

Міністерство освіти і науки України
Вищий державний навчальний заклад
«Приазовський державний технічний університет»
Біомедичної інженерії

Верескун М.В.

Конспект лекцій
з дисципліни «Технології програмування».

Частина 2. «Навчання машин та штучний інтелект»

для студентів напряму підготовки 163 «Біомедична інженерія»
всіх форм навчання



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

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

Маріуполь
2019

УДК 519.7:007.52(075)

Навчання машин та штучний інтелект: конспект лекцій з дисципліни «Технології програмування» ч. 2 «Навчання машин та штучний інтелект» для студентів напряму підготовки 163 «Біомедична інженерія» всіх форм навчання / уклад. Верескун М.В.. – Маріуполь : ДВНЗ «ПДТУ», 2019. – 84 с.

Конспект лекцій містить матеріал із загальними положеннями з навчання машин та теорії штучного інтелекту, представлення знань та еволюційного моделювання.

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

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

Рекомендовано
на засіданні кафедри вищої та прикладної математики,
протокол № 21 від 24 червня 2019 р.

Затверджено
методичною комісією
факультету інформаційних технологій
протокол № 10 від 24 червня 2019 р.

© ДВНЗ «ПДТУ», 2019

© М.В. Верескун, 2019

ЗМІСТ

ВСТУП.....	5
Лекція 1. НАПРЯМИ ТА ПІДХОДИ ДО ДОСЛІДЖЕНЬ В ОБЛАСТІ ШТУЧНОГО ІНТЕЛЕКТУ.....	6
1.1 Основні підходи до дослідження штучного інтелекту.....	7
1.2 Основні напрямки досліджень в області штучного інтелекту.....	9
1.3 Визначення інтелектуальної інформаційної системи.....	15
Лекція 2. ОСНОВИ ТЕОРІЇ ШТУЧНОГО ІНТЕЛЕКТУ. ПОДАННЯ ЗНАНЬ.....	17
2.1 Дані і знання.....	17
2.2 Класифікація моделей подання знань.....	23
2.2.1 Логіко-алгебраїчні моделі подання знань.....	25
2.2.2 Продукційні моделі подання знань.....	28
2.2.3 Семантичні мережі.....	29
2.2.4 Фрейми.....	31
Лекція 3. НЕЙРОННІ МЕРЕЖІ.....	35
3.1 Класифікація штучних нейронних мереж.....	39
3.2 Одношарові штучні нейронні мережі.....	41
3.3 Багатошарові нейронні мережі.....	43
3.4 Завдання, які вирішуються нейронними мережами.....	47
Лекція 4. ЕВОЛЮЦІЙНЕ МОДЕЛЮВАННЯ.....	49
4.1 Генетичні алгоритми.....	52
4.2. Схема функціонування генетичного алгоритму.....	53
4.3 Види генетичних алгоритмів.....	61
Контрольні питання.....	65
Лекція 5. ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ. ЕКСПЕРТНІ СИСТЕМИ.....	66
5.1 Модель експертних систем.....	69
5.2 Класифікація експертних систем і оболонок експертних систем.....	73
5.3 Засоби розробки експертних систем.....	76

ВИСНОВОК.....	81
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	82

ВСТУП

В даний час не існує єдиного і визнаного всіма точного і всеосяжного визначення передового наукового напрямку, званого «штучний інтелект» (ШІ), як, втім, немає і універсального визначення людського інтелекту.

Штучний інтелект це: 1) наука і технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм; 2) властивість інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини.

Серед багатьох точок зору на штучний інтелект домінують три. Згідно з першою, дослідження в галузі ШІ є фундаментальними дослідженнями, в рамках яких розробляються моделі і методи розв'язання задач, що традиційно вважаються інтелектуальними і не піддаються раніше формалізації і автоматизації. Відповідно до другої точки зору новий напрямок пов'язано з новими ідеями вирішення завдань на комп'ютерах, з розробкою принциповіших технологій програмування і створенням нових архітектур, що відкидають класичну архітектуру, висхідну ще до перших розробок обчислювальної техніки. Нарешті, третя точка зору, мабуть, найбільш прагматична, полягає в тому, що в результаті робіт в області ШІ народжується безліч прикладних систем, що вирішують завдання, для яких раніше створювані системи були непридатні.

Завданням штучного інтелекту як науки є відтворення за допомогою штучних пристроїв (в основному ноутбуків, планшетів, смартфонів, носяться пристроїв, таких як годинник, брелоки та ін., А також робототехнічних пристроїв: домашніх роботів, безпілотних автомобілів і ін.) Розумних дій і міркувань. Вивчення теоретичних основ даної науки дозволяє відшукувати найбільш доцільні шляхи розробки нових інформаційних технологій і вирішення перспективних інтелектуальних завдань.

Лекція 1.

НАПРЯМИ ТА ПІДХОДИ ДО ДОСЛІДЖЕНЬ В ОБЛАСТІ ШТУЧНОГО ІНТЕЛЕКТУ

План

- 1.1 Основні підходи до дослідження штучного інтелекту
- 1.2 Основні напрямки досліджень в області штучного інтелекту
- 1.3 Визначення інтелектуальної інформаційної системи
- 1.4 Класифікація інтелектуальних систем

Термін «штучний інтелект» (artificial intelligence) був запропонований в 1956 р. Слово intelligence означає «уміння міркувати розумно», а зовсім не «інтелект», для якого є термін intellect.

Штучний інтелект займається вивченням розумної поведінки (у людей, тварин і машин) і спроба знайти способи моделювання подібної поведінки в будь-якому типі штучно створеного механізму. Незважаючи на те, що терміну більше півстоліття єдиного визначення не існує. Різні дослідники по-різному визначають цю науку, в залежності від свого погляду на неї і працюють над створенням систем, які

- думають подібно людям;
- думають раціонально;
- діють подібно людям;
- діють раціонально.

Синтезуючи десятки визначень штучного інтелекту з різних джерел як робоче визначення можна запропонувати наступне.

Штучний інтелект – це один з напрямків інформатики, метою якого є розробка апаратно-програмних засобів, що дозволяють користувачеві-непрограмістів ставити і вирішувати свої, що традиційно вважаються інтелектуальними завдання, спілкуючись з ЕОМ на обмеженій підмножині природної мови.

При відтворенні розумних міркувань і дій виникають певні труднощі. По-пір'яних в більшості випадків, виконуючи якісь дії, людина не усвідомлює, як це робить, ніхто не знає точний

спосіб, метод або алгоритм розуміння тексту, розпізнавання осіб, доказів теорем, рішення задач, твори віршів і т. д. По-друге, на сучасному рівні розвитку комп'ютер занадто далекий від людського рівня компетентності і працює за іншими принципами.

Штучний інтелект завжди був міждисциплінарною наукою, будучи одночасно і наукою і мистецтвом, і технікою і психологією. Методи штучного інтелекту різноманітні. Вони активно запозичуються з інших наук, адаптуються і змінюються під решаему завдання. Для створення інтелектуальної системи необхідно залучати фахівців з прикладної області, в рамках штучного інтелекту співпрацюють лінгвісти, нейрофізіологи, психологи, економісти, інформатики, програмісти і т. д.

1.1 Основні підходи до дослідження штучного інтелекту

Незабаром після визнання штучного інтелекту окремою галуззю науки відбувся поділ його на два напрямки: нейрокібернетика і кібернетика чорного ящика. Ці напрямки розвиваються практично незалежно, істотно розрізняючись як в методології, так і в технології.

Нейрокібернетики взяли за основу структуру і принципи функціонування єдиного створеного природою пристрою, здатного міркувати – мозку. Клітини мозку називаються нейронами, звідси і назва напрямку. Дослідники вважали, що, змодельовавши мозок, зможуть відтворити і його роботу.

Дослідники напрямки кібернетика чорного ящика дотримувалися думки, що не важливо за якими принципами працює пристрій, які засоби і методи лежать в його основі, головне, імітувати функції мозку, навіть якщо крім результату це не матиме нічого спільного з природним розумом.

Нині стали помітні тенденції до об'єднання цих частин знову в єдине ціле. Стало з'являтися безліч гібридних методів і систем, наприклад, експертна система на базі нейронної мережі або нейронна мережа, навчається генетичним алгоритмом.

Дослідники, що моделюють лише окремі функції інтелекту, наприклад, розпізнавання образів, синтез мови, прийняття рішень, працюють у рамках напряму слабкий штучний інтелект. Спроби відтворити роботу інтелекту в повному обсязі відносяться до напрямку сильний штучний інтелект. Всі основні досягнення в області штучного інтелекту відносяться до слабого штучного інтелекту.

Крім цього, виділяють спадний (семіотичний) і висхідний (біологічний) підходи.

Спадний підхід передбачає моделювання високорівневих психічних процесів, таких як мислення, мова, емоції і т. д.

Висхідний підхід досліджує інтелектуальну поведінку систем на базі більш дрібних «неінтелектуальних» елементів. Нейронні мережі та еволюційне моделювання відносяться до цього підходу.

Інтелектуальні системи розробляються із залученням різних засобів і методів. Існує чотири основні підходи до їх побудови: логічний, структурний, еволюційний і імітаційний.

Основою для даного логічного підходу служить Булева алгебра. Така інтелектуальна система являє собою машину доведення теорем. При цьому вихідні дані зберігаються в базі даних у вигляді аксіом, правила логічного висновку як відносини між ними. Крім того, кожна така машина має блок генерації цілі, і система виведення намагається довести дану мету як теорему. Якщо мета доведена, то трасування застосованих правил дозволяє отримати ланцюжок дій, необхідних для реалізації поставленої мети. Потужність такої системи визначається можливостями генератора цілей і машиною доведення теорем. Для більшості логічних методів характерна велика трудомісткість, оскільки під час пошуку докази можливий повний перебір варіантів. Тому даний підхід вимагає ефективної реалізації обчислювального процесу,

Під структурним підходом мається на увазі спроби побудови інтелектуальної системи шляхом моделювання структури людського мозку, тобто системи, побудовані в рамках напряму нейрокібернетика.

При побудові інтелектуальної системи за допомогою еволюційного підходу основна увага приділяється побудові початкової моделі, і правилам, за якими вона може змінюватися (еволюціонувати). Причому модель може бути складена з найрізноманітніших методів, це може бути і нейронна мережа і набір логічних правил і будь-яка інша модель. На підставі перевірки моделей відбирає найкращі з них, на підставі яких по самим різним правилам генеруються нові моделі, з яких знову вибираються найкращі і т. Д.

Імітаційний підхід використовується в рамках напряму кібернетика чорного ящика. Інтелектуальні системи при такому підході повинні моделювати якусь інтелектуальну функцію, тобто встановити необхідне відповідність між входами і виходами системи.

1.2 Основні напрямки досліджень в області штучного інтелекту

Тематика штучного інтелекту охоплює величезний перелік наукових напрямків, починаючи з таких завдань загального характеру, як навчання і сприйняття, і закінчуючи такими спеціальними завданнями, як гра в шахи, доказ математичних теорем, твір поетичних творів і діагностика захворювань. У штучному інтелекті систематизуються і автоматизуються інтелектуальні завдання і тому ця область стосується будь-якої сфери інтелектуальної діяльності людини.

Серед безлічі напрямків штучного інтелекту є кілька ведучих, які в даний час викликають найбільший інтерес у дослідників і практиків.

Подання знань і розробка систем, заснованих на знаннях

Це основний напрямок в області розробки систем штучного інтелекту. Воно пов'язане з розробкою моделей представлення знань, створенням баз знань, які становлять ядро експертних систем.

Програмне забезпечення систем штучного інтелекту

В рамках цього напрямку розробляються спеціальні мови для вирішення інтелектуальних завдань, в яких наголос робиться на переважання логічного і символічної обробки над обчислювальними процедурами. Мови орієнтовані на символічну обробку інформації – LISP, PROLOG, РЕФАЛ і ін. Крім цього створюються пакети прикладних програм, орієнтовані на промислову розробку інтелектуальних систем, або програмні інструментарії штучного інтелекту.

Розробка природно-мовних інтерфейсів і машинний переклад

Починаючи з 50-х років однією з популярних тем досліджень в галузі штучного інтелекту є комп'ютерна лінгвістика, і, зокрема, машинний переклад. Ідея машинного перекладу виявилася зовсім не так проста, як здавалося першим дослідникам і розробникам.

Інтелектуальні роботи

Роботи – це електротехнічні пристрої, призначені для автоматизації людської праці. Виділяють кілька поколінь роботів:

I покоління. Роботи з жорсткою схемою управління. Практично всі сучасні промислові роботи належать до першого покоління. Фактично це програмовані маніпулятори.

II покоління. Адаптивні роботи з сенсорними пристроями. Є зразки таких роботів, але в промисловості вони поки використовуються мало.

III покоління. Самоорганізуються або інтелектуальні роботи. Це – кінцева мета розвитку робототехніки. Основні невідомі проблеми при створенні інтелектуальних роботів – проблема машинного зору і адекватного зберігання і обробки тривимірної візуальної інформації.

В даний час в світі виготовляється більше 60 000 роботів в рік. Фактично робототехніка сьогодні – це інженерна наука, не

відкидаєш технологій штучного інтелекту, але не готова поки до їх впровадження в силу різних причин.

Навчання і самонавчання

Активно розвивається область штучного інтелекту. Включає моделі, методи і алгоритми, орієнтовані на автоматичне накопичення і формування знань на основі аналізу та узагальнення даних, навчання за прикладами (або індуктивний), а також традиційні підходи з теорії розпізнавання образів.

В останні роки до цього напрямку тісно примикають стрімко розвиваються системи аналізу даних і пошуку закономірностей в базах даних.

Розпізнавання образів

Напрямок штучного інтелекту, що бере початок у самих його витоків, але в даний час виділилася в самостійну науку. Її основний підхід – опис класів об'єктів через певні значення значущих ознак. Кожному об'єкту ставиться у відповідність матриця ознак, по якій відбувається його розпізнавання. Процедура розпізнавання використовує найчастіше спеціальні математичні процедури і функції, що розділяють об'єкти на класи. Цей напрямок близько до машинного навчання і тісно пов'язане з нейрокібернетики.

Нові архітектури комп'ютерів

Найсучасніші процесори сьогодні засновані на традиційній послідовній архітектурі фон Неймана, використовуваної ще в комп'ютерах перших поколінь. Ця архітектура вкрай неефективна для символної обробки. Тому зусилля багатьох наукових колективів і фірм вже десятки років націлені на розробку апаратних архітектур, призначених для обробки символних і логічних даних. Створюються Пролог- і Лісп-машини, комп'ютери V і VI поколінь. Останні розробки присвячені комп'ютерам баз даних, паралельним і векторним комп'ютерів. І хоча вдалі промислові рішення існують, висока

вартість, недостатнє програмне оснащення і апаратна несумісність з традиційними комп'ютерами істотно гальмують широке використання нових архітектур.

Ігри

Це, стало швидше історичним, напрямок пов'язаний з тим, що на зорі досліджень штучного інтелекту традиційно включав в себе ігрові інтелектуальні завдання – шахи, шашки, го. В основі перших програм лежить один з ранніх підходів – лабіринтова модель мислення плюс евристики.

Зараз це швидше комерційний напрямок, так як в науковому плані ці ідеї вважаються тупиковими. В даний час в комп'ютерних іграх (наприклад, Unreal Tournament, Return to Castle Wolfenstein, Black & White, Doom, Sim) стали застосовуватися інші ідеї штучного інтелекту – нейронні мережі, інтелектуальні агенти, генетичні алгоритми і т. д., які дозволяють створювати персонажів (ботів) з різним ступенем «інтелекту». Використання методів штучного інтелекту в іграх дозволяє отримувати нові ефективні рішення, створювати шаблони проектування, підвищити розважальність і достовірність ігор.

Машинна творчість

Напрямок охоплює твір комп'ютером музики (Айзексон, Хіллер, Зармпов), віршів (Д. Лінк), живопису (Х. Фарід, Л. Моура) і навіть казок і афоризмів. Основним методом подібного «творчості» є метод пермутації (перестановок) плюс використання деяких баз знань і даних, що містять результати досліджень по структурам текстів, рим, сценаріями і т. п.

Нечіткі моделі та м'які обчислення.

Цей напрямок представлено нечіткими схемами «виведення за аналогією», поглядом на теорію нечітких заходів з імовірнісних позицій, нечітким поданням аналітичними моделями для опису геометричних об'єктів, алгоритмами еволюційного моделювання з динамікою, такими як час життя і

розмір популяції, методами вирішення оптимізаційних задач з використанням технологій генетичного пошуку, гомеостатических і синергетичних принципів і елементів самоорганізації.

Евристичне програмування

В рамках напряму досліджують послідовності розумових операцій, виконання яких призводить до успішного вирішення того чи іншого завдання, моделюють розумову діяльність людини для вирішення завдань, що не мають суворого формалізованого алгоритму або пов'язаних з неповнотою вихідних даних.

Штучне життя

Напрямок досліджень, метою якого є створення штучних істот, здатних діяти не менш ефективно, ніж живі істоти. М'яка штучне життя створює обчислювальні системи і моделі, що діють на базі біологічних і еволюційних принципів. Волога штучне життя синтезує нові штучні біологічні форми. В рамках цього напрямку використовують генетичні алгоритми, клітинні автомати, автономні агенти і т. д.

Когнітивне моделювання

Науковий напрямок, що є плідним синтезом когнітивної графіки і обчислювального моделювання, що дозволяє істотно підвищити пізнавальну ефективність сучасних ЕОМ. Методологія когнітивного моделювання призначена для аналізу і прийняття рішень в погано певних ситуаціях, ґрунтується на моделюванні суб'єктивних уявлень експерта.

Еволюційне моделювання

При еволюційному моделюванні процес моделювання складної соціально-економічної системи зводиться до створення моделі його еволюції або до пошуку допустимих станів системи, до процедури (алгоритму) відстеження множини допустимих станів (траєкторій).

Багатоагентні системи

Напрямок штучного інтелекту, яке розглядає рішення однієї задачі кількома інтелектуальними підсистемами – агентами. Агент – апаратна або програмна сутність, здатна діяти в інтересах досягнення мети, поставленої перед всією системою.

Соціальні системи дають ще одне модельне уявлення інтелекту за допомогою глобальної поведінки, яке дозволяє їм вирішувати проблеми, які б не вдалося вирішити окремим їх членам. Агенти в таких системах автономні або напівавтономні, у кожного агента є певне коло підзадач, причому він має в своєму розпорядженні малим знанням (або зовсім не має в своєму розпорядженні знанням) про те, що роблять інші агенти або як вони це роблять. Кожен агент виконує свою незалежну частину рішення проблеми і або видає власне результати (щось робить) або повідомляє результати іншим агентам.

Онтології

В рамках цього напрямку досліджується можливість всеосяжної і детальної формалізації деякої області знань за допомогою концептуальної схеми – ієрархічної структури даних, що містить всі релевантні класи об'єктів, їх зв'язку і правила предметної області. Онтології використовуються і людьми і програмними агентами, дозволяють повторно використовувати знання предметної області, відокремити їх від оперативних знань і аналізувати їх. Розробляються мови опису онтологій (RDF, DAML, OWL, KIF).

Комп'ютерні віруси

Останнє покоління комп'ютерних вірусів володіють всіма атрибутами систем штучного інтелекту. Вони здатні до розмноження, мутації, еволюції, навчання. Сучасні проблеми щодо захисту від них виявляються незначними, коли вони повністю проникнуть в сферу штучного інтелекту. Методи штучного інтелекту необхідні як для їх створення, так і для розробки ефективних засобів захисту.

Інтелектуальне математичне моделювання

В даному напрямку системи імітують творчу діяльність математика-професіонала, що займається рішенням, наприклад, крайових задач математичної фізики. Для цього використовуються бази знань, що містять теореми, математичні залежності, евристичні правила, такі системи здатні до навчання і самонавчання.

Це далеко не всі напрямки штучного інтелекту, існує безліч напрямків для вирішення безлічі завдань.

1.3 Визначення інтелектуальної інформаційної системи

Існує велика безліч інтелектуальних інформаційних систем. Однак загальноприйнятого єдиного визначення інтелектуальної інформаційної системи немає.

Інтелектуальної інформаційною системою називають автоматизовану інформаційну систему, засновану на знаннях, або комплекс програмних, лінгвістичних і логіко-математичних засобів для реалізації основного завдання – здійснення підтримки діяльності людини і пошуку інформації в режимі просунутого діалогу на природній мові.

Крім того, інформаційно-обчислювальними системами з інтелектуальною підтримкою для вирішення складних завдань називають ті системи, в яких логічна обробка інформації превалює над обчислювальною.

Таким чином, будь-яка інформаційна система, яка вирішує інтелектуальну задачу або в яких беруть методи штучного інтелекту, відноситься до інтелектуальних.

Для інтелектуальних інформаційних систем характерні наступні ознаки:

- розвинені комунікативні здібності.
- вміння вирішувати складні погано формалізуються завдання.
- здатність до самонавчання.
- адаптивність.

Комунікативні здібності ІВС характеризують спосіб взаємодії (інтерфейсу) кінцевого користувача з системою, зокрема, можливість формулювання довільного запиту в діалозі з ІВС на мові, максимально наближеному до природного.

Складні погано формалізуються завдання – це завдання, які вимагають побудови оригінального алгоритму рішення в залежності від конкретної ситуації, для якої можуть бути характерні невизначеність і динамічність вихідних даних і знань.

Здатність до самонавчання – це можливість автоматичного вилучення знань для вирішення завдань з накопиченого досвіду конкретних ситуацій

Адаптивність – здатність до розвитку системи відповідно до об'єктивними змінами моделі проблемної області.

Контрольні питання

1. Типи та форми подання знань
2. Загальна архітектура систем штучного інтелекту
3. Філософські, технічні, наукові передумови для створення штучного розуму. Поняття «Штучний інтелект».
4. Сучасні області досліджень і теоретичні проблеми штучного інтелекту
5. Основні теоретичні завдання, які вирішуються ШІ.
6. Области практичного застосування методів ШІ.
7. Ефективність вирішення практичних завдань методами ШІ та критерії її вимірювання.
8. Принципи ефективного застосування методів ШІ.

Лекція 2. ОСНОВИ ТЕОРІЇ ШТУЧНОГО ІНТЕЛЕКТУ. ПОДАННЯ ЗНАНЬ

План

2.1 Дані і знання

2.2 Класифікація моделей подання знань

2.1 Дані і знання

Дані – це інформація, отримана в результаті спостережень або вимірювань окремих властивостей (атрибутів), що характеризують об'єкти, процеси і явища предметної області.

Знання – форма існування і систематизації результатів пізнавальної діяльності людини. Знання допомагає людям раціонально організовувати свою діяльність і вирішувати різні проблеми виникають в її процесі; суб'єктивний образ об'єктивної реальності, тобто адекватне віддзеркалення зовнішнього і внутрішнього світу в свідомості людини у формі уявлень, понять, суджень, теорій.

Знання (в широкому сенсі) – сукупність понять, теоретичних побудов і уявлень.

Знання (у вузькому сенсі) – ознака певного обсягу інформації, що визначає її статус і відокремлює від всієї іншої інформації за критерієм здатності до вирішення поставленого завдання.

Знання (з точки зору представлення знань в інтелектуальних системах) – це зв'язки і закономірності предметної області (принципи, моделі, закони), отримані в результаті практичної діяльності і професійного досвіду, що дозволяє фахівцям ставити і вирішувати завдання в цій галузі.

Знання від даних відрізняються рядом **властивостей**:

- внутрішня інтерпретируемість;
- структурованість;
- зв'язність;
- семантична метрика;

- активність.

Внутрішня інтерпретируемість. Дані, що зберігаються в пам'яті або на зовнішніх носіях, позбавлені імен, таким чином, відсутня можливість їх однозначної ідентифікації системою. Дані може ідентифікувати лише програма, витягати їх за певним алгоритмом. При переході до знань в пам'ять вводиться додаткова інформація (атрибути: прізвище, рік народження, спеціальність, стаж). Атрибути можуть грати роль імен. За ним можна здійснювати пошук потрібної інформації.

Структурованість. Інформаційні одиниці повинні володіти гнучкою структурою. Інакше кажучи, повинна існувати можливість довільного встановлення між окремими інформаційними одиницями відносин типу «частина-ціле», «рід-вид» або «елемент-клас».

Зв'язність. Між інформаційними одиницями повинна бути передбачена можливість встановлення зв'язків різного типу. Семантика відносин може носити декларативний або процедурний характер. Наприклад, дві і більше інформаційні одиниці можуть бути пов'язані відношенням «одночасно», дві інформаційні одиниці – відношенням «причина-наслідок» або «бути поруч».

Семантична метрика. На безлічі інформаційних одиниць в деяких випадках корисно задавати відношення, що характеризує їх ситуаційну близькість, тобто силу асоціативного зв'язку. Його можна було б назвати відношенням релевантності для інформаційних одиниць. Воно дає можливість виділяти в інформаційній базі деякі типові ситуації (наприклад, «покупка», «регулювання руху на перехресті»). Ставлення релевантності при роботі з інформаційними одиницями дозволяє знаходити знання, близькі до вже знайденим.

Активність. Всі обчислювальні процеси ініціюються командами, а дані використовуються цими командами лише в разі потреби. Інакше кажучи, дані пасивні, а команди активні.

Знання дозволяють адаптуватися і діяти в реальній дійсності. Існує величезна безліч різних знань, починаючи від

рецепта приготування омлету до квантової фізики. Знання можна класифікувати за кількома критеріями (рис. 2.1).

Знання *синтаксичного* типу характеризує синтаксичну структуру потоку інформації, яка не залежить від змісту і змісту, які використовуються при цьому поняття, тобто інтелектуальну систему не утворює.

Семантичне знання розглядається як структура, що утворює поточний контекст. Воно містить інформацію, безпосередньо пов'язану з поточними значеннями і змістом описуваних понять і визначає стан зв'язків даних в інформаційній базі.

Прагматичне знання зумовлює найбільш ймовірні зв'язки, що описують дані з точки зору розв'язуваної задачі (узагальнений або «об'єктивний» контекст), наприклад, з урахуванням діючих в даній задачі специфічних критеріїв і угод.

Декларативні знання містять у собі уявлення про структуру понять. Ці знання наближені до даних, фактів. Наприклад: вищий навчальний заклад є сукупність факультетів, а кожен факультет в свою чергу є сукупність кафедр.

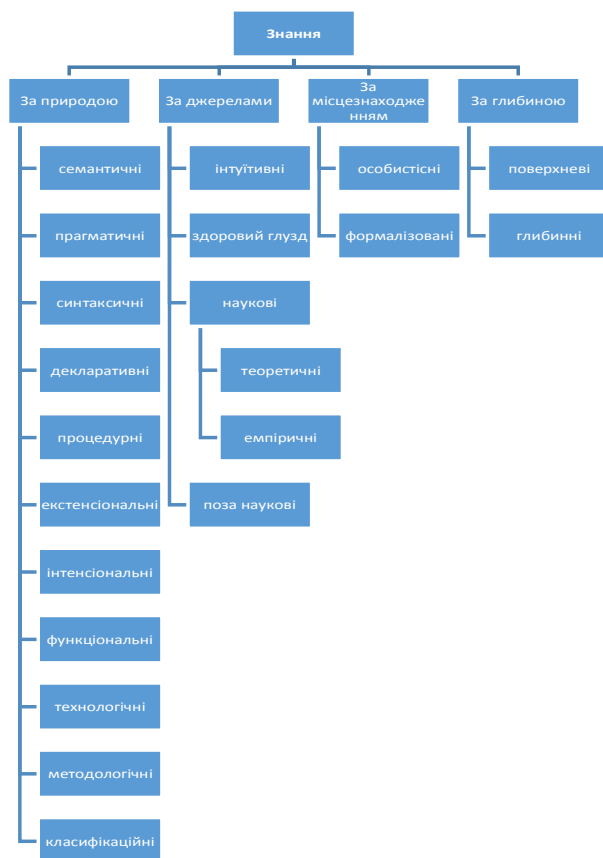


Рисунок 2.1 – Класифікація знань

Процедурні знання, мають активну природу. Вони визначають уявлення про засоби і шляхи отримання нових знань, перевірки знань. Це алгоритми різного роду. З розвитком інформатики все більша частина знань зосереджувалася в структурах даних (таблиці, списки, абстрактні типи даних), т. Е. Збільшувалася роль декларативних.

Істотним для розуміння природи знань є способи визначення понять. Один з широко застосовуваних способів заснований на ідеї iнтенсiонал і екстенсiонала.

Інтенціонал поняття – це визначення його через співвіднесення з поняттям більш високого рівня абстракції із зазначенням специфічних властивостей.

Екстенціонал поняття – це визначення поняття через перерахування його конкретних прикладів, т. е. понять більш низького рівня абстракції. Інтенціоналом формують знання про об'єкти, в той час як екстенціонал об'єднує дані.

Звідси *інтенціональні* знання – це знання про предметну область, яка відображає факти, закономірності, властивості і характеристики, справедливі для будь-яких ситуацій, які можуть виникнути в цій предметній області.

Екстенціональності знання – це знання про предметну область, що відображають факти, закономірності, властивості і характеристики, типові для конкретних ситуацій або класів однотипних ситуацій, які можуть виникнути в цій області.

Функціональні знання – це знання про виконувані функції окремих предметів і про застосування їх в реальній дійсності.

Технологічні знання – спеціалізовані знання, щоб забезпечити підтримку технологічних параметрів виробництва; виробничий досвід і навички, які використовуються при вирішенні повсякденних виробничих питань. Це може бути знання послідовності операцій або знання технологічного ланцюжка дозволяють досягати поставлених цілей відповідно до прийнятої технологією.

Методологічні знання – знання про методи перетворення дійсності, наукові знання про побудову ефективної діяльності. До методологічних знань відносять знання цілей, форм і напрямків розвитку теорії, методів і способів ефективного перетворення практики.

Класифікаційні знання застосовуються головним чином в науці, є узагальненими, системними знання. Приклад – система елементів Д.І. Менделєєва.

Інтуїція – це вид знання, специфіка якого обумовлена способом його придбання. Це знання, що не потребує доведення і сприймається як достовірне. За способом отримання інтуїція – це пряме розсуд об'єктивної зв'язку речей, що не спирається на

доказ (інтуїція, від лат. *Intueri* – споглядати, – є розсуд внутрішнім зором).

Під *здоровим глуздом* розуміють знання дозволяють приймати правильні рішення і робити правильні припущення, ґрунтуючись на логічному мисленні і накопичений досвід. У цьому значенні термін часто акцентує увагу на здатності людського розуму протистояти забобонам, помилкам, містифікацій.

Наукові знання в будь-якому випадку повинні бути обґрунтованими на емпіричній або теоретичній доказовій основі.

Теоретичні знання- абстракції, аналогії, схеми, що відображають структуру і природу процесів, що протікають в предметній області. Ці знання пояснюють явища і можуть використовуватися для прогнозування поведінки об'єктів. Теоретичний рівень наукового знання припускає встановлення законів, що дають можливість ідеалізованого сприйняття, описи і пояснення емпіричних ситуацій, тобто пізнання сутності явищ. Теоретичні закони мають більш строгий, формальний характер, в порівнянні з емпіричними. Терміни опису теоретичного знання відносяться до ідеалізованим, абстрактним об'єктам. Подібні об'єкти неможливо піддати безпосередній експериментальній перевірці.

Емпіричні знання отримують в результаті застосування емпіричних методів пізнання – спостереження, вимірювання, експерименту. Це знання про видимі взаємозв'язки між окремими подіями і фактами в предметній області. Воно, як правило, констатує якісні та кількісні характеристики об'єктів і явищ. Емпіричні закони часто носять імовірнісний характер і не є строгими.

Позанаукові знання можуть бути різними. *Паранормальні* знання-знання несумісні з наявним ґносеологічним стандартом. Широкий клас *паранаукового* (пара від грец. – близько, при) знання включає в себе вчення або роздуми про феномени, пояснення яких не є переконливим з точки зору критеріїв науковості. *Лженаукові* знання – свідомо експлуатуючі домисли

і забобони. Як симптоми лженауки виділяють малограмотний пафос, принципову нетерпимість до спростовує доводам, а також претензійність. Лженаучні знання співіснують з науковими знаннями.

Особистісні (неявні, приховані) знання – це знання людей, отримані з практики і досвіду.

Формалізовані (явні) знання – знання містяться в документах, на компакт дисках, в персональних комп'ютерах, в Інтернеті, в базах знань, в експертних системах. Формалізовані знання об'єктивізуються знаковими засобами мови, охоплюють ті знання, про які ми знаємо, їх можна записати, повідомити іншим.

2.2 Класифікація моделей подання знань

Для зберігання даних використовуються бази даних (для них характерні великий обсяг і відносно невелика питома вартість інформації), для зберігання знань – бази знань (невеликого обсягу, але виключно дорогі інформаційні масиви).

База знань – основа будь-якої інтелектуальної системи, де знання описані на деякій мові представлення знань, наближеному до природного. Сьогодні знання придбали чисто декларативну форму, т. Е. Знаннями вважаються пропозиції, записані на мовах подання знань, наближених до природної мови і зрозумілих неспеціалістам.

Сукупність знань потрібних для прийняття рішень, прийнято називати предметною областю або **знаннями про предметну область**. У будь-якій предметній області є свої поняття і зв'язки між ними, своя термінологія, свої закони, що зв'язують між собою об'єкти даних предметної області, свої процеси і події. Крім того, кожна предметна область має свої методи вирішення завдань. Вирішуючи завдання такого виду на ЕОМ використовують інформаційні системи, ядром яких є база знань, яка містить основні характеристики предметних областей.

Бази знань базуються на **моделях представлення знань**, подібно баз даних, які засновані на моделях представлення

даних (ієрархічної, мережевої, реляційної, постреляційної і т. д.)

При поданні знань в пам'яті інтелектуальної системи традиційні мови, засновані на чисельному поданні даних, є неефективними. Для цього використовуються спеціальні мови представлення знань, засновані на символічному поданні даних. Вони діляться на типи за формальними моделям представлення знань. Найбільш часто використовується на практиці класифікація моделей подання знань, наведена на рис. 2.2, де моделі представлення знанні діляться на детерміновані (жорсткі) і м'які.

Детерменіровані моделі включають в себе фрейми, логіко-алгебраїчні моделі, семантичні мережі і продукційні моделі. М'які моделі включають в себе нечіткі системи, нейронні мережі, еволюційні моделі, гібридні системи.

З моделюванням знань безпосередньо пов'язана проблема вибору мови уявлення. З метою класифікації моделей уявлення знань виділяється дев'ять ключових вимог до моделей знань:

- 1) спільність (універсальність);
- 2) наочність представлення знань;
- 3) однорідність;
- 4) реалізація в моделі властивості активності знань;
- 5) відкритість;
- 6) можливість відображення структурних відносин об'єктів предметної області;
- 7) наявність механізму «проектування» знань на систему семантичних шкал;
- 8) можливість оперування нечіткими знаннями;
- 9) використання багаторівневих уявлень (дані, моделі, метамодягли, метаметамоделі і т. д.).

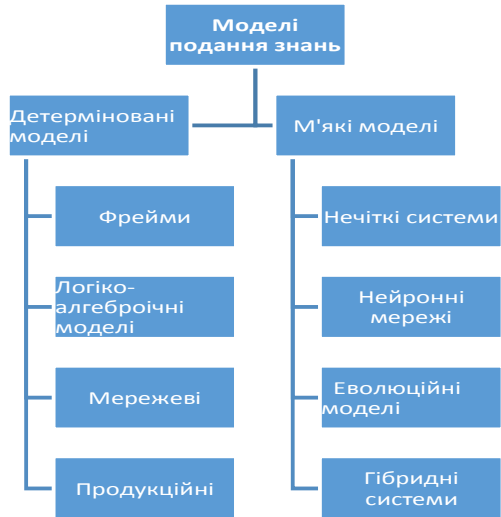


Рисунок 2.2 – Моделі представлення знань

Моделі подання знань не задовольняють повністю ці вимоги, чим і пояснюється їх різноманіття і активний розвиток цього напрямку.

2.2.1 Логіко-алгебраїчні моделі подання знань

У логічних моделях знання представляються у вигляді сукупності правильно побудованих формул будь-якої формальної системи, яка задається четвіркою

$$S = \langle T, P, A, R \rangle,$$

де T – множина базових (термінальних) елементів, з яких формуються всі вирази; P – множина синтаксичних правил, що визначають синтаксично правильні вирази з термінальних елементів формальної системи; A – множина аксіом формальної системи, відповідних синтаксично правильним виразами, які в рамках даної ФС апріорно вважаються істинними; R – кінцева множина відносин $\{r_1, r_2, \dots, r_n\}$ між формулами, званими

правилами виведення, що дозволяють отримувати з одних синтаксично правильних виразів інші.

Для будь-якого r_i існує ціле додатне число j , таке, що для кожної множини, що складається з j формул, і для кожної формули F ефективно вирішується питання про те, чи знаходяться ці j -формули відносно r_i з формулою F . Якщо r_i виконується, то F називають безпосереднім наслідком F -формул за правилом r_i .

Наслідком (висновком) формули в теорії S називається така послідовність правил, що для будь-якого з них представлена формула ϵ або аксіомою теорії S , або безпосереднім наслідком.

Найпростішою логічною моделлю є числення висловів, яке представляє собою один з початкових розділів математичної логіки, що є основою для побудови більш складних формалізмів. У практичному плані обчислення висловлювань застосовується в ряді предметних областях (зокрема, при проектуванні цифрових електронних схем). Розвиток логіки висловлювань знайшло відображення в численні предикатів першого порядку.

Під *обчисленням предикатів* розуміється формальний мову для подання відносин в деякій предметній області. Основна перевага числення предикатів – добре зрозумілий механізм математичного виведення, який може бути безпосередньо запрограмований. Предикатом називають пропозицію, яка приймає тільки два значення: «істина» або «брехня». Для позначення предикатів застосовуються логічні зв'язки між висловлюваннями: \neg – ні, \vee – або, \wedge – і, \supset – якщо, а також квантор \exists існування і квантор загальності \forall .

Таким чином, логіка предикатів оперує логічними зв'язками між висловлюваннями, наприклад, вона вирішує питання: чи можна на основі висловлювання A отримати висловлювання B і т. д.

Допустимі висловлювання на обчисленні предикатів називаються правильно побудованими формулами, що складаються з атомних формул. Атомні формули складаються з

предикатів і термів, що розділяються круглими, квадратними і фігурними дужками.

Предикатні символи – це в основному дієслівна форма (наприклад: ПИСАТИ, ВЧИТИ, ПЕРЕДАТИ), але не тільки дієслівна форма, а форма прикметників, прислівників (наприклад: ЧЕРВОНИЙ, ЗНАЧЕННЯ, ЖОВТИЙ).

Предикатні символи і константи, як правило, позначатися великими символами, функціональні символи і змінні – малими.

В абстрактних прикладах вони позначаються латинськими літерами f, g, h. У пропозиціях предикатної форми важливі відносини і елементи. Визначаючи відносини, ми визначаємо значимість елементів вираження. Елементи можуть бути предикатами і термами.

Якщо існує деяка предметна область, то предикати визначають відносини в цій предметній області, константи – елементи цієї предметної області, функціональний символ – функцію.

Розглянемо деякі приклади. Вислів «у кожної людини є батько» можна записати:

$$\forall x \exists y (\text{ЛЮДИНА } (x) \supset \text{БАТЬКО } (y, x))$$

Вираз «Антон володіє червоною машиною» записується, наприклад, так:

$$\exists x (\text{ВОЛОДІЄ } (\text{АНТОН}, x) \supset \text{МАШИНА } (x) \wedge \text{ЧЕРВОНИЙ } (x))$$

Подання знань в рамках логіки предикатів є основою напрямки II, що називається логічним програмуванням []. Методи логічного програмування в даний час широко використовуються на практиці при створенні II в ряді предметних областей. Позитивними рисами логічних моделей знань в цілому є:

- високий рівень формалізації, що забезпечує можливість реалізації системи формально точних визначень і висновків;
- узгодженість знань як єдиного цілого, що полегшує вирішення проблем верифікації БЗ, оцінки незалежності та повноти системи аксіом і т. д. ;

- єдині засоби опису як знань про предметну область, так і способів вирішення завдань в цій предметній області, що дозволяє будь-яке завдання звести до пошуку логічного виведення деякої формули в тій чи іншій формальній системі.

Однак така однаковість тягне за собою основний недолік моделі – складність використання в процесі логічного висновку евристик, що відображають специфіку предметної області. До інших недоліків логічної моделі відносять:

- «монотонність»;
- «комбінаторний вибух»;
- слабкість структурованості описів.

2.2.2 Продукційні моделі подання знань

Продукционная модель або модель, заснована на правилах, дозволяє представити знання у вигляді пропозицій типу «Якщо (умова), то (дія)».

Під «умовою» (антецедентом) розуміється деякий пропозицію-зразок, за яким здійснюється пошук в базі знань, а під «дією» (консеквентом) – дії, що виконуються при успішному результаті пошуку (вони можуть бути проміжними, які виступають далі як умови, і термінальними або цільовими, завершальними роботу системи).

Продукционная модель в чистому вигляді не має механізму виходу з тупикових станів в процесі пошуку. Вона продовжує працювати, поки не будуть вичерпані всі допустимі продукції. Практичні реалізації продукційних систем містять механізми повернення в попередній стан для управління алгоритмом пошуку.

Розглянемо приклад використання продукційних правил.

П1: Якщо «відпочинок – влітку» і «людина – активний», то «їхати в гори».

П2: Якщо «любить сонце», то «відпочинок влітку».

Припустимо, в систему надійшли дані – «людина активна» і «любить сонце».

прямий висновок – виходячи з даних, отримати відповідь.

1-й прохід.

Крок 1. Пробуємо П1, не працює (не вистачає даних «відпочинок – влітку»).

Крок 2. Пробуємо П2, працює, в базу надходить факт «відпочинок – влітку».

2-й прохід.

Крок 3. Пробуємо П1, працює, активується мета «їхати в гори», яка і виступає як порада, яку дає ЕС.

зворотний висновок – підтвердити обрану мета за допомогою наявних правил і даних.

1-й прохід.

Крок 1. Мета – «їхати в гори»: пробуємо П1 – даних «відпочинок – влітку» немає, вони стають новою метою, і шукається правило, де вона в правій частині.

Крок 2. Мета «відпочинок – влітку»: правило П2 підтверджує мету і активує її.

2-й прохід.

Крок 3. Пробуємо П1, підтверджується шукана мета.

Основні переваги продукційних систем:

- простота і гнучкість виділення знань;
- відділення знань від програми пошуку;
- модульність продукційних правил (правила не можуть «викликати» інші правила);
- можливість евристичного управління пошуком;
- можливість трасування «ланцюжка міркувань»;
- незалежність від вибору мови програмування;
- продукційні правила є правдоподібною моделлю рішення задачі людиною.

Є велика кількість програмних засобів, що реалізують Продукційний підхід (наприклад, мови високого рівня CLIPS і OPS 5; «оболонки» або «порожні» ЕС – EXSYS Professional і Карра, інструментальні KEE, ARTS, PIES, а також промислових ЕС на його основі.

2.2.3 Семантичні мережі

Семантична мережа – це орієнтований граф, вершини якого – поняття, а дуги – відносини між ними. Термін «*семантична*» означає «сміслова», а сама семантика – це наука, що встановлює відносини між символами і об'єктами, які вони позначають, т. Е. Наука, яка визначає зміст знаків. Модель на основі семантичних мереж була запропонована американським психологом Куїлліаном.

Як поняття зазвичай виступають абстрактні або конкретні об'єкти, а відносини це зв'язки типу: «це» («АКО – A-Kind-Of,» is «або» елемент класу «),» має частиною («(« has part «), «належить», «любить».

Можна запропонувати кілька класифікацій семантичних мереж, пов'язаних з типами відносин між поняттями.

За кількістю типів відносин:

- однорідні (з єдиним типом відносин);
 - неоднорідні (з різними типами відносин).

За типами відносин:

- бінарні (в яких відносини пов'язують два об'єкти);
- N-арні (в яких є спеціальні відносини, що зв'язують більше двох понять).

Найбільш часто в семантичних мережах використовуються наступні відносини:

- елемент класу (тройнда це квітка);
- атрибутивні зв'язку / мати властивість (пам'ять має властивість – обсяг);
 - значення властивості (колір має значення – жовтий);
 - приклад елемента класу (тройнда, наприклад – чайна);
 - зв'язку типу «частина-ціле» (велосипед включає кермо);
 - функціональні зв'язки (визначені зазвичай дієсловами «виробляє», «впливає» ...);
- кількісні (більше, менше, дорівнює ...);
- просторові (далеко від, близько від, за, під, над ...);
- тимчасові (раніше, пізніше, протягом ...);

- логічні зв'язки (і, або, не) і ін.

Мінімальний склад відносин в семантичній мережі такий:

- елемент класу або АКО;
- атрибутивні зв'язку / мати властивість;
- значення властивості.

Недоліком цієї моделі є складність організації процедури організації виведення.

Приклад семантичної мережі для пропозиції типу «Постачальник здійснив поставку виробів для клієнта до 1 червня 2010 року в кількості 10 000 штук» представлений на рис. 2.3.

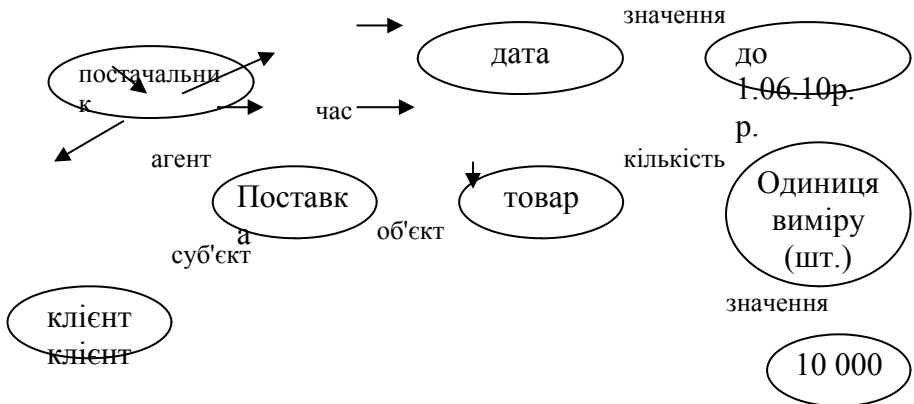


Рисунок 2.3 – Приклад семантичної мережі

2.2.4 Фрейми

Фрейм – це мінімально можливе опис сутності якої-небудь події, ситуації, процесу або об'єкта. В історичному плані розвиток фреймової моделі пов'язано з теорією фреймів М. Мінського, визначальною спосіб формалізації знань, який використовується при вирішенні задач розпізнавання образів (сцен) і розуміння мови. «Відправним моментом для даної теорії служить той факт, що людина, намагаючись пізнати нову для себе ситуацію або по-новому поглянути на вже звичні речі,

вибирає зі своєї пам'яті деяку структуру даних (образ), звану нами фреймом, з таким розрахунком, щоб шляхом зміни в ній окремих деталей зробити її придатною для розуміння більш широкого класу явищ або процесів». Іншими словами, фрейм – це форма опису знань, що окреслює рамки розглянутого (в поточній ситуації при вирішенні даного завдання) фрагмента предметної області.

Модель фрейма є досить універсальною, оскільки дозволяє відобразити все різноманіття знань про світ через:

- фрейми – структури, для позначення об'єктів і понять (позики, застава, вексель);
- фрейми – ролі (Менеджер, касир, клієнт);
- фрейми – сценарії (Банкрутство, збори акціонерів, святкування іменин);
- фрейми – ситуації (Тривога, аварія, робочий режим пристрою) і ін.

Розрізняють фрейми-зразки або прототипи і фрейми-екземпляри, які створюються для відображення реальних фактичних ситуацій на основі даних, що надходять.

Фрейм має ім'я (назва) та складається з слотів.

Традиційно структура фрейму може бути представлена як список властивостей:

(Ім'я фрейма:

(Ім'я 1-го слота: значення 1-го слота),

(Ім'я 2-го слота: значення 2-го слота),

.....

(Ім'я N-го слота: значення N-го слота)).

Ту ж запис можна представити у вигляді таблиці (див. табл. 2.1), доповнивши її двома стовпцями.

Таблиця 2.1 – Структура фрейму

Ім'я фрейму

Ім'я слота	Значення слота	Спосіб отримання значення	Приєднана процедура

У таблиці додаткові стовпці (3-й і 4-й) призначені для описання способу отримання слотом його значення і можливого приєднання до того чи іншого слоту спеціальних процедур, що допускається в теорії фреймів. Як значення слота може виступати ім'я іншої фрейма, так утворюються мережі фреймів.

Широко відомі такі фрейми-орієнтовані експертні системи, як ANALYST, МОДІС.

Контрольні питання

1. Класифікація знань. Властивості знань. Організація бази знань.
2. Методи представлення знань.
3. Загальна схема моделей подання знань.
4. Логічна модель для представлення знань
5. Види логічних моделей, загальні терміни і визначення.
6. Методи автоматичного доведення теорем (обчислення предикатів).
7. Продукційна модель для представлення знань.
8. Опис предметної області правилами і фактами.
9. Методи повного перебору в ширину і в глибину.
10. Евристичні методи пошуку в просторі станів.
11. Алгоритми локального пошуку і завдання оптимізації.
12. Фрейми для представлення знань.
13. Представлення знань про об'єкт за допомогою фреймів, приклади.
14. Практична реалізація фреймової моделі.
15. Поняття про об'єктно-орієнтованому аналізі предметної області.
16. Об'єктно-орієнтовані мови програмування.
17. Семантичні мережі для подання знань.
18. Типи вузлів і типи відносин.

19. «Поверховість» і «глибинність» знань, як основні відмінності моделі семантичних мереж від продукційної моделі.

20. Предметні області, де семантичні мережі набули поширення. Приклади.

Лекція 3. НЕЙРОННІ МЕРЕЖІ

План

- 3.1 Класифікація штучних нейронних мереж
- 3.2 Одношарові штучні нейронні мережі
- 3.3 Багатошарові нейронні мережі
- 3.4 Завдання, які вирішуються нейронними мережами

Так як основне завдання штучного інтелекту – моделювання міркувань, а природне «пристрій здатне думати» – це мозок, то очевидною завданням є створення «штучного мозку» «за образом і подобою» людського. Дослідженням цього питання стали займатися в рамках напрямку штучного інтелекту нейрокібернетика.

Пристрої мозку вивчають такі науки як психологія, нейрофізіологія, нейробиологія, що володіють достатнім обсягом знань. Основною ідеєю нейрокібернетики стало відтворення «в залізі» клітини мозку – нейрони.

Нейрони – спеціалізовані клітини, здатні приймати, обробляти, кодувати, передавати і зберігати інформацію, організовувати реакції на подразнення, встановлювати контакти з іншими нейронами, клітинами органів. Унікальними особливостями нейрона є здатність генерувати електричні розряди і передавати інформацію за допомогою спеціалізованих закінчень – синапсів.

Число нейронів мозку людини наближається до 10^{11} . На одному нейроні може бути до 10 000 синапсів. Якщо тільки ці елементи вважати осередками зберігання інформації, то можна прийти до висновку, що нервова система може зберігати 10^{19} од. інформації, т. е. здатна вмістити практично все знання, накопичені людством.

На рис. 3.1 приведена схема будови «типового» нейрона.

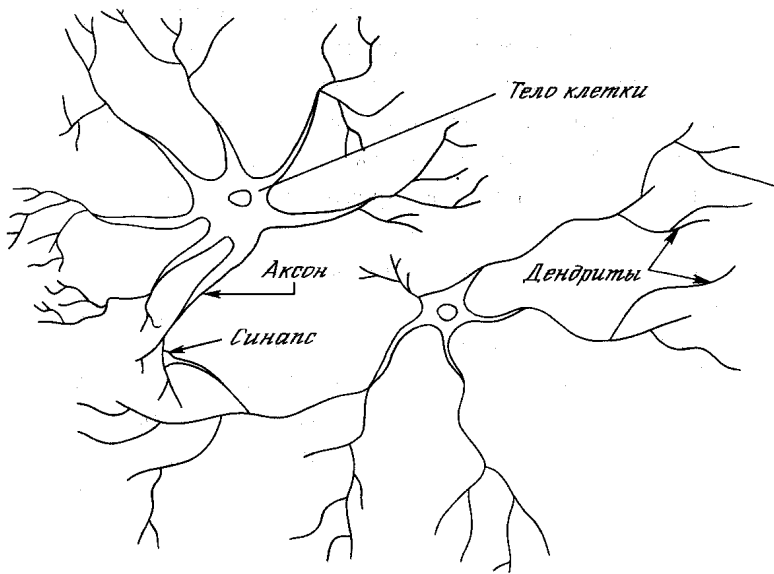


Рисунок 3.1 – Загальна схема будови біологічного нейрона

Тіло клітини містить безліч розгалужених відростків двох типів. Відростки першого типу, звані дендритами за їх схожість з кроною розлогого дерева, служать в якості входних каналів для нервових імпульсів від інших нейронів. Ці імпульси надходять в сому або тіло клітини розміром від 3 до 100 мікрон, викликаючи її специфічне збудження, яке потім поширюється по вивідній відростка другого типу – аксону. Довжина аксонів зазвичай помітно перевершує розміри дендритів, в окремих випадках досягаючи десятків сантиметрів і навіть метрів. Нейрон може перебувати в 2 станах: збудженому або збудженому.

1943 рік став роком народження теорії штучних нейронних мереж. Дж. Маккалок і У. Пітт запропонували модель формального нейрона (рис. 9) і описали основні принципи побудови нейронних мереж.

Синапси нейрона – місця контактів нервових волокон – відповідно до цієї моделі, передають сигнали, визначаючи силу впливу сигналу з цього входу на вихідний сигнал, що надходить

на аксон. Для цього кожному входу ставиться у відповідність ваговий коефіцієнт w_i . Дендрити отримують вхідний сигнал, представлений вектором x_i . Потім нейрон обробляє надійшов сигнал, виробляючи зважене підсумовування і нелінійне перетворення, використовуючи для цього активаційну функцію (функції активації можуть бути різними, найбільш поширені лінійна, порогова і сігмоїда), аргументом якої буде результат підсумовування мінус граничне значення. Це значення визначає рівень сигналу, на який нейрон буде реагувати.

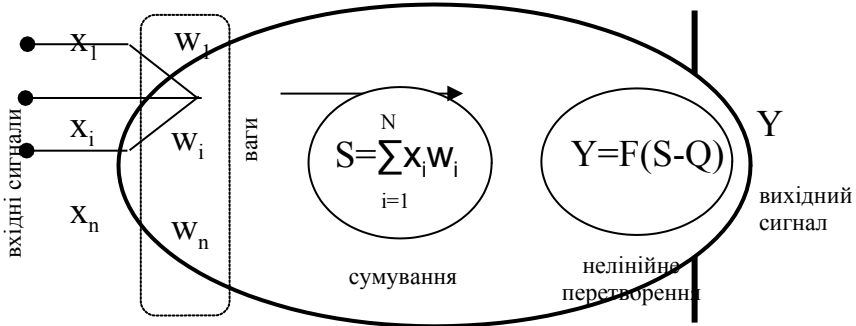


Рисунок 3.2 – Модель формального нейрона

Головною властивістю біологічного нейрона є його здатність до навчання, його універсальність, здатність вирішувати різні завдання. Описана вище модель не здатна до цього. Навченою вона стала лише в 1949 році завдяки Д. Хеббу (D. Hebb), який, спираючись на фізіологічні та психологічні дослідження, висунув гіпотезу про навченості біологічних нейронів. Його метод навчання став відправною точкою для алгоритмів навчання нейронних мереж без вчителя.

Уже через рік після цього в 1957 році в світовому науковому світі стався другий за значимістю в історії нейронних мереж подія: американський фізіолог Ф. Розенблатт розробив модель зорового сприйняття і розпізнавання – перцептрон (perceptron), а потім і побудував перший нейрокомп'ютер Марк-1.

Нейронні мережі була прості, зрозумілі і багатообіцяючі. Складні процеси мислення, здавалося, були готові розкритися перед людиною, для їх опису були потрібні лише елементарні математичні операції: додавання, множення і лінійна функція. На перцептрон, а потім і багатошаровий перцептрон покладалися великі надії, тому бурхливий ентузіазм, з яким він був прийнятий, досить швидко змінився жорсткою критикою.

У 1969 році вийшла в світ книга «Перцептрони» М. Мінського і С. Паперті, яка ознаменувала закінчення першого етапу в історії нейронних мереж. У цій книзі було проведено аналіз можливостей одношарових нейронних мереж та їх обмежень.

Обмеження, виявлені Мінським, були характерні одношаровим нейронних мереж, але не багатошаровим нейронних мереж, яким була властива інша проблема – їх навчання.

Запропонований 1986 році Д. Румельхардом алгоритм зворотного поширення помилки став одним з провідних чинників, що породили сучасний нейросетевий бум, тому що був ефективним способом навчання нейронної мережі досить довільної структури.

Нейронні мережі можуть реалізовуватися як програмно, так і апаратно. Поступово напрямок нейрокібернетика перетворилося в нейрокомп'ютинг, шосте покоління комп'ютерів, що базується на нейронних мережах, має свої переваги і недоліки. Порівняльний аналіз машини фон Неймана, ідеї якої втілені в перших 4 поколіннях, і біологічної нейронної системи, що є прообразом нейрокомп'ютера, представлений в таблиці 3.1.

Таблиця 3.1 Порівняння машини фон Неймана з біологічної нейронної системою

	Машина фон Неймана	Біологічна нейронна система
Процесор	складний	простий
	високошвидкісний	низькошвидкісний
	один або кілька	велика кількість
Пам'ять	відокремлена від процесора	інтегрована в процесор
	локалізована	розподілена
	адресація не за змістом	адресація за змістом
Обчислення	централізовані	розподілені
	послідовні	паралельні
	збережені програми	самонавчання
Надійність	висока вразливість	живучість
Спеціалізація	чисельні і символні операції	проблеми сприйняття
Середовище функціонування	строго певна	погано певна
	строго обмежена	без обмежень

3.1 Класифікація штучних нейронних мереж

За топології (рис. 3.3, рис. 3.4):

- повнозв'язні (кожен нейрон пов'язаний з усіма іншими нейронами, в тому числі і сам з собою);
- багатошарові (нейрони розташовуються шарами і кожен нейрон наступного шару пов'язаний з усіма нейронами поточного шару).

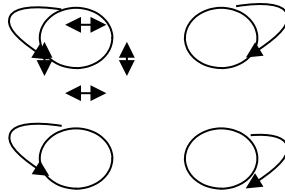


Рисунок 3.3 – Повнозв'язна нейронна мережа

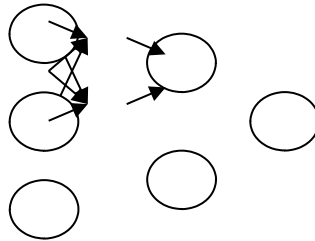


Рисунок 3.4 – Шарувата нейронна мережа

За організації навчання:

- з учителем (нейронну мережу навчають, подаючи на вхід значення навчальної вибірки і надаючи необхідні вихідні значення);

- без вчителя (на входи нейронної мережі подають множину об'єктів і нейронна мережа сама ділить їх на кластери або класи).

За типами структур (рис. 3.5):

- нейрони з одним типом функції активації (всі нейрони мережі мають одну функцію активації $f(x)$, наприклад, лінійну);

- нейрони з декількома типами функцій активації (нейрони мережі мають різні функції активації).

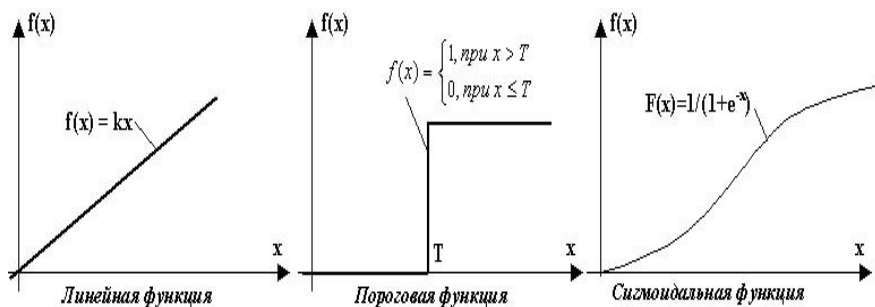


Рисунок 3.5 – Активаційні функції

За типом зв'язків:

- прямого поширення (без зворотних зв'язків між нейронами, до таких мереж відносяться одношаровий і багатшаровий персептрони, мережа радіальних базисних функцій);
- рекурентні (зі зворотним зв'язком, від виходів нейронів до входів, до таких мереж відносяться змагальні мережі і мережу Хопфілда).

За типом сигналу:

- бінарні (на входи подаються тільки нулі і одиниці);
- аналогові (на входи нейронів подаються значення безперервних функцій).

3.2 Одношарові штучні нейронні мережі

Один нейрон здатний виконувати найпростіші процедури розпізнавання, але тільки з'єднання декількох нейронів здатне вирішити практично корисну завдання. Найпростіша мережа (рис. 3.6) складається з групи нейронів, що утворюють шар).

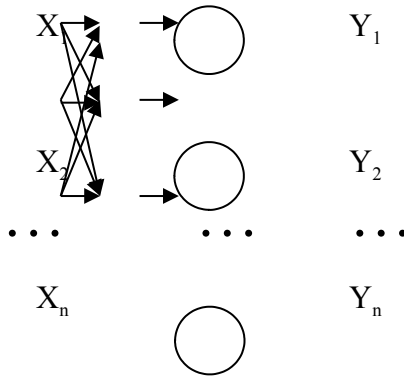


Рисунок 3.6 – Одношарова нейронна мережа

Навчання за дельта-правилом

Дельта-правило є узагальненням алгоритму навчання перцептрона. Дельта-правило працює тільки з безперервними, диференційовними функціями в режимі навчання з учителем (supervised learning). Помилка, що обчислюється в процесі навчання мережі, – це функція, що характеризує якість навчання даної мережі, тому процес навчання нейронної мережі можна уявити як процес мінімізації функції помилки. Напрямок зміни значення функції можна встановити, обчисливши похідну для функції одного змінного або градієнт для функції багатьох змінних. При мінімізації значення функції багатьох змінних менше значення необхідно шукати в напрямку антиградієнта.

В даному алгоритмі навчання початкові ваги можуть бути будь-якими.

Процес навчання можна вважати завершеним, якщо досягнута якась заздалегідь встановлена мінімальна помилка або алгоритм пропрацював умовлене кількість разів.

Алгоритм навчання по дельта-правилом:

1 крок: ініціалізація матриці ваг (і порогів, у разі використання порогової функції активації) випадковим чином.

2 крок: пред'явлення нейронної мережі способу (на вхід подаються значення з навчальної вибірки – вектор X), береться відповідний вихід (вектор D).

3 крок: обчислення вихідних значень нейронної мережі (вектор Y).

4 крок: обчислення для кожного нейрона величини розбіжності реального результату з бажаним.

$$\varepsilon_i = (d_i - y_i),$$

де d_i – бажане вихідне значення на i -нейроні, y_i – реальне значення на i -нейроні.

5 крок: зміна ваг (і порогів при використанні порогової функції) за формулами:

$$\begin{aligned}w_{ij}(t+1) &= w_{ij}(t) - \eta \cdot \varepsilon_i \cdot x_j, \\ \theta_i(t+1) &= \theta_i(t) - \eta \cdot \varepsilon_i,\end{aligned}$$

где t – номер поточної ітерації циклу навчання,

w_{ij} – вага зв'язку j -входу з i -нейроном,

η – коефіцієнт навчання, задається від 0 до 1, x_j – вхідне значення,

θ_i – порогове значення i -нейрона.

6 крок: перевірка умови продовження навчання (обчислення значення помилки i / або перевірка заданого кількості ітерацій). Якщо навчання не завершено, то 2 крок, інакше закінчуємо навчання.

3.3 Багатошарові нейронні мережі

Одношарові нейронні мережі мають свої обмеження, на які вказав Мінський, багатошарові мережі вільні від них. Вони володіють великими обчислювальними можливостями. Хоча створені мережі всіх конфігурацій, які тільки можна собі уявити, послійная організація нейронів копіює шаруваті структури певних відділів мозку. Багатошарові мережі утворюються

каскадами шарів. Вихід одного шару є входом для наступного шару, така організація мережі утворює багатозарову нейронну мережу з прямим розповсюдженням (рис. 3.7).

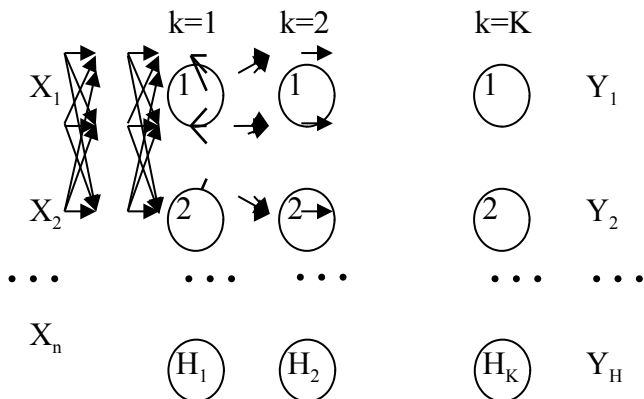


Рисунок 3.7 – Багатозарова нейронна мережа прямого поширення

Багатозарова мережа може містити довільну кількість шарів (K), кожен шар полягає з декількох нейронів, число яких також може бути довільно (H_k – кількість нейронів в шарі), кількість входів n , кількість виходів $H = H_k$ – числу нейронів у вихідному (останньому) шарі.

Шари між першим і останнім називаються проміжними або прихованими. Ваги в такій мережі мають три індексу i – номер нейрона наступного шару, для якого зв'язок вхідний, j – номер входу або нейрона поточного шару, для якого зв'язок вихідний, k – номер поточного шару в нейронній мережі (для входів, вектора X , $k = 0$).

Навчання методом зворотного поширення помилки

Навчання алгоритмом зворотного поширення помилки передбачає два проходи по всім верствам мережі: прямого і зворотного.

При прямому проході вхідний вектор подається на вхідний прошарок нейронної мережі, після чого поширяється по мережі

від шару до шару. В результаті генерується набір вихідних сигналів, який і є фактичною реакцією мережі на даний вхідний образ. Під час прямого проходу все синаптичні ваги мережі фіксовані.

Під час зворотного проходу все синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного, в результаті чого формується сигнал помилки. Цей сигнал згодом поширюється по мережі в напрямку, протилежному напрямку синаптичних зв'язків. Звідси і назва – алгоритм зворотного поширення помилки. Синаптичні ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного.

Алгоритм навчання по дельта-правилом:

1 крок: ініціалізація матриць ваг випадковим чином (в циклах).

2 крок: пред'явлення нейронної мережі способу (на вхід подаються значення з навчальної вибірки – вектор X) і береться відповідний вихід (вектор D).

3 крок (прямий прохід): обчислення в циклах виходів всіх верств і отримання вихідних значень нейронної мережі (вектор Y).

$$y_i^k = f \left(\sum_{j=0}^{H_{k-1}} w_{ij}^k \cdot y_j^{k-1} \right),$$
$$y_j^0 = x_j,$$
$$y_0^{k-1} = 1,$$
$$x_0 = 1,$$

де y_i^k – вихід і-нейрона k-шару;
 f – функція активації;

w_{ij}^k – синаптичний зв'язок між j -нейроном шару $k-1$ і i -нейроном шару k ;

x_j – вхідне значення.

4 крок (зворотний прохід): зміна ваг в циклах за формулами:

$$w_{ij}^k(t+1) = w_{ij}^k(t) + \eta \cdot \delta_i^k \cdot y_j^{k-1},$$
$$\delta_i^k = (d_i - y_i) \cdot y_i \cdot (1 - y_i) -$$

для останнього (вихідного) шару,

$$\delta_i^k = y_i \cdot (1 - y_i) \cdot \sum_{l=1}^{H_{k+1}} \delta_l^{k+1} \cdot w_l^{k+1} \quad -$$

для проміжних шарів,

де t – номер поточної ітерації циклу навчання (номер епохи);

η – коефіцієнт навчання задається від 0 до 1;

y_i^k – вихід i -нейрона k -шару;

w_{ij}^k – синаптичний зв'язок між j -нейроном шару $k-1$ і i -нейроном шару k ;

d_i – бажане вихідне значення на i -нейроні;

y_i – реальне значення на i -нейроні вихідного шару.

5 крок: перевірка умови продовження навчання (обчислення значення помилки ε / або перевірка заданого кількості ітерацій). Якщо навчання не завершено, то 2 крок, інакше закінчуємо навчання. Ефективне значення помилка обчислюється таким чином:

$$\varepsilon = \frac{1}{Q} \cdot \sum_{q=1}^Q \sum_{i=1}^H (d_i - y_i)^2,$$

де Q – загальна кількість прикладів;
 N – кількість нейронів у вихідному шарі;
 d_i – бажане вихідне значення на i -нейроні;
 y_i – реальне значення на i -нейроні вихідного шару.

3.4 Завдання, які вирішуються нейронними мережами

1. **Класифікація образів.** Завдання полягає у визначенні приналежності вхідного образу (наприклад, мовного сигналу або рукописного символу), представленого вектором ознак до одного або декількох попередньо визначених класам. До відомих додатків відносяться розпізнавання букв, розпізнавання мови, класифікація сигналу електрокардіограми, класифікація клітин крові..

2. **Кластеризація / категоризація.** При вирішенні задачі кластеризації навчальну множину не має міток класів. Алгоритм кластеризації заснований на подібності образів і поміщає схожі образи в один кластер. Відомі випадки застосування кластеризації для видобутку знань, стиснення даних і дослідження властивостей даних.

3. **Апроксимація функцій.** Припустимо, що є навчальна вибірка $((x_1, y_1), (x_2, y_2) \dots, (x_n, y_n))$ (пари даних вхід-вихід), яка генерується невідомою функцією F , спотвореної шумом. Завдання апроксимації полягає в знаходженні невідомої функції F . Апроксимація функцій необхідна при вирішенні численних інженерних і наукових задач моделювання.

4. **Передбачення / прогноз.** Нехай задані n дискретних відліків $\{y(t_1), y(t_2), \dots, y(t_n)\}$ в послідовні моменти часу t_1, t_2, \dots, t_n . Завдання полягає в передбаченні значення $y(t_{n+1})$ в наступний момент часу t_{n+1} . Передбачення / прогноз мають велике значення для прийняття рішень в бізнесі, науці і техніці (передбачення цін на фондовій біржі, прогноз погоди).

5. **Оптимізація.** Численні проблеми в математиці, статистиці, техніці, науці, медицині та економіці можуть розглядатися як проблеми оптимізації. Завданням алгоритму

оптимізації є знаходження такого рішення, яке задовольняє системі обмежень і максимізує або мінімізує цільову функцію.

6. **Пам'ять, що адресується за змістом.** У традиційних комп'ютерах звернення до пам'яті доступно тільки за допомогою адреси, що не залежить від змісту пам'яті. Більш того, якщо допущена помилка в обчисленні адреси, то може бути знайдена зовсім інша інформація. Асоціативна пам'ять або пам'ять, що адресується за змістом, доступна за вказівкою заданого змісту. Вміст пам'яті може бути викликано навіть по частковому входу або пошкодженому змісті. Асоціативна пам'ять може бути використана в мультимедійних інформаційних базах даних.

7. **Управління.** Розглянемо динамічну систему, задану сукупністю $\{u(t), y(t)\}$, де $u(t)$ – вхідний керуючий вплив, а $y(t)$ – вихід системи в момент часу t . У системах управління з еталонною моделлю метою управління є розрахунок такого вхідного впливу $u(t)$, при якому система діє по бажаній траєкторії, заданій еталонною моделлю. Прикладом є оптимальне управління двигуном.

Контрольні питання

1. Нейронні мережі як основний тип сучасних моделей ШІ.
2. Структура і функції центральної нервової системи.
3. Біологічний нейрон. Формальний нейрон.
4. Функціонування формального нейрона.
5. Поняття нейронної мережі.
6. Класифікація топологій нейронних мереж.
7. Завдання, які вирішуються НС.

Лекція 4. ЕВОЛЮЦІЙНЕ МОДЕЛЮВАННЯ

План

4.1 Генетичні алгоритми

4.2 Схема функціонування генетичного алгоритму

4.3 Види генетичних алгоритмів

Однією з головних характеристик штучного інтелекту як науки є його міждисциплінарність, що дозволяє залучати цікаві ідеї, теорії з інших областей знань, адаптувати і використовувати готові розробки для своїх завдань. Так було з нейронними мережами, з моделюванням міркувань, з комп'ютерною лінгвістикою тощо. Багато значущі теорії науки були так чи інакше буде переглянуто через призму штучного інтелекту. Теорія Ч. Дарвіна (1859 г.) стала відправною точкою для ще одного напрямку досліджень – еволюційного моделювання.

Основна теза еволюційного моделювання – замінити процес моделювання складного об'єкта моделюванням його еволюції. Він спрямований на застосування механізмів природної еволюції при синтезі складних систем обробки інформації. Дарвін сформулював основний закон розвитку органічного світу, охарактеризувавши його взаємодією трьох наступних факторів:

- спадковість (нащадки зберігають властивості батьків);
- мінливість (нащадки майже завжди не ідентичні);
- природний відбір (виживають найбільш пристосовані).

Теорія Дарвіна, доповнена генетичними знаннями, називається синтетичної теорією еволюції. Випадкову появу нових ознак вона пояснила мутаціями – змінами, що виникають в ДНК організмів.

Поняття «еволюційне моделювання» сформувалося в роботах Л. Фогеля, А. Оуена, М. Уолша. У 1966 році вийшла їхня спільна книга «Штучний інтелект і еволюційне моделювання». Історія еволюційних обчислень почалася з розробки ряду різних незалежних моделей. Основними з них

були генетичні алгоритми і класифікаційні системи Д. Холланда (Holland), опубліковані на початку 60-х років і одержали загальне визнання після виходу в світ книги, що стала класикою в цій області, – «Адаптація в природних і штучних системах» («Adaptation in Natural and Artificial Systems», 1975). У 70-х роках в рамках теорії випадкового пошуку Л.А. Растригина був запропонований ряд алгоритмів, що використовують ідеї біонічного поведінки особин. Розвиток цих ідей знайшло відображення в циклі робіт І. Л. Букатовой по еволюційному моделювання. Розвиваючи ідеї М. Л. Цетлін про доцільний і оптимальному поведінці стохастичних автоматів, Ю. І. Неймарк запропонував здійснювати пошук глобального екстремуму на основі колективу незалежних автоматів, що моделюють процеси розвитку та елімінації особин. Незважаючи на різницю в підходах, всі вони базувалися на принципах еволюції.

В рамках еволюційного моделювання створювалися і досліджувалися моделі походження молекулярно-генетичних систем обробки інформації, моделі, що характеризують загальні закономірності еволюційних процесів, і проводився аналіз моделей штучної «еволюції» з метою застосування методу еволюційного пошуку до практичних завдань оптимізації.

На початку 70-х років лауреат Нобелівської премії М. Ейген зробив вражаючу спробу побудови моделей виникнення в ранній біосфері Землі молекулярно-генетичних систем обробки інформації (в моделі квазівидів розглядається поетапна еволюція популяції інформаційних послідовностей (векторів), компоненти яких беруть невелике число дискретних значень.).

Слідом за Ейген в 1980-му новосибірські вчені В. Ратнером і В. Шамінім була запропонована модель «сайзер» (модель сайзер в найпростішому випадку розглядає систему з трьох типів макромолекул: полінуклеотидних матриці і ферментів трансляції та реплікації, кодованих цієї матрицею, полінуклеотидная матриця – це як би пристрій, в якому зберігається інформація про функціональних одиницях сайзера – ферментах.). С. Кауфман зі співробітниками з Пенсильванського

університету досліджує еволюцію автоматів, що складаються із сполучених між собою логічних елементів.

Д. Коза розробив метод, що дозволяє удосконалювати реальні технічні системи методами генетичного програмування. При цьому еволюція зачіпає не окремі чисельні параметри, а цілі системи (за допомогою цього методу вдалося заново відкрити 15 запатентованих схемотехнік: підсилювачі, фільтри, контролери і т. д.).

Переваги еволюційних обчислень:

- 1) широка область застосування;
- 2) можливість проблемно-орієнтованого кодування рішень, підбору початкової популяції, комбінування еволюційних обчислень з нееволуційним алгоритмами, продовження процесу еволюції до тих пір, поки є необхідні ресурси;
- 3) придатність для пошуку в складному просторі рішень великої розмірності;
- 4) відсутність обмежень на вид цільової функції;
- 5) ясність схеми і базових принципів еволюційних обчислень;
- 6) інтегрованість еволюційних обчислень з іншими неklasичними парадигмами штучного інтелекту, такими як штучні нейромережі і нечітка логіка.

Недоліки еволюційних обчислень:

- 1) евристичний характер еволюційних обчислень не гарантує оптимальності отриманого рішення;
- 2) відносно висока обчислювальна трудомісткість, яка долається за рахунок розпаралелювання на рівні організації еволюційних обчислень і на рівні їх безпосередньої реалізації в обчислювальній системі;
- 3) відносно невисока ефективність на заключних фазах моделювання еволюції (оператори пошуку в еволюційних алгоритмах не орієнтовані на швидке потрапляння в локальний оптимум);
- 4) невирішеність питань самоадаптації.

До методів еволюційного моделювання відносяться

- метод групового урахування аргументів
 - 1) береться самий останній шар класифікаторів;
 - 2) генерується з них за певними правилами новий шар класифікаторів, які тепер самі стають останнім шаром;
 - 3) відбирається з них F кращих, де F – ширина відбору (селекції);

якщо одна з вимог припинення селекції (наступ виродження), перехід на п. 1, інакше кращий класифікатор оголошується шуканим рішенням задачі ідентифікації;

- еволюційне (генетичне) програмування (дані, які закодовані в генотипі, можуть являти собою команди будь-якої віртуальної машини, в найпростішому випадку нічого не змінюється в генетичному алгоритмі, але тоді довжина одержуваної послідовності дій (програми) виходить не відрізняється від початкових, сучасні алгоритми генетичного програмування поширюють генетичні алгоритми для систем зі змінною довжиною генотипу);

- генетичні алгоритми.

4.1 Генетичні алгоритми

«Батьком» генетичних алгоритмів по праву вважається Д. Холланд, метод спочатку називався репродуктивним планом Холланда. Надалі генетичні алгоритми розвивалися в роботах учнів Холланда: Д. Голдберга. і К. Де Йонга, – саме в них і закріпилася назва методу.

Генетичні алгоритми – це розділ еволюційного моделювання, запозичують методичні прийоми з теоретичних положень генетики.

Генетичні алгоритми – адаптивні методи пошуку, які використовуються для вирішення задач функціональної оптимізації. Являють собою свого роду моделі машинного дослідження пошукового простору, побудоване на еволюційній метафорі. Характерні особливості: використання рядків фіксованої довжини для подання генетичної інформації, робота

з населенням рядків, використання генетичних операторів для формування майбутніх поколінь.

Генетичні алгоритми оперують сукупністю особин (популяцією), які представляють собою рядки, які кодують одне з рішень задачі. Цим метод відрізняється від більшості інших алгоритмів оптимізації, які оперують лише з одним рішенням, покращуючи його.

Генетичні алгоритми застосовуються для вирішення наступних завдань:

- оптимізація функцій;
- різноманітні завдання на графах (завдання комівояжера, розфарбування і т. д.);
- настройка і навчання штучної нейронної мережі;
- завдання компонування;
- складання розкладів;
- ігрові стратегії;
- апроксимація функцій;
- штучне життя;
- біоінформатика.

переваги генетичних алгоритмів:

- 1) універсальність;
- 2) висока оглядовість пошуку;
- 3) немає обмежень на цільову функцію;
- 4) будь-який спосіб завдання функції.

недоліки генетичних алгоритмів:

- 1) відносно висока обчислювальна вартість;
- 2) Квазіоптимальний.

Коли треба використовувати генетичний алгоритм: багато параметрів, погана цільова функція, комбінаторні задачі.

Коли не треба використовувати генетичний алгоритм: задачі добре вирішуються традиційними методами, потрібна висока точність рішення.

4.2. Схема функціонування генетичного алгоритму

З біології відомо, що будь-який організм може бути представлений своїм фенотипом, який фактично визначає, чим є об'єкт в реальному світі, і генотипом, який містить всю інформацію про об'єкт на рівні хромосомного набору. При цьому кожен ген, тобто елемент інформації генотипу, має своє відображення в фенотипі. Таким чином, для вирішення завдань необхідно представити кожен ознаку об'єкта в формі, придатній для використання в генетичному алгоритмі. Все подальше функціонування механізмів генетичного алгоритму проводиться на рівні генотипу, дозволяючи обійтися без інформації про внутрішню структуру об'єкта, що й обумовлює його широке застосування в самих різних завданнях.

У найбільш часто зустрічається різновиди генетичного алгоритму для подання генотипу об'єкту застосовуються бітові рядки. При цьому кожному атрибуту об'єкта в фенотипі відповідає один ген в генотипі об'єкта. Ген являє собою бітову рядок, найчастіше фіксованої довжини, яка представляє собою значення цієї ознаки. Для кодування ген в бінарній реалізації генетичного алгоритму часто використовують код Грея (див. Приклад в табл.4.1).

Таблиця 4.1 – Приклад фенотипу

ознака	Двійкове значення ознаки	Десяткове значення ознаки
ознака 1	0011	3
ознака 2	1100	12
ознака 3	1110	14
ознака 4	0111	7
ознака 5	1001	9

Після визначення фенотипу генетичний алгоритм функціонує за наступною схемою дій (рис. 4.1).

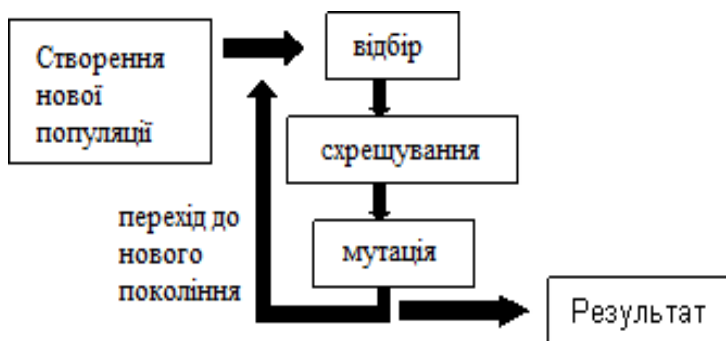


Рисунок 4.1 – Схема функціонування генетичного алгоритму

1. Формування початкової популяції.
2. Оцінка особин популяції.
3. Відбір (селекція).
4. Схрещування.
5. Мутація.
6. Формування нової популяції.
7. Якщо популяція не зійшлася, то 2, інакше – останов (припинення функціонування генетичного алгоритму).

Формування початкової популяції

Стандартний генетичний алгоритм починає свою роботу з формування початкової популяції – кінцевого набору допустимих рішень задачі. Ці рішення можуть бути обрані випадковим чином або отримані за допомогою простих наближених алгоритмів. Вибір початкової популяції не має значення для збіжності процесу, проте формування «хорошою» початкової популяції (наприклад, з множини локальних оптимумів) може помітно скоротити час досягнення глобального оптимуму. Якщо відсутні припущення про місцезнаходження глобального оптимуму, то індивіди з початкової популяції бажано розподілити рівномірно по всьому простору пошуку рішення.

Популяція ініціюється в початковий момент часу $t = 0$ і складається з k особин, кожна з яких представляє можливе рішення даної проблеми. Особина представляє собою одну або кілька хромосом (зазвичай одну). Хромосома складається з ген, – тобто це бітова рядок (хромосоми не обмежені бінарним представленням, є реалізації генетичного алгоритму, побудовані на векторах дійсних чисел). Гени розташовуються в різних позиціях хромосоми, і приймають значення, звані алелями.

Оцінка особин популяції

Для вирішення завдання за допомогою генетичного алгоритму необхідно задати міру якості для кожного індивіда в просторі пошуку. Для цієї мети використовується функція пристосованості (fitness function). Функція пристосованості повинна приймати невід'ємні значення на обмеженій області визначення, при цьому абсолютно не потрібні безперервність і дифференцируемість, значення цієї функції визначає, наскільки добре підходить особина для вирішення завдання.

У завданнях максимізації цільова функція часто сама виступає в якості функції пристосованості, для задач мінімізації цільову функцію слід інвертувати. Якщо можна вирішити завдання має обмеження, виконання яких неможливо контролювати алгоритмічно, то функція пристосованості, як правило, включає також штрафи за невиконання обмежень (вони зменшують її значення).

Відбір (селекція)

На кожному кроці еволюції за допомогою імовірнісного оператора селекції (відбору) вибираються два рішення-батька для їх подальшого схрещування. Серед операторів селекції найбільш поширеними є два імовірнісних оператора пропорційної і турнірної селекції. У деяких випадках також застосовується відбір урізанням.

Пропорційний відбір (Proportional selection)

Кожної особини призначає ймовірність $P_s(i)$, що дорівнює відношенню її пристосованості до сумарної пристосованості

популяції. Потім відбувається відбір (із заміщенням) усіх n особин для подальшої генетичної обробки, відповідно до величини $P_s(i)$.

Найпростіший пропорційний відбір – рулетка – відбирає особин за допомогою n «запусків» рулетки. Колесо рулетки містить по одному сектору для кожного члена популяції. Розмір i -ого сектора пропорційний відповідно до величини $P_s(i)$. При такому відборі члени популяції з більш високою пристосованістю з більшою ймовірністю будуть частіше вибиратися, ніж особини з низькою пристосованістю.

Турнірний відбір

Турнірний відбір реалізується в такий спосіб: з популяції, що містить m особин, вибирається випадковим чином t особин і найбільш пристосована особина записується в проміжний масив (між обраними особами проводиться турнір). Ця операція повторюється m раз. Рядки в отриманому проміжному масиві потім використовуються для схрещування (випадковим чином). Розмір групи рядків, що відбираються для турніру, часто дорівнює 2. У цьому випадку говорять про довічний / парному турнірі.

Відбір урізанням

Дана стратегія використовує відсортовану по спадаючій популяцію. Число особин для схрещування вибирається відповідно до порогу $T \in [0; 1]$. Поріг визначає, яка частка особин, починаючи з найпершої (самої пристосованої), братиме участь у відборі. В принципі, поріг можна задати і рівним 1, тоді все особини поточної популяції будуть допущені до відбору. Серед особин, допущених до схрещування випадковим чином $m / 2$ раз, вибираються батьківські пари, нащадки яких утворюють нову популяцію.

Рангові відбір

Цей вид відбір має на увазі наступне: для кожної особини її ймовірність потрапити в проміжну популяцію пропорційна її порядковому номеру в відсортованої по зростанню пристосованості популяції. Такий вид відбору не залежить від середньої пристосованості популяції.

Елітний відбір

Елітні методи відбору гарантують, що при відборі обов'язково будуть виживати кращий або кращі члени популяції. Найбільш поширена процедура обов'язкового збереження тільки однієї кращої особини, якщо вона не пройшла, як інші, через процес відбору, кросовера і мутації. Елітизм може бути впроваджений практично в будь-який стандартний метод відбору. Використання елітизму дозволяє не втратити гарне проміжне рішення, але в той же час з-за цього алгоритм може «застрягти» в локальному екстремумі. У більшості випадків елітизм не шкодить пошуку рішення, і головне – це надати алгоритму можливість аналізувати різні хромосоми з простору пошуку.

Схрещування

Як відомо, в теорії еволюції важливу роль відіграє те, яким чином ознаки батьків передаються нащадкам. У генетичних алгоритмах за передачу ознак батьків нащадкам відповідає оператор, який називається схрещування (його також називають кросовер або кросинговер). Цей оператор визначає передачу ознак батьків нащадкам. До них застосовується імовірнісний оператор схрещування, який будує на їх основі нові (1 або 2) рішення-нащадка. Відібрані особи піддаються кросоверу (іноді званого рекомбінацією) із заданою вірогідністю P_c . Якщо кожна пара батьків породжує двох нащадків, для відтворення популяції необхідно схрестити $m / 2$ пари. Для кожної пари з імовірністю P_c застосовується кросовер. Відповідно, з імовірністю $1 - P_c$ кросовер не відбувається – і тоді незмінені особини переходять на наступну стадію (мутації).

Існує велика кількість різновидів оператора схрещування. Найпростіший одноточковий кросовер працює наступним чином.

Спочатку випадковим чином вибирається одна з можливих точок розриву. (Точка розриву – ділянка між сусідніми бітами в рядку.) Обидві батьківські структури розриваються на два

сегменти по цій точці. Потім відповідні сегменти різних батьків склеюються і виходять два генотипу нащадків.

Батько 1 1 0 0 1 0 1 1 | 0 1 0 0 1 **Нащадок 1** 1 0 0 1 0 1 1 | 0 0 1 1 1

Батько 2 0 1 0 0 0 1 1 | 0 0 1 1 1 **Нащадок 2** 0 1 0 0 0 1 1 | 0 1 0 0 1

В даний час дослідники ГА пропонують багато інших операторів схрещування. Двоточковий кросовер (і рівномірний кросовер – цілком гідні альтернативи одноточкового оператору. У двоточковому кросовері обираються дві точки розриву, і батьківські хромосоми обмінюються сегментом, який знаходиться між двома цими точками. У рівномірному кросовері кожен біт першого нащадка випадковим чином успадковується від одного з батьків; другому нащадку дістається біт другого з батьків.

Мутація

Наступний генетичний оператор призначений для того, щоб підтримувати різноманітність особин з популяції, – це оператор мутації. Після того, як закінчиться стадія кросовера, нащадки можуть піддаватися випадковим модифікаціям. У найпростішому випадку в кожній хромосомі, яка піддається мутації, кожен біт з імовірністю P_m змінюється на протилежний (це так звана одноточковий мутація).

Особина до мутації: 1 0 0 1 0 1 1 0 0 1 1 1

Особина після мутації: 1 0 0 1 0 1 0 0 0 1 1 1

Складнішою різновидом мутації є оператори інверсії і транслокації. Інверсія – це перестановка генів в зворотному порядку всередині навмання вибраної ділянки хромосоми.

Особина до інверсії: 1 0 0 1 1 1 1 0 0 1 1 1

Особина після інверсії: 1 0 0 1 0 0 1 1 1 1 1 1

Транслокація – це перенесення якої-небудь ділянки хромосоми в інший сегмент цієї ж хромосоми.

Особина до транслокації: 1 0 0 1 1 1 1 0 0 1 1 1

Особина після транслокації: 1 1 1 0 0 0 1 1 0 1 1 1

Всі перераховані генетичні оператори (одноточковий і многоточечний кросовер, одноточковий мутація, інверсія, транслокація) мають схожі біологічні аналоги.

Формування нового покоління

Після схрещування і мутації особин необхідно вирішити проблему: які з нових особин увійдуть в наступне покоління, а будь – немає, і що робити з їхніми предками. Є два найпоширеніші способи.

1. Нові особини (нащадки) займають місця своїх батьків. Після цього настає наступний етап, в якому нащадки оцінюються, відбираються, дають потомство і поступаються місцем своїм «дітям».

2. Наступна популяція включає в себе як батьків, так і їх нащадків.

У другому випадку необхідно додатково визначити, які із особин батьків та нащадків потраплять в нове покоління. У найпростішому випадку, в нього після кожного схрещування включаються дві кращих особини з четвірки батьків та їхніх нащадків. Більш ефективним є механізм витіснення, який реалізується таким чином, що прагне видаляти «схожі» хромосоми з популяції і залишати відрізняються.

Критерії зупинки

Інший важливий момент функціонування алгоритму – визначення критеріїв зупинки. Взагалі кажучи, такий процес еволюції може тривати до нескінченності. Зазвичай в якості них застосовуються або обмеження на максимальне число епох функціонування алгоритму, або визначення його збіжності, зазвичай шляхом порівнювання пристосованості популяції на кількох епохах і зупинки при стабілізації цього параметра.

Сходженням називається такий стан популяції, коли всі рядки популяції майже однакові і знаходяться в області деякого екстремуму (рис. 4.2).

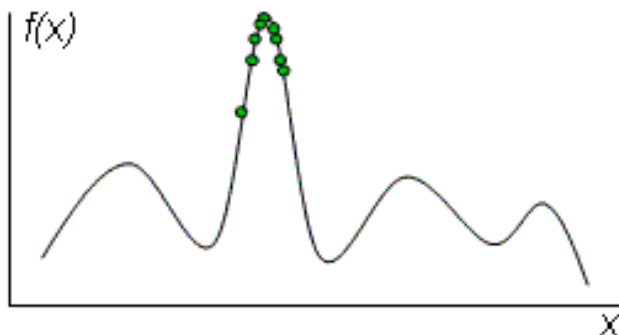


Рисунок 4.2 – Сходження генетичного алгоритму

У такій ситуації кросовер практично ніяк не змінює популяції. А що вийшли з цієї області за рахунок мутації особини схильні вимирати, так як частіше мають меншу пристосованість, особливо якщо цей екстремум є глобальним максимумом. Таким чином, сходження популяції зазвичай означає, що знайдене краще або близьке до нього рішення.

4.3 Види генетичних алгоритмів

Існують різні моделі генетичного алгоритму (класичний, простий генетичний алгоритм, гібридний, СНС генетичний алгоритм та ін.) Вони розрізняються по стратегіям відбору та формування нового покоління особин, операторами генетичного алгоритму, кодуванням ген і т. д.

СНС-алгоритм

СНС (Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation) був запропонований Ешхелманом і характеризується наступними параметрами:

1. Для нового покоління вибираються N кращих різних особин серед батьків і дітей. Дублювання рядків не допускається.
2. Для схрещування вибирається випадкова пара, але не допускається, щоб між батьками було мало Хеммінгово відстань або мало відстань між крайніми розрізняються битами.

3. Для схрещування використовується різновид однорідного кросовера HUX (Half Uniform Crossover): дитині переходить рівно половина бітів кожного з батьків.

4. Розмір популяції невеликий, близько 50 особин. Цим виправдано використання однорідного кросовера.

СНС протиставляє агресивний відбір агресивному кросоверу, проте все одно малий розмір популяції швидко викликають стан, коли створюються тільки більш-менш однакові рядки. В такому випадку СНС застосовує cataclysmic mutation: всі рядки, крім самої пристосованої, піддаються сильній мутації (змінюється близько третини бітів). Таким чином, алгоритм перезапускається і далі продовжує роботу, застосовуючи тільки кросовер.

Genitor

Цей алгоритм був створений Д. Уитли. Genitor-подібні алгоритми відрізняються від класичного ГА наступними трьома властивостями:

1. На кожному кроці тільки одна пара випадкових батьків створює тільки одну дитину.

2. Ця дитина замінює не батьки, а одну з найгірших особин популяції (в первісному Genitor – найгіршу).

3. Відбір особини для заміни проводиться по її ранку (рейтингу), а не по пристосованості.

У Genitor пошук гіперплоскостей відбувається краще, а збіжність швидше, ніж у класичного генетичного алгоритму, запропонованого Холландом.

Гібридні алгоритми

Ідея гібридних алгоритмів (hybrid algorithms) полягає в поєднанні генетичного алгоритму з деяким іншим методом пошуку, відповідним в даному завданні. На кожному поколінні кожен отриманий нащадок оптимізується цим методом, після чого виробляються звичайні для генетичного алгоритму дії.

Такий вид розвитку називається Ламарковою еволюцією, при якій особина здатна навчатися, а потім отримані навички

записувати в свій генотип, щоб потім передати їх нащадкам. І хоча такий метод погіршує здатність алгоритму шукати рішення за допомогою відбору гіперплоскостей, однак на практиці гібридні алгоритми виявляються дуже вдалими. Це пов'язано з тим, що зазвичай велика ймовірність того, що одна з особин потрапить в область глобального максимуму і після оптимізації виявиться рішенням завдання.

Паралельні генетичні алгоритми

Генетичні алгоритми можна організувати як кілька паралельно виконуються процесів, це збільшить їх продуктивність.

Розглянемо перехід від класичного генетичного алгоритму до паралельного. Для цього будемо використовувати турнірний відбір. Зведемо $N/2$ процесів (тут і далі процес мається на увазі як деяка машина, процесор, який може працювати незалежно). Кожен з них буде вибирати випадково з популяції 4 особи, проводити 2 турніру і схрещувати переможців. Отримані діти будуть записуватися в нове покоління. Таким чином, за один цикл роботи одного процесу буде змінюватися ціле покоління.

Острівна модель

Острівна модель (island model, рис. 17) – це теж модель паралельного генетичного алгоритму. Вона полягає в наступному: нехай у нас є 16 процесів і 1600 особин. Розіб'ємо їх на 16 підпопуляцій по 100 особин. Кожна з них буде розвиватися окремо за допомогою якогось генетичного алгоритму. Таким чином, можна сказати, що ми розселили особини по 16-ти ізольованим островам.

Зрідка (наприклад, кожні 5 поколінь) процеси (або острова) обмінюватимуться декількома хорошими особинами. Це називається міграція. Вона дозволяє островам обмінюватися генетичним матеріалом.

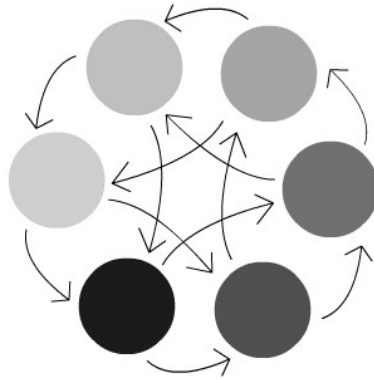


Рисунок 4.3 – Острівна модель генетичного алгоритму

Так як населеність островів зазвичай буває невелика, підпопуляції будуть схильні до передчасної збіжності. Тому важливо правильно встановити частоту міграції. Надто часта міграція (або міграція занадто великого числа особин) призведе до змішання всіх підпопуляцій, та тоді острівна модель буде несильно відрізнятися від звичайного генетичного алгоритму. Якщо ж міграція буде занадто рідкою, то вона не зможе запобігти передчасного сходження підпопуляцій.

Генетичні алгоритми стохастичного, тож при різних його запусках популяція може сходитися до різним рішенням (хоча всі вони в деякій мірі «хороші»). Острівна модель дозволяє запустити алгоритм відразу кілька раз і намагатися поєднувати «досягнення» різних островів для отримання в одній з підпопуляцій найкращого рішення.

Ніздрюваті генетичні алгоритми

Ніздрюваті генетичні алгоритми (Cellular Genetic Algorithms) – модель паралельних генетичних алгоритмів. Нехай дано 2500 процесів, які розташовані на сітці розміром 50×50 осередків, замкнутої, як показано на малюнку 18 (ліва сторона замикається з правого, верхня з нижньої, виходить тор).

Кожен процес може взаємодіяти тільки з чотирма своїми сусідами (зверху, знизу, зліва, справа). У кожному осередку знаходиться рівно одна особина. Кожен процес обиратиме

найкращу особина серед своїх сусідів, схрещувати з нею особина з свого осередку і одного отриманого дитини поміщати в свій осередок замість батька.

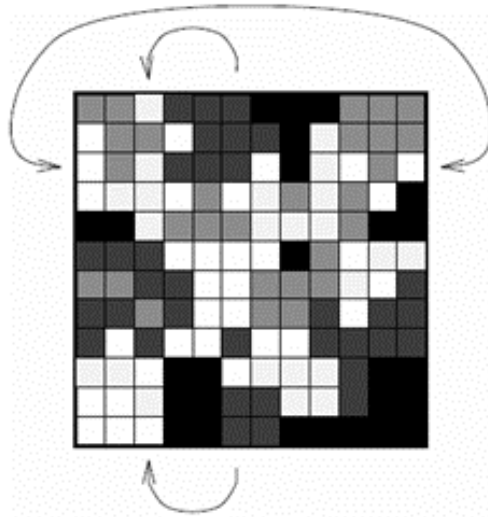


Рисунок 4.4 – Ніздрюватий генетичний алгоритм

У міру роботи такого алгоритму виникають ефекти, схожі на острівну модель. Спочатку все особини мають випадкову пристосованість (на малюнку вона визначається за кольором). Через кілька поколінь утворюються невеликі області схожих особин з близької пристосованістю. У міру роботи алгоритму ці області ростуть і конкурують між собою.

Контрольні питання

1. Поясніть роботу генетичних алгоритмів
2. Для вирішення яких задач використовуються генетичні алгоритми?
3. Наведіть приклади застосування генетичних алгоритмів в медицині.
4. Які Ви знаєте недоліки генетичних алгоритмів?
5. Із яких етапів складаються генетичні алгоритми?
6. Для чого використовується мутація в генетичних алгоритмах?

Лекція 5. ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ. ЕКСПЕРТНІ СИСТЕМИ

План

5.1 Модель експертних систем

5.2. Класифікація експертних систем і оболонок експертних систем

5.3 Засоби розробки експертних систем

На початкових етапах розвитку штучний інтелект піддавався жорсткій критиці і одним з доводів був той, що практичної користі від нього немає, він займається іграшками. Експертні системи одними з перших довели практичну користь цього наукового напрямку, приносячи на початку 80-х років ХХ століття комерційний прибуток своїм творцям.

Термін «системи, засновані на знаннях» (knowledge-based systems) з'явився в 1976 році одночасно з першими системами, акумулюють досвід і знання експертів.

Експертні системи – це прикладні системи штучного інтелекту, в яких база знань являє собою формалізовані емпіричні знання висококваліфікованих фахівців (експертів) в будь-якої вузької предметної області, вони акумулюють ці знання і тиражують їх для консультації менш кваліфікованих фахівців.

У 1965 році Е. Фейгенбаум (колишній студент Герберта Саймона), Б. Б'юкенен (філософ за освітою) і Д. Ледербергом (лауреат Нобелівської премії в області генетики) почали роботи над першою експертною системою DENDRAL. У 1969 році була створена математична експертна система MACSYMA В. Мартіном і Д. Мозесом. Перша експертна система для медичної діагностики була створена в 1973 році Е. Шортліффом і називалася MYCIN, вона спричинила за собою розробку першого командного інтерпретатора EMYCIN (В. Мілле, Шортліфф і Б'юкенен). У 1976 році Дуда і Харт почали роботу

над експертною системою PROSPECTOR, призначеної для розвідки корисних копалин, в 1984 році система точно передбачила існування родовища молібдену, оціненого в багатомільйонну суму.

Ці експертні системи, розроблені в 60-70-х роках стали класичними. За походженням, предметних областях і по наступності застосовуваних ідей, методів і інструментальних програмних засобів їх можна розділити на кілька сімейств.

1. META-DENDRAL. Система DENDRAL дозволяє визначити найбільш ймовірну структуру хімічної сполуки за експериментальними даними (мас-спектрограф, даними ядерному магнітного резонансу та ін.). Meta-DENDRAL автоматизує процес придбання знань для DENDRAL. Вона генерує правила побудови фрагментів хімічних структур.

2. MYCIN-EMYCIN-TEIREIAS-PUFF-NEOMYCIN. Це сімейство медичних ЕС і сервісних програмних засобів для їх побудови.

3. PROSPECTOR-KAS. PROSPECTOR – призначена для пошуку (передбачення) родовищ на основі геологічних аналізів. KAS – система придбання знань для PROSPECTOR.

4. CASNET-EXPERT. Система CASNET – медична ЕС для діагностики видачі рекомендацій по лікуванню очних захворювань. На її основі розроблена мова інженерії знань EXPERT, за допомогою якої створено ряд інших медичних діагностичних систем.

5. HEARSAY-HEARSAY-2-HEARSAY-3-AGE. Перші дві системи цього ряду є розвитком інтелектуальної системи розпізнавання зливої людської мови, слова якої беруться із заданого словника. Ці системи відрізняються оригінальною структурою, заснованої на використанні дошки оголошень-глобальної бази даних, що містить поточні результати роботи системи. Надалі на основі цих систем були створені інструментальні системи HEARSAY-3 і AGE (Attempt to Generalize- спроба спілкування) для побудови експертних систем.

6. Системи АМ (Artificial Mathematician – штучний математик) і EURISCO були розроблені в Стенфордському університеті доктором Дугласом Ленатом для дослідницьких і навчальних цілей. У систему АМ спочатку було закладено близько 100 правил виводу і більше 200 евристичних алгоритмів навчання, що дозволяють будувати довільні математичні теорії та уявлення. EURISCO – це розвиток системи АМ, з її допомогою у військово-стратегічній грі, що проводиться ВМФ США, була розроблена стратегія, яка містить ряд оригінальних тактичних ходів.

Крім розробки самих експертних систем дослідники зайнялися створенням інструментальне засіб для експертних систем: в 1983 році компанія IntelliCorp створила КЕЕ, а в 1985 році агентство NASA випустило першу версію CLIPS.

Експертні системи швидко завоювали позиції на інформаційному ринку і набули широкого распостраненіє. Уже в 1987 році опитування користувачів, проведене журналом «Intelligent Technologies» (США), показав, що приблизно:

- 25% користувачів використовують ЕС;
- 25% збираються придбати ЕС в найближчі 2–3 роки;
- 50% вважають за краще провести дослідження про ефективність їх використання.

У Росії в дослідження і розробку ЕС великий внесок внесли роботи Д. А. Поспелова (засновника Російської асоціації штучного інтелекту і його першого президента), Е. В. Попова, В. Ф. Хорошевского, В. Л. Стефанюка, Г. С. Осипова, В. К. Фінна, В. Л. Вагіна, В. І. Городецького і багатьох інших.

Експертні системи 60-90-х років є першим поколінням експертних систем, для них характерно:

- 1) знаннями системи є тільки знання експерта, накопичення знань не передбачається;
- 2) методи представлення знань дозволяють описувати лише статичні предметні області;

3) моделі представлення знань орієнтовані на прості предметні області.

Розвиваючись, експертні системи вийшли за ці рамки. Принципи представлення знань в експертних системах другого покоління змінилися:

1) використовуються не поверхневі знання, а більш глибокі;

2) для представлення знань залучаються кошти і методи інших напрямків штучного інтелекту, наприклад, нейронних мереж;

3) системи мають динамічну базу знань.

Поява інтернету не могло не вплинути на розвиток експертних систем. Можливість отримувати знання через мережу і отримувати знання з мережі не могла не бути використана розробниками. Тому зараз розвиваються розподілені і web-орієнтовані експертні системи.

Зараз кількість експертних систем обчислюється тисячами і десятками тисяч. У розвинених зарубіжних країнах сотні фірм займаються їх розробкою та впровадженням в різні сфери життя.

Як сучасних ЕС можна назвати швидкодіючу систему OMEGAMON (фірма Candle, з 2004 р IBM) для відстеження стану корпоративної інформаційної мережі і G2 (фірма Gensym) – комерційну експертну систему для роботи з динамічними об'єктами.

Експертні системи використовують в тих випадках, коли недостатньо експертів, в небезпечних (шкідливих) для них умовах, в процесі навчання. Експертні системи вирішують завдання, для вирішення яких відсутні чіткі алгоритми рішення.

5.1 Модель експертних систем

Експертні системи працюють в діалоговому режимі (відповідають на поставлені користувачем питання), при цьому, вони повинні вміти пояснювати, звідки отримано те чи інше рішення, будь-яка експертна система містить як мінімум п'ять компонентів або підсистем (рис. 6.1).

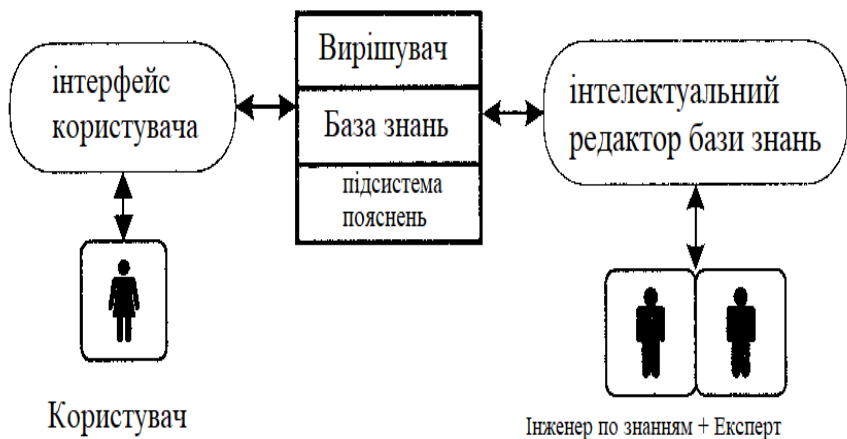


Рисунок 6.1 – Базова структура експертної системи

Користувач експертної системи – фахівець предметної області, для якого призначена система. Зазвичай його кваліфікація недостатньо висока і тому він потребує допомоги та інформаційної підтримки своєї діяльності.

Інженер по знаннях – фахівець в області штучного інтелекту, який працює з експертами і формує базу знань. Синоніми: когнітологией, інженер-інтерпретатор, аналітик.

інтерфейс користувача – комплекс програм, що реалізують діалог користувача з ЕС як на стадії введення інформації, так і при отриманні результатів.

База знань – ядро експертної системи, сукупність знань предметної області, записана на машинний носій в формі, зрозумілою експерту і користувачу (зазвичай на деякій мові, наближеному до природного).

Вирішувач – програма, що моделює хід міркувань експерта на підставі знань, наявних в базі знань. Синоніми: дедуктивна машина, машина виведення, блок логічного висновку.

Підсистема пояснень – програма, що дозволяє користувачеві отримати відповіді на питання: «Як була

отримана та чи інша рекомендація?» і «Чому система прийняла таке рішення?» Відповідь на питання «Як» – це трасування всього процесу отримання рішення із зазначенням використаних фрагментів бази знань, т. Е. Всіх кроків ланцюга умовиводів. Відповідь на питання «чому» – посилення на умовивід, безпосередньо передувала отриманому рішенню, т. Е. Відхід на один крок назад. Розвинені підсистеми пояснень підтримують і інші типи питань.

Інтелектуальний редактор бази знань – програма, що представляє інженеру по знаннях можливість створювати базу знань в діалоговому режимі. Включає в себе систему вкладених меню, шаблонів мови представлення знань, підказок і інших сервісних засобів, що полегшують роботу з базою.

Описана структура є базовою і може розширюватися (рис. 6.2).

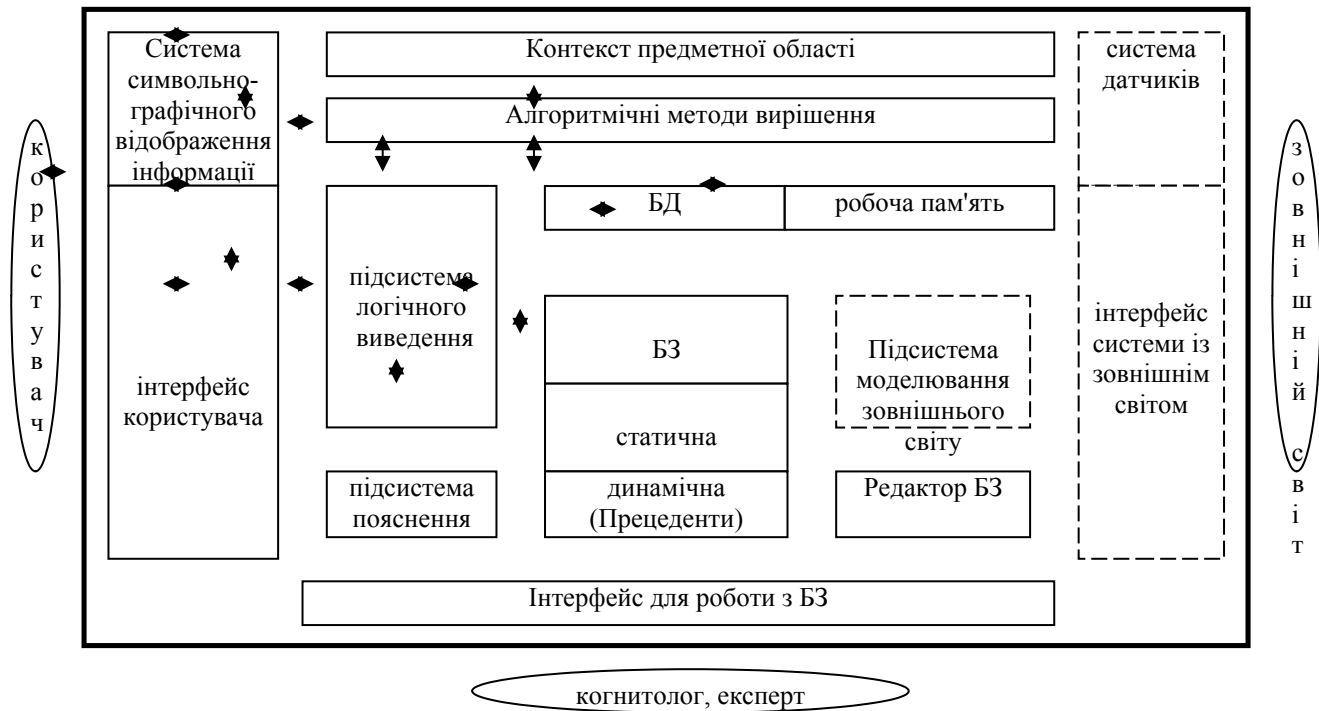


Рисунок 6.2 – Структура експертної системи

Окреслені штриховий лінією підсистеми моделювання зовнішнього світу, інтерфейс із зовнішнім світом, система датчиків – необхідні для експертних систем реального часу для отримання даних та їх інтерпретації. Експертні системи можуть накопичувати досвід у вигляді прецедентів (вже дозволених ситуацій), які зберігаються в базі знань і використовуються в подальшому.

Блок алгоритмічних методів рішення включає в себе всі обчислювальні операції і алгоритми, що реалізуються методами традиційного програмування, інтегровані в експертну систему. Об'єднання в рамках експертної системи методів традиційного програмування і штучного інтелекту дозволяє значно підвищити ефективність і якість прийнятих рішень.

Специфіка предметної області, для якої будується система, відображається описується не тільки в базі знань, а й в підсистемі «Контекст предметної області», яка дозволяє більш наочно уявити вхідну і вихідну інформацію в прийнятому для конкретної предметної області вигляді.

Сучасні інформаційні системи часто використовують архітектуру клієнт-сервер, що дозволяє будувати розподілені і мережеві додатки. При клієнт-серверній архітектурі експертної системи на стороні клієнта знаходяться інтерфейси користувача, експерта, зовнішнього середовища і система датчиків. Решта блоки розташовуються на серверній стороні.

5.2 Класифікація експертних систем і оболонок експертних систем

Існуюче безліч експертних систем ділиться на кілька класів (рис. 6.3) за різними критеріями.

За призначенням виділяють системи загального призначення, які претендують на універсальність у вирішенні завдань (CASNET), спеціалізовані, вирішальні конкретну задачу (1-st Clas, Еліс) Або орієнтовані на певну предметну область (MYCIN, MACSYMA, МОДІС, ДІАГЕН, INTERNIST-I).

За критерієм взаємодії із зовнішнім середовищем розрізняють статичні системи, в яких є тільки інтерфейс користувача, а механізму взаємодії з зовнішнім світом, наприклад, через датчики, відсутня. Динамічні експертні системи за допомогою вбудованих інтерфейсів отримують інформацію з зовнішніх датчиків або інших пристроїв. Квазідінамічні експертні системи отримують інформацію про зміну ситуації в зовнішньому середовищі через заданий проміжок часу (більше декількох секунд).

Експертні системи розробляються для різних ЕОМ і розрізняються по апаратно-програмній платформі. Вони розробляються і експлуатуються на персональних (PROSPECTOR), на символічних (Picon), на міні (СПЕІС) і на суперкомп'ютерах (ЕКСПЕРТИЗА).

За ступенем інтеграції ділять експертні системи на автономні програмні комплекси (ДАМП), які працюють самостійно, або експертна система може бути частиною більш загальної системи (інтегровані системи) або ж бути ланкою в ланцюжку програм, обробляють інформацію з спільною метою, наприклад управління підприємством.

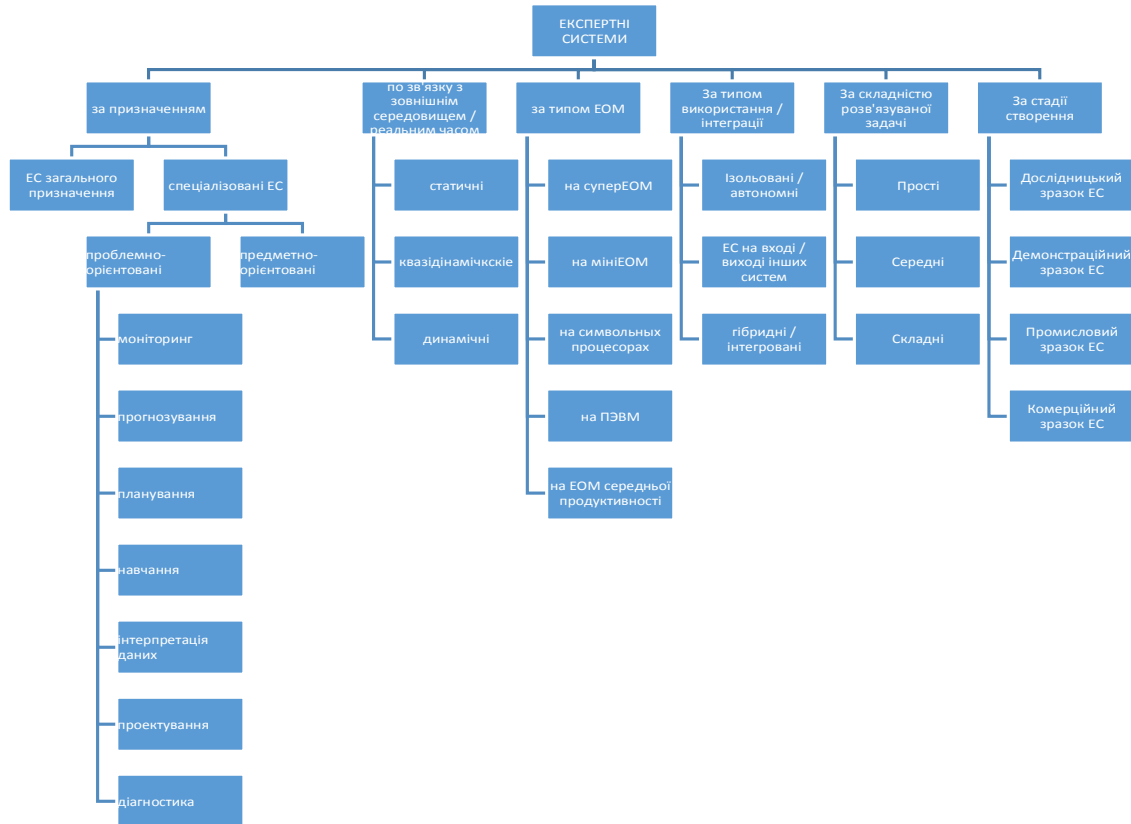


Рисунок 6.3 – Класифікація експертних систем

Залежно від розміру бази знань виділяють прості експертні системи – до 1000 простих правил (GUIDON, Гребля), – середні – від 1000 до 10000 структурованих правил (XCON, GOSSEYN, ДІАГЕН) – і складні – більше 10000 структурованих правил.

Експертні системи розрізняються по стадії існування, тобто за ступенем завершеності системи. Перша стадія існування експертної системи – це дослідницький зразок – розробляється 3-6 місяці з мінімальною базою знань (SYSTEM-D, SYN), друга – демонстраційний зразок – розробляється 6–12 місяці (THYROID MODEL), третя – промисловий зразок – розробляється 1–1,5 року з повною базою знань (PUFF, FOSSIL) і остання – комерційний зразок – розробляється 1,5–3 роки з повною базою знань (KNEECAP, MACSYMA).

5.3 Засоби розробки експертних систем

Існуючі засоби розробки експертних систем можна розділити на 3 класу (рис. 6.4). Традиційні мови програмування (C++, Java, Delphi) дозволяють побудувати експертні системи «з нуля» для конкретного завдання або предметної області, забезпечивши хороші показники якості і необхідну функціональність системи, але на розробку потрібні значні часові та фінансові ресурси. Так створюють експертні системи будь-якій стадії існування, особливо, комерційні системи, продаж яких відшкодує витрати.

Мови штучного інтелекту(LISP, PROLOG, Рефал) були розроблені спеціально для представлення знань, побудова з їх допомогою експертних систем дозволяє більш легко оперувати експертними знаннями, але обмежують спосіб їх подання структурою мови. За допомогою мов штучного інтелекту створюються дослідницькі та демонстраційні зразки.

Наступний клас засобів побудови експертних систем – спеціальний програмний інструментарій – орієнтований тільки на створення інтелектуальних інформаційних систем і ділиться

на два підкласу: оболонки і середовища розробки інтелектуальних систем.

Середовища розробки є програмними комплексами, що дозволяють будувати системи з окремих готових блоків, на їх основі створюються демонстраційні і промислові зразки експертних систем.

Оболонка експертних систем- інструментальне засіб для проектування і створення експертних систем. До складу оболонки входять засоби проектування бази знань з різними формами представлення знань і вибору режиму роботи решателя завдань. Для конкретної предметної області інженер по знаннях визначає потрібне подання знань і стратегії вирішення завдань, а потім, вводячи їх в оболонку, створює конкретну експертну систему.

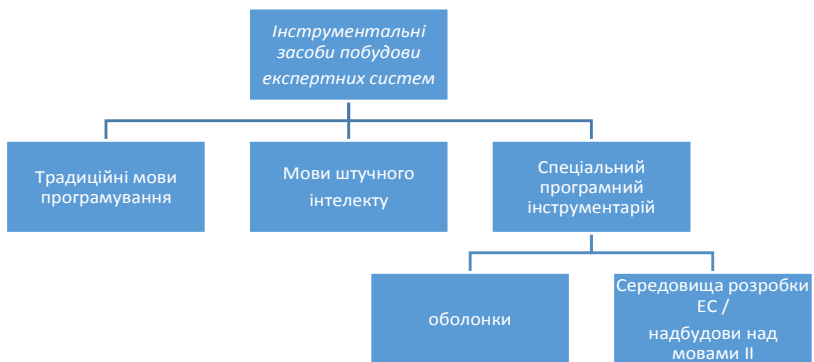


Рисунок 6.4 – Класифікація інструментальних засобів розробки ЕС

Застосування оболонки дозволяє досить швидко і з мінімальними витратами створити дослідницьку, демонстраційну або промислову експертну систему. Оболонки можна класифікувати наступним чином (рис. 6.5). За ступеня обробки виділяють експериментальну (GPSI), дослідницьку (Expert) і комерційну (EXSYS) оболонки.

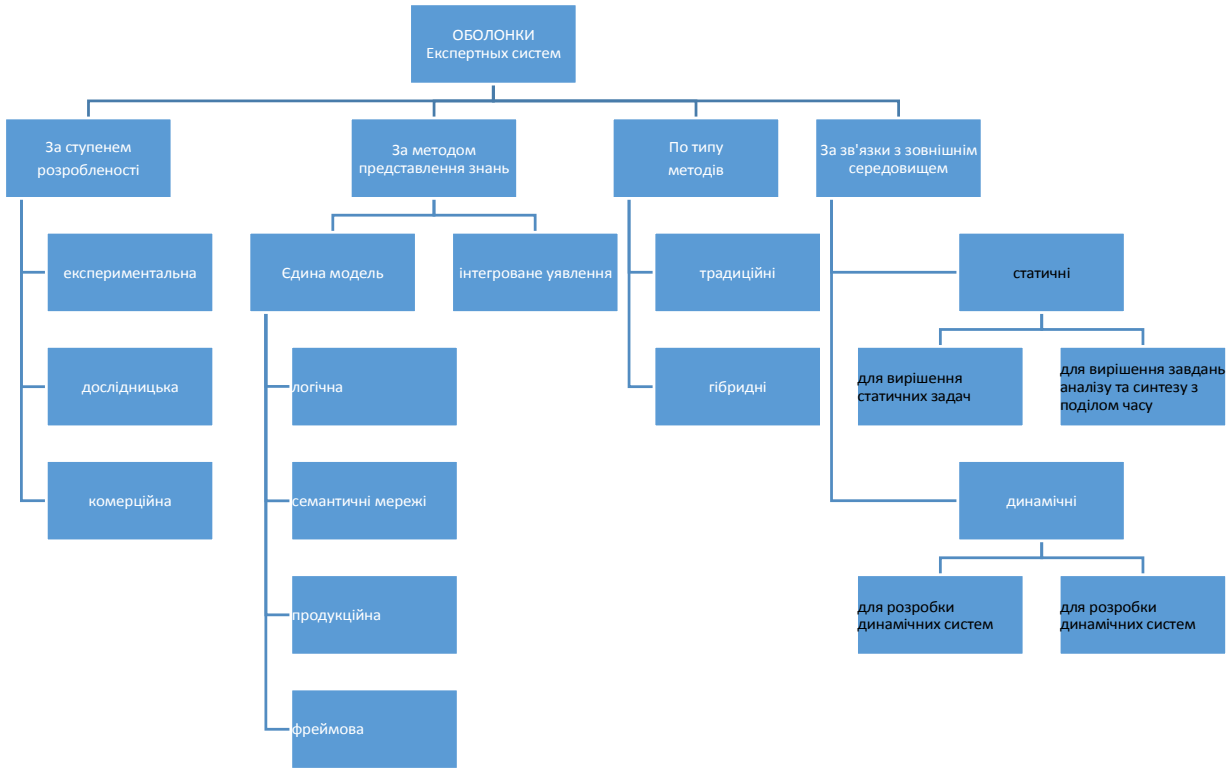


Рисунок 6.5 – Класифікація оболонок експертних систем

Знання в основі можуть бути представлені одним способом (EMYCIN, CLIPS) – семантичної мережею, продукціям, фреймами і т. д.) або ж декількома (MINEVRA, EsWin), для створення більш повної, гнучкою і наочної моделі предметної області.

Використовувані в оболонці методи можуть бути традиційними (CubiCalc, NEXPERT, Алеф) – алгоритми, дерева виводу і т. д. – і гібридними (FuzzyCLIPS, MultiNeuron) – спільно з традиційними використовуються нейронні мережі, нечітка логіка і т. д.

Існують статичні оболонки, призначені для вирішення статичних задач (1-st Clas, Еліс). Вони характеризуються використанням поверхневої технології, загальних правил і пошуку рішення від мети до даних, застосовуються для вирішення завдань аналізу.

Статичні оболонки, призначені для вирішення завдань аналізу та синтезу з поділом часу (KAPPA, Clips), використовують глибинний і структурний підходи, здійснюють пошук рішень – від мети до даних і від даних до мети.

Оболонки для проектування динамічних систем (Framework, NExpert) застосовують поверхневий підхід, приймають рішення на основі правил загального вигляду.

Оболонки для розробки динамічних систем (G2, Rethink, RkWorks) мають підсистему моделювання, планувальника рішень, використовують змішану технологію, правила загального і приватного виду, рішення задачі аналізу та синтезу в реальному часі.

EMYCIN – перша оболонка, заснована на MYCIN. Принципи, розроблені для PROSPECTOR, були використані при створенні таких систем, як KAS, SAGE, SAVOIR.

Зміна принципів побудови веде до розвитку інструментарію. Тому оболонки пройшли той же еволюційний шлях, що і ЕС. Сучасні оболонки пропонують наступні можливості (в кожній конкретній оболонці представлені частково):

- гібридне представлення знань (EsWin);
- вибір з кількох стратегій виведення (G2, CLIPS);
- підключення бібліотек та інших систем (ACTIVATION FRAMEWORK);
- архітектура на основі «дошки оголошень» (HEARSAY-III);
- архітектура «клієнт-Сервер» (JESS);
- інтеграція в Інтернет / інтранет (Egg2Lite, Exsys Corvid);
- графічний інтерфейс (WindExS, WxCLIPS);
- підсистема моделювання (G2);
- модульне побудова системи (ReThink, G2);
- візуалізація структури БЗ (WEST) і т. д.

Контрольні питання

1. Визначення, історія розвитку і області застосування експертних систем.
2. Технологія роботи з ЕС, відміну від технології застосування «звичайних» програм. Приклади практичних експертних систем.
3. Архітектура експертних систем.
4. Призначення та особливості експертних систем (ЕС).
5. Узагальнена схема структури ЕС.
6. Етапи проектування експертної системи.
7. Класифікація ЕС і сучасні тенденції їх розвитку.
8. Технологія розробки експертних систем: отримання знань, вибір моделі представлення знань, робота інженера по знаннях.

ВИСНОВОК

У другій частині дисципліни «Технології програмування», що носить назву «Навчання машин та штучний інтелект» вивчаються основи представлення знань в сучасних комп'ютерах, методологія математичного моделювання та автоматизації логічного та аналітичного мислення, відомі системи навчання машин та штучного інтелекту. У конспекті лекцій визначаються основні завдання досліджень в області НМ та ШІ, основні методи і моделі подання знань в комп'ютерах, розглядаються етапи проектування систем штучного інтелекту і розробки експертних систем.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Базові

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

Допоміжні

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

13. Акинин, М. В. Нейросетевые системы искусственного интеллекта в задачах обработки изображений / М.В. Акинин, М.Б. Никифоров, А.И. Таганов. - М.: РиС, 2016. - 152 с.

Інформаційні ресурси

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

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