**Інтелектуальні мехатронні системи**

**Опорний конспект лекцій**

**Тема 1. Основні поняття та визначення інтелектуальних систем керування**.

Штучний інтелект (ШІ) — це галузь комп'ютерних наук, яка займається створенням машин, здатних виконувати завдання, які традиційно потребують людського інтелекту. Це може включати такі дії, як розпізнавання образів, вирішення проблем, навчання та прийняття рішень. ШІ прагне моделювати або навіть перевищувати людські когнітивні здібності, використовуючи складні алгоритми, дані та обчислювальні системи.

Штучний інтелект (ШІ) — це галузь комп'ютерних наук, яка займається створенням машин, здатних виконувати завдання, які традиційно потребують людського інтелекту. Це може включати такі дії, як розпізнавання образів, вирішення проблем, навчання та прийняття рішень. ШІ прагне моделювати або навіть перевищувати людські когнітивні здібності, використовуючи складні алгоритми, дані та обчислювальні системи.

Основні напрямки досліджень систем штучного інтелекту включають:

1. **Машинне навчання (Machine Learning)**:
   * Це підгалузь ШІ, що зосереджується на тому, як комп'ютери можуть вивчати закономірності та прогнози, використовуючи дані, без явного програмування.
   * Алгоритми навчання бувають супервізованими, несупервізованими та з підкріпленням.
2. **Обробка природної мови (Natural Language Processing, NLP)**:
   * Вивчення методів взаємодії між комп'ютерами та людською мовою. Це включає розуміння, аналіз і генерацію тексту або мовлення.
3. **Комп'ютерне бачення (Computer Vision)**:
   * Дослідження технологій, які дозволяють машинам аналізувати та розуміти візуальну інформацію, як-от зображення або відео.
4. **Робототехніка (Robotics)**:
   * Створення автономних або напівавтономних роботів, які використовують ШІ для навігації, прийняття рішень та виконання складних завдань.
5. **Експертні системи**:
   * Комп'ютерні програми, що використовують знання та правила для прийняття рішень в певних доменах, наприклад, медицина чи фінанси.
6. **Глибинне навчання (Deep Learning)**:
   * Вид машинного навчання, що використовує багатошарові нейронні мережі для аналізу великих масивів даних і виконання складних завдань.
7. **Етичні та соціальні аспекти**:
   * Дослідження впливу ШІ на суспільство, вирішення питань конфіденційності, безпеки даних і упередженості в алгоритмах.

Штучний інтелект є міждисциплінарною сферою, яка охоплює математику, статистику, комп'ютерні науки, когнітивні науки, а також етику і філософію. Його дослідження та застосування продовжують трансформувати різні галузі, від медицини та освіти до промисловості й мистецтва.

**Основні підходи машинного навчання:**

1. **Навчання з учителем (Supervised Learning):**
   * Цей підхід базується на використанні "розмічених" даних, де кожен вхід має відоме значення виходу (мітку).
   * Наприклад, для прогнозування цін на нерухомість за даними про її розташування, розмір та кількість кімнат. Популярні алгоритми: лінійна регресія, дерева рішень, нейронні мережі.
2. **Навчання без учителя (Unsupervised Learning):**
   * Використовується для аналізу даних без заздалегідь відомих міток.
   * Алгоритми шукають схожі групи або закономірності в даних.
   * Приклад: кластеризація клієнтів за поведінковими характеристиками. Алгоритми: K-середніх (K-means), DBSCAN.
3. **Навчання з підкріпленням (Reinforcement Learning):**
   * Система вчиться взаємодіяти з середовищем, отримуючи винагороди за правильні дії і штрафи за неправильні.
   * Використовується у робототехніці, грі в шахи, автопілотах.
4. **Глибоке навчання (Deep Learning):**
   * Підкатегорія ML, яка використовує багатошарові нейронні мережі для аналізу великих даних і вирішення складних завдань.
   * Наприклад, розпізнавання облич на фотографіях або синтез тексту.

**Основні застосування:**

* **Рекомендаційні системи:** YouTube, Netflix, Amazon використовують ML для персоналізованих рекомендацій.
* **Розпізнавання образів:** Використовується для ідентифікації облич, обробки медичних зображень тощо.
* **Аналіз тексту:** Обробка природної мови (NLP), наприклад, чат-боти або автоматичне резюме текстів.
* **Фінанси:** Пошук шахрайських транзакцій, автоматична торгівля.
* **Транспорт:** Системи автономного водіння та управління транспортними потоками.

Машинне навчання активно розвивається завдяки збільшенню обчислювальної потужності та доступності великих даних. Ця галузь поєднує математику, програмування, статистику і знання про роботу людського мозку.

Комп'ютерне бачення (Computer Vision) — це галузь штучного інтелекту, яка дозволяє машинам «бачити» і розуміти візуальну інформацію, таку як зображення, відео або 3D-моделі. Основна мета комп'ютерного бачення — аналіз і інтерпретація візуальних даних для автоматизації процесів, які традиційно виконувалися людьми.

**Основні концепції комп'ютерного бачення:**

1. **Розпізнавання образів:**
   * Процес виявлення та класифікації об'єктів на зображеннях. Наприклад, ідентифікація облич на фотографіях або розпізнавання автомобілів на дорозі.
2. **Сегментація зображення:**
   * Розподіл зображення на частини (пікселі), які представляють окремі об'єкти. Використовується в медицині для аналізу рентгенівських знімків.
3. **Виявлення об'єктів (Object Detection):**
   * Ідентифікація місцезнаходження об'єктів у зображенні та їх класифікація. Наприклад, система контролю на виробництві може виявляти дефекти на продукції.
4. **Оптичне розпізнавання символів (OCR):**
   * Перетворення друкованого або рукописного тексту із зображення в машинозчитуваний формат. Використовується для автоматизації введення даних.
5. **Розпізнавання дій (Action Recognition):**
   * Аналіз відео для ідентифікації дій або поведінки, наприклад, виявлення підозрілої активності на відео спостереження.

**Технології та методи:**

* **Глибинне навчання (Deep Learning):**
  + Використання глибоких нейронних мереж для вирішення завдань комп'ютерного бачення, як-от розпізнавання облич чи об'єктів.
  + Популярні архітектури: Convolutional Neural Networks (CNN), YOLO, Faster R-CNN.
* **Алгоритми кластеризації:**
  + Використовуються для аналізу даних, які не мають попередніх міток.
* **Augmented Reality (AR):**
  + Поєднання комп'ютерного бачення з технологіями доповненої реальності.

**Застосування комп'ютерного бачення:**

* **Медицина:** Автоматична діагностика хвороб за аналізом медичних зображень, як-от рентгенівські знімки або МРТ.
* **Автономні транспортні засоби:** Розпізнавання доріг, знаків, інших учасників руху.
* **Системи безпеки:** Виявлення незвичайної поведінки на камерах відеоспостереження.
* **Роздрібна торгівля:** Автоматизовані каси або аналіз поведінки покупців.

Комп'ютерне бачення має величезний потенціал для розвитку нових технологій і покращення існуючих процесів.

Експертні системи — це комп'ютерні програми, які імітують процес прийняття рішень експертів у певній галузі знань. Вони використовують базу знань та правила логіки, щоб аналізувати ситуації, надавати рекомендації, робити прогнози або вирішувати проблеми. Експертні системи стали важливим інструментом у багатьох сферах, таких як медицина, фінанси, виробництво і навіть наука.

**Основні компоненти експертних систем:**

1. **База знань:**
   * Містить факти, правила та інформацію, пов'язані з конкретною галуззю. Ця база формується за допомогою знань експертів у даній сфері.
2. **Машина виведення (Inference Engine):**
   * Логічний механізм, який використовує правила з бази знань для розв'язання задач або формування висновків.
3. **Інтерфейс користувача:**
   * Інструмент, через який користувач взаємодіє з системою, вводить дані і отримує результати або рекомендації.

**Типи експертних систем:**

1. **Експертні системи на основі правил:**
   * Використовують набори правил "якщо-то" для прийняття рішень. Наприклад, в медичних діагностичних системах використовуються правила для визначення хвороби за симптомами.
2. **Експертні системи на основі кейсів:**
   * Базуються на порівнянні нових ситуацій із схожими випадками, які зберігаються в базі даних.
3. **Гібридні експертні системи:**
   * Поєднують кілька підходів, наприклад, правила з машинним навчанням, щоб забезпечити більш точні результати.

**Сфери застосування:**

* **Медицина:** Допомога в діагностиці захворювань, виборі лікування.
* **Фінанси:** Аналіз ризиків, виявлення шахрайства, прогнозування ринкових тенденцій.
* **Інженерія:** Контроль якості, діагностика технічних систем.
* **Юридична практика:** Автоматизована перевірка контрактів, аналіз законодавства.

**Переваги експертних систем:**

* Швидке прийняття рішень.
* Зменшення людських помилок.
* Доступність знань експертів у будь-який момент.

**Виклики:**

* Застарівання бази знань через зміни в галузі.
* Складність у виявленні та формалізації експертних знань.
* Обмежена адаптивність до нових ситуацій.

Експертні системи продовжують еволюціонувати завдяки інтеграції машинного навчання, глибинних нейронних мереж та великих даних.

Інтелектуальні системи — це комп'ютерні системи, що імітують або відтворюють когнітивні здібності людини, такі як навчання, прийняття рішень, аналіз даних і взаємодія з зовнішнім середовищем. Вони будуються на принципах інтеграції штучного інтелекту, машинного навчання, обробки природної мови та інших сучасних технологій.

**Загальні принципи побудови інтелектуальних систем:**

1. **Модульність та ієрархічність:**
   * Інтелектуальні системи часто побудовані за модульним принципом, де кожен модуль виконує конкретне завдання (наприклад, обробка даних, прийняття рішень, взаємодія з користувачем).
   * Система організована ієрархічно, що дозволяє розділяти завдання за рівнями складності.
2. **Адаптивність:**
   * Можливість системи навчатися та адаптуватися до змін у середовищі або нових умов. Використовуються алгоритми машинного навчання для поліпшення її продуктивності з часом.
3. **Інтерактивність:**
   * Взаємодія з користувачем або іншими системами. Це може включати голосові інтерфейси, графічні інтерфейси, сенсори тощо.
4. **Обробка даних:**
   * Системи повинні ефективно збирати, обробляти та аналізувати великі обсяги даних, що дозволяє приймати обґрунтовані рішення.
5. **Розподіленість:**
   * Інтелектуальні системи можуть працювати у розподілених середовищах, де завдання виконуються паралельно на кількох вузлах.
6. **Прозорість та пояснюваність:**
   * Система повинна надавати можливість користувачам зрозуміти, як були прийняті ті чи інші рішення (особливо важливо для систем у медицині, юриспруденції тощо).

**Типові схеми вирішення задач із залученням інтелектуальних систем:**

1. **Формалізація задачі:**
   * Визначення проблеми, постановка чітких цілей і критеріїв успіху.
   * Вибір методу розв'язання: машинне навчання, евристика, методи оптимізації тощо.
2. **Збір даних:**
   * Система збирає дані з навколишнього середовища через сенсори, бази даних або користувачів.
3. **Аналіз даних:**
   * Використання алгоритмів для аналізу структури, закономірностей і трендів в даних.
4. **Прийняття рішень:**
   * Система генерує можливі варіанти рішень і вибирає найкращий варіант на основі заданих критеріїв.
5. **Реалізація:**
   * Виконання обраного рішення, наприклад, активація механізму, генерація звіту або взаємодія з користувачем.
6. **Оцінка результатів та зворотний зв’язок:**
   * Аналіз ефективності рішень. Якщо результат не відповідає очікуванням, система може використовувати зворотний зв'язок для коригування своїх дій.

**Приклад використання:**

Для медичної діагностики інтелектуальна система може:

* Збирати дані про пацієнта (анамнез, аналізи).
* Аналізувати ці дані у порівнянні з базою знань.
* Пропонувати діагнози або рекомендоване лікування.
* Надати лікарю обґрунтовані пояснення своїх висновків.

Інтелектуальні системи знаходять застосування у численних сферах, таких як охорона здоров'я, транспорт, промисловість, торгівля та освіта.

Експертні системи апріорного визначення керування та систем реального часу — це спеціалізовані інтелектуальні системи, які використовуються для автоматизації управлінських процесів і прийняття рішень в реальному часі, особливо у складних і критичних середовищах. Давайте розглянемо ці два типи систем.

**Експертні системи апріорного визначення керування**

Ці системи приймають рішення на основі попередньо визначених знань і правил, що враховують можливі сценарії та дії.

**Особливості:**

1. **Апріорність знань:**
   * Всі знання, необхідні для прийняття рішення, вводяться в систему заздалегідь. Система не вчиться, а лише використовує наявну базу знань.
2. **Побудова бази знань:**
   * Включає факти, правила "якщо-то" та евристики, які визначають дії в конкретних ситуаціях.
3. **Логіка виведення:**
   * Використовується машина виведення, яка застосовує правила до введених даних для отримання результатів.
4. **Автономність:**
   * Система здатна працювати без втручання людини, виконуючи задані дії відповідно до бази знань.

**Приклади:**

* Управління технологічними процесами в промисловості, наприклад, контроль температури в печах.
* Автоматизоване керування механізмами на основі попередньо закладених алгоритмів.

**Експертні системи реального часу**

Ці системи працюють в умовах, де дані надходять постійно, а рішення мають бути прийняті в реальному часі, часто у критично важливих ситуаціях.

**Особливості:**

1. **Збір і обробка даних у реальному часі:**
   * Дані надходять із сенсорів, мережевих пристроїв або інших джерел, і система одразу їх аналізує.
2. **Швидкість прийняття рішень:**
   * Рішення повинні бути прийняті миттєво, без затримок.
3. **Адаптивність:**
   * У таких системах часто використовуються алгоритми машинного навчання для адаптації до змін у середовищі.
4. **Реалізація дій:**
   * Після прийняття рішення система виконує відповідні дії, наприклад, активація аварійного механізму.

**Приклади:**

* Автономні транспортні системи (автопілоти автомобілів чи літаків).
* Управління потоками електроенергії в мережах.
* Системи безпеки, які реагують на підозрілу активність.

**Порівняння**

| **Характеристика** | **Апріорні системи** | **Системи реального часу** |
| --- | --- | --- |
| **База знань** | Статична, визначена наперед | Динамічна, може змінюватися |
| **Швидкість прийняття рішень** | Відносно швидка | Дуже швидка |
| **Адаптивність** | Обмежена | Висока |
| **Сфера застосування** | Плановані процеси | Критичні ситуації |

**Структура побудови інтелектуальних систем керування**

1. **Сенсори (вхідні дані):**
   * Забезпечують збір інформації про об'єкт або середовище. Ці дані можуть стосуватися температури, тиску, швидкості, стану обладнання тощо.
2. **Модуль аналізу даних:**
   * Приймає вхідні дані, аналізує їх за допомогою методів обробки та інтерпретації. Часто використовуються алгоритми машинного навчання для розпізнавання паттернів у даних.
3. **База знань:**
   * Зберігає набір правил, моделей або алгоритмів, які система використовує для прийняття рішень. Це можуть бути евристики, алгоритми оптимізації, або навчені моделі.
4. **Машина виведення:**
   * Механізм, який на основі даних і бази знань генерує рішення. Наприклад, це може бути вибір оптимальної стратегії управління.
5. **Виконавчі механізми (актуатори):**
   * Виконують прийняті рішення. Це фізичні механізми або програмне забезпечення, яке впливає на об'єкт управління.
6. **Інтерфейс користувача:**
   * Забезпечує можливість взаємодії користувача з системою, контролю її роботи та отримання рекомендацій.

**Інтелектуальний зворотній зв’язок**

Інтелектуальний зворотний зв’язок — це процес, при якому інтелектуальна система аналізує результати своїх дій і використовує ці дані для вдосконалення майбутніх рішень. Це відбувається через замкнутий цикл управління:

1. **Моніторинг результатів:**
   * Система постійно оцінює стан об'єкта після виконання своїх дій.
2. **Аналіз відхилень:**
   * Якщо результати відрізняються від очікуваних, система визначає причини цих відхилень.
3. **Адаптація:**
   * На основі аналізу система вносить зміни в свої алгоритми або правила прийняття рішень. Це дозволяє їй пристосовуватись до нових умов або покращувати продуктивність.
4. **Навчання на основі досвіду:**
   * З кожним циклом система накопичує знання, що підвищує її ефективність у майбутньому.

**Переваги інтелектуального зворотного зв’язку**

* **Адаптивність:** Можливість працювати у змінних умовах і враховувати фактори, які не були передбачені на етапі проєктування.
* **Підвищена точність:** Аналіз помилок та їх виправлення забезпечують високу ефективність.
* **Автономність:** Зменшується потреба у втручанні людини.

**Приклад застосування**

В системах керування безпілотними автомобілями інтелектуальний зворотний зв’язок дозволяє автомобілю:

* Аналізувати ситуацію на дорозі (перешкоди, погодні умови).
* Скоригувати траєкторію руху у реальному часі.
* Вчитися на основі даних про минулі ситуації для покращення майбутнього керування.

**Формалізація задач інтелектуального керування**

Формалізація задач інтелектуального керування полягає у створенні структурованого опису проблеми та методів її вирішення. Це ключовий етап у проєктуванні інтелектуальних систем, який включає кілька основних кроків:

1. **Опис системи та середовища:**
   * Визначення об'єкта управління (система чи процес) і умов, у яких він функціонує.
   * Наприклад, це може бути автономний автомобіль, який працює на дорогах зі змінним трафіком і погодою.
2. **Формулювання цілей:**
   * Постановка цілей керування, таких як оптимізація витрат, максимізація ефективності або забезпечення безпеки.
3. **Моделювання:**
   * Розробка математичної моделі об'єкта або процесу, яка враховує його параметри, закономірності та обмеження.
4. **Вибір підходу до розв'язання:**
   * Вибір методів (евристики, алгоритми машинного навчання, логічне виведення), які будуть застосовані для досягнення поставлених цілей.
5. **Аналіз і оптимізація:**
   * Створення алгоритмів для аналізу та прийняття рішень, а також перевірка їх ефективності в тестових або симуляційних середовищах.
6. **Реалізація та інтеграція:**
   * Впровадження системи в реальну роботу з урахуванням усіх обмежень і цілей.

**Тема 2. Мехатронні системи**

**Концепція побудови мехатронних систем**

Мехатронні системи — це інтегровані системи, які об’єднують механічні компоненти, електроніку, комп’ютерне управління та інформаційні технології для створення високотехнологічних пристроїв із широким функціоналом. Концепція побудови мехатронних систем полягає у забезпеченні максимальної ефективності, точності та автоматизації шляхом гармонійного об'єднання складових.

**Структура мехатронних систем**

Мехатронні системи, як правило, мають такі основні компоненти:

1. **Механічна частина:**
   * Включає фізичні елементи, такі як двигуни, редуктори, датчики руху, приводи. Вона відповідає за виконання механічних рухів або дій.
2. **Електронна частина:**
   * Забезпечує збирання та передачу даних, включає схеми живлення, контролери, підсилювачі, мікропроцесори.
3. **Програмне забезпечення (комп’ютерна частина):**
   * Алгоритми управління, які обробляють вхідні дані та формують вихідні сигнали для виконавчих механізмів. Також можуть бути інтегровані алгоритми штучного інтелекту.
4. **Сенсори (датчики):**
   * Реєструють параметри середовища або об’єкта (швидкість, температура, тиск) і передають інформацію в систему.
5. **Інтерфейс користувача:**
   * Забезпечує взаємодію людини із системою, може бути графічним, сенсорним або голосовим.

**Принципи інтеграції мехатронних систем**

1. **Модульність:**
   * Система будується як набір модулів, кожен із яких виконує свою функцію. Це дозволяє легко додавати нові компоненти або змінювати існуючі.
2. **Системний підхід:**
   * Інтеграція здійснюється на основі системного аналізу, який враховує всі аспекти функціонування, включно із взаємодією компонентів.
3. **Інтероперабельність:**
   * Усі компоненти системи повинні бути сумісними та забезпечувати обмін даними без втрати точності.
4. **Адаптивність:**
   * Система повинна бути здатна реагувати на зміни в середовищі та адаптувати алгоритми управління до нових умов.
5. **Енергоефективність:**
   * Компоненти інтегруються таким чином, щоб мінімізувати енергоспоживання без втрати продуктивності.
6. **Прозорість управління:**
   * Інтеграція включає механізми моніторингу та пояснення рішень системи, особливо важливо в критичних додатках.

**Застосування мехатронних систем**

Мехатронні системи широко використовуються в:

* **Автоматизації виробництва:** Роботизовані лінії, 3D-принтери.
* **Медичних технологіях:** Системи для хірургії та діагностики.
* **Транспорті:** Автопілоти, системи управління двигуном.
* **Побутовій техніці:** Смарт-пристрої, як-от пральні машини чи пилососи-роботи.

**Електромеханічні модулі мехатронних систем**

Електромеханічні модулі є ключовими компонентами мехатронних систем, які забезпечують виконання механічних дій шляхом конвертації електричної енергії в механічну. Вони складаються з двигунів, редукторів і виконавчих механізмів, що працюють разом для досягнення високої точності та ефективності.

**Двигуни-редуктори**

1. **Двигуни:**
   * Електричні двигуни є джерелом механічного руху, яке генерується за рахунок електромагнітних процесів.
   * Типи двигунів:
     + **Асинхронні двигуни:** використовуються для загального застосування у промисловості.
     + **Серводвигуни:** забезпечують високу точність керування, особливо в робототехніці.
     + **Крокові двигуни:** ідеальні для позиційного керування у принтерах, CNC-машинах.
     + **Постійного струму (DC):** застосовуються там, де потрібен плавний контроль швидкості.
2. **Редуктори:**
   * Редуктори — це механізми, які змінюють швидкість і крутний момент, переданий від двигуна до виконавчого механізму.
   * Основні типи:
     + **Циліндричні редуктори:** забезпечують прямолінійний передачу.
     + **Планетарні редуктори:** компактні, ідеальні для високих крутних моментів.
     + **Черв'ячні редуктори:** використовуються для точного позиціонування.

**Поступальні модулі**

1. **Суть:**
   * Поступальні модулі призначені для створення лінійного переміщення. Вони використовуються у системах, де необхідна точна зміна позиції об'єкта або компонента.
2. **Компоненти:**
   * **Лінійні приводи:** основний елемент, який створює поступальний рух.
   * **Гвинтові механізми:** конвертують обертальний рух у лінійний (кулькові або трапецієподібні гвинти).
   * **Лінійні напрямні:** забезпечують стабільність і точність переміщення.
3. **Застосування:**
   * Роботизовані маніпулятори.
   * 3D-принтери для переміщення друкуючої головки.
   * Машини для автоматичного збирання.

**Обертальні модулі**

1. **Суть:**
   * Обертальні модулі забезпечують обертальний рух і використовуються в задачах, де потрібно змінити кутове положення компонентів.
2. **Компоненти:**
   * **Двигуни:** генерують обертальний рух.
   * **Редуктори:** адаптують швидкість і крутний момент для виконання задачі.
   * **Обертальні платформи:** дають можливість змінювати положення в просторі.
3. **Застосування:**
   * Роботизовані маніпулятори для обертання захвату.
   * Камери з функцією панорамного обертання.
   * Системи позиціонування антен або сонячних панелей.

**Взаємодія поступальних та обертальних модулів**

Інтеграція поступальних і обертальних модулів створює можливість для складних багатовимірних рухів. Наприклад:

* Роботизовані руки: поєднання поступальних та обертальних рухів для маніпулювання предметами в просторі.
* Системи автоматизації виробництва: одночасне переміщення і позиціонування компонентів.

**Інтелектуальні мехатронні модулі руху**

Інтелектуальні мехатронні модулі руху — це складні компоненти мехатронних систем, які інтегрують сенсори, виконавчі механізми, алгоритми управління та системи адаптивного регулювання для забезпечення точного і ефективного керування рухом. Вони дозволяють автоматизувати складні процеси і підвищити продуктивність у різних сферах.

**Основні характеристики інтелектуальних мехатронних модулів**

1. **Інтегрованість:**
   * Об'єднання механічних, електронних, програмних і сенсорних компонентів в одному модулі для забезпечення автономного функціонування.
2. **Адаптивність:**
   * Модулі здатні адаптувати свої параметри роботи залежно від умов середовища або змін у системі.
3. **Висока точність:**
   * Забезпечується за допомогою сенсорів і алгоритмів, які дозволяють компенсувати похибки або нестабільності.
4. **Можливість самодіагностики:**
   * Інтелектуальні модулі можуть аналізувати власний стан і повідомляти про необхідність технічного обслуговування або потенційні проблеми.
5. **Зворотній зв’язок:**
   * Використовується для постійного моніторингу стану системи і корекції дій на основі отриманих даних.

**Структура інтелектуальних мехатронних модулів руху**

1. **Сенсори:**
   * Датчики для вимірювання параметрів руху (швидкість, прискорення, позиція) і умов середовища (температура, тиск, вологість).
2. **Електромеханічні приводи:**
   * Включають двигуни (серводвигуни, крокові двигуни) та редуктори для забезпечення руху.
3. **Контролери:**
   * Комп'ютеризовані системи управління, які обробляють дані з сенсорів і формують сигнали для виконавчих механізмів.
4. **Програмне забезпечення:**
   * Алгоритми керування, включаючи методи штучного інтелекту, забезпечують оптимізацію руху і адаптацію до змінних умов.
5. **Системи передачі даних:**
   * Забезпечують зв’язок між модулями та центральною системою управління.

**Принципи роботи інтелектуальних мехатронних модулів**

1. **Моніторинг параметрів руху:**
   * Сенсори збирають дані про рух і умови середовища.
2. **Аналіз даних:**
   * Контролер обробляє дані, порівнює їх із заданими параметрами і визначає необхідні коригування.
3. **Прийняття рішень:**
   * Модуль вибирає оптимальні дії для забезпечення точного руху.
4. **Корекція:**
   * Зворотній зв’язок дозволяє модулю постійно коригувати свої дії на основі отриманих даних.

**Переваги використання**

1. **Ефективність:**
   * Автоматизовані модулі підвищують продуктивність систем.
2. **Точність:**
   * Забезпечують високий рівень точності у виконанні задач.
3. **Адаптивність:**
   * Можливість роботи у змінних або нестабільних умовах.
4. **Автономність:**
   * Зменшують потребу у втручанні людини.

**Застосування**

Інтелектуальні мехатронні модулі руху широко використовуються у:

* **Робототехніці:** Автономні маніпулятори, роботизовані платформи.
* **Промисловості:** Системи автоматизації виробничих ліній.
* **Медицині:** Роботизовані хірургічні системи та системи діагностики.
* **Транспорті:** Автопілоти, системи стабілізації.
* **Побуті:** Пилососи-роботи, автоматизовані системи догляду за будинком.

**Мобільні роботи для ремонту підземних комунікацій**

Мобільні роботи для ремонту підземних комунікацій — це спеціалізовані роботизовані системи, які використовуються для обслуговування, діагностики та ремонту підземних інфраструктур, таких як трубопроводи, кабелі та каналізаційні системи. Вони дозволяють виконувати роботи в умовах, які є небезпечними або недоступними для людей.

**Основні функції:**

1. **Діагностика:**
   * Роботи оснащені камерами, сенсорами та іншими пристроями для виявлення пошкоджень, витоків або засмічень.
2. **Ремонт:**
   * Виконують зварювання, герметизацію, очищення або заміну пошкоджених елементів.
3. **Моніторинг:**
   * Постійно збирають дані про стан комунікацій для прогнозування можливих проблем.

**Переваги:**

* Зменшення ризиків для людей.
* Економія часу та ресурсів.
* Можливість роботи в складних умовах (висока температура, токсичні середовища).

**Приклади:**

* Роботи для очищення та ремонту трубопроводів.
* Інспекційні роботи з камерами для виявлення тріщин у трубах.

**Гексаподи**

Гексаподи — це роботизовані системи, які мають шість ніжок і здатні імітувати рухи комах. Вони використовуються в різних галузях завдяки своїй стабільності, маневреності та здатності працювати на нерівних поверхнях.

**Особливості:**

1. **Шість ніжок:**
   * Забезпечують високу стійкість навіть на складних поверхнях.
2. **Біонічний дизайн:**
   * Імітують рухи комах, що дозволяє їм долати перешкоди.
3. **Гнучкість:**
   * Можуть змінювати траєкторію руху залежно від умов середовища.

**Застосування:**

* **Наукові дослідження:** Використовуються для вивчення важкодоступних місць, таких як вулкани чи глибокі печери.
* **Рятувальні операції:** Пошук людей у завалах після катастроф.
* **Промисловість:** Інспекція обладнання в небезпечних умовах.

**Лазерний робото-технічний комплекс**

Лазерний робото-технічний комплекс — це високотехнологічна система, яка використовує лазерне випромінювання для виконання різноманітних завдань, таких як зварювання, різання, маркування, термообробка та інші операції. Ці комплекси інтегрують лазерні технології з робототехнікою, що дозволяє автоматизувати процеси і досягати високої точності.

**Особливості:**

* Використання лазерного променя для локального впливу на матеріал.
* Програмне керування для точного виконання завдань.
* Можливість роботи з різними матеріалами, включаючи метали, пластики та композити.

**Застосування:**

* Машинобудування: виготовлення деталей із високою точністю.
* Електроніка: створення мікросхем та компонентів.
* Медицина: лазерна хірургія та виготовлення медичних інструментів.

**Робототехнологічні комплекси механообробки**

Робототехнологічні комплекси механообробки — це системи, які об'єднують роботів, механічні пристрої та програмне забезпечення для автоматизації процесів обробки матеріалів. Вони використовуються для виконання таких операцій, як свердління, фрезерування, токарна обробка та шліфування.

**Особливості:**

* Інтеграція роботів із механообробними верстатами.
* Використання датчиків для контролю якості обробки.
* Програмне забезпечення для оптимізації процесів.

**Застосування:**

* Автомобільна промисловість: виготовлення деталей двигунів та кузовів.
* Аерокосмічна галузь: виробництво компонентів для літаків та ракет.
* Енергетика: створення турбін та генераторів.

**Транспортні мехатронні засоби**

Транспортні мехатронні засоби — це сучасні транспортні системи, які інтегрують механічні, електронні та інформаційні технології для забезпечення ефективного та безпечного руху. Вони включають автомобілі, залізничні системи, літаки та кораблі.

**Особливості:**

* Використання сенсорів для моніторингу стану транспортного засобу.
* Автоматизовані системи керування, такі як автопілоти.
* Енергоефективність завдяки оптимізації роботи двигунів.

**Застосування:**

* Безпілотні автомобілі: автономне керування без участі водія.
* Залізничні системи: автоматизоване управління рухом поїздів.
* Морський транспорт: навігація та контроль суден.

**Мікроманіпуляційні системи (ММС)**

Мікроманіпуляційні системи — це високоточні механізми, які дозволяють виконувати маніпуляції на мікроскопічному рівні. Вони використовуються для роботи з клітинами, мікроорганізмами, біологічними зразками, а також у нанотехнологіях та електроніці.

**Структура ММС**

1. **Механічна частина:**
   * Включає мікроманіпулятори, які забезпечують точне переміщення інструментів або зразків.
   * Механічні елементи можуть бути моторизованими або ручними.
2. **Сенсори:**
   * Датчики для вимірювання параметрів, таких як позиція, сила або температура.
3. **Приводи:**
   * Використовуються для переміщення маніпуляторів. Найчастіше застосовуються п’єзоелектричні приводи.
4. **Контролери:**
   * Забезпечують управління системою, обробку даних і взаємодію з користувачем.
5. **Інтерфейс користувача:**
   * Графічний або фізичний інтерфейс для налаштування параметрів і контролю роботи системи.

**Модулі руху ММС**

Модулі руху — це компоненти, які забезпечують переміщення маніпуляторів у просторі. Вони можуть бути:

* **Лінійними:** для поступального руху.
* **Обертальними:** для зміни кутового положення.
* **Комбінованими:** для багатовимірного переміщення.

Модулі руху часто оснащуються високоточними двигунами та редукторами, що забезпечують плавність і точність руху.

**Датчики та п’єзо приводи ММС**

1. **Датчики:**
   * Використовуються для моніторингу параметрів системи. Наприклад:
     + П’єзоелектричні датчики для вимірювання сили або тиску.
     + Оптичні датчики для визначення позиції.
2. **П’єзо приводи:**
   * Забезпечують надточне переміщення маніпуляторів.
   * Використовують п’єзоелектричний ефект, який дозволяє конвертувати електричну енергію в механічну.
   * Переваги:
     + Висока точність.
     + Швидка реакція.
     + Компактність.

Мікроманіпуляційні системи є важливим інструментом у сучасній науці та технологіях, забезпечуючи можливість роботи на мікроскопічному рівні з високою точністю.

**Лекція 3. Нечіткі системи та теорія нечітких множин**

**Поняття нечітких множин**

Нечіткі множини (fuzzy sets) — це математичне узагальнення класичних множин, де елементи можуть належати до множини не тільки повністю (1) або не належати взагалі (0), але й мати ступінь належності, який варіюється від 0 до 1. Ця концепція була введена Лотфі Заде у 1965 році для моделювання невизначеностей і нечіткості в даних.

**Ключові характеристики нечітких множин:**

1. **Функція належності:**
   * Кожен елемент множини має значення функції належності, яке показує ступінь його приналежності до множини. Це значення лежить у діапазоні [0, 1].
   * Наприклад, якщо говорити про множину "високих людей", людина зі зростом 180 см може належати до цієї множини зі ступенем 0.8, а людина зі зростом 150 см — зі ступенем 0.2.
2. **Гнучкість опису:**
   * Нечіткі множини дозволяють описувати невизначені або багатозначні поняття, які не можна точно визначити за допомогою класичної математики.
3. **Операції над нечіткими множинами:**
   * Включають об'єднання (union), перетин (intersection) і доповнення (complement), які виконуються з урахуванням функцій належності.

**Нечітка логіка**

Нечітка логіка (fuzzy logic) — це метод логічного мислення, який працює із нечіткими множинами і дозволяє виконувати аналіз, прийняття рішень та моделювання на основі нечіткої інформації. Вона є розширенням класичної булевої логіки, де істинність може мати не тільки значення "правда" (1) або "хиба" (0), але й будь-яке значення між ними.

**Основні принципи нечіткої логіки:**

1. **Нечіткі правила:**
   * Використовуються правила типу "Якщо-то", де умови та висновки формулюються із нечіткими множинами.
   * Наприклад: "Якщо температура *висока*, то вентилятор працює *швидко*".
2. **Агрегація інформації:**
   * Нечітка логіка дозволяє об'єднувати інформацію з різних джерел для формування рішення.
3. **Дефазифікація:**
   * Процес перетворення нечітких висновків у чіткі значення. Наприклад, визначення точного рівня швидкості вентилятора на основі нечіткої логіки.

**Застосування нечітких множин і нечіткої логіки**

1. **Автоматизація і керування:**
   * Управління кліматом, адаптивне регулювання температури, швидкості автомобіля тощо.
2. **Штучний інтелект:**
   * Нечіткі системи використовуються для моделювання невизначеності в робототехніці, розпізнаванні образів, аналізі тексту.
3. **Економіка та фінанси:**
   * Прийняття рішень за умов невизначеності, прогнозування ринкових трендів.
4. **Медицина:**
   * Діагностика хвороб, де враховуються багатозначні симптоми.

**Історія розвитку нечіткої логіки**

Нечітка логіка була вперше запропонована Лотфі Заде у 1965 році в його роботі "Fuzzy Sets". Вона виникла як відповідь на потребу моделювати невизначеність і нечіткість у реальному світі, що не могли бути описані класичною булевою логікою. Основи теорії нечітких множин заклали фундамент для створення інтелектуальних систем, здатних працювати з нечіткими даними.

У 1980-х і 1990-х роках нечітка логіка почала активно застосовуватися в промисловості, особливо в Японії. Наприклад, компанія Matsushita створила пральну машину з нечітким керуванням, яка автоматично налаштовувала параметри прання залежно від типу тканини. Згодом нечітка логіка знайшла застосування в автомобільній промисловості, побутовій техніці, медицині та інших галузях.

**Практика застосування нечіткої логіки**

Нечітка логіка використовується для побудови систем, які працюють в умовах невизначеності. Вона дозволяє створювати адаптивні системи керування, які можуть приймати рішення на основі нечітких правил. Наприклад:

* **Автомобільна промисловість:** Системи стабілізації, адаптивний круїз-контроль.
* **Медицина:** Діагностичні системи, які враховують нечіткі симптоми.
* **Економіка:** Прогнозування ринкових трендів і аналіз ризиків.

**Сфери використання нечітких систем**

Нечіткі системи знаходять застосування в багатьох галузях:

1. **Промисловість:** Автоматизація виробничих процесів, контроль якості.
2. **Транспорт:** Автопілоти, системи навігації.
3. **Енергетика:** Оптимізація роботи електростанцій.
4. **Побутова техніка:** Пральні машини, кондиціонери, пилососи.
5. **Штучний інтелект:** Розпізнавання образів, обробка природної мови.

**Доля ринку пристроїв із нечітким керуванням**

Ринок пристроїв із нечітким керуванням є частиною ширшого сегмента інтелектуальних систем. Очікується, що до 2030 року ринок розумних пристроїв, які часто використовують нечітке керування, досягне $150 мільярдів. Зростання цього ринку обумовлене попитом на автоматизацію, енергоефективність і адаптивність у різних галузях.

Процес проєктування нечіткої системи включає кілька послідовних етапів, кожен з яких забезпечує поступове створення системи, що здатна працювати з нечіткими даними та приймати оптимальні рішення. Давайте розглянемо всі етапи цього процесу.

**1. Етап опису змінних та правил функціонування системи**

На цьому етапі визначаються основні параметри системи:

1. **Опис змінних:**
   * Визначаються вхідні, проміжні та вихідні змінні системи.
   * Для кожної змінної створюється її нечітке представлення за допомогою функцій належності (наприклад, трикутні, трапецієподібні, гауссові функції).
     + Приклад: Якщо змінна "температура" представлена, функції належності можуть бути "низька", "середня" і "висока".
2. **Формулювання правил функціонування:**
   * Визначаються нечіткі правила типу *"Якщо-то"* для всіх сценаріїв роботи системи.
   * Правила описують залежності між змінними.
     + Приклад: "Якщо температура *низька* і швидкість вентилятора *висока*, то споживання енергії *низьке*".
3. **Побудова бази знань:**
   * Всі правила записуються у базу знань, яка використовується для подальшої роботи системи.

**2. Offline-оптимізація**

На цьому етапі проводиться попереднє налаштування системи за відсутності необхідності її активного функціонування.

1. **Аналіз вхідних даних:**
   * Всі вхідні параметри системи досліджуються для визначення їх розподілу, впливу на роботу системи та розробки оптимальних стратегій управління.
2. **Оптимізація функцій належності:**
   * Параметри функцій належності (наприклад, положення піків чи ширина) коригуються для забезпечення точнішого опису вхідних та вихідних даних.
3. **Перевірка правил:**
   * На основі даних тестуються й уточнюються правила системи.
   * Алгоритми, такі як генетичні алгоритми чи методи оптимізації, можуть використовуватись для вибору найкращого набору правил.
4. **Симуляційне тестування:**
   * Система перевіряється на моделях або симуляціях для виявлення помилок і слабких місць.

**3. Online-оптимізація**

На етапі online-оптимізації система функціонує в реальному часі, і її параметри адаптуються до змінних умов.

1. **Адаптація до змін середовища:**
   * Система вивчає поведінку вхідних сигналів і коригує свої функції належності або правила для забезпечення найкращої продуктивності.
2. **Алгоритми навчання:**
   * Використовуються алгоритми, такі як нейронні мережі або методи підкріплення, для автоматичного вдосконалення правил.
   * Наприклад, система може покращувати свою реакцію на зміну температури або тиску в умовах реального функціонування.
3. **Моніторинг роботи:**
   * Постійно збираються дані про ефективність роботи системи для виявлення потенційних проблем.

**4. Реалізація системи**

Фінальний етап полягає у впровадженні системи у практичне застосування.

1. **Програмна реалізація:**
   * Нечітка система реалізується на основі обраного програмного забезпечення (наприклад, MATLAB, Fuzzy Logic Toolbox, Python із бібліотеками нечіткої логіки).
2. **Інтеграція з обладнанням:**
   * Система підключається до сенсорів і виконавчих механізмів для роботи з фізичними об'єктами.
3. **Тестування у реальних умовах:**
   * Система проходить реальні випробування, щоб переконатися в її надійності.
4. **Обслуговування та оновлення:**
   * Система отримує регулярне технічне обслуговування, включаючи оновлення правил і оптимізацію функцій.

**Приклад практичного процесу**

Уявімо, що ми проектуємо систему управління кондиціонером:

* **Опис змінних:** Вхідні змінні — температура, вологість. Вихідна змінна — швидкість вентилятора.
* **Offline-оптимізація:** Тестуємо різні комбінації температури і вологості, оптимізуючи реакцію вентилятора.
* **Online-оптимізація:** Кондиціонер самостійно регулює швидкість, враховуючи змінні умови (наприклад, несподіване підвищення температури).
* **Реалізація:** Інтегруємо систему з датчиками та мотором вентилятора.

Це дозволяє створити ефективну й адаптивну систему, яка працює в умовах невизначеності.

**Лекція 4. Проектування та моделювання нечітких регуляторів**

**Математичний апарат нечіткої логіки**

Математичний апарат нечіткої логіки базується на теорії нечітких множин, яка є розширенням класичної теорії множин. Основна ідея полягає в тому, що кожен елемент множини має ступінь належності, визначений у діапазоні [0, 1], на відміну від традиційної булевої логіки, де належність обмежується значеннями "0" або "1".

**Основні компоненти математичного апарату:**

1. **Нечіткі множини:**
   * **Визначення:** Нечітка множина ( A ) у певному універсумі ( X ) визначається за допомогою функції належності ().
   * (): означає, що елемент ( x ) повністю належить множині ( A ).
   * (): означає, що ( x ) взагалі не належить ( A ).
   * (): часткове належання ( x ) до множини ( A ).
2. **Функції належності:**
   * Математичні функції, які моделюють нечіткі характеристики (наприклад, "низький", "високий").
   * Типові приклади:
     + **Трикутна функція:** визначається піком і шириною.
     + **Трапецієподібна функція:** має плато, що описує інтервал.
     + **Гауссова функція:** описує плавний розподіл.
3. **Операції над нечіткими множинами:**
   * Для роботи з нечіткими множинами використовуються такі операції, як об'єднання, перетин і доповнення.

**Операції із нечіткими множинами**

Операції над нечіткими множинами є аналогами класичних операцій, але виконуються із врахуванням ступеня належності.

**1. Об'єднання (Union):**

* Для двох нечітких множин ( A ) і ( B ) функція належності об'єднання визначається як:
* Інтерпретація: ступінь належності елемента до об'єднаної множини дорівнює максимальному значенню з його ступенів належності до кожної з множин.

**2. Перетин (Intersection):**

* Для двох множин ( A ) і ( B ) функція належності перетину визначається як:
* Інтерпретація: ступінь належності елемента до перетину множин дорівнює мінімуму його ступенів належності до кожної з множин.

**3. Доповнення (Complement):**

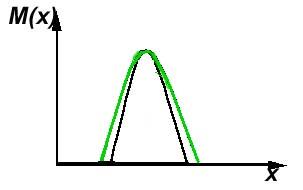
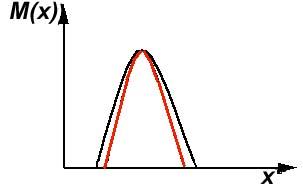
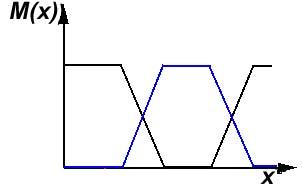
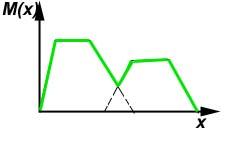
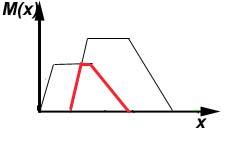
* Доповнення нечіткої множини ( A ) визначається як:
* Інтерпретація: чим більше елемент належить множині ( A ), тим менше він належить до її доповнення.

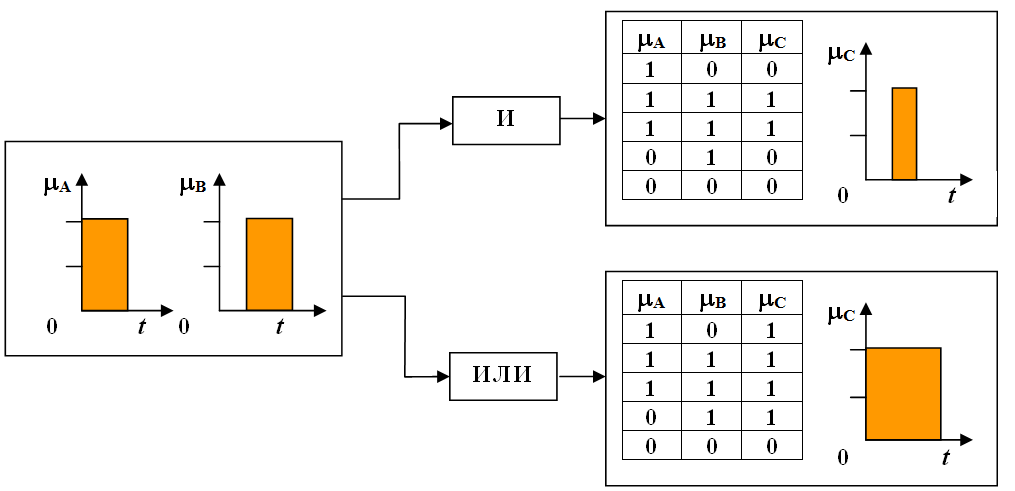
**4. Різниця множин (Difference):**

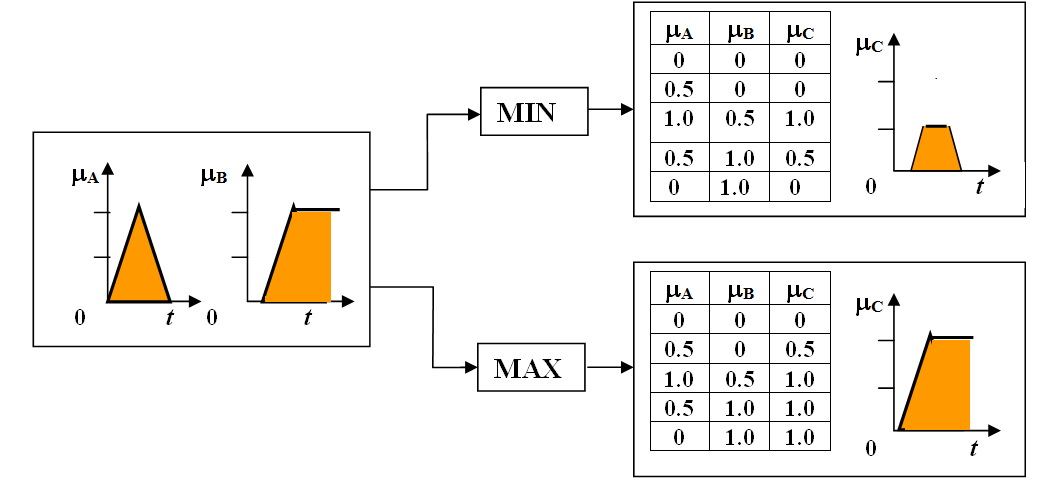
* Різниця нечітких множин ( A ) і ( B ) визначається як:

**5. Декартовий добуток (Cartesian Product):**

* Для множин ( A ) та ( B ) декартовий добуток визначається як множина пар ( (x, y) ), для яких:

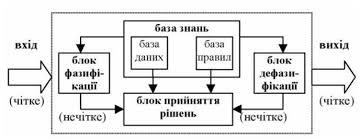






**Моделювання із використанням нечіткої логіки**

1. **Фазифікація (Fuzzification):**
   * Перетворення чітких (детермінованих) змінних на нечіткі множини. Наприклад, температуру 35°C можна віднести до множини "висока температура" зі ступенем належності ( 0.8 ).
2. **Агрегація та інтерференція:**
   * На основі нечітких правил (*"Якщо температура висока, то вентилятор працює швидко"*) виконується аналіз вхідних даних і формуються висновки.
3. **Дефазифікація (Defuzzification):**
   * Перетворення нечітких вихідних даних у чітке значення. Наприклад, швидкість вентилятора переводиться у конкретну кількість обертів.



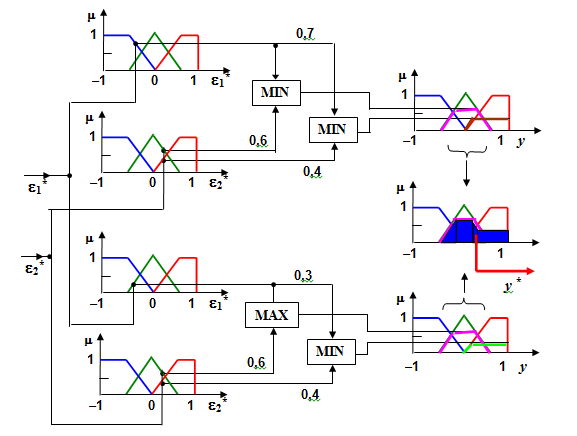
**Приклад практичного використання**

**Задача:** Управління інтенсивністю кондиціонера залежно від температури.

* **Вхідні нечіткі множини:** "низька температура", "середня температура", "висока температура".
* **Вихідні нечіткі множини:** "низька швидкість вентилятора", "середня швидкість", "висока швидкість".
* **Правило:** "Якщо температура *середня*, то вентилятор працює на *середній швидкості*".

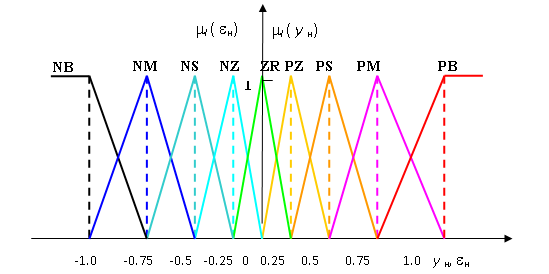
**Поняття лінгвістичних змінних**

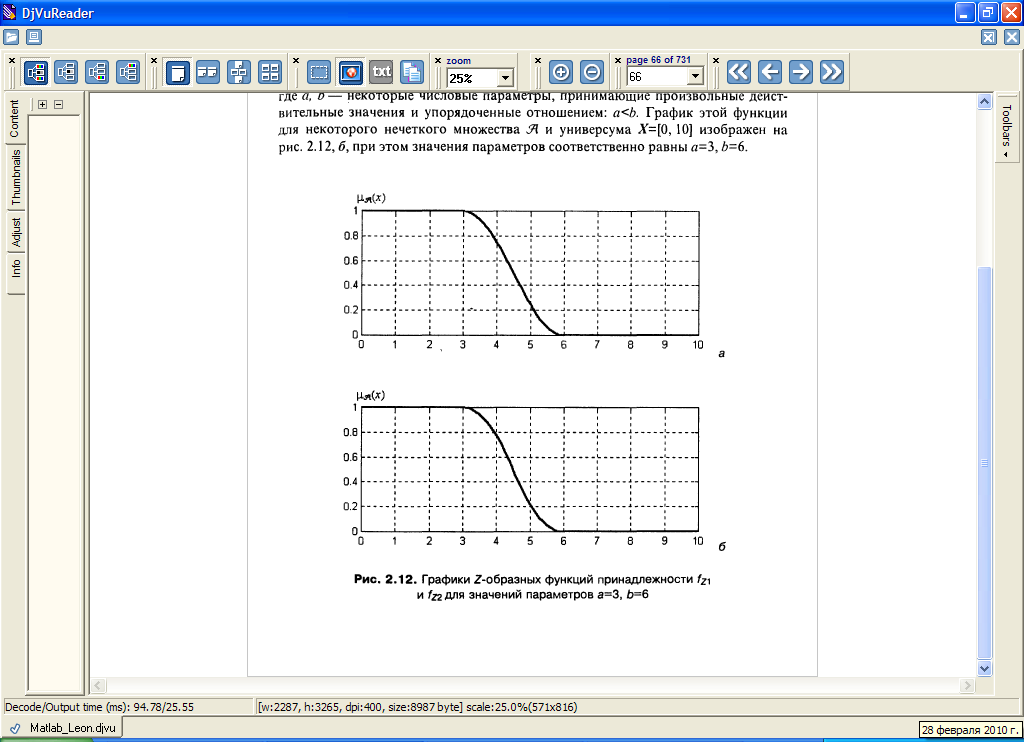
Лінгвістична змінна — це змінна, значення якої не є числом або точним значенням, а представлено за допомогою термінів природної мови (наприклад, "високий", "низький", "середній"). Вона широко використовується в нечіткій логіці для опису невизначених або багатозначних характеристик.



**Компоненти лінгвістичної змінної:**

1. **Ім'я змінної:**
   * Назва змінної, яка описує її властивість (наприклад, "температура").
2. **Універсум дискурсу:**
   * Множина всіх можливих значень змінної (наприклад, діапазон температур від -30°C до 50°C).
3. **Термін множини:**
   * Набір термінів, які використовуються для опису значень змінної (наприклад, "низька температура", "середня температура", "висока температура").
4. **Функції належності:**
   * Математичні функції, які описують ступінь належності кожного значення до терміну множини.





**Приклад використання**

**Сценарій:** Управління швидкістю вентилятора залежно від температури.

* Лінгвістична змінна: Температура.
* Функції належності: "низька температура", "середня температура", "висока температура".
* Правила:
  + "Якщо температура *висока*, то швидкість *висока*".
  + "Якщо температура *середня*, то швидкість *середня*".

**Поняття нечіткого алгоритму**

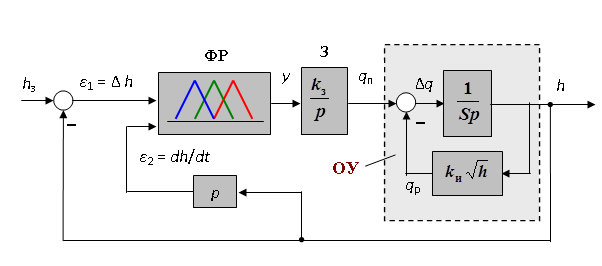
Нечіткий алгоритм — це процедура або набір правил, які використовуються для вирішення задач в умовах невизначеності або багатозначності. У його основі лежить концепція нечіткої логіки, яка дозволяє працювати з лінгвістичними змінними, нечіткими множинами та функціями належності.

**Особливості нечітких алгоритмів:**

1. **Гнучкість:**
   * Нечіткий алгоритм працює з даними, які не є чіткими (наприклад, "середня температура"), дозволяючи знаходити рішення в складних умовах.
2. **Адаптивність:**
   * Алгоритм може адаптуватися до змін середовища або невизначеностей.
3. **Лінгвістичний опис:**
   * Використовуються терміни природної мови для формулювання умов та рішень.

**Структура побудови правил прийняття рішення**

Нечіткі системи прийняття рішень базуються на наборі правил типу "Якщо-то". Ці правила дозволяють виконувати логічні операції на основі нечітких множин.



**Етапи побудови структури правил:**

1. **Формалізація задачі:**
   * Визначення мети, умов середовища та змінних, які впливають на прийняття рішень.
   * Наприклад, задача: регулювання швидкості вентилятора залежно від температури.
2. **Визначення лінгвістичних змінних:**
   * Вхідні змінні (наприклад, температура): "низька", "середня", "висока".
   * Вихідні змінні (наприклад, швидкість вентилятора): "низька", "середня", "висока".
3. **Формулювання нечітких правил:**
   * Правила формулюються у вигляді умов типу "Якщо-то".
   * Наприклад:
     + "Якщо температура *низька*, то швидкість вентилятора *низька*."
     + "Якщо температура *висока*, то швидкість вентилятора *висока*."
4. **Побудова бази знань:**
   * Усі сформульовані правила записуються в базу знань, яка використовується для логічного виведення.
5. **Агрегація даних:**
   * На основі нечітких правил система агрегує вхідні дані для отримання висновків.
   * Приклад: Якщо температура 30°C має належність 0.7 до множини "середня" і 0.3 до множини "висока", система аналізує ці значення і приймає оптимальне рішення.
6. **Дефазифікація (Defuzzification):**
   * Нечіткі вихідні дані перетворюються в чітке рішення.
   * Наприклад, система визначає точну швидкість вентилятора — 70 обертів за хвилину.

**Приклад практичної побудови правил**

**Задача:** Автоматичне регулювання освітлення в кімнаті залежно від рівня природного світла.

1. **Вхідні змінні:** Рівень освітлення:
   * "низький", "середній", "високий".
2. **Вихідні змінні:** Інтенсивність лампи:
   * "слабка", "помірна", "яскрава".
3. **Правила:**
   * "Якщо рівень освітлення *низький*, то інтенсивність лампи *яскрава*."
   * "Якщо рівень освітлення *середній*, то інтенсивність лампи *помірна*."
   * "Якщо рівень освітлення *високий*, то інтенсивність лампи *слабка*."

**Переваги нечітких алгоритмів:**

* Здатність працювати з нечіткою інформацією.
* Інтуїтивно зрозумілі правила на основі природної мови.
* Адаптивність до змін середовища.
* Широкий спектр застосувань, від побутової техніки до складних систем керування.

**Процедура фаззифікації вихідної інформації**

Фаззифікація — це процес перетворення чітких (детермінованих) величин у нечіткі змінні, представлені за допомогою функцій належності до нечітких множин. Вона дозволяє описувати вхідні дані нечіткої системи за допомогою лінгвістичних термінів і нечітких множин.

**Етапи фаззифікації:**

1. **Ідентифікація вхідних даних:**
   * Визначення параметрів, які впливають на вихід системи (наприклад, температура, швидкість).
2. **Опис лінгвістичних змінних:**
   * Перетворення значень параметрів у лінгвістичні змінні (наприклад, "висока температура", "низька швидкість").
3. **Формування функцій належності:**
   * Побудова функцій належності для кожної змінної з використанням трикутних, трапецієподібних або гауссових форм.

**Формування механізму (функції) виводу**

Механізм виводу — це логічний процес, за допомогою якого система аналізує фаззифіковані дані, застосовує нечіткі правила і визначає відповідну вихідну величину.

**Етапи формування механізму виводу:**

1. **Застосування нечітких правил:**
   * Використовуються логічні залежності типу "Якщо-то" для аналізу даних. Наприклад:
     + "Якщо температура *висока*, то швидкість вентилятора *велика*."
2. **Агрегація правил:**
   * Для кожного правила визначається ступінь виконання, що залежить від функцій належності вхідних змінних.
3. **Композиція висновків:**
   * Об'єднання результатів усіх правил для формування вихідної нечіткої множини.
   * Наприклад, використання логічних операцій об'єднання (максимум) або перетину (мінімум).

**Процедура дефаззифікації вихідної множини**

Дефаззифікація — це перетворення нечіткої вихідної множини у чітке числове значення. Цей етап необхідний для того, щоб система могла видавати конкретний результат, який може бути застосований у реальному світі.

**Етапи дефаззифікації:**

1. **Оцінка вихідної множини:**
   * Визначення нечіткої множини, що описує вихідну величину.
2. **Вибір методу дефаззифікації:**
   * Залежно від задачі обирається один із методів перетворення нечіткої множини в чітке значення.
3. **Розрахунок керуючого впливу:**
   * На основі отриманого числового значення визначається керуючий вплив (наприклад, швидкість обертання вентилятора).

**Основні види механізмів виводу**

1. **Мамдані:**
   * Найпоширеніший метод.
   * Використовує нечіткі правила та композицію логіки, а вихідна множина дефаззифікується.
   * Підходить для задач, де потрібна інтуїтивність.
2. **Сугено:**
   * Використовує правила з чітким вихідним значенням (лінійна чи константна функція).
   * Підходить для задач, де важлива точність і швидкість.
3. **Цанко:**
   * Використовує комплексні математичні моделі для формування висновків.

**Основні види дефаззифікації**

1. **Центр тяжіння (Centroid):**
   * Визначення центру мас вихідної функції належності.
   * Найпоширеніший метод через його точність.
2. **Максимум:**
   * Вибір значення, що відповідає максимуму функції належності.
3. **Середнє значення:**
   * Розрахунок середнього з кількох точок максимуму.
4. **Метод середньозваженого центру:**
   * Враховує значення функції належності для кожної точки.

**Лекція 5. Побудова інтелектуальних систем керування за використанням нечіткої логіки**

Інтелектуальні системи керування базуються на принципах, які дозволяють створювати адаптивні, ефективні та гнучкі системи для вирішення складних задач. Основні аспекти:

**Загальні принципи побудови інтелектуальних систем керування:**

1. **Системний підхід**: Включає аналіз і моделювання об'єкта керування, враховуючи всі його компоненти та взаємозв'язки.
2. **Адаптивність**: Системи повинні мати здатність до самонавчання та адаптації до змін у зовнішньому середовищі.
3. **Інтеграція методів штучного інтелекту**: Використання нейронних мереж, генетичних алгоритмів, нечіткої логіки та інших методів для підвищення ефективності.
4. **Оптимізація**: Застосування алгоритмів для пошуку найкращих рішень у багатопараметричних задачах.
5. **Модульність**: Побудова системи з окремих модулів, які можуть бути легко замінені або оновлені.

**Призначення та сфери застосування нечітких моделей:**

Нечіткі моделі використовуються для вирішення задач, де традиційні математичні методи неефективні через невизначеність або складність. Їх основне призначення:

* **Моделювання складних систем**: Наприклад, прогнозування погоди, економічних процесів, поведінки користувачів.
* **Автоматизація керування**: У промисловості, транспорті, робототехніці.
* **Прийняття рішень**: У медицині, фінансах, управлінні проектами.
* **Розробка експертних систем**: Для аналізу даних та підтримки рішень.

Сфери застосування:

* **Промисловість**: Автоматизація виробничих процесів.
* **Медицина**: Діагностика та лікування на основі аналізу даних.
* **Транспорт**: Управління рухом, оптимізація маршрутів.
* **Екологія**: Моніторинг та прогнозування стану навколишнього середовища.

Типова структурна схема систем нечіткого керування включає кілька основних елементів, кожен з яких виконує свою функцію:

**Елементи структурної схеми:**

1. **Блок фаззифікації**: Перетворює вхідні дані у нечіткі множини, використовуючи функції належності.
2. **База правил**: Містить набір правил, які визначають взаємозв'язок між вхідними та вихідними змінними.
3. **Блок нечіткої логіки**: Виконує обчислення на основі правил і нечітких множин.
4. **Блок дефаззифікації**: Перетворює нечіткі вихідні дані у чіткі значення.
5. **Виконавчий механізм**: Реалізує вихідні рішення системи.

**Фаззифікація за принципом Мамдані:**

Метод Мамдані є одним із найпоширеніших підходів у нечіткій логіці. Він передбачає:

* Використання функцій належності для визначення ступеня належності вхідних змінних до нечітких множин.
* Застосування логічних операцій для обчислення вихідних значень на основі правил.
* Дефаззифікація, яка перетворює нечіткі результати у чіткі значення, наприклад, методом центру ваги.

**Фаззифікація за принципом Сугено:**

Метод Сугено відрізняється тим, що вихідні значення визначаються як лінійні або константні функції вхідних змінних. Основні етапи:

* Фаззифікація вхідних даних.
* Застосування правил, де вихідні значення представлені у вигляді функцій.
* Дефаззифікація, яка часто є простішою, оскільки вихідні значення вже є чіткими.

Формування нечітких правил співставлення, концепція керування на випередження та блок-схема нечіткого регулятора є ключовими аспектами у розробці інтелектуальних систем керування. Ось детальний опис:

**Формування та опис нечітких правил співставлення**

Нечіткі правила співставлення базуються на лінгвістичних змінних та нечітких множинах. Основні етапи:

1. **Визначення вхідних і вихідних змінних**: Наприклад, температура, швидкість, тиск тощо.
2. **Формування функцій належності**: Для кожної змінної створюються функції належності, які визначають ступінь належності до певної нечіткої множини.
3. **Створення бази правил**: Правила формуються у вигляді "Якщо...то...", наприклад:
   * Якщо температура висока, то швидкість вентилятора збільшується.
4. **Логічне виведення**: Використовуються методи нечіткої логіки для обчислення вихідних значень на основі правил.

**Концепція керування на випередження**

Керування на випередження передбачає прогнозування майбутнього стану системи та прийняття рішень, які мінімізують відхилення від бажаного стану. Основні аспекти:

* **Прогнозування**: Використання моделей для передбачення поведінки системи.
* **Оптимізація**: Вибір найкращих дій для досягнення цілей.
* **Реалізація**: Використання алгоритмів для впровадження рішень.

**Блок-схема нечіткого регулятора**

Типова блок-схема включає:

1. **Блок фаззифікації**: Перетворює вхідні дані у нечіткі множини.
2. **База правил**: Містить набір правил для керування.
3. **Блок логічного виведення**: Обчислює вихідні значення на основі правил.
4. **Блок дефаззифікації**: Перетворює нечіткі результати у чіткі значення.
5. **Виконавчий механізм**: Реалізує вихідні рішення.

**Лекція 6. Побудова адаптивних систем управління з використанням нечіткої логіки**

Структури інтелектуальних систем керування (ІСК) із нечіткими та ПІД-регуляторами мають важливе значення для забезпечення високої ефективності керування складними об'єктами. Ось детальний опис:

**Структури ІСК із нечіткими та ПІД-регуляторами**

1. **Нечіткий регулятор**:
   * Використовує нечітку логіку для аналізу вхідних даних та прийняття рішень.
   * Складається з блоків фаззифікації, бази правил, блоку логічного виведення та дефаззифікації.
   * Підходить для систем із високим рівнем невизначеності або складністю.
2. **ПІД-регулятор**:
   * Класичний регулятор, який базується на пропорційно-інтегрально-диференційному принципі.
   * Забезпечує стабільність і точність керування для лінійних систем.
   * Складається з трьох компонентів: пропорційного (P), інтегрального (I) та диференціального (D).
3. **Комбіновані схеми**:
   * Поєднують переваги нечітких та ПІД-регуляторів.
   * Нечіткий регулятор використовується для адаптації параметрів ПІД-регулятора в реальному часі.
   * Забезпечують гнучкість і точність керування навіть у складних умовах.

**Переваги застосування комбінованих схем керування**

* **Адаптивність**: Комбіновані схеми можуть адаптуватися до змін у зовнішньому середовищі.
* **Покращена точність**: Нечіткий регулятор допомагає оптимізувати параметри ПІД-регулятора.
* **Стійкість**: Забезпечують стабільну роботу навіть за наявності збурень.
* **Гнучкість**: Підходять для керування як лінійними, так і нелінійними системами.
* **Ефективність**: Зменшують час регулювання та покращують якість перехідних процесів.

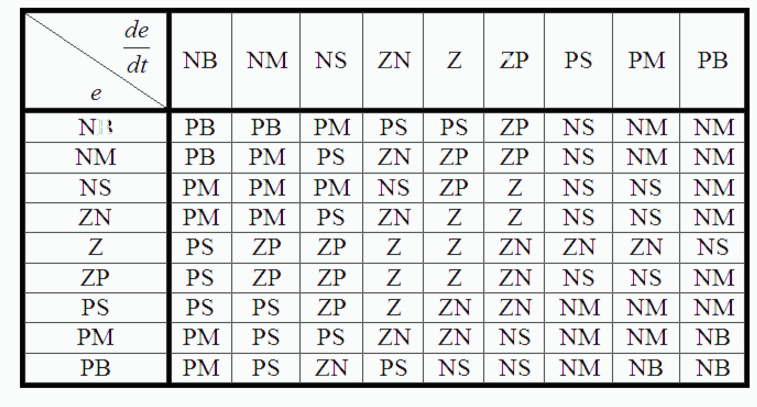
Узагальнена процедура синтезу нечіткого алгоритму керування включає кілька ключових етапів, які забезпечують ефективність і точність роботи системи. Ось детальний опис:

**Узагальнена процедура синтезу нечіткого алгоритму керування:**

1. **Визначення вхідних і вихідних змінних**:
   * Вхідні змінні — це параметри, які впливають на систему (наприклад, температура, тиск).
   * Вихідні змінні — це результати, які система повинна забезпечити (наприклад, швидкість вентилятора).
2. **Фаззифікація вхідних змінних**:
   * Перетворення чітких значень вхідних змінних у нечіткі множини за допомогою функцій належності.
3. **Формування бази правил**:
   * Створення набору правил у вигляді "Якщо...то...". Наприклад:
     + Якщо температура висока, то швидкість вентилятора висока.
   * Правила базуються на експертних знаннях або даних.
4. **Логічне виведення**:
   * Використання методів нечіткої логіки (наприклад, Мамдані або Сугено) для обчислення вихідних значень.
5. **Дефаззифікація**:
   * Перетворення нечітких вихідних значень у чіткі результати.
6. **Аналіз і оптимізація**:
   * Перевірка роботи алгоритму та його налаштування для досягнення оптимальних результатів.

**Формування таблиці правил:**

* Таблиця правил створюється на основі вхідних і вихідних змінних.
* Кожне правило відповідає певному поєднанню значень вхідних змінних.
* Наприклад, для двох вхідних змінних



**Аналіз повноти зв’язків:**

* Перевіряється, чи всі можливі комбінації вхідних змінних враховані в таблиці правил.
* Якщо деякі комбінації відсутні, це може призвести до некоректної роботи системи.

Адаптивна система автоматичного керування (САК) із еталонною моделлю, побудованою за використанням нечіткої логіки, є складною системою, яка забезпечує високу ефективність керування навіть за умов змінних параметрів об'єкта. Ось детальний опис:

**Основні етапи синтезу САК із еталонною моделлю:**

1. **Визначення еталонної моделі**:
   * Еталонна модель представляє бажану поведінку системи.
   * Вона використовується як орієнтир для адаптації параметрів керування.
2. **Формування нечіткої логіки**:
   * Використовуються функції належності для опису вхідних і вихідних змінних.
   * Створюється база правил у вигляді "Якщо...то...", яка визначає взаємозв'язок між змінними.
3. **Адаптація параметрів**:
   * Параметри системи автоматично змінюються для мінімізації похибки між реальним станом системи та еталонною моделлю.
   * Застосовуються алгоритми оптимізації, такі як градієнтний метод.
4. **Контроль похибки**:
   * Похибка між еталонною моделлю та реальним станом системи використовується для корекції параметрів.

**Переваги використання нечіткої логіки:**

* **Гнучкість**: Система може працювати за умов невизначеності.
* **Адаптивність**: Автоматичне налаштування параметрів забезпечує стабільність роботи.
* **Ефективність**: Зменшення часу регулювання та покращення якості керування.

**Типова блок-схема САК із еталонною моделлю:**

1. **Еталонна модель**: Визначає бажану поведінку системи.
2. **Блок фаззифікації**: Перетворює вхідні дані у нечіткі множини.
3. **База правил**: Містить набір правил для керування.
4. **Блок логічного виведення**: Обчислює вихідні значення на основі правил.
5. **Блок дефаззифікації**: Перетворює нечіткі результати у чіткі значення.
6. **Блок адаптації**: Коригує параметри системи на основі похибки.

Практичні приклади застосування інтелектуальних систем керування (ІСК) з нечіткими регуляторами демонструють їх ефективність у вирішенні складних задач, особливо для керування нестійкими об'єктами. Ось кілька ключових аспектів:

**Приклади застосування:**

1. **Робототехніка**:
   * Нечіткі регулятори використовуються для керування роботами, які виконують складні маневри або працюють у змінних умовах. Наприклад, керування балансуванням двоколісного робота.
2. **Автомобільна промисловість**:
   * Системи нечіткої логіки застосовуються для керування двигуном, трансмісією та системами стабілізації автомобіля.
3. **Авіація**:
   * Нечіткі регулятори використовуються для автоматичного керування польотом літаків, особливо в умовах турбулентності.
4. **Енергетика**:
   * Керування нестійкими об'єктами, такими як електричні мережі, де нечіткі регулятори допомагають стабілізувати напругу та частоту.
5. **Медицина**:
   * Нечіткі регулятори застосовуються для керування медичними пристроями, такими як апарати штучного дихання.

**Особливості керування нестійкими об'єктами:**

* Нечіткі регулятори дозволяють враховувати невизначеність і змінність параметрів об'єкта.
* Вони забезпечують стабільність і точність керування навіть за умов складних динамічних змін.

**Лекція 7. Штучні нейронні мережі (ШНМ). Проектування нечітких регуляторів на базі ШНМ**

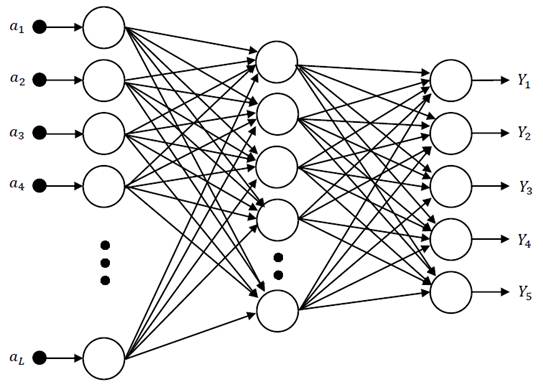
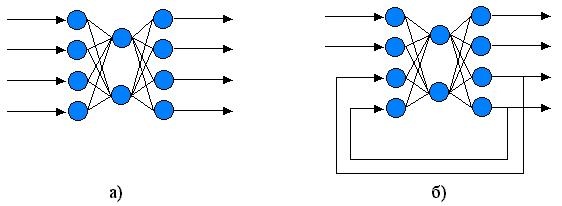
Проектування інтелектуальних систем керування (ІСК) на базі нейронних мереж є важливим напрямком у сучасних технологіях, що дозволяє створювати адаптивні та ефективні системи для вирішення складних задач. Ось детальний опис:

**Загальні поняття**

Штучна нейронна мережа (ШНМ) — це обчислювальна модель, натхненна біологічними нейронними мережами, які складають мозок. Вона складається з вузлів (нейронів), з'єднаних між собою, які обробляють інформацію за допомогою вагових коефіцієнтів. Основні компоненти ШНМ:

* **Вхідний шар**: Отримує дані ззовні.
* **Приховані шари**: Виконують обчислення та аналіз.
* **Вихідний шар**: Видає результати.

ШНМ використовуються для задач класифікації, прогнозування, розпізнавання образів, оптимізації та багатьох інших.



**Визначення та історія виникнення ШНМ**

Штучні нейронні мережі були вперше запропоновані в 1943 році Уорреном Маккалохом і Волтером Піттсом, які створили математичну модель нейрона. Основні етапи розвитку:

1. **1943 рік**: Модель Маккалоха-Піттса, яка описувала роботу нейрона.
2. **1958 рік**: Розробка перцептрона Френком Розенблаттом, що став основою для навчання нейронних мереж.
3. **1980-ті роки**: Відродження інтересу до ШНМ завдяки алгоритму зворотного поширення помилки.
4. **2010-ті роки**: Розвиток глибинного навчання (Deep Learning), що дозволило використовувати багатошарові нейронні мережі для складних задач.

**Проектування ІСК на базі нейронних мереж**

Процес проектування включає:

1. **Визначення задачі**: Наприклад, класифікація зображень або прогнозування даних.
2. **Вибір архітектури мережі**: Одношарові, багатошарові, згорткові (CNN), рекурентні (RNN) тощо.
3. **Підготовка даних**: Збір, очищення та нормалізація даних.
4. **Навчання мережі**: Використання алгоритмів, таких як градієнтний спуск.
5. **Тестування та оптимізація**: Перевірка точності та налаштування параметрів.

**Структурна схема нейронної мережі**

Структурна схема штучної нейронної мережі (ШНМ) включає три основні компоненти:

1. **Вхідний шар**:
   * Отримує початкові дані, які подаються у нейронну мережу.
   * Кожен нейрон вхідного шару відповідає за передачу певної величини до прихованих шарів.
2. **Приховані шари**:
   * Складаються з одного або більше шарів нейронів, які виконують обчислення.
   * Відповідають за обробку інформації, виявлення прихованих закономірностей і перетворення даних.
   * Залежно від типу мережі, приховані шари можуть включати рекурентні, згорткові або інші спеціалізовані механізми.
3. **Вихідний шар**:
   * Надає результати роботи мережі.
   * Кількість нейронів у вихідному шарі залежить від задачі: наприклад, один нейрон для регресії або кілька для класифікації.

**Математична модель нейрона та принципи його функціонування**

Штучний нейрон є базовим елементом ШНМ і моделюється за допомогою математичних рівнянь:

1. **Вхідні сигнали**: — вхідні дані, які отримує нейрон.
2. **Вагові коефіцієнти**: — коефіцієнти, що визначають важливість кожного входу.
3. **Зважена сума**: Нейрон обчислює зважену суму вхідних сигналів: — це зсув (bias), який дозволяє модифікувати вихід.
4. **Функція активації**: — застосовується для визначення вихідного сигналу нейрона. Вона додає нелінійність моделі.

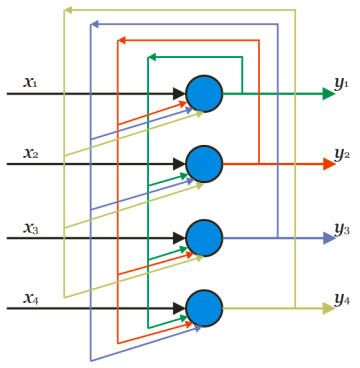
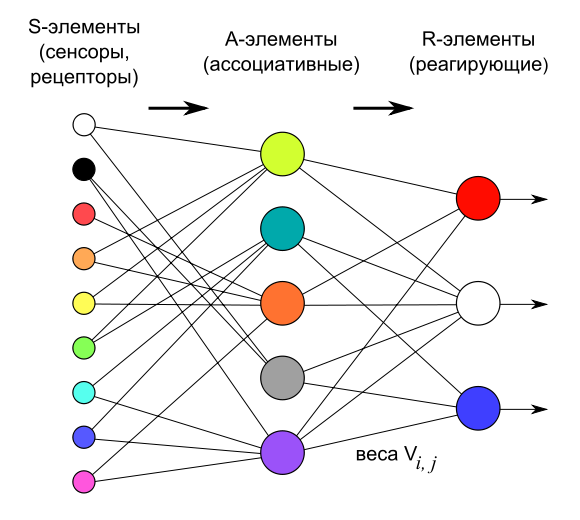
Принципи функціонування:

* Нейрон отримує вхідні сигнали, зважує їх відповідно до коефіцієнтів, додає зсув і передає через функцію активації.
* Сукупність нейронів спільно працюють, щоб навчитися вирішувати задачу.

**Функції активації нейрона**

Функція активації визначає, як нейрон реагує на вхідний сигнал. Основні функції:

1. **Лінійна активація**:
   * Формула:
   * Використовується для регресійних задач.
2. **Сигмоїдна функція**:
   * Формула:
   * Перетворює значення у діапазон від 0 до 1, застосовується для задач класифікації.
3. **ReLU (Rectified Linear Unit)**:
   * Формула:
   * Найбільш популярна функція для прихованих шарів, дозволяє уникати проблеми затухаючого градієнта.
4. **Гіперболічний тангенс (tanh)**:
   * Формула:
   * Повертає значення у діапазоні від -1 до 1, використовується для задач з симетричними даними.
5. **Softmax**:
   * Формула:
   * Використовується у вихідному шарі для класифікації, щоб отримати ймовірності для кожного класу.



**Методи навчання ШНМ та алгоритми адаптації: метод зворотного розповсюдження**

Навчання штучної нейронної мережі (ШНМ) включає оптимізацію вагових коефіцієнтів, щоб мережа могла ефективно виконувати задані завдання, такі як класифікація або прогнозування. Метод зворотного розповсюдження помилки є одним із найпоширеніших підходів у навчанні ШНМ, особливо багатошарових перцептронів.

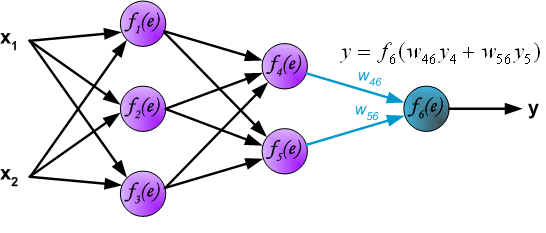
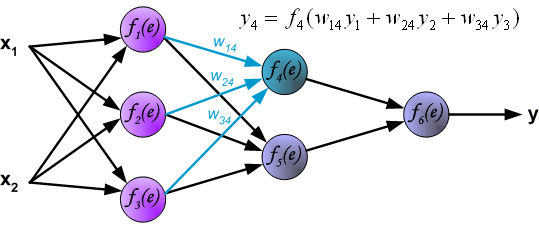
**Основні методи навчання ШНМ:**

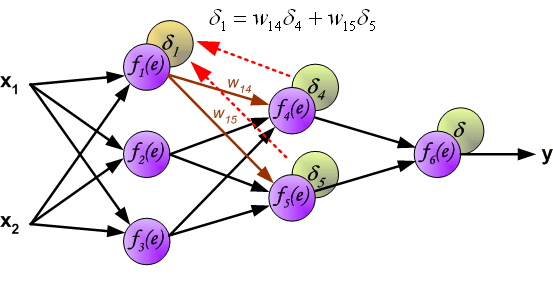
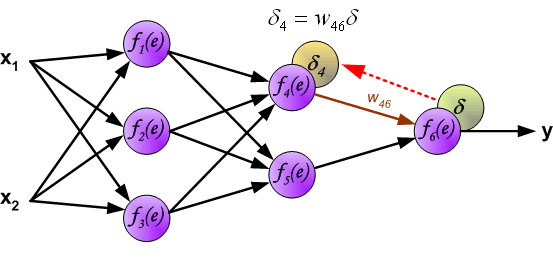
1. **Навчання з учителем**:
   * Мережа навчається на основі набору даних, де кожен вхід має очікуваний вихід (мітка).
   * Використовується функція втрат для порівняння прогнозу мережі з реальним значенням.
   * Застосовується метод зворотного розповсюдження для корекції ваг.
2. **Навчання без учителя**:
   * Мережа самостійно знаходить структури вхідних даних, не маючи міток.
   * Використовується для кластеризації або зменшення розмірності даних.
3. **Навчання з підкріпленням**:
   * Мережа отримує винагороду або покарання за свої дії, вчиться оптимізувати стратегію.
   * Застосовується у завданнях управління або ігор.

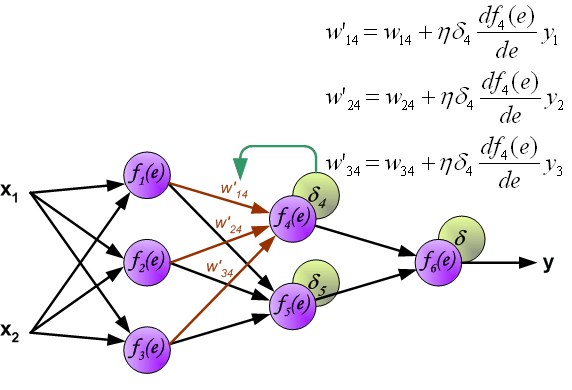
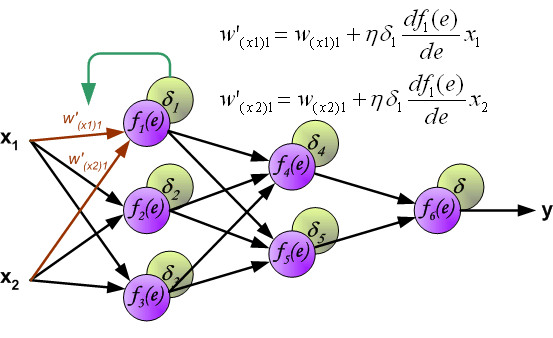
**Метод зворотного розповсюдження помилки:**

Цей метод використовується для навчання багатошарових ШНМ і базується на алгоритмі градієнтного спуску. Основні етапи:

1. **Передній прохід (Forward Pass)**:
   * Вхідні дані передаються через мережу, і вихід обчислюється.
   * Використовується функція активації для обробки значень нейронів.
2. **Розрахунок похибки (Loss Calculation)**:
   * Визначається похибка між прогнозованим виходом і реальним значенням.
   * Застосовується функція втрат, наприклад, середньоквадратична похибка:
3. **Зворотній прохід (Backward Pass)**:
   * Похибка розповсюджується назад через шари мережі.
   * Градієнти обчислюються для кожної ваги, щоб визначити, наскільки кожен параметр впливає на похибку.
4. **Оновлення ваг (Weight Update)**:
   * Використовується правило градієнтного спуску: — швидкість навчання, — градієнт похибки відносно ваги.







**Адаптація алгоритмів навчання:**

Адаптивні методи навчання передбачають динамічне регулювання параметрів, таких як швидкість навчання. Основні підходи:

* **Adam**: Комбінує моменти градієнтів для швидшого збіжності.
* **RMSprop**: Регулює швидкість навчання для кожного параметра окремо.
* **Adagrad**: Зменшує швидкість навчання для параметрів, які часто оновлюються.

**Сфери застосування штучних нейронних систем (ШНС)**

Штучні нейронні системи (ШНС) успішно використовуються у різних галузях завдяки своїй здатності адаптуватися, обробляти великі обсяги даних і знаходити складні взаємозв'язки між змінними. Ось ключові сфери застосування:

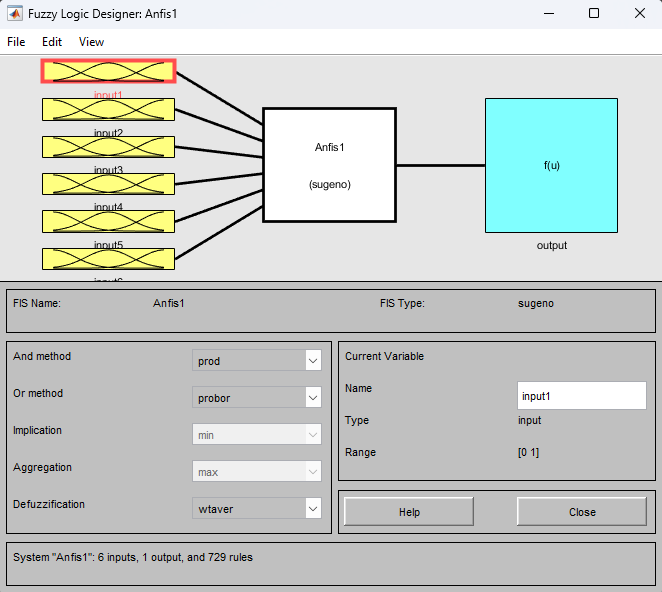
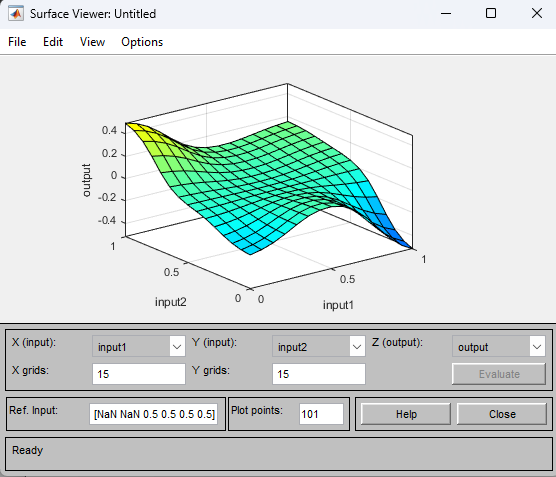
1. **Медицина**:
   * Діагностика захворювань на основі аналізу медичних зображень (рентген, МРТ, КТ).
   * Прогнозування результатів лікування та створення персоналізованих терапевтичних планів.
2. **Фінанси**:
   * Оцінка ризиків у кредитуванні.
   * Виявлення шахрайства у фінансових транзакціях.
   * Прогнозування ринкових трендів та автоматизована торгівля.
3. **Робототехніка та автономні системи**:
   * Навчання автономних транспортних засобів (наприклад, самокеровані автомобілі).
   * Контроль дронів і роботів у реальному часі.
4. **Енергетика**:
   * Оптимізація роботи електричних мереж.
   * Прогнозування споживання енергії та балансування навантаження.
5. **Телекомунікації**:
   * Управління мережевим трафіком.
   * Розпізнавання мовлення для голосових помічників.
6. **Освіта**:
   * Адаптивні системи навчання, які аналізують прогрес студентів.
   * Оцінка знань за допомогою інтелектуальних тестових систем.
7. **Безпека**:
   * Розпізнавання обличчя і голосу.
   * Моніторинг та аналіз відео для забезпечення громадської безпеки.

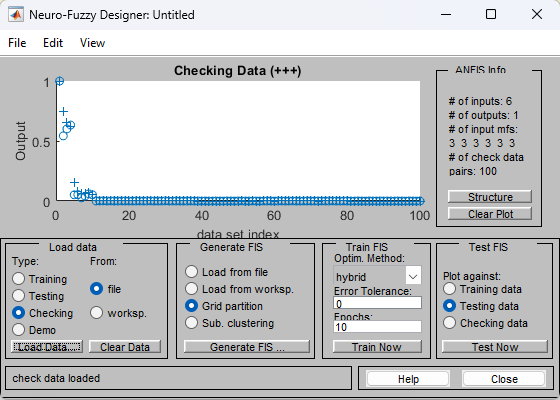
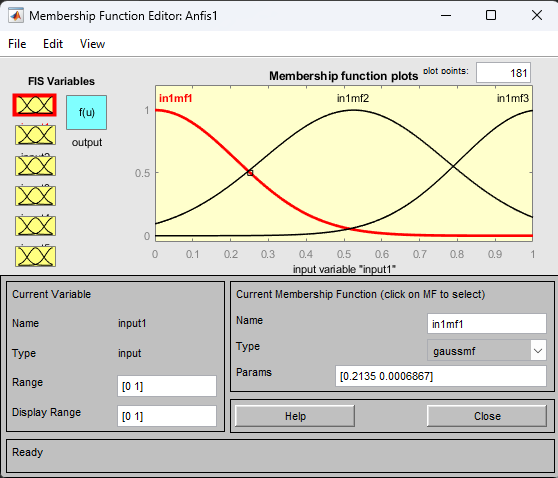
**Побудова нечіткого регулятора за методом FAM (Fuzzy Associative Matrix)**

Метод FAM (Fuzzy Associative Matrix) використовується для створення нечітких регуляторів, які забезпечують високу ефективність керування об'єктами зі складною динамікою. Основні етапи:

1. **Формування нечітких змінних**:
   * Визначення вхідних (наприклад, похибка і похідна похибки) та вихідних (керуюча дія) змінних.
   * Визначення функцій належності для кожної змінної (трикутні, трапецієподібні або гауссові).
2. **Створення матриці правил (Fuzzy Associative Matrix)**:
   * Встановлення взаємозв'язків між вхідними і вихідними змінними у вигляді таблиці правил.
   * Кожен елемент матриці відповідає конкретному правилу типу "Якщо...то...". Наприклад:
     + Якщо похибка велика і зростає, то керуюча дія повинна бути великою.
3. **Логічне виведення**:
   * Використання бази правил для обчислення вихідних значень. Найчастіше застосовуються методи Мамдані чи Сугено.
4. **Дефаззифікація**:
   * Перетворення нечітких значень вихідної змінної у чітке значення. Це може виконуватися методом центру ваги, середнього максимуму тощо.
5. **Тестування та оптимізація**:
   * Аналіз ефективності нечіткого регулятора для конкретної задачі та його налаштування.

Метод FAM забезпечує ефективне керування в системах, де неможливо створити точну математичну модель об'єкта.





**Застосування нейронних мереж та контролерів для мехатронних систем**

Мехатронні системи — це інтегровані комплекси механічних, електронних і інформаційних компонентів, які забезпечують виконання складних завдань, таких як автоматизація, оптимізація та управління. Використання нейронних мереж і контролерів значно підвищує функціональність, надійність і адаптивність таких систем.

**Роль нейронних мереж:**

Нейронні мережі використовуються для:

* **Аналізу даних**: Виявлення складних закономірностей у поведінці систем.
* **Прогнозування**: Передбачення можливих збурень і адаптація системи.
* **Автоматичного навчання**: Удосконалення параметрів системи без потреби в ручному налаштуванні.

**Роль контролерів:**

Контролери забезпечують:

* **Стабільність**: Утримання мехатронної системи в заданому стані.
* **Точність**: Забезпечення точного виконання завдань.
* **Адаптивність**: Здатність реагувати на змінні умови роботи.

**Комбінація нейронних мереж та контролерів:**

Інтеграція нейронних мереж і контролерів дозволяє створювати адаптивні системи, які:

* Навчаються під час роботи, оптимізуючи свої дії.
* Можуть працювати в умовах невизначеності або складної динаміки.
* Підвищують загальну ефективність та швидкодію.

**Тема 4. Розпізнавання зображень**

**Системи розпізнавання зображень**

Системи розпізнавання зображень є однією з ключових областей сучасного машинного навчання та штучного інтелекту. Їх призначення полягає в аналізі, інтерпретації та ідентифікації візуальної інформації, отриманої з камер або інших сенсорів. Основні аспекти:

**Компоненти систем розпізнавання зображень:**

1. **Збір даних**:
   * Отримання зображень за допомогою сенсорів, камер або інших пристроїв.
   * Попередня обробка, включаючи фільтрацію шумів, покращення контрасту та нормалізацію.
2. **Особливості зображень**:
   * Виділення ключових характеристик (краї, кути, текстури) для аналізу.
   * Використання алгоритмів детекції, таких як SIFT, SURF або сучасних підходів з використанням CNN (згорткових нейронних мереж).
3. **Класифікація та інтерпретація**:
   * Використання моделей машинного навчання, наприклад, нейронних мереж або Support Vector Machines (SVM), для ідентифікації об’єктів.
   * Створення ієрархій об'єктів для більш точного аналізу.
4. **Додаткові елементи**:
   * Пост-обробка: об'єднання результатів, усунення хибнопозитивних результатів.
   * Збереження та інтеграція даних у систему.

**Використання сучасних технологій:**

* **Глибинне навчання (Deep Learning)**:
  + Застосування згорткових нейронних мереж (CNN), які автоматично виділяють особливості зображень і виконують класифікацію.
  + Архітектури, такі як ResNet, VGG, YOLO, активно використовуються в задачах розпізнавання.
* **Комп'ютерний зір (Computer Vision)**:
  + Інтеграція аналізу зображень із сенсорними даними для створення 3D-моделей, відстеження об'єктів або виявлення руху.

**Формування зображень в технічних системах**

Формування зображень у технічних системах є процесом отримання, обробки та відображення візуальної інформації. Його основні етапи включають:

1. **Отримання зображення**:
   * Використання камер, сенсорів, томографів, мікроскопів та інших пристроїв.
   * Сенсори можуть бути оптичними, інфрачервоними, ультразвуковими або рентгенівськими.
2. **Перетворення сигналів**:
   * Аналоговий сигнал, отриманий від сенсора, оцифровується для подальшої обробки.
   * Виконується калібрування і корекція отриманих даних.
3. **Обробка зображення**:
   * Фільтрування для зменшення шуму або покращення якості.
   * Виділення контурів, сегментація зображень або покращення глибини кольору.
4. **Реалізація алгоритмів покращення**:
   * Використання сучасних алгоритмів, таких як алгоритми гістонів (Histograms), розтяжки контрасту, адаптивної фільтрації.
   * Додавання кольорових масок або спеціальних фільтрів для досягнення специфічного ефекту.
5. **Збереження і відображення**:
   * Оброблені зображення зберігаються в цифровому форматі для аналізу або передаються на пристрої відображення (монітори, VR-гарнітури).

**Застосування**

Системи розпізнавання зображень та технічні системи формування зображень використовуються в:

* **Медицині** (діагностика за знімками МРТ, КТ).
* **Промисловості** (контроль якості товарів).
* **Безпеці** (розпізнавання облич, автомобільних номерів).
* **Робототехніці** (розпізнавання об’єктів для автономного руху).

**Методи розпізнавання зображень**

Методи розпізнавання зображень базуються на алгоритмах та технологіях, які дозволяють виділяти та класифікувати об'єкти, що містяться в зображеннях. Ці методи можна розділити на традиційні та сучасні підходи.

**Традиційні методи:**

1. **Методи виділення ознак**:
   * Виділення контурів, кутів, текстур або фрагментів за допомогою алгоритмів, таких як:
     + **SIFT (Scale-Invariant Feature Transform)**: Виділяє ключові точки незалежно від масштабу та обертання.
     + **SURF (Speeded-Up Robust Features)**: Швидший аналог SIFT, застосовується для реального часу.
   * Ці ознаки використовуються для подальшої класифікації або зіставлення.
2. **Методи сегментації зображень**:
   * Поділ зображення на області або об'єкти, які мають певні спільні характеристики (кольори, текстури).
   * Використання алгоритмів кластеризації (наприклад, K-means) або порогових методів.
3. **Класифікація за допомогою класичних алгоритмів**:
   * Використання таких методів, як **Support Vector Machines (SVM)** чи **k-Nearest Neighbors (k-NN)**, для розпізнавання зображень на основі виділених особливостей.

**Сучасні методи (глибинне навчання):**

1. **Згорткові нейронні мережі (CNN)**:
   * Використовуються для автоматичного виділення особливостей і класифікації зображень.
   * Популярні архітектури: **ResNet, VGG, AlexNet, YOLO (You Only Look Once)**.
   * Наприклад, CNN аналізує зображення пошарово, виділяючи текстури, краєві ознаки та більш складні патерни.
2. **Трансформери (Vision Transformers)**:
   * Новітні підходи до аналізу зображень, які використовують механізм уваги для обробки всієї інформації зображення одночасно.
   * Вони мають меншу залежність від великих обсягів анотованих даних.
3. **Сегментаційні мережі**:
   * Спеціалізовані нейронні мережі для виділення окремих об'єктів на зображенні (наприклад, **U-Net, Mask R-CNN**).

**Фільтрація та обробка зображень**

Фільтрація та обробка є основними етапами підготовки зображень до аналізу, які дозволяють зменшити шум, покращити якість або виділити ключові області.

**Етапи фільтрації:**

1. **Фільтри низьких частот (розмивання)**:
   * Використовуються для згладжування зображення, видалення дрібних деталей або шумів.
   * Наприклад, **Гауссовий фільтр** застосовується для видалення гауссового шуму.
2. **Фільтри високих частот (виділення країв)**:
   * Використовуються для виділення контурів і границь.
   * Алгоритми: **Sobel, Prewitt, Canny**.
3. **Медіанна фільтрація**:
   * Видаляє імпульсні шуми (білі плями), залишаючи різкість контурів.
4. **Адаптивні фільтри**:
   * Регулюють фільтрацію залежно від локальної інформації, наприклад, для видалення нерівномірного освітлення.

**Методи обробки:**

1. **Нормалізація зображень**:
   * Приведення значень пікселів до стандартного діапазону для зменшення різниці в освітленні або контрасті.
2. **Сегментація**:
   * Поділ зображення на області для виділення об'єктів або фонів. Методи: глобальна порогова сегментація або сегментація з використанням кластеризації.
3. **Зміна простору кольорів**:
   * Перетворення RGB у HSV, LAB або градацію сірого для простішого аналізу.
4. **Морфологічні операції**:
   * Застосовуються для аналізу форм і структур об'єктів на бінаризованих зображеннях.
   * Операції: ерозія, дилатація, відкриття, закриття.
5. **Покращення контрасту**:
   * Алгоритми, такі як гістонівська рівняння, дозволяють покращити видимість деталей на зображенні.

**Використання персептронних мереж для навчання**

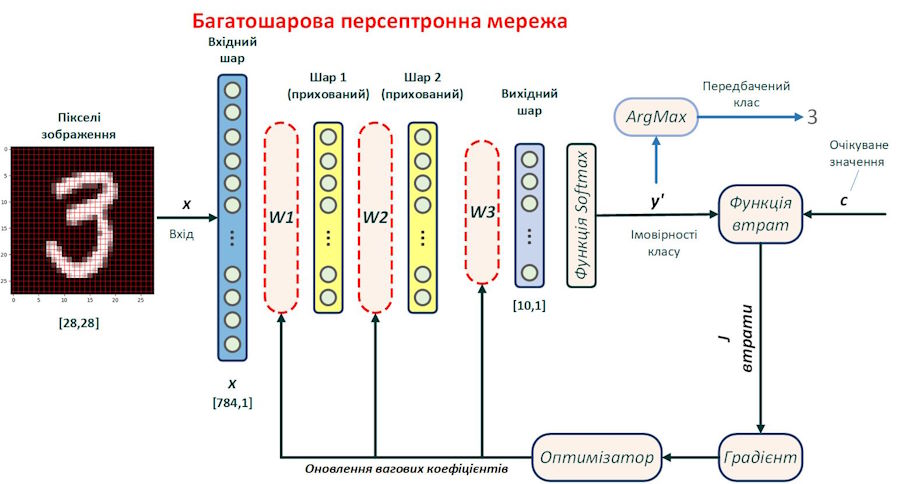
Персептронні мережі є одним із перших типів штучних нейронних мереж, розроблених для вирішення задач класифікації. Вони прості у реалізації, але ефективні для навчання при розв'язанні лінійно роздільних задач. Ось детальний опис їх використання:

**Використання персептронних мереж для навчання комп'ютерного зору**

Персептронні мережі, включно з багатошаровими персептронами (MLP), відіграють важливу роль у навчанні комп'ютерного зору. Комп'ютерний зір—це галузь, яка займається обробкою та аналізом візуальних даних для ідентифікації об'єктів, розпізнавання патернів та розуміння візуальної інформації.

**Застосування персептронних мереж у комп'ютерному зорі:**

1. **Розпізнавання об'єктів**:
   * Персептрони можуть навчатися класифікувати об'єкти на зображеннях. Наприклад, розрізняти, чи є на зображенні автомобіль, кішка чи дерево.
   * Використовуються функції активації (наприклад, сигмоїдна чи softmax) для визначення ймовірності належності зображення до конкретного класу.
2. **Обробка зображень**:
   * Персептрони можуть виділяти особливості зображення, такі як краї, текстури чи контури, для подальшого аналізу.
   * Використовуються для покращення якості зображень або видалення шумів.
3. **Розпізнавання символів та тексту**:
   * Персептрони застосовуються для задач OCR (оптичного розпізнавання символів), де вони допомагають розпізнавати текст з відсканованих документів чи зображень.
4. **Сегментація зображень**:
   * Завдяки персептронним мережам можна розділяти зображення на окремі області для аналізу (наприклад, відокремлення об'єктів від фону).

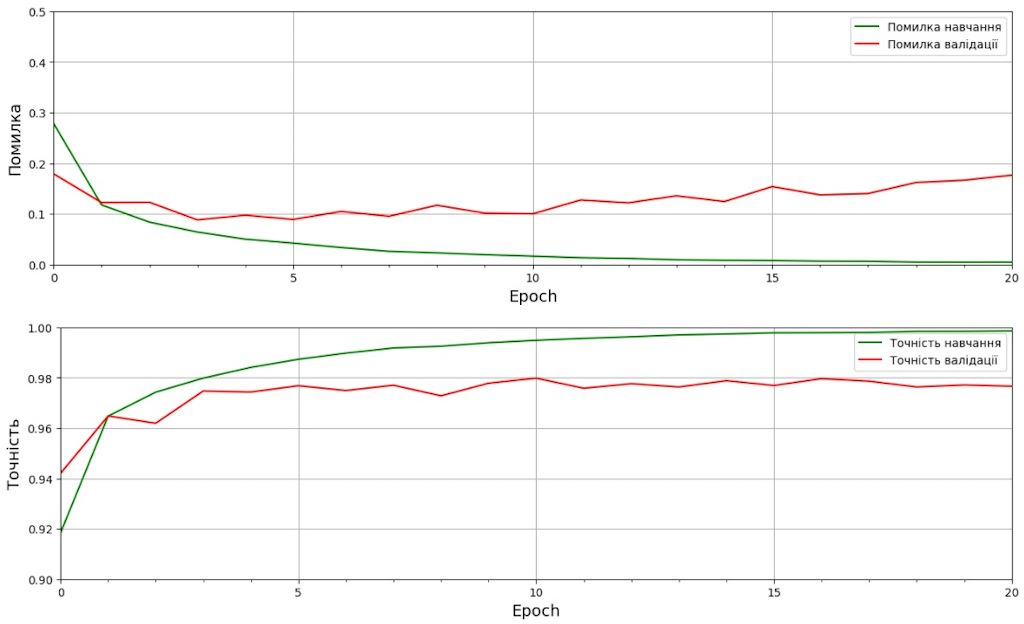


**Переваги використання персептронних мереж у комп'ютерному зорі:**

1. **Обробка багатовимірних даних**:
   * Персептронні мережі легко справляються з великими наборами візуальних даних.
2. **Гнучкість**:
   * Їх можна налаштувати для вирішення різних задач, від розпізнавання до класифікації.
3. **Поєднання з іншими методами**:
   * Персептрони можна інтегрувати з іншими інструментами (наприклад, згортковими нейронними мережами) для більш складних завдань.

**Процес навчання персептронних мереж для комп'ютерного зору:**

1. **Підготовка даних**:
   * Зображення перетворюються у числові вектори, які подаються до персептрона.
   * Виконується нормалізація пікселів (зазвичай значення перетворюються в діапазон від 0 до 1).
2. **Навчання на вибірці**:
   * Використовується набір даних із маркованими зображеннями (наприклад, MNIST для рукописних цифр).
   * Персептрон оптимізується для мінімізації похибки між реальним та прогнозованим класом.
3. **Тестування та валідація**:
   * Після навчання мережа перевіряється на нових даних, які не були використані у навчанні.
   * Визначається її точність і продуктивність.

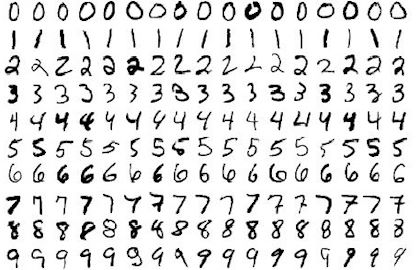


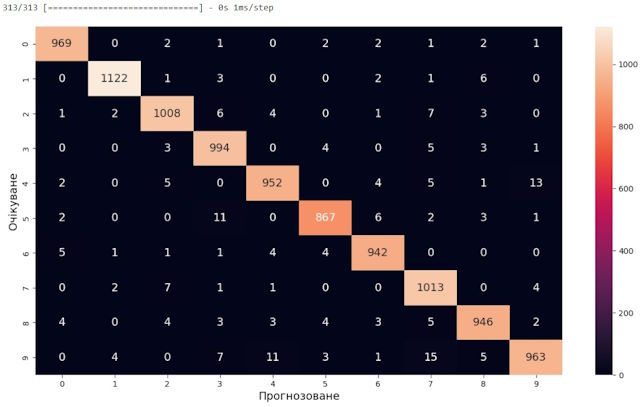
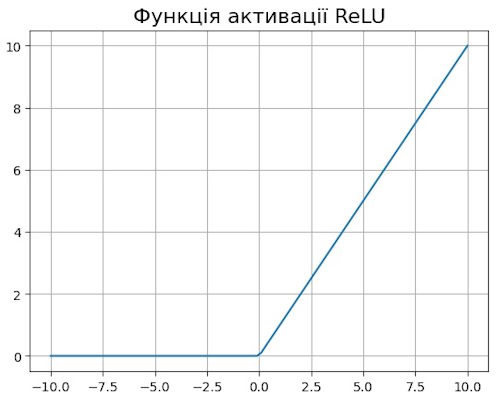
**Обмеження персептронних мереж:**

* **Лінійність**: Одношаровий персептрон може розв'язувати лише лінійно роздільні задачі. Для складніших проблем потрібні багатошарові персептрони.
* **Масштабованість**: Персептрони обмежені при роботі з великими та складними зображеннями. Для складних завдань доцільніше застосовувати згорткові нейронні мережі (CNN).

**Реальні приклади:**

1. **Розпізнавання рукописних цифр**:
   * Використання персептронних мереж для класифікації цифр із набору MNIST.
2. **Ідентифікація простих об'єктів**:
   * Аналіз базових зображень для навчання систем розпізнавання на ранніх етапах.





Матриця невідповідності представляє інформацію у вигляді таблиці або матриці, де одна вісь представляє основні мітки істинності для кожного класу, а інша вісь представляє прогнозовані мітки з мережі.

Створення матриці плутанини в TensorFlow здійснюється шляхом виклику функції tf.math.confusion\_matrix(), яка приймає два обов’язкові аргументи, якими є список базових міток істинності та пов’язаних передбачених міток.