

Интеллектуальные системы

1. Понятие интеллектуальной системы и основные компоненты

Интеллектуальная система (ИС) — это программная или программно-аппаратная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы.

Основные характеристики интеллектуальных систем:

- Адаптивность — способность приспосабливаться к изменяющимся условиям
- Обучаемость — возможность улучшать характеристики с накоплением опыта
- Способность к обобщению — умение применять знания к новым ситуациям
- Объяснимость решений — возможность объяснить процесс принятия решения
- Автономность — способность функционировать без постоянного участия человека

Основные компоненты интеллектуальной системы:

1. **База знаний** — центральный компонент, который хранит формализованную информацию о предметной области:
 - Факты — утверждения об объектах и их свойствах
 - Правила — описание зависимостей между фактами
 - Эвристики — эмпирические правила, основанные на опыте
 - Метазнания — знания о структуре и организации знаний
2. **Механизм логического вывода** — компонент, обрабатывающий знания для получения решений:
 - Поиск в базе знаний необходимой информации
 - Сопоставление фактов с условиями правил
 - Генерация новых фактов на основе существующих
 - Выбор стратегии вывода (прямой, обратный, смешанный)
3. **Подсистема объяснений** — компонент, позволяющий системе объяснять свои рассуждения:
 - Трассировка процесса вывода
 - Объяснение запросов системы
 - Обоснование полученных результатов
 - Предоставление альтернативных решений
4. **Подсистема приобретения знаний** — компонент для пополнения базы знаний:
 - Интерфейс для ручного ввода знаний
 - Автоматическое извлечение знаний из данных
 - Проверка непротиворечивости новых знаний

- Интеграция новых знаний с существующими

5. **Интеллектуальный интерфейс** — компонент для взаимодействия с пользователем:

- Поддержка естественного языка
- Визуализация знаний и результатов
- Адаптация к уровню пользователя
- Различные формы представления информации

6. **Модуль обучения** — компонент, позволяющий системе улучшать свою работу:

- Анализ результатов работы
- Выявление закономерностей
- Корректировка существующих знаний
- Генерация новых правил и моделей

7. **Подсистема моделирования** — компонент для создания и анализа моделей:

- Модели предметной области
- Имитационное моделирование
- Прогнозирование поведения
- "Что если" анализ

2. Структура и принцип работы ИНС Хемминга. Какие задачи можно решать этой нейронной сетью?

Структура ИНС Хемминга

Искусственная нейронная сеть (ИНС) Хемминга — разновидность рекуррентной нейронной сети, предназначенная для распознавания образов и ассоциативной памяти. Она использует расстояние Хемминга для измерения сходства между векторами.

Структура ИНС Хемминга включает:

1. **Входной слой** — слой нейронов, принимающий входной образ
2. **Z-слой** (слой сравнения) — слой, выполняющий вычисление степени сходства входного образа с эталонными
3. **A-слой** (слой соревнования) — выполняет выбор наиболее похожего эталона по принципу "победитель получает всё"

Принцип работы ИНС Хемминга

Работа сети Хемминга происходит в два этапа:

1. Этап инициализации (Z-слой):

- На вход сети подается вектор X , представляющий входной образ
- Каждый нейрон Z-слоя вычисляет скалярное произведение входного вектора X с вектором весов $W_{(i)}$ (представляющим i -й эталонный образ)

- К результату добавляется смещение $n/2$ (где n — размерность входного вектора)
- Выход Z-слоя представляет меру сходства входного образа с эталонными

2. Этап соревнования (А-слой):

- Выходы Z-слоя становятся входами для А-слоя
- А-слой реализует латеральное торможение — нейроны конкурируют между собой
- В процессе итераций нейроны с наибольшими начальными значениями усиливаются, а остальные подавляются
- В конечном итоге активным остается только один нейрон — "победитель"

Задачи, решаемые ИНС Хемминга

ИНС Хемминга эффективно решает следующие задачи:

1. Распознавание образов:

- Классификация зашумленных и искаженных образов
- Идентификация символов и простых изображений
- Распознавание шаблонов в данных

2. Ассоциативная память:

- Восстановление полного образа по частичной информации
- Исправление искаженных образов
- Извлечение наиболее близкого эталона из памяти

3. Классификация бинарных образов:

- Отнесение входного образа к одному из заранее определенных классов
- Распознавание принадлежности объекта к известным категориям

4. Фильтрация шума:

- Восстановление исходного сигнала из зашумленного
- Коррекция ошибок в двоичной информации

Преимущества и ограничения ИНС Хемминга

Преимущества:

- Быстрая сходимость (малое число итераций)
- Гарантированное нахождение устойчивого состояния
- Простота реализации и настройки
- Устойчивость к шуму и искажениям

Ограничения:

- Работает только с дискретными (обычно бинарными) входными данными
- Ограниченная емкость памяти
- Не может распознавать образы, существенно отличающиеся от эталонных

- Чувствительность к корреляциям между эталонными образами

3. Продукционная модель представления знаний и принцип работы систем, основанных на правилах

Продукционная модель представления знаний

Продукционная модель — один из наиболее распространенных способов представления знаний в интеллектуальных системах. Она основана на представлении знаний в виде набора правил вида "Если условие, То действие" (IF-THEN).

Формальная структура продукции:

(i) Q; P; A → B; N

где:

- i — имя или идентификатор правила
- Q — область применимости правила (метаусловие)
- P — условие применимости правила (предусловие)
- A → B — ядро продукции (если A, то B)
- N — постусловие (комментарии, указания о дальнейших действиях)

Типы условий в продукционных правилах:

1. **Простые условия** — элементарные утверждения о значениях переменных
2. **Составные условия** — комбинации простых условий с использованием логических операторов
3. **Нечеткие условия** — условия с использованием лингвистических переменных
4. **Темпоральные условия** — условия, учитывающие временные отношения

Типы действий в продукционных правилах:

1. **Добавление новых фактов** в рабочую память
2. **Модификация существующих фактов**
3. **Удаление фактов** из рабочей памяти
4. **Выполнение процедур** или вызов функций
5. **Активация других правил**

Принцип работы систем, основанных на правилах

Системы, основанные на правилах (Rule-Based Systems, RBS), используют продукционную модель для представления знаний и механизм логического вывода для решения задач.

Архитектура продукционной системы:

1. **База правил** — хранилище всех правил системы
2. **Рабочая память** — хранилище фактов и промежуточных результатов
3. **Механизм логического вывода** — управляет процессом вывода
4. **Интерфейс пользователя** — обеспечивает взаимодействие с пользователем
5. **Модуль объяснений** — объясняет процесс рассуждений
6. **Модуль приобретения знаний** — предоставляет средства для добавления и модификации правил

Цикл работы механизма логического вывода:

1. **Сопоставление (matching)** — определение правил, условия которых удовлетворяются текущими фактами
2. **Разрешение конфликтов (conflict resolution)** — выбор одного правила из всех подходящих
3. **Выполнение (execution)** — выполнение действий, указанных в выбранном правиле
4. **Повторение** — возврат к шагу 1, пока не будет достигнуто решение или не останется применимых правил

Стратегии логического вывода:

1. **Прямой вывод (forward chaining)** — от фактов к заключениям (подход "от данных к цели")
2. **Обратный вывод (backward chaining)** — от гипотезы к подтверждающим ее фактам (подход "от цели к данным")
3. **Смешанный вывод (mixed chaining)** — комбинация прямого и обратного вывода

Стратегии разрешения конфликтов:

1. **По порядку определения** — выбирается правило, определенное раньше других
2. **По специфичности** — выбирается правило с более конкретными условиями
3. **По новизне данных** — выбирается правило, использующее наиболее новые факты
4. **По приоритету** — выбирается правило с наивысшим приоритетом
5. **По количеству условий** — выбирается правило с наибольшим числом условий

Преимущества и недостатки продукционных систем

Преимущества:

- Модульность и независимость правил
- Естественность представления знаний
- Простота модификации и расширения
- Объяснимость процесса вывода
- Единообразие представления знаний

Недостатки:

- Неэффективность при большом числе правил
- Сложность представления неопределенности
- Трудность представления процедурных знаний

- Возможность противоречий в правилах
- Отсутствие гибкости обучения

4. Искусственная нейронная сеть прямого распространения и метод ее обучения

Искусственная нейронная сеть прямого распространения

Искусственная нейронная сеть (ИНС) прямого распространения — это вид нейронной сети, в которой информация движется только в одном направлении, от входного слоя через скрытые слои к выходному слою, без обратных связей.

Структура ИНС прямого распространения:

1. **Входной слой** — принимает входные данные без их преобразования
2. **Скрытые слои** — выполняют нелинейные преобразования входных данных
3. **Выходной слой** — формирует результат работы сети

Нейрон как базовый элемент сети:

Каждый нейрон выполняет следующие операции:

1. Взвешенное суммирование входных сигналов: $NET = \sum (w_{(i)} \times x_{(i)}) + b$
2. Преобразование суммы с помощью функции активации: $OUT = f(NET)$

Функции активации:

1. Пороговая функция (функция Хевисайда):

$$f(x) = 1, \text{ если } x \geq 0$$
$$f(x) = 0, \text{ если } x < 0$$

2. Сигмоидальная функция (логистическая):

$$f(x) = 1 / (1 + e^{(-x)})$$

3. Гиперболический тангенс:

$$f(x) = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$$

4. ReLU (Rectified Linear Unit):

$$f(x) = \max(0, x)$$

5. Leaky ReLU:

$$f(x) = \max(0.01x, x)$$

Метод обучения: алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки (Backpropagation) — основной метод обучения нейронных сетей прямого распространения, основанный на минимизации функции ошибки с помощью градиентного спуска.

Этапы алгоритма:

1. Прямое распространение (Forward pass):

- На вход сети подается обучающий пример
- Сигнал последовательно проходит через все слои сети
- На выходе сети формируется результат

2. Вычисление ошибки:

- Сравнение выхода сети с желаемым результатом
- Вычисление функции ошибки (обычно используется среднеквадратичная ошибка или кросс-энтропия)

3. Обратное распространение (Backward pass):

- Вычисление градиента функции ошибки по весам сети
- Распространение ошибки от выходного слоя к входному

4. Корректировка весов:

- Изменение весов и смещений в направлении, противоположном градиенту
- Формула обновления весов: $w_{(ij)} = w_{(ij)} - \eta \times \partial E / \partial w_{(ij)}$, где η — скорость обучения

5. Повторение процесса для всех обучающих примеров и эпох обучения

Особенности и модификации алгоритма:

1. Выбор скорости обучения:

- Слишком малая скорость — медленная сходимость
- Слишком большая скорость — возможно расхождение
- Адаптивная скорость обучения — изменение скорости в процессе обучения

2. Момент (инерция):

- Добавление члена инерции для ускорения обучения
- Формула с моментом: $\Delta w_{(t)} = -\eta \times \partial E / \partial w + \alpha \times \Delta w_{(t-1)}$, где α — коэффициент момента

3. Пакетное обучение:

- Стохастический градиентный спуск (SGD) — обновление весов после каждого примера

- Пакетный градиентный спуск (Batch GD) — обновление весов после всего набора данных
- Мини-пакетный градиентный спуск (Mini-batch GD) — обновление весов после группы примеров

4. Проблемы обучения:

- Локальные минимумы — застревание в неоптимальных решениях
- Переобучение — слишком хорошая подгонка под обучающие данные в ущерб обобщению
- Исчезающий и взрывной градиент — проблемы с распространением ошибки в глубоких сетях

Применения ИНС прямого распространения:

- Распознавание образов (изображений, речи, рукописного текста)
- Прогнозирование временных рядов
- Классификация и кластеризация данных
- Аппроксимация функций
- Системы принятия решений

5. Представление знаний с помощью семантических сетей. Виды семантических сетей и их назначение

Семантические сети

Семантическая сеть — модель представления знаний в виде графа, вершины которого соответствуют объектам (понятиям, событиям), а дуги — отношениям между ними.

Основные элементы семантической сети:

1. **Вершины (узлы)** — представляют понятия, объекты, события или свойства
2. **Дуги (ребра)** — представляют отношения между понятиями
3. **Метки дуг** — определяют тип отношения (is-a, has-part, causes и др.)

Преимущества семантических сетей:

- Наглядность представления знаний
- Близость к естественному языку и человеческому мышлению
- Гибкость и расширяемость
- Возможность отражения иерархических структур
- Эффективность при поиске связей между объектами

Недостатки семантических сетей:

- Сложность обработки исключений
- Проблемы с представлением процедурных знаний
- Отсутствие стандартизации
- Сложность при больших объемах данных

- Трудности с представлением неопределенности

Виды семантических сетей и их назначение

1. Классифицирующие сети

Назначение: Отражение иерархии понятий и категоризация объектов.

Основные отношения:

- IS-A (является, принадлежит классу) — отношение наследования
- A-KIND-OF (является разновидностью) — отношение "вид-род"
- INSTANCE-OF (является экземпляром) — отношение экземпляр-класс

Пример: "Собака IS-A млекопитающее", "Такса A-KIND-OF собака", "Рекс INSTANCE-OF такса"

2. Функциональные сети

Назначение: Описание процессов, функций и действий объектов.

Основные отношения:

- AGENT (исполнитель) — кто выполняет действие
- OBJECT (объект) — над чем выполняется действие
- INSTRUMENT (инструмент) — с помощью чего выполняется действие
- TIME (время) — когда выполняется действие
- LOCATION (место) — где выполняется действие

Пример: "Иван(AGENT) читает(ACTION) книгу(OBJECT) в библиотеке(LOCATION)"

3. Сценарные сети (фреймы-сценарии)

Назначение: Представление стереотипных ситуаций, последовательности событий.

Основные отношения:

- NEXT (следующий) — временная последовательность
- CAUSE (причина) — причинно-следственная связь
- CONDITION (условие) — условная связь
- GOAL (цель) — целевая связь

Пример: "Закипание воды(CAUSE) ведет к появлению пара(EFFECT)"

4. Партитивные сети (часть-целое)

Назначение: Описание структуры объектов, их составных частей.

Основные отношения:

- HAS-PART (имеет часть) — структурная связь
- PART-OF (является частью) — обратная структурная связь

- MADE-OF (сделан из) — материальная связь

Пример: "Компьютер(WHOLE) HAS-PART процессор(PART)", "Колесо(PART) PART-OF автомобиль(WHOLE)"

5. Лингвистические сети

Назначение: Представление языковых структур и семантики языка.

Основные отношения:

- SYNONYM (синоним) — смысловая эквивалентность
- ANTONYM (антоним) — противоположность значения
- HOMONYM (омоним) — совпадение формы при различии значений
- HYPONYM (гипоним) — подчиненное понятие

Пример: "Автомобиль(SYNONYM) машина", "Горячий(ANTONYM) холодный"

6. Ситуационные сети (сети Петри)

Назначение: Моделирование динамических систем и процессов.

Основные элементы:

- Позиции — состояния системы
- Переходы — события, изменяющие состояния
- Маркеры — ресурсы системы

Пример: Моделирование производственного процесса, системы обслуживания

Применение семантических сетей

- Представление знаний в экспертных системах
- Моделирование естественного языка
- Информационный поиск и семантический веб
- Системы машинного перевода
- Представление онтологий
- Системы вопросно-ответного поиска

6. Формальная постановка задачи кластеризации и методы ее решения

Формальная постановка задачи кластеризации

Кластеризация — это задача разбиения множества объектов на группы (кластеры) на основе их сходства между собой и отличий от объектов из других групп.

Формально задача кластеризации может быть определена следующим образом:

Дано:

- Множество объектов $X = \{x_1, x_2, \dots, x_n\}$
- Функция расстояния (метрика) $d(x_i, x_j)$ между объектами
- Критерий качества кластеризации Q

Требуется: Найти такое разбиение множества X на непересекающиеся подмножества (кластеры) $C = \{C_1, C_2, \dots, C_k\}$, чтобы:

1. $\cup C_i = X$ (объединение всех кластеров дает исходное множество)
2. $C_i \cap C_j = \emptyset$ для $i \neq j$ (кластеры не пересекаются)
3. Критерий качества Q принимает оптимальное значение

Типичные критерии качества кластеризации:

- Минимизация суммы внутрикластерных расстояний
- Максимизация межкластерных расстояний
- Минимизация отношения внутрикластерного разброса к межкластерному расстоянию

Методы решения задачи кластеризации

1. Иерархические методы

Принцип работы: Построение иерархической структуры вложенных кластеров.

Агломеративные методы (снизу вверх):

1. Каждый объект помещается в отдельный кластер
2. На каждом шаге два ближайших кластера объединяются
3. Процесс продолжается до достижения необходимого числа кластеров или порогового расстояния

Дивизимные методы (сверху вниз):

1. Все объекты помещаются в один кластер
2. На каждом шаге кластер разделяется на два
3. Процесс продолжается до достижения необходимого числа кластеров

Методы определения расстояния между кластерами:

- Метод ближайшего соседа (single linkage)
- Метод дальнего соседа (complete linkage)
- Метод среднего расстояния (average linkage)
- Метод Уорда (Ward's method)

2. Метод k-средних (k-means)

Принцип работы:

1. Случайно выбираются k точек как начальные центры кластеров
2. Каждый объект относится к ближайшему центру кластера
3. Центры кластеров пересчитываются как средние всех объектов кластера

4. Шаги 2-3 повторяются до стабилизации кластеров

Преимущества:

- Простота реализации
- Высокая скорость работы
- Хорошая масштабируемость

Недостатки:

- Необходимость заранее задавать число кластеров
- Чувствительность к выбору начальных центров
- Плохая работа с кластерами сложной формы
- Чувствительность к выбросам

3. EM-алгоритм (Expectation-Maximization)

Принцип работы:

1. Инициализация параметров модели смеси распределений
2. Е-шаг: вычисление вероятностей принадлежности объектов к кластерам
3. М-шаг: переоценка параметров модели на основе вычисленных вероятностей
4. Повторение шагов 2-3 до сходимости

Преимущества:

- Мягкая кластеризация (вероятностная принадлежность к кластерам)
- Работа с данными различной природы
- Теоретическое обоснование

Недостатки:

- Сложность реализации
- Вычислительная трудоемкость
- Чувствительность к начальной инициализации

4. Плотностные методы (DBSCAN, OPTICS)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

Принцип работы:

1. Для каждого объекта определяется его ϵ -окрестность
2. Если окрестность содержит не менее MinPts объектов, создается новый кластер
3. Кластер расширяется, включая все достижимые по плотности объекты
4. Объекты, не включенные ни в один кластер, считаются шумом

Преимущества:

- Не требует предварительного задания числа кластеров
- Обнаруживает кластеры произвольной формы
- Устойчивость к шуму

- Работа с кластерами различной плотности

Недостатки:

- Чувствительность к параметрам ϵ и MinPts
- Проблемы с кластерами различной плотности
- Вычислительная сложность для больших наборов данных

5. Сетевые методы (Self-Organizing Maps)**Самоорганизующиеся карты Кохонена (SOM):****Принцип работы:**

1. Создание двумерной решетки нейронов
2. Инициализация весовых векторов нейронов
3. Для каждого входного объекта:
 - Найти нейрон-победитель (BMU)
 - Обновить веса BMU и его соседей
4. Повторение шага 3 с уменьшением радиуса соседства и скорости обучения

Преимущества:

- Визуализация многомерных данных в 2D
- Сохранение топологической структуры данных
- Хорошая работа с зашумленными данными

Недостатки:

- Сложность выбора параметров
- Необходимость предварительного определения размера карты
- Длительное время обучения

Оценка качества кластеризации**Внутренние метрики (без эталонной кластеризации):**

- **Силуэтный коэффициент** — оценивает компактность и разделенность кластеров
- **Индекс Дэвиса-Болдина** — оценивает отношение внутрикластерного разброса к межкластерному расстоянию
- **Индекс Данна** — отношение минимального межкластерного расстояния к максимальному внутрикластерному диаметру

Внешние метрики (с эталонной кластеризацией):

- **Rand index** — доля пар объектов, корректно отнесенных к одному или разным кластерам
- **Adjusted Rand index** — скорректированный Rand index с учетом случайного совпадения
- **Взаимная информация (Mutual Information)** — мера зависимости между двумя кластеризациями

- **F-мера** — комбинация точности и полноты