому початку раунду. Потім ці дані (D1) передаються базовому учню (скажімо, L1). Неправильно класифікованим екземплярам L1 присвоюється вага, що вища, ніж правильно класифікованим екземплярам, але з урахуванням того, що загальний розподіл ймовірностей буде дорівнювати 1. Ці покращені дані (скажімо, D2) потім передаються другому базовому учню (скажімо, L2) і так далі. Потім результати об’єднуються у формі голосування.

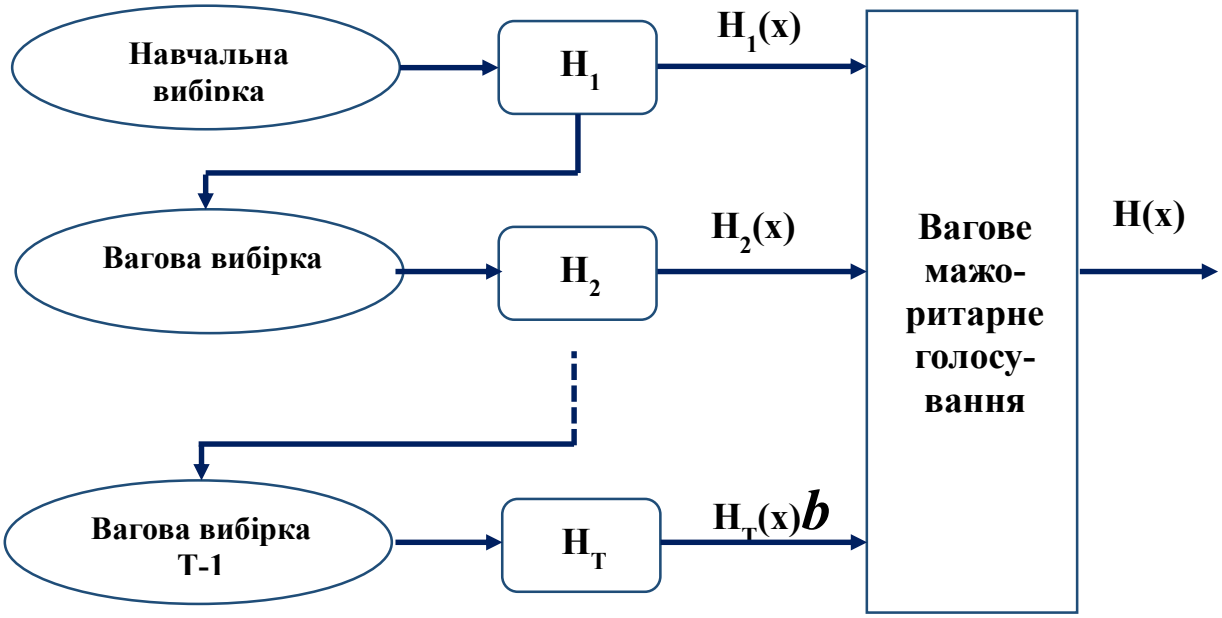


Рисунок 20 – Boosting (загальна схема)

Слабкі класифікатори утворюються послідовно, розрізняючись тільки вагами навчальних даних, які залежать від точності попередніх класифікаторів.

Базові класифікатори повинні бути слабкими, з сильних хорошу композицію не збудувати ("бритва Оккама ").

Причини цього: сильний класифікатор, даючи нульову помилку на навчальних даних, адаптується і композиція буде складатися з одного класифікатора один, навіть сильний, класифікатор може дати "поганий" прогноз на даних тестування, даючи " хороші" результати на навчальних даних.

Stacking (укладання)

Ідея – навчаємо кілька різних алгоритмів і передаємо їх результати на вхід останньому, який приймає остаточне рішення (рис. 21).

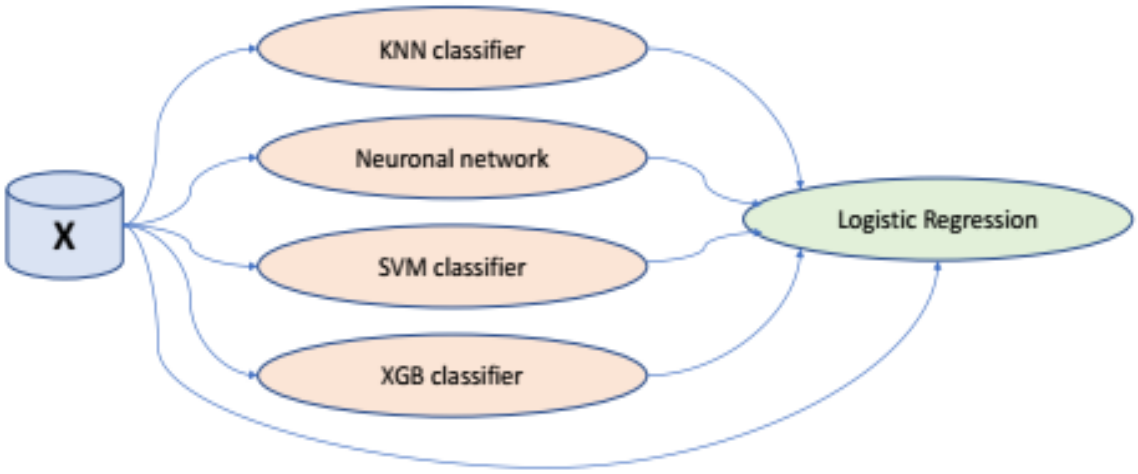


Рисунок 21 – Stacking (загальна схема)

Перед початком стекованого навчання нам потрібно вибрати кількість M алгоритмів, які ми хочемо об'єднати: L_1,. … ., L_M. І ми позначаємо через L алгоритм навчання для нашої метамоделі.

- Нехай D = f(X_1,Y_1),. … .,(X_n,Y_n)g позначають наш набір даних. На першому етапі, щоб створити так звані моделі рівня 0 або узагальнювачі, ми застосовуємо алгоритми навчання рівня 0 до вихідного набору даних, випадковим чином розділеного за допомогою перехресної перевірки для навчання та тестування майбутніх моделей. Це h_m=L_m(D) для m = 1,. . ., M.
- Для кожного X_i тестових зразків рівня 0 ми використовуємо кожен узагальнювач для прогнозування їх змінної відповіді. Тоді ми позначимо вихід як z_{mi}=h_m(Xi).
- Ці виходи разом із їхніми справжніми значеннями Y_i утворюють вектор передбачення з M + 1 координатами (Y_i, z_{1i} ,. ... .,$z_{M,i}$). Для кожного екземпляра тестової вибірки ці вектори збираються в новий набір даних D_0, дані рівня 1.
- Останній дозволяє навчати та тестувати L, алгоритм навчання рівня 1. Ми застосовуємо його до нового набору даних так само, як і раніше. У позначеннях h=L(D_0).
- Прогнозувати за новими даними, отримані узагальнювачі h_1,. …, h_M. Остаточним прогнозом для нових даних x є: f(x) = h(h_1(x), ..., h_M(x))

Контрольні запитання:

1. Як визначається поняття машинного навчання та у чому полягає його відмінність від традиційного програмування?

2. Яку роль відіграють три основні компоненти машинного навчання — дані (Data), ознаки (Features) та алгоритм (Algorithm) — у побудові моделей?

3. Які основні типи навчання виділяють у машинному навчанні, та в чому полягають їхні ключові відмінності?

4. Поясніть, як реалізується підхід Supervised Learning (навчання з учителем) і які завдання до нього належать.

5. У чому полягає специфіка Unsupervised Learning (навчання без учителя) і які приклади його практичного застосування?

6. Який принцип покладено в основу Reinforcement Learning (навчання з підкріпленням), та як реалізується взаємодія агента із середовищем?

7. Які основні типи задач вирішуються методами машинного навчання (регресія, класифікація, кластеризація, прогнозування, виявлення аномалій)?

8. У чому полягає гіпотеза компактності у задачах класифікації та як вона впливає на побудову систем розпізнавання образів?

9. Опишіть основні підходи до побудови систем розпізнавання образів і класифікаторів: геометричний, статистичний, біологічний, нечіткий.

10. Які принципи покладено в основу ансамблевих методів (Bagging, Boosting, Stacking) і яку перевагу вони забезпечують порівняно з окремими моделями?

ТЕМА 3. ШТУЧНІ НЕЙРОННІ МЕРЕЖІ

Штучні нейромережі є моделями нейронної структури мозку, який здатен сприймати, обробляти, зберігати та продукувати інформацію, що представлена образами. Особливістю мозку також є навчання та самонавчання на власному досвіді. Адаптивні системи на основі штучних нейронних мереж дозволяють з успіхом вирішувати проблеми розпізнавання образів, виконання прогнозів, оптимізації, асоціативної пам'яті, керування та інші інтелектуальні завдання, не використовуючи традиційного програмування.

У порівнянні з традиційними комп'ютерами з архітектурою фон Неймана, штучні нейронні мережі мають інші принципи обробки інформації та нові якості:

- Розподілене представлення інформації і паралельні обчислення.
- Адаптивність до змін у середовищі.
- Здатність до навчання й узагальнення.
- Толерантність до помилок.
- Низьке енергоспоживання.

Нейромережі не можна вважати доцільним рішенням для всіх обчислювальних проблем. Для багатьох завдань ідеальними є традиційні комп'ютери та обчислювальні методи. Сучасні комп’ютери перевершують людину за здатністю робити числові й символьні обчислення. Однак людина може без зусиль вирішувати складні задачі сприйняття зовнішніх даних (наприклад, впізнавання людини в натовпі) з такою швидкістю і точністю, що значно перевищує можливості потужних комп’ютерів.

1. Моделі штучного нейрону

Всі штучні нейромережі конструюються з базового формуючого блоку - штучного нейрону, що моделює основні функції природного нейрона.

При функціонуванні штучний нейрон одночасно отримує багато вхідних сигналів. Кожен вхід має власний коефіцієнт, що називається синаптичною вагою і моделює різноманітні синаптичні з’єднання біологічних нейронів. Ваговий коефіцієнт збільшує або зменшує значення входу, тому, ваги суттєвого входу підсилюються і, навпаки, вага несуттєвого входу примусово зменшується,

що визначає інтенсивність вхідного сигналу. Ваги можуть змінюватись відповідно до навчальних прикладів, топології мережі та навчальних правил.

Вхідні сигнали x_n помножені на вагові коефіцієнти з'єднання w_n додаються (блок суматора), проходять через передатну функцію та генерують результат (рис.1).

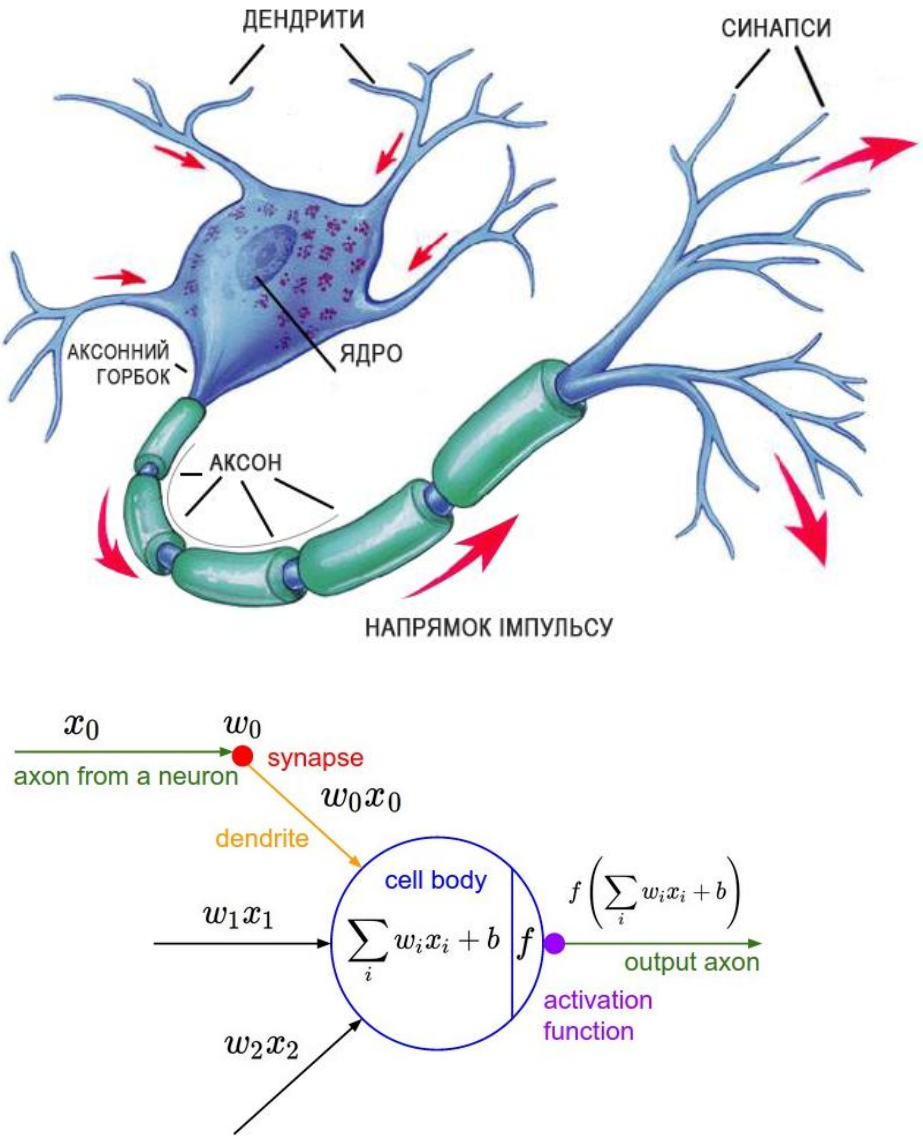


Рис. 1. Базовий штучний нейрон

Поглиблені знання щодо будови біологічного нейрона, як ефективного перетворюючого інструмента, можна розглядати як джерело базових ідей та концепцій по створенню нових типів нейромереж.

В наявних на цей час нейронних мережах штучні нейрони називаються «процесорами» або «елементами обробки» і вони мають значно більше можливостей, ніж базовий нейрон, що описано вище. На рис. 2 зображено детальну схему штучного нейрону.

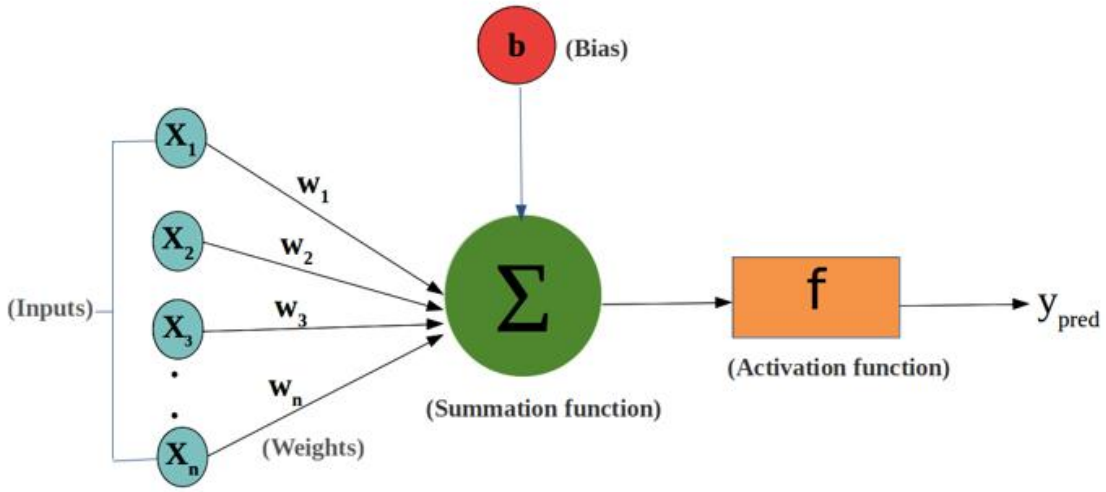


Рис. 2. Розширена модель штучного нейрону

Модифіковані входи (зважені синаптичними коефіцієнтами) надходять до блока суматора, де, окрім додавання входів можна виконувати інші операції, наприклад, обрати середнє, найбільше чи найменше значення входів, здійснити логічні операції AND чи OR, або виконати інші функції. Інколи до блоку суматора додається функції активації, яка дозволяє функції суматора зміщуватися в часі.

Значення, що спродуковано в блоці суматора надсилається до передатної функції, яка за допомогою певного алгоритму, обмежує вихід в певному діапазоні, наприклад, [0 ÷ 1] чи [-1÷1]. В існуючих нейромережах в якості передатних функцій можуть бути використані сигмоїда, синус, гіперболічний тангенс та поліноми. Приклад передатної функції показано на рис. 3.

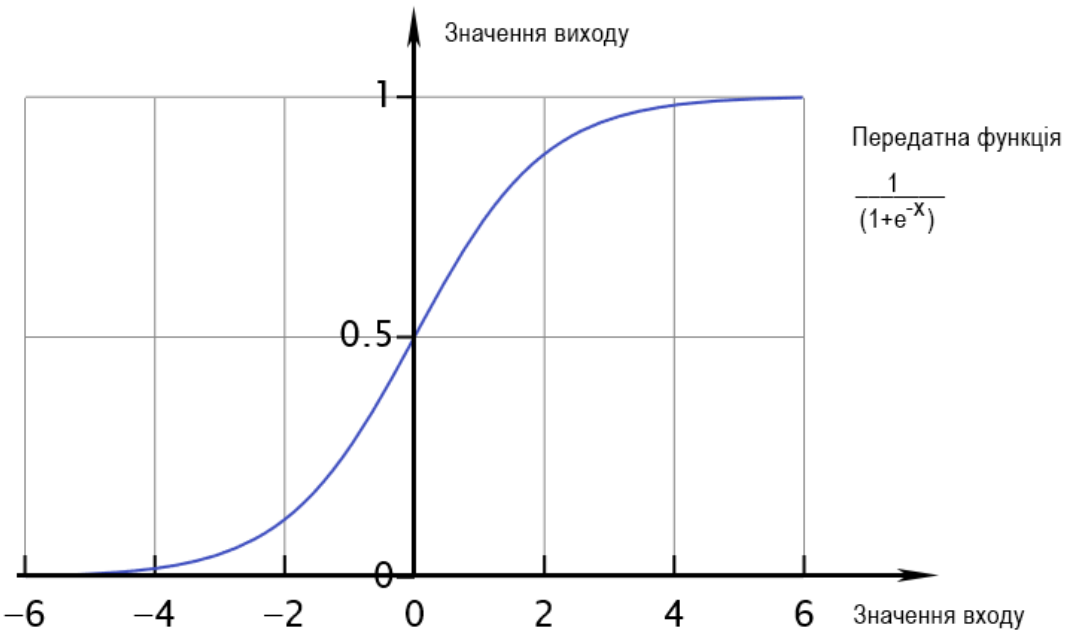


Рис. 3. Сигмоїдна передатна функція

Результат передатної функції надходить до входів інших нейронів або до зовнішнього з'єднання, відповідно до структури нейромережі.

2. Компоненти штучного нейрона

Незалежно від розташування та функціонального призначення, штучні нейрони мають спільні компоненти. Нижче представлено сім основних компонентів.

Компонента 1. Нормалізація входів

Нормалізація вхідних даних - це процес, при якому всі вхідні дані проходять процес "вирівнювання", тобто приведення до певного інтервалу, наприклад [0,1] або [-1,1]. Нормалізація дозволяє значно підвищити швидкість збіжності алгоритму навчання нейронної мережі, оскільки не можна порівнювати величини різних порядків. Якщо не провести нормалізацію, то вхідні дані будуть надавати додатковий вплив на нейрон, що може привести до невірних рішень.

Компонента 2. Вагові коефіцієнти

До нейрону одночасно надходять сигнали від багатьох входів. Кожен вхід змінює своє значення відповідно до коефіцієнта синаптичної ваги. Ваги суттєвого входу підсилюються і навпаки вага несуттєвого входу примусово зменшується, що визначає інтенсивність вхідного сигналу. Ваги можуть змінюватись відповідно до алгоритму навчання. Після проходження навчання ваги фіксуються і в подальшому використовуються.

Компонента 3. Функція суматора

Першим кроком дії нейрону є обчислення зваженої суми всіх входів. Математично, вхідні сигнали та відповідні їм ваги представлено векторами (x_{10}, x_{20} ... x_{n0}) та (w_{10}, w_{20} . . . w_{n0}). Добуток цих векторів є загальним вхідним сигналом.

Спрощеною функцією суматора є множення кожної компоненти вектора x на відповідну компоненту вектора w: вхід$_1$ = x_1 * w_1, вхід$_2$ = x_2 * w_2, і знаходження суми всіх добутків: вхід$_1$ + вхід$_2$ + . . . + вхід$_n$. Результатом є не багатоелементний вектор, а єдине число.

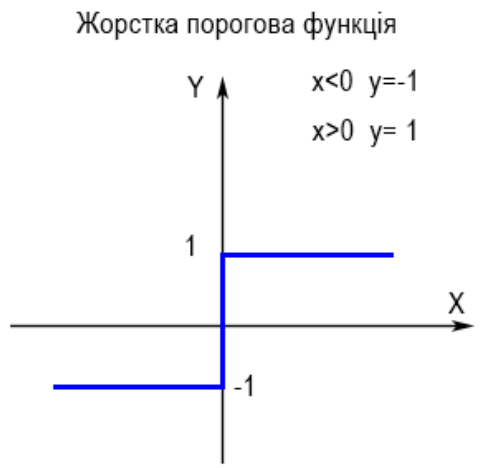
Функція суматора може бути складнішою, наприклад, вибір мінімуму, максимуму, середнього арифметичного, добутку або інший нормалізуючий алгоритм. Вхідні сигнали та вагові коефіцієнти можуть комбінуватись багатьма способами перед надходженням до передатної функції. Особливі алгоритми для комбінування входів нейронів визначаються обраними мережною архітектурою та парадигмою (алгоритмом навчання).

В деяких нейромережах функції суматора виконують додаткову обробку, так звану функцію активації, яка зміщує вихід функції суматора відносно часу. Функції активації на сьогодні час використовуються обмежено і більшість сучасних нейронних реалізацій використовують функцію активації "тотожності", яка еквівалентна її відсутності. Цю функцію доцільніше використовувати як компоненту мережі в цілому, ніж як компоненту окремого нейрона.

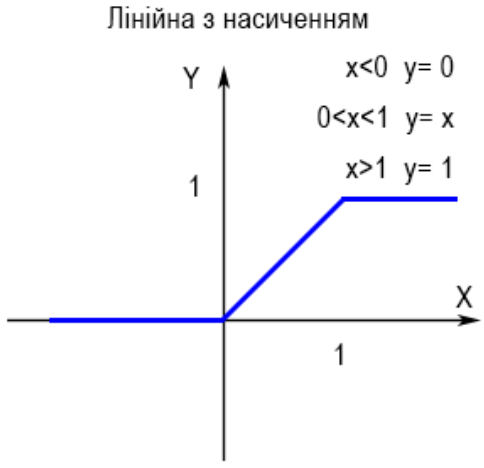
Компонента 4. Передатна функція

Результат функції суматора перетворюється у вихідний сигнал через алгоритмічний процес відомий як передатна функція. У передатній функції для визначення виходу нейрона загальна сума порівнюється з деяким порогом. Якщо сума є більшою за значення порогу, елемент обробки генерує сигнал, в противному випадку сигнал не генерується або генерується гальмуючий сигнал.

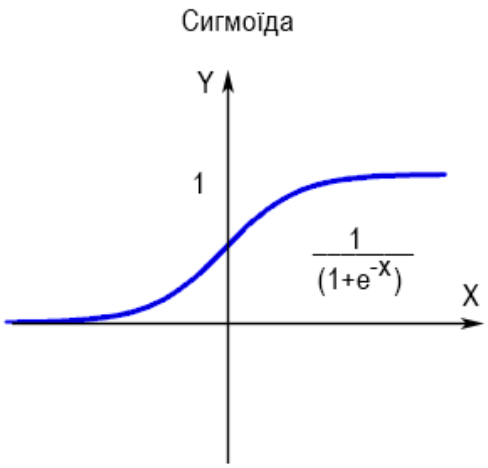
Переважно застосовують нелінійну передатну функцію, оскільки лінійні функції є обмеженими, а вихід є пропорційним до входу. На рис. 9.4 зображено типові передатні функції.



Для простої передатної функції нейромережа може видавати 0 та 1, 1 та -1 або інші числові комбінації. Передатна функція в таких випадках є "жорстким обмежувачем" або пороговою функцією (рис. 4 а).



Передатна функція лінійна з насиченням віддзеркалює вхід всередині заданого діапазону і діє як жорсткий обмежувач за межами цього діапазону. Це лінійна функція, яка відсікається до мінімальних та максимальних значень, роблячи її нелінійною (рис. 4б).



S-подібна передатна функція наближує мінімальне та максимальне значення у асимптотах і називається сигмоїдою (рис. 4в), коли її діапазон [0, 1].

S-подібна передатна функція називається гіперболічним тангенсом (рис. 4г), при діапазоні [-1, 1]. Важливою рисою цих кривих є неперервність функцій та їх похідних. Застосування сигмоїдних функцій надає добрі результати і має широке застосування.

Рис.4. Типові передатні функції

Для різних нейромереж можуть вибиратись інші передатні функції, зокрема поліноміальні.

Перед надходженням до передатної функції до вхідного сигналу деколи додають однорідно розподілений випадковий шум, джерело та кількість якого визначається режимом навчання. Цей шум називається "температурою" штучних нейронів, що надає математичній моделі елемент реальності.

Компонента 5. Вихідна функція (змагання)

По аналогії з біологічним нейроном, кожний штучний нейрон має один вихідний сигнал, який передається до сотень інших нейронів. Переважно, вихід

є прямо пропорційним до результату передатної функції. В деяких мережних топологіях результати передатної функції змінюються для створення змагання між сусідніми нейронами. Нейронам дозволяється змагатися між собою, блокуючи дії нейронів, що мають слабший сигнал. Змагання (конкуренція) може відбуватись між нейронами, які знаходяться на одному або різних прошарках. По-перше, конкуренція визначає, який штучний нейрон буде активним і забезпечить вихідний сигнал. По-друге, конкуруючі виходи допомагають визначити, який нейрон буде брати участь у процесі навчання.

Компонента 6. Функція навчання

Метою функції навчання є налаштування вагових коефіцієнтів на входах кожного елемента обробки відповідно до певного алгоритму навчання для досягнення бажаного результату. Існує два типи навчання: контрольоване та неконтрольоване. Контрольоване навчання вимагає навчальної множини даних або вчителя, що ранжирує ефективність результатів мережі. У випадку неконтрольованого навчання система самоорганізовується за внутрішнім критерієм, закладеним в алгоритм навчання.

Компонента 7. Функція похибки та поширюване назад значення

У більшості мереж, що застосовують контрольоване навчання обчислюється різниця між спродукованим та бажаним виходом. Похибка відхилення (біжуча похибка) обробляється функцією похибки відповідно до заданої мережної архітектури. Після проходження всіх прошарків поточна похибка поширюється назад до попереднього прошарку, алгоритм навчання корегує вагові коефіцієнти, що враховується в наступному циклі навчання.

3. Архітектура з'єднань штучних нейронів

Об'єднуючись у мережі, нейрони утворюють системи обробки інформації, які забезпечують ефективну адаптацію моделі до постійних змін з боку зовнішнього середовища. В процесі функціонування мережі відбувається перетворення вхідного вектору сигналів у вихідний.

Конкретний вид перетворення визначається як архітектурою нейромережі так і характеристиками нейронних елементів, засобами керування та синхронізації інформаційних потоків між нейронами. Важливим фактором

ефективності мережі є встановлення оптимальної кількості нейронів, типів зв'язків між ними та відповідних правил передачі інформації.

При описі нейромереж використовують кілька усталених термінів, які в різних джерелах можуть мати різне трактування, зокрема:

- Структура нейромережі - спосіб зв'язків нейронів у нейромережі.
- Архітектура нейромережі - структура нейромережі та типи нейронів.
- Парадигма нейромережі - спосіб навчання та використання; іноді містить і поняття архітектури.

На основі однієї архітектури можуть бути реалізовані різні парадигми нейромережі і навпаки.

Серед відомих архітектурних рішень виділяють групу слабозв'язаних нейронних мереж, у випадку, коли кожний нейрон мережі зв'язаний лише із сусідніми (рис.5). Якщо входи кожного нейрона зв'язані з виходами всіх решта нейронів, тоді мова йде про повнозв'язані нейромережі (рис.6).

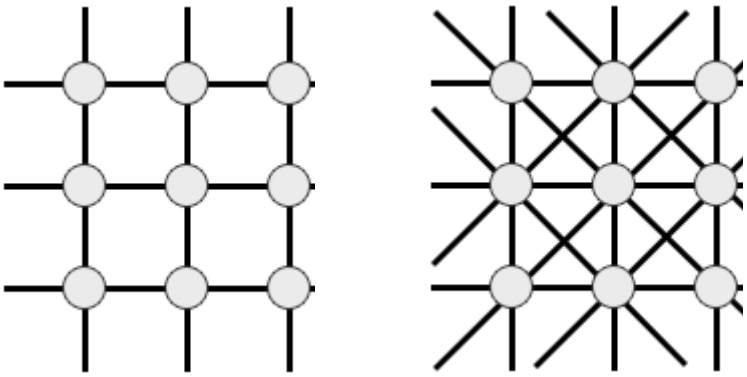


Рис. 5. Слабозв'язані нейромережі

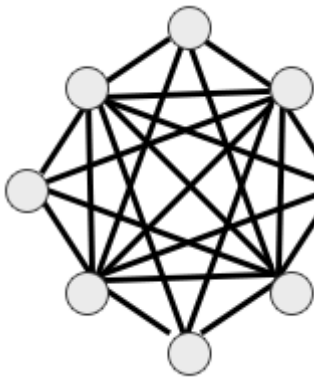


Рис. 6.
Повнозв'язана мережа

Зрозуміло, що такий поділ носить дещо теоретичний характер. Самим поширеним варіантом архітектури є багатошарові мережі. Нейрони в даному випадку об'єднуються у з'єднані між собою прошарки з єдиним вхідним вектором сигналів.

Зовнішній вхідний вектор подається на вхідний прошарок нейронної мережі (рецептори). Виходами нейронної мережі є вихідні сигнали останнього прошарку (ефектори). Окрім вхідного та вихідного прошарків, нейромережа має один або декілька прихованих прошарків нейронів, які не мають контактів із зовнішнім середовищем.

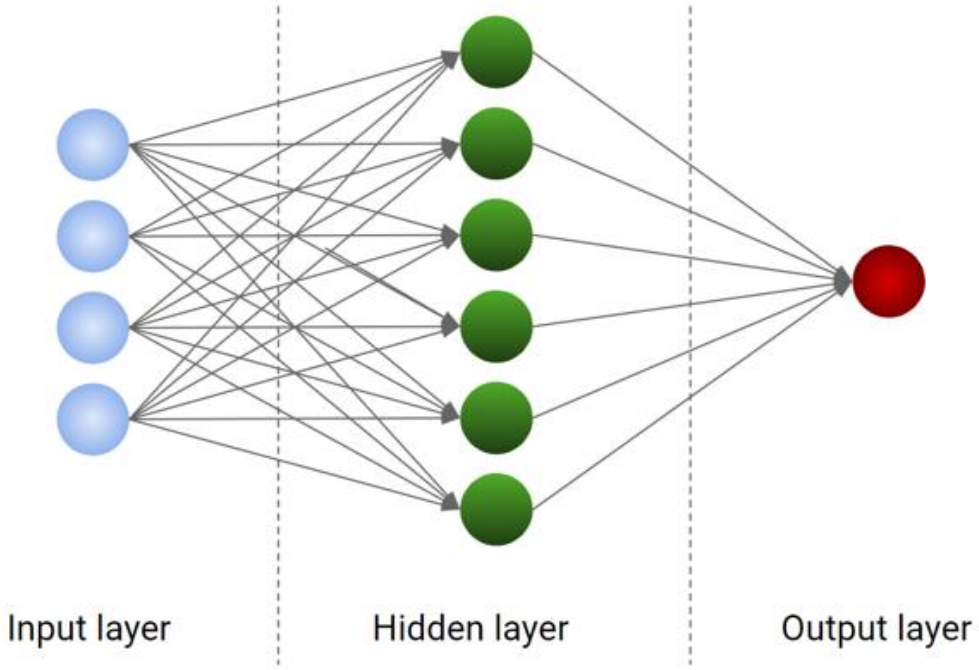


Рис.7. Багатошаровий тип з'єднання нейронів

На рис.7 показано типову структуру штучних нейромереж. Хоча існують мережі, які містять лише один прошарок, або навіть один елемент, більшість застосувань вимагають мережі, які містять як мінімум три типи прошарків - вхідний, прихований та вихідний. Прошарок вхідних нейронів отримує дані або з вхідних файлів, або безпосередньо з електронних давачів. Вихідний прошарок надсилає інформацію безпосередньо до зовнішнього середовища, до вторинного комп'ютерного процесу або до інших пристроїв. Між цими двома прошарками може бути кілька прихованих прошарків, які містять багато нейронів у різноманітних зв'язаних структурах. Входи та виходи кожного з прихованих нейронів надходять до інших нейронів.

Зв'язки між нейронами різних прошарків називають проективними. Зв'язки, що скеровані від вхідних прошарків до вихідних називаються аферентними, в інакшому випадку, при зворотному напрямку зв'язків -

еферентними. Зв'язки між нейронами одного прошарку відносять до бічних (латеральних).

Важливим аспектом нейромереж є напрямок зв'язку від одного нейрону до іншого. У більшості мереж кожен нейрон прихованого прошарку отримує сигнали від нейронів попереднього прошарку та зазвичай від нейронів вхідного прошарку. Після виконання операцій над сигналами, нейрон передає свій вихід до всіх нейронів наступних прошарків, забезпечуючи шлях передачі вперед (Feedforward) на вихідний прошарок (рис.8).

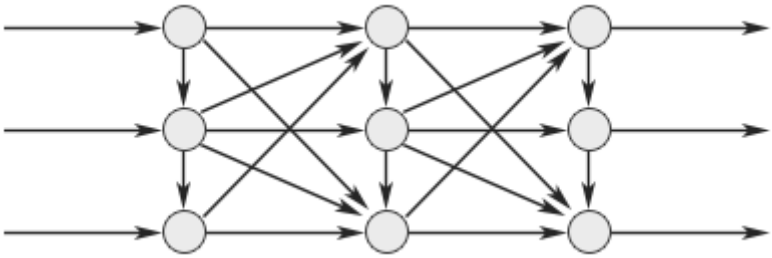


Рис. 8. Мережа з прямими і бічними зв’язками (прямого поширення)

При зворотному зв'язку, вихід нейронів прошарку скеровується до нейронів попереднього прошарку (рис. 9).

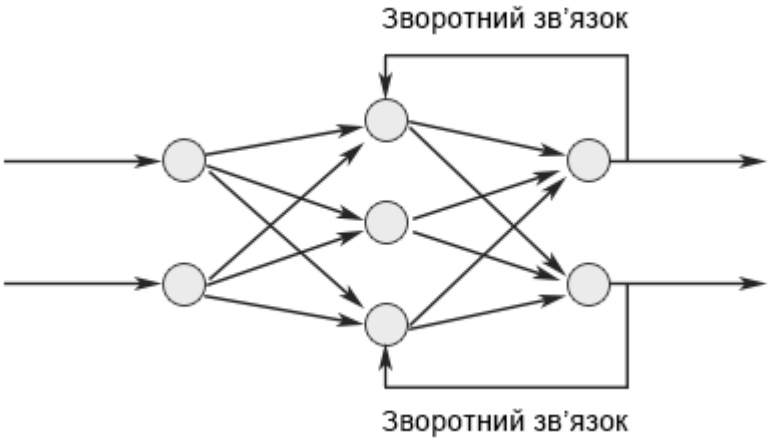


Рис.9. Мережа зі зворотнім зв'язком (зворотного поширення)

Спосіб, яким нейрони з'єднуються між собою має значний вплив на роботу мережі. Більшість програмних моделей дозволяють користувачу додавати, видаляти та керувати з'єднаннями як завгодно. Корегуючи параметри, зв'язки можна робити як збуджуючими так і гальмуючими.

За архітектурою зв'язків, більшість відомих нейромереж, що знайшли практичне застосування, можна згрупувати у два великих класи:

- Мережі прямого поширення (з односкерованими послідовними зв'язками).
- Мережі зворотного поширення (з рекурентними зворотними зв'язками).

Мережі прямого поширення відносять до статичних, оскільки на задані входи нейронів надходить вектор вхідних сигналів, що не залежить від попереднього стану мережі. Рекурентні мережі вважаються динамічними, оскільки за рахунок зворотних зв'язків (петель) входи нейронів модифікуються в часі, що приводить до змін станів мережі.

4. Навчання нейронних мереж

Оригінальність нейромереж, як аналога біологічного мозку, полягає у здібності до навчання за прикладами, що складають навчальну множину. Процес навчання нейромереж розглядається як налаштування архітектури та вагових коефіцієнтів синаптичних зв'язків відповідно до даних навчальної множини так, щоб ефективно вирішити поставлену задачу.

Після визначення архітектури нейронної мережі її потрібно «навчити», тобто підібрати такі значення її ваг, щоб вона працювала належним чином. Навчити нейронну мережу - значить, повідомити їй, чого від неї домагаються. Цей процес дуже схожий на навчання дитини алфавіту. Після того, як показали дитині зображення літери "А", слідує запитання: "Яка це літера?" Якщо відповідь невірна, то дитині підказують вірну відповідь: "Це літера А".

Дитина запам'ятовує приклад разом з вірною відповіддю, тобто в її пам'яті відбуваються деякі зміни в потрібному напрямку. Процес пред'явлення літер повторюється, поки всі літери будуть твердо запам'ятовані. Такий процес називають "навчання з вчителем". Навчання нейронної мережі відбувається аналогічно.

Процес навчання нейронної мережі полягає в налаштуванні її внутрішніх параметрів під конкретну задачу (рис.10).

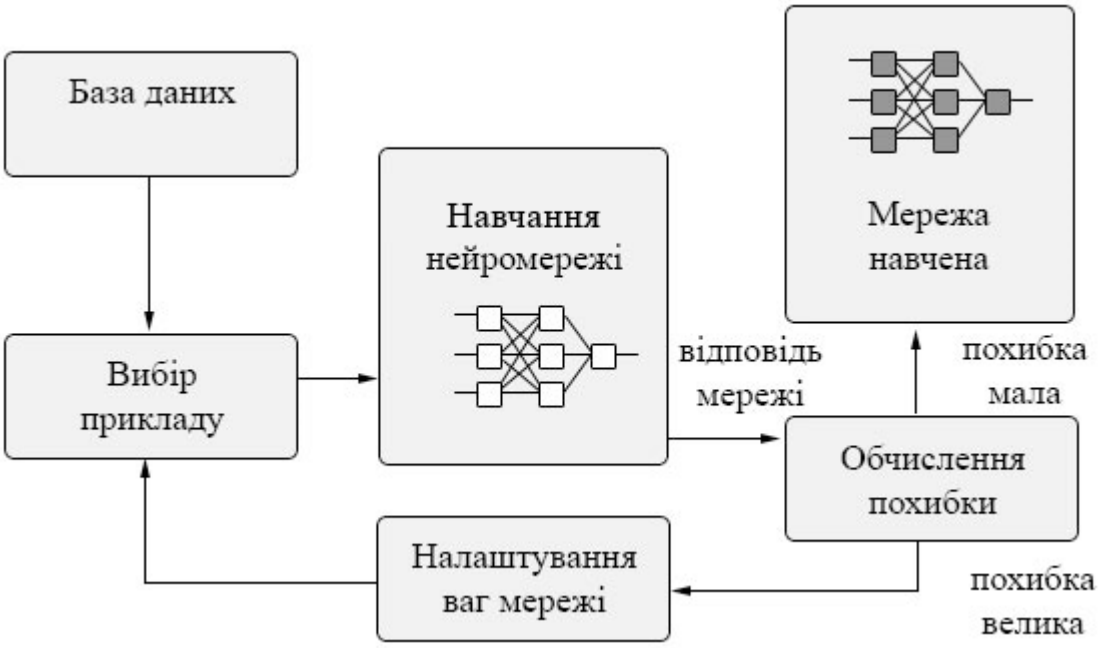


Рис. 10. Процес навчання нейромережі

Алгоритм роботи нейронної мережі є ітеративним, його кроки називають епохами або циклами. Епоха - одна ітерація в процесі навчання, що містить пред'явлення всіх прикладів з навчальної множини і, можливо, перевірку якості навчання на контрольній множині.

Процес навчання здійснюється на навчальній вибірці (множині). Навчальна вибірка містить набір даних про предметну область (об’єкт, явище, процес), що поділяється на вхідні значення і відповідні їм вихідні значення. Наприклад, після пред'явлення зображення літери "А" на вхід нейронної мережі, вона видає деяку відповідь, не обов'язково вірну.

В навчальній множині присутня вірна (бажана) відповідь, і сенсом навчання є те, щоб на виході нейронної мережі з міткою "А" рівень сигналу був максимальний. В ході навчання нейронна мережа знаходить певні залежності вихідних полів від вхідних.

Зазвичай, в якості бажаного виходу в задачі класифікації беруть набір (1, 0, 0, ...), де 1 стоїть на виході з міткою "А", а 0 - на всіх інших виходах. Обчислюючи різницю між бажаною і реальною відповідями мережі, утворюється похибка, що надалі обробляється відповідною функцією. Функція

похибки - це цільова функція, що дозволяє оцінити якість роботи нейронної мережі і потребує мінімізації в процесі керованого навчання нейронної мережі.

Алгоритмом навчання є набір формул, який дозволяє за функцією похибки обчислити необхідні зміни для вагових коефіцієнтів зв’язків нейронної мережі.

Одну літеру (або різні зображення однієї літери) можна пред'являти нейронній мережі багато разів. У цьому сенсі навчання швидше нагадує повторення вправ в спорті - тренування. Після багаторазового пред'явлення прикладів вагові коефіцієнти зв’язків нейронної мережі стабілізуються, причому нейронна мережа надає правильні відповіді на всі (або майже всі) приклади з навчальної множини. У такому випадку говорять, що "нейронна мережа вивчила всі приклади", "нейронна мережа навчена", або "нейронна мережа натренована".

Які вхідні поля (ознаки) необхідно використовувати? Спочатку здійснюється евристичний вибір, далі кількість входів може бути змінено. Складність може викликати питання про кількість прикладів в наборі даних. Вся інформація, яку нейронна мережа має про завдання, міститься в наборі прикладів. Тому, якість навчання нейронної мережі безпосередньо залежить від кількості прикладів в навчальній вибірці, а також від того, наскільки повно ці приклади описують це завдання.

Так, наприклад, безглуздо використовувати нейронну мережу для передбачення фінансової кризи, якщо в навчальній вибірці не представлено жодної кризи. Для повноцінного навчання нейронної мережі потрібна репрезентативна і доволі велика вибірка з сотні (а краще тисячі) прикладів. Кількість необхідних прикладів залежить від складності розв'язуваної задачі.

Розробник має надати можливості вибору кількості прошарків у мережі і кількості нейронів в кожному прошарку. Далі необхідно призначити такі значення ваг, які зможуть мінімізувати похибку рішення. Від якості навчання нейронної мережі залежить її здатність вирішувати поставлені перед нею завдання.

4.1. Перенавчання нейронної мережі

При навчанні нейронних мереж часто виникає проблема перенавчання (overfitting). Перенавчання, або надмірно близька підгонка - надлишкова точна

відповідність нейронної мережі до конкретного набору навчальних прикладів, при якому мережа втрачає здатність до узагальнення.

Перенавчання виникає у разі занадто довгого навчання, недостатньої кількості навчальних прикладів або занадто складної структури нейронної мережі.

Перенавчання пов'язано з тим, що вибір навчальної множини є випадковим. З перших кроків навчання відбувається зменшення похибки. На наступних кроках з метою зменшення похибки (цільової функції) параметри підлаштовуються під особливості навчальної множини. Однак при цьому відбувається "підлаштування" не під загальні закономірності ряду, а під особливості його частини - навчальної підмножини. При цьому точність прогнозу зменшується.

Один з варіантів боротьби з перенавчанням мережі - розділення навчальної вибірки на дві множини (навчальну і тестову). На навчальній множині відбувається навчання нейронної мережі. На тестовій множині здійснюється перевірка побудованої моделі. Ці множини не повинні перетинатися.

4.2. Застосування нейронної мережі

Після того, як нейронна мережа навчена, її можна застосовувати для вирішення різних завдань (рис.11).

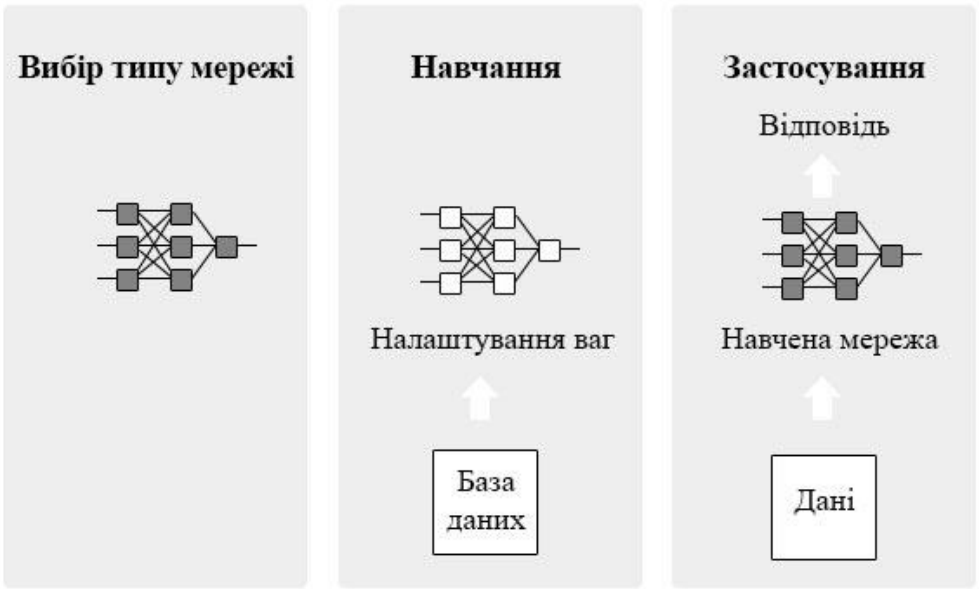


Рис.11. Застосування нейромережі

Найважливіша особливість людського мозку полягає в тому, що, одного разу навчившись певного процесу, він може вірно діяти і в тих ситуаціях, в яких він не бував під час навчання. Наприклад, людина може читати майже будь-який почерк, навіть якщо бачить його вперше в житті.

Добре навчена нейронна мережа може з великою ймовірністю правильно реагувати на нові дані, що не були їй відомі раніше. Наприклад, можна написати літеру "А" іншим почерком, а потім запропонувати нейронній мережі класифікувати нове зображення. Ваги навченої нейронної мережі зберігають багато інформації про схожість і відмінності літер, тому, можна розраховувати на правильну відповідь і для нового варіанту зображення.

4.3. Контрольоване навчання

Значну кількість рішень отримано від нейромереж з контрольованим навчанням, де поточне значення виходу постійно порівнюється з бажаним. Ваги на початку встановлюються випадково, але під час наступних ітерації корегуються для досягнення близької відповідності між бажаним та поточним значенням на виході. Існуючі алгоритми навчання націлені на мінімізацію поточних похибок всіх елементів обробки, що відбувається за певний час неперервною зміною вагових коефіцієнтів до досягнення прийнятної точності мережі.

Перед використанням, нейромережа з контрольованим навчанням повинна бути навченою. Фаза навчання може тривати довго, зокрема, в прототипах систем з невідповідною процесорною потужністю навчання може займати кілька годин.

Навчання вважається завершеним при досягненні визначеного рівня ефективності нейромережі. Цей рівень означає, що мережа досягла бажаної статистичної точності, оскільки вона видає бажані виходи для заданої послідовності входів. Після навчання коефіцієнти синаптичних ваг фіксуються для подальшого застосування. Деякі типи мереж допускають навчання під час використання, що допомагає мережі адаптуватись до умов, які повільно змінюються.

Навчальні множини повинні бути достатньо великими, щоб містити всю необхідну інформацію для виявлення важливих особливостей і зв'язків. Також, навчальні приклади повинні містити широке різноманіття даних.

Якщо мережа навчається лише для одного прикладу, ваги старанно встановлені для цього прикладу, радикально змінюються у навчанні для наступного прикладу. Попередні приклади при навчанні наступних просто забуваються. В результаті система повинна навчатись всьому разом, знаходячи найкращі вагові коефіцієнти для загальної множини прикладів.

Наприклад, у навчанні системи розпізнавання піксельних образів для десятьох цифр, які представлені двадцятьма прикладами кожної цифри, всі приклади цифри "сім" не доцільно представляти послідовно. Краще надати мережі спочатку один тип представлення всіх цифр, потім другий тип і так далі.

Головним чинником для успішної роботи мережі є представлення вхідних і вихідних даних. Штучні мережі працюють лише з числовими вхідними даними, отже, необроблені дані, що надходять із зовнішнього середовища проходять процедуру нормалізації даних відповідно до діапазону всіх значень.

Якщо після контрольованого навчання нейромережа ефективно опрацьовує дані навчальної множини, важливим стає її ефективність при роботі з даними, які не використовувались для навчання. У випадку отримання незадовільних результатів для тестової множини, навчання продовжується. Тестування використовується для забезпечення запам'ятовування не лише даних заданої навчальної множини, але і створення загальних образів, що можуть міститись в даних.

4.4. Неконтрольоване навчання

Неконтрольоване навчання (Unsupervised Learning) - це один із підходів до навчання нейронних мереж, при якому модель навчається на даних, що не мають явних міток або цільових змінних. Натомість модель шукає приховані структури або патерни в даних самостійно.

На відміну від контрольованого навчання, де кожен навчальний приклад має відповідну мітку, у неконтрольованому навчанні дані не містять явних міток або цільових змінних. Модель повинна самостійно виявити приховані структури

та патерни у даних. Це може застосовуватися для кластеризації даних, пошуку прихованих змінних або вивчення залежностей між ознаками.

Часто дані у неконтрольованому навчанні мають високу розмірність, тобто містять велику кількість ознак. Це може створювати складнощі в аналізі та обробці даних, а також вимагати застосування методів зниження розмірності. У неконтрольованому навчанні може бути складно інтерпретувати отримані результати. Оскільки немає явних міток, оцінка якості моделі та інтерпретація прихованих структур потребують додаткових методів та експертного аналізу.

Важливим також є проведення попередньої обробки даних, наприклад масштабування, нормалізацію або видалення викидів, щоб забезпечити ефективніше навчання моделі. Неконтрольоване навчання може бути чутливим до якості попередньої обробки даних.

Відносна ефективність неконтрольованого навчання залежить від конкретної задачі та даних. У деяких випадках, наприклад, завдання кластеризації, неконтрольоване навчання може бути дуже ефективним і допоможе виявити приховані групи або кластери в даних. Однак, без явних міток, важко оцінити якість та точність отриманих результатів. Ефективність неконтрольованого навчання значною мірою залежить від правильного вибору моделі, передобробки даних та інтерпретації отриманих результатів.

4.5. Оцінки навчання

Оцінка ефективності навчання нейромережі залежить від кількох керованих факторів. Розглядають три фундаментальні властивості, що пов'язані з навчанням: ємність, складність зразків і обчислювальна складність.

- **Ємність** вказує, скільки зразків може запам'ятати мережа, і які межі прийняття рішень можуть бути на ній сформовані.
- **Складність зразків** визначає число навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення.
- **Обчислювальна складність** напряму пов'язана з потужністю комп’ютера.

5. Завдання, які вирішують за допомогою нейромереж

Нейронні мережі широко застосовуються для вирішення різних завдань у різних галузях.

- **Класифікація**. Нейронні мережі можуть бути використані для класифікації об'єктів на різні класи. Наприклад, розпізнавання зображень, де нейронна мережа може визначати, чи є зображення кішкою чи собакою.
- **Регресія.** Нейронні мережі можна використовувати для передбачення чисельних значень. Наприклад, прогноз ціни нерухомості на основі її характеристик або прогнозування часових рядів.
- **Кластеризація**. Нейронні мережі можуть використовуватися для угруповання об'єктів у кластери на основі їх подібності. Наприклад, кластеризація користувачів соціальних мереж для персоналізації рекомендацій.
- **Обробка природної мови.** Нейронні мережі можуть використовуватись для аналізу та розуміння природної мови. Це може включати завдання, такі як визначення тональності тексту, машинний переклад, системи голосового керування або перетворення аудіо на текст.
- **Голосові помічники та чат-боти**. Нейронні мережі використовуються для розробки голосових помічників і чат-ботів, які можуть розуміти і відповідати на природну мову користувачів, виконувати завдання та надавати інформацію.
- **Рекомендаційні системи**. Нейронні мережі можуть використовуватися для створення персональних рекомендацій для користувачів на основі їх переваг та поведінки. Це може містити рекомендації товарів, фільмів, музики тощо.
- **Обробка зображень та відео**. Нейронні мережі можуть використовуватись для аналізу та обробки зображень та відео. Це може включати завдання, такі як розпізнавання об'єктів на зображеннях, сегментація зображень, анотування відео тощо.
- **Розпізнавання жестів та емоцій**. Нейронні мережі можуть використовуватися для розпізнавання жестів та емоцій на основі відео або зображень. Це може бути корисним, наприклад, для розробки систем розпізнавання жестових команд або систем аналізу емоційного стану.

- **Генерація контенту**. Нейронні мережі можуть використовуватися для створення нового контенту, таких як зображення, тексти, музика та ін. Наприклад, створення автоматичних описів зображень або створення музики.
- **Автоматичне керування.** Нейронні мережі можуть бути використані для розробки систем автоматичного керування. Це може включати керування роботами, автомобілями, дронами чи іншими складними системами.
- **Робототехніка**. Нейронні мережі застосовуються в робототехніці для вирішення задач навігації, розпізнавання об'єктів, планування рухів та взаємодії з навколишнім середовищем. Вони дозволяють роботам навчатися та адаптуватися до нових умов та завдань.

Це лише деякі приклади областей, де нейронні мережі знаходять застосування. З появою нових ідей і технологій, застосування нейронних мереж продовжує розширюватися і включати все більше областей.

Незважаючи на переваги нейронних мереж в певних областях над традиційними обчисленнями, існуючі нейромережі не є досконалими рішеннями. Вони навчаються і можуть робити "помилки". Окрім того, не можна гарантувати, що розроблена мережа є оптимальною мережею.

Застосування нейромереж вимагає від розробника виконання ряду умов:

- Наявність репрезентативної та достатньої за розміром множини даних для навчання й тестування мережі.
- Розуміння базової природи проблеми, яку буде вирішено.
- Вибір методів навчання та відповідна потужність обробки.

6. Доцільність використання нейронних мереж

Переваги перед традиційними обчислювальними методами

Рішення задач в умовах невизначеності. Завдяки здатності до навчання нейронна мережа дозволяє вирішувати завдання з невідомими закономірностями і залежностями між вхідними та вихідними даними, що дозволяє працювати з неповними даними.

Стійкість до шумів у вхідних даних. Нейронна мережа може самостійно виявляти неінформативні для аналізу параметри і видаляти їх, в зв'язку з чим відпадає необхідність у попередньому аналізі вхідних даних.

Гнучкість структури нейронних мереж. Компоненти нейрокомп'ютерів - нейрони і зв'язки між ними - можна комбінувати в різний спосіб. За рахунок цього один нейрокомп'ютер можна застосовувати для вирішення різних завдань, часто не пов'язаних між собою.

Висока швидкодія. Вхідні дані обробляються багатьма нейронами одночасно, завдяки чому нейронні мережі вирішують завдання швидше, ніж більшість інших алгоритмів.

Адаптація до змін навколишнього середовища. Нейронні мережі, навчаючись на даних, здатні підлаштовуватися під змінне навколишнє середовище (наприклад, зміни ситуації на ринку). Якщо необхідно вирішувати якесь завдання в умовах нестаціонарного середовища, то можуть бути створені нейронні мережі, що перенавчаються в режимі реального часу. Чим вище адаптивні здібності системи, тим більш стійкою буде її робота в нестаціонарному середовищі.

Відмовостійкість нейронних мереж. На несприятливу зміну умов нейромережа реагує лише незначним зниженням продуктивності. Ця особливість пояснюється розподіленим характером зберігання інформації в нейронної мережі, тому істотно вплинути на працездатність нейромережі можуть лише серйозні пошкодження структури.

Недоліки нейронних мереж

Відповідь, що надає нейронна мережа, є завжди приблизною. Нейронні мережі не здатні давати точні і однозначні відповіді. Але завдання, в яких треба застосовувати нейромережі і одночасно отримувати точні відповіді, зустрічаються досить рідко.

Нездатність прийняття рішень в кілька етапів. Нейронна мережа не може вирішувати завдання, які вимагають послідовного виконання кількох кроків; вона здатна вирішувати завдання тільки "в один захід". Тому нейромережа не може, наприклад, довести математичну теорему.

Трудомісткість і тривалість навчання. Для того щоб нейронна мережа могла вірно вирішувати поставлені завдання, потрібно провести її навчання на десятках мільйонів наборів вхідних даних. Розроблені різні технології прискореного навчання та сучасні відеокарти дозволяють навчати нейромережі в сотні разів швидше. З'являються готові, попередньо навчені нейромережі, зокрема, що розпізнають образи. На основі таких нейромереж можна створювати додатки, не займаючись тривалим навчанням.

Загалом, нейромережа це машинна інтерпретація мозку людини, в якому знаходяться мільйони нейронів, що передають інформацію у вигляді електричних імпульсів і з'єднані між собою синаптичними зв’язками. Нейронні мережі використовуються для вирішення складних завдань, які вимагають аналітичних обчислень подібно до того, що робить людський мозок.

Структура нейронної мережі прийшла в світ програмування прямо з біології. Завдяки такій структурі, машина отримує можливість аналізувати і обробляти різну інформацію. Нейронні мережі також здатні відтворювати вхідну інформацію зі своєї пам'яті та надавати результати обчислень.

Контрольні запитання:

1. У чому полягає принципова відмінність штучних нейронних мереж від традиційних обчислювальних систем із архітектурою фон Неймана?
2. Які основні властивості штучних нейронних мереж визначають їхню ефективність при розв’язанні інтелектуальних задач?
3. Опишіть структуру штучного нейрона та поясніть роль вагових коефіцієнтів у процесі навчання мережі.
4. Які функції виконують суматор, передатна функція та функція активації в моделі штучного нейрона?
5. Які типи архітектур нейронних мереж існують і чим відрізняються мережі прямого поширення від рекурентних?
6. Як здійснюється процес контрольованого навчання нейронної мережі? Які його основні етапи та мета?
7. Що таке перенавчання нейронної мережі, які його причини та способи запобігання?

8. Які завдання найчастіше розв’язуються за допомогою нейронних мереж у сучасних системах штучного інтелекту?
9. Які переваги мають нейронні мережі над класичними алгоритмічними підходами при роботі з неповними або шумними даними?
10. У чому полягають обмеження та недоліки сучасних нейронних мереж, і які напрями досліджень спрямовані на їх подолання?

ТЕМА 4. ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ ШІ

Штучні нейронні мережі, або нейромережі, є одним з найбільш захоплюючих і динамічних напрямків в галузі інформаційних технологій. За останні кілька десятиліть вони перетворилися з концепції в реальність, створивши неймовірні можливості для розвитку та інновацій в різних галузях.

1. Основні типи моделей генерації зображень.

Сучасні системи штучного інтелекту здатні генерувати реалістичні зображення на основі навчання на великих наборах даних. В основі таких систем лежать різні архітектури генеративних моделей. Найбільш відомими типами є:

- Variational Autoencoders (VAE) – варіаційні автокодери;
- Generative Adversarial Networks (GAN) – генеративні змагальні мережі;
- Дифузійні моделі (Diffusion Models) – моделі послідовного шумування/денойзингу.

Кожна з цих архітектур має свої особливості, переваги та недоліки. *VAE*-моделі забезпечують високу різноманітність згенерованих зображень, але часто страждають на розмитість деталей. *GAN*-мережі відомі здатністю створювати дуже фотореалістичні зображення, однак їх складно тренувати і вони можуть охоплювати не всю різноманітність даних. *Дифузійні моделі* поєднують високу якість і різноманітність генерації, але потребують багато обчислень і часу для генерації одного зображення. Таким чином, вибір моделі залежить від конкретних вимог задачі – чи важливіше отримати максимальну реалістичність, чи різноманітність, чи швидкість генерації.

Variational Autoencoders (VAE)

Варіаційний автокодер (VAE) – це тип нейронної мережі, який поєднує властивості автокодера та методів варіаційного байєсівського виведення. Модель VAE складається з двох основних частин: енкодера та декодера. Енкодер стискає вхідне зображення у компактне подання – латентний простір, і замість того щоб кодувати його в єдину точку, він оцінює параметри розподілу (зазвичай багатовимірного нормального) для цього коду. Таким чином, кожне зображення зі складного датасету відображається не в одну точку латентного простору, а в розподіл, що дозволяє уникнути перенавчання та забезпечує безперервність латентного простору. Декодер виконує обернену функцію – генерує зображення на основі вибірки зі знайденого енкодером латентного розподілу, відновлюючи його до початкового простору пікселів.

Навчання VAE відбувається шляхом одночасної мінімізації двох складових функції втрат: *помилки реконструкції* (наскільки добре декодоване зображення відновлює оригінал) та *KL-дивергенції* (наскільки розподіл

латентного коду наближений до заданого апріорного). Для ефективного навчання застосовується прийом reparametrization trick – вибірка з латентного розподілу здійснюється через детерміноване перетворення випадкового шуму, що дозволяє виконувати диференціювання під час backpropagation. Завдяки цьому VAE навчається відновлювати дані і водночас тримає компактний латентний простір, придатний для генерації нових семплів.

Генерація зображень із VAE. Після навчання модель може генерувати нові приклади шляхом вибірки випадкового вектора з латентного простору (з використанням розподілу, що його навчилася моделювати мережа) та пропускання його через декодер. Завдяки безперервності латентного простору такі вибірки дають нові зображення, схожі на навчальні, але не ідентичні їм. VAE забезпечують різноманітність результатів, бо вони оптимізовані на покриття всієї статистичної розмаїтості даних. Однак якість і різкість зображень VAE нижча, ніж у інших підходів: типово згенеровані зображення можуть бути трохи розмитими або спрощеними. Це спричинено тим, що VAE оптимізують *піксельну* помилку відновлення, яка за природою усереднює варіанти, і тому дрібні деталі та високочастотні структури розмиваються.

Історія та значення. Архітектура VAE була вперше запропонована у 2013 році (Diederik P. Kingma, Max Welling) як генеративна модель для ненаглядуваного навчання. VAE стали важливим кроком уперед, довівши можливість навчити нейронну мережу генерувати дані через латентний простір. Попри те, що пізніші моделі (як-от GAN та дифузійні) перевершили VAE у фотореалістичності, варіаційні автокодери заклали основу для подальших досліджень і досі застосовуються там, де потрібна інтерпретованість латентного простору чи поєднання генеративних моделей з енкодерами (наприклад, гібридні моделі VAE-GAN).

Generative Adversarial Networks (GAN)

Генеративно-змагальна мережа (GAN) – це підхід, в якому дві нейронні мережі змагаються одна з одною у рамках *нульової суми*. Концепцію GAN запропонував Ян Гудфелоу у 2014 році, що викликало справжній бум у галузі генерації зображень. Архітектура GAN складається з двох компонентів: генератора та дискримінатора. Генератор бере на вхід випадковий шумовий вектор (зазвичай зі стандартного нормального розподілу) і перетворює його на зображення, прагнучи зробити його максимально схожим на реальні приклади. Дискримінатор натомість отримує на вхід або справжнє зображення з тренувального набору, або зображення, згенероване генератором, і має навчитися відрізняти справжнє від підробки. По суті, генератор виступає в ролі «фальшивомонетника», а дискримінатор – «експерта», що розпізнає підробки.

Навчання GAN організоване як гра: генератор намагається обдурити дискримінатор, а дискримінатор – не датися обдурити. Формально вони оптимізують протилежні цільові функції (генератор мінімізує ймовірність того, що дискримінатор правильно віднесе його зображення до «підробок», а

дискримінатор – максимізує цю ймовірність). Така конкуренція призводить до того, що генератор поступово вчиться створювати все реалістичніші зображення, наближаючись за розподілом до даних, на яких його тренують. Коли GAN успішно навчився, згенеровані ним зображення можуть бути настільки правдоподібними, що людина не відрізнить їх від справжніх. Це зробило GAN дуже популярними для задач на кшталт генерування фотографій людей, які ніколи не існували (наприклад, знаменитий проект *This Person Does Not Exist*).

Проблеми тренування GAN. На практиці навчання GAN є складним – баланс між генератором і дискримінатором легко порушується. Типові проблеми:
- *Зникання градієнта:* якщо дискримінатор занадто сильний, він майже безпомилково відрізняє фейки, і генератор не отримує корисного сигналу для навчання. Це блокує покращення генератора. Вирішенням є, наприклад, використання альтернативних функцій втрат – Wasserstein-GAN з відповідним метрикою та штрафами, що забезпечує ненульові градієнти навіть при сильному дискримінаторі.
- *Колапс моди:* генератор може навчитися випускати дуже обмежений піднабір образів, які добре обдурюють дискримінатор, і ігнорувати різноманітність справжніх даних. В такому разі мережа буде генерувати одноманітні зображення (одну «моду»), наприклад, лише обличчя одного типу, незважаючи на широкий діапазон облич у тренувальних даних. Для боротьби з цим пропонуються методи, що стимулюють генератор до різноманітності: *minibatch discrimination*, *unrolled GAN* та знову ж таки підхід WGAN, що покращує стійкість навчання.
- *Нестабільність і незіходження:* гра між двома мережами може не сходитися – параметри починають коливатися, не наближаючись до рівноваги. Дослідники застосовують різні прийоми регуляризації (наприклад, додавання випадкового шуму до входів дискримінатора, штрафи на великі ваги дискримінатора) для підвищення стабільності навчання.

Попри труднощі навчання, GAN залишаються одними з найуспішніших генеративних моделей для зображень. Вони здатні напряму видавати зображення за один прохід генератора (без ітерацій), що робить генерацію дуже швидкою у порівнянні з дифузійними моделями. Результати GAN-моделей вражають фотореалістичністю дрібних деталей – вони генерують чіткі обличчя, текстури, складні об’єкти, перевершуючи ранні VAE за якістю. Сфери застосування включають *генерацію портретів*, *пейзажів*, *штифтів* (pin art), *покращення роздільної здатності зображень* (через SRGAN) тощо.

Дифузійні моделі.

Дифузійні моделі – це відносно новий клас генеративних моделей, що привернув увагу дослідників починаючи з 2015 року. Ідея дифузійної (diffusion) моделі полягає в тому, щоб навчитися послідовно перетворювати розподіл випадкового шуму на розподіл реальних даних. Процес складається з двох напрямків: прямий дифузійний процес (forward diffusion) та обернений процес.

Прямий процес поступово додає шум до справжніх зображень – крок за кроком пікселі змінюються випадково, аж доки зображення не перетвориться на чистий шум. Таким чином модель задає *марковський ланцюг* деградації даних. Обернений процес – це те, що моделює нейронна мережа: починаючи з випадкового шуму, вона в кілька кроків навчається прибирати шум, рухаючись у протилежному напрямку до вихідного розподілу даних. Мережа навчається денойзити – на кожному кроці трохи очищувати зображення від шуму, спираючись на патерни, засвоєні з тренувальних даних. Після достатньої кількості кроків (декілька сотень або десятків) із чистого шуму поступово формується осмислене зображення. Цей процес можна порівняти з *проявленням фотографії*: тільки замість хімічного проявника тут алгоритм, що крок за кроком «проявляє» зображення з хаосу шуму.

Навчання дифузійної моделі відбувається шляхом навчання нейронної мережі (часто архітектури *U-Net*) *передбачати або прибирати шум* на кожному кроці. Використовується спрощений обчислювально зручний підхід, де модель тренують на завданні передбачення початкового зображення або шуму з зашумленої версії за один крок. Завдяки цьому після навчання модель здатна *ітеративно застосовуватися кілька разів*, відновлюючи все більше сигналу і зменшуючи шум. Високофіделітні результати дифузійних моделей пояснюються тим, що вони оптимізують ймовірність даних напряму (через максимізацію правдоподібності шляхом оберненого процесу), а не намагаються виграти змагання як GAN. Це дозволяє уникнути проблеми колапсу мод і дає чудове покриття різноманітності даних. Практично це означає, що дифузійні моделі можуть генерувати дуже різноманітні зображення і водночас з високою деталізацією та реалістичністю – вони поєднують плюси VAE (різноманітність) та GAN (якість).

Недоліком такого підходу є низька швидкість генерації. Оскільки потрібна серія з десятків або сотень ітерацій денойзингу, генерація одного зображення займає значно більше часу, ніж один прохід GAN. Наприклад, якщо GAN може згенерувати зображення за мілісекунди, то дифузійна модель – за секунди (залежить від кількості кроків). Для подолання цього розробляють методи прискорення: зменшення кількості кроків (шляхом більш агресивного денойзингу на крок або використання диференційних методів розв’язання), а також використовують потужні апаратні прискорювачі. Проте на 2023–2025 роки дифузійні моделі стали основою багатьох найкращих систем генерації зображень саме завдяки якості результату, прийнятно жертвуючи часом генерації.

2. Stable Diffusion.

Stable Diffusion – це знакова генеративна модель, що втілює потужність дифузійного підходу та є однією з найбільш популярних відкритих систем текст-ту-зображення. Випущена у серпні 2022 року компанією Stability AI спільно з дослідниками з CompVis (Мюнхенський ун-т) та RunwayML, вона стала

доступною спільноті як відкритий проект з відкритим кодом і моделями. Stable Diffusion є латентною дифузійною моделлю – тобто дифузійний процес виконується не в просторі пікселів, а в стислому латентному просторі ознак, який отримують за допомогою VAE-енкодера. Це дозволило різко знизити вимоги до обчислень: модель може працювати на звичайному споживацькому GPU з ~8 ГБ пам’яті (а оптимізовані версії – навіть з ~2.4 ГБ VRAM), генеруючи зображення досить швидко.

3. Нейромережі MIDJORNEY, LEONARDO, DALL-E, IMAGEFX.

Нейромережа Midjourney: Міджорні - це перспективна нейромережа, призначена для обробки зображень та графіки. Вона пропонує широкий спектр можливостей, починаючи від редакції та відновлення зображень до створення графічних ефектів і фільтрів. Міджорні дозволяє автоматизувати процеси обробки зображень, полегшуючи завдання графічних дизайнерів та фотографів.

Leonardo.ai: Leonardo.ai - це платформа, яка користується нейромережами та штучним інтелектом для створення різноманітних графічних рішень. Вона надає інструменти для створення арт-робіт, дизайну логотипів, обробки зображень та багато іншого. Leonardo.ai дозволяє користувачам швидко та ефективно створювати вражаючі графічні елементи за допомогою штучного інтелекту.

DALL-E 3, ImageFX: DALL-E 3 та ImageFX - це інноваційні нейромережі, призначені для генерації зображень з текстового опису та застосування різноманітних графічних ефектів до зображень відповідно. Вони здатні створювати складні та реалістичні графічні об'єкти за допомогою алгоритмів штучного інтелекту.

4. Робота нейромереж на практиці та порівняння.

Для проведення аналізу ефективності нейромереж у вдосконаленні індивідуалізованого навчання студентів був використаний невеликий дослід, що включав порівняння різних нейромереж за результатами їхньої роботи у відповідь на певний запит. Цей дослід став підґрунтям для аналізу ефективності та можливостей нейромереж у контексті навчального процесу.

Під час експерименту було використано наступний промт, згенерований у чаті з мовною моделлю штучного розуму GPT:

"Створити високоякісне, реалістичне зображення Лесі Українки в традиційній українській вишиванці, відображаючи сучасний дух і культурну ідентичність. Вона повинна стояти на оживленому місці в сучасній Україні, на фоні Хрещатика. На задньому плані повинен бути видимий напис “Україна”,

вбудований у міський пейзаж, підкреслюючи зв'язок між фігурою та місцем. Кожна деталь повинна бути ретельно відтворена."

На рисунку 1 показано нейромережу від Google, а саме ImageFX.

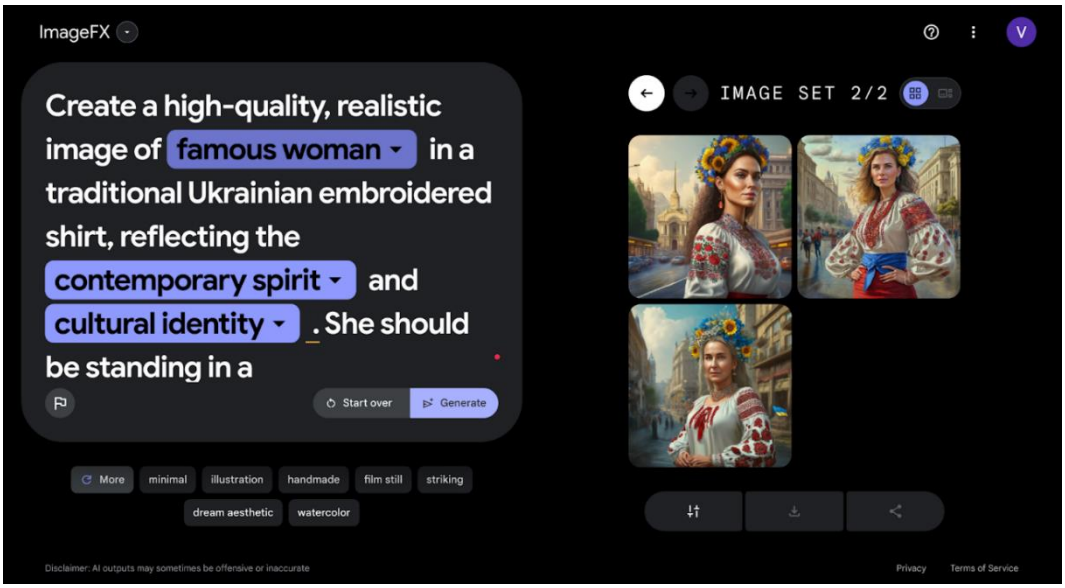


Рис. 1: Нейромережуа від Google, а саме ImageFX

ImageFX, нейромережа від Google, надала досить красиві зображення з хорошою деталізацією людини, включаючи Лесю Українку в українській вишиванці. Проте, виникла проблема з чіткістю фону та генерацією тексту на задньому плані, що ускладнило створення реалістичного зображення.

На рис. 2 показано нейромережу від Adobe Firefly

Рис.2: Нейромережа від Adobe Firefly

Firefly, нейромережа від Adobe, також надала якісні зображення, але не завжди вдалося згенерувати фон міста та текст, що відчутно вплинуло на реалістичність створеного зображення.

На рисунку 3 показано Нейромережа leonardo

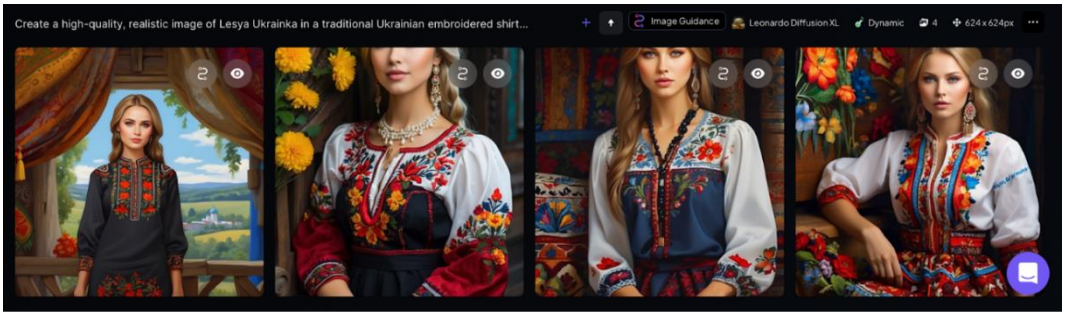


Рис.3: Нейромережа від leonardo

Нейромережа Leonardo не забезпечила очікувані результати, що може бути пов'язано з її технічними особливостями та обмеженнями.

На рисунку 4 показано Нейромережа Midjorney

Рис.4: Нейромережа від Midjorney

Midjorney також не відповіла очікуванням, що свідчить про ймовірні обмеження цієї нейромережі в контексті створення реалістичних зображень.

На рисунку 5 показано Нейромережа Copilot на платформі DALL·E 3

Рис.5: Нейромережа від Copilot на платформі DALL·E 3

Нейромережа Copilot на платформі DALL·E 3 продемонструвала чудові результати у генерації зображень та тексту, що зробило її однією з найуспішніших серед випробуваних нейромереж.

Отримані результати свідчать про те, що нейромережі можуть бути ефективним інструментом у вдосконаленні індивідуалізованого навчання, проте вони мають свої обмеження. Деякі нейромережі, такі як Copilot на платформі DALL·E 3, показали добрі результати у генерації зображень та тексту, але навіть вони не є ідеальними і вимагають подальшого вдосконалення.

	Image FX	Adobe Firefly	Leonardo	Midjourney	DALL-E 3
Representation of Lesya Ukrainka's face	+	+	+	+	+
Embroidery details	+	+	+	+	+
Clarity of the background	-	-	-	+	+
Generation of text in the background	-	-	-	-	+

Рис. 6 – Підсумкове порівняння платформ за критеріями якості.

Нейромережі мають кілька переваг, які роблять їх потужним інструментом у сфері штучного інтелекту. По-перше, вони відомі своєю високою точністю. Це означає, що вони здатні розв'язувати складні завдання з великою точністю і ефективністю. Наприклад, нейромережі успішно використовуються для розпізнавання образів, класифікації тексту та аналізу даних.

По-друге, нейромережі можуть автоматизувати багато процесів, що раніше вимагали б значної людської участі. Вони можуть виконувати рутинні завдання, такі як обробка даних, редакція зображень або синтез мовлення, забезпечуючи при цьому швидкість та ефективність.

Крім того, нейромережі можуть навчатися на великій кількості даних, а це дозволяє їм ставати все кращими з часом. Вони можуть адаптуватися до нових умов та завдань, що робить їх універсальними інструментами для різних галузей та областей діяльності.

Також важливо відзначити гнучкість нейромереж. Вони можуть застосовуватися до різних завдань, від простих до складних, та адаптуватися до різних типів даних. Це робить їх корисними для широкого спектру застосувань від медицини до фінансів і від реклами до мистецтва.

Незважаючи на свої переваги, нейромережі також мають кілька недоліків. Наприклад, вони потребують великої кількості даних для навчання, і недостатня кількість даних може призвести до низької точності моделі. Крім того, навчання нейромереж може бути часомістким і вимагати значних обчислювальних ресурсів.

Ще одним недоліком є складність інтерпретації результатів. Нейромережі часто вважаються "чорними коробками", оскільки їхні рішення можуть бути складними для розуміння і пояснення. Це може ускладнювати процеси прийняття рішень та впровадження моделей у практику.

Крім того, з мірою збільшення складності нейромереж збільшується і потреба в обчислювальних ресурсах. Великі та складні нейромережі можуть вимагати значних обчислювальних потужностей для навчання та використання, що може бути важким для багатьох організацій та дослідників.

Загалом, розглянуто різні типи нейромереж, включаючи глибокі нейронні мережі, зглибшися у їхню структуру та принципи роботи. Також розглянуто популярні архітектури нейромереж і їхнє застосування у вирішенні різних завдань.

Крім того, обговорено переваги і недоліки використання нейромереж. Вони можуть бути потужним інструментом у руках кваліфікованих фахівців, але водночас вимагають уважного підходу до навчання, використання і валідації моделей.

Наприкінці, важливо підкреслити, що нейромережі мають великий потенціал у багатьох галузях, і вони продовжують розвиватися та вдосконалюватися з кожним днем. Вони можуть змінити спосіб, яким ми працюємо, навчаємося і спілкуємося, і важливо продовжувати досліджувати їхні можливості для максимального використання в майбутньому.

5. Ризики та етичні питання.

Як і будь-яка потужна технологія, генеративні моделі несуть не лише користь, а й виклики. Ми розглянемо три основні аспекти: авторське право, етичні проблеми (включно з упередженнями) та дезінформацію/deepfakes.

Вже є реальні випадки, коли генеровані зображення спричинили суперечки і навіть судові позови. Суспільство реагує по-різному: хтось захоплюється можливостями, а хтось б'є тривогу. Наша задача – зважити ці ризики, навести приклади і подивитися, які є пропозиції вирішення.

Спершу загальне питання: Хто володіє згенерованим зображенням? Чиє авторське право – якщо картинку фактично "намалювала" нейромережа, навчена на мільйонах чужих зображень? Це поки що сіра зона в законодавстві. У деяких юрисдикціях (наприклад, США) результат, створений не людиною, не підлягає захисту авторським правом – є прецеденти, коли реєстрацію на AI-арт відхиляли, мовляв, нема людського автора. В інших випадках автором вважають того, хто запустив процес (людина, що дала prompt). Але є й зустрічне: самі творці оригінальних даних вимагають захистити свої права, якщо їхні роботи використали для навчання без дозволу.

Генеративні моделі часто навчаються на величезних датасетах, зібраних із мережі. Наприклад, Stable Diffusion тренували на LAION-5B – це 5 мільярдів пар "картинка+текст" з інтернету. Туди потрапили і фотографії, і малюнки, і ілюстрації, багато з яких захищені авторським правом (скажімо, роботи сучасних художників, фото з Getty Images тощо). Модель під час навчання "увібрала" в себе статистичні особливості цих зображень. Вона не зберігає їх 1-в-1, але може видавати щось дуже схоже на оригінал або стилізоване під конкретного художника. Художники забили на сполох: «Наші роботи використали без згоди, навчили на них AI, який тепер імітує наш стиль – це порушення наших прав.»

Загалом, OpenAI теж обмежує: DALL-E 2 не генерує облич відомих персон, не видає політичні контенти певні. Критики кажуть, що ці фільтри занадто грубі: наприклад, Midjourney довго забороняв слово "Arabic" чи "Islamic", бо боявся образити релігію – і люди не могли нічого в цьому стилі робити. Зараз тюниться. Тут складно знайти межу між цензурою і відповідальністю. Одні вимагають максимальної свободи: мовляв, інструмент як кисть, малюй що хочеш, відповідальність на людині. Інші – що це як зброя, потрібні замки і контроль, щоб не наробили зла.

Отже, генерація зображень ШІ – це потужний інструмент, що демократизує творчість, але одночасно кидає виклик нашим уявленням про авторство, правду і художню цінність. Як кожна нова технологія, вона не добра і не зла сама по собі – все залежить від того, в чиїх руках і з яким наміром. Наша з вами задача – навчитися використовувати її на благо: як засіб натхнення, економії часу, навчання, збереження спадщини, – і запобігати її зловживанням.

1. Контрольні запитання:

1. Які основні архітектури генеративних моделей використовуються для створення зображень (VAE, GAN, Diffusion Models) і в чому полягають їхні принципові відмінності?
2. Як реалізується процес навчання у варіаційних автокодерах (VAE), і яку роль відіграє латентний простір у генерації зображень?
3. Який принцип змагальної взаємодії між генератором і дискримінатором лежить в основі генеративно-змагальних мереж (GAN)?
4. Які основні проблеми виникають під час навчання GAN і які підходи застосовуються для їхнього подолання (наприклад, Wasserstein-GAN, minibatch discrimination)?
5. У чому полягає сутність дифузійних моделей і як відбувається процес послідовного шумування та відновлення зображень?
6. Як поєднання латентного простору та дифузійного процесу в моделі **Stable Diffusion** дозволяє зменшити обчислювальні витрати при генерації?
7. Які сильні та слабкі сторони мають популярні генеративні платформи **DALL·E 3, Midjourney, Leonardo, Firefly, ImageFX** у створенні реалістичних зображень?
8. Які критерії (деталізація обличчя, чіткість фону, генерація тексту, кольорова палітра) використовуються для оцінки якості роботи нейромереж при генерації зображень?
9. Які основні етичні та правові питання виникають під час використання генеративних моделей, зокрема щодо авторських прав і deepfake-контенту?
10. Яким чином генеративні нейромережі можуть бути інтегровані в навчальний процес і використані для розвитку креативного мислення студентів?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Штучний інтелект. Нейромережева обробка інформації : архітектури, навчання, застосування : навчальний посібник у 2-х ч. : Ч. 1 / О. Г. Руденко, О. О. Безсонов, С. П. Євсеєв, О. Б. Ахієзер, Ю. І. Зайцев ; за заг. ред. С. П. Євсеєва. – Харків : НТУ «ХПІ», – Львів : «Новий Світ-2000», 2025. – 426 с. – (Серія «Кібербезпека та штучний інтелект»). ISBN 978-966-418-517-9
2. Штучний інтелект. Нейромережева обробка інформації : архітектури, навчання, застосування : навчальний посібник у 2-х ч. : Ч. 2 / О. Г. Руденко, О. О. Безсонов, С. П. Євсеєв, О. Б. Ахієзер, Ю. І. Зайцев ; за заг. ред. С. П. Євсеєва. – Харків : НТУ «ХПІ», – Львів : «Новий Світ-2000», 2025. – 376 с. – (Серія «Кібербезпека та штучний інтелект»). ISBN 978-966-418-518-6
3. Глибовець М.М., Олецький О.В. Штучний інтелект. Підручник для студентів вищих навчальних закладів, які навчаються за спеціальностями "Комп'ютерні науки" та "Прикладна математика". - К.:Вид.дім "КМ Академія", 2002. - 366 с.
4. Глибовець М.М., Кравченко І.В., Олецький О.В., Терещенко В.М. Програмування в Пролозі. Навчальний посібник. - К.: ВЦ "Київський університет", 1998. - 110 с.
5. Гладун А. Я., Рагушина Ю. В. Data Mining: пошук знань в даних. К.: ТОВ «ВД «АДЕФ Україна», 2016. 452 с.
6. Литвин В. В., Пасічник В. В., Нікольський Ю. В. Аналіз даних та знань : навчальний посібник. Львів: «Магнолія 2006», 2015. 276 с.
7. Снитюк В. Є. Прогнозування. Моделі. Методи. Алгоритми : навчальний посібник. К.: Маклаут, 2008. 364 с.
8. Литвин В.В., Пасічник В.В., Яцишин Ю.В. Інтелектуальні системи : підручник. Львів: Новий світ – 2000, 2009. 406с.
9. Математичне забезпечення інформаційно-керуючих систем: підручник / А. М. Гуржій, З. В. Дудар, В. М. Левикін, Б. В. Шамша. Х. : Компанія Сміт, 2006. 448с.
10. Haykin, S., 2009 “Neural Networks and Learning Machines”, Third Edition, McMaster University, Hamilton, Ontario, Canada, 938
11. Alpayndin, E. (2010). Introduction to machine learning. The Knowledge Engineering Review, 25(3), 353–353.
12. Yasniy, O., Mytnyk, M., Maruschak, P., Mykytyshyn, A., & Didych, I. (2024). Machine learning methods as applied to modelling thermal conductivity of epoxy-based composites with different fillers for aircraft. Aviation, 28(2), 64-71.
13. Yasniy, O., Menou, A., Mykytyshyn, A., Kubashok, V., Didych, I. (2024). Application of neural network platforms for text-based image generation. CEUR Workshop Proceedings, 3842, pp. 232–240.

14. Tymoshchuk, D., Yasniy, O., Maruschak, P., Iasnii, V., & Didych, I. (2024). Loading Frequency Classification in Shape Memory Alloys: A Machine Learning Approach. Computers, 13(12), 339.
15. Iryna, D., Andrii, M., Andrii, S., & Mykola, M. (2024). Application of machine learning methods to the prediction of NO2 concentration in the air environment. In Ceur Workshop Proceedings Open source preview, 2024, 3896, pp. 569–577.
16. Iryna Didych, Oleh Yasniy. (2025). Structural integrity & Lifetime estimation by machine learning methods. Important structural elements. LAP Lambert Academic Publishing, Republik of Moldova, Europe. 124 p.
17. Yasniy, O., Maruschak, P., Mykytyshyn, A., Didych, I., & Tymoshchuk, D. (2025). Artificial intelligence as applied to classifying epoxy composites for aircraft. Aviation, 29(1), 22-29.
18. Tymoshchuk, D., Didych, I., Maruschak, P., Yasniy, O., Mykytyshyn, A., & Mytnyk, M. (2025). Machine Learning Approaches for Classification of Composite Materials. Modelling, 6(4), 118.

ДЛЯ НОТАТОК