

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ильшат Ринатович Мухаметзянов

Должность: директор

Дата подписания: 13.07.2023 14:34:25

Уникальный идентификатор:

aba80b84033c9ef196388e9ea0434f90a83a40954ba270e84bcb664f02d1d8d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования «Казанский национальный исследовательский**

**технический
университет им. А.Н. Туполева-КАИ»
(КНИТУ-КАИ)
Чистопольский филиал «Восток»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ
по дисциплине
ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ЭКСПЕРТНЫХ
ИЗМЕРИТЕЛЬНЫХ СИСТЕМАХ**

Индекс по учебному плану: **Б1.О.23**

Направление подготовки: **12.03.01 Приборостроение**

Квалификация: **Бакалавр**

Профиль подготовки: **Приборостроение**

Типы задач профессиональной деятельности: **производственно-технологический, проектно-конструкторский**

Рекомендовано УМК ЧФ КНИТУ-КАИ

Чистополь
2023 г.

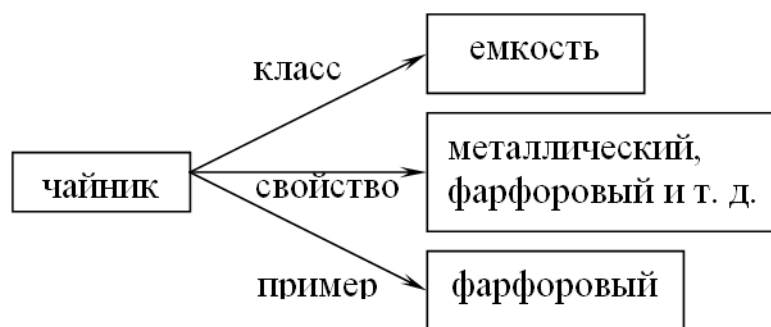
Лабораторная работа №1

Формализация знаний

Использование семантических сетей для представления знаний

Семантическая сеть – это один из способов представления знаний. Изначально семантическая сеть была задумана как модель представления долговременной памяти в психологии, но впоследствии стала одним из способов представления знаний в ЭС.

Семантика – означает общие отношения между символами и объектами из этих символов.



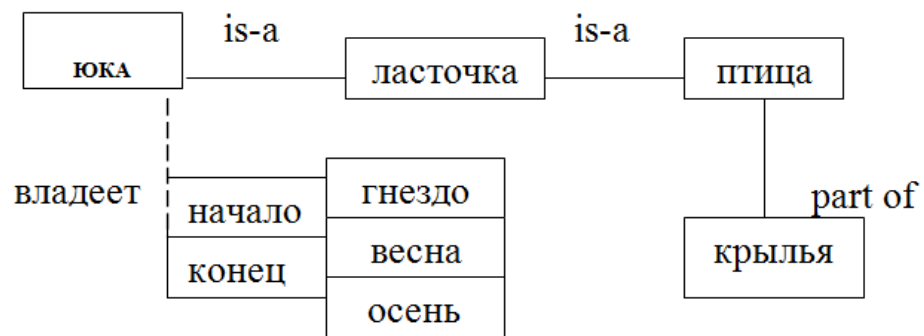
Простейший образец семантической сети

Вершины – это объекты, дуги – это отношения. Семантическая модель не раскрывает сама по себе каким образом осуществляется представление знаний. Поэтому семантическая сеть рассматривается как метод представления знаний и структурирования знаний. При расширении семантической сети в ней возникают другие отношения:

IS – A (принадлежит) и PART OF (является частью) отношение:

целое → часть.

Ласточка IS – A птица, «нос» PART OF «тело». Например:



Расширение семантической сети

Могут быть и другие отношения: владеет. Тогда семантическая сеть расширяется иерархически (вершина имеет две ветви). Кроме того, можно расширить сеть и другим отношением:

период → «весна – лето».

Получается иерархическая структура понятия ЮКО. Можно разбить на подсхемы. Большой проблемой для семантических сетей является то, что результат вывода не гарантирует достоверности, так как вывод есть просто наследование свойств ветви is-a.

Для отображения иерархических отношений между объектами и введения единой семантики в семантические сети было предложено использовать процедурные сети. Сеть строится на основе класса (понятия); вершины, дуги и процедуры представлены как объекты.

Используя соответствующие дуги построить семантическую сеть, касающуюся:

1. географии какого-либо региона. Дуги: государство, страна, континент, широта.
2. диагностики глазных заболеваний. Дуги: категории болезней, патофизиологическое состояние, наблюдения, симптомы.
3. распознавания химических структур. Дуги: формула вещества, свойства вещества, область применения, меры предосторожности.

4. процедуры поиска полезных ископаемых. Дуги: наименование ископаемого, расположение месторождения, глубина залегания, методы добычи.

5. судебной процедуры. Дуги: юридическое лицо, событие, меры воздействия, способы расследования.

6. распределения продуктов по магазинам. Дуги: источник снабжения, наименование продукта, способ транспортировки, конечный пункт транспортировки.

7. определения принадлежности животного к определенному виду, типу, семейству. Дуги: место обитания, строение, особенности поведения, вид питания.

8. классификации пищевых продуктов. Дуги: наименование продукта, составляющие части, способ приготовления, срок хранения.

9. распознавания типа компьютера. Дуги: страна изготовитель, стандартная конфигурация, область применения, используемое программное обеспечение.

10. иерархической структуры БД. Дуги: система, состояние, назначение, взаимодействие составляющих.

Использование фреймов для представления знаний

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, IF-ADDED, 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.

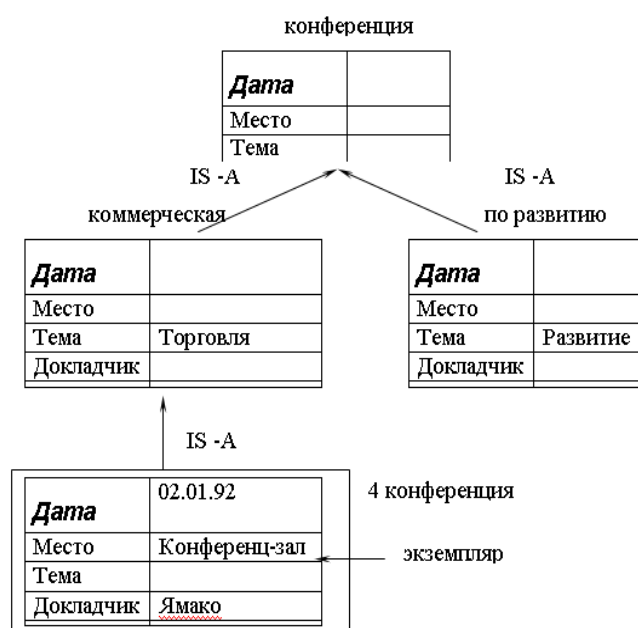
7. Присоединенная процедура. В качестве значения слота можно использовать программу процедурного типа. Когда мы говорим, что в моделях представления знаний фреймами объединяются процедурные и декларативные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

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

Например: Фреймовое представление конференции.

Иерархические фреймовые структуры базируются на отношениях IS – А 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.



Пример фреймовой модели

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. использования 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.

Лабораторная работа №2

Реализация нечеткой модели

Работа с оболочкой проектирования нечетких систем CubiCalc

Нечеткое управление. Нечеткие и лингвистические переменные.

Управляющие контроллеры, построенные на принципах нечеткой логики – наиболее важное применение теории нечетких множеств. Отличие их функционирования от обычных контроллеров заключается в том, что для описания системы управления используются знания экспертов, выраженные в лингвистической форме. Эти знания могут быть выражены естественным образом в виде лингвистических переменных, которые принимают в качестве своих значений слова и выражения естественного языка, их значениями являются нечеткие переменные.

Понятие нечеткой и лингвистической переменных используется при описании объектов и явлений с помощью нечетких множеств.

Нечеткая переменная характеризуется тройкой $\langle \alpha, X, \tilde{A} \rangle$, где α – наименование переменной, X – универсальное множество, \tilde{A} – нечеткое множество на X , описывающее ограничения на значения нечеткой переменной α .

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$, где β – наименование лингвистической переменной;

T – множество ее значений (терм-множество), представляющих собой наименования нечетких переменных, областью определения каждой из 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). M – семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G , в нечеткую переменную, т.е. сформировать соответствующее нечеткое множество.

Замечание. Чтобы избежать большого количества символов

символ β используют как для названия самой переменной, так и для всех ее значений;

пользуются одним и тем же символом для обозначения нечеткого множества и его названия, например терм "*молодой*", являющийся значением лингвистической переменной $\beta = \text{"возраст"}$, одновременно есть и нечеткое множество M ("*молодой*").

Пример: Пусть эксперт определяет толщину выпускаемого изделия с помощью понятий "*малая толщина*", "*средняя толщина*" и "*большая толщина*", при этом минимальная толщина равна 10 мм, а максимальная - 80 мм. Формализация такого описания может быть проведена с помощью следующей лингвистической переменной $\langle \beta, T, X, G, M \rangle$, где

β - толщина изделия; T - {"*малая 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: "*малая или средняя толщина*", "*очень малая толщина*" и др.; M - процедура задания на $X = [10, 80]$ нечетких подмножеств $A_1 = \text{"малая толщина"}$, $A_2 = \text{"средняя толщина"}$, $A_3 = \text{"большая толщина"}$, а также нечетких множеств для термов из $G(T)$ в соответствии с правилами трансляции нечетких связок и модификаторов "*и*", "*или*", "*не*", "*очень*", "*слегка*" и др. операции над нечеткими множествами вида: $A \cap B$, $A \cup B$, A^2 , $A^{0.5}$ и др.

Функции принадлежности нечетких множеств:

"*малая толщина*" = A_1 , "*средняя толщина*" = A_2 , "*большая толщина*" = A_3 представлены на рис. 1.

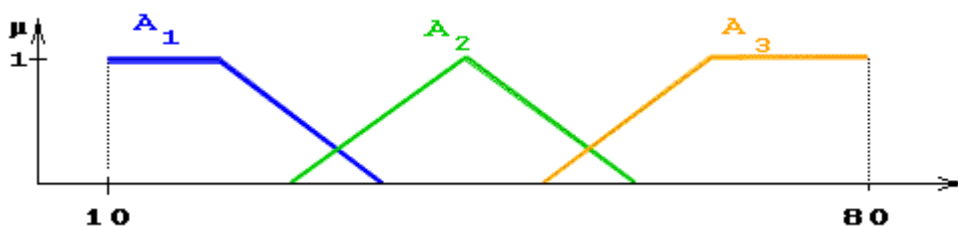


Рис. 1. Функции принадлежности значений лингвистической переменной «Толщина»

Функция принадлежности нечеткого множества "малая или средняя толщина" представлена на рис. 2.

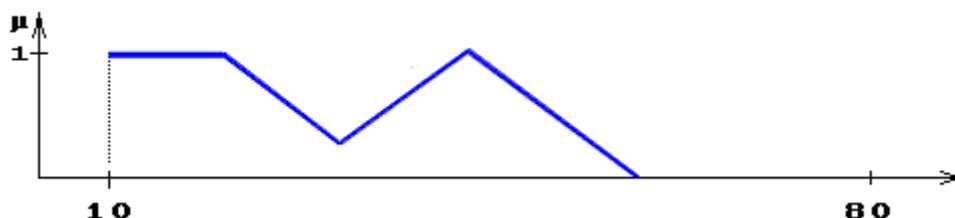


Рис. 2. Функция принадлежности понятия «Малая или средняя толщина»

Логико-лингвистическое описание систем. Нечеткие модели.

Логико-лингвистические методы описания систем основаны на том, что поведение исследуемой системы описывается на естественном (или близком к естественному) языке в терминах лингвистических переменных.

Входные и выходные параметры системы рассматриваются как лингвистические переменные, а качественное описание процесса задается совокупностью высказываний следующего вида:

L_1 : если $\langle A_1 \rangle$ то $\langle B_1 \rangle$,

L_2 : если $\langle A_2 \rangle$ то $\langle B_2 \rangle$,

.....

L_k : если $\langle A_k \rangle$ то $\langle B_k \rangle$,

где $\langle A_i \rangle$, $i=1,2,...,k$ - составные нечеткие высказывания, определенные на значениях входных лингвистических переменных, а $\langle B_i \rangle$, $i = 1,2,...,k$ - высказывания, определенные на значениях выходных лингвистических переменных.

Рассмотрим пример решения задачи нечеткого логического управления: построение модели управления паровым котлом.

Модель управления паровым котлом

Прототипом модели послужил паровой двигатель (лабораторный) с двумя входами (подача тепла, открытие дросселя) и двумя выходами (давление в котле, скорость двигателя).

Цель управления: поддержание заданного давления в котле (зависит от подачи тепла) и заданной скорости двигателя (зависит от открытия дросселя). В соответствии с этим, схема системы управления двигателем выглядит следующим образом:

Рассмотрим одну часть задачи - управление давлением.

Входные лингвистические переменные:

РЕ - отклонение давления (разность между текущим и заданным значениями);

СРЕ - скорость изменения отклонения давления.

Выходная лингвистическая переменная:

НС - изменение количества тепла.

Значения лингвистических переменных:

NB - отрицательное большое;

NM- отрицательное среднее;

NS- отрицательное малое;

NO- отрицательное близкое к нулю;

ZO- близкое к нулю;

PO - положительное близкое к нулю;

PS - положительное малое;

PM - положительное среднее;

PB - положительное большое.

Управляющие правила (15 правил), связывающие лингвистические значения входных и выходных переменных, имеют вид: *"Если отклонение давления = A_i и, если скорость отклонения давления = B_i , то изменение*

количества подаваемого тепла равно C_i'' , где A_i, B_i, C_i - перечисленные выше лингвистические значения.

Полный набор правил задавался таблицей:

N	Отклонение давления РЕ	Скорость изменения отклонения давления СРЕ	Изменение количества подаваемого тепла НС
1	NB	NB или NM	PB
2	NB или NM	NS	PM
3	NS	PS или NO	PM
4	NO	PB или PM	PM
5	NO	NB или NM	NM
6	PO или ZO	NO	NO
7	PO	NB или NM	PM
8	PO	PB или PM	NM
9	PS	PS или NO	NM
10	PB или PM	NS	NM
11	PB	NB или NM	NB
12	NO	PS	PS
13	NO	NS	NS
14	PO	PS	PS
15	PO	PS	NS

Нечеткие системы

Под *нечеткой системой* понимают модель с одним или несколькими входами, заданными в виде лингвистических переменных, с одним либо

несколькими выходами (четкими либо нечеткими), функционирующую на базе нечетких правил.

Нечеткие правила, обычно имеют продукционную форму, а их вид зависит от типа модели.

Наиболее распространены в настоящее время модели Мамдани, нечеткие правила в которых имеют следующую форму:

$$R_i: \text{ЕСЛИ } x_1 \text{ есть } A_{i1} \text{ И } x_2 \text{ есть } A_{i2} \text{ И } \dots \text{ И } x_n \text{ есть } A_{in} \text{ ТО } y \text{ есть } B_i \quad (1)$$

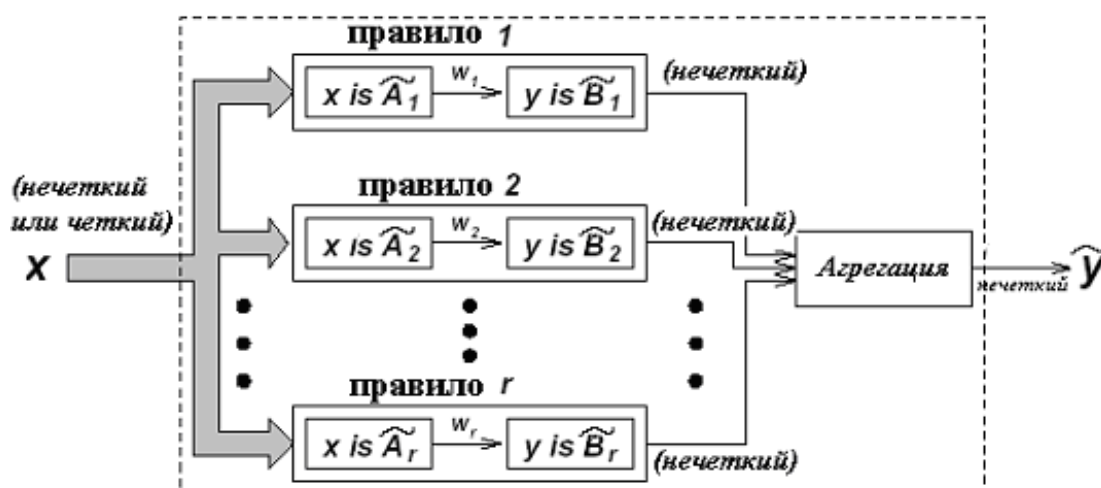
где x_1, \dots, x_n - входные лингвистические переменные, y - выходная лингвистическая переменная, а A_{ij}, B_i - нечеткие переменные, определяющие их значения.

В нечетких моделях Мамдани как на входе, так и на выходе мы имеем информацию, задаваемую значениями лингвистических переменных.

Пример правила в нечеткой модели Мамдани:

ЕСЛИ влажность ВЫСОКАЯ и температура СРЕДНЯЯ ТО установить угол поворота клапана МАЛЕНЬКИМ.

Общая схема нечетких систем Мамдани представлена ниже



Когда на входы нечеткой системы поступают конкретные значения параметров, модель осуществляет нечеткий вывод и формирует непосредственное значение на выходе модели. Пример нечеткого вывода в максиминных моделях Мамдани с дефаззификацией по методу центра тяжести представлен на рисунке 1.

Нечеткий вывод на каждом из правил R_i нечеткой модели здесь осуществляется следующим образом:

Вычисляем степень принадлежности $w_j = \mu_{\tilde{A}_{ij}}(x_j^*)$

Находим $w = \min_j w_j$

Находим нечеткое множество $\tilde{B}_i' = w \wedge \tilde{B}_i$

Результаты нечеткого вывода каждого из нечетких правил объединяются.

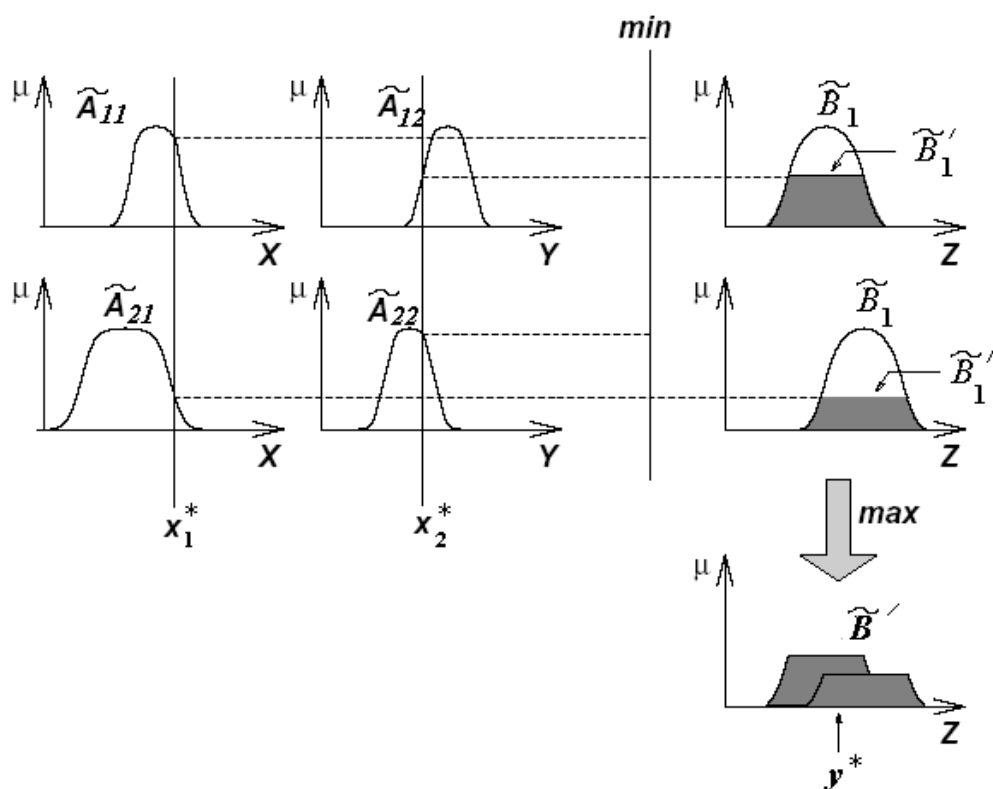


Рис. 1. Пример вывода на нечеткой модели Мамдани

Конструктор нечетких систем CubiCalc

Система CubiCalc является интерактивной оболочкой для проектирования моделей систем нечеткого управления, основанных на нечетких продукционных правилах.

Принимая на вход четкие значения переменных лингвистического характера, она способна обработать их с привлечением нечетких продукционных правил согласно модели Мамдани, и сформировать на выходе системы значения выходных переменных.

Нечеткие производционные правила в системе CubiCalc имеют следующий вид (1).

Примером такого правила может служить следующее

ЕСЛИ давление в резервуаре маленькое И температура воды большая И рост давления маленький И рост температуры небольшой ТО немного повернуть регулятор потока воды.

В данном правиле

1. давление,
2. температура,
3. рост давления,
4. рост температуры,
5. угол поворота регулятора потока воды

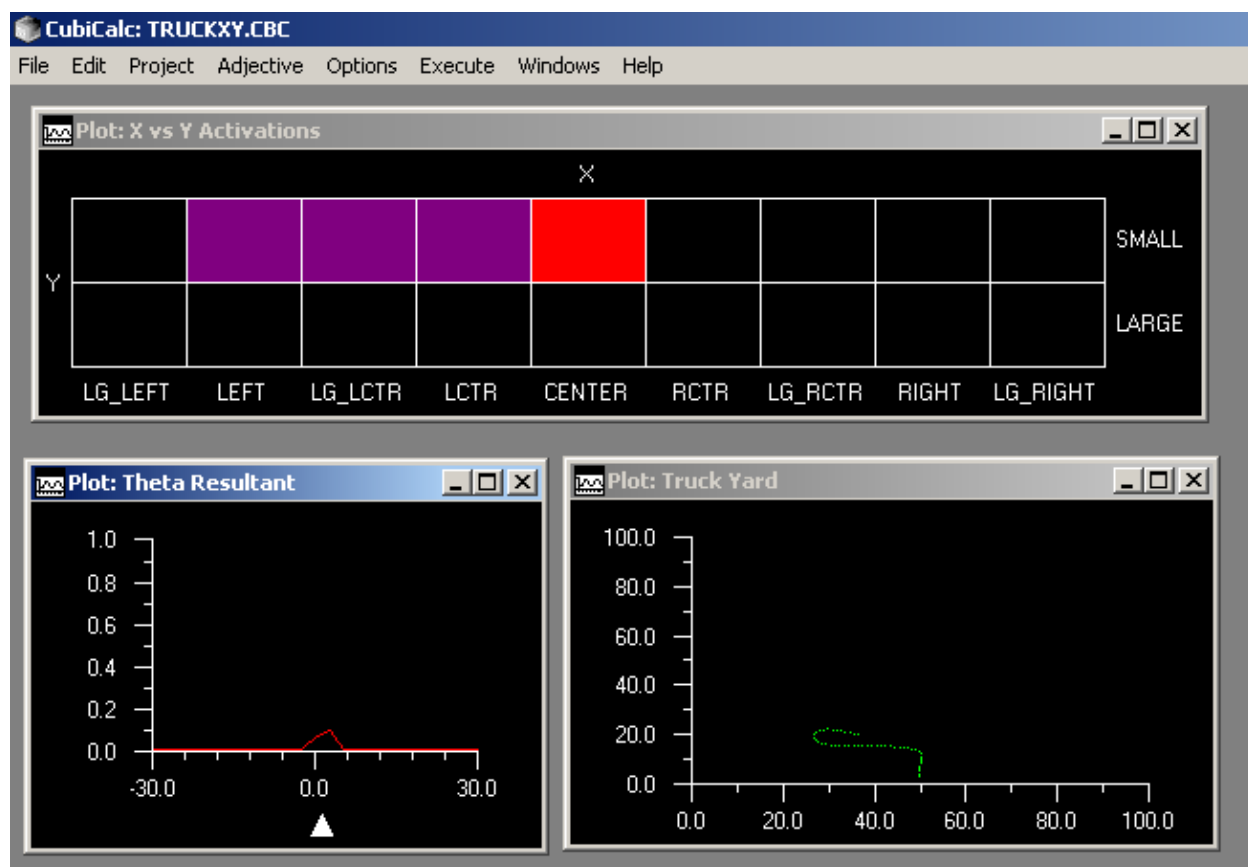
- есть лингвистические переменные, принимающие соответственно следующие значения в виде нечетких переменных –

1. маленькое,
2. большая,
3. маленький,
4. небольшой,
5. немного.

Знакомство с системой CubiCalc на примере модели управления грузовиком TRACKXY

1. Запустить систему CubiCalc и познакомиться с основными функциями меню данной системы.
2. Загрузить демонстрационную программу TRACKXY -модель системы управления грузовиком для въезда его в узкие ворота.
3. В режиме пошагового исполнения (клавиша (F8)) поработать с данной моделью, запустив ее несколько раз, исследовав траекторию движения грузовика (окно Track Yard), активацию нечетких правил (окно X

vs Y activation), результат нечеткого вывода в виде угла нечеткого множества угла поворота руля (окно Theta Resultant) (рис. 2).



Формально модель работы данной системы задается в разделе PROJECT меню и включает в себя следующие основные разделы:

Меню Variables - конструктор входных, выходных лингвистических и временных переменных модели.

Меню Adjectives – конструктор лингвистических переменных модели, позволяющий формировать их значения – нечеткие переменные.

Меню Values – исследование текущих значений определенных в модели переменных.

Меню Rules – конструктор нечетких продукционных правил модели вида (1), согласно которым происходит функционирование системы.

Меню Initialization – раздел инициализации значений переменных модели.

Меню Preprocessing (предобработка) – раздел действий, выполняемых перед каждым циклом отработки нечетких правил.

Меню Postprocessing (постобработка) – раздел действий, выполняемых после каждого цикла отработки нечетких правил.

Меню Simulation (моделирование) – раздел действий, определяющих функционирование нечеткой модели (изменение значений переменных моделей по результатам нечеткого вывода).

Plots – графики, отображающие работу модели.

Вернемся к модели управления грузовиком. Работа данной модели основана на следующих интуитивных соображениях эксперта –

Расстояние грузовика до ворот по Y описывается с помощью двух категорий – БОЛЬШОЕ и МАЛЕНЬКОЕ.

Если расстояние БОЛЬШОЕ, то поступаем по обычным правилам модели (в модели они заданы), если МАЛЕНЬКОЕ, то стараемся отогнать грузовик от нижней границы и выгнать на середину площадки.

В модели управления грузовиком кроме расстояния до ворот Y вводятся, также следующие переменные –

Эти переменные

Значения лингвистической переменной Phi (ориентация грузовика):

VL0 – Намного левее от нулевого угла.

L0 – Левее от нулевого угла

M0 – Более-менее нулевой угол.

R0 – Правее от нулевого угла.

VR0 – Намного правее от нулевого угла.

VL90 – Намного левее 90 градусов

L90 – Левее 90 градусов

M90 – Более менее 90 градусов

R90 – Правее 90 градусов

VR90 – Намного правее 90 градусов

VL180 – Намного левее 180 градусов

L180 – Левее 180 градусов

M180 – Более-менее 180 градусов

R180 – Правее 180 градусов

VR180 – Намного правее 180 градусов

Значения лингвистической переменной Phi45 (ориентация грузовика по отношению к 45 градусам):

VL45 – Намного правее 45 градусов

L45 – Левее 45 градусов

M45 – Более – менее 45 градусов

R45 – Правее 45 градусов

VR45 – Намного правее 45 градусов

Значения лингвистической переменной Phi135 (ориентация грузовика по отношению к 135 градусам):

VL135 – Намного левее 135 градусов

L135 - Левее 135 градусов

M135 – Более-менее 135 градусов

R135 – Правее 135 градусов

VR135 – Намного правее 135 градусов

Значение лингвистической переменной X (горизонтальная позиция):

LG_LEFT – Намного левее от центра

LEFT – Левее центра

LG_LCTR – Близко к центру слева

LCTR – Очень близко к центру слева

CENTER – Более-менее в центре

RCTR – Очень близко к центру справа

LG_RCTR – Близко к центру справа

RIGHT – Правее центра

LG_RIGHT – Намного правее справа от центра

Значение лингвистической переменной Theta (Поворот руля):

NB – Намного против часовой стрелки

NM – Средне против часовой стрелки

NS – Немного против часовой стрелки

ZE – Нулевой поворот

PS – Немного по часовой стрелке

PM – Средне по часовой стрелке

PB – Намного по часовой стрелке

3. Вызвать пункт меню Project -> Variables, изучить все лингвистические переменные модели TRACKXY, изучить их семантику и ответить на следующие вопросы :

3.1. Сколько переменных, и какие присутствуют в разработанной модели?

3.2. Какие типы переменных поддерживает система CubiCalc?

3.3. Какие переменные в модели TRACKXY являются входными, какие выходными, какие временными?

3.4. Что понимается в модели CubiCalc под понятием «Повернуть руль намного против часовой стрелки»? «Находиться очень близко к центру слева»? Дайте естественно языковую интерпретацию их значениям.

3.5. Для каждой их нечетких переменных модели проинтерпретировать два произвольных их значения (их функции принадлежности). Проинтерпретировать семантику.

4. Войти в раздел Project -> Rules и изучить правила, по которым функционирует модель TRUCKXY. Ответить на следующие вопросы:

4.1. Сколько правил включает нечеткая модель системы?

4.2. Переведите на естественный язык 10 любых правил данной системы и дайте их естественно-языковую интерпретацию

5. Изучите действия, которые выполняются на фазе инициализации системы?

6. Во вкладке Simulation изучите действия, по которым моделируется поведение системы на каждой итерации.

7. Во вкладке Plots изучите графики, которые отражают результаты работы системы. Какие типы графиков доступны для создания?

8. Запустить модель на выполнение и посмотреть результаты работы модели.

9. Попытайтесь изменить функционирование модели – например, увеличьте скорость автомобиля, измените значения лингвистических переменных (например, переопределите ряд значений лингвистической переменной X). После этого заново запустить модель и изучить, насколько корректно она функционирует, будет ли грузовик въезжать в ворота в данном случае. Показать результаты модифицированной модели преподавателю.

Задание на лабораторную работу

Задача: имеется некая техническая система, на вход которой подается информация с двух датчиков – датчика температуры (пределы изменения 0 – 100° С) и давления (пределы изменения 100 – 1000МПа).

Назначение системы – управление вентилем подачи пара согласно следующему набору правил.

Вентиль может быть повернут влево или вправо максимум на 90 градусов (влево - отрицательный угол, то есть пределы изменения угла поворота: [-90, 90])

Набор правил:

Если температура маленькая и давление маленькое, то повернуть вентиль очень сильно влево.

Если температура маленькая и давление среднее, то повернуть вентиль сильно влево.

Если температура маленькая и давление большое, то повернуть вентиль немного влево.

Если температура средняя и давление маленькое, то повернуть вентиль немного влево.

Если температура средняя и давление среднее, то повернуть вентиль в нейтральное положение.

Если температура средняя и давление большое, то повернуть вентиль немного вправо.

Если температура большая и давление малое, то повернуть вентиль немного вправо.

Если температура большая и давление среднее, то повернуть вентиль сильно вправо.

Если температура большая и давление большое, то повернуть вентиль очень сильно вправо.

Модель изменения температуры и давления (simulation) после выполнения каждого цикла имеет следующий вид:

1. Температура = температура - угол поворота вентиль/4+uniform()*10-uniform()*10+Давление/100.

2. Давление = давление - угол поворота вентиль/4+uniform()*10-uniform()*10+Температура/10.

3. В 5 случаях из 100 после выполнения каждого цикла температура поднимается вверх на 5 градусов.

4. В 5 случаях из 100 после выполнения каждого цикла давление поднимается на 25 МПа.

Где $\text{uniform}()$ – случайное число от 0 до 1.

Перед первым запуском, проинициализировать данные переменные следующим образом:

Температура = 50

Давление = 600

Описать систему, функционирующую по данным правилам и отображающую диаграмму срабатывания правил, графики изменения значений переменных давления и температуры с течением времени.

Поработать с созданной моделью. Что Вы можете сказать по поводу ее устойчивости? Добиться устойчивой работы модели (без выходов параметры за предельные для них границы) в течение длительного времени.

Лабораторная работа №3

Реализация нечеткой модели при задании правил в нормализованной форме

В системе CubiCalc возможно задание правил не в альтернативной форме, а в нормализованной. В данном случае указывать селектор SYNTAX_ALTERNATE в блоке правил не нужно.

При формировании нечетких правил в нормализованной форме, их форма записи является более расширенной, чем в альтернативной. В нормализованной форме правила функционирования модели нечеткой системы записываются следующим образом:

(Вес правила) IF Условие THEN заключение

В условии перечисляются перечень условий вида Лингвистическая переменная Is значение, объединенные связками И (AND), ИЛИ (OR), НЕ (NOT). Вместо AND может использоваться знак &, вместо OR знак |, вместо NOT знак !.

Вес правила определяет степень его универсальности (достоверности).

К значениям переменных в условиях могут применяться модификаторы ОЧЕНЬ (VERY) и немного (SOMEWHAT).

Пример правила –

(0.7) IF X is Large AND (Y is Small OR U is Negative) THEN Z is Large

естественно-языковая интерпретация которого выглядит следующим образом: «С достоверностью 0.7, если X является большим и (Y малое или U отрицательное), то Z является большим». Где X, Y, U – лингвистические переменные, а большое, маленькое, отрицательное – их значения.

Задание на лабораторную работу

Задание: входами технической системы является информация с трех датчиков – яркость света (1-1000Лк), температура воды (0-60), давление (100-

1000МПа). Назначение системы – управление углом поворота вентилей [-90;90] согласно следующему набору правил.

Набор правил:

(Вес 1) ЕСЛИ освещенность малая И температура малая И давление малое ТО повернуть вентиль сильно влево.

(Вес 0.8) ЕСЛИ освещенность малая И температура малая И давление среднее ТО повернуть вентиль сильно влево.

(Вес 0.6) ЕСЛИ освещенность малая И температура малая И давление большое ТО повернуть вентиль достаточно влево.

(Вес 1) ЕСЛИ освещенность малая И температура средняя И давление малое ТО повернуть вентиль сильно влево.

(Вес 0.3) ЕСЛИ освещенность малая И температура средняя И давление среднее ТО повернуть вентиль в нейтральное положение.

(Вес 0.9) ЕСЛИ освещенность малая И температура средняя И давление большое ТО повернуть вентиль достаточно вправо.

(Вес 0.8) ЕСЛИ освещенность малая И температура большая И давление малое ТО повернуть вентиль достаточно вправо.

(Вес 1) ЕСЛИ освещенность малая И температура большая И давление среднее ТО повернуть вентиль сильно вправо.

(Вес 1) ЕСЛИ освещенность малая И температура немного (SOMEWHAT) большая И давление немного (SOMEWHAT) большое ТО повернуть вентиль сильно вправо.

(Вес 1) ЕСЛИ освещенность большая И (Температура очень (VERY) малая ИЛИ давление ОЧЕНЬ (VERY) малое) ТО повернуть вентиль сильно влево.

(Вес 0.6) ЕСЛИ освещенность большая И температура средняя И давление среднее ТО повернуть вентиль в нейтральное положение.

(Вес 0.8) ЕСЛИ температура большая И давление большое ТО повернуть вентиль достаточно вправо.

Таким образом, техническая система имеет 3 входных лингвистических переменных:

Яркость со значениями:

Малая;

Большая.

Температура со значениями

Малая;

Средняя;

Большая.

Давление со значениями

Малое;

Среднее;

Высокое

И одну выходную лингвистическую переменную – угол поворота вентиля с базовым терм-множеством значений:

Сильно влево;

Достаточно влево;

Нейтральное положение;

Достаточно вправо;

Сильно вправо.

Остальные значения лингвистических переменных образуются от элементов базового терм-множества с помощью модификаторов ОЧЕНЬ (VERY), НЕМНОГО (SOMEWHAT), НЕ (NOT)

Модель изменения яркости, температуры и давления (simulation) после выполнения каждого цикла имеет следующий вид:

1. Температура = температура - угол поворота вентиля/4+uniform()*10-uniform()*10+Давление/200+Яркость/200.

2. Давление = давление - угол поворота вентиля/4+uniform()*10-uniform()*10+Температура/10.

3. В 5 случаях из 100 после выполнения каждого цикла температура поднимается вверх на 5 градусов.

4. В 5 случаях из 100 после выполнения каждого цикла давление поднимается на 5 МПа.

5. Яркость = Яркость + uniform()-uniform()); ЕСЛИ Яркость > 999 ТО Яркость = 999; ЕСЛИ Яркость < 1 ТО Яркость = 1;

Где uniform () – случайное число от 0 до 1.

Перед первым запуском, проинициализировать данные переменные следующим образом:

Температура = 50

Давление = 600

Яркость = 500;

Создать модель в системе CubiCalc, функционирующую по выше описанным законам и отображающую диаграмму срабатывания правил, графики изменения значений переменных давления и температуры с течением времени.. Исследовать работу модели и сделать вывод по устойчивости ее работы. Добиться того, чтобы модель работала устойчиво с течением длительного времени.

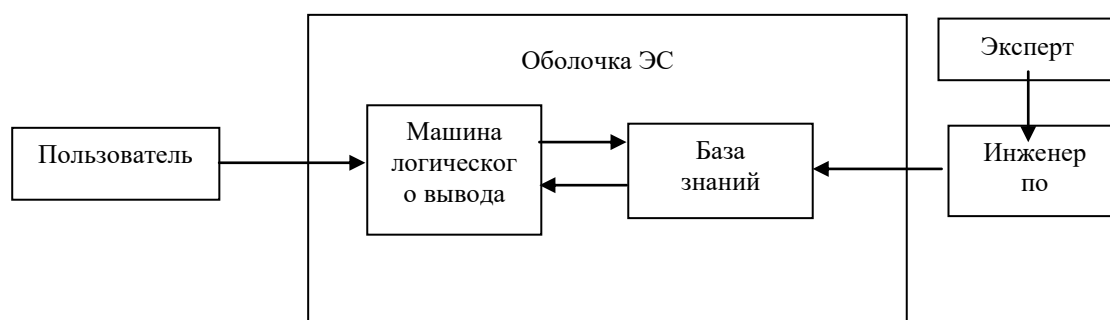
Лабораторная работа №4

Разработка базы знаний и машины вывода прототипа ЭИС

Экспертные системы (ЭС) являются одним из наиболее распространенных типов интеллектуальных информационных систем, применяемых в настоящее время в различных предметных областях.

В базы знаний (БЗ) экспертных систем закладываются знания людей-экспертов о некоторой проблемной области, в которой предполагается использование системы. Использование экспертных систем наиболее актуально в задачах принятия решений и прогнозирования ситуаций. Цель использования экспертных систем – не заменить человека-эксперта при решении некоторой проблемы, а помочь ему в осуществлении этого, отбросить заведомо ложные пути решения, посоветовать по его мнению лучший путь. В любом случае, окончательное решение должна принимать не экспертная система, а человек на основе сформированных экспертной системой рекомендаций.

Основными компонентами любой экспертной системы являются: база знаний и машина логического вывода. Взаимодействие между ними обеспечивается программой, которую принято называть оболочкой экспертной системы. Обобщенная структура любой экспертной системы может быть представлена следующим образом.



База знаний экспертной системы хранит знания о предметной области, в которой осуществляется решение задачи. Эти знания могут формулироваться экспертами в предметной области, извлекаться из литературы, из научных источников и т.д. Основными составляющими

элементами баз знаний большинства экспертных систем являются факты (декларативные знания о предметной области) и правила (процедурные знания о предметной области), описывающие алгоритмы действий экспертной системы в определенных ситуациях.

Машина логического вывода определяет внутренние механизмы функционирования экспертной системы, внутренние механизмы логического вывода на правилах. Машина логического вывода позволяет пользователю получить информацию и о том, как было получено некоторое решение.

2. Основные элементы программирования в ЭС CLIPS

Программирование для большинства экспертных систем подразумевает управление базой знаний (внесение и удаление фактов и правил), а также механизмами логического вывода (управление работой правил, способами умозаключений и т.д.).

При работе из командной строки в системе CLIPS, стандартным приглашением системы является строка CLIPS>. Все команды языка заключаются в скобки ().

Для добавления фактов о предметной области в CLIPS используется команда `assert`. Например, (`assert (today-is sunny)`).

В базу может добавляться сразу несколько фактов (`assert (факт-1) (факт-2) ... (факт N)`).

Для удаления фактов из базы используется команда (`retract N1 N2 ... Nm`), где `N1`, `N2`, ..., `Nm` – номера фактов в базе знаний.

Для просмотра всей базы фактов, которые были заложены в систему, используется команда (`facts`). Для удаления всех фактов из базы, используются команды (`reset`) и (`clear`). Команда (`clear`) очищает всю базу знаний.

Факты в базу можно включать не по одиночке, а целым списком по команде `deffacts`.

(`deffacts имя`
“описание”

(факт-1)

...

(факт-n)

)

После этого, все факты занесутся в базу по команде (reset).

Например,

(*deffacts h*

(hunter Dick)

(hunter Nick)

(hunter Bob)

)

(*reset*)

По команде (undeffacts имя) с последующим (reset), факты удаляются из памяти.

Правила в CLIPS имеют вид продукций, то есть могут быть записаны в виде

ЕСЛИ перечень условий ТО перечень действий

Синтаксис правил в языке CLIPS следующий:

(defrule имя-правила

“необязательный комментарий” ; необязательный

(salience N) ; необязательный

(предпосылка_1)

.....

(предпосылка_m)

=>

(действие_1)

(действие_n)

)

Пример:

(*defrule fun*

“Better things to do on Sunday”

(salience 100)

(today-is Sunday)

(weather-is warm)

=>

(assert (drink cold_water))

(assert (walk in_park))

)

Переменные в CLIPS записываются следующим образом: ?<имя переменной>. Например, ?color.

(defrule make-rain

(today-weather ?weather)

=>

(assert (weather-is ?weather))

)

Если при запуске программы будет найден факт *(today-weather rainy)*, то выполненное правило внесет в базу знаний новый факт *(weather-is rainy)*.

Запуск программы на CLIPS осуществляется по команде *(run)*.

Вывод информации на экран из системы CLIPS осуществляется посредством команды *printout t “<текст>” crlf*. Здесь, *t* указывает направление вывода на экран, а *crlf* – возврат каретки (можно опустить).

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Выполнить перечисленные ниже задания и ответить на вопросы. Результат исполнения каждого задания показать преподавателю.

Задание 1. Основы работы с фактами

Запустить систему CLIPS и вывести на экран диалоговое окно фактов данной системы, отражающее текущее наполнение фактами базы знаний.

Посредством команды *assert* внести в систему следующие факты: 1) *today-credit-rating high* 2) *yesterday-credit-rating low* 3) *tomorrow-credit-rating*

high. Обратить внимание на факты, отображающиеся в окне фактов. Какой номер имеет факт yesterday-credit-rating low?

Вывести на экран список всех фактов.

Удалить из базы фактов yesterday-credit-rating low. Какие факты остались в системе? Какой номер у факта today-credit-rating high? Какой номер у факта tomorrow-credit-rating high?

Дать команду на удаление факта с номером 1. Какая ошибка при этом возникла? Объяснить ошибку?

Удалить факт с номером 0. Внести в систему факт today-credit-rating low. Какой номер имеет данный факт? Удалить все факты из базы командой (clear). С какого номера начнут заноситься следующие факты? Дать команду (reset). С какого номера начнут теперь заноситься следующие факты? Что за факт добавился в систему после команды reset?

Задание 2. Работа с фактами

С помощью команды deffacts внести в базу следующие факты: 1) today-is rainy 2) yesterday-was rainy 3) tomorrow-will sunny.

Вывести список фактов, существующих в базе знаний на экран. Какой номер имеет каждый из внесенных фактов?

Удалить из базы факт yesterday-was rain.

С помощью команды undeffacts удалить из базы все внесенные факты. Дать команду (facts). Убедиться, что в базе не осталось ни одного факта. Если факты остались, то объяснить, где Вы допустили ошибку и исправить ее.

Очистить базу фактов системы.

Задание 3. Работа с типами данных FuzzyCLIPS.

Внести в FuzzyCLIPS факты о том, что 1) число num имеет значение 19; 2) Строка z имеет значение "big" 3) Символьная строка s имеет значение BAD 4) Координаты точки в четырехмерном пространстве равны 10,2,4,3.

Очистить базу фактов.

Задание 4. Работа с правилами. Разработать систему моделирования действий пешехода на светофоре, функционирующую по правилам: 1. Если

цвет зеленый, то добавить в базу факт о том, что пешеход идет; 2. Если цвет желтый, то добавить факт о том, что пешеход готовится идти; 3. Если цвет красный, то добавить в базу факт о том, что пешеход стоит. Проверить работу системы при различных вариантах цвета светофора.

Задание 5. Работа с переменными и с адресами фактов. Внести в базу знаний системы множество фактов о величине прибыли людей. Факты должны иметь вид ВЕЛИЧИНА-ПРИБЫЛИ ИМЯ-ЧЕЛОВЕКА. Величину прибыли выразить в качественном виде (Высокая (HIGH), низкая (LOW), средняя (MIDDLE)). Внести в базу знаний системы правило, вносящее в базу знаний факты вида (RICH ИМЯ-ЧЕЛОВЕКА) для всех людей, прибыль которых является высокой. Запустить программу и проверить результат ее выполнения, а также выводящее имена данных людей на экран.

РАЗДЕЛ 2

Теоретический материал

Принятие решения о выдаче клиенту банковского кредита можно реализовать согласно следующим 8 правилам.

1. ЕСЛИ клиент имеет постоянную работу

И клиент имеет адекватный доход.

И имущество приемлемое

И клиент имеет хороший кредитный рейтинг

И величина кредита меньше 80% стоимости имущества

И клиент имеет 20% стоимости имущества в кассе

ТО одобрить кредит

2. ЕСЛИ клиент имеет адекватные активы

И клиент имеет адекватный доход

И имущество приемлемо

И клиент имеет хороший кредитный рейтинг

И величина кредита меньше 80% стоимости имущества

И клиент имеет 20% стоимости имущества в кассе

ТО одобрить кредит

3. ЕСЛИ клиент имеет работу

И клиент имел более чем два года эту работу

ТО клиент имеет постоянную работу

4. ЕСЛИ имущество находится в зоне расположения банка

ИЛИ имущество не содержится в запрещенном списке

ТО имущество приемлемо

5. ЕСЛИ доход имеется и активен

ИЛИ доход одиночки активен

ТО клиент имеет адекватный доход

6. ЕСЛИ клиент женат

И оплата закладной меньше 60% чистого дохода семьи

ТО доход семьи адекватен

ЕСЛИ клиент не женат

И оплата закладной меньше 70% чистого дохода клиента

ТО доход одиночки активен

8. ЕСЛИ стоимость имущества клиента больше 10-кратной величины
кредита

ИЛИ ликвидные активы клиента больше 5-кратной величины
кредита

ТО клиент имеет адекватные активы

Видим, что на вход экспертной систем подается следующий набор
фактов

Факт о том, имеет клиент работу или не имеет.

Факт о том, имеет ли клиент работу более 2 лет.

Факт о том, женат клиент или не женат.

Находится ли имущество в зоне расположения банка.

Содержится или не содержится имущество в запрещенном списке.

Имеется ли доход?

Является ли оплата закладной меньше 60% чистого дохода семьи.

Является ли оплата закладной меньше 70% чистого дохода клиента.

Является ли стоимость имущества клиента больше 10-кратной величины кредита?

Являются ли ликвидные активы клиента больше 5-кратной величины кредита?

Является ли величина кредита меньше 80% стоимости имущества.

Является ли кредитный рейтинг хорошим?

Имеет ли клиент 20% стоимости имущества в кассе?

На основе данных фактов экспертная система формирует новые. В качестве окончательных решений, экспертная система выдает рекомендацию об одобрении или отклонении в выдаче кредита.

Дополнительный теоретический материал по CLIPS

При описании правил на языке CLIPS возможно объединение посылок путем использования логических связок И, ИЛИ, НЕ.

Формат использования связок в правиле следующий:

(СВЯЗКА (ПОСЫЛКА_1) (ПОСЫЛКА_2))

Например:

(*defrule r*
(*or (temperature high) (pressure low)*)

=>

(*assert (.....)*)

)

Логическая связка И записывается как *and*, ИЛИ как *or*, НЕ как *not*.

Задание на лабораторную работу

Разработать экспертную систему на языке CLIPS, функционирующую по выше приведенным правилам.

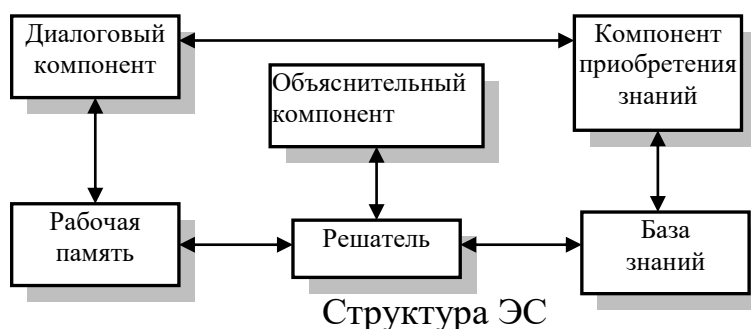
Для облегчения процесса разработки, рекомендуется правила и факты оформлять в виде двух текстовых файлов, которые будут считываться в CLIPS. При задании совокупности фактов использовать команду *deffacts*.

Лабораторная работа №5

Разработка экспертной интеллектуальной системы

Под экспертной системой (ЭС) понимают набор программ, выполняющий функции эксперта при решении задач из некоторой предметной области. ЭС выдают советы, проводят анализ, дают консультации, ставят диагноз.

Структура традиционной статической ЭС включает следующие основные компоненты: решатель (интерпретатор), рабочую память, базу знаний, компонент приобретения знаний, объяснительный компонент, диалоговый компонент.



В базе знаний содержатся факты, на основе которых производится выработка решения. Решатель – алгоритм, программа, набор правил, по которым осуществляется решение задачи. Процесс рассуждений реализуется на основе базы знаний и рабочей памяти. Решатель выполняет две функции: во-первых, просмотр существующих фактов из рабочей памяти и правил из базы знаний и добавление (по мере возможности) в рабочую память новых фактов и, во-вторых, определение порядка просмотра и применения правил. На диалоговый компонент возложена задача ведения диалога о решаемой задаче на языке пользователя (эксперта). Компонент приобретения знаний как программный модуль может в ЭС отсутствовать. Его задача – приобретать в ходе диалога новые знания. Наличие объяснительного компонента дает ЭС способность при решении задачи следовать линии

рассуждений, понятной пользователю (эксперту), и объяснять ход рассуждений.

Задания, порядок выполнения работы

При выполнении лабораторной работы используется программа Mini Expert System, описание работы с которой приведено ниже.

Общее задание на лабораторную работу:

1) В соответствии с вариантом составить список вопросов, необходимых для получения определенного решения и список вариантов ответов. Списки оформить в виде текстового файла с расширением .DAT (см. примеры из программы Mini Expert System).

2) Отладить экспертную систему, проверить ее работоспособность на примерах. При недостаточной точности выводов ЭС увеличить количество вопросов и уточнить вероятностные характеристики исходов-ответов.

3) По результатам работы происходит устная защита следующего содержания:

- задание,
- списки вопросов и вариантов ответов (распечатка .DAT-файла),
- протоколы проверки работоспособности на примерах.

Варианты заданий

Вариант 1. Идентификация типа транспортного средства (велосипед, мотоцикл, автобус, грузовик, легковая).

Вариант 2. Распознавание геометрических примитивов (окружность, эллипс, квадрат, треугольник, прямоугольник, параллелограмм, трапеция).

Вариант 3. Выбор принтера для покупки (матричного, струйного, лазерного).

Вариант 4. Идентификация животного (собака, гусь, рыба, слон, крокодил).

Вариант 5. Идентификация мебели (стул, стол, кровать, шкаф, диван).

Вариант 6. Идентификация овощей (огурцы, томаты, лук, кабачок, капуста).

Вариант 7. Идентификация фруктов (яблоко, апельсин, лимон, банан, груша).

Вариант 8. Идентификация ягод (малина, смородина, клубника, вишня, арбуз).

Вариант 9. Идентификация косметики (помада, тушь, тональный крем, пудра, тени).

Вариант 10. Распознавание геометрических тел (сфера, параллелепипед, куб, пирамида, конус).

Вариант 11. Идентификация моющих средств (шампунь, стиральный порошок, мыло, гель для душа, средство для мытья посуды).

Вариант 12. Выбор мобильного устройства для покупки (КПК, смартфон, ноутбук, мобильный телефон).

Вариант 13. Выбор бытовой техники для покупки (MP3-плеер, телевизор, аудио магнитола, музыкальный центр, домашний кинотеатр).

Вариант 14. Выбор бытовой техники для покупки (тостер, микроволновая печь, блендер, миксер, кухонный комбайн).

Вариант 15. Идентификация столовых приборов (вилка, ложка, стакан, рюмка, блюдец, тарелка).

Подготовка базы знаний

Программа Mini Expert System представляет собой простую экспертную систему.

На первом этапе создания базы знаний необходимо сформулировать знания о рассматриваемой области в виде двух наборов: $Q = \{q_j\}$ – набор вопросов (симптомов, свидетельств) и $V = \{v_i\}$ – набор вариантов исхода (вариантов решения), а также двух матриц вероятностей: $P_y = \{p_{yij}\}$ и $P_n = \{p_{nij}\}$ размером $m \times n$, где p_{yij} – вероятность получения положительного ответа на j -й вопрос, если i -й исход верен, p_{nij} – вероятность получения отрицательного ответа на j -й вопрос, если i -й исход верен, n и m – количества вопросов и исходов соответственно. Кроме того, каждому исходу ставится в

соответствие *априорная вероятность* данного исхода P , т.е. *вероятность исхода в случае отсутствия дополнительной информации*.

Исходная информация оформляется в виде текстового файла с расширением .DAT со следующей структурой:

Описание базы знаний, имя автора, комментариев и т.д.

(можно в несколько строк; эта информация выводится после загрузки базы знаний; данная секция заканчивается после первой пустой строки)

Вопрос № 0 (любой текст, заканчивающийся переносом строки)

Вопрос № 1

Вопрос № 2

...

Вопрос № N (после последнего вопроса следует одна пустая строка, и вторая секция заканчивается)

Исход № 0, $P [, i, P_y, P_n]$

Исход № 1, $P [, i, P_y, P_n]$

Исход № 2, $P [, i, P_y, P_n]$

...

Исход № M, $P [, i, P_y, P_n]$

В начале описания правила вывода задаётся исход, вероятность которого меняется в соответствии с данным правилом. Это текст, включающий любые символы, кроме запятых. После запятой указывается априорная вероятность данного исхода P . После этого через запятую идёт ряд повторяющихся полей из трёх элементов. Первый элемент i – номер соответствующего вопроса. Следующие два элемента P_{yij} и P_{nij} – соответственно вероятности получения ответа «Да» на этот вопрос, если

возможный исход верен и неверен. Эти данные указываются для каждого вопроса, связанного с данным исходом.

Примечание: $P \leq 0.00001$ считается равной нулю, а $P \geq 0.99999$ – единице, поэтому не следует указывать такие значения – исход с подобной априорной вероятностью обрабатываться не будет.

Например:

Грипп, 0.01, 1,0.9,0.01, 2,1,0.01, 3,0,0.01

Здесь сказано: существует априорная вероятность $P = 0,01$ того, что любой наугад взятый человек болеет гриппом.

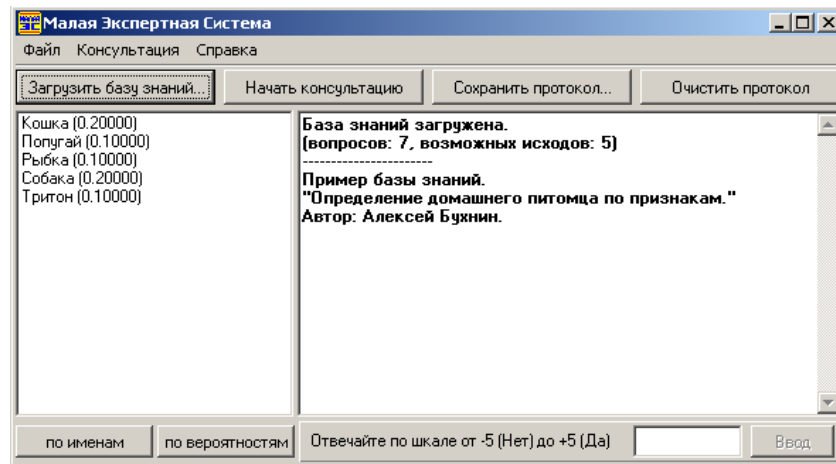
Первому вопросу ($i = 1$) соответствует запись «1,0.9,0.01». Отсюда следуют значения $P_{y_{i1}} = 0,9$ и $P_{n_{i1}} = 0,01$, которые означают, что если у пациента грипп, то он в девяти случаях из десяти ответит «Да» на этот вопрос, а если у него нет гриппа, он ответит «Да» лишь в одном случае из ста (т.е. данный симптом встречается довольно редко при других болезнях). Ответ «Да» подтверждает гипотезу о том, что у него грипп. Ответ «Нет» позволяет предположить, что человек гриппом не болеет.

Для второго вопроса имеем запись «2,1,0.01». То есть, если у человека грипп, то этот симптом обязательно должен присутствовать ($P_{y_{i2}} = 1$) и он обязательно ответит «Да». Соответствующий симптом может иметь место и при отсутствии гриппа ($P_{n_{i2}} = 0,01$), но это маловероятно.

Примечание: При большом количестве вопросов нет необходимости в каждой строке последней секции перечислять их все, тем более, если ответ на какой-либо вопрос не влияет на вероятность данного исхода.

Работа с программой Mini Expert System.

Запускающим файлом программы является MiniES.exe. После запуска появляется диалоговое окно (см. рисунок 2 на след. странице), кнопки на котором выполняют функции: «Загрузить базу знаний» - загрузка заранее подготовленного .DAT-файла; «Начать консультацию» - запуск решателя, ответы на задаваемые вопросы вводятся в нижнее поле по шкале от -5 (однозначно нет) до 5 (однозначно да).



Диалоговое окно программы Mini Expert System

Технология работы

Откройте текстовый редактор Блокнот.

Наберите текст:

"Определение домашнего питомца по признакам."

Вопросы:

Морда вытянута?

Крылья есть?

На поглаживания по спине отвечает довольным урчанием?

Живёт в аквариуме (или другом резервуаре с водой)?

Есть лапы?

При встрече с хозяином виляет хвостом?

Собака, 0.2, 1,0.7,0.5, 2,0,0.5, 3,0.01,0.5, 4,0,0.5, 6,0.9,0.05

Кошка, 0.2, 1,0.1,0.5, 2,0,0.5, 3,0.95,0, 4,0,0.5

Попугай, 0.1, 2,1,0.3, 4,0,0.5

Рыбка, 0.1, 2,0,0.5, 4,1,0.1, 5,0,0.5

Тритон, 0.1, 2,0,0.5, 4,1,0.2, 5,1,0.5

Сохраните текст с расширением .DAT.

Запустите файл MiniES.exe.

Загрузите подготовленную базу знаний.

Протестируйте работу экспертной системы на данном примере
(загадайте всех перечисленных животных)

Согласно рассмотренному примеру выполнить задание по своему варианту.

Лабораторная работа № 6

Прогнозирующие системы на базе нейросетевых систем

Прогноз курса валюты на будущее

Теоретический материал

Задача прогнозирования временных рядов является одной из классических задач, эффективно решаемых с помощью нейронных сетей. Способность нейронных сетей после обучения к обобщению и пролонгации результатов создает потенциальные предпосылки на их базе различного рода прогнозирующих систем.

Пусть дан временной ряд $x(t)$ на промежутке $t = \overline{1, m}$. Тогда задача прогнозирования состоит в том, чтобы найти продолжение временного ряда на неизвестном промежутке, то есть необходимо определить $x(m+1)$, $x(m+2)$, и т.д. (рис. 1).



Рис. 1. Иллюстрация задачи прогноза с помощью нейронных сетей

Совокупность известных значений временного ряда образует обучающую выборку, размерность которой характеризуется значением m . Для прогнозирования временных рядов используется метод "скользящего окна". Он характеризуется длиной окна p , равной числу элементов ряда, одновременно подаваемых на нейронную сеть. Данный факт определяет структуру нейронной сети, которая состоит из p распределительных нейронов и одного выходного нейрона.

Построенная модель скользящего окна для нейронных сетей с линейной функцией активации соответствует линейной авторегрессии и описывается выражением

$$\overline{x(n)} = \sum_{k=1}^p w_k \cdot x(n - p + k - 1)$$

где $w_k, k = \overline{1, p}$ - весовые коэффициенты нейронной сети; $\overline{x(n)}$ - оценка значения ряда $x(n)$ в момент времени n .

Ошибка прогнозирования определяется выражением $e(n) = x(n) - \overline{x(n)}$.

Модель линейной авторегрессии формирует значение ряда $x(n)$, как взвешенную сумму предыдущих значений ряда. Обучающую выборку нейронной сети можно представить в виде матрицы, строки которой характеризуют векторы, подаваемые на вход нейронной сети:

$$X = \begin{bmatrix} x(1) & x(2) & \dots & x(p) \\ x(2) & x(3) & \dots & x(p+1) \\ \dots & \dots & \dots & \dots \\ x(m-p) & x(m-p+1) & \dots & x(m-1) \end{bmatrix} \quad (1)$$

Это эквивалентно перемещению окна по ряду $x(t)$ с единичным шагом.

Таким образом, для обучения нейронной сети прогнозированию используется выборка известных членов ряда. После обучения сеть должна прогнозировать временной ряд на упреждающий промежуток времени.

Задание 1. Задача прогнозирования курса доллара на день вперед

Рассмотрим задачу прогнозирования курса доллара на день вперед. Пусть имеется база данных, содержащая значения курса доллара за последний месяц. Необходимо построить прогноз цены доллара на завтра, послезавтра и т.д. на основе курсов за последние несколько дней на основе обучения нейронной сети.

Прогнозирующая нейронная сеть должна иметь один выход и количество входов, равное ширине окна p , - сколько предыдущих значений

мы хотим использовать для прогноза, например, четыре последних значения ($p=4$)

Для обучения нейронной сети необходимо составить обучающую выборку в виде (1), в которой входными значениями будут курсы доллара за последних p дней, а желаемых выходом – известный курс в следующий за ними день. Эту выборку можно подготовить в пакете Excel. Подготовка обучающей выборки в данном случае будет состоять из следующих операций:

1. Внести значения курса доллара за последний месяц последовательно в столбец таблицы.
2. Скопировать значения котировок в p соседних столбца.
3. Сдвинуть второй столбец на 1 ячейку вверх, третий столбец на 2 и т.д. В этом случае сформируется обучающая выборка вида (1).

Особенность этой подготовки состоит в том, что каждая строка таблицы теперь представляет собой обучающий пример, где первые p чисел – входные значения сети, а $p+1$ число – желаемое значение выхода. Исключение составляют последние $p-1$ строк, где данных недостаточно, поэтому эти строки необходимо убрать из обучающей выборки. В p строке заданы все p входов, но неизвестно значение выхода. Для этой строки можно спрогнозировать результат. Это будет результатом прогноза на следующий день за m -ым.

Для создания и обучения нейронной сети предлагается использовать пакет Neural Network Wizard.

Работа с пакетом Neural Network Wizard

Neural Network Wizard является пакетом для создания моделей искусственных многослойных нейронных сетей прямого распространения.

Для создания модели нейронной сети необходимо выполнить следующие шаги:

1. Создать файл с обучающей выборкой, в котором должны быть представлены представительные примеры решения задачи (вход-выход). Например, зависимости курса валют на следующий день в зависимости от предыдущих p дней.

2. Выбрать файл с обучающей выборкой в Neural Network Wizard (рис. 2.)

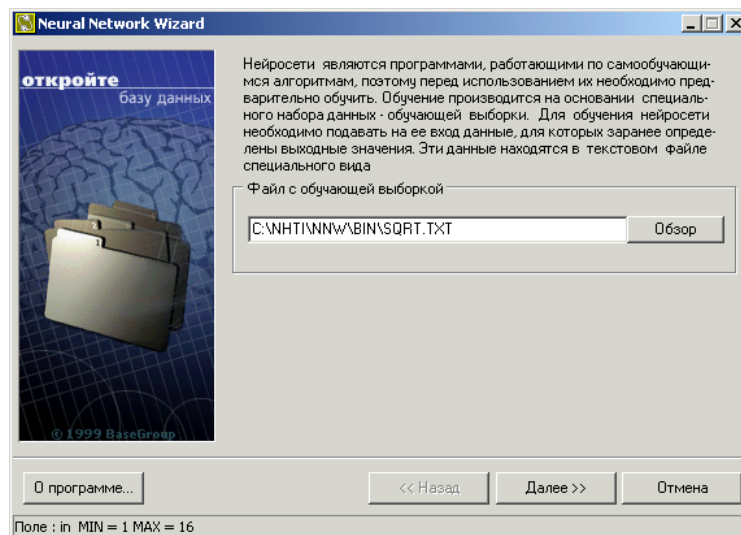


Рис. 2. Выбор файла с обучающей выборкой

3. Задать входные и выходные элементы нейронной сети (рис. 3.)

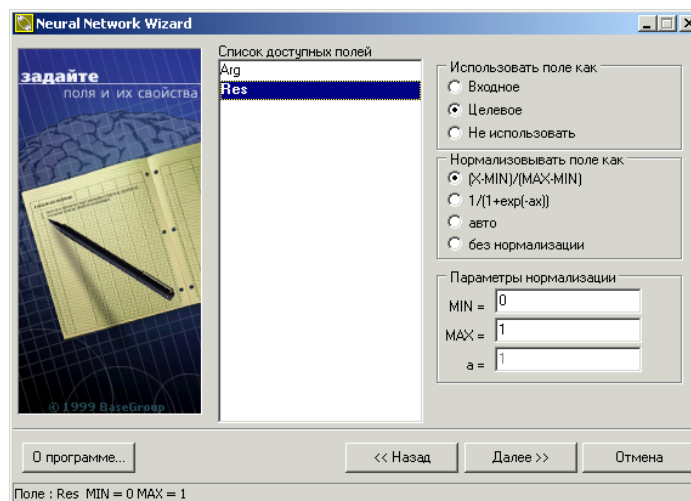


Рис. 3. Определение входов и выходов нейронной сети.

4. Указать количество слоев, количество нейронов в скрытых слоях и вид функции активации (рис. 4).

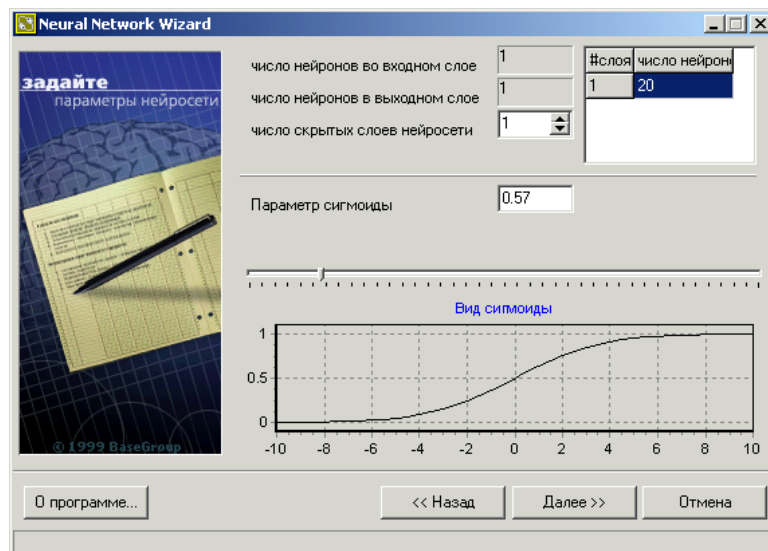


Рис. 4. Выбор количества скрытых слоев, количества нейронов в скрытых слоях и вид функции активации

5. Задать параметры обучения (рис. 5).

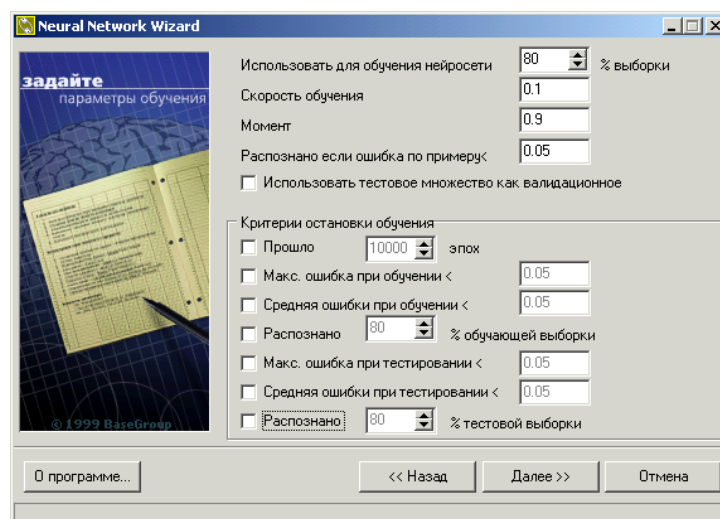


Рис. 5. Параметры обучения нейронной сети.

6. Обучить нейронную сеть.

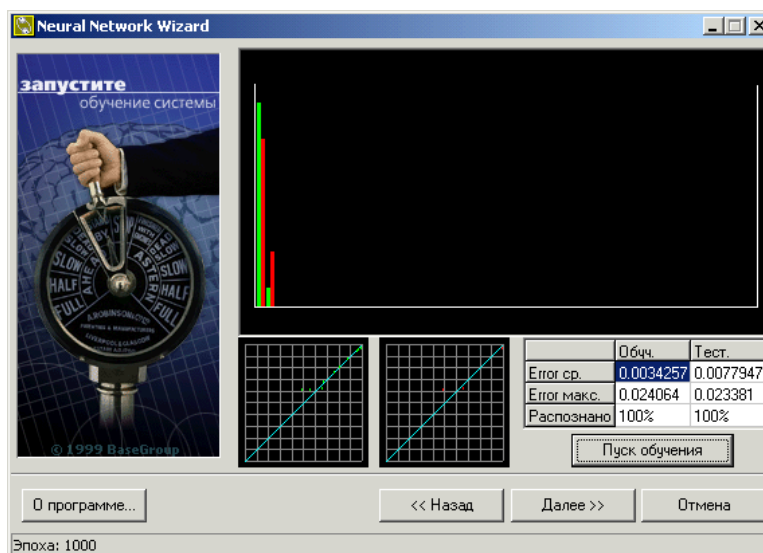


Рис. 6. Обучение нейронной сети

7. Подача на вход нейронной сети необходимых значений и нахождение результата (рис. 7).

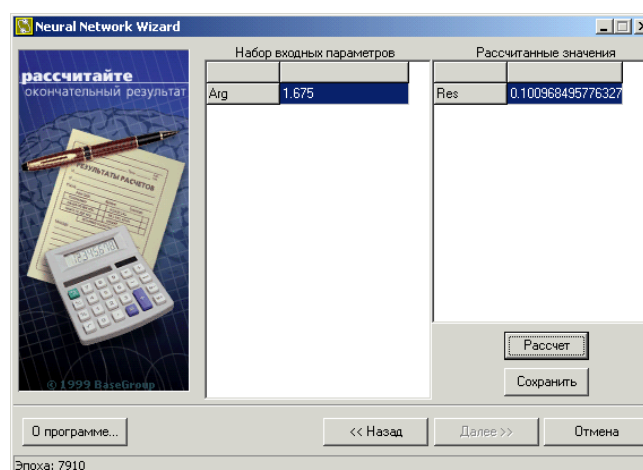


Рис. 7. Работа с нейронной сетью.

Задание на лабораторную работу.

1. Подготовьте обучающую выборку для обучения нейронной сети прогнозирования курса доллара. При подготовке обучающей выборки и данных для опроса можно воспользоваться, например, информацией с сайта www.micex.ru, www.akm.ru/rus, <http://www.inline.ru/>, либо получить информацию о курсе доллара от преподавателя. В качестве размера скользящего окна p взять значение 4.

2. Обучите нейронную сеть в пакете NNW для распознавания курса доллара.
3. Исследуйте корректность работы нейронной сети – задайте 10 примеров с известными курсами доллара за 4 дня и спрогнозируйте курс доллара на следующий день. Исследуйте ошибку прогнозирования. Данные тесты внесите в отчет по лабораторной работе.
4. На базе обученной нейронной сети последовательно спрогнозируйте курс доллара неделю вперед. Какова ошибка данного прогнозирования. Результаты данного исследования внесите в отчет по лабораторной работе.

Задание 2. Задача прогнозирования курса доллара на завтрашний день по известным курсам валют на сегодняшний день.

Обучите нейронную сеть для решения задачи прогнозирования курса доллара на завтрашний день по известным курсам 5 валют на сегодняшний день (<http://www.inline.ru/>). Валюты выберете произвольно.

1. Постройте обучающую выборку с 5 входами и 1 выходом. Входы – курсы 5 выбранных валют сегодня, выход – курс доллара завтра. Обучающая выборка должна содержать не менее 30 записей.
2. Обучите нейронную сеть.
3. Подготовьте данные для опроса нейронной сети, содержащую курсы выбранных 5 валют сегодня, и спрогнозируйте курс доллара на завтра. Результаты опроса внести в отчет по лабораторной работе.

Основная литература:

1. Масленникова О.Е., Гаврилова И.В. Основы искусственного интеллекта [Электронный ресурс]: учеб. пособие/ О.Е. Масленникова, И.В. Гаврилова. — 2-е изд., стер. — М. : ФЛИНТА, 2013. — 282 с. — Режим доступа: <http://ibooks.ru/reading.php?productid=337972>

2. Джонс, М.Т. Программирование искусственного интеллекта в приложениях. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2011. — 312 с. — Режим доступа: <https://e.lanbook.com/reader/book/1244/#1> — Загл. с экрана.

Дополнительная литература:

1. Ездаков А.Л. Экспертные системы САПР: Учеб.пособие. – М.: ИД «ФОРУМ»: ИНФРА-М, 2016. – 160 с.: ил.- (Высшее образование).

2. Гладков, Л. А. Генетические алгоритмы [Электронный ресурс] / Под ред. В. М. Курейчика. - 2-е изд., исправл. и доп. - М.: ФИЗМАТЛИТ, 2010. – 368 с. — Режим доступа: <http://znanium.com/bookread2.php?book=544626#>