

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

Кафедра систем автоматизированного проектирования
и информационных систем

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим занятиям по дисциплине **«Системы искусственного интел-
лекта»** для студентов направления 09.03.02 «Информационные системы и
технологии» очной формы обучения

Воронеж 2021

Составитель: канд. техн. наук Ю.В. Литвиненко
УДК 681.38+681.3

Методические указания к практическим занятиям по дисциплине «Системы искусственного интеллекта» для студентов направления 09.03.02 «Информационные системы и технологии» очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. канд. техн. наук Ю.В. Литвиненко. Воронеж, 2021. – 48 с.

Методические указания содержат краткие теоретические и практические сведения об основных подходах к решению неформализованных задач в рамках проектирования систем искусственного интеллекта.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2003 и содержатся в файле МУ_ПЗ_СИИ_ИСТЦ_2021.pdf.

Рецензент д-р техн. наук, проф. А.А. Рындин

Ответственный за выпуск зав. кафедрой
д-р техн. наук, проф. Я.Е. Львович

*Рекомендовано методическим семинаром кафедры САПРИС
и методической комиссией ФИТКБ Воронежского государственного
технического университета в качестве методических материалов*

© ФГБОУ ВО «Воронежский государственный технический университет», 2021

Практическое занятие №1

Выявление знаний в системах искусственного интеллекта. Продукционные системы.

Целью данного практического занятия является проведение исследования процесса разработки экспертных систем в заданной предметной области и реализация продукционной экспертной системы.

В результате выполнения лабораторной работы должны быть приобретены следующие знания и навыки:

- знание назначения и основных свойств экспертных систем;
- знание структуры экспертных систем;
- знание областей применения экспертных систем;
- знание основных моделей представления знаний;
- навыки формализации базы знаний экспертной системы в виде правил «Если-То».

В исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Основным назначением ЭС является разработка программных средств, которые при решении задач, трудных для человека, получают результаты, не уступающие по качеству и эффективности решения, решениям получаемым человеком-экспертом. Экспертная система - это программное средство, использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. ЭС используются для решения так называемых неформализованных задач, общим для которых является то, что:

- задачи не могут быть заданы в числовой форме;
- цели нельзя выразить в терминах точно определённой целевой функции;
- не существует алгоритмического решения задачи;

- если алгоритмическое решение есть, то его нельзя использовать из-за ограниченности ресурсов (время, память).

Кроме того, неформализованные задачи обладают ошибочностью, неполнотой, неоднозначностью и противоречивостью как исходных данных, так и знаний о решаемой задаче.

Обобщенная структура статических экспертных систем представлена на рисунке.

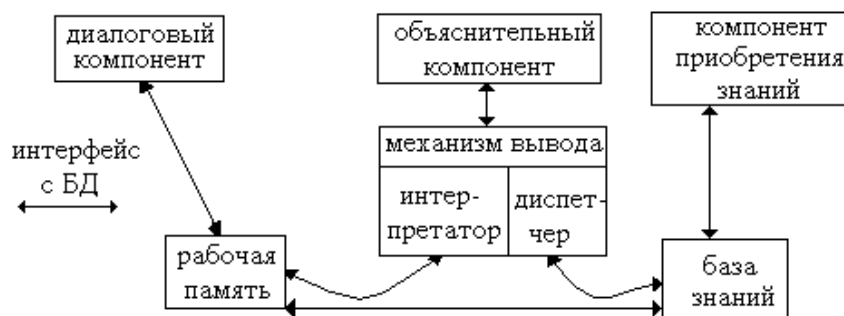


Рис.1. - Структура статической экспертной системы

База данных (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи.

База знаний (БЗ) в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области.

Механизм вывода (решатель), используя исходные данные из рабочей памяти и знания из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи.

Компонент приобретения знаний автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом.

Объяснительный компонент объясняет, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату. Ответ на вопрос «Как?»- это

трассировка всего процесса получения решения (т. е. всех шагов цепи умозаключений). Вопрос «Почему?» - ссылка на умозаключение, непосредственно предшествовавшее полученному решению (отход на один шаг назад).

Диалоговый компонент ориентирован на организацию дружественного общения с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы.

В работе ЭС можно выделить два основных режима: режим приобретения знаний и режим решения задачи (режим консультации или режим использования). В режиме приобретения знаний общение с ЭС осуществляет эксперт (при помощи инженера знаний).

Используя компонент приобретения знаний, эксперт описывает проблемную область в виде совокупности фактов и правил. Другими словами, "наполняет" ЭС знаниями, которые позволяют ей самостоятельно решать задачи из проблемной области.

Отметим, что этому режиму при традиционном подходе к программированию соответствуют этапы: алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом, в отличие от традиционного подхода в случае ЭС разработку программ осуществляет не программист, а эксперт, не владеющий программированием.

В режиме консультаций общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может:

- не быть специалистом в данной предметной области, и в этом случае он обращается к ЭС за результатом, который не умеет получить сам;
- быть специалистом, и в этом случае он обращается к ЭС с целью ускорения получения результата, возлагая на ЭС рутинную работу.

Следует отметить, что в отличие от традиционных программ ЭС при решении задачи не только исполняют предписанную алгоритмом последовательность операций, но и сама предварительно формирует её.

Хорошо построенная ЭС имеет возможность самообучаться на решаемых задачах, пополняя автоматически свою БЗ результатами полученных выводов и решений.

Особенности ЭС, отличающие их от обычных программ, заключаются в том, что они должны обладать:

1. Компетентностью: достигать экспертного уровня решений (т.е. в конкретной предметной области иметь тот же уровень профессионализма, что и эксперты-люди), быть умелой (т.е. применять знания эффективно и быстро, избегая, как и люди, ненужных вычислений), иметь адекватную робастность (т.е. способность лишь постепенно снижать качество работы по мере приближения к границам диапазона компетентности или допустимой надёжности данных).

2. Возможностью к символьным рассуждениям: представлять знания в символьном виде, переформулировать символьные знания. На жаргоне искусственного интеллекта символ — это строка знаков, соответствующая содержанию некоторого понятия. Символы объединяют, чтобы выразить отношения между ними. Когда отношения представлены в ЭС они называются символьными структурами.

3. Глубиной: работать в предметной области, содержащей трудные задачи, использовать сложные правила (т.е. использовать либо сложные конструкции правил, либо большое их количество)

4. Самосознанием: исследовать свои рассуждения (т.е. проверять их правильность), объяснять свои действия.

Существует ещё одно важное отличие ЭС. Если обычные программы разрабатываются так, чтобы каждый раз порождать правильный результат, то ЭС разработаны с тем, чтобы вести себя как эксперты. Они, как правило, дают правильные ответы, но иногда, как и люди, способны ошибаться.

Традиционные программы для решения сложных задач, тоже могут делать ошибки. Но их очень трудно исправить, поскольку алгоритмы, лежащие в их основе, явно в них не сформулированы. Следовательно, ошибки нелегко

найти и исправить. ЭС, подобно людям, имеют потенциальную возможность учиться на своих ошибках.

В настоящее время сложилась определенная технология разработки экспертных систем, которая включает следующие шесть этапов: идентификация, концептуализация, формализация, выполнение, тестирование, опытная эксплуатация.

Этап идентификации проблемной области включает определение назначения и сферы применения ЭС, подбор экспертов и инженеров по знаниям, выделение ресурсов, постановку и параметризацию решаемых задач, неформальное описание задачи.

На этапе построения концептуальной модели создается целостное и системное описание используемых знаний, отражающее сущность функционирования проблемной области. От качества построения концептуальной модели проблемной области во многом зависит насколько часто в дальнейшем по мере развития проекта будет выполняться перепроектирование базы знаний. Хорошая концептуальная модель может только уточняться (детализироваться или упрощаться), но не перестраиваться.

Результат концептуализации проблемной области обычно фиксируется в виде наглядных графических схем на объектном, функциональном и поведенческом уровнях моделирования:

- объектная модель описывает структуру предметной области как совокупности взаимосвязанных объектов, отражает знания о составе объектов, их свойствах;
- функциональная модель отражает действия и преобразования над объектами;
- поведенческая модель рассматривает взаимодействия объектов во временном аспекте, отражает изменение состояний объектов в результате возникновения некоторых событий, влекущих за собой выполнение определенных действий.

Первые две модели описывают статические аспекты функционирования проблемной области, а третья модель - динамику изменения ее состояний.

На этапе формализации базы знаний осуществляется выбор метода представления знаний. В рамках выбранного формализма осуществляется проектирование логической структуры базы знаний. Выходом этапа формализации является описание того, как рассматриваемая задача может быть представлена в выбранном формализме и определение способов манипулирования знаниями.

Цель этапа выполнения - создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний.

В ходе этапа тестирования производится оценка выбранного способа представления знаний в ЭС в целом. Для этого инженер по знаниям подбирает примеры, обеспечивающие проверку всех возможностей разработанной ЭС.

На этапе опытной эксплуатации проверяется пригодность ЭС для конечного пользователя. Пригодность ЭС для пользователя определяется в основном удобством работы с ней и ее полезностью. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). В свою очередь, удобство работы с ЭС подразумевает естественность взаимодействия с ней (общение в привычном, не утомляющем пользователя виде), гибкость ЭС (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях неопытного пользователя).

В ходе разработки ЭС почти всегда осуществляется ее модификация. Выделяют следующие виды модификации системы: переформулирование понятий и требований, переконструирование представления знаний в системе и усовершенствование прототипа.

На начальном этапе проектирования экспертной системы необходимо изучить и описать предметную область, для которой она разрабатывается. Знания обычно собираются и систематизируются путем диалога инженера по знаниям с несколькими специалистами-экспертами в данной проблемной области.

В рамках исследования искусственного интеллекта созданы многочисленные экспертные системы для разных областей знания, таких, например, как медицинская диагностика и обследование пациентов, генные и молекулярные исследования, составление конфигурации вычислительных машин, образование, поиск неисправностей в устройствах и системах и многие другие практические приложения. В рамках данной лабораторной работы предложено разработать экспертную систему, связанную с диагностикой заболеваний или экспертную систему определения породы дерева.

Например, пусть предметной областью будут заболевания гортани: ларингит острый, ларингит хронический катаральный, ларингит хронический гипертрофический, ларингит хронический атрофический. В результате диалога с врачом-отоларингологом установлено, что каждое из перечисленных заболеваний характеризуется набором признаков (симптомов). Для удобства восприятия анализ предметной области можно свести в таблицу.

Столбцы этой таблицы соответствуют названиям заболеваний, а строки - симптомам. На пересечении столбцов и строк стоит знак "+", если симптом соответствует исследуемой болезни. Одни и те же симптомы могут являться признаками разных заболеваний.

Описание предметной области

Симптом/ заболевание	ларингит острый	ларингит хроническ ий катаральн ый	ларингит хроническ ий гипертроф ический	ларингит хроническ ий атрофичес кий
общее недомогание	+	+	+	+
сухость першение, саднение	+	+		+
кашель сначала сухой, затем с мокротой	+			
голос хриплый или беззвучный	+	+		+
иногда боль при глотании	+			
головная боль	+			
повышение температуры тела	+			
быстрая утомляемость голоса		+		
периодический кашель с мокротой		+		
охриплость с афонией			+	
ощущение неловкости			+	
жжение в горле			+	
кашель при обострении			+	
сухой кашель				+
слизистая покрыта густой слизью				+
откашливание с прожилками крови				+

Пусть предметной областью будет определение породы дерева. В результате диалога со специалистом-экспертом в данной области для идентификации породы внутри каждого подмножества сформирован список атрибутов. Количество характеристик определяет степень точности классификации. Различающей не обязательно является какая-нибудь единственная характеристика - все множество атрибутов используется для достижения целей в строящихся правилах. Все перечисленные ниже атрибуты являются необходимыми, так как ни один из них не характерен для всех пород одновременно.

Таким образом, древесина может быть: лиственная; хвойная; мягкая; твердая; очень твердая; серо-коричневая; светло-красная; светлая; темная; смолистая; очень смолистая; крупная текстура; мелкая текстура. Каждая характеристика для конкретной породы либо верна, либо не верна.

Для каждой породы справедливы следующие характеристики. Дуб принадлежит к деревьям лиственной породы, древесина твердая, светло-коричневого цвета с мелкой текстурой. Бук также является лиственным деревом с твердой древесиной крупной текстуры светло-красного цвета. Осина – дерево лиственной породы с мягкой древесиной светлого цвета и мелкой текстуры. Тис - дерево лиственной породы с очень твердой древесиной темной окраски. Ель – дерево хвойной породы с мягкой смолистой древесиной светлой окраски.

Основу любой ЭС составляет база знаний (БЗ) о предметной области, которая накапливается в процессе построения и эксплуатации ЭС. Поэтому следующим этапом проектирования экспертной системы является разработка (выбор) способа представления знаний, полученных от эксперта в данной проблемной области в базе знаний.

В настоящее время существует большое количество моделей представления знаний, многие из которых можно свести к нескольким классам: фреймы, семантические сети, логические модели, продукционные модели.

Один из часто используемых в экспертных системах способов представления знаний – продукционная модель. Основная идея заключается в ассоциировании с соответствующими действиями набора условий в виде правил типа "если-то", называемых также продукциями: ЕСЛИ условие ТО действия. "Если-то"-правила обычно оказываются весьма естественным и выразительным средством представления знаний. Кроме того, они обладают следующими положительными свойствами:

- модульность: каждое правило описывает небольшой, относительно независимый фрагмент знаний;

-возможность инкрементного наращивания: добавление новых правил в базу знаний происходит относительно независимо от других правил;

- удобство модификации (как следствие модульности): старые правила можно изменять и заменять на новые относительно независимо от других правил;

- применение правил способствует прозрачности системы, т.е. способности к объяснению принятых решений и полученных результатов.

Однако, продукционные системы не свободны от недостатков:

- процесс вывода менее эффективен, чем в других системах, поскольку большая часть времени при выводе затрачивается на непроизводительную проверку применимости правил;

- этот процесс трудно поддается управлению;

- сложно представить родовидовую иерархию понятий.

Представление знаний, основанное на фреймах, является альтернативным по отношению к системам продукций: оно дает возможность хранить родовидовую иерархию понятий в базе знаний в явной форме. Фреймом называется структура для описания стереотипной ситуации, состоящая из характеристик этой ситуации и их значений. Характеристики называются слотами, а значения – значениями слотов. Слот может содержать не только конкретное значение, но и имя процедуры, позволяющей вычислить его по заданному алгоритму, а также одну или несколько продукций (эвристик), с помощью которых это значение можно найти. В слот может входить не одно, а несколько значений. Иногда слот включает компонент, называемый фасетом, который задает диапазон или перечень его возможных значений. Фасет указывает также граничные значения заполнителя слота.

Совокупность фреймов, моделирующая какую-нибудь предметную область, представляет собой иерархическую структуру, в которую фреймы соединяются с помощью родовидовых связей. На верхнем уровне

иерархии находится фрейм, содержащий наиболее полную информацию, истинную для всех остальных фреймов. Фреймы обладают способностью наследовать значения характеристик своих родителей, находящихся на более высоком уровне иерархии. Значения характеристик фреймов могут передаваться по умолчанию фреймам, находящимся ниже них в иерархии, но если последние содержат собственные значения данных характеристик, то в качестве истинных принимаются именно они.

Достоинства фреймовых систем представления знаний проявляются в том случае, если родовидовые связи изменяются нечасто и предметная область насчитывает немного исключений. Во фреймовых системах данные о родовидовых связях хранятся явно, т.е. так же, как и значения всех других типов. Значения слотов представляются в системе в единственном экземпляре, поскольку включаются только в один фрейм, описывающий наиболее общее понятие из всех тех, которые содержат слот с данным именем. Такое свойство систем фреймов дает возможность уменьшить объем памяти, необходимый для их размещения в компьютере. Еще одно достоинство фреймов состоит в том, что значение любого слота при необходимости может быть вычислено с помощью соответствующих процедур или найдено эвристическими методами.

Как недостаток фреймовых систем следует отметить их относительно высокую сложность, что проявляется в снижении скорости работы механизма вывода и в увеличении трудоемкости внесения изменений в родовидовую иерархию. Кроме того, во фреймовых системах затруднена обработка исключений.

Семантическая сеть, по мнению специалистов, - наиболее общий способ представления знаний. В ней понятия и классы, а также отношения и связи между ними представлены в виде сети. Семантическая сеть отображает совокупность объектов предметной области и отношений между ними, при этом объектам соответствуют вершины (или узлы) сети, а отношениям - со-

единяющие их дуги. В качестве объектов могут выступать события, действия, обобщенные понятия или свойства объектов. Свойства представляются в сети также в виде вершин и служат для описания классов объектов. Вершины сети соединяются дугой, если соответствующие объекты предметной области находятся в каком-либо отношении.

Как и в системе, основанной на фреймах, в семантической сети могут быть представлены родовидовые отношения, которые позволяют реализовать наследование свойств от объектов-родителей. Это обстоятельство приводит к тому, что семантические сети приобретают большинство недостатков и достоинств представления знаний в виде фреймов. Но основное преимущество семантических сетей заключается в их наглядности и непосредственной связанности понятий через сеть, которая позволяет быстро находить связи понятий и на этой основе управлять принимаемыми решениями. Основной недостаток сетей - сложность обработки исключений.

Выбор модели представления знаний определяется характеристиками предметной области и требованиями, предъявляемыми к системе. В рамках данной работы предложено формализовать знания в виде продукционной модели.

Задание на лабораторную работу

1. Выбрать предметную область и сформулировать перечень вопросов, подлежащих разработке.
2. Провести анализ предметной области и выбрать способ представления знаний.
3. Представить знания в проблемной области в рамках выбранного формализма.
4. Выбрать инструментальное средство разработки экспертной системы.
5. Разработать меню, обеспечивающее взаимодействие пользователя с системой, предусмотрев режимы добавления знаний в базу знаний, просмотра

правил, содержащихся в базе знаний, корректировки знаний, сохранения правил, удаления правил, логического вывода, объяснения логического вывода.

6. Протестировать разработанную систему в режиме ввода дополнительных правил.

7. Протестировать разработанную систему в режиме консультаций.

Практическое занятие №2

Выявление знаний в системах искусственного интеллекта. Фрейм-модели представления знаний.

Цель работы заключается в изучении принципов использования фреймов для представления знаний в базах знаний экспертных систем.

Фреймы - один из распространенных формализмов представления знаний в ЭС. Фрейм можно представить себе как структуру, состоящую из набора ячеек - слотов. Каждый слот состоит из имени и ассоциируемых с ним значений. Значения могут представлять собой данные, процедуры, ссылки на другие фреймы или быть пустыми. Такое построение оказывается очень удобным для моделирования аналогий, описания областей с родовидовыми связями понятий и т.п. Любой фрейм состоит из некоторых составляющих, имена и содержание которых описано ниже:

1. Имя фрейма. Это идентификатор, присваиваемый фрейму, фрейм должен иметь имя уникальное в данной фреймовой системе.

2. Имя слота. Это идентификатор, присваиваемый слоту; слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота.

3. Указатели наследования. Эти указатели касаются только фреймовых систем иерархического типа, основанные на отношениях “абстрактное-конкретное”, они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего

уровня. Типичные указатели наследования Unique (U: - уникальный), Same (S: такой же), Range (R: установление границ), Override (O: игнорировать) и т.п. U показывает, что фрейм может иметь слоты с разными значениями: S - все слоты должны иметь одинаковые значения, R - значение слотов фрейма нижнего уровня должны находиться в пределах, указанных значениями слотов фрейма верхнего уровня, O - при отсутствии указания значение слота фрейма верхнего уровня становится значением слота фрейма нижнего уровня, но в случае определения нового значения слотов фреймов нижних уровней указываются в качестве значений слотов.

4. Указание типа данных. указывается, что слот имеет численное значение, либо служит указателем другого фрейма. К типам данных относятся: FRAME (указатель), INTEGER (целый), REAL (действительный), BOOL (булев), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение) и др.

5. Значение слота. Пункт ввода значения слота. Значение слота должно совпадать с указанным типом данных этого слота, кроме того должно выполняться условие наследования.

6. Демон. Здесь дается определение демонов типа IF-NEEDED, IFADDED, IF-REMOVED и т.д. Демоном называется процедура, автоматически запускаемая при выполнении некоторого условия. демоны запускаются при обращении к соответствующему слоту. Кроме того, демон является разновидностью присоединенной процедуры.

7. Присоединенная процедура. В качестве значения слота можно использовать программу процедурного типа. Когда мы говорим, что в моделях представления знаний фреймами объединяются процедурные и декларативные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

Структура фрейма может выглядеть следующим образом.

ИМЯ ФРЕЙМА			
Имя слота	Значение слота	Способ получения значения	Демон

Особенностью иерархической структуры является то, что информация об атрибутах фрейма на верхнем уровне совместно используется всеми фреймами нижних уровней, связанных с ним.

В качестве примера рассмотрим фреймовое представление понятия конференции. Иерархические фреймовые структуры базируются на отношениях IS – A между фреймами, описывающими некоторую конференцию. Все фреймы должны содержать информацию о ДАТЕ, МЕСТЕ, НАЗВАНИИ ТЕМЫ, ДОКЛАДЧИКЕ. Таким образом, на самом верхнем уровне определен фрейм КОНФЕРЕНЦИЯ. Конференции разделяются на коммерческие и по развитию. Они составляют дочерние фреймы. В них могут быть добавлены слоты: объем торговли и бюджет.

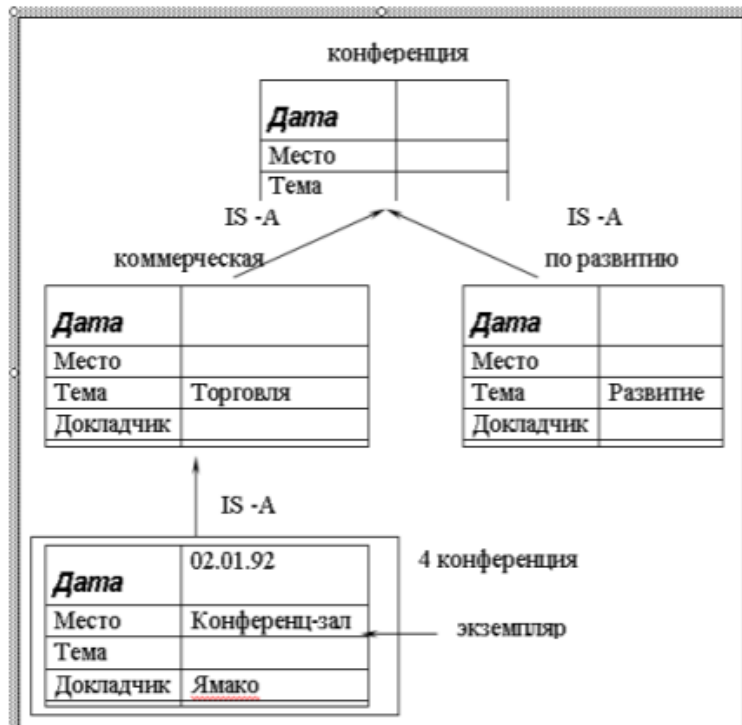


Рис.3. Пример фреймовой модели

Задание на лабораторную работу

Необходимо построить фреймовую модель заданного объекта и реализовать программу с использованием фреймовой модели

Варианты заданий

Используя фреймовую модель представления знаний реализовать структуру отношений, описывающие следующие ситуации:

1. Экзамен по дисциплине за семестр у преподавателя при составляющих: семестр, экзамен, преподаватель, оценка, студент, получать.
2. Ведомость при составляющих: дисциплина, студент, экзамен, семестр, преподаватель, оценка.
3. Конференция по коммерческим вопросам при составляющих: дата, место проведения, тема, цель выступающие.
4. Получение оценки при составляющих: преподаватель, студент, оценка, получать.
5. Использование изделия при составляющих: организация, разработка технологического решения, исследование «физического эффекта», методы создания изделия.
6. Информационная структура БД в машиностроении при составляющих: физические эффекты, технические решения, изделия, объект поставки изделия, приборы и стенды, нормативы.
7. Классификация продукта при составляющих: название, область применения, способ хранения, способ транспортировки.
8. Аудитория (описание) при составляющих: вместимость, назначение, составляющие, местонахождение.
9. Животный мир при составляющих: вид, тип, среда обитания, особенности поведения.

Практическое занятие №3
Выявление знаний в системах искусственного интеллекта.
Нечеткие системы.

Отличительной особенностью естественной компетенции (интеллекта) является способность принимать правильные решения в условиях неполной и нечеткой информации. Построение моделей приближенных размышлений человека и использование их в интеллектуальных системах представляет сегодня одну из важнейших проблем научного направления, получившего название «искусственный интеллект».

Основы нечеткой логики были заложены в конце 60-х лет в работах известного американского математика Латфи Заде. Для создания интеллектуальных систем, в полной мере соответствующих тем требованиям, которые к ним предъявляются и способных адекватно взаимодействовать с человеком, был необходим новый математический аппарат, который переводит неоднозначные жизненные утверждения в язык четких и формальных математических формул. Первым серьезным шагом в этом направлении стала теория нечетких множеств, разработанная Заде. Он заложил основы моделирования интеллектуальной деятельности человека и дал толчок к развитию новой математической теории. Он же дал и название для новой области науки - "fuzzy logic" (fuzzy - нечеткий, размытый, мягкий). Дальнейшие работы профессора Латфи Заде и его последователей заложили фундамент новой теории и создали предпосылки для внедрения методов нечеткого управления в инженерную практику.

Аппарат теории нечетких множеств, продемонстрировав ряд многообещающих возможностей применения - от систем управления летательными аппаратами до прогнозирования итогов выборов, оказался вместе с тем сложным для воплощения. Учитывая имеющийся уровень технологии, нечеткая логика заняла свое место среди других специальных научных дисциплин - где-то посредине между экспертными системами и нейронными сетями.

Свое второе рождение теория нечеткой логики пережила в начале восьмидесятых годов, когда несколько групп исследователей (в основном в США и Японии) всерьез занялись созданием электронных систем различного применения, использующих нечеткие управляющие алгоритмы. Теоретические основы для этого были заложены в ранних работах Коско и других ученых.

Третий период начался с конца 80-х годов и продолжается до сих пор. Этот период характеризуется бумом практического применения теории нечеткой логики в разных сферах науки и техники. Результатом стало появление целого ряда новых массовых микрочипов, базирующихся на нечеткой логике. Сегодня их можно найти в стиральных машинах и видеокамерах, цехах заводов и моторных отсеках автомобилей, в системах управления складскими роботами и боевыми вертолетами. В США развитие нечеткой логики идет по пути создания систем для большого бизнеса и военных. Нечеткая логика применяется при анализе новых рынков, биржевой игре, оценки политических рейтингов, выборе оптимальной ценовой стратегии и т.п. Появились и коммерческие системы массового применения.

Смещение центра исследований нечетких систем в сторону практических применений привело к постановке целого ряда проблем, в частности:

- новые архитектуры компьютеров для нечетких вычислений;
 - элементная база нечетких компьютеров и контроллеров;
 - инструментальные средства разработки;
 - инженерные методы расчета и разработки нечетких систем управления,
- и т.п..

Пусть E - универсальное множество, x - элемент E , а R - определенное свойство. Обычное (четкое) подмножество A универсального множества E , элементы которого удовлетворяют свойству R , определяется как множество упорядоченной пары $A = \{\mu_A(x)/x\}$, где $\mu_A(x)$ - характеристическая функция, принимающая значение 1, когда x удовлетворяет свойству R , и 0 - в другом случае.

Нечеткое подмножество отличается от обычного тем, что для элементов x из E нет однозначного ответа "нет" относительно свойства R . В связи с этим, нечеткое подмножество A универсального множества E определяется как множество упорядоченных пар $A = \{\mu_A(x)/x\}$, где $\mu_A(x)$ - характеристическая функция принадлежности (или просто функция принадлежности), принимающая значение в некотором упорядоченном множестве M (например, $M = [0,1]$).

Функция принадлежности указывает степень (или уровень) принадлежности элемента x к подмножеству A . Множество M называют множеством принадлежностей. Если $M = \{0,1\}$, тогда нечеткое подмножество A может рассматриваться как обычное или четкое множество.

Рассмотрим множество X всех чисел от 0 до 10. Определим подмножество A множества X всех действительных чисел от 5 до 8: $A = [5,8]$. Покажем функцию принадлежности множества A : эта функция ставит в соответствие число 1 или 0 каждому элементу в X в зависимости от того, принадлежит данный элемент подмножеству A или нет. Результат представлен на рисунке 1:

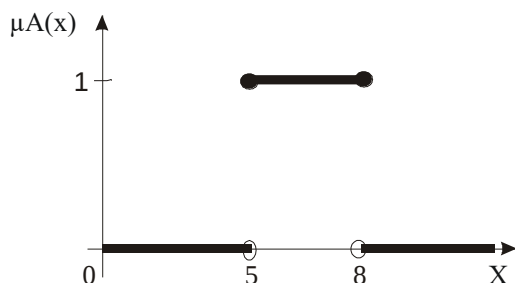


Рис. 1.

Можно интерпретировать элементы, соответствующие 1, как элементы, находящиеся во множестве A , а элементы, соответствующие 0, как элементы, не находящиеся в множестве A . Эта концепция используется во многих областях. Но существуют ситуации, в которых ей будет не хватать гибкости.

Например, опишем множество молодых людей. Формально можно записать так:

$$B = \{\text{множество молодых людей}\}.$$

Т.к. возраст человека начинается с 0, то нижняя граница этого множества должна быть нулем. Верхнюю границу определить сложнее. Сначала установим верхнюю границу равную, например, 20 годам. Таким образом, имеем В как четко ограниченный интервал: $B = [0, 20]$. Возникает следующая структурная проблема: почему человек в двадцать лет - молодой, а сразу на следующий день уже не молодой? Если передвинуть верхнюю границу в другую точку, проблема не решится.

Более естественный путь создания множества В состоит в ослаблении строгого деления на молодых и не молодых. Сделаем это, вынося не только четкие суждения "Да, он принадлежит множеству молодых людей" или "Нет, она не принадлежит множеству молодых людей", но и гибкие формулировки "Да, он принадлежит к довольно молодым людям" или "Нет, он не очень молодой".

Рассмотрим, как с помощью нечеткого множества определить выражение "он еще молодой". В первом примере мы кодировали все элементы множества с помощью 0 или 1. Простым способом обобщить данную концепцию является введение значений между 0 и 1. Реально можно даже допустить бесконечное число значений между 0 и 1, в единичном интервале $I = [0, 1]$. Интерпретация чисел при соотношении всех элементов множества становится теперь сложнее. Конечно, число 1 соответствует элементу, принадлежащему множеству В, а 0 означает, что элемент точно не принадлежит множеству В. Все другие значения определяют степень принадлежности к множеству В. Для наглядности приведем характеристическую функцию множества молодых людей, как и в первом примере (рис.2). Т.е., все 25-летние молоды со степенью 50%.

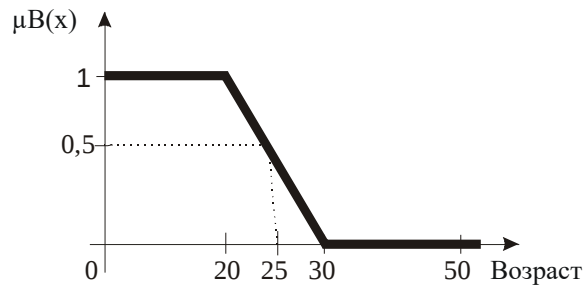


Рис.2.

Пусть $E = \{x_1, x_2, x_3, x_4, x_5\}$, $M = [0,1]$; A - нечеткое множество, для которого

$$\mu_A(x_1)=0,3; \mu_A(x_2)=0; \mu_A(x_3)=1; \mu_A(x_4)=0,5; \mu_A(x_5)=0,9$$

Тогда A можно представить в виде: $A = \{0,3/x_1; 0/x_2; 1/x_3; 0,5/x_4; 0,9/x_5\}$, или $A = 0,3/x_1 + 0/x_2 + 1/x_3 + 0,5/x_4 + 0,9/x_5$, где "+" является операцией объединения, или:

$$A = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ & 0,3 & 0 & 1 & 0,5 & 0,9 \end{matrix}$$

Рассмотрим основные характеристики нечетких множеств. Пусть $M = [0,1]$ и A - нечеткое множество с элементами из универсального множества E и множеством принадлежностей M

Величина $\sup_{x \in E} \mu_A(x)$ называется высотой нечеткого множества A . Нечеткое множество A является нормальным, если его высота равняется 1, то есть верхняя граница его функции принадлежности равняется 1 ($\sup_{x \in E} \mu_A(x) = 1$). При $\mu_A(x) < 1$ нечеткое множество называется субнормальным.

Нечеткое множество является пустым, если $\square \forall x \in E \mu_A(x)=0$. Непустое субнормальное множество можно нормализовать по формуле :

$$\mu_A(x) := \frac{\mu_a(x)}{\sup_{x \in E} \mu_a(x)}$$

Нечеткое множество является унимодальным, если $\mu_A(x)=1$ лишь для одного x из E .

Носителем нечеткого множества A является обычное подмножество со свойством $\mu_A(x) > 0$, то есть носитель $A = \{x / \mu_A(x) > 0\} \subseteq E$.

Элементы $x \in E$, для которых $\mu_A(x) = 0,5$ называются точками перехода множества A .

Примеры нечетких множеств:

1. Пусть $E = \{0,1,2,...,10\}$, $M = [0,1]$. Нечеткое множество "несколько" можно определить таким образом: "несколько" = $0,5/3 + 0,8/4 + 1/5 + 1/6 + 0,8/7 + 0,5/8$. Его характеристики: высота = 1, носитель = $\{3,4,5,6,7,8\}$, точки перехода - $\{3,8\}$.

2. Пусть $E = \{0,1,2,3,...,n,...\}$. Нечеткое множество "малый" можно определить:

$$\mu_{\text{"малый"}}(x) = \begin{cases} \mu_{\text{"малый"}}(n) = \frac{1}{1 + \left(\frac{n}{10}\right)^2} / n \end{cases}$$

3. Пусть $E = \{1,2,3,...,100\}$ и соответствует понятию "возраст", тогда нечеткое множество "молодой", можно определить с помощью:

$$\mu_{\text{"молодой"}}(x) = \begin{cases} 1, x \in [1,25] \\ 1 + \left(\frac{x-25}{5}\right)^2, x \geq 25 \end{cases}$$

Нечеткое множество "молодой" на универсальном множестве $E' = \{\text{Иванов, Петров, Сидоров},...\}$ задается с помощью функции принадлежности $\mu_{\text{"молодой"}}(x)$ на $E = \{1,2,3,...,100\}$ (возраст), что называется относительно E' функцией совместимости, при этом: $\mu_{\text{"молодой"}}(\text{Сидоров}) = \mu_{\text{"молодой"}}(x)$, где x - возраст Сидорова.

4. Пусть $E = \{\text{Запорожец, Жигули, Мерседес},...\}$ - множество марок автомобилей, а $E' = [0,\mu]$ - универсальное множество "стоимость", тогда на E' мы можем определить нечеткие множества типа: "для небогатых", "для среднего класса", "престижные", с функциями принадлежности типа (рис. 3):

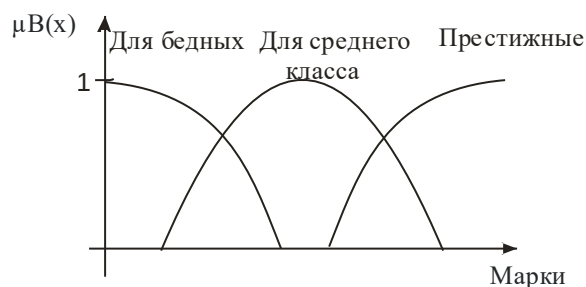


Рис.3.

Имея эти функции и зная цены автомобилей из E в данный момент времени, определим на E' нечеткие множества с этими же названиями. Так, например, нечеткое множество "для небогатых", заданное на универсальном множестве $E = \{\text{Запорожец, Жигули, Мерседес, ...}\}$ выглядит таким образом (рис.4):



Рис.4.

Аналогично можно определить нечеткое множество "скоростные", "средние", "тихоходные" и т.д.

В приведенных выше примерах использованы прямые методы, когда эксперт или просто задает для любого $x \in E$ значение $\mu_A(x)$, или определяет функцию принадлежности. Как правило, прямые методы задания функции принадлежности используются для измеримых понятий, таких как скорость, час, расстояние, давление, температура и т.д., то есть когда выделяются полярные значения. Во многих задачах при характеристике объекта можно выделить набор признаков и для любого из них определить полярные значения, отвечающие значениям функции принадлежности 0 или 1.

Например, в задаче распознавания лица можно выделить следующие пункты:

		0	1
x1	высота лба	низкий	широкий
x2	профиль носа	курносый	горбатый
x3	длина носа	короткий	длинный
x4	разрез глаз	узкий	широкий
x5	цвет глаз	светлый	темный
x6	форма подбородка	острый	квадратный
x7	толщина губ	тонкие	толстые
x8	цвет лица	темный	светлый
x9	овал лица	овальное	квадратное

Для конкретного лица А эксперт, исходя из приведенной шкалы, задает $\mu_A(x) \in [0,1]$, формируя векторную функцию принадлежности $\{\mu_A(x1), \mu_A(x2), \dots, \mu_A(x9)\}$.

Косвенные методы определения значений функции принадлежности используются в случаях, когда нет элементарных измеримых свойств для определения нечеткого множества. Как правило, это методы попарных сравнений. Если бы значения функций принадлежности были известны, например, $\mu_A(x_i) = w_i, i=1,2,\dots,n$, тогда попарные сравнения можно представить матрицей отношений $A = \{a_{ij}\}$, где $a_{ij} = w_i/w_j$ (операция деления).

Рассмотрим основные операции над нечеткими множествами. Пусть А и В - нечеткие множества на универсальном множестве Е. Говорят, что А **содержится** в В, если $\forall x \in E \mu_A(x) < \mu_B(x)$. Обозначение: $A \subset B$. Иногда используют термин "доминирование", то есть в случае если $A \subset B$, говорят, что В доминирует А.

Нечеткие множества А и В **равны**, если $\square \forall x \in E \mu_A(x) = \mu_B(x)$. Обозначение: $A = B$.

Пусть $M = [0,1]$, А и В - нечеткие множества, заданные на Е. А и В **дополняют** друг друга, если $\forall x \in E \square \mu_A(x) = 1 - \mu_B(x)$. Обозначение: $B = \bar{A}$ или

$A = \bar{\bar{A}}$. Очевидно, что $\bar{\bar{A}} = A$. Дополнение определено для $M = [0,1]$, но очевидно, что его можно определить для любого упорядоченного M .

Пересечением двух нечетких множеств $A \cap B$ называется наибольшее нечеткое подмножество, которое содержится одновременно в A и B с функцией принадлежности:

$$\mu(A(x) \cap B(x)) = \min(\mu A(x), \mu B(x)).$$

Объединением двух нечетких множеств $A \cup B$ называется наименьшее нечеткое подмножество, которое включает как A , так и B с функцией принадлежности:

$$\mu(A(x) \cup B(x)) = \max(\mu A(x), \mu B(x)).$$

Разностью двух нечетких множеств является: $A - B = A \cap \bar{B}$ с функцией принадлежности $\mu(A(x) - B(x)) = \mu(A(x) \cap \bar{B}(x)) = \min(\mu A(x), 1 - \mu B(x))$.

Дизъюнктивной суммой $A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$ с функцией принадлежности $\mu(A(x) \oplus B(x)) = \max\{\min\{\mu A(x), 1 - \mu B(x)\}; \min\{1 - \mu A(x), \mu B(x)\}\}$

Например, пусть:

$$A = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4;$$

$$B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4;$$

$$C = 0,1/x_1 + 1/x_2 + 0,2/x_3 + 0,9/x_4.$$

Здесь:

1. $A \subset B$, то есть A содержится в B или B доминирует A . Нечеткое множество C несравнимо ни с A , ни с B , то есть пары $\{A, C\}$ и $\{A, C\}$ - пары недоминируемых нечетких множеств.

$$2. A \neq B \neq C.$$

$$3. \bar{A} = 0,6/x_1 + 0,8/x_2 + 1/x_3 + 0/x_4;$$

$$\bar{B} = 0,3/x_1 + 0,1/x_2 + 0,9/x_3 + 0/x_4.$$

$$4. A \cap B = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4.$$

$$5. A \cup B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4.$$

$$6. A - B = A \cap \bar{B} = 0,3/x_1 + 0,1/x_2 + 0/x_3 + 0/x_4;$$

$$B - A = B \cap \bar{A} = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4.$$

$$7. A \oplus B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4.$$

Для нечетких множеств можно применить визуальное представление. Рассмотрим прямоугольную систему координат, на оси ординат которой откладываются значения $\mu_A(x)$, на оси абсцисс в произвольном порядке расположены элементы E . Если E по своей природе упорядочено, то этот порядок желательно сохранить в расположении элементов на оси абсцисс. Такое представление делает наглядными простые операции над нечеткими множествами.

Пусть A - нечеткий интервал от 5 до 8, а B - нечеткое число около 4, как показано на рис.5.

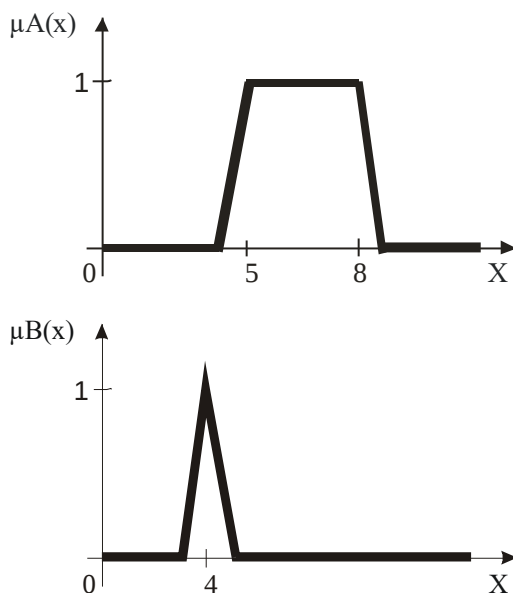


Рис.5.

Проиллюстрируем нечеткое множество $A \cap B$:
(от 5 до 8)И(около 4) (рис.6).

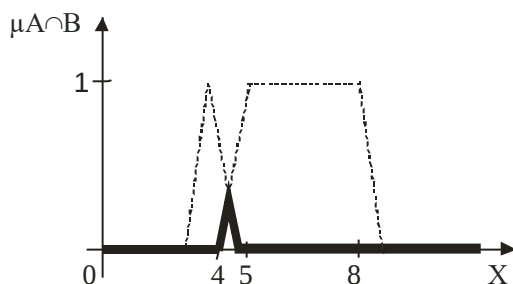


Рис.6.

Нечеткое множество $A \cup B$: (от 5 до 8) ИЛИ (около 4) показано на рис.7.

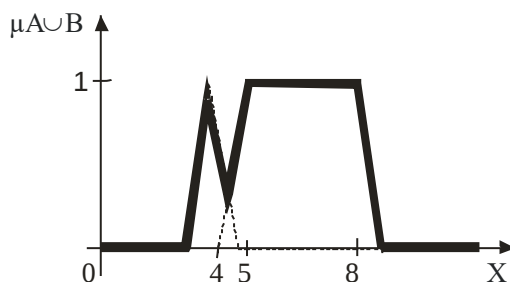


Рис.7.

Рис.8 иллюстрирует операцию отрицания нечеткого множества A .

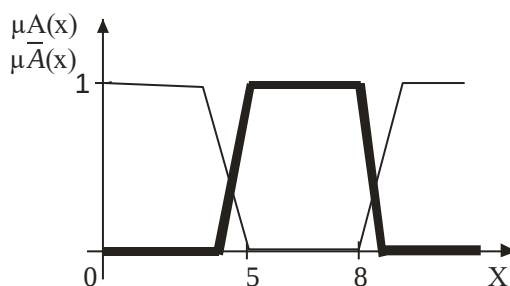


Рис.8.

На рис.9 заштрихованная часть соответствует нечеткому множеству A и изображает область значений A и всех нечетких множеств, содержащихся в A . Остальные рисунки изображают соответственно $\bar{A}$, $A \cap \bar{A}$, $A \cup \bar{A}$.

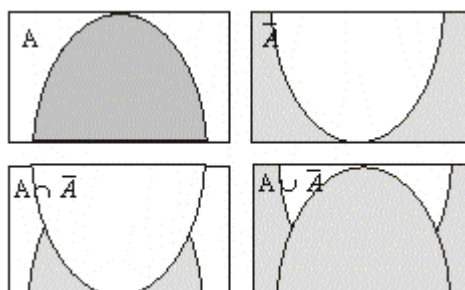


Рис.9.

Рассмотрим свойства операций $\cap$ и $\cup$. Пусть A, B, C - нечеткие множества, тогда выполняются следующие свойства:

$$\left. \begin{aligned} A \cap B &= B \cap A \\ A \cup B &= B \cup A \end{aligned} \right\} - \text{коммутативность};$$

$$\left. \begin{aligned} (A \cap B) \cap C &= A \cap (B \cap C) \\ (A \cup B) \cap C &= A \cap (B \cap C) \end{aligned} \right\} - \text{ассоциативность};$$

$$\left. \begin{aligned} A \cap A &= A \\ A \cup A &= A \end{aligned} \right\} - \text{идемпотентность};$$

$$\left. \begin{aligned} A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \end{aligned} \right\} - \text{дистрибутивность};$$

$$A \cap \square = \square, \text{ где } \square - \text{пустое множество, то есть } \mu_{\square}(x) = 0 \quad \forall x \in E;$$

$$A \cup \square = A;$$

$$A \cap E = A, \text{ где } E - \text{универсальное множество};$$

$$A \cup E = E;$$

$$\left. \begin{aligned} \overline{A \cap B} &= \bar{A} \cup \bar{B} \\ \overline{A \cup B} &= \bar{A} \cap \bar{B} \end{aligned} \right\} - \text{теоремы де Моргана.}$$

В отличие от четких множеств, для нечетких множеств в общем случае (это проиллюстрировано выше в примере представления нечетких множеств):

$$A \cap \bar{A} \neq \square,$$

$$A \cup E \neq E.$$

При работе с лингвистическими переменными используются следующие операции (рис.10):

$$\text{CON}(A) = A^2 - \text{операция концентрирования},$$

$$\text{DIL}(A) = A^{0,5} - \text{операция размывания}.$$

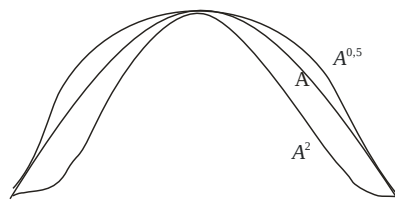


Рис.10.

Если α - положительное число, такое, что $\alpha \max_{x \in A} \mu_A(x) \leq 1$, тогда нечеткое множество αA имеет функцию принадлежности $\mu_{\alpha A}(x) = \alpha \mu_A(x)$.

При описании объектов и явлений с помощью нечетких множеств используется понятие *нечеткой и лингвистической переменных*.

Нечеткая переменная характеризуется тройкой $\langle \alpha, X, A \rangle$, где

α - имя переменной,

X - универсальное множество (область определения α),

A - нечеткое множество на X , описывающее ограничение (то есть $\mu_A(x)$) на значение нечеткой переменной α .

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$, где

β - имя лингвистической переменной;

T - множество её значений (терм-множество), представляющие имена нечетких переменных, областью определения которых является множество X . Множество T называется базовым терм-множеством лингвистической переменной;

G - синтаксическая процедура, позволяющая оперировать элементами терм-множества T , в частности, генерировать новые термы (значения). Множество $T \cup G(T)$, где $G(T)$ - множество сгенерированных термов, называется расширенным терм-множеством лингвистической переменной;

M - семантическая процедура, позволяющая преобразовать новое значение лингвистической переменной, образованной процедурой G , в нечеткую переменную, то есть сформировать соответствующее нечеткое множество.

Во избежание большого количества символов символ β используют как для названия самой переменной, так и для всех его значений, а для обозначения нечеткого множества и его названия пользуются одним символом, например, терм "молодой", является значением лингвистической переменной b = "возраст", и одновременно нечетким множеством M ("молодой"). Присваивание нескольких значений символам предполагает, что контекст допускает неопределенности.

Например, пусть эксперт определяет толщину изделия, с помощью понятия "маленькая толщина", "средняя толщина" и "большая толщина", при этом минимальная толщина равняется 10 мм, а максимальная - 80 мм.

Формализация этого описания может быть проведена с помощью лингвистической переменной $\langle \beta, T, X, G, M \rangle$, где

β - толщина изделия;

T - {"маленькая толщина", "средняя толщина", "большая толщина"};

X - [10, 80];

G - процедура образования новых термов с помощью связок "и", "или" и модификаторов типа "очень", "не", "слегка" и др. Например, "маленькая или средняя толщина", "очень маленькая толщина" и др.;

M - процедура задания на $X = [10, 80]$ нечетких подмножеств $A1$ ="маленькая толщина", $A2$ = "средняя толщина", $A3$ ="большая толщина", а также нечетких множеств для термов из $G(T)$ соответственно правилам трансляции нечетких связок и модификаторов "и", "или", "не", "очень", "слегка", операции над нечеткими множествами вида: $A \cap C$, $A \cup C$, $\bar{A}$, $CON(A) = A^2$, $DIL(A) = A^{0.5}$.

Вместе с рассмотренными выше базовыми значениями лингвистической переменной "толщина" ($T=\{"маленькая толщина", "средняя толщина", "большая толщина"\}$) существуют значения, зависящие от области определения X . В данном случае значения лингвистической переменной "толщина изделия" могут быть определены как "около 20 мм", "около 50 мм", "около 70 мм", то есть в виде нечетких чисел.

Функции принадлежности нечетких множеств: "маленькая толщина" = $A1$, "средняя толщина" = $A2$, "большая толщина" = $A3$ (рис.11).

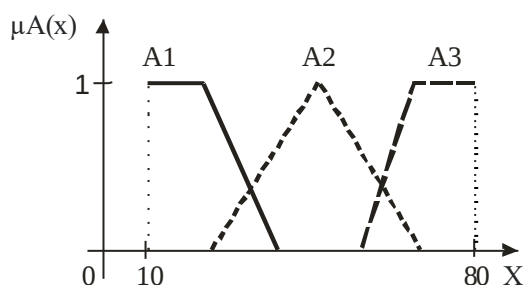


Рис.11.

Функция принадлежности нечеткого множества "маленькая или средняя толщина" $A1 \cup A2$ (рис.12).

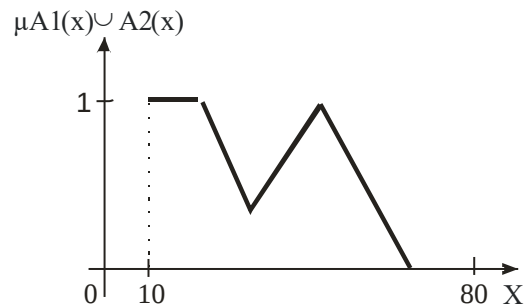


Рис.12.

Нечеткими высказываниями называются высказывания следующего вида:

- высказывание $\langle \beta \text{ есть } \beta' \rangle$, где β - имя лингвистической переменной, β' - ее значение, которому соответствует нечеткое множество на универсальном множестве X . Например, высказывание $\langle \text{давление большое} \rangle$ предполагает, что лингвистической переменной "давление" предоставляется значение "большой", для которого на универсальном множестве X переменной "давление" определено, соответственно данному значению "большой", нечеткое множество;

- высказывание $\langle \beta \text{ есть } m\beta' \rangle$, где m - модификатор, которому соответствуют слова "очень", "более или менее", "намного больше" и др.;

- сложные высказывания, образованные из высказываний вида 1 и 2 и операций "И", "ИЛИ", "ЕСЛИ.., ТО...", "ЕСЛИ.., ТО.., ИНАЧЕ".

Если значения фиксированной лингвистической переменной соответствуют нечетким множествам одного универсального множества X , можно отождествлять модификаторы "очень" или "нет" с операциями "CON" и "дополнение", а союзы "И", "ИЛИ" с операциями "пересечение" и "объединение" над нечеткими множествами.

Для иллюстрации понятия лингвистической переменной мы в качестве примера рассматривали лингвистическую переменную "толщина изделия" с базовым терм-множеством $T = \{\text{"маленькая"}, \text{"средняя"}, \text{"большая"}\}$. При этом

на $X = [10, 80]$ мы определили нечеткие множества A_1, A_2, A_3 , соответствующие базовым значениям: "маленькая", "средняя", "большая".

В этом случае высказыванию <толщина изделия очень маленькая> отвечает нечеткое множество $CON(A)=A^2$; высказыванию <толщина изделия не большая или средняя> - нечеткое множество $A_2 \cup \overline{A_3}$; высказыванию <толщина изделия не маленькая и не большая> $\overline{A_1} \cap \overline{A_3}$.

Наиболее важным применением теории нечетких множеств являются контроллеры нечеткой логики. Их функционирование несколько отличается от работы обычных контроллеров; для описания системы вместо дифференциальных уравнений используются знания экспертов. Эти знания могут быть выражены с помощью лингвистических переменных, которые описаны нечеткими множествами.

Общая структура микроконтроллера, использующего нечеткую логику, показана на рис.13. Она содержит блок фаззификации; базу знаний; блок решений; блок дефаззификации. Блок фаззификации преобразует четкие величины, измеренные на выходе объекта управления, в нечеткие величины, которые описаны лингвистическими переменными в базе знаний. Блок решений использует нечеткие условные (if - then) правила, заложенные в базу знаний, для преобразования нечетких входных данных в необходимые управляющие влияния, которые также носят нечеткий характер. Блок дефаззификации превращает нечеткие данные с выхода блока решений в четкую величину, которая используется для управления объектом.

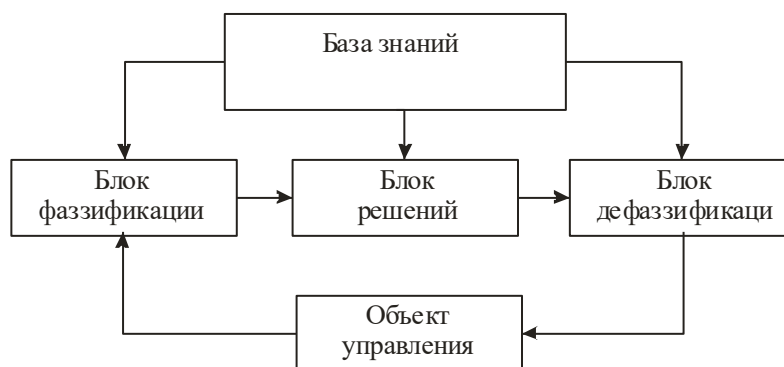


Рис. 13. Общая структура нечеткого микроконтроллера

Все системы с нечеткой логикой функционируют по одному принципу: показания измерительных приборов фаззифицируются (превращаются в нечеткий формат), обрабатываются, дефаззифицируются и в виде обычных сигналов подаются на исполнительные устройства.

Задание на лабораторную работу

1. Выбрать лингвистическую переменную (например, вес, рост, возраст и др.) и задать для нее нечеткие множества.

Рассмотрим в качестве примера лингвистическую переменную «Температура». Множеством её значений (терм-множеством Т), представляющим имена нечетких переменных может быть: $T = \{\text{«низкая»}, \text{«норма»}, \text{«высокая»}\}$. Областью определения для этих нечетких переменных является множество $X = [35; 36; 37; 38; 39; 40]$. В результате экспертного оценивания (четырьмя экспертами) получили матрицу определения расплывчатого подмножества для лингвистической переменной «Температура» и соответствующие функции принадлежности нечетких переменных (оценки выбраны условно):

$$A_T = \begin{bmatrix} \text{мннббб} \\ \text{ннннбб} \\ \text{мнннбб} \\ \text{мнннбб} \end{bmatrix}$$

$$\mu_i = \left\{ \frac{3}{4}, 0, 0, 0, 0, 0 \right\}, \mu_{\bar{i}} = \left\{ \frac{1}{4}, 1, 1, \frac{3}{4}, \frac{1}{4}, 0 \right\}, \mu_{\bar{a}} = \left\{ 0, 0, 0, \frac{1}{4}, \frac{3}{4}, 1 \right\}$$

Построим функции принадлежности нечетких множеств "малая"(A1), "норма"(A2), "большая"(A3) для лингвистической переменной «Температура» с использованием средств Excel.

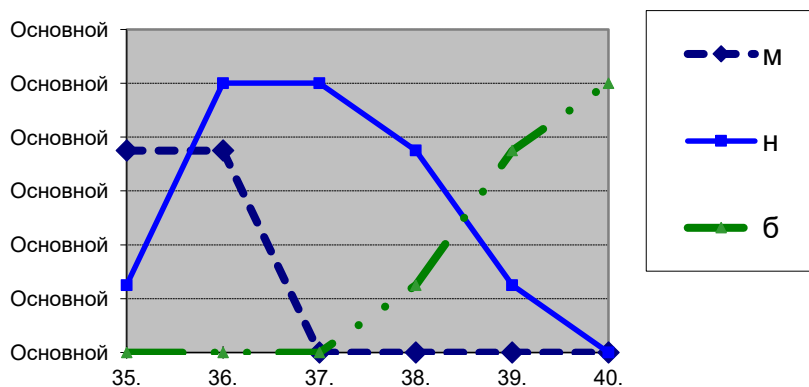
	A	B	C	D	E	F	G
1	Всего экспертов:						
2	4						
3	Экспертные		Числовая шкала				

4	оценки	35	36	37	38	39	40
5	1-й эксперт	м	н	н	б	б	б
6	2-й эксперт	н	н	н	н	б	б
7	3-й эксперт	м	н	н	н	н	б
8	4-й эксперт	м	м	н	н	б	б
9							
10							
11	Нечеткие множества	35	36	37	38	39	40
12	«малая»	0,75	0,75	0	0	0	0
13	«норма»	0,25	0,75	1	0,75	0,25	0
14	«большая»	0	0	0	0,25	0,75	1

Для вычисления значений функций принадлежности в клетках b12:g14 используется функция "счет, если", отнесенная к числу экспертов - клетка A2. Фрагмент формул приведен для значения температуры 36⁰С и графики для полученных значений функций принадлежности нечетких множеств.

36
=СЧЁТЕСЛИ(C5C8;A12)/A2
=СЧЁТЕСЛИ(C5:C8;A13)/A2
=СЧЁТЕСЛИ(C5:C8;A14)/A2

Функции принадлежности



2. Построить таблицу экспертных оценок и провести их обработку по аналогии с приведенным примером.

3. Построить графики функций принадлежности нечетких множеств.
4. Исследовать, как изменяются функции принадлежности при изменении экспертных оценок.
5. По сочетанию параметров построить конъюнктивную или дизъюнктивную модель для оценки свойств объекта (используя, например, операции объединения, пересечения, отрицания).

Практическое занятие №4

Нейронные сети в системах искусственного интеллекта.

Моделирование работы однослойного персептрона.

Цель лабораторной работы - изучение основных принципов функционирования однослойного персептрона, получение практических навыков по его программной реализации.

Основу каждой искусственной нейронной сети (ИНС) составляют относительно простые, в большинстве случаев – однотипные, элементы (ячейки), имитирующие работу нейронов мозга. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рисунке 1.

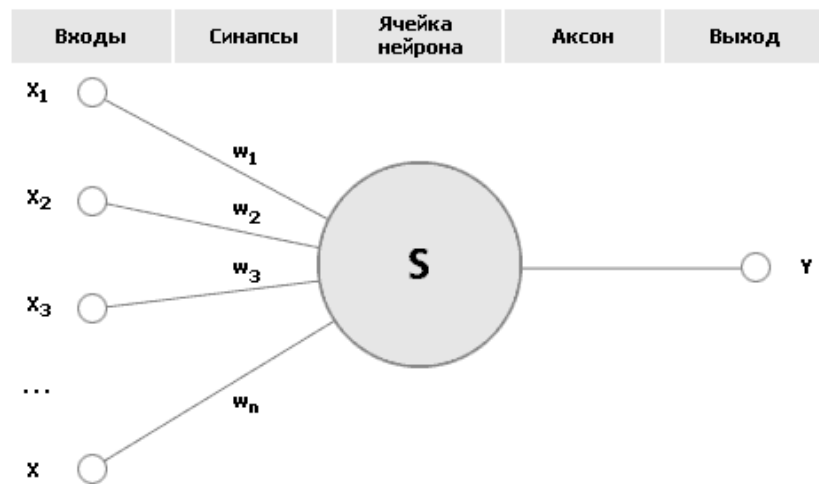


Рис 1. Искусственный нейрон

Текущее состояние нейрона определяется, как взвешенная сумма его входов:

$$s = \sum_{i=1}^n x_i \cdot w_i . \quad (1)$$

Выход нейрона есть функция его состояния:

$$y = f(s) . \quad (2)$$

Нелинейная функция f называется активационной и может иметь различный вид, как показано на рисунке 2.

Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т.е. функция S-образного вида):

$$f(x) = \frac{1}{1 + e^{-\alpha x}} . \quad (3)$$

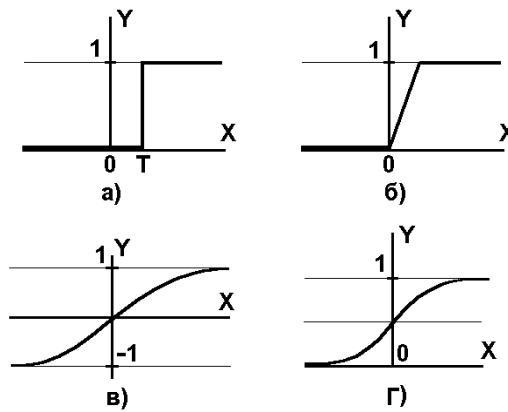


Рис. 2. а) функция единичного скачка; б) линейный порог (гистерезис); в) сигмоид – гиперболический тангенс; г) сигмоид – формула (3)

При уменьшении α сигмоид становится более пологим, в пределе при $\alpha=0$ вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке $x=0$. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне $[0,1]$. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной:

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)). \quad (4)$$

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

В качестве примера простейшей ИНС рассмотрим трехнейронный перцептрон (рис.3), то есть такую сеть, нейроны которой имеют активационную функцию в виде единичного скачка. На n входов поступают некие сигналы, проходящие по синапсам на 3 нейрона, образующие единственный слой этой ИНС и выдающие три выходных сигнала:

$$y_j = f \left[\sum_{i=1}^n x_i \cdot w_{ij} \right], \quad j=1...3 \quad . \quad (5)$$

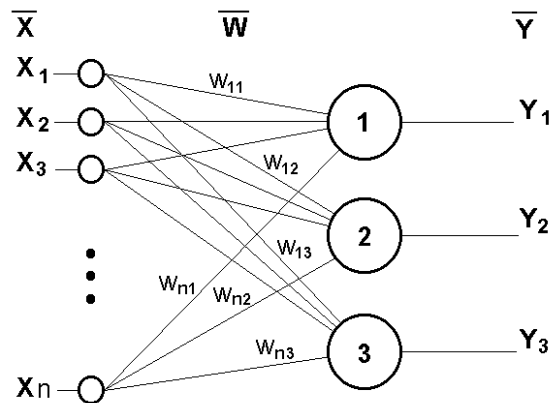


Рис.3. Однослойный перцептрон

Очевидно, что процесс функционирования ИНС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов. Этот этап называется обучением ИНС. На этапе обучения кроме параметра качества подбора весов важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса.

Обучение ИНС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы ИНС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует великое множество различных алгоритмов обучения, которые, однако, делятся на два больших класса: детерминистские и стохастические. В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором – она производится на основе действий, подчиняющихся некоторому случайному процессу.

Из рисунка функции единичного скачка видно, что пороговое значение T , в общем случае, может принимать произвольное значение. Более того, оно должно принимать некое произвольное, неизвестное заранее значение, которое подбирается на стадии обучения вместе с весовыми коэффициентами. То же самое относится и к центральной точке сигмоидной зависимости, которая может сдвигаться вправо или влево по оси X , а также и ко всем другим активационным функциям. Это, однако, не отражено в формуле (1), которая должна была бы выглядеть так:

$$s = \sum_{i=1}^n x_i \cdot w_i - T \quad . \quad (6)$$

Такое смещение обычно вводится путем добавления к слою нейронов еще одного входа, возбуждающего дополнительный синапс каждого из нейронов, значение которого всегда равняется 1. Присвоим этому входу номер 0. Тогда

$$s = \sum_{i=0}^n x_i \cdot w_i \quad , \quad (7)$$

где $w_0 = -T$, $x_0 = 1$.

Очевидно, что различие формул (1) и (6) состоит лишь в способе нумерации входов.

Работа всех сетей сводится к классификации (обобщению) входных сигналов, принадлежащих n -мерному гиперпространству, по некоторому числу классов. С математической точки зрения это происходит путем разбиения гиперпространства гиперплоскостями (запись для случая однослойного перцептрона)

$$\sum_{i=1}^n x_i \cdot w_{ik} = T_k \quad , \quad k=1...m \quad . \quad (8)$$

Каждая полученная область является областью определения отдельного класса. Число таких классов для одной ИНС перцептронного типа не превышает 2^m , где m – число выходов сети. Однако не все из них могут быть разделены данной ИНС.

Например, однослойный перцептрон, состоящий из одного нейрона с двумя входами, представленный на рисунке 4, не способен разделить плоскость (двумерное гиперпространство) на две полуплоскости так, чтобы осуществить классификацию входных сигналов по классам А и В (таблица 1).

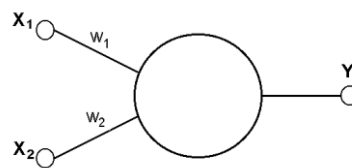


Рис.4. Однонейронный перцептрон

Таблица 1

x_1	x_2	
	0	1
0	А	В
1	В	А

Уравнение сети для этого случая

$$x_1 \cdot w_1 + x_2 \cdot w_2 = T \quad (9)$$

является уравнением прямой (одномерной гиперплоскости), которая ни при каких условиях не может разделить плоскость так, чтобы точки из множества входных сигналов, принадлежащие разным классам, оказались по разные стороны от прямой (рис. 5).

Рассмотрим вопрос обучения ИНС на примере однослойного перцептрона с рисунка 3.

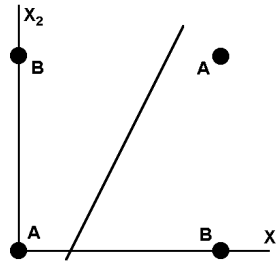


Рис.5. Визуальное представление работы ИНС

Рассмотрим алгоритм обучения с учителем.

1. Проинициализировать элементы весовой матрицы (обычно небольшими случайными значениями).
2. Подать на входы один из входных векторов, которые сеть должна научиться различать, и вычислить ее выход.
3. Если выход правильный, перейти на шаг 4.

Иначе вычислить разницу между идеальным и полученным значениями выхода:

$$\delta = Y_I - Y \quad . \quad (10)$$

Модифицировать веса в соответствии с формулой:

$$w_{ij}(t+1) = w_{ij}(t) + \nu \cdot \delta \cdot x_i \quad , \quad (11)$$

где t и $t+1$ – номера соответственно текущей и следующей итераций; ν – коэффициент скорости обучения, $0 < \nu < 1$; i – номер входа; j – номер нейрона в слое.

Очевидно, что если $Y_I > Y$ весовые коэффициенты будут увеличены и тем самым уменьшат ошибку. В противном случае они будут уменьшены, и Y тоже уменьшится, приближаясь к Y_I .

4. Цикл с шага 2, пока сеть не перестанет ошибаться.

Для выполнения лабораторной работы необходимо:

1. Изучить теоретическое введение.
2. Разработать программные процедуры моделирования функционирования однослойного персептрона и его обучения на основе предложенного алгоритма
3. Протестировать программы на обучающей выборке данных.
4. Провести анализ влияния структуры и параметров сети на скорость обучения.