**1. Фундаментальні поняття: інтелектуальна система, агент, середовище, задачі штучного інтелекту, сильний і слабкий штучний інтелект**

Звичайно, залежність на основних поняттях штучного інтелекту - це важлива перша крок для розуміння самої технології. Ось короткий огляд кожного з цих понять:

1. Інтелектуальна система: Це комп'ютерна система або програма, яка може вирішувати завдання, які зазвичай потребують інтелектуальних зусиль. Ці системи можуть використовувати різні методи та алгоритми для прийняття рішень.

2. Агент - це частина інтелектуальної системи, яка сприймає оточуюче середовище через сенсори та діє на нього через виконавчі органи. Агент може бути реалізований як програмний модуль або фізичний об'єкт.

3. Середовище: Це все, з чим взаємодіє агент. Це може бути реальний світ, віртуальне середовище або навіть інша інтелектуальна система.

4. Задачі штучного інтелекту: Це завдання, які інтелектуальні системи можуть вирішувати. Вони можуть включати в себе розпізнавання образів, вирішення проблем, планування дій, машинне навчання тощо.

5. Сильний штучний інтелект: Це концепція, за якою інтелектуальна система має здатність розуміти та діяти в будь-якому середовищі так само, як це робить людина. Це підвищений рівень штучного інтелекту, який потребує універсальної гнучкості та здатності до вирішення різноманітних завдань.

6. Слабкий штучний інтелект: На відміну від сильного штучного інтелекту, слабкий штучний інтелект використовується для вирішення конкретних завдань та обмеженої області знань. Він не має загального розуміння або свідомості. Такі системи дуже ефективні в обмежених контекстах, але не здатні адаптуватися до нових ситуацій без втручання людини.

**2. Пошук у просторі станів та подання знань**

**2.1 Стратегії пошуку у просторі станів: пошук вшир, пошук вглиб, прямий, зворотний та двонаправлений пошук.**

Розглянемо основні стратегії пошуку у просторі станів, які використовуються в інтелектуальних системах:

1. Пошук вшир: Ця стратегія полягає в тому, щоб розглядати всі можливі варіанти на кожному рівні глибини перед тим, як переходити на наступний рівень. Це означає, що система спочатку досліджує всі можливі дії з поточного стану, перш ніж переходити до дослідження наступного рівня.

2. Пошук вглиб: На відміну від пошуку вшир, ця стратегія передбачає глибше дослідження конкретного шляху перед тим, як розглядати інші альтернативи. Тобто система буде спочатку досліджувати всі можливі наслідки вибраної дії перед тим, як переходити до дослідження інших дій.

3. Прямий пошук: Ця стратегія полягає в тому, щоб здійснювати пошук від початкового стану до цільового стану без повернення або збереження інформації про проміжні кроки. Вона часто використовується в задачах, де можливий прямий шлях до розв'язку.

4. Зворотний пошук: Ця стратегія, навпаки, розглядає вибір дій від цільового стану до початкового. Вона використовується в тих випадках, коли легше знаходити шляхи до цілі, ніж з початку до кінця.

5. Двонаправлений пошук: Ця стратегія комбінує як прямий, так і зворотний пошук. Вона використовується для ефективного знаходження шляхів до цілі, розглядаючи шляхи як від початкового до цільового стану, так і в зворотньому напрямку. Це може допомогти скоротити час пошуку, знаходячи шляхи з обох напрямків і об'єднуючи їх у оптимальний шлях.

**2.2 Моделі подання знань (семантична мережа, продукційна модель).**

Розглянемо дві основні моделі подання знань:

1. Семантична мережа:

У семантичних мережах знання представляються у вигляді графа, де вузлами є поняття, а зв'язки між ними відображають взаємозв'язки між цими поняттями. Кожен вузол представляє концепт або об'єкт, а зв'язки між вузлами вказують на семантичні або логічні залежності між ними.

Наприклад, якщо ми маємо семантичну мережу для тварин, то можливі зв'язки будуть "є хижаком", "є частинкою", "є домашнім", тощо. Ця модель дозволяє представити складні відносини між різними концептами та швидко здійснювати логічні виводи.

2. Продукційна модель:

Продукційні моделі представляють знання у вигляді правил, відомих як продукції. Кожна продукція має умову (яка перевіряється) та дії (які виконуються при виконанні умови). Ці моделі особливо корисні для моделювання ситуацій, де можливі дії можна розглядати як реакції на певні умови.

Наприклад, у продукційній системі для роботи можуть бути правила на кшталт: "Якщо робот бачить перешкоду, то він повинен зупинитися" або "Якщо робот відчуває падіння тиску, то він повинен зупинитися і викликати технічне обслуговування". Ці правила можуть мати різні пріоритети та взаємодіяти між собою для прийняття рішень.

Обидві моделі мають свої переваги та недоліки і використовуються в залежності від конкретних вимог та характеру проблеми.

**3. Машинне навчання**

**3.1 Задача класифікації. Навчання з вчителем та без учителя.**

Задача класифікації - це одна з основних задач машинного навчання, яка полягає в призначенні об'єктам вхідних даних однієї або декількох міток класу на підставі їхніх характеристик. Наприклад, задача класифікації може полягати в розпізнаванні електронних листів як "спам" або "не спам", розпізнаванні зображень котів і собак або класифікації новин за їхньою темою.

Існують два основних підходи до розв'язання задачі класифікації:

1. Навчання з вчителем (Supervised Learning):

У навчанні з вчителем для кожного прикладу вхідних даних надаються відповідні мітки класу. Модель навчається на цих парах вхідних даних та міток класу з метою здійснення прогнозу для нових даних.

Під час навчання з вчителем використовуються алгоритми, які вчаться згідно з цими паровими даними, такі як алгоритми класифікації дерев рішень, метод опорних векторів (SVM), нейронні мережі тощо.

2. Навчання без вчителя (Unsupervised Learning):

У навчанні без вчителя модель навчається з невмітими мітками класу або зовсім без міток. Метою є виявлення прихованих структур у вихідних даних, таких як групи схожих об'єктів (кластеризація), зменшення розмірності даних або знаходження аномалій.

До алгоритмів навчання без вчителя входять методи кластеризації, такі як k-середніх та ієрархічна кластеризація, методи зменшення розмірності даних, такі як метод головних компонент (PCA), та алгоритми виявлення аномалій.

Обидва підходи мають свої використання залежно від конкретної задачі та доступних даних. Навчання з вчителем вимагає наявності правильних міток класу, тоді як навчання без вчителя може застосовуватися у випадках, коли мітки класу не доступні або коли дослідникам цікаво виявити структуру даних без апріорного знання про класи.

**3.2 Вибір тренувальних та валідацІйних даних для навчання.**

Вибір тренувальних та валідаційних даних є критичним етапом у процесі навчання моделі машинного навчання. Цей процес допомагає переконатися, що модель буде добре узгоджуватися з новими даними, які вона раніше не бачила. Ось кілька кроків, які зазвичай включаються у вибір тренувальних та валідаційних даних:

1. Розділення даних: Спочатку весь набір даних розділяється на дві окремі групи: тренувальний набір та валідаційний набір. Зазвичай від 70% до 80% даних використовується для тренування моделі, а решта для валідації.

2. Випадкове розділення: Дані зазвичай розподіляються випадковим чином між тренувальним та валідаційним наборами. Це допомагає забезпечити, що обидва набори будуть репрезентативними для всього набору даних.

3. Збалансованість класів: Якщо ваші дані містять категоріальні класи, важливо, щоб обидва набори (тренувальний та валідаційний) мали б схожий розподіл класів. Це допоможе уникнути перекосу в оцінці ефективності моделі.

4. Перехресна перевірка (Cross-Validation): У деяких випадках може бути використана перехресна перевірка, де весь набір даних розділяється на кілька підмножин, і модель тренується та валідується декілька разів на різних комбінаціях тренувальних та валідаційних наборів. Це допомагає отримати більш надійну оцінку ефективності моделі.

5. Врахування часового аспекту: Якщо дані мають часову компоненту, важливо, щоб тренувальний та валідаційний набори були розділені за часом. Це допоможе уникнути перенавчання на майбутні дані.

Вибір правильних тренувальних та валідаційних даних допомагає створити надійні моделі, які можуть ефективно узгоджуватися з новими даними.

**3.3 Поняття: штучний нейрон, штучна нейронна мережа, функції активації штучного нейрона**

**(лінійна, порогова, сигмоїдна, радіально-базисна Гауса).**

1. Штучний нейрон (Артіфіційний нейрон):

Штучний нейрон - це математична модель, яка імітує роботу біологічного нейрона в мозку. Він приймає деякі вхідні значення, обчислює їхню вагову суму, додає зміщення (bias), та застосовує до результату функцію активації для генерації вихідного сигналу. Виходить що, штучний нейрон працює за принципом «сигнал-вага-сума».

2. Штучна нейронна мережа (ШНМ):

Штучна нейронна мережа - це система з'єднаних штучних нейронів, яка використовується для моделювання різноманітних завдань, таких як класифікація, прогнозування, розпізнавання образів та інше. Вона складається з різних шарів нейронів, таких як вхідний шар, приховані шари та вихідний шар, і використовується для вирішення складних завдань завдяки її здатності до самонавчання.

3. Функції активації штучного нейрона:

Функція активації - це необхідна складова штучного нейрона, яка визначає вихідний сигнал відповідно до вагової суми вхідних значень. Ось деякі типові функції активації:

- Лінійна функція активації: Просто передає вихідне значення, що дозволяє моделі навчатися лінійним залежностям.

- Порогова функція активації: Видає 1, якщо вхідний сигнал більше порогового значення, і 0 в іншому випадку.

- Сигмоїдна функція активації: Перетворює вхід в діапазоні між 0 та 1, допомагаючи моделі вирішувати проблеми класифікації.

- Радіально-базисна функція активації (RBF): Використовується для апроксимації нелінійних функцій, видаючи велике значення для точок, які знаходяться близько до центра, і менше значення для тих, що віддалені.

Ці функції активації грають ключову роль у прогнозуванні вихідного сигналу штучного нейрона та допомагають нейронним мережам вирішувати різноманітні завдання з машинного навчання.